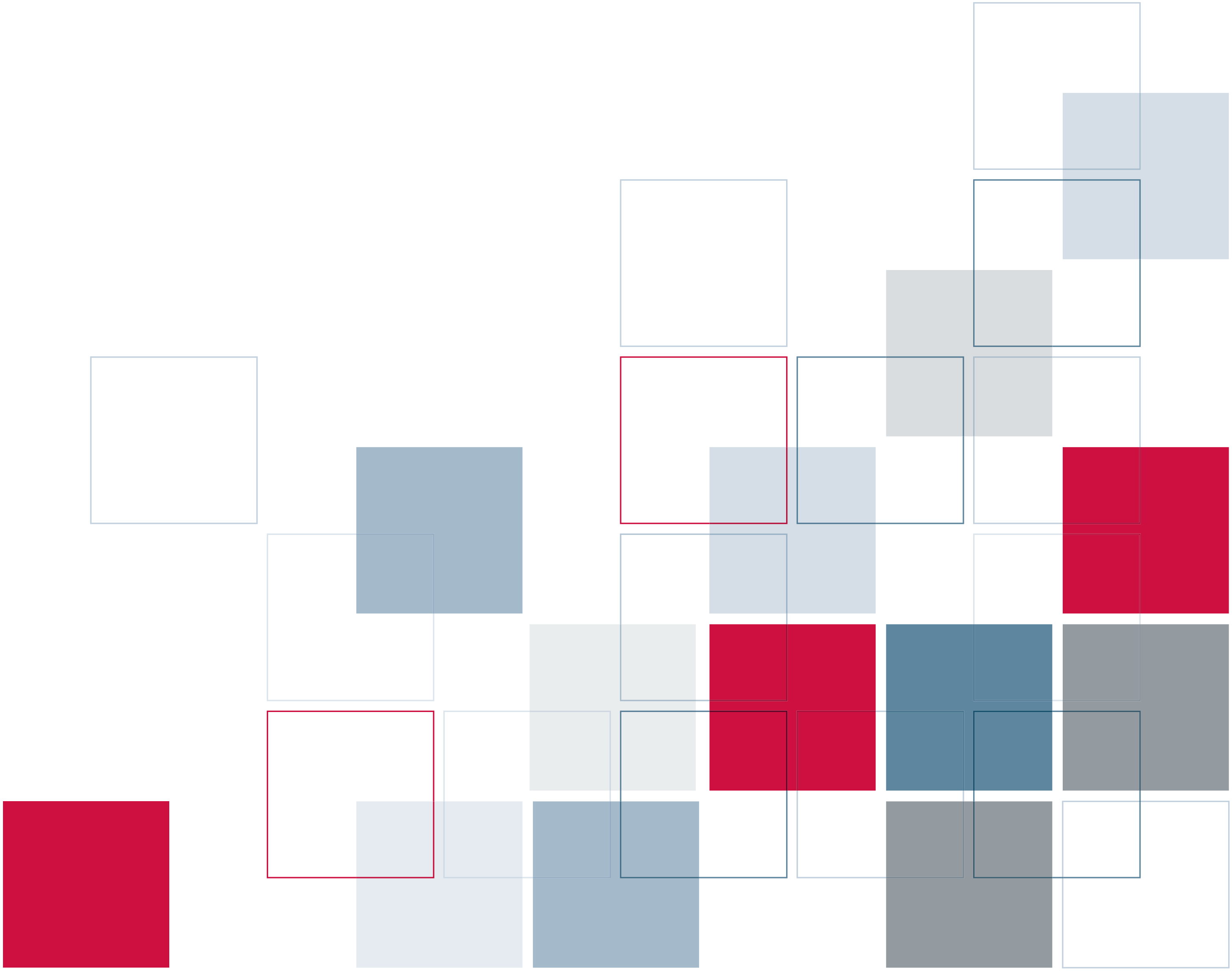
PASW® Statistics 18 Core System User's Guide



For more information about SPSS Inc. software products, please visit our Web site at *http://www.spss.com* or contact

SPSS Inc.
233 South Wacker Drive, 11th Floor Chicago, IL 60606-6412
Tel: (312) 651-3000 Fax: (312) 651-3668

Printed in the United States of America.

ISBN-13: 978-1-56827-406-5
ISBN-10: 1-56827-406-8

1 2 3 4 5 6 7 8 9 0   12 11 10 09

Preface

PASW Statistics 18

PASW Statistics 18 is a comprehensive system for analyzing data. PASW Statistics can take data from almost any type of file and use them to generate tabulated reports, charts and plots of distributions and trends, descriptive statistics, and complex statistical analyses.

This manual, the *PASW® Statistics 18 Core System User's Guide*, documents the graphical user interface of PASW Statistics. Examples using the statistical procedures found in PASW Statistics Base 18 are provided in the Help system, installed with the software.

In addition, beneath the menus and dialog boxes, PASW Statistics uses a command language. Some extended features of the system can be accessed only via command syntax. (Those features are not available in the Student Version.) Detailed command syntax reference information is available in two forms: integrated into the overall Help system and as a separate document in PDF form in the *Command Syntax Reference*, also available from the Help menu.

PASW Statistics Options

The following options are available as add-on enhancements to the full (not Student Version) PASW Statistics Base system:

Statistics Base gives you a wide range of statistical procedures for basic analyses and reports, including counts, crosstabs and descriptive statistics, OLAP Cubes and codebook reports. It also provides a wide variety of dimension reduction, classification and segmentation techniques such as factor analysis, cluster analysis, nearest neighbor analysis and discriminant function analysis. Additionally, PASW Statistics Base offers a broad range of algorithms for comparing means and predictive techniques such as t-test, analysis of variance, linear regression and ordinal regression.

Advanced Statistics focuses on techniques often used in sophisticated experimental and biomedical research. It includes procedures for general linear models (GLM), linear mixed models, variance components analysis, loglinear analysis, ordinal regression, actuarial life tables, Kaplan-Meier survival analysis, and basic and extended Cox regression.

Bootstrapping is a method for deriving robust estimates of standard errors and confidence intervals for estimates such as the mean, median, proportion, odds ratio, correlation coefficient or regression coefficient.

Categories performs optimal scaling procedures, including correspondence analysis.

Complex Samples allows survey, market, health, and public opinion researchers, as well as social scientists who use sample survey methodology, to incorporate their complex sample designs into data analysis.

Conjoint provides a realistic way to measure how individual product attributes affect consumer and citizen preferences. With Conjoint, you can easily measure the trade-off effect of each product attribute in the context of a set of product attributes—as consumers do when making purchasing decisions.

Custom Tables creates a variety of presentation-quality tabular reports, including complex stub-and-banner tables and displays of multiple response data.

Data Preparation provides a quick visual snapshot of your data. It provides the ability to apply validation rules that identify invalid data values. You can create rules that flag out-of-range values, missing values, or blank values. You can also save variables that record individual rule violations and the total number of rule violations per case. A limited set of predefined rules that you can copy or modify is provided.

Decision Trees creates a tree-based classification model. It classifies cases into groups or predicts values of a dependent (target) variable based on values of independent (predictor) variables. The procedure provides validation tools for exploratory and confirmatory classification analysis.

Direct Marketing allows organizations to ensure their marketing programs are as effective as possible, through techniques specifically designed for direct marketing.

Exact Tests calculates exact *p* values for statistical tests when small or very unevenly distributed samples could make the usual tests inaccurate. This option is available only on Windows operating systems.

Forecasting performs comprehensive forecasting and time series analyses with multiple curve-fitting models, smoothing models, and methods for estimating autoregressive functions.

Missing Values describes patterns of missing data, estimates means and other statistics, and imputes values for missing observations.

Neural Networks can be used to make business decisions by forecasting demand for a product as a function of price and other variables, or by categorizing customers based on buying habits and demographic characteristics. Neural networks are non-linear data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

Regression provides techniques for analyzing data that do not fit traditional linear statistical models. It includes procedures for probit analysis, logistic regression, weight estimation, two-stage least-squares regression, and general nonlinear regression.

Amos™ (**a**nalysis of **mo**ment **s**tructures) uses structural equation modeling to confirm and explain conceptual models that involve attitudes, perceptions, and other factors that drive behavior.

Installation

To install the Base system, run the License Authorization Wizard using the authorization code that you received from SPSS Inc. For more information, see the installation instructions supplied with the Base system.

Compatibility

PASW Statistics is designed to run on many computer systems. See the installation instructions that came with your system for specific information on minimum and recommended requirements.

Serial Numbers

Your serial number is your identification number with SPSS Inc. You will need this serial number when you contact SPSS Inc. for information regarding support, payment, or an upgraded system. The serial number was provided with your Core system.

Customer Service

If you have any questions concerning your shipment or account, contact your local office, listed on the Web site at *http://www.spss.com/worldwide*. Please have your serial number ready for identification.

Training Seminars

SPSS Inc. provides both public and onsite training seminars. All seminars feature hands-on workshops. Seminars will be offered in major cities on a regular basis. For more information on these seminars, contact your local office, listed on the Web site at *http://www.spss.com/worldwide*.

Technical Support

Technical Support services are available to maintenance customers. Customers may contact Technical Support for assistance in using PASW Statistics or for installation help for one of the supported hardware environments. To reach Technical Support, see the Web site at *http://www.spss.com*, or contact your local office, listed on the Web site at *http://www.spss.com/worldwide*. Be prepared to identify yourself, your organization, and the serial number of your system.

Additional Publications

The *SPSS Statistics Statistical Procedures Companion*, by Marija Norušis, has been published by Prentice Hall. A new version of this book, updated for PASW Statistics 18, is planned. The *SPSS Statistics Advanced Statistical Procedures Companion*, also based on PASW Statistics 18, is forthcoming. The *SPSS Statistics Guide to Data Analysis* for PASW Statistics 18 is also in development. Announcements of publications available exclusively through Prentice Hall will be available on the Web site at *http://www.spss.com/estore* (select your home country, and then click Books).

Contents

1 Overview 1

2 Getting Help 9

3 Data Files 11

10 Working with Output 205

11 Pivot Tables 228

18 Creating and Managing Custom Dialogs 319

19 Production Jobs 345

20 Output Management System 353

21 Scripting Facility 379

Appendix

Chapter

1

Overview

What's New in Version 18?

Automated data preparation. Automated Data Preparation (ADP) handles the task of preparing data for analysis, analyzing your data and identifying fixes, screening out fields (variables) that are problematic or not likely to be useful, deriving new attributes when appropriate, and improving performance through intelligent screening techniques. You can use the algorithm in fully automatic fashion, allowing it to choose and apply fixes, or you can use it in interactive fashion, previewing the changes before they are made and accept or reject them as desired. Automated Data Preparation is available in the Data Preparation add-on option.

Bootstrapping. Bootstrapping is a robust method for determining the properties of population estimators (like the mean, median, percentiles, and correlation and regression coefficients) when parametric assumptions do not hold, or when inferences based on parametric assumptions are difficult to compute. Bootstrapping is available in the new Bootstrapping add-on option.

New nonparametric tests. Nonparametric tests make minimal assumptions about the underlying distribution of the data. The new nonparametric tests provide a new user interface and Model Viewer output, and include all of the tests available in the legacy nonparametric tests, including: one-sample Wilcoxon signed-rank test, one-sample confidence intervals for the binomial distribution, the related-samples marginal homogeneity test, and the Hodges-Lehman confidence interval for the median of the difference in paired-samples and the difference in medians of two independent samples. Pairwise and stepwise step-down multiple comparisons are also available for all *k* independent samples and *k* related samples tests. The Jonckheere-Terpstra test is available without requiring the Exact Tests add-on option. The new nonparametric tests are available in the Statistics Base add-on option.

Programmability enhancements. The R Integration Plug-in now supports R debugging features. Additionally, you can create pivot tables from R with multiple row and column dimensions and you can nest multiple pivot tables under a common outline heading. R extension commands can be implemented directly from R source code files, bypassing the need to distribute them as R packages. Also, you can bundle together all components of a custom R or Python procedure, allowing end users to easily install the procedure without manually copying files. Complete documentation for the Python and R Integration Plug-ins is now integrated with the Help system.

Direct marketing tools. The new Direct Marketing add-on option provides a set of tools designed to improve the results of direct marketing campaigns by identifying demographic, purchasing, and other characteristics that define various groups of consumers and targeting specific groups to maximize positive response rates.

Custom Tables enhancements. The Custom Tables add-on option now offers computed categories and significance results integrated into the same table as the values being tested.

Improved SAS data file support. You can now write data files in SAS 9 format. For more information, see the topic Saving Data: Data File Types in Chapter 3 on p. 40.

Improved Custom Dialog Builder. The Custom Dialog Builder now has a list box control that supports single or multiple selection. Also, list items for combo box and list box controls can now be dynamically populated with values associated with the variables in a specified target list. In addition, radio buttons can now contain a set of nested controls. For more information, see the topic Creating and Managing Custom Dialogs in Chapter 18 on p. 319.

Improved display of large pivot tables. New display options are now available that make it easier to view and navigate large pivot tables (tables with hundreds or thousands of rows). For more information, see the topic Set Rows to Display in Chapter 11 on p. 238.

Improved Twostep Cluster output. The Twostep Cluster procedure now provides interactive model viewer output. Twostep Cluster is available in the Statitics Base option.

Additional rule-checking on quality control charts. Rule-checking is now performed on several additional control charts. When rule-checking is requested for an X-bar chart, it will also be performed on the accompanying R (range) or s (standard deviation) chart. Similarly, when rule-checking is requested for an Individuals (Runs) chart, it will also be performed on the accompanying Moving Range chart. Quality control charts are available in the Statistics Base option.

Windows

There are a number of different types of windows in PASW Statistics:

Data Editor. The Data Editor displays the contents of the data file. You can create new data files or modify existing data files with the Data Editor. If you have more than one data file open, there is a separate Data Editor window for each data file.

Viewer. All statistical results, tables, and charts are displayed in the Viewer. You can edit the output and save it for later use. A Viewer window opens automatically the first time you run a procedure that generates output.

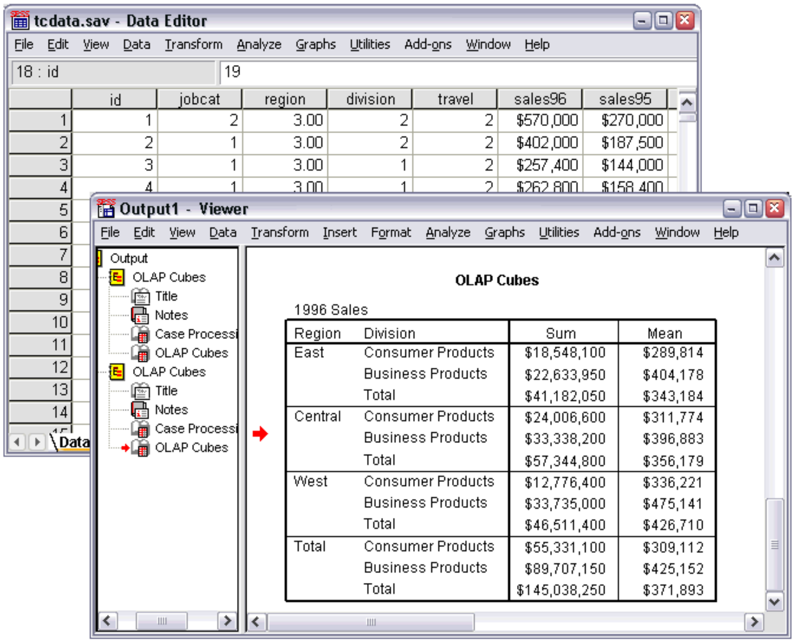
Pivot Table Editor. Output that is displayed in pivot tables can be modified in many ways with the Pivot Table Editor. You can edit text, swap data in rows and columns, add color, create multidimensional tables, and selectively hide and show results.

Chart Editor. You can modify high-resolution charts and plots in chart windows. You can change the colors, select different type fonts or sizes, switch the horizontal and vertical axes, rotate 3-D scatterplots, and even change the chart type.

Text Output Editor. Text output that is not displayed in pivot tables can be modified with the Text Output Editor. You can edit the output and change font characteristics (type, style, color, size).

Syntax Editor. You can paste your dialog box choices into a syntax window, where your selections appear in the form of command syntax. You can then edit the command syntax to use special features that are not available through dialog boxes. You can save these commands in a file for use in subsequent sessions.

Figure 1-1
Data Editor and Viewer


tcdata.sav - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Output1 - Viewer
File Edit View Data Transform Insert Format Analyze Graphs Utilities Add-ons Window Help
Output
OLAP Cubes
Title
Notes
Case Processi
OLAP Cubes
OLAP Cubes
Title
Notes
Case Processi
OLAP Cubes
Data

18 : id 19

	id	jobcat	region	division	travel	sales96	sales95
1	1	2	3.00	2	2	$570,000	$270,000
2	2	1	3.00	2	2	$402,000	$187,500
3	3	1	3.00	1	2	$257,400	$144,000
4	4	1	3.00	1	2	$262,800	$158,400

OLAP Cubes

1996 Sales

Region	Division	Sum	Mean
East	Consumer Products	$18,548,100	$289,814
	Business Products	$22,633,950	$404,178
	Total	$41,182,050	$343,184
Central	Consumer Products	$24,006,600	$311,774
	Business Products	$33,338,200	$396,883
	Total	$57,344,800	$356,179
West	Consumer Products	$12,776,400	$336,221
	Business Products	$33,735,000	$475,141
	Total	$46,511,400	$426,710
Total	Consumer Products	$55,331,100	$309,112
	Business Products	$89,707,150	$425,152
	Total	$145,038,250	$371,893

Designated Window versus Active Window

If you have more than one open Viewer window, output is routed to the **designated** Viewer window. If you have more than one open Syntax Editor window, command syntax is pasted into the designated Syntax Editor window. The designated windows are indicated by a plus sign in the icon in the title bar. You can change the designated windows at any time.

The designated window should not be confused with the **active** window, which is the currently selected window. If you have overlapping windows, the active window appears in the foreground. If you open a window, that window automatically becomes the active window and the designated window.

Changing the Designated Window

▶ Make the window that you want to designate the active window (click anywhere in the window).

▶ Click the Designate Window button on the toolbar (the plus sign icon).

or

▶ From the menus choose:

Utilities
  Designate Window

Note: For Data Editor windows, the active Data Editor window determines the dataset that is used in subsequent calculations or analyses. There is no "designated" Data Editor window. For more information, see the topic Basic Handling of Multiple Data Sources in Chapter 6 on p. 92.

Status Bar

The status bar at the bottom of each PASW Statistics window provides the following information:

Command status. For each procedure or command that you run, a case counter indicates the number of cases processed so far. For statistical procedures that require iterative processing, the number of iterations is displayed.

Filter status. If you have selected a random sample or a subset of cases for analysis, the message Filter on indicates that some type of case filtering is currently in effect and not all cases in the data file are included in the analysis.

Weight status. The message Weight on indicates that a weight variable is being used to weight cases for analysis.

Split File status. The message Split File on indicates that the data file has been split into separate groups for analysis, based on the values of one or more grouping variables.

Dialog Boxes

Most menu selections open dialog boxes. You use dialog boxes to select variables and options for analysis.

Dialog boxes for statistical procedures and charts typically have two basic components:

Source variable list. A list of variables in the active dataset. Only variable types that are allowed by the selected procedure are displayed in the source list. Use of short string and long string variables is restricted in many procedures.

Target variable list(s). One or more lists indicating the variables that you have chosen for the analysis, such as dependent and independent variable lists.

Variable Names and Variable Labels in Dialog Box Lists

You can display either variable names or variable labels in dialog box lists, and you can control the sort order of variables in source variable lists. To control the default display attributes of variables in source lists, choose Options on the Edit menu. For more information, see the topic General Options in Chapter 16 on p. 291.

You can also change the variable list display attributes within dialogs. The method for changing the display attributes depends on the dialog:

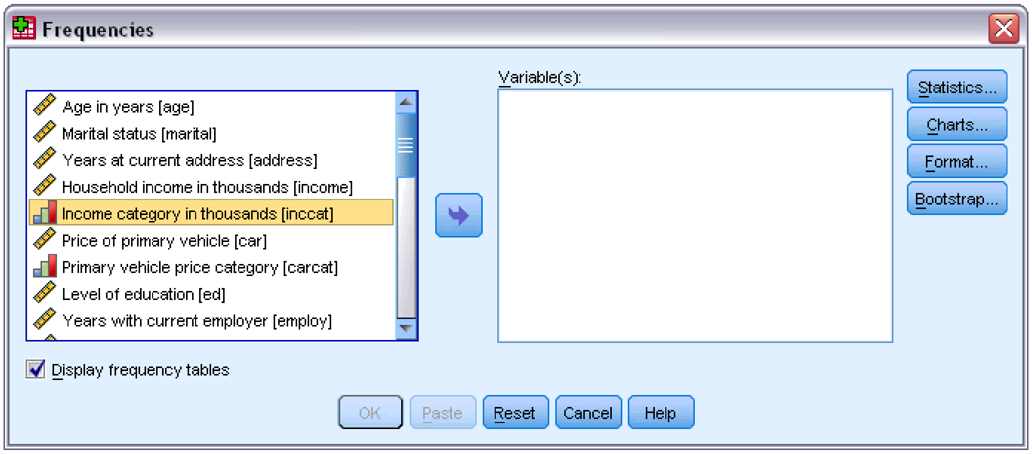
- If the dialog provides sorting and display controls above the source variable list, use those controls to change the display attributes.
- If the dialog does not contain sorting controls above the source variable list, right-click on any variable in the source list and select the display attributes from the context menu.

You can display either variable names or variable labels (names are displayed for any variables without defined labels), and you can sort the source list by file order, alphabetical order, or measurement level. (In dialogs with sorting controls above the source variable list, the default selection of **None** sorts the list in file order.)

Resizing Dialog Boxes

You can resize dialog boxes just like windows, by clicking and dragging the outside borders or corners. For example, if you make the dialog box wider, the variable lists will also be wider.

Figure 1-2
Resized dialog box


Frequencies
Age in years [age]
Marital status [marital]
Years at current address [address]
Household income in thousands [income]
Income category in thousands [inccat]
Price of primary vehicle [car]
Primary vehicle price category [carcat]
Level of education [ed]
Years with current employer [employ]
Variable(s):
Statistics...
Charts...
Format...
Bootstrap...
☑ Display frequency tables
OK
Paste
Reset
Cancel
Help

Dialog Box Controls

There are five standard controls in most dialog boxes:

OK. Runs the procedure. After you select your variables and choose any additional specifications, click OK to run the procedure and close the dialog box. Some dialogs have a Run button instead of the OK button.

Paste. Generates command syntax from the dialog box selections and pastes the syntax into a syntax window. You can then customize the commands with additional features that are not available from dialog boxes.

Reset. Deselects any variables in the selected variable list(s) and resets all specifications in the dialog box and any subdialog boxes to the default state.

Cancel. Cancels any changes that were made in the dialog box settings since the last time it was opened and closes the dialog box. Within a session, dialog box settings are persistent. A dialog box retains your last set of specifications until you override them.

Help. Provides context-sensitive Help. This control takes you to a Help window that contains information about the current dialog box.

Selecting Variables

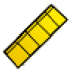
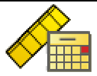
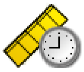
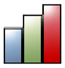
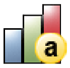
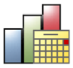
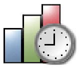
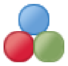
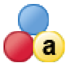
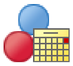
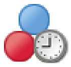
To select a single variable, simply select it in the source variable list and drag and drop it into the target variable list. You can also use arrow button to move variables from the source list to the target lists. If there is only one target variable list, you can double-click individual variables to move them from the source list to the target list.

You can also select multiple variables:

- To select multiple variables that are grouped together in the variable list, click the first variable and then Shift-click the last variable in the group.
- To select multiple variables that are not grouped together in the variable list, click the first variable, then Ctrl-click the next variable, and so on (Macintosh: Command-click).

Data Type, Measurement Level, and Variable List Icons

The icons that are displayed next to variables in dialog box lists provide information about the variable type and measurement level.

Measurement Level	**Data Type**			
	Numeric	String	Date	Time
Scale (Continuous)		n/a		
Ordinal				
Nominal				

- For more information on measurement level, see Variable Measurement Level on p. 71.
- For more information on numeric, string, date, and time data types, see Variable Type on p. 72.

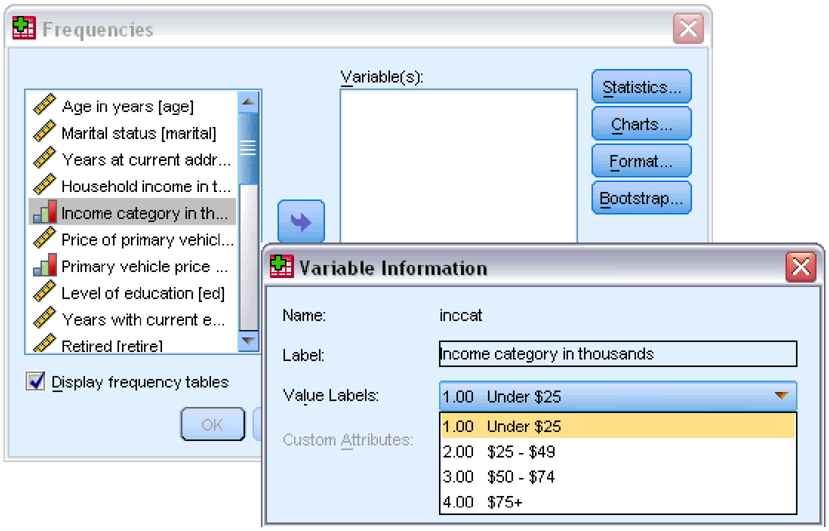
Getting Information about Variables in Dialog Boxes

Many dialogs provide the ability to find out more about the variables displayed in the variable lists.

▶ Right-click a variable in the source or target variable list.

▶ Choose Variable Information.

Figure 1-3
Variable information



Name: inccat

Label: Income category in thousands

Value Labels: 1.00 Under $25

1.00	Under $25
2.00	$25 - $49
3.00	$50 - $74
4.00	$75+

Basic Steps in Data Analysis

Analyzing data with PASW Statistics is easy. All you have to do is:

Get your data into PASW Statistics. You can open a previously saved PASW Statistics data file, you can read a spreadsheet, database, or text data file, or you can enter your data directly in the Data Editor.

Select a procedure. Select a procedure from the menus to calculate statistics or to create a chart.

Select the variables for the analysis. The variables in the data file are displayed in a dialog box for the procedure.

Run the procedure and look at the results. Results are displayed in the Viewer.

Statistics Coach

If you are unfamiliar with PASW Statistics or with the available statistical procedures, the Statistics Coach can help you get started by prompting you with simple questions, nontechnical language, and visual examples that help you select the basic statistical and charting features that are best suited for your data.

To use the Statistics Coach, from the menus in any PASW Statistics window choose:

Help
  Statistics Coach

The Statistics Coach covers only a selected subset of procedures. It is designed to provide general assistance for many of the basic, commonly used statistical techniques.

Finding Out More

For a comprehensive overview of the basics, see the online tutorial. From any PASW Statistics menu choose:

Help
  Tutorial

Chapter
2

Getting Help

Help is provided in many different forms:

Help menu. The Help menu in most windows provides access to the main Help system, plus tutorials and technical reference material.

- **Topics.** Provides access to the Contents, Index, and Search tabs, which you can use to find specific Help topics.
- **Tutorial.** Illustrated, step-by-step instructions on how to use many of the basic features. You don't have to view the whole tutorial from start to finish. You can choose the topics you want to view, skip around and view topics in any order, and use the index or table of contents to find specific topics.
- **Case Studies.** Hands-on examples of how to create various types of statistical analyses and how to interpret the results. The sample data files used in the examples are also provided so that you can work through the examples to see exactly how the results were produced. You can choose the specific procedure(s) that you want to learn about from the table of contents or search for relevant topics in the index.
- **Statistics Coach.** A wizard-like approach to guide you through the process of finding the procedure that you want to use. After you make a series of selections, the Statistics Coach opens the dialog box for the statistical, reporting, or charting procedure that meets your selected criteria.
- **Command Syntax Reference.** Detailed command syntax reference information is available in two forms: integrated into the overall Help system and as a separate document in PDF form in the *Command Syntax Reference*, available from the Help menu.
- **Statistical Algorithms.** The algorithms used for most statistical procedures are available in two forms: integrated into the overall Help system and as a separate document in PDF form available on the manuals CD. For links to specific algorithms in the Help system, choose Algorithms from the Help menu.

Context-sensitive Help. In many places in the user interface, you can get context-sensitive Help.

- **Dialog box Help buttons.** Most dialog boxes have a Help button that takes you directly to a Help topic for that dialog box. The Help topic provides general information and links to related topics.

- **Pivot table context menu Help.** Right-click on terms in an activated pivot table in the Viewer and choose What's This? from the context menu to display definitions of the terms.
- **Command syntax.** In a command syntax window, position the cursor anywhere within a syntax block for a command and press F1 on the keyboard. A complete command syntax chart for that command will be displayed. Complete command syntax documentation is available from the links in the list of related topics and from the Help Contents tab.

Other Resources

Technical Support Web site. Answers to many common problems can be found at *http://support.spss.com*. (The Technical Support Web site requires a login ID and password. Information on how to obtain an ID and password is provided at the URL listed above.)

Developer Central. Developer Central has resources for all levels of users and application developers. Download utilities, graphics examples, new statistical modules, and articles. Visit Developer Central at *http://www.spss.com/devcentral*.

Getting Help on Output Terms

To see a definition for a term in pivot table output in the Viewer:

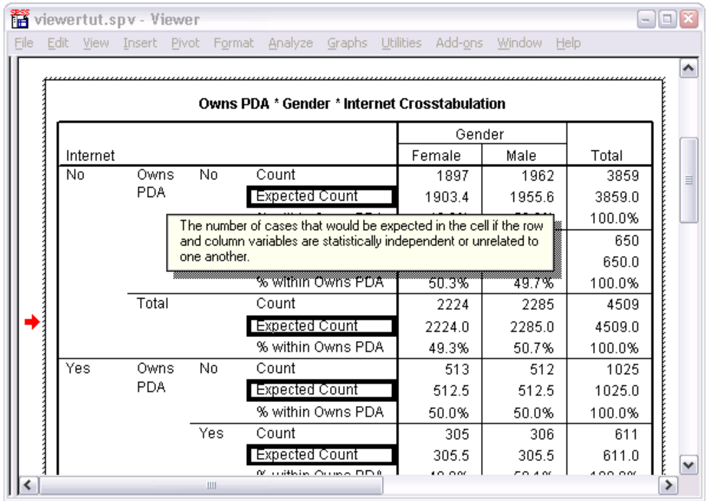
▶ Double-click the pivot table to activate it.

▶ Right-click on the term that you want explained.

▶ Choose What's This? from the context menu.

A definition of the term is displayed in a pop-up window.

Figure 2-1
Activated pivot table glossary Help with right mouse button



Owns PDA * Gender * Internet Crosstabulation

				Gender		
Internet				Female	Male	Total
No	Owns PDA	No	Count	1897	1962	3859
			Expected Count	1903.4	1955.6	3859.0
						100.0%
						650
						650.0
			% within Owns PDA	50.3%	49.7%	100.0%
	Total		Count	2224	2285	4509
			Expected Count	2224.0	2285.0	4509.0
			% within Owns PDA	49.3%	50.7%	100.0%
Yes	Owns PDA	No	Count	513	512	1025
			Expected Count	512.5	512.5	1025.0
			% within Owns PDA	50.0%	50.0%	100.0%
		Yes	Count	305	306	611
			Expected Count	305.5	305.5	611.0

The number of cases that would be expected in the cell if the row and column variables are statistically independent or unrelated to one another.

Chapter

3

Data Files

Data files come in a wide variety of formats, and this software is designed to handle many of them, including:

- Spreadsheets created with Excel and Lotus
- Database tables from many database sources, including Oracle, SQLServer, Access, dBASE, and others
- Tab-delimited and other types of simple text files
- Data files in PASW Statistics format created on other operating systems
- SYSTAT data files
- SAS data files
- Stata data files

Opening Data Files

In addition to files saved in PASW Statistics format, you can open Excel, SAS, Stata, tab-delimited, and other files without converting the files to an intermediate format or entering data definition information.

- Opening a data file makes it the active dataset. If you already have one or more open data files, they remain open and available for subsequent use in the session. Clicking anywhere in the Data Editor window for an open data file will make it the active dataset. For more information, see the topic Working with Multiple Data Sources in Chapter 6 on p. 92.
- In distributed analysis mode using a remote server to process commands and run procedures, the available data files, folders, and drives are dependent on what is available on or from the remote server. The current server name is indicated at the top of the dialog box. You will not have access to data files on your local computer unless you specify the drive as a shared device and the folders containing your data files as shared folders. For more information, see the topic Distributed Analysis Mode in Chapter 4 on p. 62.

To Open Data Files

▶ From the menus choose:

File
  Open
    Data...

▶ In the Open Data dialog box, select the file that you want to open.

▶ Click Open.

Optionally, you can:

- Automatically set the width of each string variable to the longest observed value for that variable using Minimize string widths based on observed values. This is particularly useful when reading code page data files in Unicode mode. For more information, see the topic General Options in Chapter 16 on p. 291.
- Read variable names from the first row of spreadsheet files.
- Specify a range of cells to read from spreadsheet files.
- Specify a worksheet within an Excel file to read (Excel 95 or later).

For information on reading data from databases, see Reading Database Files on p. 14. For information on reading data from text data files, see Text Wizard on p. 28.

Data File Types

PASW Statistics. Opens data files saved in PASW Statistics format and also the DOS product SPSS/PC+.

SPSS/PC+. Opens SPSS/PC+ data files. This is available only on Windows operating systems.

SYSTAT. Opens SYSTAT data files.

PASW Statistics Portable. Opens data files saved in portable format. Saving a file in portable format takes considerably longer than saving the file in PASW Statistics format.

Excel. Opens Excel files.

Lotus 1-2-3. Opens data files saved in 1-2-3 format for release 3.0, 2.0, or 1A of Lotus.

SYLK. Opens data files saved in SYLK (symbolic link) format, a format used by some spreadsheet applications.

dBASE. Opens dBASE-format files for either dBASE IV, dBASE III or III PLUS, or dBASE II. Each case is a record. Variable and value labels and missing-value specifications are lost when you save a file in this format.

SAS. SAS versions 6–9 and SAS transport files. Using command syntax, you can also read value labels from a SAS format catalog file.

Stata. Stata versions 4–8.

Opening File Options

Read variable names. For spreadsheets, you can read variable names from the first row of the file or the first row of the defined range. The values are converted as necessary to create valid variable names, including converting spaces to underscores.

Worksheet. Excel 95 or later files can contain multiple worksheets. By default, the Data Editor reads the first worksheet. To read a different worksheet, select the worksheet from the drop-down list.

Range. For spreadsheet data files, you can also read a range of cells. Use the same method for specifying cell ranges as you would with the spreadsheet application.

Reading Excel 95 or Later Files

The following rules apply to reading Excel 95 or later files:

Data type and width. Each column is a variable. The data type and width for each variable are determined by the data type and width in the Excel file. If the column contains more than one data type (for example, date and numeric), the data type is set to string, and all values are read as valid string values.

Blank cells. For numeric variables, blank cells are converted to the system-missing value, indicated by a period. For string variables, a blank is a valid string value, and blank cells are treated as valid string values.

Variable names. If you read the first row of the Excel file (or the first row of the specified range) as variable names, values that don't conform to variable naming rules are converted to valid variable names, and the original names are used as variable labels. If you do not read variable names from the Excel file, default variable names are assigned.

Reading Older Excel Files and Other Spreadsheets

The following rules apply to reading Excel files prior to Excel 95 and other spreadsheet data:

Data type and width. The data type and width for each variable are determined by the column width and data type of the first data cell in the column. Values of other types are converted to the system-missing value. If the first data cell in the column is blank, the global default data type for the spreadsheet (usually numeric) is used.

Blank cells. For numeric variables, blank cells are converted to the system-missing value, indicated by a period. For string variables, a blank is a valid string value, and blank cells are treated as valid string values.

Variable names. If you do not read variable names from the spreadsheet, the column letters (*A*, *B*, *C*, ...) are used for variable names for Excel and Lotus files. For SYLK files and Excel files saved in R1C1 display format, the software uses the column number preceded by the letter *C* for variable names (*C1*, *C2*, *C3*, ...).

Reading dBASE Files

Database files are logically very similar to PASW Statistics data files. The following general rules apply to dBASE files:

- Field names are converted to valid variable names.
- Colons used in dBASE field names are translated to underscores.
- Records marked for deletion but not actually purged are included. The software creates a new string variable, *D_R*, which contains an asterisk for cases marked for deletion.

Reading Stata Files

The following general rules apply to Stata data files:

- **Variable names.** Stata variable names are converted to PASW Statistics variable names in case-sensitive form. Stata variable names that are identical except for case are converted to valid variable names by appending an underscore and a sequential letter (*_A*, *_B*, *_C*, ..., *_Z*, *_AA*, *_AB*, ..., and so forth).
- **Variable labels.** Stata variable labels are converted to PASW Statistics variable labels.
- **Value labels.** Stata value labels are converted to PASW Statistics value labels, except for Stata value labels assigned to "extended" missing values.
- **Missing values.** Stata "extended" missing values are converted to system-missing values.
- **Date conversion.** Stata date format values are converted to PASW Statistics `DATE` format (d-m-y) values. Stata "time-series" date format values (weeks, months, quarters, and so on) are converted to simple numeric (F) format, preserving the original, internal integer value, which is the number of weeks, months, quarters, and so on, since the start of 1960.

Reading Database Files

You can read data from any database format for which you have a database driver. In local analysis mode, the necessary drivers must be installed on your local computer. In distributed analysis mode (available with PASW Statistics Server), the drivers must be installed on the remote server.For more information, see the topic Distributed Analysis Mode in Chapter 4 on p. 62.

To Read Database Files

▶ From the menus choose:

File
  Open Database
    New Query...

▶ Select the data source.

▶ If necessary (depending on the data source), select the database file and/or enter a login name, password, and other information.

▶ Select the table(s) and fields. For OLE DB data sources (available only on Windows operating systems), you can only select one table.

▶ Specify any relationships between your tables.

▶ Optionally:

- Specify any selection criteria for your data.
- Add a prompt for user input to create a parameter query.
- Save your constructed query before running it.

To Edit Saved Database Queries

▶ From the menus choose:

File
  Open Database
    Edit Query...

▶ Select the query file (**.spq*) that you want to edit.

▶ Follow the instructions for creating a new query.

To Read Database Files with Saved Queries

▶ From the menus choose:

File
  Open Database
    Run Query...

▶ Select the query file (**.spq*) that you want to run.

▶ If necessary (depending on the database file), enter a login name and password.

▶ If the query has an embedded prompt, enter other information if necessary (for example, the quarter for which you want to retrieve sales figures).

Selecting a Data Source

Use the first screen of the Database Wizard to select the type of data source to read.

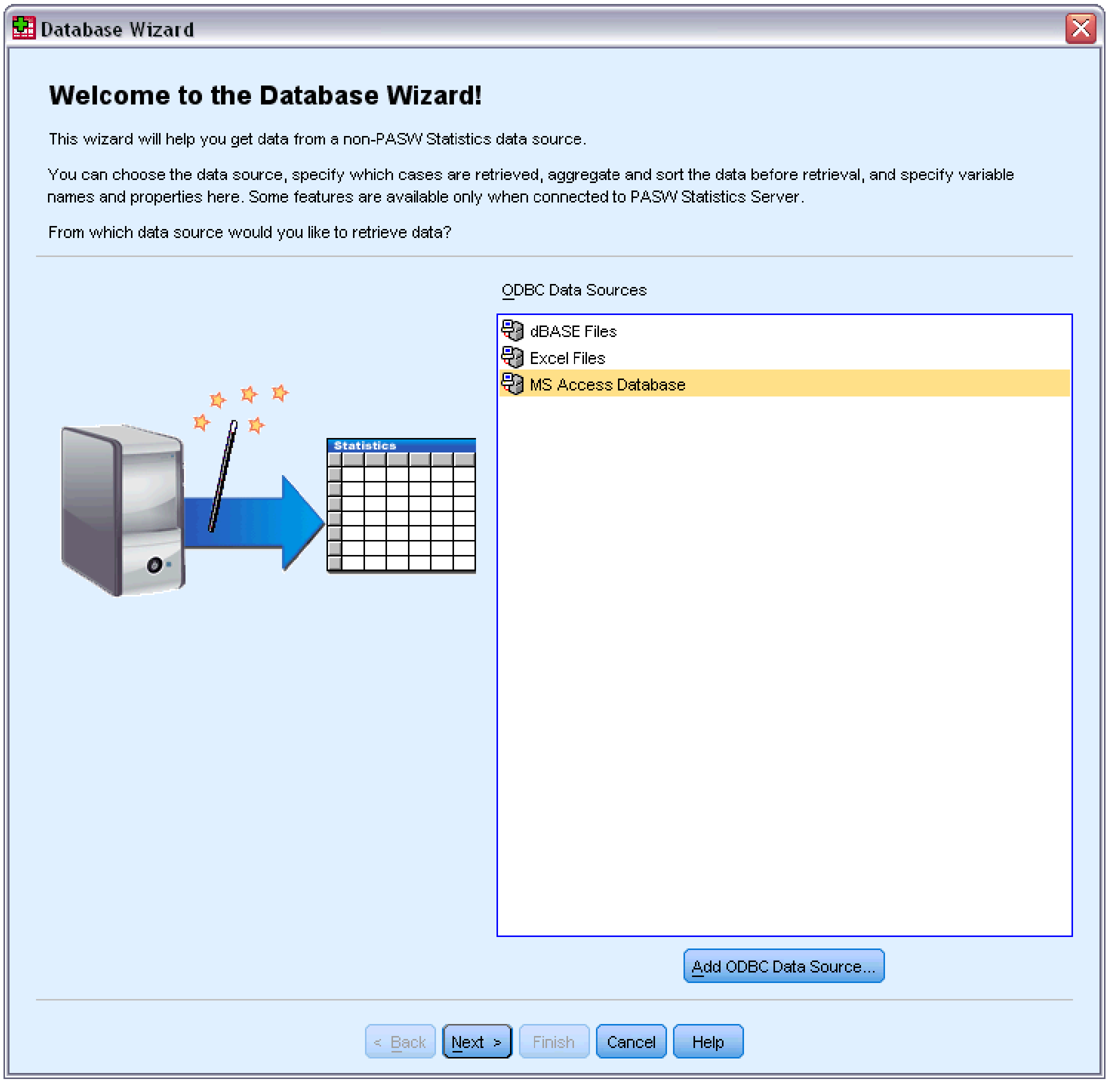
ODBC Data Sources

If you do not have any ODBC data sources configured, or if you want to add a new data source, click Add ODBC Data Source.

- On Linux operating systems, this button is not available. ODBC data sources are specified in *odbc.ini*, and the *ODBCINI* environment variables must be set to the location of that file. For more information, see the documentation for your database drivers.
- In distributed analysis mode (available with PASW Statistics Server), this button is not available. To add data sources in distributed analysis mode, see your system administrator.

An ODBC data source consists of two essential pieces of information: the driver that will be used to access the data and the location of the database you want to access. To specify data sources, you must have the appropriate drivers installed. Drivers for a variety of database formats are available at *http://www.spss.com/drivers*.

Figure 3-1
Database Wizard



Database Wizard

Welcome to the Database Wizard!

This wizard will help you get data from a non-PASW Statistics data source.

You can choose the data source, specify which cases are retrieved, aggregate and sort the data before retrieval, and specify variable names and properties here. Some features are available only when connected to PASW Statistics Server.

From which data source would you like to retrieve data?

ODBC Data Sources

- dBASE Files
- Excel Files
- MS Access Database

OLE DB Data Sources

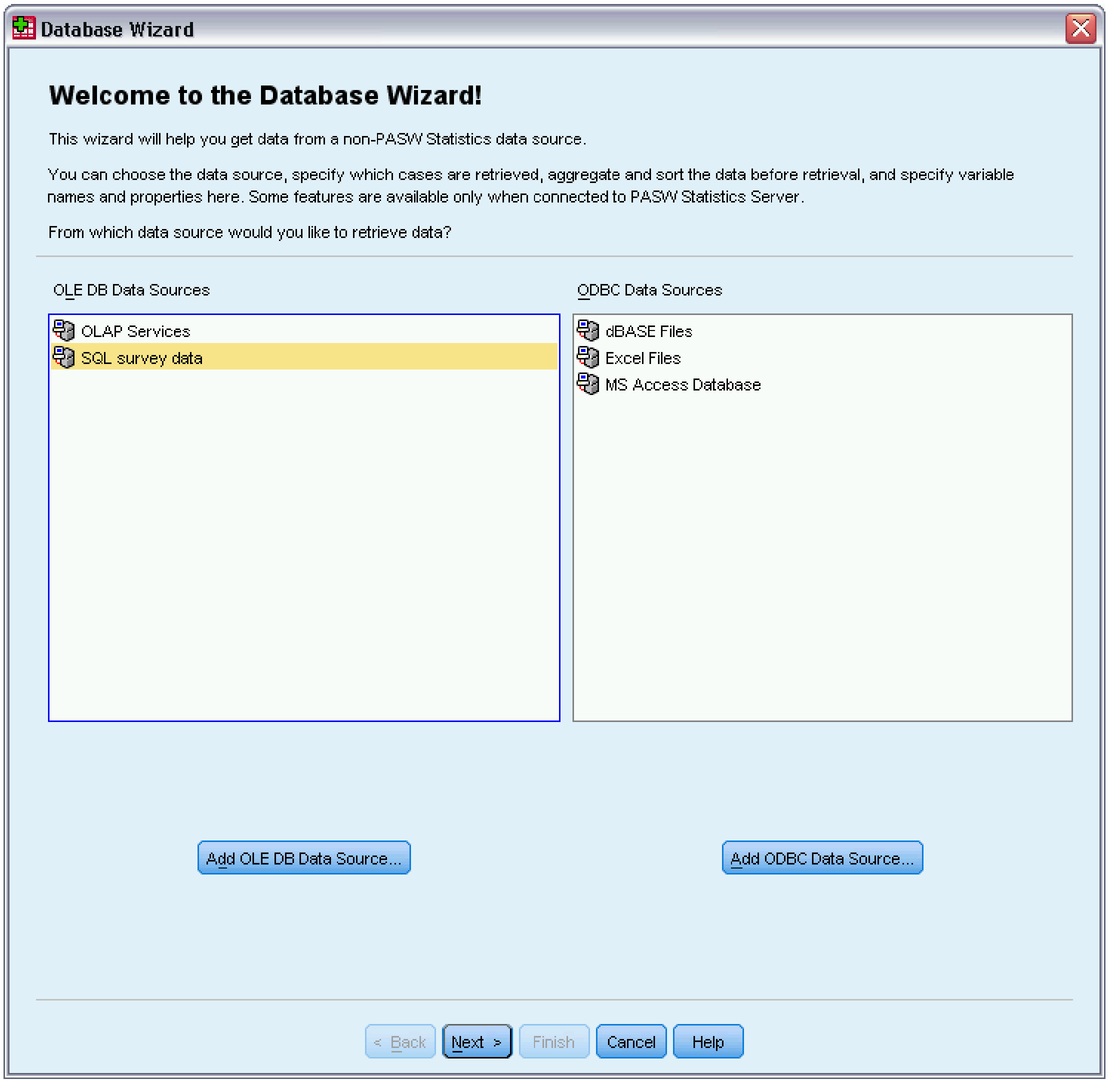
To access OLE DB data sources (available only on Microsoft Windows operating systems), you must have the following items installed:

- .NET framework. To obtain the most recent version of the .NET framework, go to *http://www.microsoft.com/net*.
- PASW Reports for Surveys Components. A version that is compatible with this release can be installed from the installation media. If you are using PASW Statistics Developer, you can download a compatible version from the Downloads tab at www.spss.com/statistics (*http://www.spss.com/statistics/*).

The following limitations apply to OLE DB data sources:

- Table joins are not available for OLE DB data sources. You can read only one table at a time.
- You can add OLE DB data sources only in local analysis mode. To add OLE DB data sources in distributed analysis mode on a Windows server, consult your system administrator.
- In distributed analysis mode (available with PASW Statistics Server), OLE DB data sources are available only on Windows servers, and both .NET and PASW Reports for Surveys Components must be installed on the server.

Figure 3-2
Database Wizard with access to OLE DB data sources



Database Wizard

Welcome to the Database Wizard!

This wizard will help you get data from a non-PASW Statistics data source.

You can choose the data source, specify which cases are retrieved, aggregate and sort the data before retrieval, and specify variable names and properties here. Some features are available only when connected to PASW Statistics Server.

From which data source would you like to retrieve data?

OLE DB Data Sources
- OLAP Services
- SQL survey data

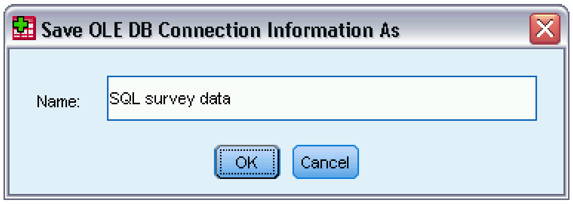
ODBC Data Sources
- dBASE Files
- Excel Files
- MS Access Database

To add an OLE DB data source:

- ▶ Click Add OLE DB Data Source.
- ▶ In Data Link Properties, click the Provider tab and select the OLE DB provider.

- Click Next or click the Connection tab.
- Select the database by entering the directory location and database name or by clicking the button to browse to a database. (A user name and password may also be required.)
- Click OK after entering all necessary information. (You can make sure the specified database is available by clicking the Test Connection button.)
- Enter a name for the database connection information. (This name will be displayed in the list of available OLE DB data sources.)

Figure 3-3
Save OLE DB Connection Information As dialog box



Name: SQL survey data

- Click OK.

This takes you back to the first screen of the Database Wizard, where you can select the saved name from the list of OLE DB data sources and continue to the next step of the wizard.

Deleting OLE DB Data Sources

To delete data source names from the list of OLE DB data sources, delete the UDL file with the name of the data source in:

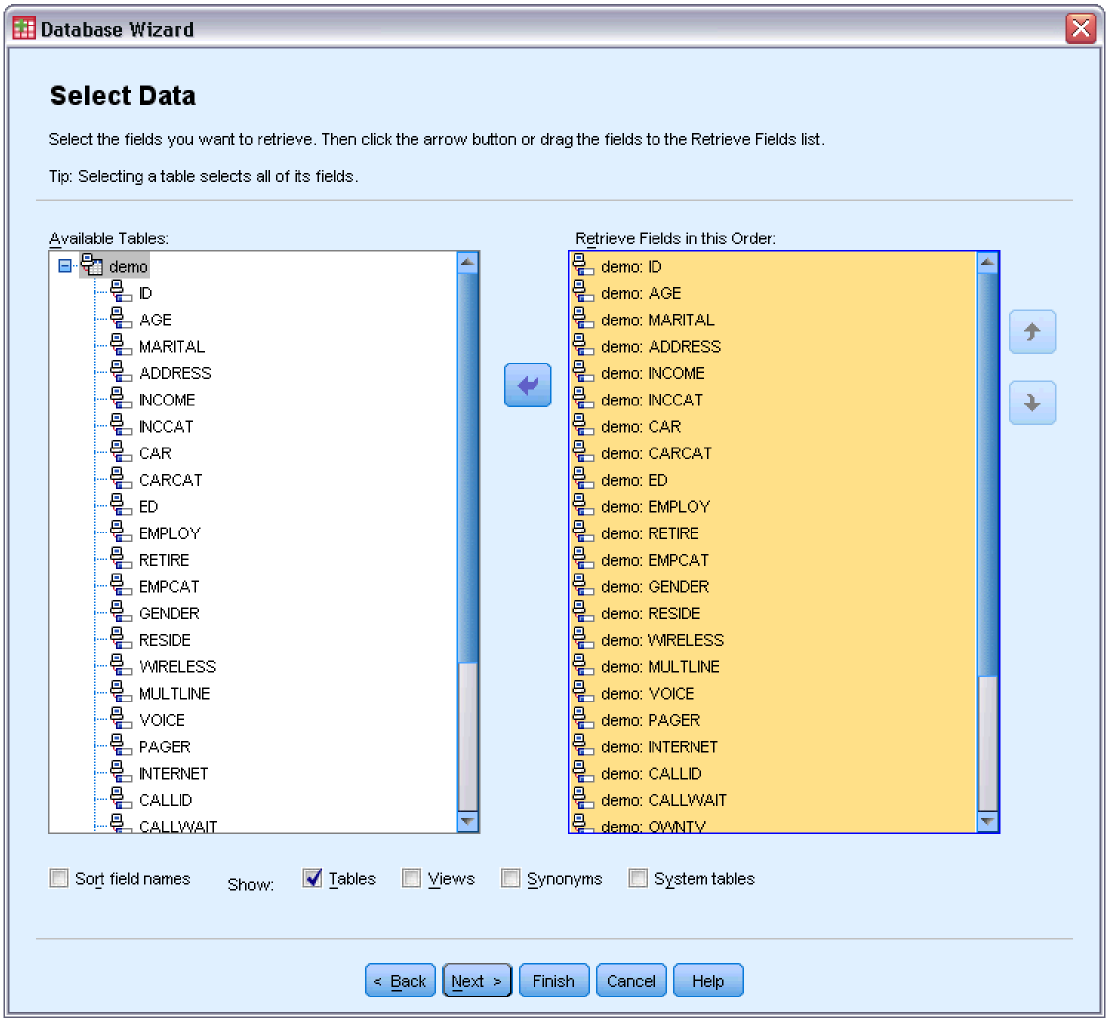
[drive]:\Documents and Settings\[user login]\Local Settings\Application Data\SPSS\UDL

Selecting Data Fields

The Select Data step controls which tables and fields are read. Database fields (columns) are read as variables.

If a table has any field(s) selected, all of its fields will be visible in the following Database Wizard windows, but only fields that are selected in this step will be imported as variables. This enables you to create table joins and to specify criteria by using fields that you are not importing.

Figure 3-4
Database Wizard, selecting data



Select Data

Select the fields you want to retrieve. Then click the arrow button or drag the fields to the Retrieve Fields list.

Tip: Selecting a table selects all of its fields.

Available Tables:

- demo
  - ID
  - AGE
  - MARITAL
  - ADDRESS
  - INCOME
  - INCCAT
  - CAR
  - CARCAT
  - ED
  - EMPLOY
  - RETIRE
  - EMPCAT
  - GENDER
  - RESIDE
  - WIRELESS
  - MULTLINE
  - VOICE
  - PAGER
  - INTERNET
  - CALLID
  - CALLWAIT

Retrieve Fields in this Order:

- demo: ID
- demo: AGE
- demo: MARITAL
- demo: ADDRESS
- demo: INCOME
- demo: INCCAT
- demo: CAR
- demo: CARCAT
- demo: ED
- demo: EMPLOY
- demo: RETIRE
- demo: EMPCAT
- demo: GENDER
- demo: RESIDE
- demo: WIRELESS
- demo: MULTLINE
- demo: VOICE
- demo: PAGER
- demo: INTERNET
- demo: CALLID
- demo: CALLWAIT
- demo: OWNTV

□ Sort field names Show: ☑ Tables □ Views □ Synonyms □ System tables

Displaying field names. To list the fields in a table, click the plus sign (+) to the left of a table name. To hide the fields, click the minus sign (–) to the left of a table name.

To add a field. Double-click any field in the Available Tables list, or drag it to the Retrieve Fields In This Order list. Fields can be reordered by dragging and dropping them within the fields list.

To remove a field. Double-click any field in the Retrieve Fields In This Order list, or drag it to the Available Tables list.

Sort field names. If this check box is selected, the Database Wizard will display your available fields in alphabetical order.

By default, the list of available tables displays only standard database tables. You can control the type of items that are displayed in the list:

- **Tables.** Standard database tables.

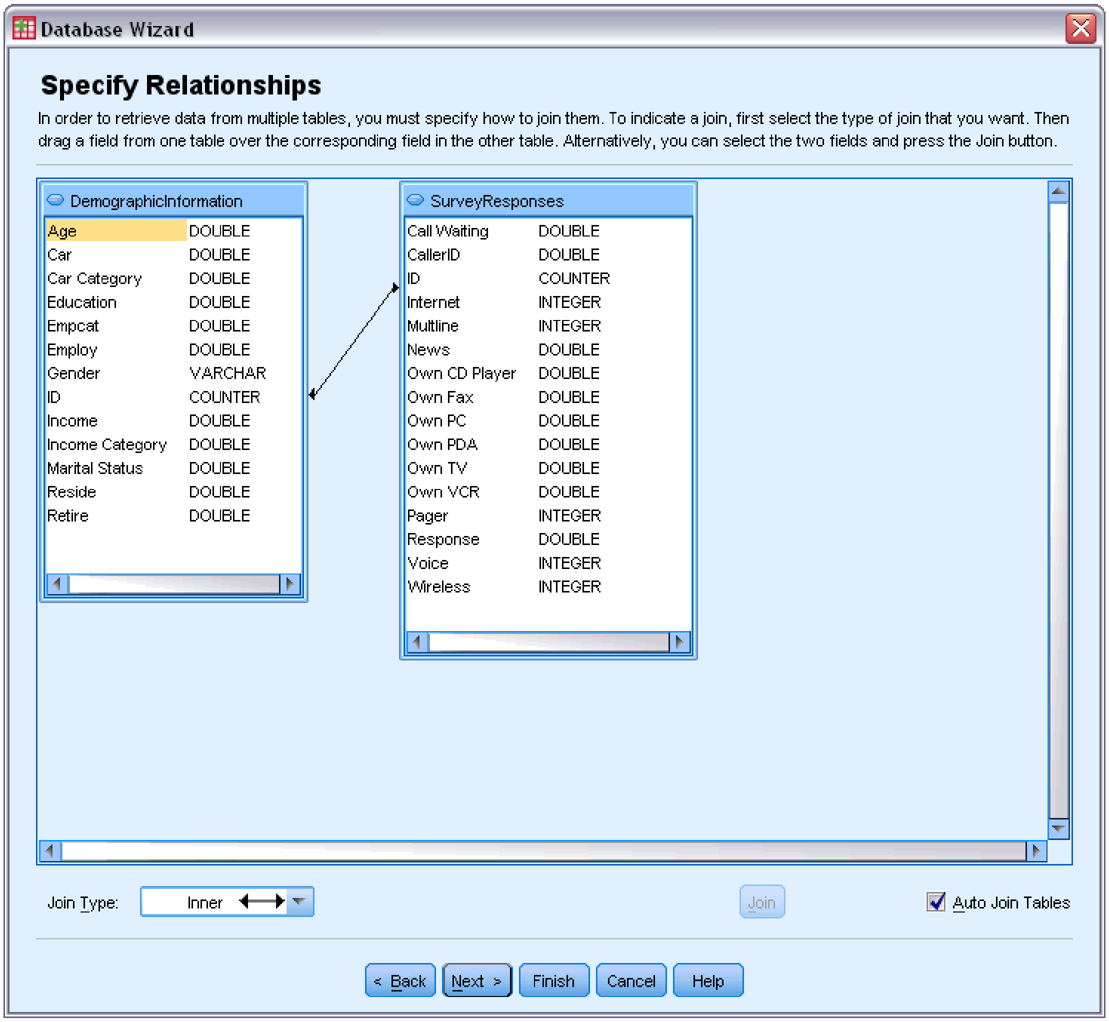
- **Views.** Views are virtual or dynamic "tables" defined by queries. These can include joins of multiple tables and/or fields derived from calculations based on the values of other fields.
- **Synonyms.** A synonym is an alias for a table or view, typically defined in a query.
- **System tables.** System tables define database properties. In some cases, standard database tables may be classified as system tables and will only be displayed if you select this option. Access to real system tables is often restricted to database administrators.

Note: For OLE DB data sources (available only on Windows operating systems), you can select fields only from a single table. Multiple table joins are not supported for OLE DB data sources.

Creating a Relationship between Tables

The Specify Relationships step allows you to define the relationships between the tables for ODBC data sources. If fields from more than one table are selected, you must define at least one join.

Figure 3-5
Database Wizard, specifying relationships



Database Wizard

Specify Relationships

In order to retrieve data from multiple tables, you must specify how to join them. To indicate a join, first select the type of join that you want. Then drag a field from one table over the corresponding field in the other table. Alternatively, you can select the two fields and press the Join button.

DemographicInformation	
Age	DOUBLE
Car	DOUBLE
Car Category	DOUBLE
Education	DOUBLE
Empcat	DOUBLE
Employ	DOUBLE
Gender	VARCHAR
ID	COUNTER
Income	DOUBLE
Income Category	DOUBLE
Marital Status	DOUBLE
Reside	DOUBLE
Retire	DOUBLE

SurveyResponses	
Call Waiting	DOUBLE
CallerID	DOUBLE
ID	COUNTER
Internet	INTEGER
Multline	INTEGER
News	DOUBLE
Own CD Player	DOUBLE
Own Fax	DOUBLE
Own PC	DOUBLE
Own PDA	DOUBLE
Own TV	DOUBLE
Own VCR	DOUBLE
Pager	INTEGER
Response	DOUBLE
Voice	INTEGER
Wireless	INTEGER

Join Type: Inner ⟷

☑ Auto Join Tables

Establishing relationships. To create a relationship, drag a field from any table onto the field to which you want to join it. The Database Wizard will draw a **join line** between the two fields, indicating their relationship. These fields must be of the same data type.

Auto Join Tables. Attempts to automatically join tables based on primary/foreign keys or matching field names and data type.

Join Type. If outer joins are supported by your driver, you can specify inner joins, left outer joins, or right outer joins.

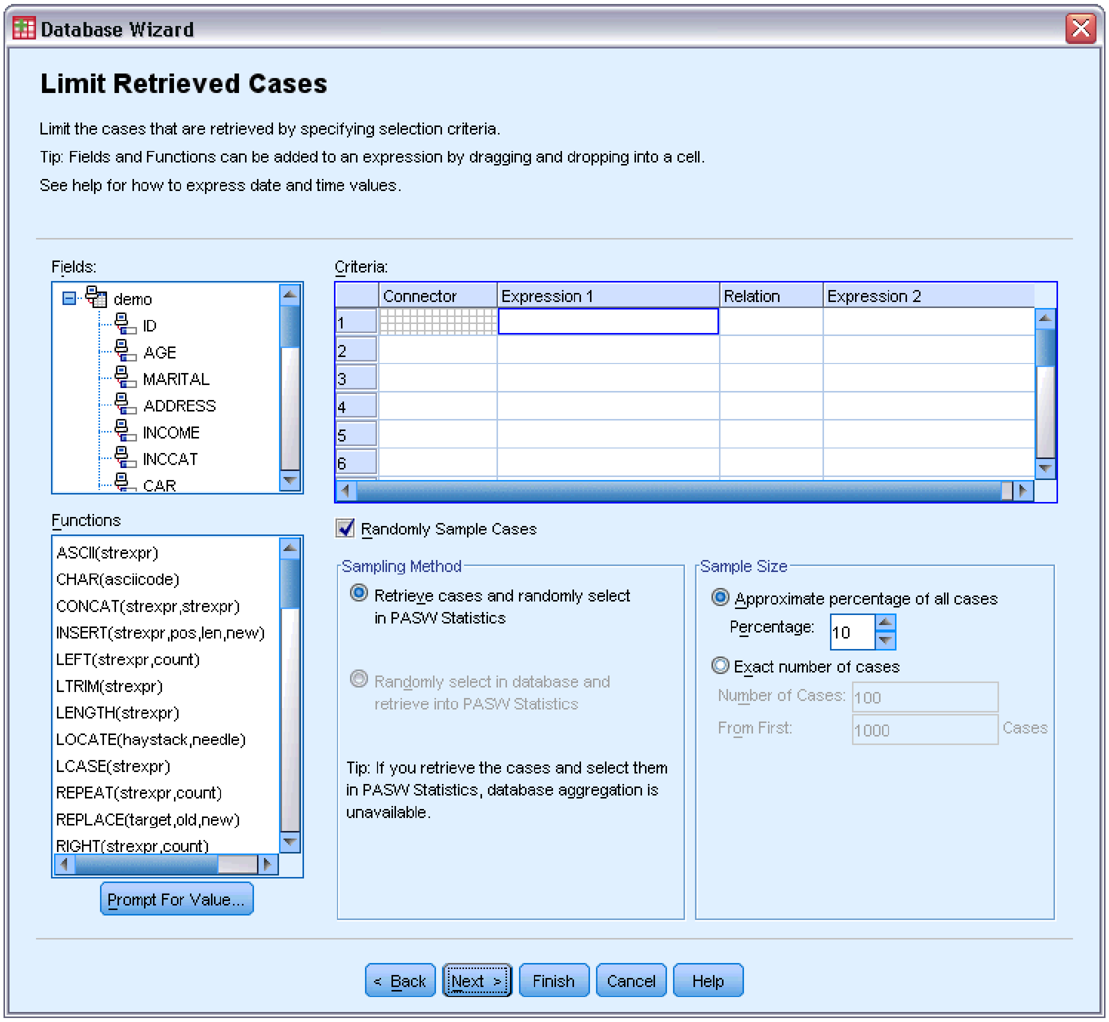
- **Inner joins.** An inner join includes only rows where the related fields are equal. In this example, all rows with matching ID values in the two tables will be included.
- **Outer joins.** In addition to one-to-one matching with inner joins, you can also use outer joins to merge tables with a one-to-many matching scheme. For example, you could match a table in which there are only a few records representing data values and associated descriptive labels with values in a table containing hundreds or thousands of records representing survey respondents. A left outer join includes all records from the table on the left and, from the table on the right, includes only those records in which the related fields are equal. In a right outer join, the join imports all records from the table on the right and, from the table on the left, imports only those records in which the related fields are equal.

Limiting Retrieved Cases

The Limit Retrieved Cases step allows you to specify the criteria to select subsets of cases (rows). Limiting cases generally consists of filling the criteria grid with criteria. Criteria consist of two expressions and some relation between them. The expressions return a value of *true*, *false*, or *missing* for each case.

- If the result is *true*, the case is selected.
- If the result is *false* or *missing*, the case is not selected.
- Most criteria use one or more of the six relational operators (<, >, <=, >=, =, and <>).
- Expressions can include field names, constants, arithmetic operators, numeric and other functions, and logical variables. You can use fields that you do not plan to import as variables.

Figure 3-6
Database Wizard, limiting retrieved cases



Limit Retrieved Cases

Limit the cases that are retrieved by specifying selection criteria.

Tip: Fields and Functions can be added to an expression by dragging and dropping into a cell.

See help for how to express date and time values.

	Connector	Expression 1	Relation	Expression 2
1				
2				
3				
4				
5				
6				

☑ Randomly Sample Cases

Sampling Method

● Retrieve cases and randomly select in PASW Statistics

○ Randomly select in database and retrieve into PASW Statistics

Tip: If you retrieve the cases and select them in PASW Statistics, database aggregation is unavailable.

Sample Size

● Approximate percentage of all cases

Percentage: 10

○ Exact number of cases

Number of Cases: 100

From First: 1000 Cases

To build your criteria, you need at least two expressions and a relation to connect the expressions.

▶ To build an expression, choose one of the following methods:

- In an Expression cell, type field names, constants, arithmetic operators, numeric and other functions, or logical variables.
- Double-click the field in the Fields list.
- Drag the field from the Fields list to an Expression cell.
- Choose a field from the drop-down menu in any active Expression cell.

▶ To choose the relational operator (such as = or >), put your cursor in the Relation cell and either type the operator or choose it from the drop-down menu.

If the SQL contains `WHERE` clauses with expressions for case selection, dates and times in expressions need to be specified in a special manner (including the curly braces shown in the examples):

- Date literals should be specified using the general form `{d 'yyyy-mm-dd'}`.
- Time literals should be specified using the general form `{t 'hh:mm:ss'}`.
- Date/time literals (timestamps) should be specified using the general form `{ts 'yyyy-mm-dd hh:mm:ss'}`.
- The entire date and/or time value must be enclosed in single quotes. Years must be expressed in four-digit form, and dates and times must contain two digits for each portion of the value. For example January 1, 2005, 1:05 AM would be expressed as:

  ```
  {ts '2005-01-01 01:05:00'}
  ```

Functions. A selection of built-in arithmetic, logical, string, date, and time SQL functions is provided. You can drag a function from the list into the expression, or you can enter any valid SQL function. See your database documentation for valid SQL functions. A list of standard functions is available at:

http://msdn2.microsoft.com/en-us/library/ms711813.aspx

Use Random Sampling. This option selects a random sample of cases from the data source. For large data sources, you may want to limit the number of cases to a small, representative sample, which can significantly reduce the time that it takes to run procedures. Native random sampling, if available for the data source, is faster than PASW Statistics random sampling, because PASW Statistics random sampling must still read the entire data source to extract a random sample.

- **Approximately.** Generates a random sample of approximately the specified percentage of cases. Since this routine makes an independent pseudorandom decision for each case, the percentage of cases selected can only approximate the specified percentage. The more cases there are in the data file, the closer the percentage of cases selected is to the specified percentage.
- **Exactly.** Selects a random sample of the specified number of cases from the specified total number of cases. If the total number of cases specified exceeds the total number of cases in the data file, the sample will contain proportionally fewer cases than the requested number.

Note: If you use random sampling, aggregation (available in distributed mode with PASW Statistics Server) is not available.

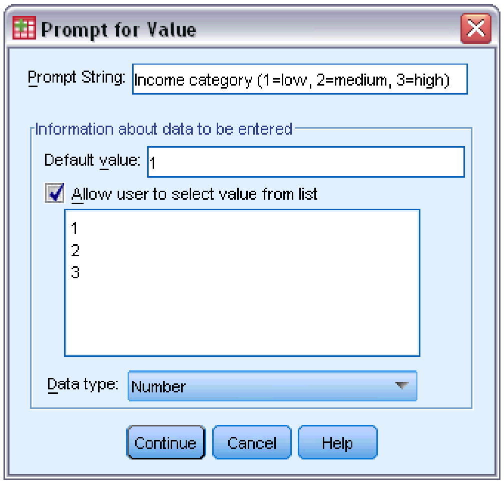
Prompt For Value. You can embed a prompt in your query to create a **parameter query**. When users run the query, they will be asked to enter information (based on what is specified here). You might want to do this if you need to see different views of the same data. For example, you may want to run the same query to see sales figures for different fiscal quarters.

▶ Place your cursor in any Expression cell, and click Prompt For Value to create a prompt.

Creating a Parameter Query

Use the Prompt for Value step to create a dialog box that solicits information from users each time someone runs your query. This feature is useful if you want to query the same data source by using different criteria.

Figure 3-7
Prompt for Value



Prompt for Value

Prompt String: Income category (1=low, 2=medium, 3=high)

Information about data to be entered

Default value: 1

☑ Allow user to select value from list

1
2
3

Data type: Number

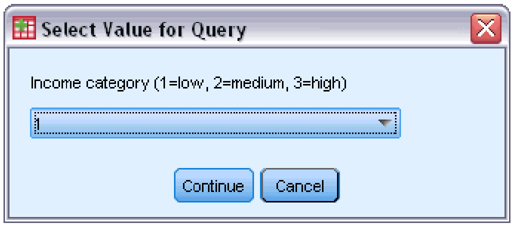
To build a prompt, enter a prompt string and a default value. The prompt string is displayed each time a user runs your query. The string should specify the kind of information to enter. If the user is not selecting from a list, the string should give hints about how the input should be formatted. An example is as follows: Enter a Quarter (Q1, Q2, Q3, ...).

Allow user to select value from list. If this check box is selected, you can limit the user to the values that you place here. Ensure that your values are separated by returns.

Data type. Choose the data type here (Number, String, or Date).

The final result looks like this:

Figure 3-8
User-defined prompt



Select Value for Query

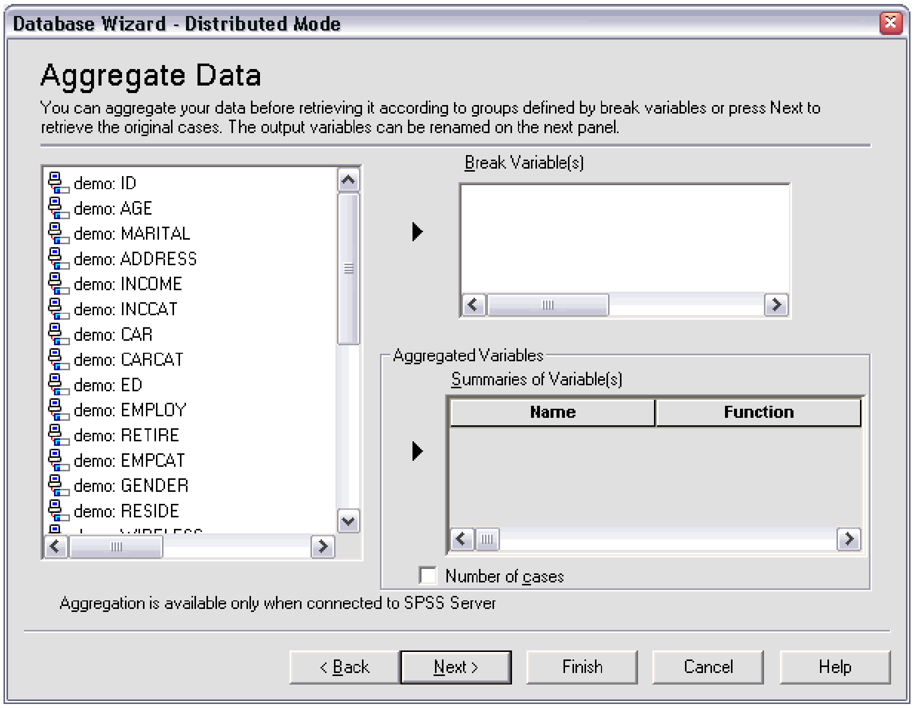
Income category (1=low, 2=medium, 3=high)

1

Aggregating Data

If you are in distributed mode, connected to a remote server (available with PASW Statistics Server), you can aggregate the data before reading it into PASW Statistics.

Figure 3-9
Database Wizard, aggregating data



Database Wizard - Distributed Mode

Aggregate Data

You can aggregate your data before retrieving it according to groups defined by break variables or press Next to retrieve the original cases. The output variables can be renamed on the next panel.

Aggregation is available only when connected to SPSS Server

You can also aggregate data after reading it into PASW Statistics, but preaggregating may save time for large data sources.

▶ To create aggregated data, select one or more break variables that define how cases are grouped.

▶ Select one or more aggregated variables.

▶ Select an aggregate function for each aggregate variable.

▶ Optionally, create a variable that contains the number of cases in each break group.

Note: If you use PASW Statistics random sampling, aggregation is not available.

Defining Variables

Variable names and labels. The complete database field (column) name is used as the variable label. Unless you modify the variable name, the Database Wizard assigns variable names to each column from the database in one of two ways:

- If the name of the database field forms a valid, unique variable name, the name is used as the variable name.
- If the name of the database field does not form a valid, unique variable name, a new, unique name is automatically generated.

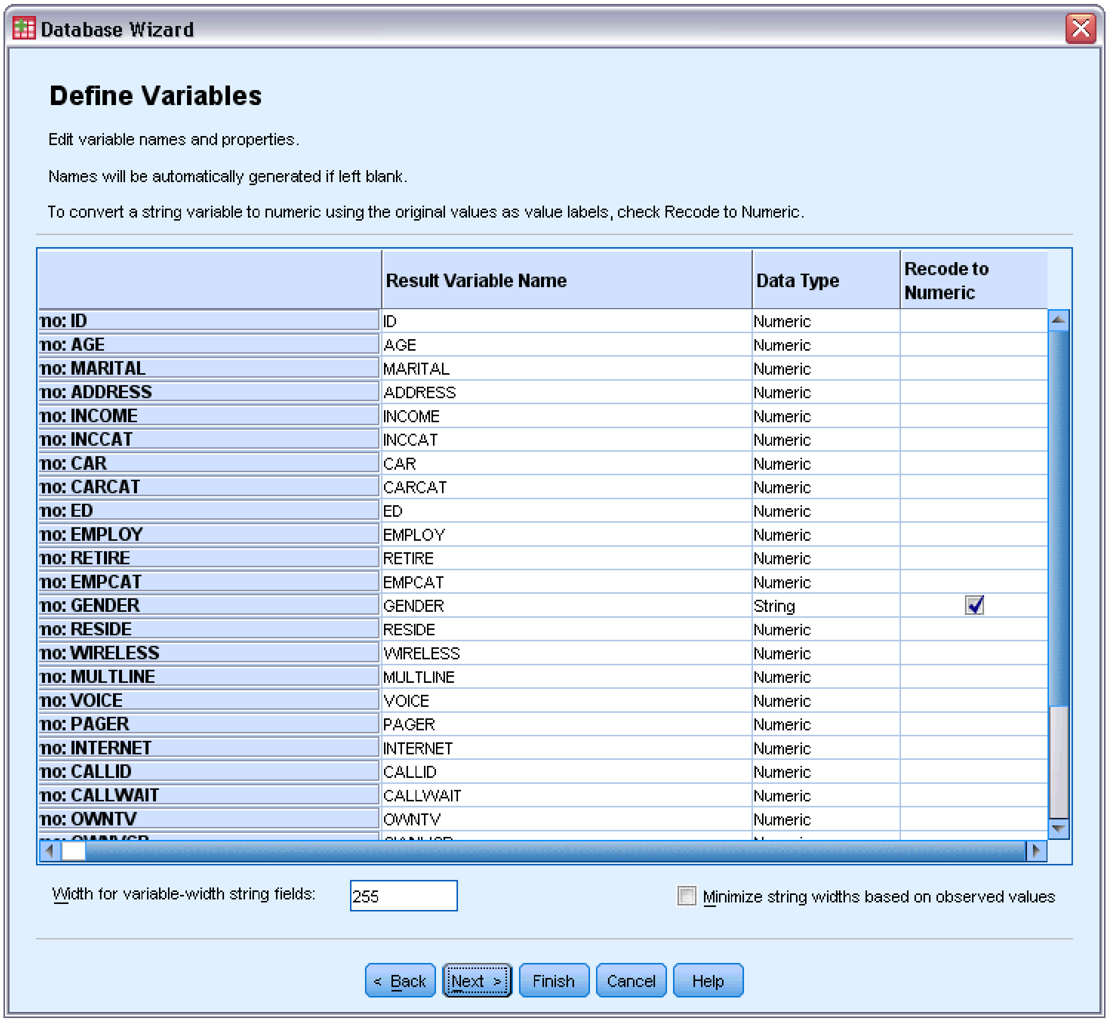
Click any cell to edit the variable name.

Converting strings to numeric values. Select the Recode to Numeric box for a string variable if you want to automatically convert it to a numeric variable. String values are converted to consecutive integer values based on alphabetical order of the original values. The original values are retained as value labels for the new variables.

Width for variable-width string fields. This option controls the width of variable-width string values. By default, the width is 255 bytes, and only the first 255 bytes (typically 255 characters in single-byte languages) will be read. The width can be up to 32,767 bytes. Although you probably don't want to truncate string values, you also don't want to specify an unnecessarily large value, which will cause processing to be inefficient.

Minimize string widths based on observed values. Automatically set the width of each string variable to the longest observed value.

Figure 3-10
Database Wizard, defining variables



Define Variables

Edit variable names and properties.

Names will be automatically generated if left blank.

To convert a string variable to numeric using the original values as value labels, check Recode to Numeric.

	Result Variable Name	Data Type	Recode to Numeric
no: ID	ID	Numeric	
no: AGE	AGE	Numeric	
no: MARITAL	MARITAL	Numeric	
no: ADDRESS	ADDRESS	Numeric	
no: INCOME	INCOME	Numeric	
no: INCCAT	INCCAT	Numeric	
no: CAR	CAR	Numeric	
no: CARCAT	CARCAT	Numeric	
no: ED	ED	Numeric	
no: EMPLOY	EMPLOY	Numeric	
no: RETIRE	RETIRE	Numeric	
no: EMPCAT	EMPCAT	Numeric	
no: GENDER	GENDER	String	☑
no: RESIDE	RESIDE	Numeric	
no: WIRELESS	WIRELESS	Numeric	
no: MULTLINE	MULTLINE	Numeric	
no: VOICE	VOICE	Numeric	
no: PAGER	PAGER	Numeric	
no: INTERNET	INTERNET	Numeric	
no: CALLID	CALLID	Numeric	
no: CALLWAIT	CALLWAIT	Numeric	
no: OWNTV	OWNTV	Numeric	

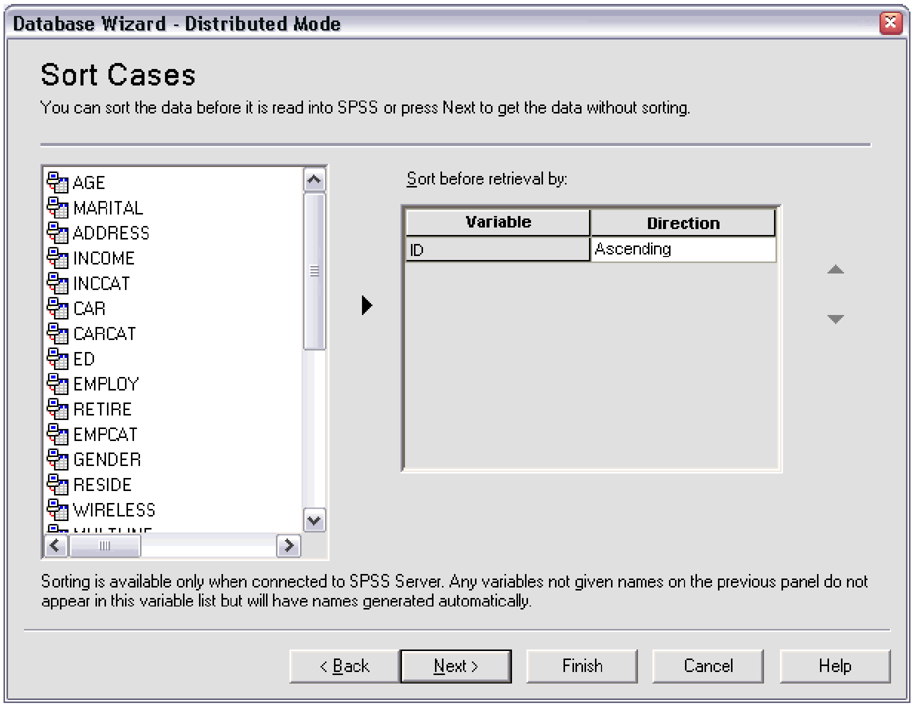
Width for variable-width string fields: 255

☐ Minimize string widths based on observed values

Sorting Cases

If you are in distributed mode, connected to a remote server (available with PASW Statistics Server), you can sort the data before reading it into PASW Statistics.

Figure 3-11
Database Wizard, sorting cases



Database Wizard - Distributed Mode

Sort Cases

You can sort the data before it is read into SPSS or press Next to get the data without sorting.

Sort before retrieval by:

Variable	Direction
ID	Ascending

Sorting is available only when connected to SPSS Server. Any variables not given names on the previous panel do not appear in this variable list but will have names generated automatically.

You can also sort data after reading it into PASW Statistics, but presorting may save time for large data sources.

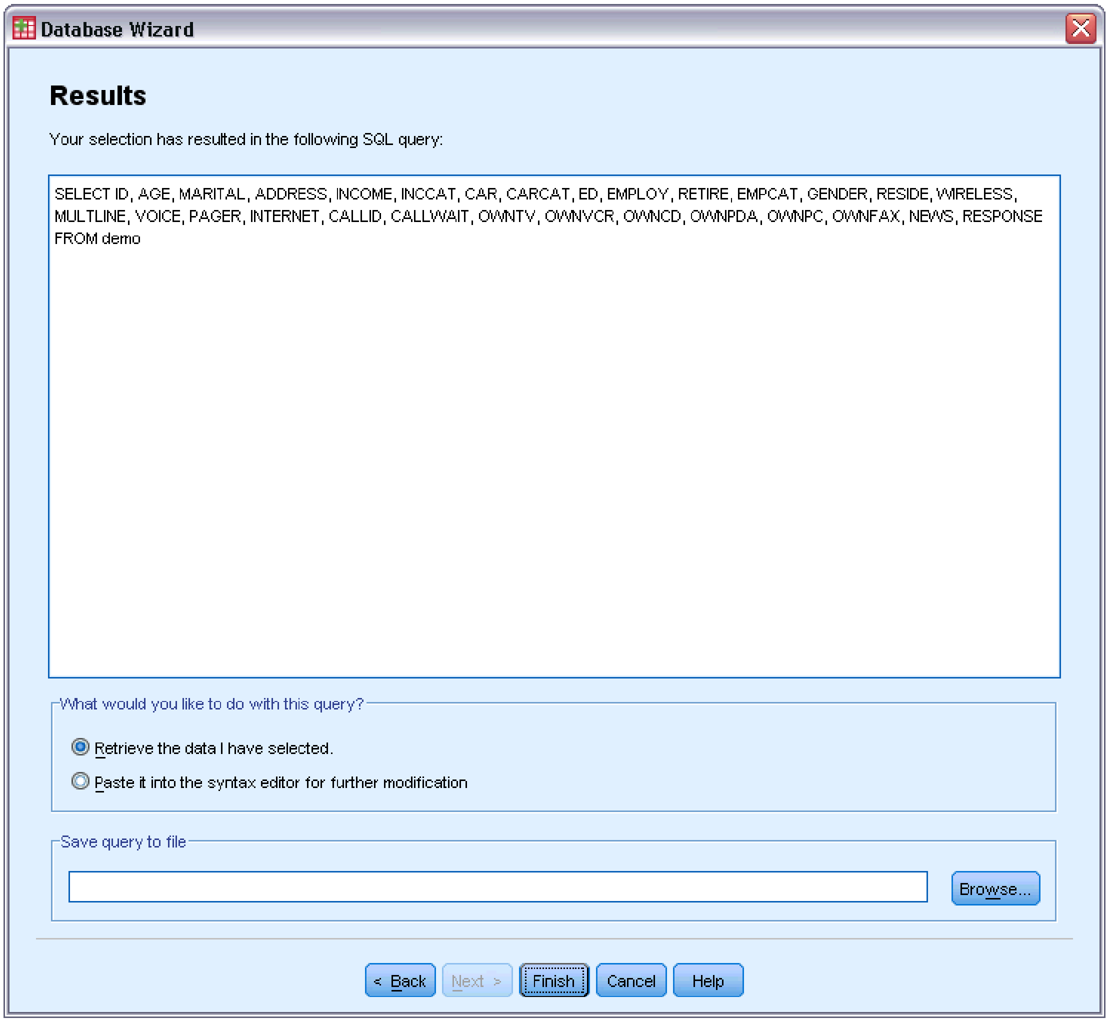
Results

The Results step displays the SQL Select statement for your query.

- You can edit the SQL Select statement before you run the query, but if you click the Back button to make changes in previous steps, the changes to the Select statement will be lost.
- To save the query for future use, use the Save query to file section.
- To paste complete `GET DATA` syntax into a syntax window, select Paste it into the syntax editor for further modification. Copying and pasting the Select statement from the Results window will not paste the necessary command syntax.

Note: The pasted syntax contains a blank space before the closing quote on each line of SQL that is generated by the wizard. These blanks are not superfluous. When the command is processed, all lines of the SQL statement are merged together in a very literal fashion. Without the space, there would be no space between the last character on one line and first character on the next line.

Figure 3-12
Database Wizard, results panel



Results

Your selection has resulted in the following SQL query:

```
SELECT ID, AGE, MARITAL, ADDRESS, INCOME, INCCAT, CAR, CARCAT, ED, EMPLOY, RETIRE, EMPCAT, GENDER, RESIDE, WIRELESS,
MULTLINE, VOICE, PAGER, INTERNET, CALLID, CALLWAIT, OWNTV, OWNVCR, OWNCD, OWNPDA, OWNPC, OWNFAX, NEWS, RESPONSE
FROM demo
```

What would you like to do with this query?

● Retrieve the data I have selected.

○ Paste it into the syntax editor for further modification

Save query to file

Text Wizard

The Text Wizard can read text data files formatted in a variety of ways:

- Tab-delimited files
- Space-delimited files
- Comma-delimited files
- Fixed-field format files

For delimited files, you can also specify other characters as delimiters between values, and you can specify multiple delimiters.

To Read Text Data Files

- ▶ From the menus choose:

  File

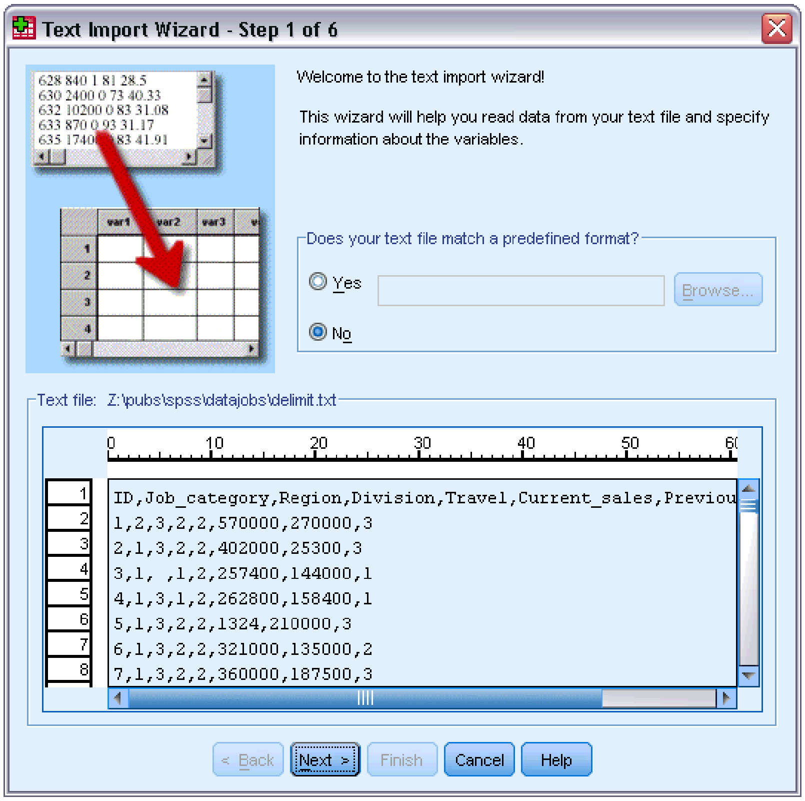
  Read Text Data...

- ▶ Select the text file in the Open Data dialog box.

- ▶ Follow the steps in the Text Wizard to define how to read the data file.

Text Wizard: Step 1

Figure 3-13
Text Wizard: Step 1



Text Import Wizard - Step 1 of 6

Welcome to the text import wizard!

This wizard will help you read data from your text file and specify information about the variables.

Does your text file match a predefined format?

○ Yes   Browse...

● No

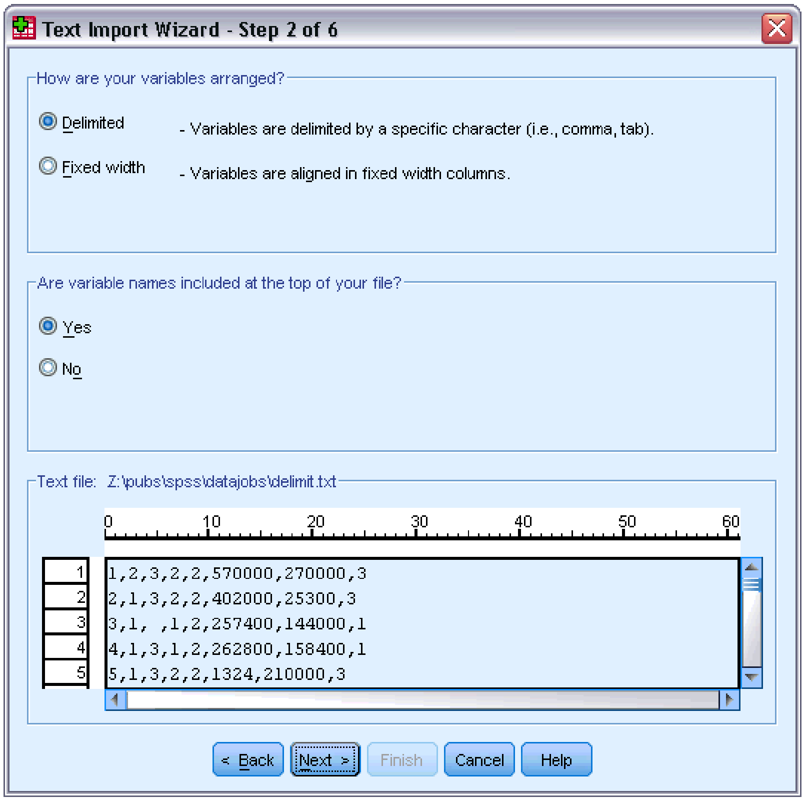
Text file: Z:\pubs\spss\datajobs\delimit.txt

```
0         10        20        30        40        50        60
1 ID,Job_category,Region,Division,Travel,Current_sales,Previou
2 1,2,3,2,2,570000,270000,3
3 2,1,3,2,2,402000,25300,3
4 3,1, ,1,2,257400,144000,1
5 4,1,3,1,2,262800,158400,1
6 5,1,3,2,2,1324,210000,3
7 6,1,3,2,2,321000,135000,2
8 7,1,3,2,2,360000,187500,3
```

The text file is displayed in a preview window. You can apply a predefined format (previously saved from the Text Wizard) or follow the steps in the Text Wizard to specify how the data should be read.

Text Wizard: Step 2

Figure 3-14
Text Wizard: Step 2



How are your variables arranged?

● Delimited - Variables are delimited by a specific character (i.e., comma, tab).

○ Fixed width - Variables are aligned in fixed width columns.

Are variable names included at the top of your file?

● Yes

○ No

Text file: Z:\pubs\spss\datajobs\delimit.txt

```
1  1,2,3,2,2,570000,270000,3
2  2,1,3,2,2,402000,25300,3
3  3,1, ,1,2,257400,144000,1
4  4,1,3,1,2,262800,158400,1
5  5,1,3,2,2,1324,210000,3
```

This step provides information about variables. A variable is similar to a field in a database. For example, each item in a questionnaire is a variable.

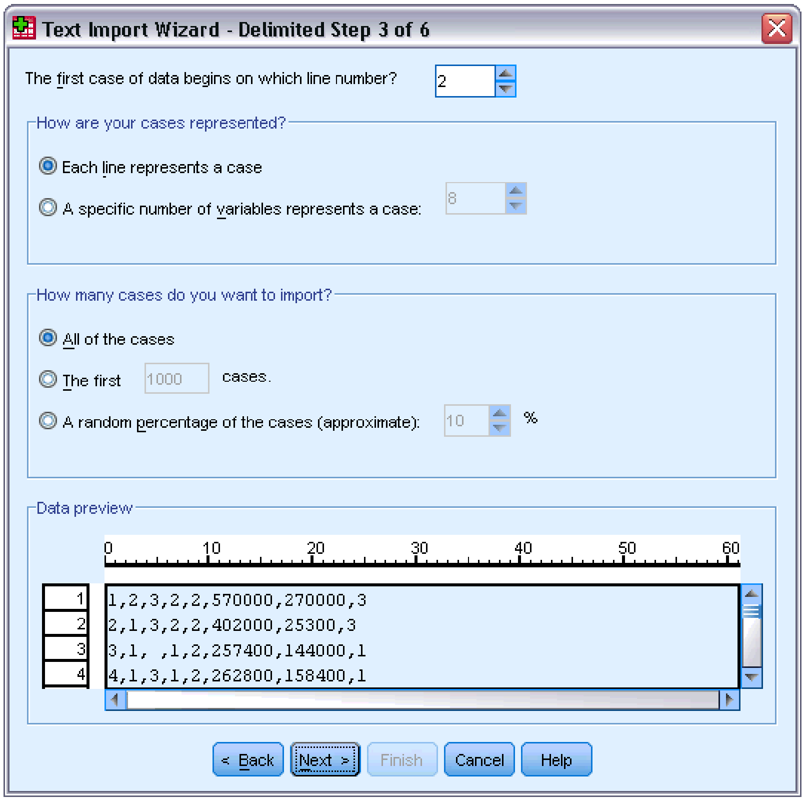
How are your variables arranged? To read your data properly, the Text Wizard needs to know how to determine where the data value for one variable ends and the data value for the next variable begins. The arrangement of variables defines the method used to differentiate one variable from the next.

- **Delimited.** Spaces, commas, tabs, or other characters are used to separate variables. The variables are recorded in the same order for each case but not necessarily in the same column locations.
- **Fixed width.** Each variable is recorded in the same column location on the same record (line) for each case in the data file. No delimiter is required between variables. In fact, in many text data files generated by computer programs, data values may appear to run together without even spaces separating them. The column location determines which variable is being read.

Are variable names included at the top of your file? If the first row of the data file contains descriptive labels for each variable, you can use these labels as variable names. Values that don’t conform to variable naming rules are converted to valid variable names.

Text Wizard: Step 3 (Delimited Files)

Figure 3-15
Text Wizard: Step 3 (for delimited files)



Text Import Wizard - Delimited Step 3 of 6

The first case of data begins on which line number? 2

How are your cases represented?

● Each line represents a case

○ A specific number of variables represents a case: 8

How many cases do you want to import?

● All of the cases

○ The first 1000 cases.

○ A random percentage of the cases (approximate): 10 %

Data preview

```
1  1,2,3,2,2,570000,270000,3
2  2,1,3,2,2,402000,25300,3
3  3,1, ,1,2,257400,144000,1
4  4,1,3,1,2,262800,158400,1
```

This step provides information about cases. A case is similar to a record in a database. For example, each respondent to a questionnaire is a case.

The first case of data begins on which line number? Indicates the first line of the data file that contains data values. If the top line(s) of the data file contain descriptive labels or other text that does not represent data values, this will *not* be line 1.

How are your cases represented? Controls how the Text Wizard determines where each case ends and the next one begins.

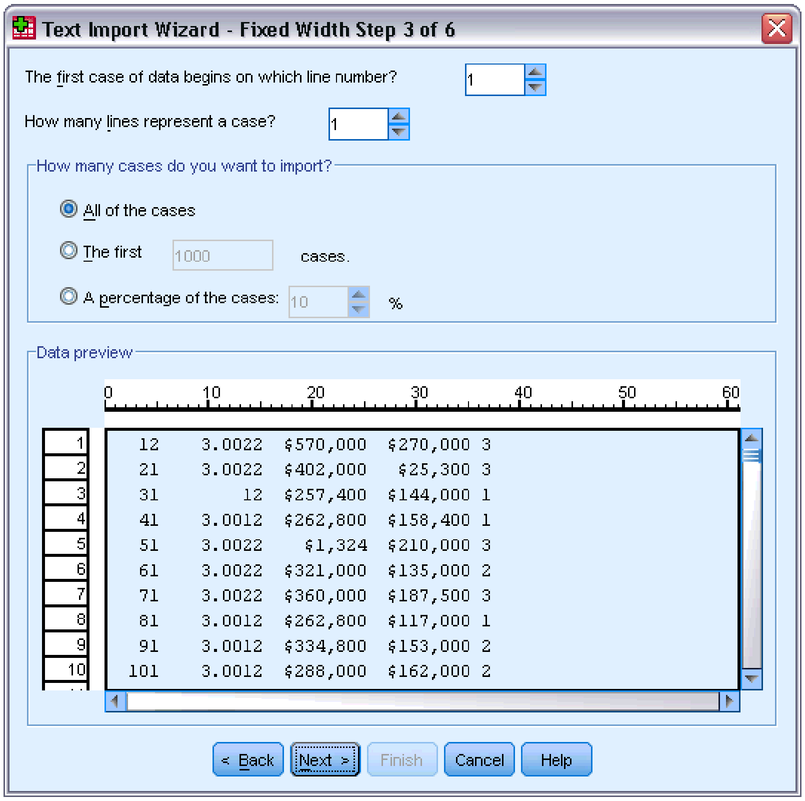
- **Each line represents a case.** Each line contains only one case. It is fairly common for each case to be contained on a single line (row), even though this can be a very long line for data files with a large number of variables. If not all lines contain the same number of data values, the number of variables for each case is determined by the line with the greatest number of data values. Cases with fewer data values are assigned missing values for the additional variables.
- **A specific number of variables represents a case.** The specified number of variables for each case tells the Text Wizard where to stop reading one case and start reading the next. Multiple cases can be contained on the same line, and cases can start in the middle of one line and be continued on the next line. The Text Wizard determines the end of each case based on the number of values read, regardless of the number of lines. Each case must contain data

values (or missing values indicated by delimiters) for all variables, or the data file will be read incorrectly.

How many cases do you want to import? You can import all cases in the data file, the first *n* cases (*n* is a number you specify), or a random sample of a specified percentage. Since the random sampling routine makes an independent pseudo-random decision for each case, the percentage of cases selected can only approximate the specified percentage. The more cases there are in the data file, the closer the percentage of cases selected is to the specified percentage.

Text Wizard: Step 3 (Fixed-Width Files)

Figure 3-16
Text Wizard: Step 3 (for fixed-width files)



Text Import Wizard - Fixed Width Step 3 of 6

The first case of data begins on which line number? 1

How many lines represent a case? 1

How many cases do you want to import?

● All of the cases

○ The first 1000 cases.

○ A percentage of the cases: 10 %

Data preview

```
       0         10        20        30        40        50        60
 1      12    3.0022  $570,000  $270,000 3
 2      21    3.0022  $402,000   $25,300 3
 3      31        12  $257,400  $144,000 1
 4      41    3.0012  $262,800  $158,400 1
 5      51    3.0022    $1,324  $210,000 3
 6      61    3.0022  $321,000  $135,000 2
 7      71    3.0022  $360,000  $187,500 3
 8      81    3.0012  $262,800  $117,000 1
 9      91    3.0012  $334,800  $153,000 2
10     101    3.0012  $288,000  $162,000 2
```

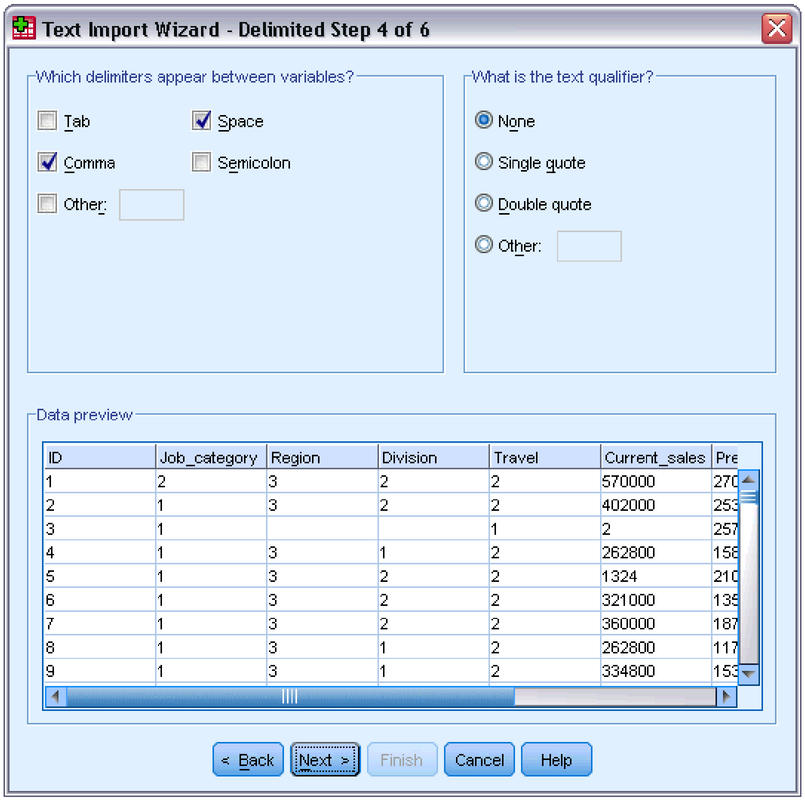
This step provides information about cases. A case is similar to a record in a database. For example, each respondent to questionnaire is a case.

The first case of data begins on which line number? Indicates the first line of the data file that contains data values. If the top line(s) of the data file contain descriptive labels or other text that does not represent data values, this will *not* be line 1.

How many lines represent a case? Controls how the Text Wizard determines where each case ends and the next one begins. Each variable is defined by its line number within the case and its column location. You need to specify the number of lines for each case to read the data correctly.

How many cases do you want to import? You can import all cases in the data file, the first *n* cases (*n* is a number you specify), or a random sample of a specified percentage. Since the random sampling routine makes an independent pseudo-random decision for each case, the percentage of cases selected can only approximate the specified percentage. The more cases there are in the data file, the closer the percentage of cases selected is to the specified percentage.

Text Wizard: Step 4 (Delimited Files)



Which delimiters appear between variables?

□ Tab ☑ Space
☑ Comma □ Semicolon
□ Other:

What is the text qualifier?

● None
○ Single quote
○ Double quote
○ Other:

Data preview

ID	Job_category	Region	Division	Travel	Current_sales	Pre
1	2	3	2	2	570000	270
2	1	3	2	2	402000	253
3	1			1	2	257
4	1	3	1	2	262800	158
5	1	3	2	2	1324	210
6	1	3	2	2	321000	135
7	1	3	2	2	360000	187
8	1	3	1	2	262800	117
9	1	3	1	2	334800	153

Figure 3-17
Text Wizard: Step 4 (for delimited files)

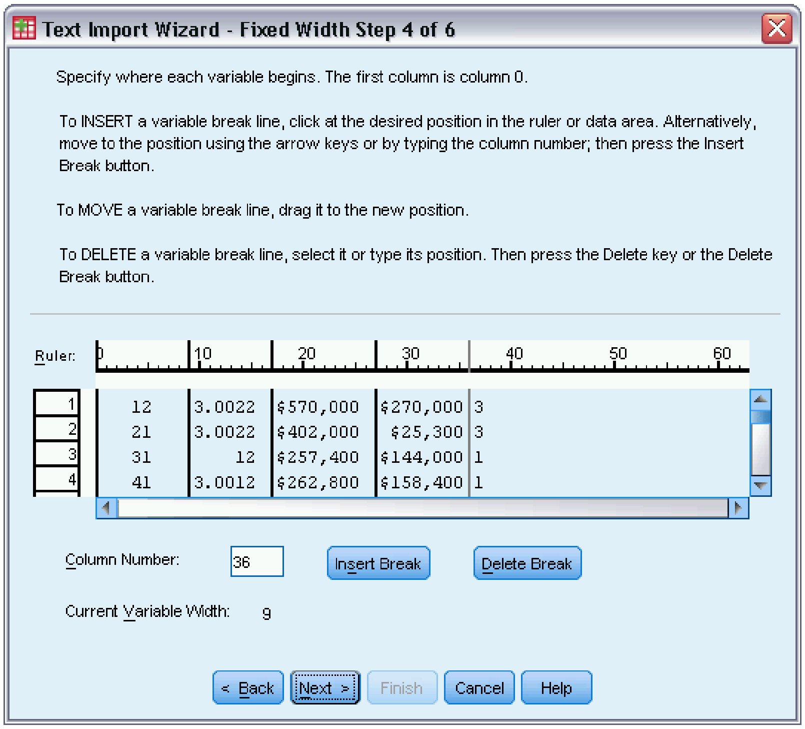
This step displays the Text Wizard's best guess on how to read the data file and allows you to modify how the Text Wizard will read variables from the data file.

Which delimiters appear between variables? Indicates the characters or symbols that separate data values. You can select any combination of spaces, commas, semicolons, tabs, or other characters. Multiple, consecutive delimiters without intervening data values are treated as missing values.

What is the text qualifier? Characters used to enclose values that contain delimiter characters. For example, if a comma is the delimiter, values that contain commas will be read incorrectly unless there is a text qualifier enclosing the value, preventing the commas in the value from being interpreted as delimiters between values. CSV-format data files exported from Excel use a double quotation mark (") as a text qualifier. The text qualifier appears at both the beginning and the end of the value, enclosing the entire value.

Text Wizard: Step 4 (Fixed-Width Files)

Figure 3-18
Text Wizard: Step 4 (for fixed-width files)



Text Import Wizard - Fixed Width Step 4 of 6

Specify where each variable begins. The first column is column 0.

To INSERT a variable break line, click at the desired position in the ruler or data area. Alternatively, move to the position using the arrow keys or by typing the column number; then press the Insert Break button.

To MOVE a variable break line, drag it to the new position.

To DELETE a variable break line, select it or type its position. Then press the Delete key or the Delete Break button.

```
Ruler: 0        10       20        30        40        50        60
1        12 |3.0022 |$570,000 |$270,000 |3
2        21 |3.0022 |$402,000 |  $25,300 |3
3        31 |     12 |$257,400 |$144,000 |1
4        41 |3.0012 |$262,800 |$158,400 |1
```

Column Number: 36

Current Variable Width: 9

This step displays the Text Wizard’s best guess on how to read the data file and allows you to modify how the Text Wizard will read variables from the data file. Vertical lines in the preview window indicate where the Text Wizard currently thinks each variable begins in the file.

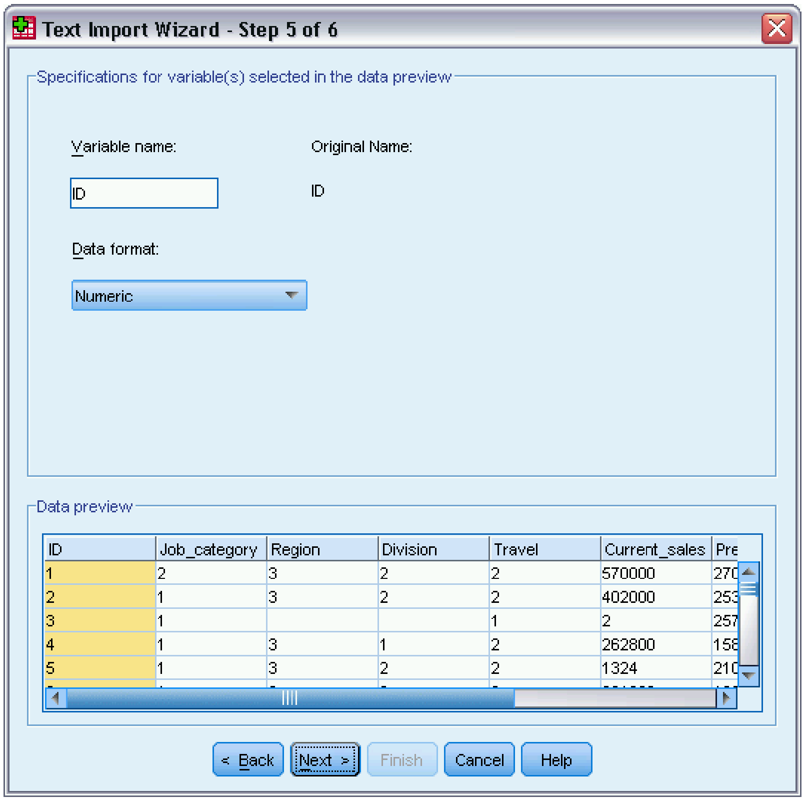
Insert, move, and delete variable break lines as necessary to separate variables. If multiple lines are used for each case, the data will be displayed as one line for each case, with subsequent lines appended to the end of the line.

Notes:

For computer-generated data files that produce a continuous stream of data values with no intervening spaces or other distinguishing characteristics, it may be difficult to determine where each variable begins. Such data files usually rely on a data definition file or some other written description that specifies the line and column location for each variable.

Text Wizard: Step 5

Figure 3-19
Text Wizard: Step 5



Text Import Wizard - Step 5 of 6
Specifications for variable(s) selected in the data preview

Variable name: ID

Original Name: ID

Data format: Numeric

ID	Job_category	Region	Division	Travel	Current_sales	Pre
1	2	3	2	2	570000	270
2	1	3	2	2	402000	253
3	1			1	2	257
4	1	3	1	2	262800	158
5	1	3	2	2	1324	210

< Back
Next >
Finish
Cancel
Help

This steps controls the variable name and the data format that the Text Wizard will use to read each variable and which variables will be included in the final data file.

Variable name. You can overwrite the default variable names with your own variable names. If you read variable names from the data file, the Text Wizard will automatically modify variable names that don't conform to variable naming rules. Select a variable in the preview window and then enter a variable name.

Data format. Select a variable in the preview window and then select a format from the drop-down list. Shift-click to select multiple contiguous variables or Ctrl-click to select multiple noncontiguous variables.

The default format is determined from the data values in the first 250 rows. If more than one format (e.g., numeric, date, string) is encountered in the first 250 rows, the default format is set to string.

Text Wizard Formatting Options

Formatting options for reading variables with the Text Wizard include:

Do not import. Omit the selected variable(s) from the imported data file.

Numeric. Valid values include numbers, a leading plus or minus sign, and a decimal indicator.

String. Valid values include virtually any keyboard characters and embedded blanks. For delimited files, you can specify the number of characters in the value, up to a maximum of 32,767. By default, the Text Wizard sets the number of characters to the longest string value encountered for the selected variable(s) in the first 250 rows of the file. For fixed-width files, the number of characters in string values is defined by the placement of variable break lines in step 4.

Date/Time. Valid values include dates of the general format *dd-mm-yyyy*, *mm/dd/yyyy*, *dd.mm.yyyy*, *yyyy/mm/dd*, *hh:mm:ss*, and a variety of other date and time formats. Months can be represented in digits, Roman numerals, or three-letter abbreviations, or they can be fully spelled out. Select a date format from the list.

Dollar. Valid values are numbers with an optional leading dollar sign and optional commas as thousands separators.

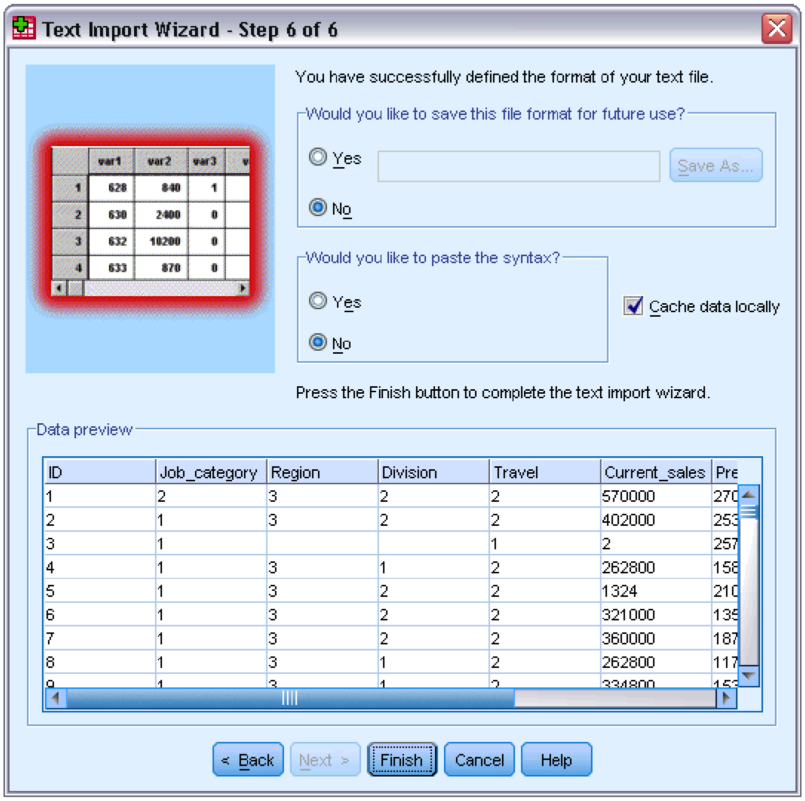
Comma. Valid values include numbers that use a period as a decimal indicator and commas as thousands separators.

Dot. Valid values include numbers that use a comma as a decimal indicator and periods as thousands separators.

Note: Values that contain invalid characters for the selected format will be treated as missing. Values that contain any of the specified delimiters will be treated as multiple values.

Text Wizard: Step 6

Figure 3-20
Text Wizard: Step 6



Text Import Wizard - Step 6 of 6

You have successfully defined the format of your text file.

Would you like to save this file format for future use?

○ Yes Save As...

● No

Would you like to paste the syntax?

○ Yes

● No

☑ Cache data locally

Press the Finish button to complete the text import wizard.

Data preview

ID	Job_category	Region	Division	Travel	Current_sales	Pre
1	2	3	2	2	570000	270
2	1	3	2	2	402000	253
3	1			1	2	257
4	1	3	1	2	262800	158
5	1	3	2	2	1324	210
6	1	3	2	2	321000	135
7	1	3	2	2	360000	187
8	1	3	1	2	262800	117
9	1	3	1	2	334800	15

This is the final step of the Text Wizard. You can save your specifications in a file for use when importing similar text data files. You can also paste the syntax generated by the Text Wizard into a syntax window. You can then customize and/or save the syntax for use in other sessions or in production jobs.

Cache data locally. A data cache is a complete copy of the data file, stored in temporary disk space. Caching the data file can improve performance.

Reading PASW Data Collection Data

On Microsoft Windows operating systems, you can read data from PASW Data Collection products. (*Note*: This feature is only available with PASW Statistics installed on Microsoft Windows operating systems.)

To read PASW Data Collection data sources, you must have the following items installed:

- .NET framework. To obtain the most recent version of the .NET framework, go to *http://www.microsoft.com/net*.
- PASW Reports for Surveys Components. A version that is compatible with this release can be installed from the installation media. If you are using PASW Statistics Developer, you can download a compatible version from the Downloads tab at www.spss.com/statistics (*http://www.spss.com/statistics/*).

You can read PASW Data Collection data sources only in local analysis mode. This feature is not available in distributed analysis mode using PASW Statistics Server.

To read data from a PASW Data Collection data source:

▶ In any open PASW Statistics window, from the menus choose:

File
  Open PASW Data Collection Data

▶ On the Connection tab of Data Link Properties, specify the metadata file, the case data type, and the case data file.

▶ Click OK.

▶ In the PASW Data Collection Data Import dialog box, select the variables that you want to include and select any case selection criteria.

▶ Click OK to read the data.

Data Link Properties Connection Tab

To read a PASW Data Collection data source, you need to specify:

Metadata Location. The metadata document file (*.mdd*) that contains questionnaire definition information.

Case Data Type. The format of the case data file. Available formats include:

- Quancept Data File (DRS). Case data in a Quancept *.drs*, *.drz*, or *.dru* file.

- Quanvert Database. Case data in a Quanvert database.
- PASW Data Collection Database (MS SQL Server). Case data in a relational database in SQL Server.
- PASW Data Collection XML Data File. Case data in an XML file.

Case Data Location. The file that contains the case data. The format of this file must be consistent with the selected case data type.

Note: The extent to which other settings on the Connection tab or any settings on the other Data Link Properties tabs may or may not affect reading PASW Data Collection data into PASW Statistics is not known, so we recommend that you do not change any of them.

Select Variables Tab

You can select a subset of variables to read. By default, all standard variables in the data source are displayed and selected.

- **Show System variables.** Displays any “system” variables, including variables that indicate interview status (*in progress*, *completed*, *finish date*, and so on). You can then select any system variables that you want to include. By default, all system variables are excluded.
- **Show Codes variables.** Displays any variables that represent codes that are used for open-ended “Other” responses for categorical variables. You can then select any Codes variables that you want to include. By default, all Codes variables are excluded.
- **Show SourceFile variables.** Displays any variables that contain filenames of images of scanned responses. You can then select any SourceFile variables that you want to include. By default, all SourceFile variables are excluded.

Case Selection Tab

For PASW Data Collection data sources that contain system variables, you can select cases based on a number of system variable criteria. You do not need to include the corresponding system variables in the list of variables to read, but the necessary system variables must exist in the source data to apply the selection criteria. If the necessary system variables do not exist in the source data, the corresponding selection criteria are ignored.

Data collection status. You can select respondent data, test data, or both. You can also select cases based on any combination of the following interview status parameters:

- Completed successfully
- Active/in progress
- Timed out
- Stopped by script
- Stopped by respondent
- Interview system shutdown
- Signal (terminated by a signal statement in the script)

Data collection finish date. You can select cases based on the data collection finish date.

- **Start Date.** Cases for which data collection finished on or after the specified date are included.
- **End Date.** Cases for which data collection finished before the specified date are included. This does *not* include cases for which data collection finished on the end date.
- If you specify both a start date and end date, this defines a range of finish dates from the start date to (but not including) the end date.

File Information

A data file contains much more than raw data. It also contains any variable definition information, including:

- Variable names
- Variable formats
- Descriptive variable and value labels

This information is stored in the dictionary portion of the data file. The Data Editor provides one way to view the variable definition information. You can also display complete dictionary information for the active dataset or any other data file.

To Display Data File Information

▶ From the menus in the Data Editor window choose:

File
  Display Data File Information

▶ For the currently open data file, choose Working File.

▶ For other data files, choose External File, and then select the data file.

The data file information is displayed in the Viewer.

Saving Data Files

In addition to saving data files in PASW Statistics format, you can save data in a wide variety of external formats, including:

- Excel and other spreadsheet formats
- Tab-delimited and CSV text files
- SAS
- Stata
- Database tables

To Save Modified Data Files

▶ Make the Data Editor the active window (click anywhere in the window to make it active).

- From the menus choose:

  File
  Save

The modified data file is saved, overwriting the previous version of the file.

Note: A data file saved in Unicode mode cannot be read by versions of PASW Statistics prior to 16.0. To save a Unicode data file in a format that can be read by earlier releases, open the file in code page mode and re-save it. The file will be saved in the encoding based on the current locale. Some data loss may occur if the file contains characters not recognized by the current locale. For information on switching between Unicode mode and code page mode, see General Options on p. 291.

Saving Data Files in External Formats

- Make the Data Editor the active window (click anywhere in the window to make it active).

- From the menus choose:

  File
  Save As...

- Select a file type from the drop-down list.

- Enter a filename for the new data file.

To write variable names to the first row of a spreadsheet or tab-delimited data file:

- Click Write variable names to spreadsheet in the Save Data As dialog box.

To save value labels instead of data values in Excel files:

- Click Save value labels where defined instead of data values in the Save Data As dialog box.

To save value labels to a SAS syntax file (active only when a SAS file type is selected):

- Click Save value labels into a .sas file in the Save Data As dialog box.

For information on exporting data to database tables, see Exporting to a Database on p. 46.

For information on exporting data for use in PASW Data Collection applications, see Exporting to PASW Data Collection on p. 58.

Saving Data: Data File Types

You can save data in the following formats:

PASW Statistics (*.sav). PASW Statistics format.

- Data files saved in PASW Statistics format cannot be read by versions of the software prior to version 7.5. Data files saved in Unicode mode cannot be read by releases of PASW Statistics prior to version 16.0 For more information, see the topic General Options in Chapter 16 on p. 291.
- When using data files with variable names longer than eight bytes in version 10.x or 11.x, unique, eight-byte versions of variable names are used—but the original variable names are preserved for use in release 12.0 or later. In releases prior to 10.0, the original long variable names are lost if you save the data file.
- When using data files with string variables longer than 255 bytes in versions prior to release 13.0, those string variables are broken up into multiple 255-byte string variables.

Version 7.0 (*.sav). Version 7.0 format. Data files saved in version 7.0 format can be read by version 7.0 and earlier versions but do not include defined multiple response sets or Data Entry for Windows information.

SPSS/PC+ (*.sys). SPSS/PC+ format. If the data file contains more than 500 variables, only the first 500 will be saved. For variables with more than one defined user-missing value, additional user-missing values will be recoded into the first defined user-missing value. This format is available only on Windows operating systems.

PASW Statistics Portable (*.por). Portable format that can be read by other versions of PASW Statistics and versions on other operating systems. Variable names are limited to eight bytes and are automatically converted to unique eight-byte names if necessary. In most cases, saving data in portable format is no longer necessary, since PASW Statistics data files should be platform/operating system independent. You cannot save data files in portable file in Unicode mode. For more information, see the topic General Options in Chapter 16 on p. 291.

Tab-delimited (*.dat). Text files with values separated by tabs. (*Note*: Tab characters embedded in string values are preserved as tab characters in the tab-delimited file. No distinction is made between tab characters embedded in values and tab characters that separate values.)

Comma-delimited (*.csv). Text files with values separated by commas or semicolons. If the current PASW Statistics decimal indicator is a period, values are separated by commas. If the current decimal indicator is a comma, values are separated by semicolons.

Fixed ASCII (*.dat). Text file in fixed format, using the default write formats for all variables. There are no tabs or spaces between variable fields.

Excel 2007 (*.xlsx). Microsoft Excel 2007 XLSX-format workbook. The maximum number of variables is 16,000; any additional variables beyond the first 16,000 are dropped. If the dataset contains more than one million cases, multiple sheets are created in the workbook.

Excel 97 through 2003 (*.xls). Microsoft Excel 97 workbook. The maximum number of variables is 256; any additional variables beyond the first 256 are dropped. If the dataset contains more than 65,356 cases, multiple sheets are created in the workbook.

Excel 2.1 (*.xls). Microsoft Excel 2.1 spreadsheet file. The maximum number of variables is 256, and the maximum number of rows is 16,384.

1-2-3 Release 3.0 (*.wk3). Lotus 1-2-3 spreadsheet file, release 3.0. The maximum number of variables that you can save is 256.

1-2-3 Release 2.0 (*.wk1). Lotus 1-2-3 spreadsheet file, release 2.0. The maximum number of variables that you can save is 256.

1-2-3 Release 1.0 (*.wks). Lotus 1-2-3 spreadsheet file, release 1A. The maximum number of variables that you can save is 256.

SYLK (*.slk). Symbolic link format for Microsoft Excel and Multiplan spreadsheet files. The maximum number of variables that you can save is 256.

dBASE IV (*.dbf). dBASE IV format.

dBASE III (*.dbf). dBASE III format.

dBASE II (*.dbf). dBASE II format.

SAS v9+ Windows (*.sas7bdat). SAS versions 9 for Windows.

SAS v9+ UNIX (*.sas7bdat). SAS versions 9 for UNIX.

SAS v7-8 Windows short extension (*.sd7). SAS versions 7–8 for Windows short filename format.

SAS v7-8 Windows long extension (*.sas7bdat). SAS versions 7–8 for Windows long filename format.

SAS v7-8 for UNIX (*.sas7bdat). SAS v8 for UNIX.

SAS v6 for Windows (*.sd2). SAS v6 file format for Windows/OS2.

SAS v6 for UNIX (*.ssd01). SAS v6 file format for UNIX (Sun, HP, IBM).

SAS v6 for Alpha/OSF (*.ssd04). SAS v6 file format for Alpha/OSF (DEC UNIX).

SAS Transport (*.xpt). SAS transport file.

Stata Version 8 Intercooled (*.dta).

Stata Version 8 SE (*.dta).

Stata Version 7 Intercooled (*.dta).

Stata Version 7 SE (*.dta).

Stata Version 6 (*.dta).

Stata Versions 4–5 (*.dta).

Saving File Options

For spreadsheet, tab-delimited files, and comma-delimited files, you can write variable names to the first row of the file.

Saving Data Files in Excel Format

You can save your data in one of three Microsoft Excel file formats. Excel 2.1, Excel 97, and Excel 2007.

- Excel 2.1 and Excel 97 are limited to 256 columns; so only the first 256 variables are included.
- Excel 2007 is limited to 16,000 columns; so only the first 16,000 variables are included.

- Excel 2.1 is limited to 16,384 rows; so only the first 16,384 cases are included.
- Excel 97 and Excel 2007 also have limits on the number of rows per sheet, but workbooks can have multiple sheets, and multiple sheets are created if the single-sheet maximum is exceeded.

Variable Types

The following table shows the variable type matching between the original data in PASW Statistics and the exported data in Excel.

PASW Statistics Variable Type	Excel Data Format
Numeric	0.00; #,##0.00; ...
Comma	0.00; #,##0.00; ...
Dollar	$#,##0_); ...
Date	d-mmm-yyyy
Time	hh:mm:ss
String	General

Saving Data Files in SAS Format

Special handling is given to various aspects of your data when saved as a SAS file. These cases include:

- Certain characters that are allowed in PASW Statistics variable names are not valid in SAS, such as @, #, and $. These illegal characters are replaced with an underscore when the data are exported.
- PASW Statistics variable names that contain multibyte characters (for example, Japanese or Chinese characters) are converted to variables names of the general form *Vnnn*, where *nnn* is an integer value.
- PASW Statistics variable labels containing more than 40 characters are truncated when exported to a SAS v6 file.
- Where they exist, PASW Statistics variable labels are mapped to the SAS variable labels. If no variable label exists in the PASW Statistics data, the variable name is mapped to the SAS variable label.
- SAS allows only one value for system-missing, whereas PASW Statistics allows numerous user-missing values in addition to system-missing. As a result, all user-missing values in PASW Statistics are mapped to a single system-missing value in the SAS file.
- SAS 6-8 data files are saved in the current PASW Statistics locale encoding, regardless of current mode (Unicode or code page). In Unicode mode, SAS 9 files are saved in UTF-8 format. In code page mode, SAS 9 files are saved in the current locale encoding.
- A maximum of 32,767 variables can be saved to SAS 6-8.

Save Value Labels

You have the option of saving the values and value labels associated with your data file to a SAS syntax file. This syntax file contains `proc format` and `proc datasets` commands that can be run in SAS to create a SAS format catalog file.

This feature is not supported for the SAS transport file.

Variable Types

The following table shows the variable type matching between the original data in PASW Statistics and the exported data in SAS.

PASW Statistics Variable Type	SAS Variable Type	SAS Data Format
Numeric	Numeric	12
Comma	Numeric	12
Dot	Numeric	12
Scientific Notation	Numeric	12
Date	Numeric	(Date) for example, MMDDYY10, ...
Date (Time)	Numeric	Time18
Dollar	Numeric	12
Custom Currency	Numeric	12
String	Character	$8

Saving Data Files in Stata Format

- Data can be written in Stata 5–8 format and in both Intercooled and SE format (versions 7 and 8 only).
- Data files that are saved in Stata 5 format can be read by Stata 4.
- The first 80 bytes of variable labels are saved as Stata variable labels.
- For numeric variables, the first 80 bytes of value labels are saved as Stata value labels. Value labels are dropped for string variables, non-integer numeric values, and numeric values greater than an absolute value of 2,147,483,647.
- For versions 7 and 8, the first 32 bytes of variable names in case-sensitive form are saved as Stata variable names. For earlier versions, the first eight bytes of variable names are saved as Stata variable names. Any characters other than letters, numbers, and underscores are converted to underscores.
- PASW Statistics variable names that contain multibyte characters (for example, Japanese or Chinese characters) are converted to variable names of the general form *Vnnn*, where *nnn* is an integer value.
- For versions 5–6 and Intercooled versions 7–8, the first 80 bytes of string values are saved. For Stata SE 7–8, the first 244 bytes of string values are saved.
- For versions 5–6 and Intercooled versions 7–8, only the first 2,047 variables are saved. For Stata SE 7–8, only the first 32,767 variables are saved.

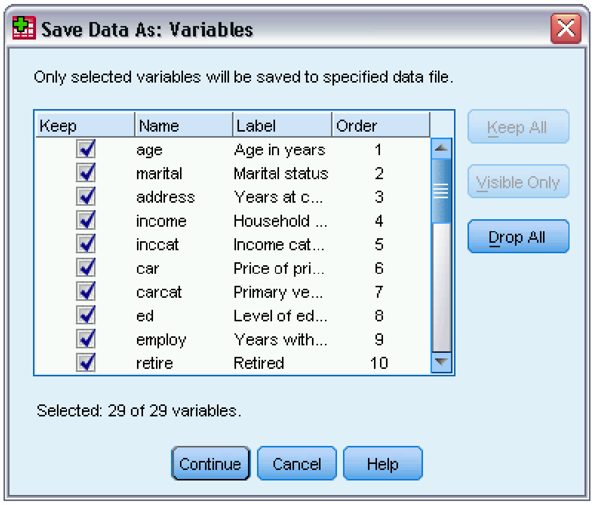
PASW Statistics Variable Type	Stata Variable Type	Stata Data Format
Numeric	Numeric	g
Comma	Numeric	g
Dot	Numeric	g
Scientific Notation	Numeric	g

PASW Statistics Variable Type	Stata Variable Type	Stata Data Format
Date*, Datetime	Numeric	D_m_Y
Time, DTime	Numeric	g (number of seconds)
Wkday	Numeric	g (1–7)
Month	Numeric	g (1–12)
Dollar	Numeric	g
Custom Currency	Numeric	g
String	String	s

*Date, Adate, Edate, SDate, Jdate, Qyr, Moyr, Wkyr

Saving Subsets of Variables

Figure 3-21
Save Data As Variables dialog box



Only selected variables will be saved to specified data file.

Keep	Name	Label	Order
☑	age	Age in years	1
☑	marital	Marital status	2
☑	address	Years at c...	3
☑	income	Household ...	4
☑	inccat	Income cat...	5
☑	car	Price of pri...	6
☑	carcat	Primary ve...	7
☑	ed	Level of ed...	8
☑	employ	Years with...	9
☑	retire	Retired	10

Selected: 29 of 29 variables.

The Save Data As Variables dialog box allows you to select the variables that you want saved in the new data file. By default, all variables will be saved. Deselect the variables that you don't want to save, or click Drop All and then select the variables that you want to save.

Visible Only. Selects only variables in variable sets currently in use. For more information, see the topic Using Variable Sets to Show and Hide Variables in Chapter 15 on p. 282.

To Save a Subset of Variables

- ▶ Make the Data Editor the active window (click anywhere in the window to make it active).
- ▶ From the menus choose:

  File
    Save As...

- ▶ Click Variables.

▶ Select the variables that you want to save.

Exporting to a Database

You can use the Export to Database Wizard to:

- Replace values in existing database table fields (columns) or add new fields to a table.
- Append new records (rows) to a database table.
- Completely replace a database table or create a new table.

To export data to a database:

▶ From the menus in the Data Editor window for the dataset that contains the data you want to export, choose:

File
  Export to Database

▶ Select the database source.

▶ Follow the instructions in the export wizard to export the data.

Creating Database Fields from PASW Statistics Variables

When creating new fields (adding fields to an existing database table, creating a new table, replacing a table), you can specify field names, data type, and width (where applicable).

Field name. The default field names are the same as the PASW Statistics variable names. You can change the field names to any names allowed by the database format. For example, many databases allow characters in field names that aren't allowed in variable names, including spaces. Therefore, a variable name like *CallWaiting* could be changed to the field name *Call Waiting*.

Type. The export wizard makes initial data type assignments based on the standard ODBC data types or data types allowed by the selected database format that most closely matches the defined PASW Statistics data format—but databases can make type distinctions that have no direct equivalent in PASW Statistics, and vice versa. For example, most numeric values in PASW Statistics are stored as double-precision floating-point values, whereas database numeric data types include float (double), integer, real, and so on. In addition, many databases don't have equivalents to PASW Statistics time formats. You can change the data type to any type available in the drop-down list.

As a general rule, the basic data type (string or numeric) for the variable should match the basic data type of the database field. If there is a data type mismatch that cannot be resolved by the database, an error results and no data are exported to the database. For example, if you export a string variable to a database field with a numeric data type, an error will result if any values of the string variable contain non-numeric characters.

Width. You can change the defined width for string (char, varchar) field types. Numeric field widths are defined by the data type.

By default, PASW Statistics variable formats are mapped to database field types based on the following general scheme. Actual database field types may vary, depending on the database.

PASW Statistics Variable Format	Database Field Type
Numeric	Float or Double
Comma	Float or Double
Dot	Float or Double
Scientific Notation	Float or Double
Date	Date or Datetime or Timestamp
Datetime	Datetime or Timestamp
Time, DTime	Float or Double (number of seconds)
Wkday	Integer (1–7)
Month	Integer (1–12)
Dollar	Float or Double
Custom Currency	Float or Double
String	Char or Varchar

User-Missing Values

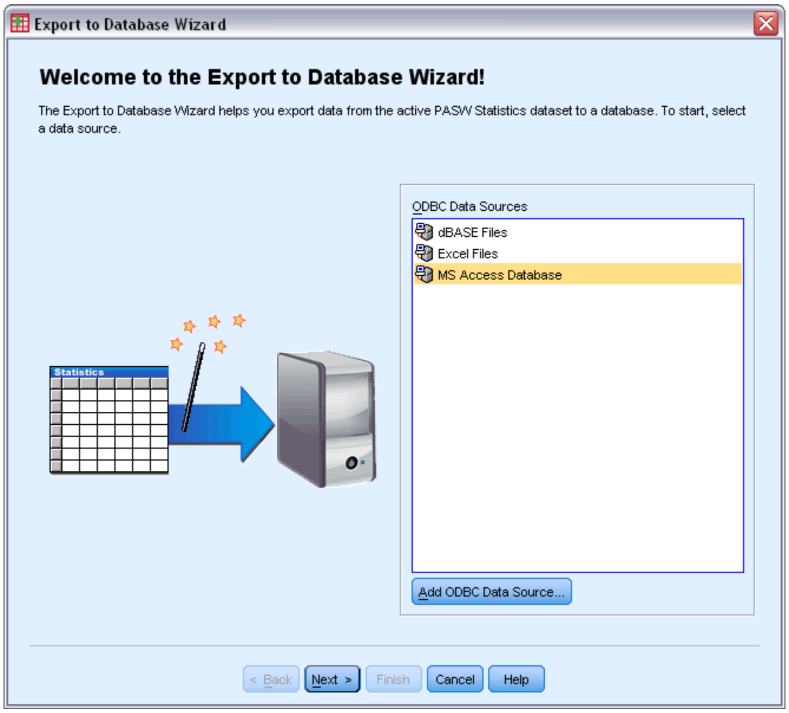
There are two options for the treatment of user-missing values when data from variables are exported to database fields:

- **Export as valid values.** User-missing values are treated as regular, valid, nonmissing values.
- **Export numeric user-missing as nulls and export string user-missing values as blank spaces.** Numeric user-missing values are treated the same as system-missing values. String user-missing values are converted to blank spaces (strings cannot be system-missing).

Selecting a Data Source

In the first panel of the Export to Database Wizard, you select the data source to which you want to export data.

Figure 3-22
Export to Database Wizard, selecting a data source



Welcome to the Export to Database Wizard!

The Export to Database Wizard helps you export data from the active PASW Statistics dataset to a database. To start, select a data source.

ODBC Data Sources

- dBASE Files
- Excel Files
- MS Access Database

You can export data to any database source for which you have the appropriate ODBC driver. (*Note*: Exporting data to OLE DB data sources is not supported.)

If you do not have any ODBC data sources configured, or if you want to add a new data source, click Add ODBC Data Source.

- On Linux operating systems, this button is not available. ODBC data sources are specified in *odbc.ini*, and the *ODBCINI* environment variables must be set to the location of that file. For more information, see the documentation for your database drivers.
- In distributed analysis mode (available with PASW Statistics Server), this button is not available. To add data sources in distributed analysis mode, see your system administrator.

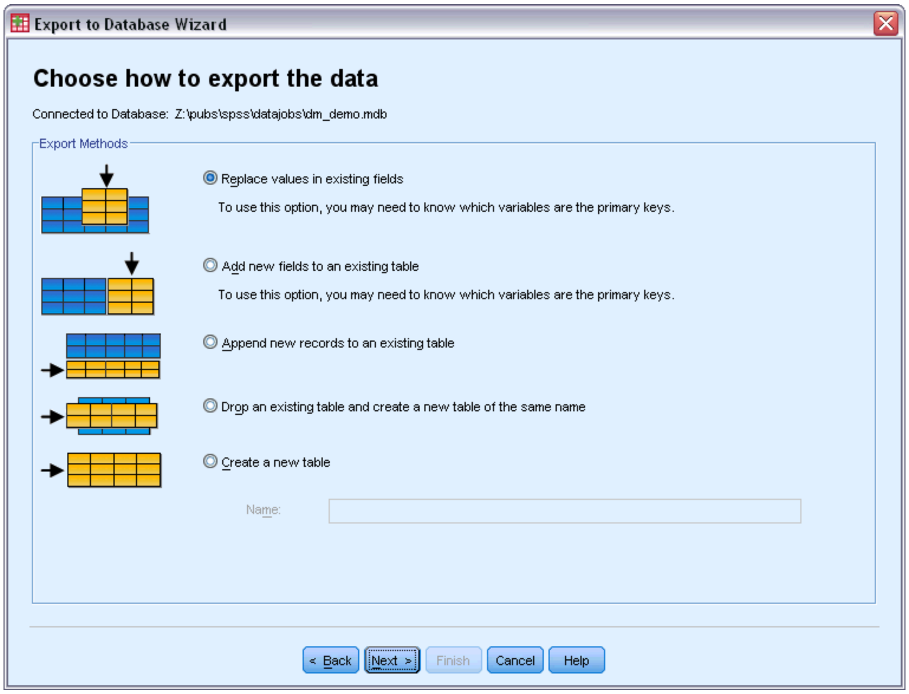
An ODBC data source consists of two essential pieces of information: the driver that will be used to access the data and the location of the database you want to access. To specify data sources, you must have the appropriate drivers installed. Drivers for a variety of database formats are available at *http://www.spss.com/drivers*.

Some data sources may require a login ID and password before you can proceed to the next step.

Choosing How to Export the Data

After you select the data source, you indicate the manner in which you want to export the data.

Figure 3-23
Export to Database Wizard, choosing how to export



Choose how to export the data

Connected to Database: Z:\pubs\spss\datajobs\dm_demo.mdb

Export Methods

● Replace values in existing fields
To use this option, you may need to know which variables are the primary keys.

○ Add new fields to an existing table
To use this option, you may need to know which variables are the primary keys.

○ Append new records to an existing table

○ Drop an existing table and create a new table of the same name

○ Create a new table

Name:

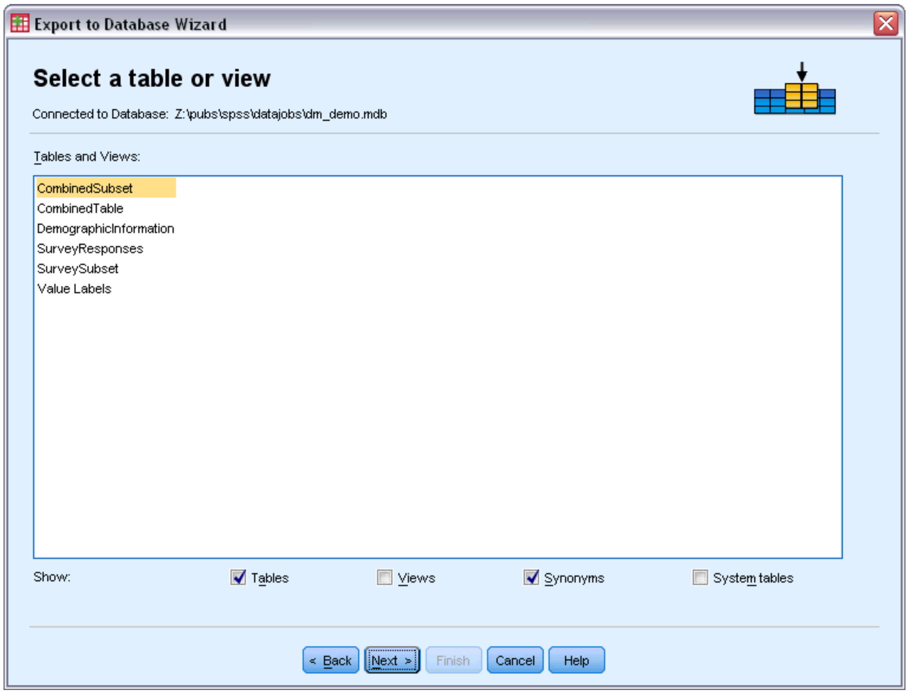
The following choices are available for exporting data to a database:

- **Replace values in existing fields.** Replaces values of selected fields in an existing table with values from the selected variables in the active dataset. For more information, see the topic Replacing Values in Existing Fields on p. 52.
- **Add new fields to an existing table.** Creates new fields in an existing table that contain the values of selected variables in the active dataset. For more information, see the topic Adding New Fields on p. 53. This option is not available for Excel files.
- **Append new records to an existing table.** Adds new records (rows) to an existing table containing the values from cases in the active dataset. For more information, see the topic Appending New Records (Cases) on p. 54.
- **Drop an existing table and create a new table of the same name.** Deletes the specified table and creates a new table of the same name that contains selected variables from the active dataset. All information from the original table, including definitions of field properties (for example, primary keys, data types) is lost. For more information, see the topic Creating a New Table or Replacing a Table on p. 56.
- **Create a new table.** Creates a new table in the database containing data from selected variables in the active dataset. The name can be any value that is allowed as a table name by the data source. The name cannot duplicate the name of an existing table or view in the database. For more information, see the topic Creating a New Table or Replacing a Table on p. 56.

Selecting a Table

When modifying or replacing a table in the database, you need to select the table to modify or replace. This panel in the Export to Database Wizard displays a list of tables and views in the selected database.

Figure 3-24
Export to Database Wizard, selecting a table or view



Export to Database Wizard

Select a table or view

Connected to Database: Z:\pubs\spss\datajobs\dm_demo.mdb

Tables and Views:

CombinedSubset
CombinedTable
DemographicInformation
SurveyResponses
SurveySubset
Value Labels

Show: ☑ Tables □ Views ☑ Synonyms □ System tables

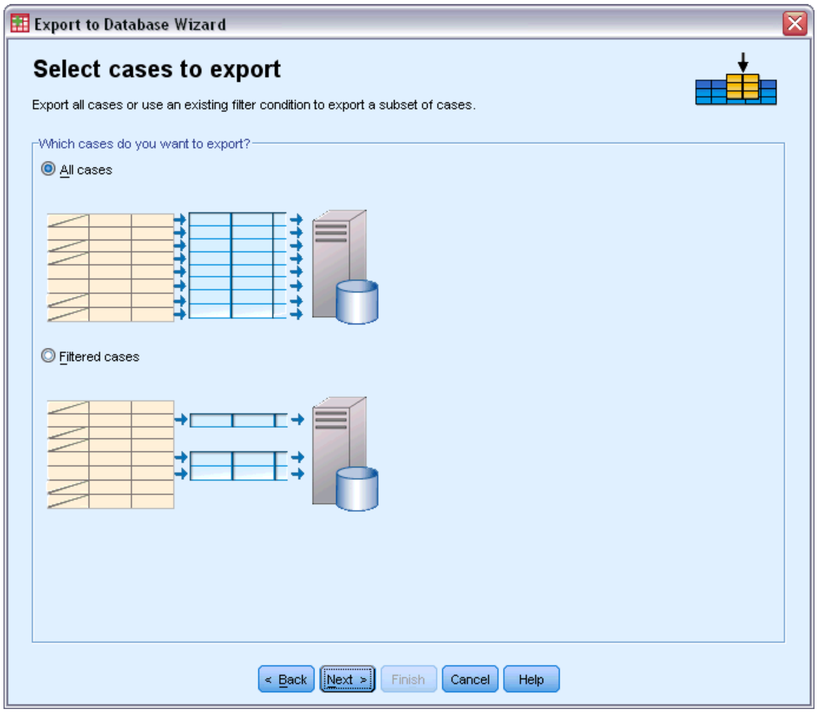
By default, the list displays only standard database tables. You can control the type of items that are displayed in the list:

- **Tables.** Standard database tables.
- **Views.** Views are virtual or dynamic “tables” defined by queries. These can include joins of multiple tables and/or fields derived from calculations based on the values of other fields. You can append records or replace values of existing fields in views, but the fields that you can modify may be restricted, depending on how the view is structured. For example, you cannot modify a derived field, add fields to a view, or replace a view.
- **Synonyms.** A synonym is an alias for a table or view, typically defined in a query.
- **System tables.** System tables define database properties. In some cases, standard database tables may be classified as system tables and will be displayed only if you select this option. Access to real system tables is often restricted to database administrators.

Selecting Cases to Export

Case selection in the Export to Database Wizard is limited either to all cases or to cases selected using a previously defined filter condition. If no case filtering is in effect, this panel will not appear, and all cases in the active dataset will be exported.

Figure 3-25
Export to Database Wizard, selecting cases to export



Export to Database Wizard
Select cases to export
Export all cases or use an existing filter condition to export a subset of cases.
Which cases do you want to export?
● All cases
○ Filtered cases
< Back
Next >
Finish
Cancel
Help

For information on defining a filter condition for case selection, see Select Cases on p. 182.

Matching Cases to Records

When adding fields (columns) to an existing table or replacing the values of existing fields, you need to make sure that each case (row) in the active dataset is correctly matched to the corresponding record in the database.

- In the database, the field or set of fields that uniquely identifies each record is often designated as the **primary key**.
- You need to identify which variable(s) correspond to the primary key field(s) or other fields that uniquely identify each record.
- The fields don’t have to be the primary key in the database, but the field value or combination of field values must be unique for each case.

To match variables with fields in the database that uniquely identify each record:

- Drag and drop the variable(s) onto the corresponding database fields.

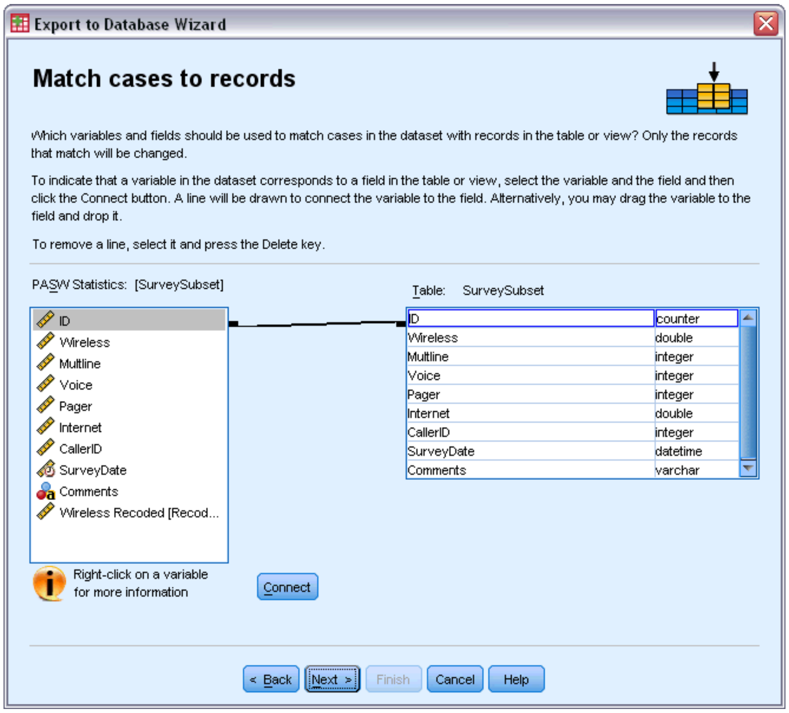
or

- Select a variable from the list of variables, select the corresponding field in the database table, and click Connect.

To delete a connection line:

- Select the connection line and press the Delete key.

Figure 3-26
Export to Database Wizard, matching cases to records



Export to Database Wizard

Match cases to records

Which variables and fields should be used to match cases in the dataset with records in the table or view? Only the records that match will be changed.

To indicate that a variable in the dataset corresponds to a field in the table or view, select the variable and the field and then click the Connect button. A line will be drawn to connect the variable to the field. Alternatively, you may drag the variable to the field and drop it.

To remove a line, select it and press the Delete key.

PASW Statistics: [SurveySubset]

ID
Wireless
Multline
Voice
Pager
Internet
CallerID
SurveyDate
Comments
Wireless Recoded [Recod...

Table: SurveySubset

ID	counter
Wireless	double
Multline	integer
Voice	integer
Pager	integer
Internet	double
CallerID	integer
SurveyDate	datetime
Comments	varchar

Right-click on a variable for more information

Note: The variable names and database field names may not be identical (since database field names may contain characters not allowed in PASW Statistics variable names), but if the active dataset was created from the database table you are modifying, either the variable names or the variable labels will usually be at least similar to the database field names.

Replacing Values in Existing Fields

To replace values of existing fields in a database:

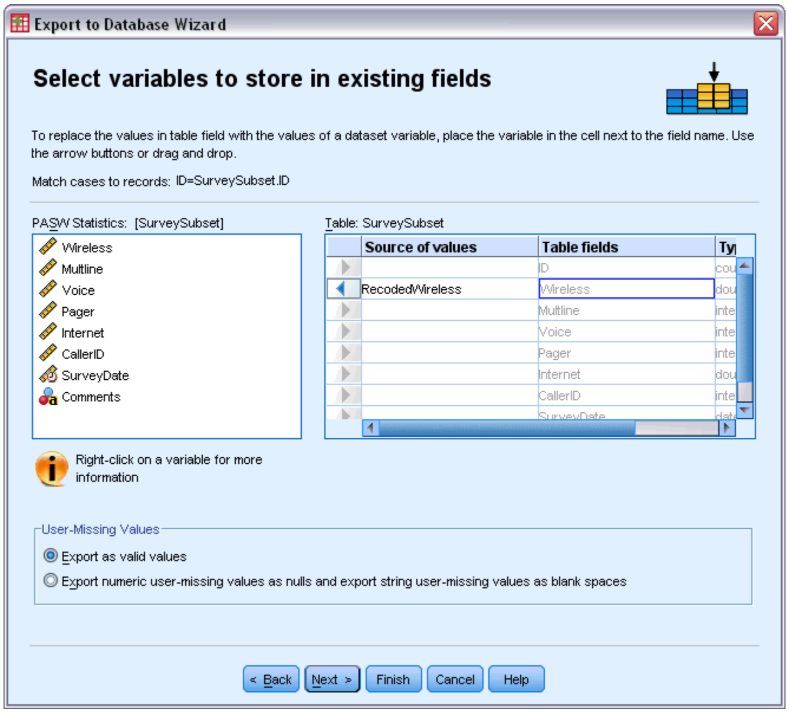
- In the Choose how to export the data panel of the Export to Database Wizard, select Replace values in existing fields.

- In the Select a table or view panel, select the database table.

▶ In the Match cases to records panel, match the variables that uniquely identify each case to the corresponding database field names.

▶ For each field for which you want to replace values, drag and drop the variable that contains the new values into the Source of values column, next to the corresponding database field name.

Figure 3-27
Export to Database Wizard, replacing values of existing fields



Export to Database Wizard

Select variables to store in existing fields

To replace the values in table field with the values of a dataset variable, place the variable in the cell next to the field name. Use the arrow buttons or drag and drop.

Match cases to records: ID=SurveySubset.ID

PASW Statistics: [SurveySubset]

- Wireless
- Multline
- Voice
- Pager
- Internet
- CallerID
- SurveyDate
- Comments

Table: SurveySubset

Source of values	Table fields	Ty
	ID	cou
RecodedWireless	Wireless	dou
	Multline	inte
	Voice	inte
	Pager	inte
	Internet	dou
	CallerID	inte
	SurveyDate	dat

Right-click on a variable for more information

User-Missing Values

● Export as valid values

○ Export numeric user-missing values as nulls and export string user-missing values as blank spaces

- As a general rule, the basic data type (string or numeric) for the variable should match the basic data type of the database field. If there is a data type mismatch that cannot be resolved by the database, an error results and no data is exported to the database. For example, if you export a string variable to a database field with a numeric data type (for example, double, real, integer), an error will result if any values of the string variable contain non-numeric characters. The letter *a* in the icon next to a variable denotes a string variable.
- You cannot modify the field name, type, or width. The original database field attributes are preserved; only the values are replaced.

Adding New Fields

To add new fields to an existing database table:

▶ In the Choose how to export the data panel of the Export to Database Wizard, select Add new fields to an existing table.

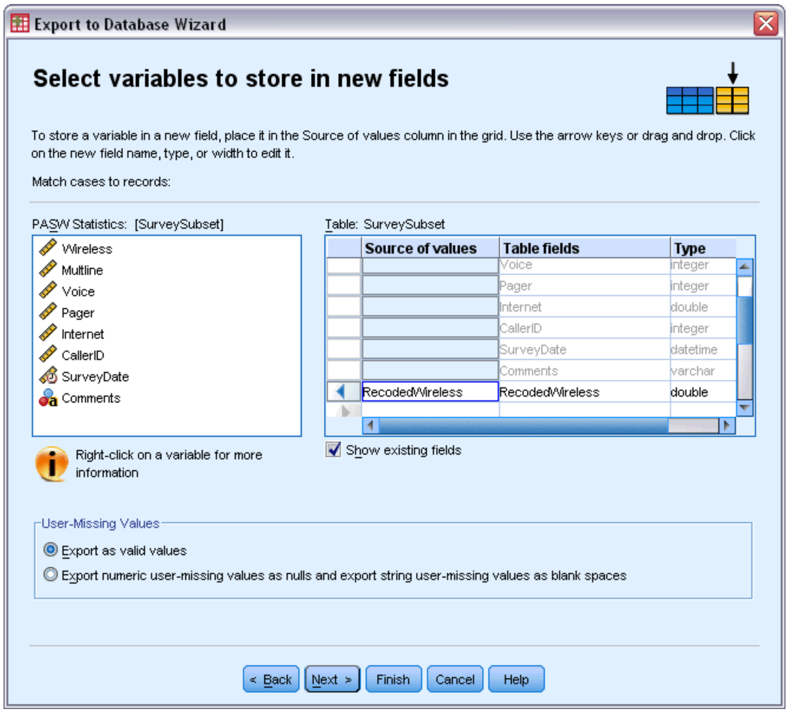
▶ In the Select a table or view panel, select the database table.

▶ In the Match cases to records panel, match the variables that uniquely identify each case to the corresponding database field names.

- ▶ Drag and drop the variables that you want to add as new fields to the Source of values column.

Figure 3-28

Export to Database Wizard, adding new fields to an existing table



Export to Database Wizard

Select variables to store in new fields

To store a variable in a new field, place it in the Source of values column in the grid. Use the arrow keys or drag and drop. Click on the new field name, type, or width to edit it.

Match cases to records:

PASW Statistics: [SurveySubset]

- Wireless
- Multline
- Voice
- Pager
- Internet
- CallerID
- SurveyDate
- Comments

Right-click on a variable for more information

Table: SurveySubset

Source of values	Table fields	Type
	Voice	integer
	Pager	integer
	Internet	double
	CallerID	integer
	SurveyDate	datetime
	Comments	varchar
RecodedWireless	RecodedWireless	double

☑ Show existing fields

User-Missing Values

● Export as valid values

○ Export numeric user-missing values as nulls and export string user-missing values as blank spaces

For information on field names and data types, see the section on creating database fields from PASW Statistics variables in Exporting to a Database on p. 46.

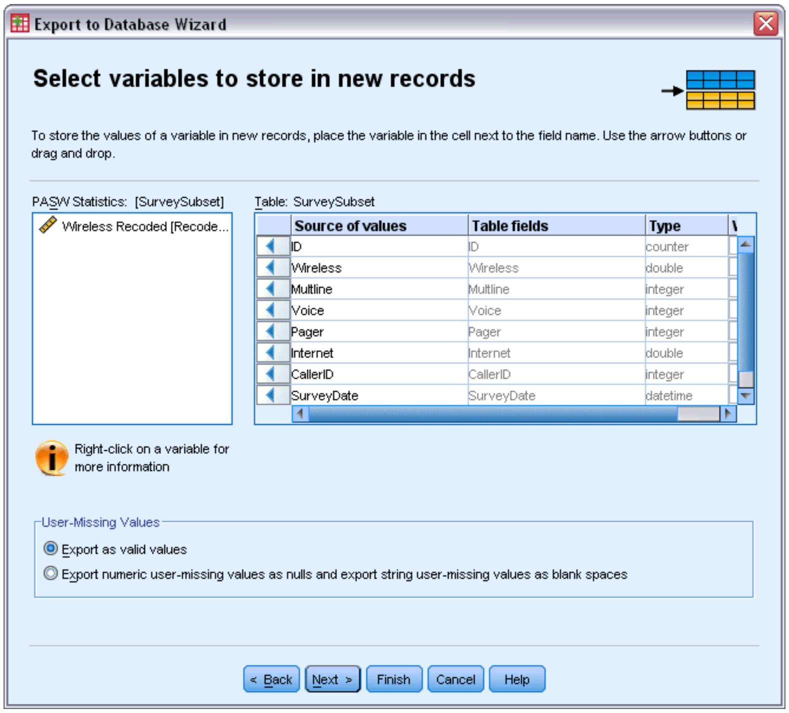
Show existing fields. Select this option to display a list of existing fields. You cannot use this panel in the Export to Database Wizard to replace existing fields, but it may be helpful to know what fields are already present in the table. If you want to replace the values of existing fields, see Replacing Values in Existing Fields on p. 52.

Appending New Records (Cases)

To append new records (cases) to a database table:

- ▶ In the Choose how to export the data panel of the Export to Database Wizard, select Append new records to an existing table.
- ▶ In the Select a table or view panel, select the database table.
- ▶ Match variables in the active dataset to table fields by dragging and dropping variables to the Source of values column.

Figure 3-29
Export to Database Wizard, adding records (cases) to a table



Export to Database Wizard

Select variables to store in new records

To store the values of a variable in new records, place the variable in the cell next to the field name. Use the arrow buttons or drag and drop.

PASW Statistics: [SurveySubset]

Wireless Recoded [Recode...

Right-click on a variable for more information

Table: SurveySubset

Source of values	Table fields	Type
ID	ID	counter
Wireless	Wireless	double
Multline	Multline	integer
Voice	Voice	integer
Pager	Pager	integer
Internet	Internet	double
CallerID	CallerID	integer
SurveyDate	SurveyDate	datetime

User-Missing Values

● Export as valid values

○ Export numeric user-missing values as nulls and export string user-missing values as blank spaces

The Export to Database Wizard will automatically select all variables that match existing fields, based on information about the original database table stored in the active dataset (if available) and/or variable names that are the same as field names. This initial automatic matching is intended only as a guide and does not prevent you from changing the way in which variables are matched with database fields.

When adding new records to an existing table, the following basic rules/limitations apply:

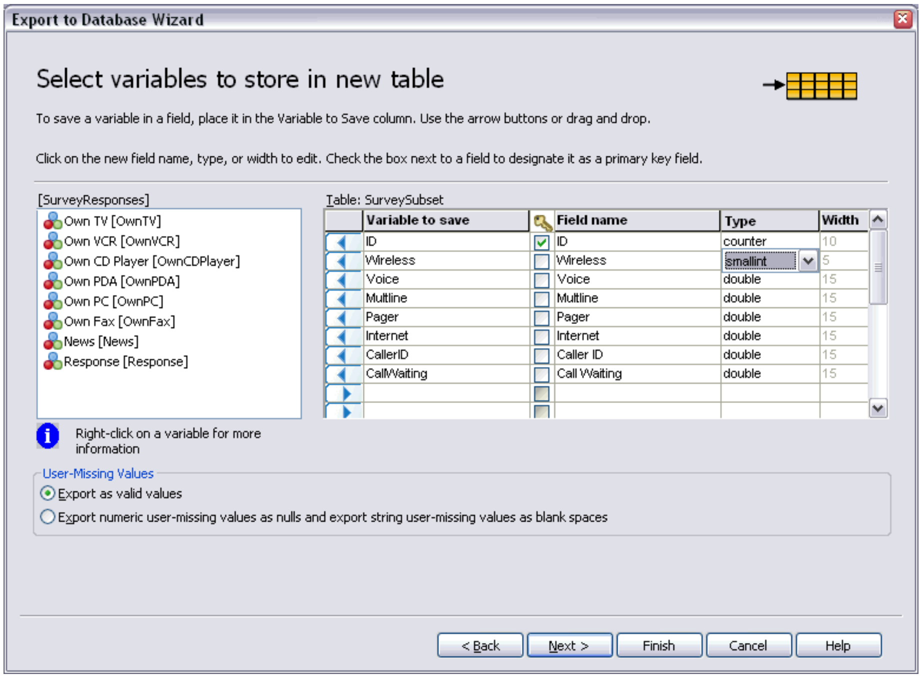
- All cases (or all selected cases) in the active dataset are added to the table. If any of these cases duplicate existing records in the database, an error may result if a duplicate key value is encountered. For information on exporting only selected cases, see Selecting Cases to Export on p. 50.
- You can use the values of new variables created in the session as the values for existing fields, but you cannot add new fields or change the names of existing fields. To add new fields to a database table, see Adding New Fields on p. 53.
- Any excluded database fields or fields not matched to a variable will have no values for the added records in the database table. (If the Source of values cell is empty, there is no variable matched to the field.)

Creating a New Table or Replacing a Table

To create a new database table or replace an existing database table:

- ▶ In the Choose how to export the data panel of the export wizard, select Drop an existing table and create a new table of the same name or select Create a new table and enter a name for the new table.
- ▶ If you are replacing an existing table, in the Select a table or view panel, select the database table.
- ▶ Drag and drop variables into the Variables to save column.
- ▶ Optionally, you can designate variables/fields that define the primary key, change field names, and change the data type.

Figure 3-30
Export to Database Wizard, selecting variables for a new table



Export to Database Wizard

Select variables to store in new table

To save a variable in a field, place it in the Variable to Save column. Use the arrow buttons or drag and drop.

Click on the new field name, type, or width to edit. Check the box next to a field to designate it as a primary key field.

[SurveyResponses]
- Own TV [OwnTV]
- Own VCR [OwnVCR]
- Own CD Player [OwnCDPlayer]
- Own PDA [OwnPDA]
- Own PC [OwnPC]
- Own Fax [OwnFax]
- News [News]
- Response [Response]

Table: SurveySubset

	Variable to save		Field name	Type	Width
◀	ID	☑	ID	counter	10
◀	Wireless	□	Wireless	smallint	5
◀	Voice	□	Voice	double	15
◀	Multline	□	Multline	double	15
◀	Pager	□	Pager	double	15
◀	Internet	□	Internet	double	15
◀	CallerID	□	Caller ID	double	15
◀	CallWaiting	□	Call Waiting	double	15
▶		□			
▶		□			

Right-click on a variable for more information

User-Missing Values
- ● Export as valid values
- ○ Export numeric user-missing values as nulls and export string user-missing values as blank spaces

Primary key. To designate variables as the primary key in the database table, select the box in the column identified with the key icon.

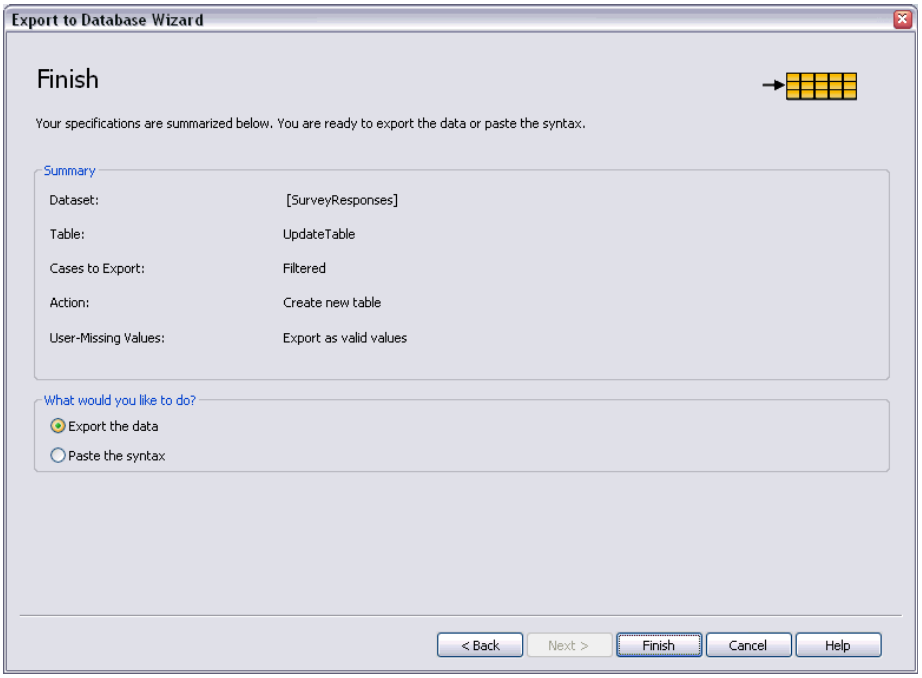
- All values of the primary key must be unique or an error will result.
- If you select a single variable as the primary key, every record (case) must have a unique value for that variable.
- If you select multiple variables as the primary key, this defines a composite primary key, and the *combination* of values for the selected variables must be unique for each case.

For information on field names and data types, see the section on creating database fields from PASW Statistics variables in Exporting to a Database on p. 46.

Completing the Database Export Wizard

The last panel of the Export to Database Wizard provides a summary that indicates what data will be exported and how it will be exported. It also gives you the option of either exporting the data or pasting the underlying command syntax to a syntax window.

Figure 3-31
Export to Database Wizard, finish panel



Export to Database Wizard

Finish

Your specifications are summarized below. You are ready to export the data or paste the syntax.

Summary

Dataset:	[SurveyResponses]
Table:	UpdateTable
Cases to Export:	Filtered
Action:	Create new table
User-Missing Values:	Export as valid values

What would you like to do?

● Export the data

○ Paste the syntax

Summary Information

- **Dataset.** The PASW Statistics session name for the dataset that will be used to export data. This information is primarily useful if you have multiple open data sources. Data sources opened using the graphical user interface (for example, the Database Wizard) are automatically assigned names such as *DataSet1*, *DataSet2*, etc. A data source opened using command syntax will have a dataset name only if one is explicitly assigned.
- **Table.** The name of the table to be modified or created.
- **Cases to Export.** Either all cases are exported or cases selected by a previously defined filter condition are exported. For more information, see the topic Selecting Cases to Export on p. 50.
- **Action.** Indicates how the database will be modified (for example, create a new table, add fields or records to an existing table).
- **User-Missing Values.** User-missing values can be exported as valid values or treated the same as system-missing for numeric variables and converted to blank spaces for string variables. This setting is controlled in the panel in which you select the variables to export.

Exporting to PASW Data Collection

The Export to PASW Data Collection dialog box creates PASW Statistics data files and PASW Data Collection metadata files that you can use to read the data into PASW Data Collection applications. This is particularly useful when “roundtripping” data between PASW Statistics and PASW Data Collection applications.

To export data for use in PASW Data Collection applications:

▶ From the menus in the Data Editor window that contains the data you want to export, choose:

File
  Export to PASW Data Collection

▶ Click Data File to specify the name and location of the PASW Statistics data file.

▶ Click Metadata File to specify the name and location of the PASW Data Collection metadata file.

For new variables and datasets not created from PASW Data Collection data sources, PASW Statistics variable attributes are mapped to PASW Data Collection metadata attributes in the metadata file according to the methods described in the SAV DSC documentation in the Data Collection Developer Library.

If the active dataset was created from a PASW Data Collection data source:

- The new metadata file is created by merging the original metadata attributes with metadata attributes for any new variables, plus any changes to original variables that might affect their metadata attributes (for example, addition of, or changes to, value labels).
- For original variables read from the PASW Data Collection data source, any metadata attributes not recognized by PASW Statistics are preserved in their original state. For example, PASW Statistics converts grid variables to regular PASW Statistics variables, but the metadata that defines these grid variables is preserved when you save the new metadata file.
- If any PASW Data Collection variables were automatically renamed to conform to PASW Statistics variable naming rules, the metadata file maps the converted names back to the original PASW Data Collection variable names.

The presence or absence of value labels can affect the metadata attributes of variables and consequently the way those variables are read by PASW Data Collection applications. If value labels have been defined for any nonmissing values of a variable, they should be defined for all nonmissing values of that variable; otherwise, the unlabeled values will be dropped when the data file is read by PASW Data Collection.

Protecting Original Data

To prevent the accidental modification or deletion of your original data, you can mark the file as read-only.

▶ From the Data Editor menus choose:

File
  Mark File Read Only

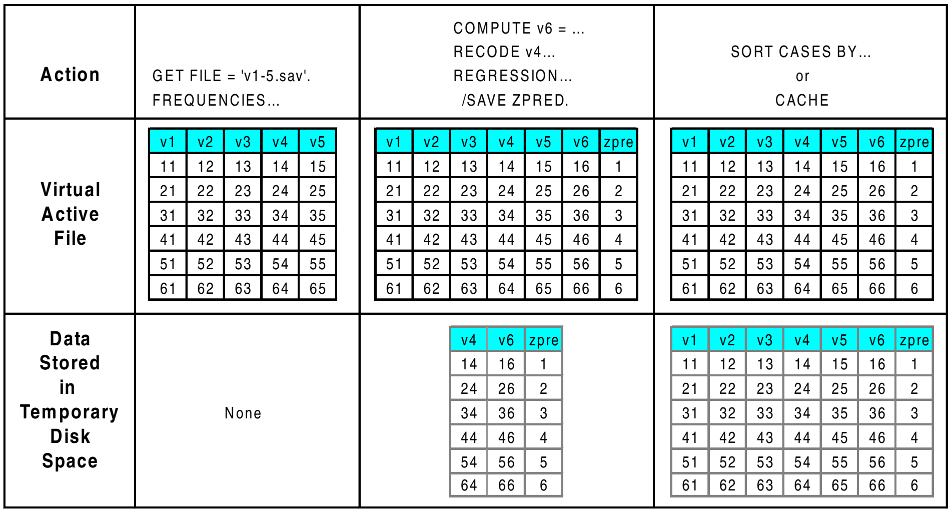
If you make subsequent modifications to the data and then try to save the data file, you can save the data only with a different filename, so the original data are not affected.

You can change the file permissions back to read-write by choosing Mark File Read Write from the File menu.

Virtual Active File

The virtual active file enables you to work with large data files without requiring equally large (or larger) amounts of temporary disk space. For most analysis and charting procedures, the original data source is reread each time you run a different procedure. Procedures that modify the data require a certain amount of temporary disk space to keep track of the changes, and some actions always require enough disk space for at least one entire copy of the data file.

Figure 3-32
Temporary disk space requirements



<table>
<tr><th>Action</th><th>GET FILE = 'v1-5.sav'.
FREQUENCIES...</th><th>COMPUTE v6 = ...
RECODE v4...
REGRESSION...
/SAVE ZPRED.</th><th>SORT CASES BY...
or
CACHE</th></tr>
<tr><td>Virtual Active File</td>
<td><table>
<tr><th>v1</th><th>v2</th><th>v3</th><th>v4</th><th>v5</th></tr>
<tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr>
<tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td></tr>
<tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td></tr>
<tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td></tr>
<tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td></tr>
</table></td>
<td><table>
<tr><th>v1</th><th>v2</th><th>v3</th><th>v4</th><th>v5</th><th>v6</th><th>zpre</th></tr>
<tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>1</td></tr>
<tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>2</td></tr>
<tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>3</td></tr>
<tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>4</td></tr>
<tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>5</td></tr>
<tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>6</td></tr>
</table></td>
<td><table>
<tr><th>v1</th><th>v2</th><th>v3</th><th>v4</th><th>v5</th><th>v6</th><th>zpre</th></tr>
<tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>1</td></tr>
<tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>2</td></tr>
<tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>3</td></tr>
<tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>4</td></tr>
<tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>5</td></tr>
<tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>6</td></tr>
</table></td></tr>
<tr><td>Data Stored in Temporary Disk Space</td>
<td>None</td>
<td><table>
<tr><th>v4</th><th>v6</th><th>zpre</th></tr>
<tr><td>14</td><td>16</td><td>1</td></tr>
<tr><td>24</td><td>26</td><td>2</td></tr>
<tr><td>34</td><td>36</td><td>3</td></tr>
<tr><td>44</td><td>46</td><td>4</td></tr>
<tr><td>54</td><td>56</td><td>5</td></tr>
<tr><td>64</td><td>66</td><td>6</td></tr>
</table></td>
<td><table>
<tr><th>v1</th><th>v2</th><th>v3</th><th>v4</th><th>v5</th><th>v6</th><th>zpre</th></tr>
<tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>1</td></tr>
<tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>2</td></tr>
<tr><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>3</td></tr>
<tr><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td><td>4</td></tr>
<tr><td>51</td><td>52</td><td>53</td><td>54</td><td>55</td><td>56</td><td>5</td></tr>
<tr><td>61</td><td>62</td><td>63</td><td>64</td><td>65</td><td>66</td><td>6</td></tr>
</table></td></tr>
</table>

Actions that don’t require any temporary disk space include:

- Reading PASW Statistics data files
- Merging two or more PASW Statistics data files
- Reading database tables with the Database Wizard
- Merging PASW Statistics data files with database tables
- Running procedures that read data (for example, Frequencies, Crosstabs, Explore)

Actions that create one or more columns of data in temporary disk space include:

- Computing new variables
- Recoding existing variables
- Running procedures that create or modify variables (for example, saving predicted values in Linear Regression)

Actions that create an entire copy of the data file in temporary disk space include:

- Reading Excel files
- Running procedures that sort data (for example, Sort Cases, Split File)
- Reading data with `GET TRANSLATE` or `DATA LIST` commands
- Using the Cache Data facility or the `CACHE` command
- Launching other applications from PASW Statistics that read the data file (for example, AnswerTree, DecisionTime)

Note: The `GET DATA` command provides functionality comparable to `DATA LIST` without creating an entire copy of the data file in temporary disk space. The `SPLIT FILE` command in command syntax does not sort the data file and therefore does not create a copy of the data file. This command, however, requires sorted data for proper operation, and the dialog box interface for this procedure will automatically sort the data file, resulting in a complete copy of the data file. (Command syntax is not available with the Student Version.)

Actions that create an entire copy of the data file by default:

- Reading databases with the Database Wizard
- Reading text files with the Text Wizard

The Text Wizard provides an optional setting to automatically cache the data. By default, this option is selected. You can turn it off by deselecting Cache data locally. For the Database Wizard, you can paste the generated command syntax and delete the `CACHE` command.

Creating a Data Cache

Although the virtual active file can vastly reduce the amount of temporary disk space required, the absence of a temporary copy of the “active” file means that the original data source has to be reread for each procedure. For large data files read from an external source, creating a temporary copy of the data may improve performance. For example, for data tables read from a database source, the SQL query that reads the information from the database must be reexecuted for any command or procedure that needs to read the data. Since virtually all statistical analysis procedures and charting procedures need to read the data, the SQL query is reexecuted for each procedure you run, which can result in a significant increase in processing time if you run a large number of procedures.

If you have sufficient disk space on the computer performing the analysis (either your local computer or a remote server), you can eliminate multiple SQL queries and improve processing time by creating a data cache of the active file. The data cache is a temporary copy of the complete data.

Note: By default, the Database Wizard automatically creates a data cache, but if you use the `GET DATA` command in command syntax to read a database, a data cache is not automatically created. (Command syntax is not available with the Student Version.)

To Create a Data Cache

▶ From the menus choose:

File
  Cache Data...

▶ Click OK or Cache Now.

OK creates a data cache the next time the program reads the data (for example, the next time you run a statistical procedure), which is usually what you want because it doesn't require an extra data pass. Cache Now creates a data cache immediately, which shouldn't be necessary under most circumstances. Cache Now is useful primarily for two reasons:

- A data source is "locked" and can't be updated by anyone until you end your session, open a different data source, or cache the data.
- For large data sources, scrolling through the contents of the Data View tab in the Data Editor will be much faster if you cache the data.

To Cache Data Automatically

You can use the `SET` command to automatically create a data cache after a specified number of changes in the active data file. By default, the active data file is automatically cached after 20 changes in the active data file.

▶ From the menus choose:

File
  New
    Syntax

▶ In the syntax window, type SET CACHE n (where *n* represents the number of changes in the active data file before the data file is cached).

▶ From the menus in the syntax window choose:

Run
  All

Note: The cache setting is not persistent across sessions. Each time you start a new session, the value is reset to the default of 20.

Chapter 4

Distributed Analysis Mode

Distributed analysis mode allows you to use a computer other than your local (or desktop) computer for memory-intensive work. Because remote servers that are used for distributed analysis are typically more powerful and faster than your local computer, distributed analysis mode can significantly reduce computer processing time. Distributed analysis with a remote server can be useful if your work involves:

- Large data files, particularly data read from database sources.
- Memory-intensive tasks. Any task that takes a long time in local analysis mode may be a good candidate for distributed analysis.

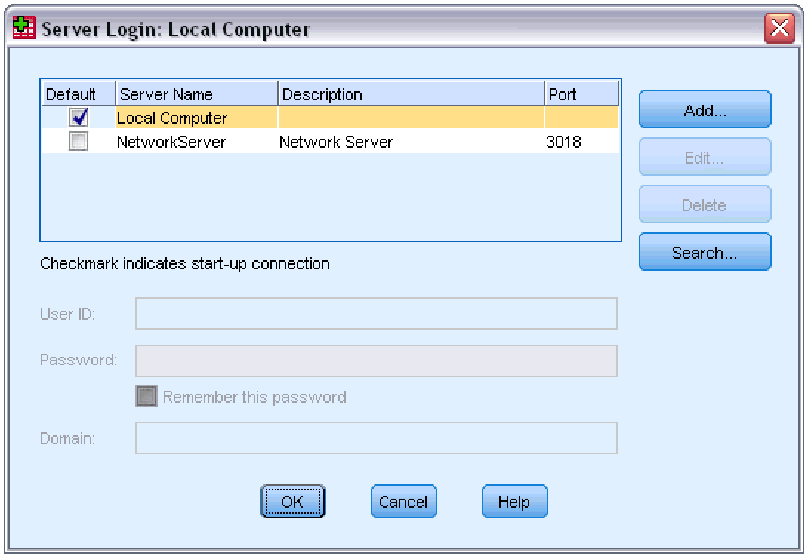
Distributed analysis affects only data-related tasks, such as reading data, transforming data, computing new variables, and calculating statistics. Distributed analysis has no effect on tasks related to editing output, such as manipulating pivot tables or modifying charts.

Note: Distributed analysis is available only if you have both a local version and access to a licensed server version of the software that is installed on a remote server.

Server Login

The Server Login dialog box allows you to select the computer that processes commands and runs procedures. You can select your local computer or a remote server.

Figure 4-1
Server Login dialog box



Server Login: Local Computer

Default	Server Name	Description	Port
☑	Local Computer		
☐	NetworkServer	Network Server	3018

Checkmark indicates start-up connection

User ID:

Password:

☐ Remember this password

Domain:

You can add, modify, or delete remote servers in the list. Remote servers usually require a user ID and password, and a domain name may also be necessary. Contact your system administrator for information about available servers, a user ID and password, domain names, and other connection information.

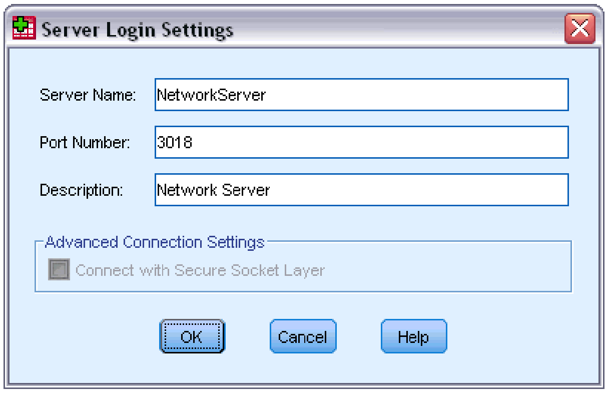
You can select a default server and save the user ID, domain name, and password that are associated with any server. You are automatically connected to the default server when you start a new session.

If you are licensed to use the Statistics Adapter and your site is running PASW Collaboration and Deployment Services 3.5 or later, you can click Search... to view a list of servers that are available on your network. If you are not logged on to a repository, you will be prompted to enter connection information before you can view the list of servers.

Adding and Editing Server Login Settings

Use the Server Login Settings dialog box to add or edit connection information for remote servers for use in distributed analysis mode.

Figure 4-2
Server Login Settings dialog box



Server Name: NetworkServer

Port Number: 3018

Description: Network Server

Advanced Connection Settings

☐ Connect with Secure Socket Layer

Contact your system administrator for a list of available servers, port numbers for the servers, and additional connection information. Do not use the Secure Socket Layer unless instructed to do so by your administrator.

Server Name. A server “name” can be an alphanumeric name that is assigned to a computer (for example, NetworkServer) or a unique IP address that is assigned to a computer (for example, 202.123.456.78).

Port Number. The port number is the port that the server software uses for communications.

Description. You can enter an optional description to display in the servers list.

Connect with Secure Socket Layer. Secure Socket Layer (SSL) encrypts requests for distributed analysis when they are sent to the remote server. Before you use SSL, check with your administrator. For this option to be enabled, SSL must be configured on your desktop computer and the server.

To Select, Switch, or Add Servers

- ▶ From the menus choose:

  File
  Switch Server...

To select a default server:

- ▶ In the server list, select the box next to the server that you want to use.
- ▶ Enter the user ID, domain name, and password that were provided by your administrator.

Note: You are automatically connected to the default server when you start a new session.

To switch to another server:

- ▶ Select the server from the list.
- ▶ Enter your user ID, domain name, and password (if necessary).

Note: When you switch servers during a session, all open windows are closed. You will be prompted to save changes before the windows are closed.

To add a server:

- ▶ Get the server connection information from your administrator.
- ▶ Click Add to open the Server Login Settings dialog box.
- ▶ Enter the connection information and optional settings, and then click OK.

To edit a server:

- ▶ Get the revised connection information from your administrator.
- ▶ Click Edit to open the Server Login Settings dialog box.
- ▶ Enter the changes and click OK.

To search for available servers:

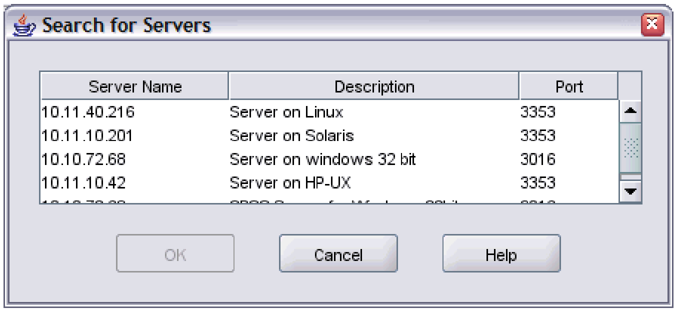
Note: The ability to search for available servers is available only if you are licensed to use the Statistics Adapter and your site is running PASW Collaboration and Deployment Services 3.5 or later.

- ▶ Click Search... to open the Search for Servers dialog box. If you are not logged on to a repository, you will be prompted for connection information.
- ▶ Select one or more available servers and click OK. The servers will now appear in the Server Login dialog box.
- ▶ To connect to one of the servers, follow the instructions “To switch to another server.”

Searching for Available Servers

Use the Search for Servers dialog box to select one or more servers that are available on your network. This dialog box appears when you click Search... on the Server Login dialog box.

Figure 4-3
Search for Servers dialog box



Server Name	Description	Port
10.11.40.216	Server on Linux	3353
10.11.10.201	Server on Solaris	3353
10.10.72.68	Server on windows 32 bit	3016
10.11.10.42	Server on HP-UX	3353

Select one or more servers and click OK to add them to the Server Login dialog box. Although you can manually add servers in the Server Login dialog box, searching for available servers lets you connect to servers without requiring that you know the correct server name and port number. This information is automatically provided. However, you still need the correct logon information, such as user name, domain, and password.

Opening Data Files from a Remote Server

In distributed analysis mode, the Open Remote File dialog box replaces the standard Open File dialog box.

- The contents of the list of available files, folders, and drives depends on what is available on or from the remote server. The current server name is indicated at the top of the dialog box.
- In distributed analysis mode, you will not have access to files on your local computer unless you specify the drive as a shared device or specify the folders containing your data files as shared folders. Consult the documentation for your operating system for information on how to “share” folders on your local computer with the server network.
- If the server is running a different operating system (for example, you are running Windows and the server is running UNIX), you probably won’t have access to local data files in distributed analysis mode even if they are in shared folders.

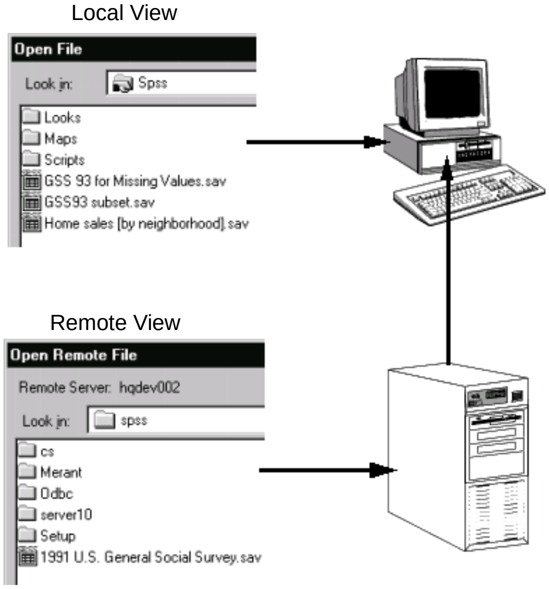
File Access in Local and Distributed Analysis Mode

The view of data folders (directories) and drives for both your local computer and the network is based on the computer that you are currently using to process commands and run procedures—which is not necessarily the computer in front of you.

Local analysis mode. When you use your local computer as your “server,” the view of data files, folders, and drives in the file access dialog box (for opening data files) is similar to what you see in other applications or in Windows Explorer. You can see all of the data files and folders on your computer and any files and folders on mounted network drives.

Distributed analysis mode. When you use another computer as a “remote server” to run commands and procedures, the view of data files, folders, and drives represents the view from the remote server computer. Although you may see familiar folder names (such as *Program Files*) and drives (such as *C*), these items are *not* the folders and drives on your computer; they are the folders and drives on the remote server.

Figure 4-4
Local and remote views



In distributed analysis mode, you will not have access to data files on your local computer unless you specify the drive as a shared device or specify the folders containing your data files as shared folders. If the server is running a different operating system (for example, you are running Windows and the server is running UNIX), you probably won’t have access to local data files in distributed analysis mode even if they are in shared folders.

Distributed analysis mode is not the same as accessing data files that reside on another computer on your network. You can access data files on other network devices in local analysis mode or in distributed analysis mode. In local mode, you access other devices from your local computer. In distributed mode, you access other network devices from the remote server.

If you’re not sure if you’re using local analysis mode or distributed analysis mode, look at the title bar in the dialog box for accessing data files. If the title of the dialog box contains the word *Remote* (as in Open Remote File), or if the text Remote Server: [server name] appears at the top of the dialog box, you’re using distributed analysis mode.

Note: This situation affects only dialog boxes for accessing data files (for example, Open Data, Save Data, Open Database, and Apply Data Dictionary). For all other file types (for example, Viewer files, syntax files, and script files), the local view is always used.

Availability of Procedures in Distributed Analysis Mode

In distributed analysis mode, procedures are available for use only if they are installed on both your local version and the version on the remote server.

If you have optional components installed locally that are not available on the remote server and you switch from your local computer to a remote server, the affected procedures will be removed from the menus and the corresponding command syntax will result in errors. Switching back to local mode will restore all affected procedures.

Absolute versus Relative Path Specifications

In distributed analysis mode, relative path specifications for data files and command syntax files are relative to the current server, not relative to your local computer. A relative path specification such as */mydocs/mydata.sav* does not point to a directory and file on your local drive; it points to a directory and file on the remote server's hard drive.

Windows UNC Path Specifications

If you are using a Windows server version, you can use universal naming convention (UNC) specifications when accessing data and syntax files with command syntax. The general form of a UNC specification is:

```
\\servername\sharename\path\filename
```

- *Servername* is the name of the computer that contains the data file.
- *Sharename* is the folder (directory) on that computer that is designated as a shared folder.
- *Path* is any additional folder (subdirectory) path below the shared folder.
- *Filename* is the name of the data file.

An example is as follows:

```
GET FILE='\\hqdev001\public\july\sales.sav'.
```

If the computer does not have a name assigned to it, you can use its IP address, as in:

```
GET FILE='\\204.125.125.53\public\july\sales.sav'.
```

Even with UNC path specifications, you can access data and syntax files only from devices and folders that are designated as shared. When you use distributed analysis mode, this situation includes data and syntax files on your local computer.

UNIX Absolute Path Specifications

For UNIX server versions, there is no equivalent to the UNC path, and all directory paths must be absolute paths that start at the root of the server; relative paths are not allowed. For example, if the data file is located in */bin/data* and the current directory is also */bin/data*, `GET FILE='sales.sav'` is not valid; you must specify the entire path, as in:

```
GET FILE='/bin/sales.sav'.
INSERT FILE='/bin/salesjob.sps'.
```

Chapter

5

Data Editor

The Data Editor provides a convenient, spreadsheet-like method for creating and editing data files. The Data Editor window opens automatically when you start a session.

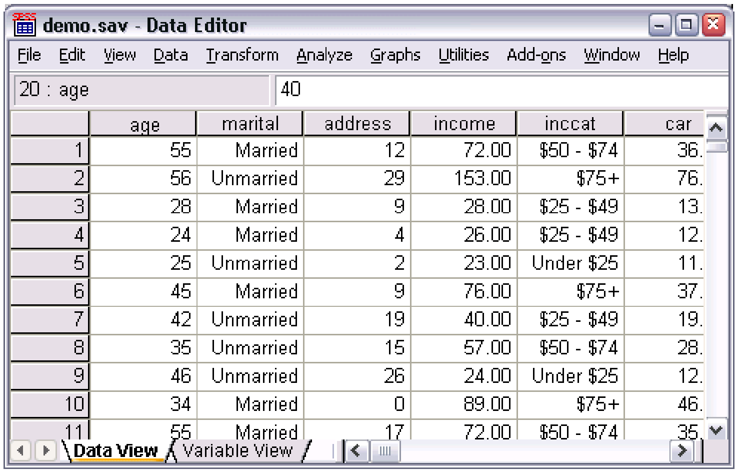
The Data Editor provides two views of your data:

- **Data View.** This view displays the actual data values or defined value labels.
- **Variable View.** This view displays variable definition information, including defined variable and value labels, data type (for example, string, date, or numeric), measurement level (nominal, ordinal, or scale), and user-defined missing values.

In both views, you can add, change, and delete information that is contained in the data file.

Data View

Figure 5-1
Data View


demo.sav - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Data View
Variable View

20 : age 40

	age	marital	address	income	inccat	car
1	55	Married	12	72.00	$50 - $74	36.
2	56	Unmarried	29	153.00	$75+	76.
3	28	Married	9	28.00	$25 - $49	13.
4	24	Married	4	26.00	$25 - $49	12.
5	25	Unmarried	2	23.00	Under $25	11.
6	45	Married	9	76.00	$75+	37.
7	42	Unmarried	19	40.00	$25 - $49	19.
8	35	Unmarried	15	57.00	$50 - $74	28.
9	46	Unmarried	26	24.00	Under $25	12.
10	34	Married	0	89.00	$75+	46.
11	55	Married	17	72.00	$50 - $74	35.

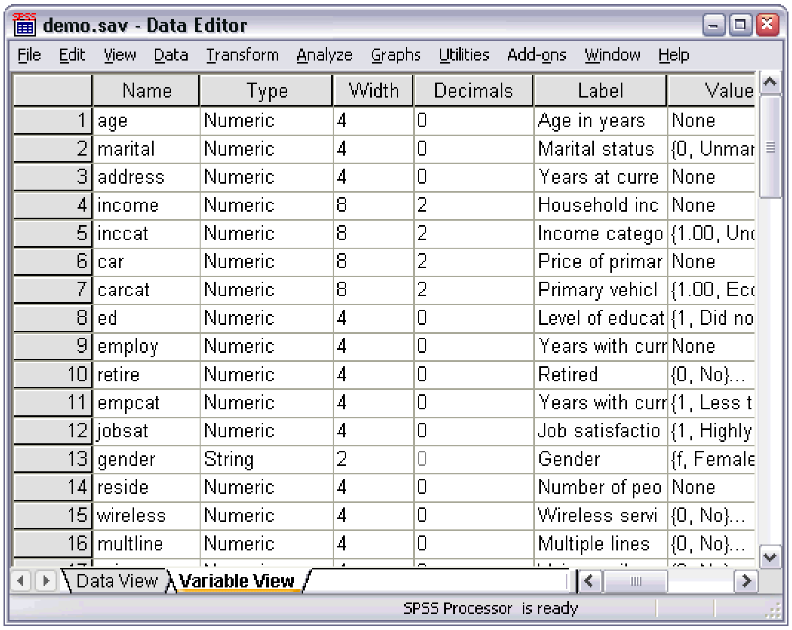
Many of the features of Data View are similar to the features that are found in spreadsheet applications. There are, however, several important distinctions:

- Rows are cases. Each row represents a case or an observation. For example, each individual respondent to a questionnaire is a case.
- Columns are variables. Each column represents a variable or characteristic that is being measured. For example, each item on a questionnaire is a variable.

- Cells contain values. Each cell contains a single value of a variable for a case. The cell is where the case and the variable intersect. Cells contain only data values. Unlike spreadsheet programs, cells in the Data Editor cannot contain formulas.
- The data file is rectangular. The dimensions of the data file are determined by the number of cases and variables. You can enter data in any cell. If you enter data in a cell outside the boundaries of the defined data file, the data rectangle is extended to include any rows and/or columns between that cell and the file boundaries. There are no "empty" cells within the boundaries of the data file. For numeric variables, blank cells are converted to the system-missing value. For string variables, a blank is considered a valid value.

Variable View

Figure 5-2
Variable View



	Name	Type	Width	Decimals	Label	Value
1	age	Numeric	4	0	Age in years	None
2	marital	Numeric	4	0	Marital status	{0, Unmar
3	address	Numeric	4	0	Years at curre	None
4	income	Numeric	8	2	Household inc	None
5	inccat	Numeric	8	2	Income catego	{1.00, Unc
6	car	Numeric	8	2	Price of primar	None
7	carcat	Numeric	8	2	Primary vehicl	{1.00, Ecc
8	ed	Numeric	4	0	Level of educat	{1, Did no
9	employ	Numeric	4	0	Years with curr	None
10	retire	Numeric	4	0	Retired	{0, No}...
11	empcat	Numeric	4	0	Years with curr	{1, Less t
12	jobsat	Numeric	4	0	Job satisfactio	{1, Highly
13	gender	String	2	0	Gender	{f, Female
14	reside	Numeric	4	0	Number of peo	None
15	wireless	Numeric	4	0	Wireless servi	{0, No}...
16	multline	Numeric	4	0	Multiple lines	{0, No}...

Variable View contains descriptions of the attributes of each variable in the data file. In Variable View:

- Rows are variables.
- Columns are variable attributes.

You can add or delete variables and modify attributes of variables, including the following attributes:

- Variable name
- Data type
- Number of digits or characters
- Number of decimal places
- Descriptive variable and value labels

- User-defined missing values
- Column width
- Measurement level

All of these attributes are saved when you save the data file.

In addition to defining variable properties in Variable View, there are two other methods for defining variable properties:

- The Copy Data Properties Wizard provides the ability to use an external PASW Statistics data file or another dataset that is available in the current session as a template for defining file and variable properties in the active dataset. You can also use variables in the active dataset as templates for other variables in the active dataset. Copy Data Properties is available on the Data menu in the Data Editor window.
- Define Variable Properties (also available on the Data menu in the Data Editor window) scans your data and lists all unique data values for any selected variables, identifies unlabeled values, and provides an auto-label feature. This method is particularly useful for categorical variables that use numeric codes to represent categories—for example, 0 = *Male*, 1 = *Female*.

To Display or Define Variable Attributes

▶ Make the Data Editor the active window.

▶ Double-click a variable name at the top of the column in Data View, or click the Variable View tab.

▶ To define new variables, enter a variable name in any blank row.

▶ Select the attribute(s) that you want to define or modify.

Variable Names

The following rules apply to variable names:

- Each variable name must be unique; duplication is not allowed.
- Variable names can be up to 64 bytes long, and the first character must be a letter or one of the characters @, #, or $. Subsequent characters can be any combination of letters, numbers, nonpunctuation characters, and a period (.). In code page mode, sixty-four bytes typically means 64 characters in single-byte languages (for example, English, French, German, Spanish, Italian, Hebrew, Russian, Greek, Arabic, and Thai) and 32 characters in double-byte languages (for example, Japanese, Chinese, and Korean). Many string characters that only take one byte in code page mode take two or more bytes in Unicode mode. For example, é is one byte in code page format but is two bytes in Unicode format; so *résumé* is six bytes in a code page file and eight bytes in Unicode mode.

  Note: Letters include any nonpunctuation characters used in writing ordinary words in the languages supported in the platform's character set.
- Variable names cannot contain spaces.

- A # character in the first position of a variable name defines a scratch variable. You can only create scratch variables with command syntax. You cannot specify a # as the first character of a variable in dialog boxes that create new variables.
- A $ sign in the first position indicates that the variable is a system variable. The $ sign is not allowed as the initial character of a user-defined variable.
- The period, the underscore, and the characters $, #, and @ can be used within variable names. For example, *A._$@#1* is a valid variable name.
- Variable names ending with a period should be avoided, since the period may be interpreted as a command terminator. You can only create variables that end with a period in command syntax. You cannot create variables that end with a period in dialog boxes that create new variables.
- Variable names ending in underscores should be avoided, since such names may conflict with names of variables automatically created by commands and procedures.
- Reserved keywords cannot be used as variable names. Reserved keywords are ALL, AND, BY, EQ, GE, GT, LE, LT, NE, NOT, OR, TO, and WITH.
- Variable names can be defined with any mixture of uppercase and lowercase characters, and case is preserved for display purposes.
- When long variable names need to wrap onto multiple lines in output, lines are broken at underscores, periods, and points where content changes from lower case to upper case.

Variable Measurement Level

You can specify the level of measurement as scale (numeric data on an interval or ratio scale), ordinal, or nominal. Nominal and ordinal data can be either string (alphanumeric) or numeric.

- **Nominal.** A variable can be treated as nominal when its values represent categories with no intrinsic ranking (for example, the department of the company in which an employee works). Examples of nominal variables include region, zip code, and religious affiliation.
- **Ordinal.** A variable can be treated as ordinal when its values represent categories with some intrinsic ranking (for example, levels of service satisfaction from highly dissatisfied to highly satisfied). Examples of ordinal variables include attitude scores representing degree of satisfaction or confidence and preference rating scores.
- **Scale.** A variable can be treated as scale (continuous) when its values represent ordered categories with a meaningful metric, so that distance comparisons between values are appropriate. Examples of scale variables include age in years and income in thousands of dollars.

Note: For ordinal string variables, the alphabetic order of string values is assumed to reflect the true order of the categories. For example, for a string variable with the values of *low*, *medium*, *high*, the order of the categories is interpreted as *high*, *low*, *medium*, which is not the correct order. In general, it is more reliable to use numeric codes to represent ordinal data.

New numeric variables created during a session are assigned the scale measurement level. For data read from external file formats and PASW Statistics data files that were created prior to version 8.0, default assignment of measurement level is based on the following rules:

- Numeric variables with fewer than 24 unique values and string variables are set to nominal.
- Numeric variables with 24 or more unique values are set to scale.

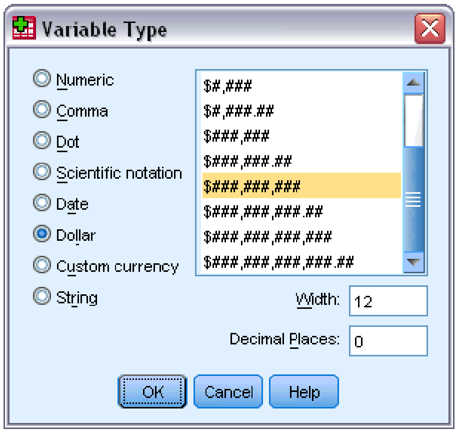
You can change the scale/nominal cutoff value for numeric variables in the Options dialog box. For more information, see the topic Data Options in Chapter 16 on p. 295.

The Define Variable Properties dialog box, available from the Data menu, can help you assign the correct measurement level. For more information, see the topic Assigning the Measurement Level in Chapter 7 on p. 100.

Variable Type

Variable Type specifies the data type for each variable. By default, all new variables are assumed to be numeric. You can use Variable Type to change the data type. The contents of the Variable Type dialog box depend on the selected data type. For some data types, there are text boxes for width and number of decimals; for other data types, you can simply select a format from a scrollable list of examples.

Figure 5-3
Variable Type dialog box



Variable Type

○ Numeric
○ Comma
○ Dot
○ Scientific notation
○ Date
● Dollar
○ Custom currency
○ String

$#,###
$#,###.##
$###,###
$###,###.##
$###,###,###
$###,###,###.##
$###,###,###,###
$###,###,###,###.##

Width: 12

Decimal Places: 0

The available data types are as follows:

Numeric. A variable whose values are numbers. Values are displayed in standard numeric format. The Data Editor accepts numeric values in standard format or in scientific notation.

Comma. A numeric variable whose values are displayed with commas delimiting every three places and displayed with the period as a decimal delimiter. The Data Editor accepts numeric values for comma variables with or without commas or in scientific notation. Values cannot contain commas to the right of the decimal indicator.

Dot. A numeric variable whose values are displayed with periods delimiting every three places and with the comma as a decimal delimiter. The Data Editor accepts numeric values for dot variables with or without periods or in scientific notation. Values cannot contain periods to the right of the decimal indicator.

Scientific notation. A numeric variable whose values are displayed with an embedded E and a signed power-of-10 exponent. The Data Editor accepts numeric values for such variables with or without an exponent. The exponent can be preceded by E or D with an optional sign or by the sign alone—for example, 123, 1.23E2, 1.23D2, 1.23E+2, and 1.23+2.

Date. A numeric variable whose values are displayed in one of several calendar-date or clock-time formats. Select a format from the list. You can enter dates with slashes, hyphens, periods, commas, or blank spaces as delimiters. The century range for two-digit year values is determined by your Options settings (from the Edit menu, choose Options, and then click the Data tab).

Dollar. A numeric variable displayed with a leading dollar sign ($), commas delimiting every three places, and a period as the decimal delimiter. You can enter data values with or without the leading dollar sign.

Custom currency. A numeric variable whose values are displayed in one of the custom currency formats that you have defined on the Currency tab of the Options dialog box. Defined custom currency characters cannot be used in data entry but are displayed in the Data Editor.

String. A variable whose values are not numeric and therefore are not used in calculations. The values can contain any characters up to the defined length. Uppercase and lowercase letters are considered distinct. This type is also known as an alphanumeric variable.

To Define Variable Type

▶ Click the button in the *Type* cell for the variable that you want to define.

▶ Select the data type in the Variable Type dialog box.

▶ Click OK.

Input versus Display Formats

Depending on the format, the display of values in Data View may differ from the actual value as entered and stored internally. Following are some general guidelines:

- For numeric, comma, and dot formats, you can enter values with any number of decimal positions (up to 16), and the entire value is stored internally. The Data View displays only the defined number of decimal places and rounds values with more decimals. However, the complete value is used in all computations.
- For string variables, all values are right-padded to the maximum width. For a string variable with a maximum width of three, a value of *No* is stored internally as `'No '` and is not equivalent to `' No'`.
- For date formats, you can use slashes, dashes, spaces, commas, or periods as delimiters between day, month, and year values, and you can enter numbers, three-letter abbreviations, or complete names for month values. Dates of the general format *dd-mmm-yy* are displayed with dashes as delimiters and three-letter abbreviations for the month. Dates of the general

format *dd/mm/yy* and *mm/dd/yy* are displayed with slashes for delimiters and numbers for the month. Internally, dates are stored as the number of seconds from October 14, 1582. The century range for dates with two-digit years is determined by your Options settings (from the Edit menu, choose Options, and then click the Data tab).

- For time formats, you can use colons, periods, or spaces as delimiters between hours, minutes, and seconds. Times are displayed with colons as delimiters. Internally, times are stored as a number of seconds that represents a time interval. For example, 10:00:00 is stored internally as 36000, which is 60 (seconds per minute) x 60 (minutes per hour) x 10 (hours).

Variable Labels

You can assign descriptive variable labels up to 256 characters (128 characters in double-byte languages). Variable labels can contain spaces and reserved characters that are not allowed in variable names.

To Specify Variable Labels

▶ Make the Data Editor the active window.

▶ Double-click a variable name at the top of the column in Data View, or click the Variable View tab.

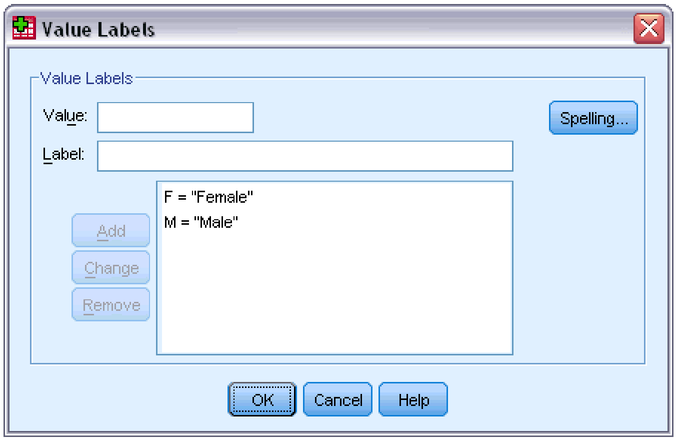
▶ In the *Label* cell for the variable, enter the descriptive variable label.

Value Labels

You can assign descriptive value labels for each value of a variable. This process is particularly useful if your data file uses numeric codes to represent non-numeric categories (for example, codes of 1 and 2 for *male* and *female*).

- Value labels can be up to 120 bytes.

Figure 5-4
Value Labels dialog box



F = "Female"
M = "Male"

To Specify Value Labels

- Click the button in the *Values* cell for the variable that you want to define.
- For each value, enter the value and a label.
- Click Add to enter the value label.
- Click OK.

Inserting Line Breaks in Labels

Variable labels and value labels automatically wrap to multiple lines in pivot tables and charts if the cell or area isn't wide enough to display the entire label on one line, and you can edit results to insert manual line breaks if you want the label to wrap at a different point. You can also create variable labels and value labels that will *always* wrap at specified points and be displayed on multiple lines.

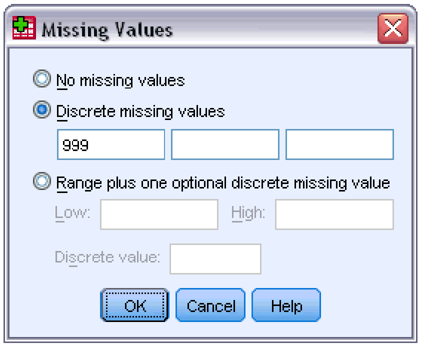
- For variable labels, select the *Label* cell for the variable in Variable View in the Data Editor.
- For value labels, select the *Values* cell for the variable in Variable View in the Data Editor, click the button in the cell, and select the label that you want to modify in the Value Labels dialog box.
- At the place in the label where you want the label to wrap, type \n.

The \n is not displayed in pivot tables or charts; it is interpreted as a line break character.

Missing Values

Missing Values defines specified data values as **user-missing**. For example, you might want to distinguish between data that are missing because a respondent refused to answer and data that are missing because the question didn't apply to that respondent. Data values that are specified as user-missing are flagged for special treatment and are excluded from most calculations.

Figure 5-5
Missing Values dialog box



Missing Values

○ No missing values
● Discrete missing values
999
○ Range plus one optional discrete missing value
Low: High:
Discrete value:

- You can enter up to three discrete (individual) missing values, a range of missing values, or a range plus one discrete value.
- Ranges can be specified only for numeric variables.

- All string values, including null or blank values, are considered to be valid unless you explicitly define them as missing.
- Missing values for string variables cannot exceed eight bytes. (There is no limit on the defined width of the string variable, but defined missing values cannot exceed eight bytes.)
- To define null or blank values as missing for a string variable, enter a single space in one of the fields under the Discrete missing values selection.

To Define Missing Values

▶ Click the button in the *Missing* cell for the variable that you want to define.

▶ Enter the values or range of values that represent missing data.

Roles

Some dialogs support predefined roles that can be used to pre-select variables for analysis. When you open one of these dialogs, variables that meet the role requirements will be automatically displayed in the destination list(s). Available roles are:

Input. The variable will be used as an input (e.g., predictor, independent variable).

Target. The variable will be used as an output or target (e.g., dependent variable).

Both. The variable will be used as both input and output.

None. The variable has no role assignment.

Partition. The variable will be used to partition the data into separate samples for training, testing, and validation.

Split. Included for round-trip compatibility with PASW Modeler. Variables with this role are *not* used as split-file variables in PASW Statistics.

- By default, all variables are assigned the *Input* role. This includes data from external file formats and data files from versions of PASW Statistics prior to version 18.
- Role assignment only affects dialogs that support role assignment. It has no effect on command syntax.

To Assign Roles

▶ Select the role from the list in the *Role* cell for the variable.

Column Width

You can specify a number of characters for the column width. Column widths can also be changed in Data View by clicking and dragging the column borders.

- Column width for proportional fonts is based on average character width. Depending on the characters used in the value, more or fewer characters may be displayed in the specified width.
- Column width affect only the display of values in the Data Editor. Changing the column width does not change the defined width of a variable.

Variable Alignment

Alignment controls the display of data values and/or value labels in Data View. The default alignment is right for numeric variables and left for string variables. This setting affects only the display in Data View.

Applying Variable Definition Attributes to Multiple Variables

After you have defined variable definition attributes for a variable, you can copy one or more attributes and apply them to one or more variables.

Basic copy and paste operations are used to apply variable definition attributes. You can:

- Copy a single attribute (for example, value labels) and paste it to the same attribute cell(s) for one or more variables.
- Copy all attributes from one variable and paste them to one or more other variables.
- Create multiple new variables with all the attributes of a copied variable.

Applying Variable Definition Attributes to Other Variables

To Apply Individual Attributes from a Defined Variable

▶ In Variable View, select the attribute cell that you want to apply to other variables.

▶ From the menus choose:

Edit
  Copy

▶ Select the attribute cell(s) to which you want to apply the attribute. (You can select multiple target variables.)

▶ From the menus choose:

Edit
  Paste

If you paste the attribute to blank rows, new variables are created with default attributes for all attributes except the selected attribute.

To Apply All Attributes from a Defined Variable

▶ In Variable View, select the row number for the variable with the attributes that you want to use. (The entire row is highlighted.)

▶ From the menus choose:

Edit
  Copy

▶ Select the row number(s) for the variable(s) to which you want to apply the attributes. (You can select multiple target variables.)

- From the menus choose:

  Edit
  Paste

Generating Multiple New Variables with the Same Attributes

- In Variable View, click the row number for the variable that has the attributes that you want to use for the new variable. (The entire row is highlighted.)
- From the menus choose:

  Edit
  Copy
- Click the empty row number beneath the last defined variable in the data file.
- From the menus choose:

  Edit
  Paste Variables...
- In the Paste Variables dialog box, enter the number of variables that you want to create.
- Enter a prefix and starting number for the new variables.
- Click OK.

  The new variable names will consist of the specified prefix plus a sequential number starting with the specified number.

Custom Variable Attributes

In addition to the standard variable attributes (for example, value labels, missing values, measurement level), you can create your own custom variable attributes. Like standard variable attributes, these custom attributes are saved with PASW Statistics data files. Therefore, you could create a variable attribute that identifies the type of response for survey questions (for example, single selection, multiple selection, fill-in-the-blank) or the formulas used for computed variables.

Creating Custom Variable Attributes

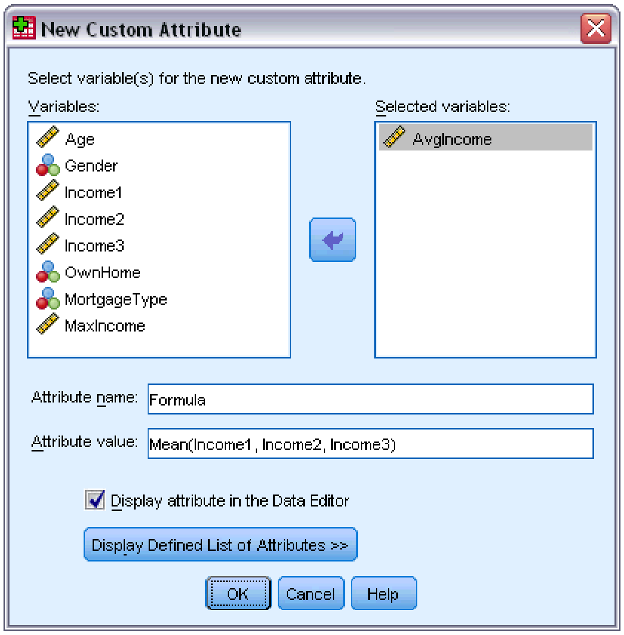
To create new custom attributes:

- In Variable View, from the menus choose:

  Data
  New Custom Attribute...
- Drag and drop the variables to which you want to assign the new attribute to the Selected Variables list.
- Enter a name for the attribute. Attribute names must follow the same rules as variable names. For more information, see the topic Variable Names on p. 70.

- ▶ Enter an optional value for the attribute. If you select multiple variables, the value is assigned to all selected variables. You can leave this blank and then enter values for each variable in Variable View.

Figure 5-6
New Custom Attribute dialog box



Select variable(s) for the new custom attribute.

Variables:
Age
Gender
Income1
Income2
Income3
OwnHome
MortgageType
MaxIncome

Selected variables:
AvgIncome

Attribute name: Formula

Attribute value: Mean(Income1 , Income2, Income3)

☑ Display attribute in the Data Editor

Display Defined List of Attributes >>

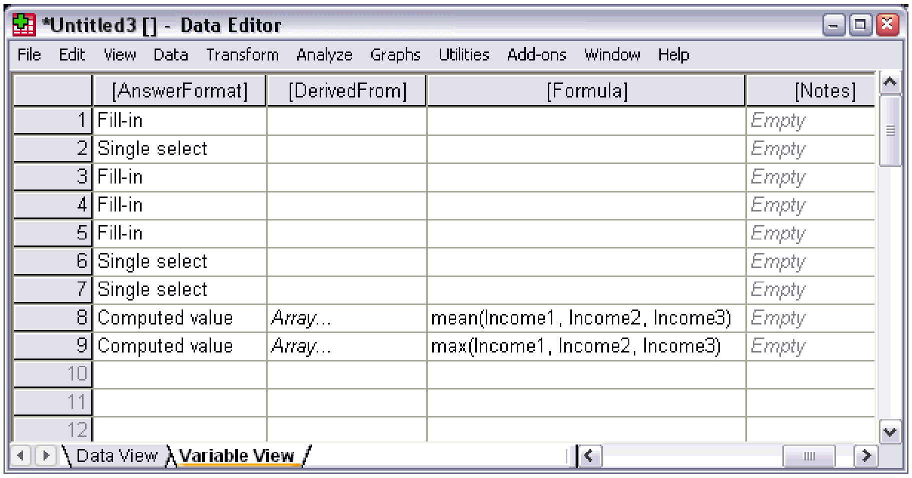
Display attribute in the Data Editor. Displays the attribute in Variable View of the Data Editor. For information on controlling the display of custom attributes, see Displaying and Editing Custom Variable Attributes below.

Display Defined List of Attributes. Displays a list of custom attributes already defined for the dataset. Attribute names that begin with a dollar sign ($) are reserved attributes that cannot be modified.

Displaying and Editing Custom Variable Attributes

Custom variable attributes can be displayed and edited in the Data Editor in Variable View.

Figure 5-7
Custom variable attributes displayed in Variable View


*Untitled3 [] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Data View
Variable View

	[AnswerFormat]	[DerivedFrom]	[Formula]	[Notes]
1	Fill-in			*Empty*
2	Single select			*Empty*
3	Fill-in			*Empty*
4	Fill-in			*Empty*
5	Fill-in			*Empty*
6	Single select			*Empty*
7	Single select			*Empty*
8	Computed value	*Array...*	mean(Income1, Income2, Income3)	*Empty*
9	Computed value	*Array...*	max(Income1, Income2, Income3)	*Empty*
10				
11				
12				

- Custom variable attribute names are enclosed in square brackets.
- Attribute names that begin with a dollar sign are reserved and cannot be modified.
- A blank cell indicates that the attribute does not exist for that variable; the text *Empty* displayed in a cell indicates that the attribute exists for that variable but no value has been assigned to the attribute for that variable. Once you enter text in the cell, the attribute exists for that variable with the value you enter.
- The text *Array...* displayed in a cell indicates that this is an attribute array—an attribute that contains multiple values. Click the button in the cell to display the list of values.

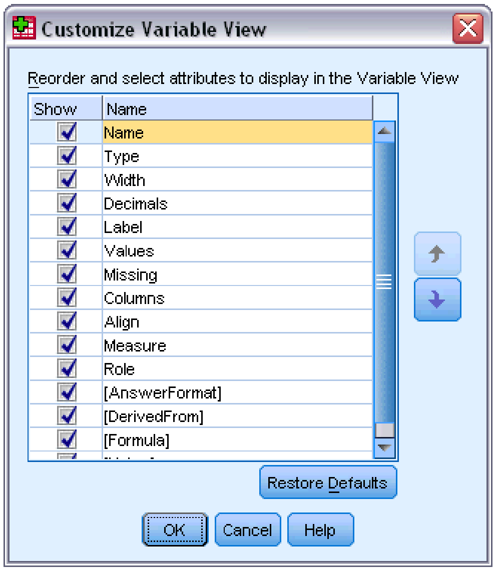
To Display and Edit Custom Variable Attributes

▶ In Variable View, from the menus choose:

View
  Customize Variable View...

▶ Select (check) the custom variable attributes you want to display. (The custom variable attributes are the ones enclosed in square brackets.)

Figure 5-8
Customize Variable View



Reorder and select attributes to display in the Variable View

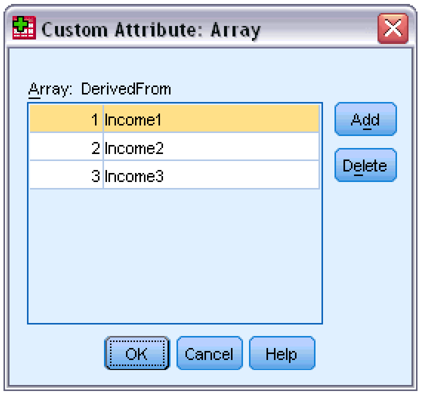
Show	Name
☑	Name
☑	Type
☑	Width
☑	Decimals
☑	Label
☑	Values
☑	Missing
☑	Columns
☑	Align
☑	Measure
☑	Role
☑	[AnswerFormat]
☑	[DerivedFrom]
☑	[Formula]

Once the attributes are displayed in Variable View, you can edit them directly in the Data Editor.

Variable Attribute Arrays

The text *Array...* (displayed in a cell for a custom variable attribute in Variable View or in the Custom Variable Properties dialog box in Define Variable Properties) indicates that this is an **attribute array**, an attribute that contains multiple values. For example, you could have an attribute array that identifies all of the source variables used to compute a derived variable. Click the button in the cell to display and edit the list of values.

Figure 5-9
Custom Attribute Array dialog box



Array: DerivedFrom

1	Income1
2	Income2
3	Income3

Customizing Variable View

You can use Customize Variable View to control which attributes are displayed in Variable View (for example, name, type, label) and the order in which they are displayed.

- Any custom variable attributes associated with the dataset are enclosed in square brackets. For more information, see the topic Creating Custom Variable Attributes on p. 78.
- Customized display settings are saved with PASW Statistics data files.
- You can also control the default display and order of attributes in Variable View. For more information, see the topic Changing the Default Variable View in Chapter 16 on p. 297.

To Customize Variable View

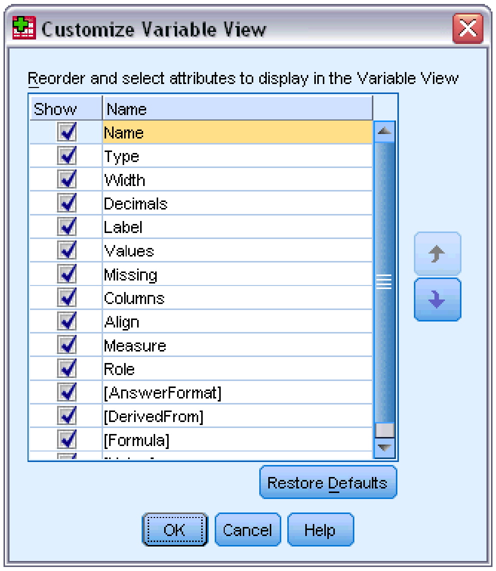
▶ In Variable View, from the menus choose:

View
  Customize Variable View...

▶ Select (check) the variable attributes you want to display.

▶ Use the up and down arrow buttons to change the display order of the attributes.

Figure 5-10
Customize Variable View dialog box



Customize Variable View

Reorder and select attributes to display in the Variable View

Show	Name
☑	Name
☑	Type
☑	Width
☑	Decimals
☑	Label
☑	Values
☑	Missing
☑	Columns
☑	Align
☑	Measure
☑	Role
☑	[AnswerFormat]
☑	[DerivedFrom]
☑	[Formula]

Restore Defaults. Apply the default display and order settings.

Spell Checking

Variable and Value Labels

To check the spelling of variable labels and value labels:

▶ Select the Variable View tab in the Data Editor window.

▶ Right-click on the *Labels* or *Values* column and from the context menu choose:
Spelling

or

▶ In Variable View, from the menus choose:
Utilities
  Spelling

or

▶ In the Value Labels dialog box, click Spelling. (This limits the spell checking to the value labels for a particular variable.)

Spell checking is limited to variable labels and value labels in Variable View of the Data Editor.

String Data Values

To check the spelling of string data values:

▶ Select the Data View tab of the Data Editor.

▶ Optionally, select one or more variables (columns) to check. To select a variable, click the variable name at the top of the column.

▶ From the menus choose:
Utilities
  Spelling

- If there are no selected variables in Data View, all string variables will be checked.
- If there are no string variables in the dataset or the none of the selected variables is a string variable, the Spelling option on the Utilities menu is disabled.

Entering Data

In Data View, you can enter data directly in the Data Editor. You can enter data in any order. You can enter data by case or by variable, for selected areas or for individual cells.

- The active cell is highlighted.
- The variable name and row number of the active cell are displayed in the top left corner of the Data Editor.
- When you select a cell and enter a data value, the value is displayed in the cell editor at the top of the Data Editor.
- Data values are not recorded until you press Enter or select another cell.
- To enter anything other than simple numeric data, you must define the variable type first.

If you enter a value in an empty column, the Data Editor automatically creates a new variable and assigns a variable name.

To Enter Numeric Data

- ▶ Select a cell in Data View.
- ▶ Enter the data value. (The value is displayed in the cell editor at the top of the Data Editor.)
- ▶ To record the value, press Enter or select another cell.

To Enter Non-Numeric Data

- ▶ Double-click a variable name at the top of the column in Data View or click the Variable View tab.
- ▶ Click the button in the *Type* cell for the variable.
- ▶ Select the data type in the Variable Type dialog box.
- ▶ Click OK.
- ▶ Double-click the row number or click the Data View tab.
- ▶ Enter the data in the column for the newly defined variable.

To Use Value Labels for Data Entry

- ▶ If value labels aren't currently displayed in Data View, from the menus choose:

  View
    Value Labels

- ▶ Click the cell in which you want to enter the value.
- ▶ Choose a value label from the drop-down list.

  The value is entered, and the value label is displayed in the cell.

  Note: This process works only if you have defined value labels for the variable.

Data Value Restrictions in the Data Editor

The defined variable type and width determine the type of value that can be entered in the cell in Data View.

- ■ If you type a character that is not allowed by the defined variable type, the character is not entered.
- ■ For string variables, characters beyond the defined width are not allowed.
- ■ For numeric variables, integer values that exceed the defined width can be entered, but the Data Editor displays either scientific notation or a portion of the value followed by an ellipsis (...) to indicate that the value is wider than the defined width. To display the value in the cell, change the defined width of the variable.

  Note: Changing the column width does not affect the variable width.

Editing Data

With the Data Editor, you can modify data values in Data View in many ways. You can:

- Change data values
- Cut, copy, and paste data values
- Add and delete cases
- Add and delete variables
- Change the order of variables

Replacing or Modifying Data Values

To Delete the Old Value and Enter a New Value

▶ In Data View, double-click the cell. (The cell value is displayed in the cell editor.)

▶ Edit the value directly in the cell or in the cell editor.

▶ Press Enter or select another cell to record the new value.

Cutting, Copying, and Pasting Data Values

You can cut, copy, and paste individual cell values or groups of values in the Data Editor. You can:

- Move or copy a single cell value to another cell
- Move or copy a single cell value to a group of cells
- Move or copy the values for a single case (row) to multiple cases
- Move or copy the values for a single variable (column) to multiple variables
- Move or copy a group of cell values to another group of cells

Data Conversion for Pasted Values in the Data Editor

If the defined variable types of the source and target cells are not the same, the Data Editor attempts to convert the value. If no conversion is possible, the system-missing value is inserted in the target cell.

Converting numeric or date into string. Numeric (for example, numeric, dollar, dot, or comma) and date formats are converted to strings if they are pasted into a string variable cell. The string value is the numeric value as displayed in the cell. For example, for a dollar format variable, the displayed dollar sign becomes part of the string value. Values that exceed the defined string variable width are truncated.

Converting string into numeric or date. String values that contain acceptable characters for the numeric or date format of the target cell are converted to the equivalent numeric or date value. For example, a string value of 25/12/91 is converted to a valid date if the format type of the target cell is one of the day-month-year formats, but the value is converted to system-missing if the format type of the target cell is one of the month-day-year formats.

Converting date into numeric. Date and time values are converted to a number of seconds if the target cell is one of the numeric formats (for example, numeric, dollar, dot, or comma). Because dates are stored internally as the number of seconds since October 14, 1582, converting dates to numeric values can yield some extremely large numbers. For example, the date 10/29/91 is converted to a numeric value of 12,908,073,600.

Converting numeric into date or time. Numeric values are converted to dates or times if the value represents a number of seconds that can produce a valid date or time. For dates, numeric values that are less than 86,400 are converted to the system-missing value.

Inserting New Cases

Entering data in a cell in a blank row automatically creates a new case. The Data Editor inserts the system-missing value for all other variables for that case. If there are any blank rows between the new case and the existing cases, the blank rows become new cases with the system-missing value for all variables. You can also insert new cases between existing cases.

To Insert New Cases between Existing Cases

▶ In Data View, select any cell in the case (row) below the position where you want to insert the new case.

▶ From the menus choose:

Edit
  Insert Cases

A new row is inserted for the case, and all variables receive the system-missing value.

Inserting New Variables

Entering data in an empty column in Data View or in an empty row in Variable View automatically creates a new variable with a default variable name (the prefix *var* and a sequential number) and a default data format type (numeric). The Data Editor inserts the system-missing value for all cases for the new variable. If there are any empty columns in Data View or empty rows in Variable View between the new variable and the existing variables, these rows or columns also become new variables with the system-missing value for all cases. You can also insert new variables between existing variables.

To Insert New Variables between Existing Variables

▶ Select any cell in the variable to the right of (Data View) or below (Variable View) the position where you want to insert the new variable.

▶ From the menus choose:

Edit
  Insert Variable

A new variable is inserted with the system-missing value for all cases.

To Move Variables

- ▶ To select the variable, click the variable name in Data View or the row number for the variable in Variable View.
- ▶ Drag and drop the variable to the new location.
- ▶ If you want to place the variable between two existing variables: In Data View, drop the variable on the variable column to the right of where you want to place the variable, or in Variable View, drop the variable on the variable row below where you want to place the variable.

To Change Data Type

You can change the data type for a variable at any time by using the Variable Type dialog box in Variable View. The Data Editor will attempt to convert existing values to the new type. If no conversion is possible, the system-missing value is assigned. The conversion rules are the same as the rules for pasting data values to a variable with a different format type. If the change in data format may result in the loss of missing-value specifications or value labels, the Data Editor displays an alert box and asks whether you want to proceed with the change or cancel it.

Finding Cases, Variables, or Imputations

The Go To dialog box finds the specified case (row) number or variable name in the Data Editor.

Cases

▶ For cases, from the menus choose:

Edit
  Go to Case...

▶ Enter an integer value that represents the current row number in Data View.

Note: The current row number for a particular case can change due to sorting and other actions.

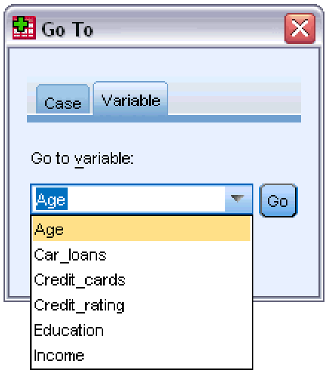
Variables

▶ For variables, from the menus choose:

Edit
  Go to Variable...

▶ Enter the variable name or select the variable from the drop-down list.

Figure 5-11
Go To dialog box



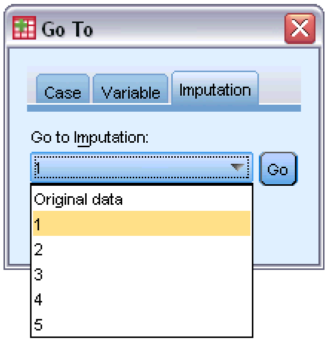
Imputations

▶ From the menus choose:

Edit
  Go to Imputation...

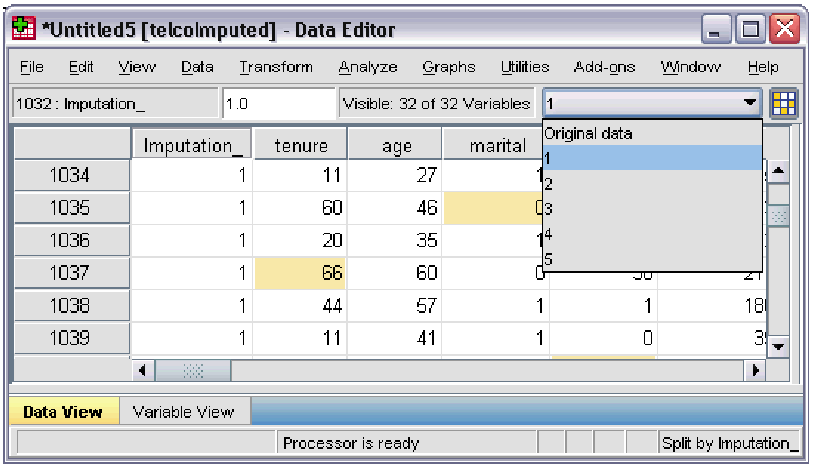
▶ Select the imputation (or Original data) from the drop-down list.

Figure 5-12
Go To dialog box



Alternatively, you can select the imputation from the drop-down list in the edit bar in Data View of the Data Editor.

Figure 5-13
Data Editor with imputation markings ON



*Untitled5 [telcoImputed] - Data Editor

1032 : Imputation_ 1.0 Visible: 32 of 32 Variables 1

	Imputation_	tenure	age	marital		
1034	1	11	27	1		
1035	1	60	46	0		
1036	1	20	35	1		
1037	1	66	60	0		
1038	1	44	57	1	1	18
1039	1	11	41	1	0	3

Processor is ready Split by Imputation_

Relative case position is preserved when selecting imputations. For example, if there are 1000 cases in the original dataset, case 1034, the 34th case in the first imputation, displays at the top of the grid. If you select imputation 2 in the dropdown, case 2034, the 34th case in imputation 2, would display at the top of the grid. If you select Original data in the dropdown, case 34 would display at the top of the grid. Column position is also preserved when navigating between imputations, so that it is easy to compare values between imputations.

Finding and Replacing Data and Attribute Values

To find and/or replace data values in Data View or attribute values in Variable View:

▶ Click a cell in the column you want to search. (Finding and replacing values is restricted to a single column.)

▶ From the menus choose:

Edit
  Find

or

Edit
  Replace

Data View

- You cannot search up in Data View. The search direction is always down.
- For dates and times, the formatted values as displayed in Data View are searched. For example, a date displayed as 10/28/2007 will not be found by a search for a date of 10-28-2007.
- For other numeric variables, Contains, Begins with, and Ends with search formatted values. For example, with the Begins with option, a search value of $123 for a Dollar format variable will find both $123.00 and $123.40 but not $1,234. With the Entire cell option, the search value can be formatted or unformatted (simple F numeric format), but only exact numeric values (to the precision displayed in the Data Editor) are matched.

- If value labels are displayed for the selected variable column, the label text is searched, not the underlying data value, and you cannot replace the label text.

Variable View

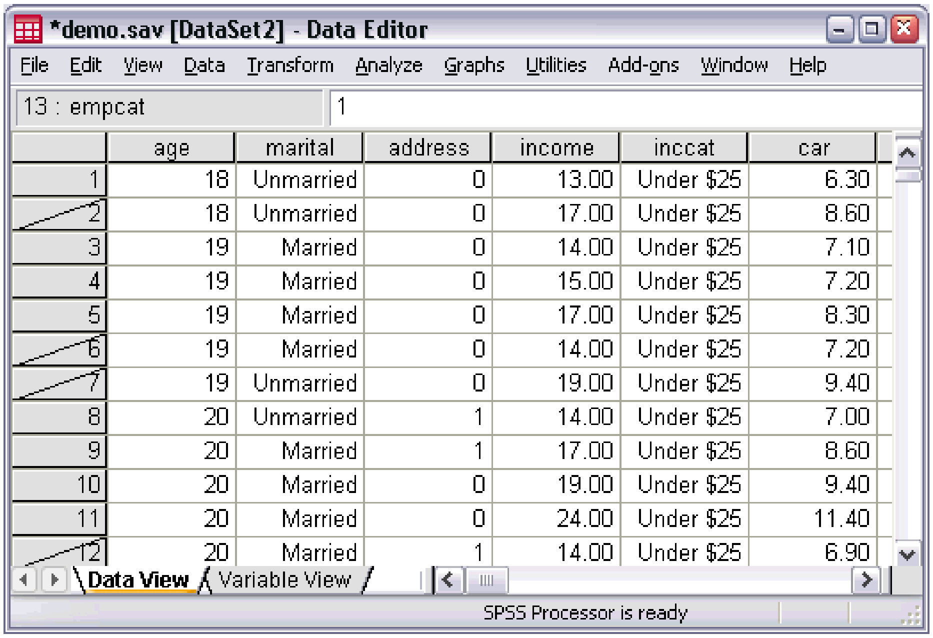
- Find is only available for the *Name*, *Label*, *Values*, *Missing*, and custom variable attribute columns.
- Replace is only available for the *Label*, *Values*, and custom attribute columns.
- In the *Values* (value labels) column, the search string can match either the data value or a value label.

  Note: Replacing the data value will delete any previous value label associated with that value.

Case Selection Status in the Data Editor

If you have selected a subset of cases but have not discarded unselected cases, unselected cases are marked in the Data Editor with a diagonal line (slash) through the row number.

Figure 5-14
Filtered cases in the Data Editor



13 : empcat 1

	age	marital	address	income	inccat	car
1	18	Unmarried	0	13.00	Under $25	6.30
2	18	Unmarried	0	17.00	Under $25	8.60
3	19	Married	0	14.00	Under $25	7.10
4	19	Married	0	15.00	Under $25	7.20
5	19	Married	0	17.00	Under $25	8.30
6	19	Married	0	14.00	Under $25	7.20
7	19	Unmarried	0	19.00	Under $25	9.40
8	20	Unmarried	1	14.00	Under $25	7.00
9	20	Married	1	17.00	Under $25	8.60
10	20	Married	0	19.00	Under $25	9.40
11	20	Married	0	24.00	Under $25	11.40
12	20	Married	1	14.00	Under $25	6.90

Data Editor Display Options

The View menu provides several display options for the Data Editor:

Fonts. This option controls the font characteristics of the data display.

Grid Lines. This option toggles the display of grid lines.

Value Labels. This option toggles between the display of actual data values and user-defined descriptive value labels. This option is available only in Data View.

Using Multiple Views

In Data View, you can create multiple views (panes) by using the splitters that are located below the horizontal scroll bar and to the right of the vertical scroll bar.

You can also use the Window menu to insert and remove pane splitters. To insert splitters:

▶ In Data View, from the menus choose:

Window
  Split

Splitters are inserted above and to the left of the selected cell.

- If the top left cell is selected, splitters are inserted to divide the current view approximately in half, both horizontally and vertically.
- If any cell other than the top cell in the first column is selected, a horizontal pane splitter is inserted above the selected cell.
- If any cell other than the first cell in the top row is selected, a vertical pane splitter is inserted to the left of the selected cell.

Data Editor Printing

A data file is printed as it appears on the screen.

- The information in the currently displayed view is printed. In Data View, the data are printed. In Variable View, data definition information is printed.
- Grid lines are printed if they are currently displayed in the selected view.
- Value labels are printed in Data View if they are currently displayed. Otherwise, the actual data values are printed.

Use the View menu in the Data Editor window to display or hide grid lines and toggle between the display of data values and value labels.

To Print Data Editor Contents

▶ Make the Data Editor the active window.

▶ Click the tab for the view that you want to print.

▶ From the menus choose:

File
  Print...

Chapter

6

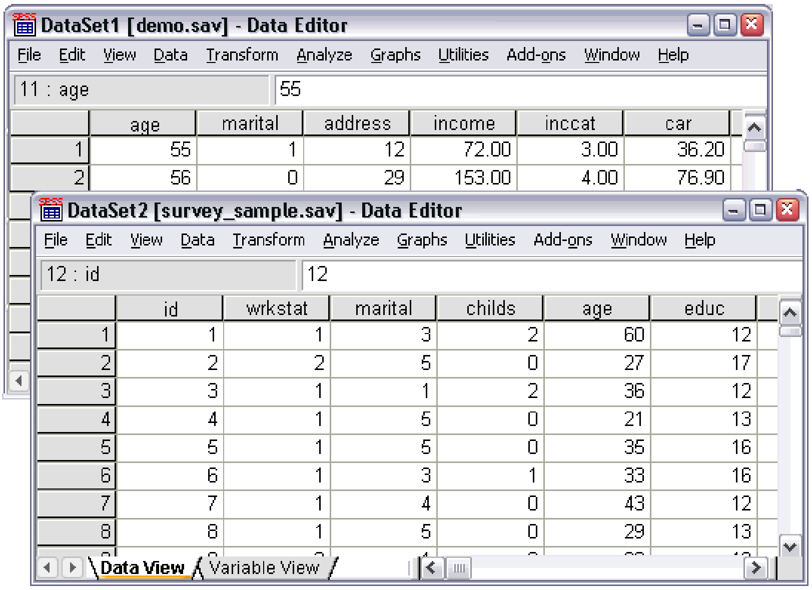
Working with Multiple Data Sources

Starting with version 14.0, multiple data sources can be open at the same time, making it easier to:

- Switch back and forth between data sources.
- Compare the contents of different data sources.
- Copy and paste data between data sources.
- Create multiple subsets of cases and/or variables for analysis.
- Merge multiple data sources from various data formats (for example, spreadsheet, database, text data) without saving each data source first.

Basic Handling of Multiple Data Sources

Figure 6-1
Two data sources open at same time


DataSet1 [demo.sav] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
DataSet2 [survey_sample.sav] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Data View
Variable View

11 : age 55

	age	marital	address	income	inccat	car
1	55	1	12	72.00	3.00	36.20
2	56	0	29	153.00	4.00	76.90

12 : id 12

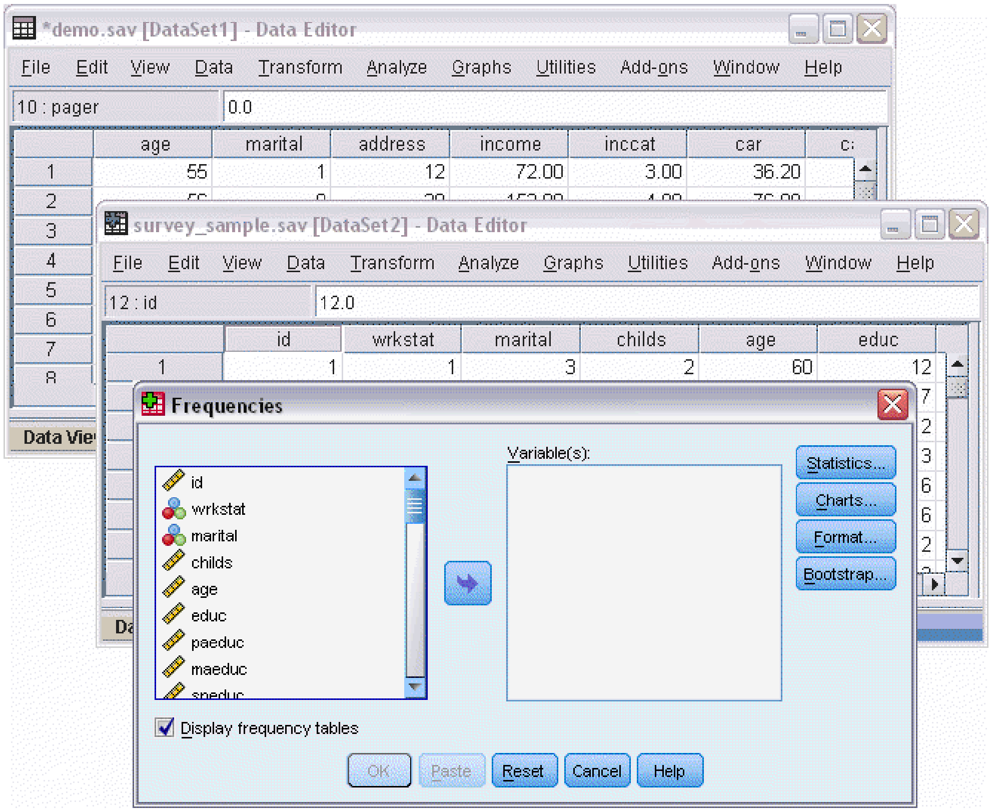
	id	wrkstat	marital	childs	age	educ
1	1	1	3	2	60	12
2	2	2	5	0	27	17
3	3	1	1	2	36	12
4	4	1	5	0	21	13
5	5	1	5	0	35	16
6	6	1	3	1	33	16
7	7	1	4	0	43	12
8	8	1	5	0	29	13

By default, each data source that you open is displayed in a new Data Editor window. (See General Options for information on changing the default behavior to only display one dataset at a time, in a single Data Editor window.)

- Any previously open data sources remain open and available for further use.
- When you first open a data source, it automatically becomes the **active dataset**.

- You can change the active dataset simply by clicking anywhere in the Data Editor window of the data source that you want to use or by selecting the Data Editor window for that data source from the Window menu.
- Only the variables in the active dataset are available for analysis.

Figure 6-2
Variable list containing variables in the active dataset


*demo.sav [DataSet1] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
10 : pager
0.0
Data View
survey_sample.sav [DataSet2] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
12 : id
12.0
Frequencies
Variable(s):
Statistics...
Charts...
Format...
Bootstrap...
OK
Paste
Reset
Cancel
Help

	age	marital	address	income	inccat	car
1	55	1	12	72.00	3.00	36.20

	id	wrkstat	marital	childs	age	educ
1	1	1	3	2	60	12

Frequencies variable list: id, wrkstat, marital, childs, age, educ, paeduc, maeduc, speduc

☑ Display frequency tables

- You cannot change the active dataset when any dialog box that accesses the data is open (including all dialog boxes that display variable lists).
- At least one Data Editor window must be open during a session. When you close the last open Data Editor window, PASW Statistics automatically shuts down, prompting you to save changes first.

Working with Multiple Datasets in Command Syntax

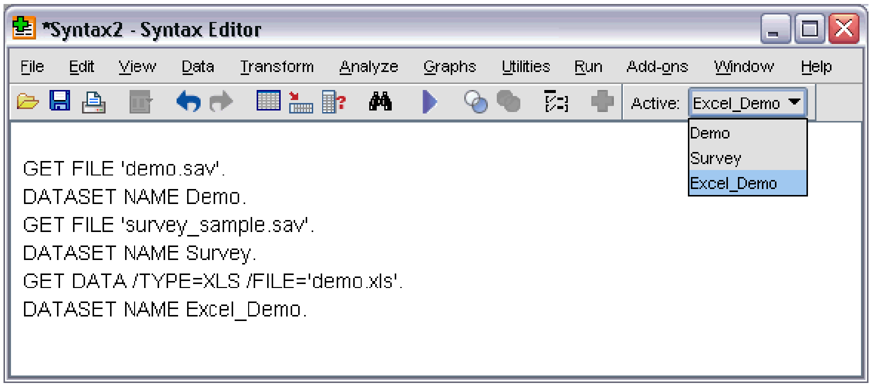
If you use command syntax to open data sources (for example, `GET FILE`, `GET DATA`), you need to use the `DATASET NAME` command to name each dataset explicitly in order to have more than one data source open at the same time.

When working with command syntax, the active dataset name is displayed on the toolbar of the syntax window. All of the following actions can change the active dataset:

- Use the `DATASET ACTIVATE` command.

- Click anywhere in the Data Editor window of a dataset.
- Select a dataset name from the toolbar in the syntax window.

Figure 6-3
Open datasets displayed on syntax window toolbar



```
GET FILE 'demo.sav'.
DATASET NAME Demo.
GET FILE 'survey_sample.sav'.
DATASET NAME Survey.
GET DATA /TYPE=XLS /FILE='demo.xls'.
DATASET NAME Excel_Demo.
```

Copying and Pasting Information between Datasets

You can copy both data and variable definition attributes from one dataset to another dataset in basically the same way that you copy and paste information within a single data file.

- Copying and pasting selected data cells in Data View pastes only the data values, with no variable definition attributes.
- Copying and pasting an entire variable in Data View by selecting the variable name at the top of the column pastes all of the data and all of the variable definition attributes for that variable.
- Copying and pasting variable definition attributes or entire variables in Variable View pastes the selected attributes (or the entire variable definition) but does not paste any data values.

Renaming Datasets

When you open a data source through the menus and dialog boxes, each data source is automatically assigned a dataset name of *DataSetn*, where *n* is a sequential integer value, and when you open a data source using command syntax, no dataset name is assigned unless you explicitly specify one with `DATASET NAME`. To provide more descriptive dataset names:

▶ From the menus in the Data Editor window for the dataset whose name you want to change choose:

File
  Rename Dataset...

▶ Enter a new dataset name that conforms to variable naming rules. For more information, see the topic Variable Names in Chapter 5 on p. 70.

Suppressing Multiple Datasets

If you prefer to have only one dataset available at a time and want to suppress the multiple dataset feature:

▶ From the menus choose:

Edit
  Options...

▶ Click the General tab.

Select (check) Open only one dataset at a time.

For more information, see the topic General Options in Chapter 16 on p. 291.

Chapter

7

Data Preparation

Once you've opened a data file or entered data in the Data Editor, you can start creating reports, charts, and analyses without any additional preliminary work. However, there are some additional data preparation features that you may find useful, including the ability to:

- Assign variable properties that describe the data and determine how certain values should be treated.
- Identify cases that may contain duplicate information and exclude those cases from analyses or delete them from the data file.
- Create new variables with a few distinct categories that represent ranges of values from variables with a large number of possible values.

Variable Properties

Data entered in the Data Editor in Data View or read from an external file format (such as an Excel spreadsheet or a text data file) lack certain variable properties that you may find very useful, including:

- Definition of descriptive value labels for numeric codes (for example, 0 = *Male* and 1 = *Female*).
- Identification of missing values codes (for example, 99 = *Not applicable*).
- Assignment of measurement level (nominal, ordinal, or scale).

All of these variable properties (and others) can be assigned in Variable View in the Data Editor. There are also several utilities that can assist you in this process:

- **Define Variable Properties** can help you define descriptive value labels and missing values. This is particularly useful for categorical data with numeric codes used for category values.
- **Copy Data Properties** provides the ability to use an existing PASW Statistics data file as a template for file and variable properties in the current data file. This is particularly useful if you frequently use external-format data files that contain similar content (such as monthly reports in Excel format).

Defining Variable Properties

Define Variable Properties is designed to assist you in the process of assigning attributes to variables, including creating descriptive value labels for categorical (nominal, ordinal) variables. Define Variable Properties:

- Scans the actual data values and lists all unique data values for each selected variable.

- Identifies unlabeled values and provides an “auto-label” feature.
- Provides the ability to copy defined value labels and other attributes from another variable to the selected variable or from the selected variable to multiple additional variables.

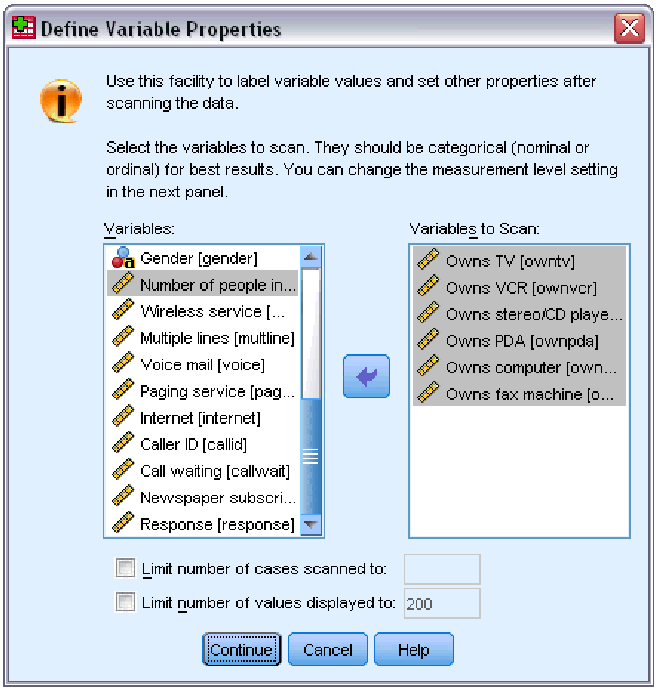
Note: To use Define Variable Properties without first scanning cases, enter 0 for the number of cases to scan.

To Define Variable Properties

▶ From the menus choose:

Data
  Define Variable Properties...

Figure 7-1
Initial dialog box for selecting variables to define



Define Variable Properties

Use this facility to label variable values and set other properties after scanning the data.

Select the variables to scan. They should be categorical (nominal or ordinal) for best results. You can change the measurement level setting in the next panel.

Variables:
Gender [gender]
Number of people in...
Wireless service [...
Multiple lines [multline]
Voice mail [voice]
Paging service [pag...
Internet [internet]
Caller ID [callid]
Call waiting [callwait]
Newspaper subscri...
Response [response]

Variables to Scan:
Owns TV [owntv]
Owns VCR [ownvcr]
Owns stereo/CD playe...
Owns PDA [ownpda]
Owns computer [own...
Owns fax machine [o...

□ Limit number of cases scanned to:
□ Limit number of values displayed to: 200

▶ Select the numeric or string variables for which you want to create value labels or define or change other variable properties, such as missing values or descriptive variable labels.

▶ Specify the number of cases to scan to generate the list of unique values. This is particularly useful for data files with a large number of cases for which a scan of the complete data file might take a significant amount of time.

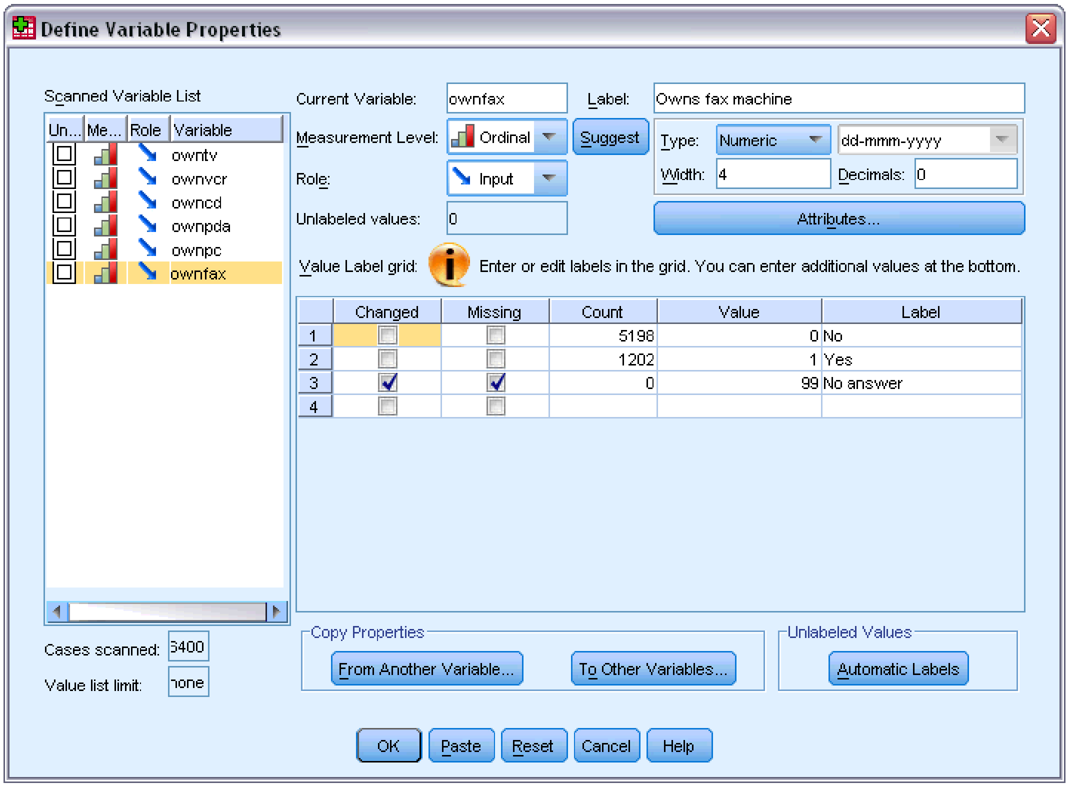
▶ Specify an upper limit for the number of unique values to display. This is primarily useful to prevent listing hundreds, thousands, or even millions of values for scale (continuous interval, ratio) variables.

▶ Click Continue to open the main Define Variable Properties dialog box.

- ▶ Select a variable for which you want to create value labels or define or change other variable properties.
- ▶ Enter the label text for any unlabeled values that are displayed in the Value Label grid.
- ▶ If there are values for which you want to create value labels but those values are not displayed, you can enter values in the *Value* column below the last scanned value.
- ▶ Repeat this process for each listed variable for which you want to create value labels.
- ▶ Click OK to apply the value labels and other variable properties.

Defining Value Labels and Other Variable Properties

Figure 7-2
Define Variable Properties, main dialog box



Define Variable Properties

Scanned Variable List

Un...	Me...	Role	Variable
□			owntv
□			ownvcr
□			owncd
□			ownpda
□			ownpc
□			ownfax

Cases scanned: 6400

Value list limit: none

Current Variable: ownfax

Label: Owns fax machine

Measurement Level: Ordinal

Type: Numeric

dd-mmm-yyyy

Role: Input

Width: 4

Decimals: 0

Unlabeled values: 0

Value Label grid: Enter or edit labels in the grid. You can enter additional values at the bottom.

	Changed	Missing	Count	Value	Label
1	□	□	5198	0	No
2	□	□	1202	1	Yes
3	☑	☑	0	99	No answer
4	□	□			

The Define Variable Properties main dialog box provides the following information for the scanned variables:

Scanned Variable List. For each scanned variable, a check mark in the *Unlabeled* (*U.*) column indicates that the variable contains values without assigned value labels.

To sort the variable list to display all variables with unlabeled values at the top of the list:

- ▶ Click the *Unlabeled* column heading under Scanned Variable List.

You can also sort by variable name or measurement level by clicking the corresponding column heading under Scanned Variable List.

Value Label Grid

- **Label.** Displays any value labels that have already been defined. You can add or change labels in this column.
- **Value.** Unique values for each selected variable. This list of unique values is based on the number of scanned cases. For example, if you scanned only the first 100 cases in the data file, then the list reflects only the unique values present in those cases. If the data file has already been sorted by the variable for which you want to assign value labels, the list may display far fewer unique values than are actually present in the data.
- **Count.** The number of times each value occurs in the scanned cases.
- **Missing.** Values defined as representing missing data. You can change the missing values designation of the category by clicking the check box. A check indicates that the category is defined as a user-missing category. If a variable already has a range of values defined as user-missing (for example, 90-99), you cannot add or delete missing values categories for that variable with Define Variable Properties. You can use Variable View in the Data Editor to modify the missing values categories for variables with missing values ranges. For more information, see the topic Missing Values in Chapter 5 on p. 75.
- **Changed.** Indicates that you have added or changed a value label.

Note: If you specified 0 for the number of cases to scan in the initial dialog box, the Value Label grid will initially be blank, except for any preexisting value labels and/or defined missing values categories for the selected variable. In addition, the Suggest button for the measurement level will be disabled.

Measurement Level. Value labels are primarily useful for categorical (nominal and ordinal) variables, and some procedures treat categorical and scale variables differently; so it is sometimes important to assign the correct measurement level. However, by default, all new numeric variables are assigned the scale measurement level. Thus, many variables that are in fact categorical may initially be displayed as scale.

If you are unsure of what measurement level to assign to a variable, click Suggest.

Role. Some dialogs support the ability to pre-select variables for analysis based on defined roles. For more information, see the topic Roles in Chapter 5 on p. 76.

Copy Properties. You can copy value labels and other variable properties from another variable to the currently selected variable or from the currently selected variable to one or more other variables.

Unlabeled Values. To create labels for unlabeled values automatically, click Automatic Labels.

Variable Label and Display Format

You can change the descriptive variable label and the display format.

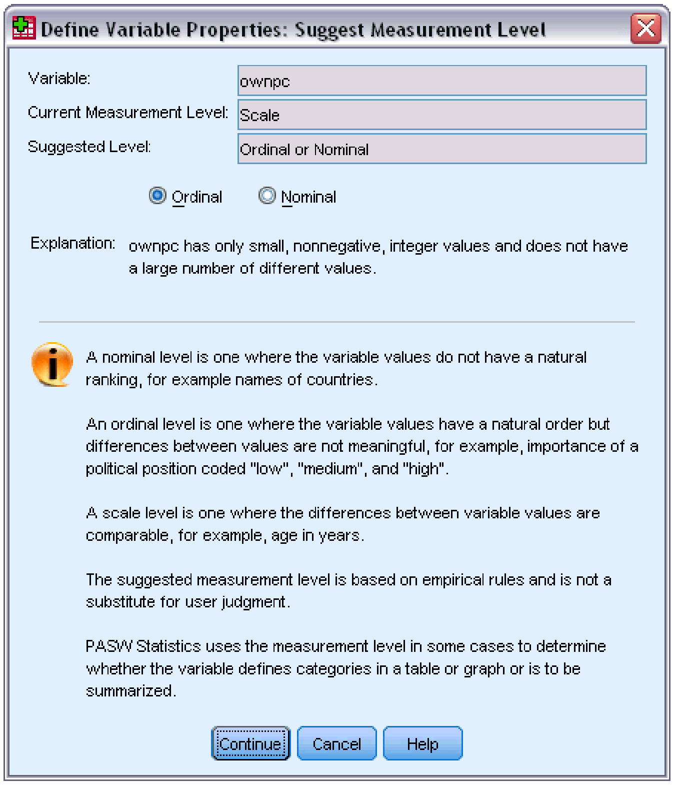
- You cannot change the variable’s fundamental type (string or numeric).
- For string variables, you can change only the variable label, not the display format.

- For numeric variables, you can change the numeric type (such as numeric, date, dollar, or custom currency), width (maximum number of digits, including any decimal and/or grouping indicators), and number of decimal positions.
- For numeric date format, you can select a specific date format (such as dd-mm-yyyy, mm/dd/yy, and yyyyddd)
- For numeric custom format, you can select one of five custom currency formats (CCA through CCE). For more information, see the topic Currency Options in Chapter 16 on p. 297.
- An asterisk is displayed in the *Value* column if the specified width is less than the width of the scanned values or the displayed values for preexisting defined value labels or missing values categories.
- A period (.) is displayed if the scanned values or the displayed values for preexisting defined value labels or missing values categories are invalid for the selected display format type. For example, an internal numeric value of less than 86,400 is invalid for a date format variable.

Assigning the Measurement Level

When you click Suggest for the measurement level in the Define Variable Properties main dialog box, the current variable is evaluated based on the scanned cases and defined value labels, and a measurement level is suggested in the Suggest Measurement Level dialog box that opens. The Explanation area provides a brief description of the criteria used to provide the suggested measurement level.

Figure 7-3
Suggest Measurement Level dialog box



Variable: ownpc

Current Measurement Level: Scale

Suggested Level: Ordinal or Nominal

● Ordinal ○ Nominal

Explanation: ownpc has only small, nonnegative, integer values and does not have a large number of different values.

A nominal level is one where the variable values do not have a natural ranking, for example names of countries.

An ordinal level is one where the variable values have a natural order but differences between values are not meaningful, for example, importance of a political position coded "low", "medium", and "high".

A scale level is one where the differences between variable values are comparable, for example, age in years.

The suggested measurement level is based on empirical rules and is not a substitute for user judgment.

PASW Statistics uses the measurement level in some cases to determine whether the variable defines categories in a table or graph or is to be summarized.

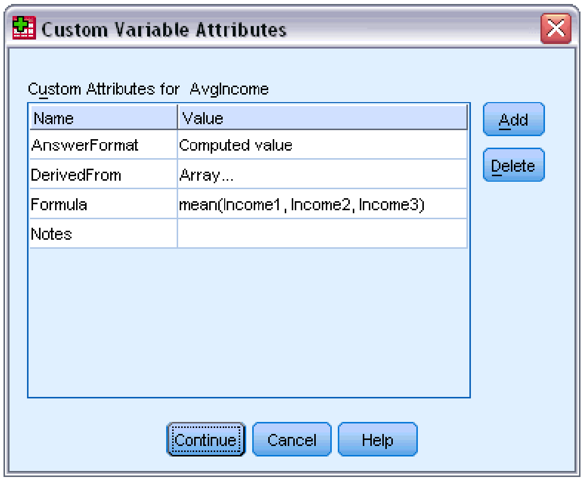
Note: Values defined as representing missing values are not included in the evaluation for measurement level. For example, the explanation for the suggested measurement level may indicate that the suggestion is in part based on the fact that the variable contains no negative values, whereas it may in fact contain negative values—but those values are already defined as missing values.

▶ Click Continue to accept the suggested level of measurement or Cancel to leave the measurement level unchanged.

Custom Variable Attributes

The Attributes button in Define Variable Properties opens the Custom Variable Attributes dialog box. In addition to the standard variable attributes, such as value labels, missing values, and measurement level, you can create your own custom variable attributes. Like standard variable attributes, these custom attributes are saved with PASW Statistics data files.

Figure 7-4
Custom Variable Attributes



Custom Attributes for AvgIncome

Name	Value
AnswerFormat	Computed value
DerivedFrom	Array...
Formula	mean(Income1, Income2, Income3)
Notes	

Name. Attribute names must follow the same rules as variable names. For more information, see the topic Variable Names in Chapter 5 on p. 70.

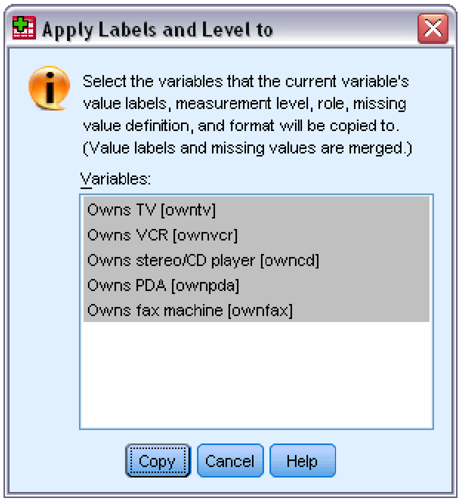
Value. The value assigned to the attribute for the selected variable.

- Attribute names that begin with a dollar sign are reserved and cannot be modified. You can view the contents of a reserved attribute by clicking on the button in the desired cell.
- The text *Array...*, displayed in a Value cell, indicates that this is an **attribute array**, an attribute that contains multiple values. Click the button in the cell to display the list of values.

Copying Variable Properties

The Apply Labels and Level dialog box is displayed when you click From Another Variable or To Other Variables in the Define Variable Properties main dialog box. It displays all of the scanned variables that match the current variable's type (numeric or string). For string variables, the defined width must also match.

Figure 7-5
Apply Labels and Level dialog box


Apply Labels and Level to

Select the variables that the current variable's value labels, measurement level, role, missing value definition, and format will be copied to. (Value labels and missing values are merged.)

Variables:

Owns TV [owntv]
Owns VCR [ownvcr]
Owns stereo/CD player [owncd]
Owns PDA [ownpda]
Owns fax machine [ownfax]

▶ Select a single variable from which to copy value labels and other variable properties (except variable label).

or

▶ Select one or more variables to which to copy value labels and other variable properties.

▶ Click Copy to copy the value labels and the measurement level.

- Existing value labels and missing value categories for target variable(s) are not replaced.
- Value labels and missing value categories for values not already defined for the target variable(s) are added to the set of value labels and missing value categories for the target variable(s).
- The measurement level for the target variable is always replaced.
- The role for the target variable is always replaced.
- If either the source or target variable has a defined range of missing values, missing values definitions are not copied.

Multiple Response Sets

Custom Tables and the Chart Builder support a special kind of “variable” called a **multiple response set**. Multiple response sets aren’t really “variables” in the normal sense. You can’t see them in the Data Editor, and other procedures don’t recognize them. Multiple response sets use multiple variables to record responses to questions where the respondent can give more than one answer. Multiple response sets are treated like categorical variables, and most of the things you can do with categorical variables, you can also do with multiple response sets.

Multiple response sets are constructed from multiple variables in the data file. A multiple response set is a special construct within a data file. You can define and save multiple response sets in PASW Statistics data files, but you cannot import or export multiple response sets from/to other file formats. You can copy multiple response sets from other PASW Statistics data files using

Copy Data Properties, which is accessed from the Data menu in the Data Editor window. For more information, see the topic Copying Data Properties on p. 106.

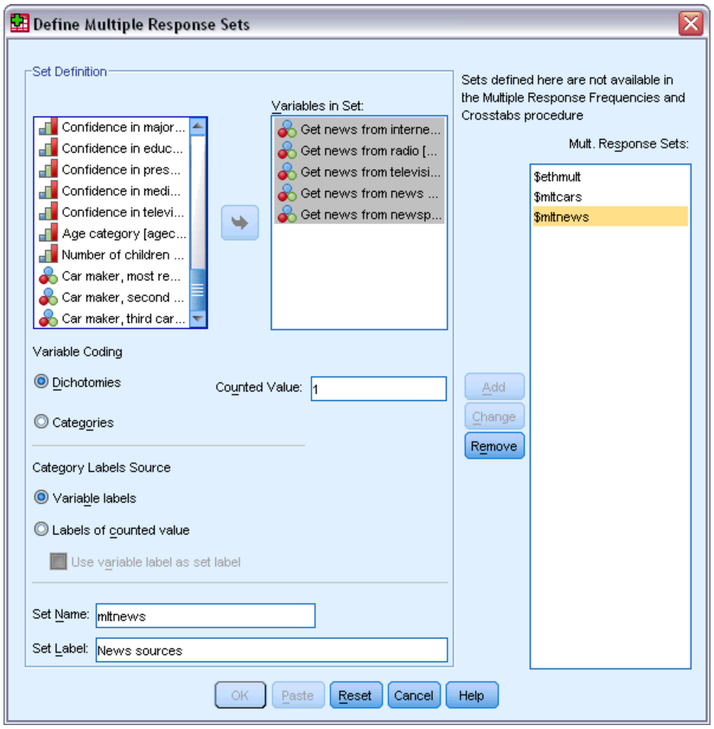
Defining Multiple Response Sets

To define multiple response sets:

▶ From the menus, choose:

Data
  Define Multiple Response Sets...

Figure 7-6
Define Multiple Response Sets dialog box



Define Multiple Response Sets

Set Definition

- Confidence in major...
- Confidence in educ...
- Confidence in pres...
- Confidence in medi...
- Confidence in televi...
- Age category [agec...
- Number of children ...
- Car maker, most re...
- Car maker, second ...
- Car maker, third car...

Variables in Set:

- Get news from interne...
- Get news from radio [...
- Get news from televisi...
- Get news from news ...
- Get news from newsp...

Sets defined here are not available in the Multiple Response Frequencies and Crosstabs procedure

Mult. Response Sets:

- $ethmult
- $mltcars
- $mltnews

Variable Coding

● Dichotomies   Counted Value: 1

○ Categories

Category Labels Source

● Variable labels

○ Labels of counted value

□ Use variable label as set label

Set Name: mltnews

Set Label: News sources

▶ Select two or more variables. If your variables are coded as dichotomies, indicate which value you want to have counted.

▶ Enter a unique name for each multiple response set. The name can be up to 63 bytes long. A dollar sign is automatically added to the beginning of the set name.

▶ Enter a descriptive label for the set. (This is optional.)

▶ Click Add to add the multiple response set to the list of defined sets.

Dichotomies

A multiple dichotomy set typically consists of multiple dichotomous variables: variables with only two possible values of a yes/no, present/absent, checked/not checked nature. Although the variables may not be strictly dichotomous, all of the variables in the set are coded the same way, and the Counted Value represents the positive/present/checked condition.

For example, a survey asks the question, “Which of the following sources do you rely on for news?” and provides five possible responses. The respondent can indicate multiple choices by checking a box next to each choice. The five responses become five variables in the data file, coded 0 for *No* (not checked) and 1 for *Yes* (checked). In the multiple dichotomy set, the Counted Value is 1.

The sample data file *survey_sample.sav* already has three defined multiple response sets. *$mltnews* is a multiple dichotomy set.

▶ Select (click) *$mltnews* in the Mult. Response Sets list.

This displays the variables and settings used to define this multiple response set.

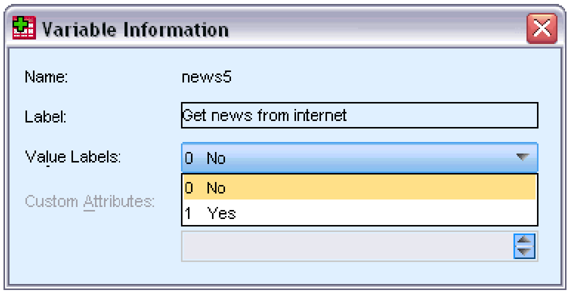
- The Variables in Set list displays the five variables used to construct the multiple response set.
- The Variable Coding group indicates that the variables are dichotomous.
- The Counted Value is 1.

▶ Select (click) one of the variables in the Variables in Set list.

▶ Right-click the variable and select Variable Information from the pop-up context menu.

▶ In the Variable Information window, click the arrow on the Value Labels drop-down list to display the entire list of defined value labels.

Figure 7-7
Variable information for multiple dichotomy source variable



Name: news5

Label: Get news from internet

Value Labels: 0 No

0 No
1 Yes

The value labels indicate that the variable is a dichotomy with values of 0 and 1, representing *No* and *Yes*, respectively. All five variables in the list are coded the same way, and the value of 1 (the code for *Yes*) is the counted value for the multiple dichotomy set.

Categories

A multiple category set consists of multiple variables, all coded the same way, often with many possible response categories. For example, a survey item states, “Name up to three nationalities that best describe your ethnic heritage.” There may be hundreds of possible responses, but for coding purposes the list is limited to the 40 most common nationalities, with everything else relegated to an "other" category. In the data file, the three choices become three variables, each with 41 categories (40 coded nationalities and one "other" category).

In the sample data file, *$ethmult* and *$mltcars* are multiple category sets.

Category Label Source

For multiple dichotomies, you can control how sets are labeled.

- **Variable labels.** Uses the defined variable labels (or variable names for variables without defined variable labels) as the set category labels. For example, if all of the variables in the set have the same value label (or no defined value labels) for the counted value (for example, *Yes*), then you should use the variable labels as the set category labels.
- **Labels of counted values.** Uses the defined value labels of the counted values as set category labels. Select this option only if all variables have a defined value label for the counted value and the value label for the counted value is different for each variable.
- **Use variable label as set label.** If you select Label of counted values, you can also use the variable label for the first variable in the set with a defined variable label as the set label. If none of the variables in the set have defined variable labels, the name of the first variable in the set is used as the set label.

Copying Data Properties

The Copy Data Properties Wizard provides the ability to use an external PASW Statistics data file as a template for defining file and variable properties in the active dataset. You can also use variables in the active dataset as templates for other variables in the active dataset. You can:

- Copy selected file properties from an external data file or open dataset to the active dataset. File properties include documents, file labels, multiple response sets, variable sets, and weighting.
- Copy selected variable properties from an external data file or open dataset to matching variables in the active dataset. Variable properties include value labels, missing values, level of measurement, variable labels, print and write formats, alignment, and column width (in the Data Editor).
- Copy selected variable properties from one variable in either an external data file, open dataset, or the active dataset to many variables in the active dataset.
- Create new variables in the active dataset based on selected variables in an external data file or open dataset.

When copying data properties, the following general rules apply:

- If you use an external data file as the source data file, it must be a data file in PASW Statistics format.
- If you use the active dataset as the source data file, it must contain at least one variable. You cannot use a completely blank active dataset as the source data file.
- Undefined (empty) properties in the source dataset do not overwrite defined properties in the active dataset.
- Variable properties are copied from the source variable only to target variables of a matching type—string (alphanumeric) or numeric (including numeric, date, and currency).

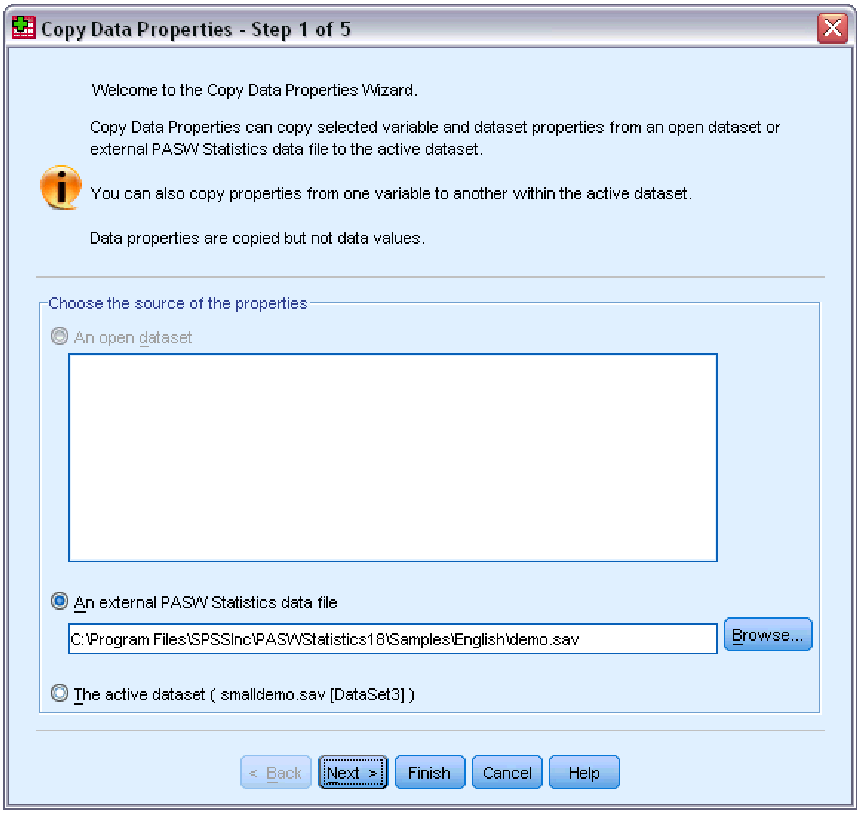
Note: Copy Data Properties replaces Apply Data Dictionary, formerly available on the File menu.

To Copy Data Properties

▶ From the menus in the Data Editor window choose:

Data
  Copy Data Properties...

Figure 7-8
Copy Data Properties Wizard: Step 1



Welcome to the Copy Data Properties Wizard.

Copy Data Properties can copy selected variable and dataset properties from an open dataset or external PASW Statistics data file to the active dataset.

You can also copy properties from one variable to another within the active dataset.

Data properties are copied but not data values.

Choose the source of the properties

○ An open dataset

● An external PASW Statistics data file

C:\Program Files\SPSSInc\PASWStatistics18\Samples\English\demo.sav

○ The active dataset (smalldemo.sav [DataSet3])

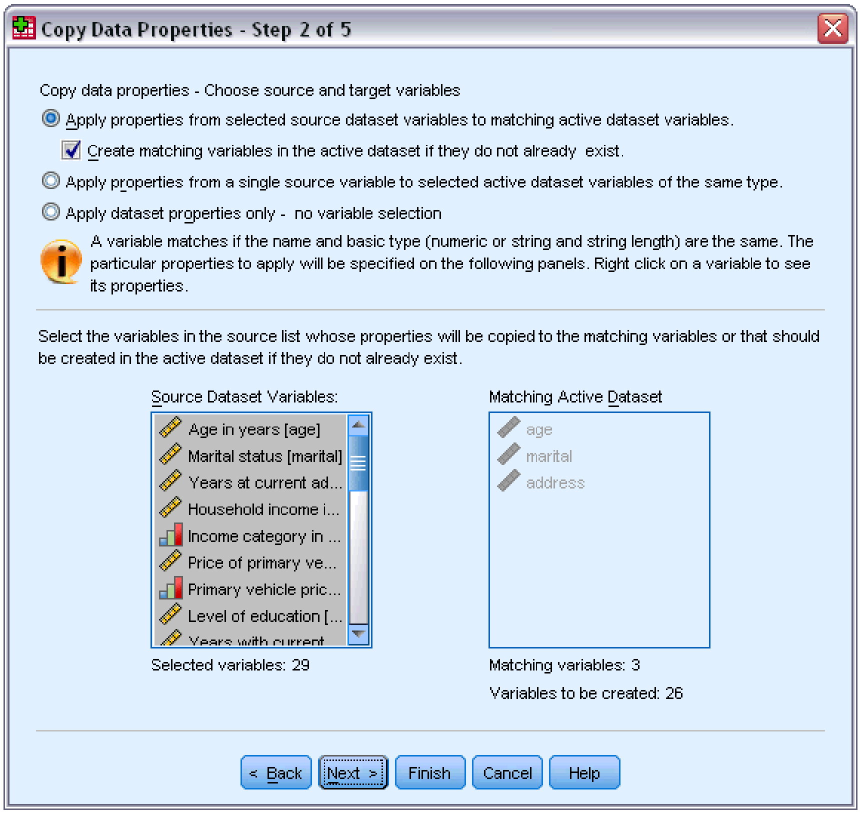
▶ Select the data file with the file and/or variable properties that you want to copy. This can be a currently open dataset, an external PASW Statistics data file, or the active dataset.

▶ Follow the step-by-step instructions in the Copy Data Properties Wizard.

Selecting Source and Target Variables

In this step, you can specify the source variables containing the variable properties that you want to copy and the target variables that will receive those variable properties.

Figure 7-9
Copy Data Properties Wizard: Step 2



Copy Data Properties - Step 2 of 5

Copy data properties - Choose source and target variables

● Apply properties from selected source dataset variables to matching active dataset variables.

☑ Create matching variables in the active dataset if they do not already exist.

○ Apply properties from a single source variable to selected active dataset variables of the same type.

○ Apply dataset properties only - no variable selection

A variable matches if the name and basic type (numeric or string and string length) are the same. The particular properties to apply will be specified on the following panels. Right click on a variable to see its properties.

Select the variables in the source list whose properties will be copied to the matching variables or that should be created in the active dataset if they do not already exist.

Source Dataset Variables:

- Age in years [age]
- Marital status [marital]
- Years at current ad...
- Household income i...
- Income category in ...
- Price of primary ve...
- Primary vehicle pric...
- Level of education [...
- Years with current ...

Selected variables: 29

Matching Active Dataset

- age
- marital
- address

Matching variables: 3

Variables to be created: 26

Apply properties from selected source dataset variables to matching active dataset variables. Variable properties are copied from one or more selected source variables to matching variables in the active dataset. Variables "match" if both the variable name and type (string or numeric) are the same. For string variables, the defined length must also be the same. By default, only matching variables are displayed in the two variable lists.

- **Create matching variables in the active dataset if they do not already exist.** This updates the source list to display all variables in the source data file. If you select source variables that do not exist in the active dataset (based on variable name), new variables will be created in the active dataset with the variable names and properties from the source data file.

If the active dataset contains no variables (a blank, new dataset), all variables in the source data file are displayed and new variables based on the selected source variables are automatically created in the active dataset.

Apply properties from a single source variable to selected active dataset variables of the same type. Variable properties from a single selected variable in the source list can be applied to one or more selected variables in the active dataset list. Only variables of the same type (numeric or string) as the selected variable in the source list are displayed in the active dataset list. For string variables, only strings of the same defined length as the source variable are displayed. This option is not available if the active dataset contains no variables.

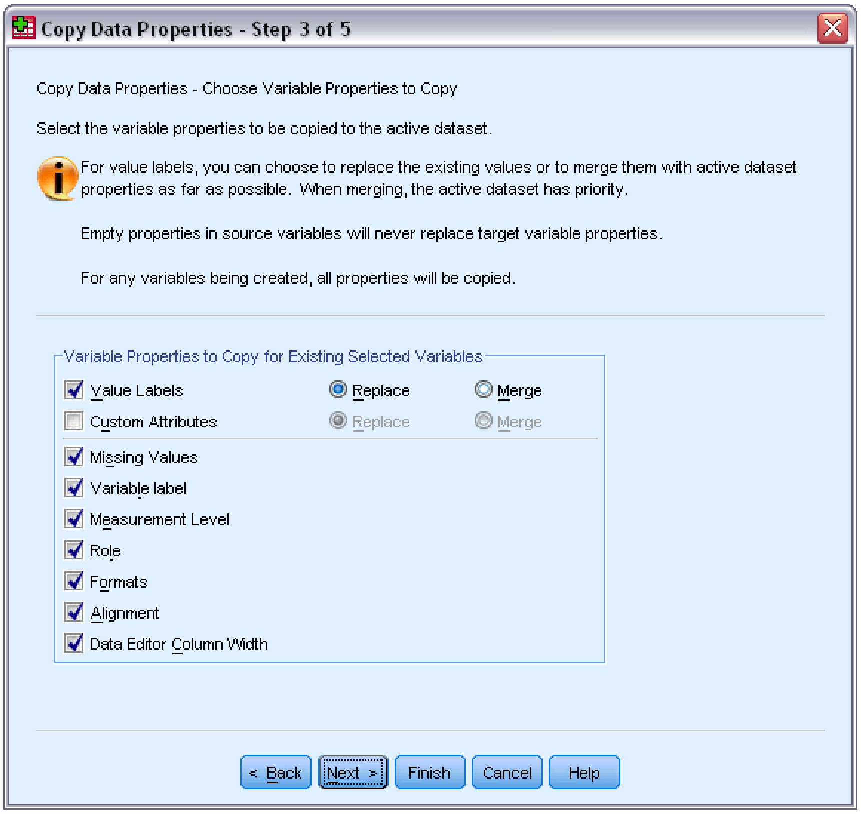
Note: You cannot create new variables in the active dataset with this option.

Apply dataset properties only—no variable selection. Only file properties (for example, documents, file label, weight) will be applied to the active dataset. No variable properties will be applied. This option is not available if the active dataset is also the source data file.

Choosing Variable Properties to Copy

You can copy selected variable properties from the source variables to the target variables. Undefined (empty) properties in the source variables do not overwrite defined properties in the target variables.

Figure 7-10
Copy Data Properties Wizard: Step 3



Copy Data Properties - Step 3 of 5

Copy Data Properties - Choose Variable Properties to Copy

Select the variable properties to be copied to the active dataset.

For value labels, you can choose to replace the existing values or to merge them with active dataset properties as far as possible. When merging, the active dataset has priority.

Empty properties in source variables will never replace target variable properties.

For any variables being created, all properties will be copied.

Variable Properties to Copy for Existing Selected Variables

☑ Value Labels ● Replace ○ Merge
☐ Custom Attributes ● Replace ○ Merge
☑ Missing Values
☑ Variable label
☑ Measurement Level
☑ Role
☑ Formats
☑ Alignment
☑ Data Editor Column Width

Value Labels. Value labels are descriptive labels associated with data values. Value labels are often used when numeric data values are used to represent non-numeric categories (for example, codes of 1 and 2 for *Male* and *Female*). You can replace or merge value labels in the target variables.

- Replace deletes any defined value labels for the target variable and replaces them with the defined value labels from the source variable.
- Merge merges the defined value labels from the source variable with any existing defined value label for the target variable. If the same value has a defined value label in both the source and target variables, the value label in the target variable is unchanged.

Custom Attributes. User-defined custom variable attributes. For more information, see the topic Custom Variable Attributes in Chapter 5 on p. 78.

- Replace deletes any custom attributes for the target variable and replaces them with the defined attributes from the source variable.
- Merge merges the defined attributes from the source variable with any existing defined attributes for the target variable.

Missing Values. Missing values are values identified as representing missing data (for example, 98 for *Do not know* and 99 for *Not applicable*). Typically, these values also have defined value labels that describe what the missing value codes stand for. Any existing defined missing values for the target variable are deleted and replaced with the defined missing values from the source variable.

Variable Label. Descriptive variable labels can contain spaces and reserved characters not allowed in variable names. If you're copying variable properties from a single source variable to multiple target variables, you might want to think twice before selecting this option.

Measurement Level. The measurement level can be nominal, ordinal, or scale.

Role. Some dialogs support the ability to pre-select variables for analysis based on defined roles. For more information, see the topic Roles in Chapter 5 on p. 76.

Formats. For numeric variables, this controls numeric type (such as numeric, date, or currency), width (total number of displayed characters, including leading and trailing characters and decimal indicator), and number of decimal places displayed. This option is ignored for string variables.

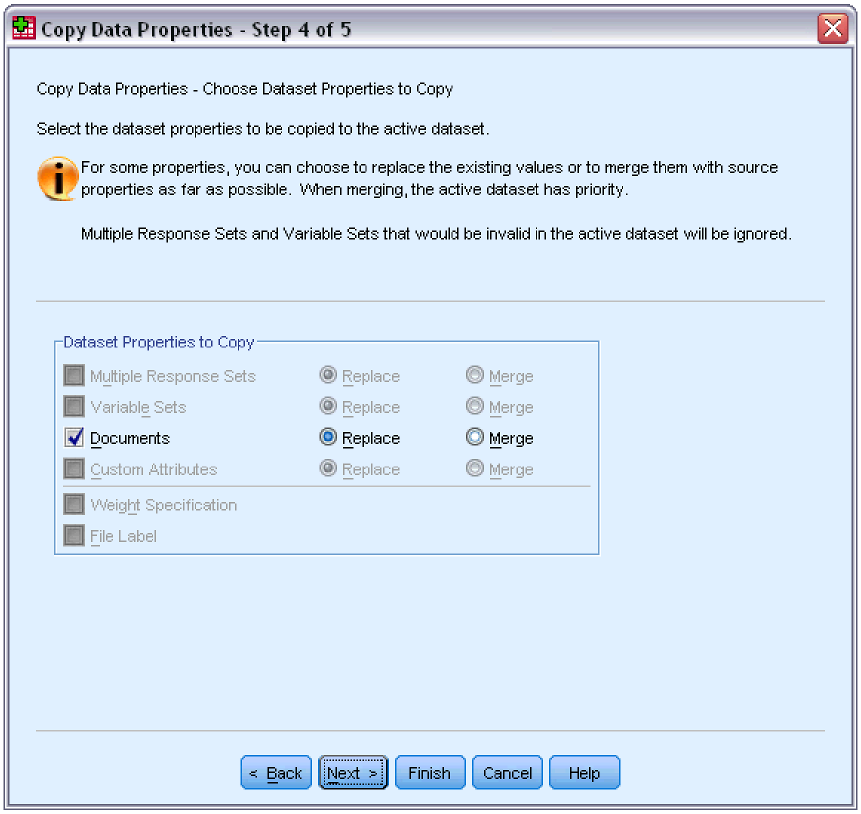
Alignment. This affects only alignment (left, right, center) in Data View in the Data Editor.

Data Editor Column Width. This affects only column width in Data View in the Data Editor.

Copying Dataset (File) Properties

You can apply selected, global dataset properties from the source data file to the active dataset. (This is not available if the active dataset is the source data file.)

Figure 7-11
Copy Data Properties Wizard: Step 4



Copy Data Properties - Step 4 of 5

Copy Data Properties - Choose Dataset Properties to Copy

Select the dataset properties to be copied to the active dataset.

For some properties, you can choose to replace the existing values or to merge them with source properties as far as possible. When merging, the active dataset has priority.

Multiple Response Sets and Variable Sets that would be invalid in the active dataset will be ignored.

Dataset Properties to Copy

□ Multiple Response Sets	● Replace	○ Merge
□ Variable Sets	● Replace	○ Merge
☑ Documents	● Replace	○ Merge
□ Custom Attributes	● Replace	○ Merge
□ Weight Specification		
□ File Label		

Multiple Response Sets. Applies multiple response set definitions from the source data file to the active dataset.

- Multiple response sets that contain no variables in the active dataset are ignored unless those variables will be created based on specifications in step 2 (Selecting Source and Target Variables) in the Copy Data Properties Wizard.
- Replace deletes all multiple response sets in the active dataset and replaces them with the multiple response sets from the source data file.
- Merge adds multiple response sets from the source data file to the collection of multiple response sets in the active dataset. If a set with the same name exists in both files, the existing set in the active dataset is unchanged.

Variable Sets. Variable sets are used to control the list of variables that are displayed in dialog boxes. Variable sets are defined by selecting Define Variable Sets from the Utilities menu.

- Sets in the source data file that contain variables that do not exist in the active dataset are ignored unless those variables will be created based on specifications in step 2 (Selecting Source and Target Variables) in the Copy Data Properties Wizard.

- Replace deletes any existing variable sets in the active dataset, replacing them with variable sets from the source data file.
- Merge adds variable sets from the source data file to the collection of variable sets in the active dataset. If a set with the same name exists in both files, the existing set in the active dataset is unchanged.

Documents. Notes appended to the data file via the `DOCUMENT` command.

- Replace deletes any existing documents in the active dataset, replacing them with the documents from the source data file.
- Merge combines documents from the source and active dataset. Unique documents in the source file that do not exist in the active dataset are added to the active dataset. All documents are then sorted by date.

Custom Attributes. Custom data file attributes, typically created with the `DATAFILE ATTRIBUTE` command in command syntax.

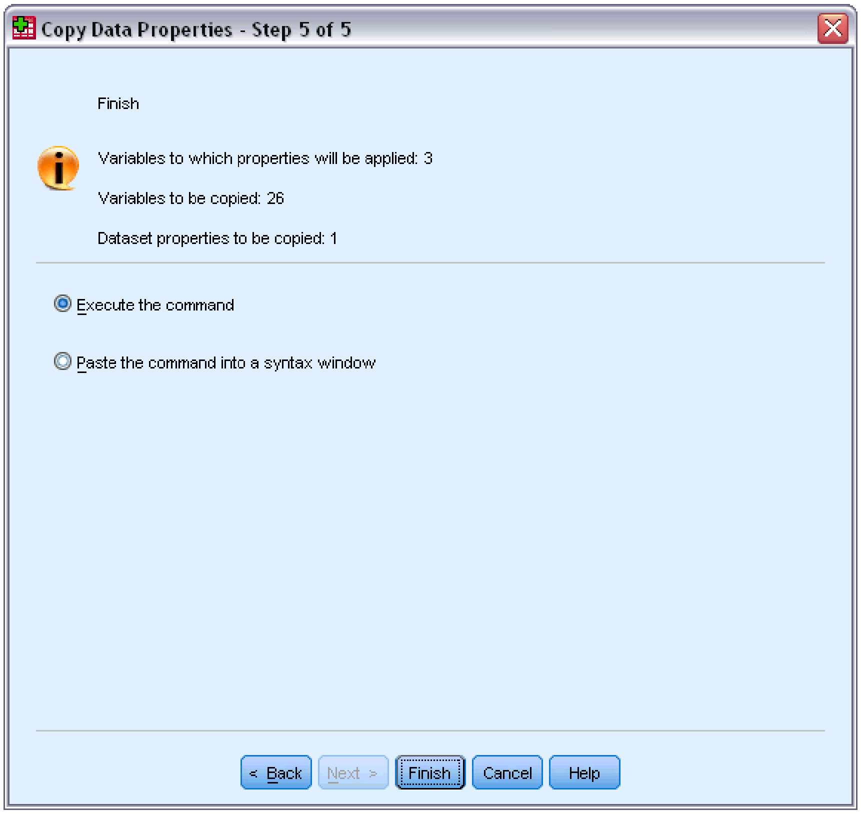
- Replace deletes any existing custom data file attributes in the active dataset, replacing them with the data file attributes from the source data file.
- Merge combines data file attributes from the source and active dataset. Unique attribute names in the source file that do not exist in the active dataset are added to the active dataset. If the same attribute name exists in both data files, the named attribute in the active dataset is unchanged.

Weight Specification. Weights cases by the current weight variable in the source data file if there is a matching variable in the active dataset. This overrides any weighting currently in effect in the active dataset.

File Label. Descriptive label applied to a data file with the `FILE LABEL` command.

Results

Figure 7-12
Copy Data Properties Wizard: Step 5



Copy Data Properties - Step 5 of 5

Finish

Variables to which properties will be applied: 3

Variables to be copied: 26

Dataset properties to be copied: 1

● Execute the command

○ Paste the command into a syntax window

The last step in the Copy Data Properties Wizard provides information on the number of variables for which variable properties will be copied from the source data file, the number of new variables that will be created, and the number of dataset (file) properties that will be copied.

You can also choose to paste the generated command syntax into a syntax window and save the syntax for later use.

Identifying Duplicate Cases

"Duplicate" cases may occur in your data for many reasons, including:

- Data entry errors in which the same case is accidentally entered more than once.
- Multiple cases share a common primary ID value but have different secondary ID values, such as family members who all live in the same house.
- Multiple cases represent the same case but with different values for variables other than those that identify the case, such as multiple purchases made by the same person or company for different products or at different times.

Identify Duplicate Cases allows you to define *duplicate* almost any way that you want and provides some control over the automatic determination of primary versus duplicate cases.

To Identify and Flag Duplicate Cases

- From the menus choose:

Data
  Identify Duplicate Cases...

- Select one or more variables that identify matching cases.

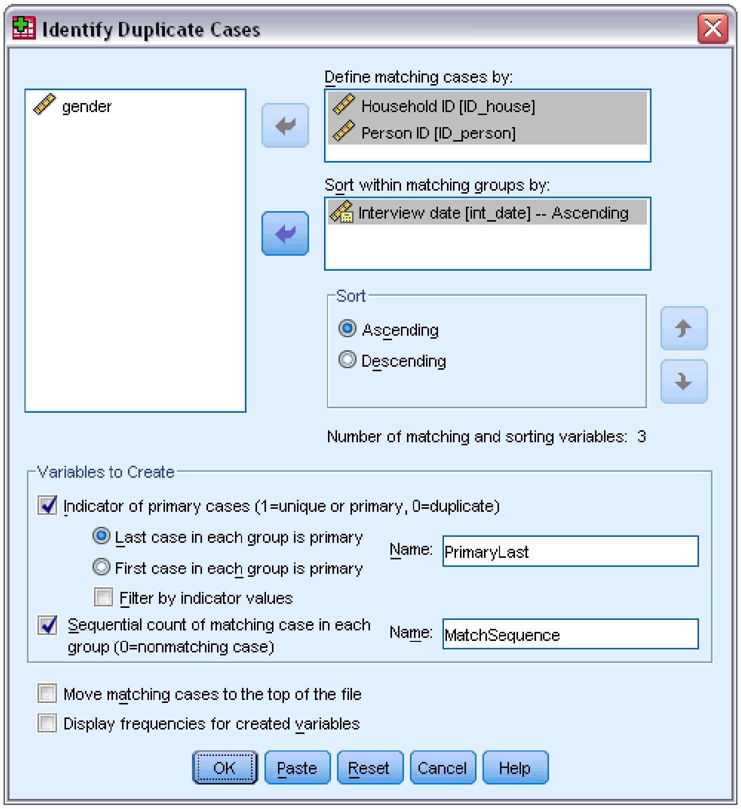
- Select one or more of the options in the Variables to Create group.

Optionally, you can:

- Select one or more variables to sort cases within groups defined by the selected matching cases variables. The sort order defined by these variables determines the “first” and “last” case in each group. Otherwise, the original file order is used.

- Automatically filter duplicate cases so that they won’t be included in reports, charts, or calculations of statistics.

Figure 7-13
Identify Duplicate Cases dialog box



Identify Duplicate Cases

gender

Define matching cases by:
Household ID [ID_house]
Person ID [ID_person]

Sort within matching groups by:
Interview date [int_date] -- Ascending

Sort
● Ascending
○ Descending

Number of matching and sorting variables: 3

Variables to Create
☑ Indicator of primary cases (1=unique or primary, 0=duplicate)
● Last case in each group is primary
○ First case in each group is primary
□ Filter by indicator values
Name: PrimaryLast
☑ Sequential count of matching case in each group (0=nonmatching case)
Name: MatchSequence

□ Move matching cases to the top of the file
□ Display frequencies for created variables

Define matching cases by. Cases are considered duplicates if their values match for *all* selected variables. If you want to identify only cases that are a 100% match in all respects, select all of the variables.

Sort within matching groups by. Cases are automatically sorted by the variables that define matching cases. You can select additional sorting variables that will determine the sequential order of cases in each matching group.

- For each sort variable, you can sort in ascending or descending order.
- If you select multiple sort variables, cases are sorted by each variable within categories of the preceding variable in the list. For example, if you select *date* as the first sorting variable and *amount* as the second sorting variable, cases will be sorted by amount within each date.
- Use the up and down arrow buttons to the right of the list to change the sort order of the variables.
- The sort order determines the "first" and "last" case within each matching group, which determines the value of the optional primary indicator variable. For example, if you want to filter out all but the most recent case in each matching group, you could sort cases within the group in ascending order of a date variable, which would make the most recent date the last date in the group.

Indicator of primary cases. Creates a variable with a value of 1 for all unique cases and the case identified as the primary case in each group of matching cases and a value of 0 for the nonprimary duplicates in each group.

- The primary case can be either the last or first case in each matching group, as determined by the sort order within the matching group. If you don't specify any sort variables, the original file order determines the order of cases within each group.
- You can use the indicator variable as a **filter variable** to exclude nonprimary duplicates from reports and analyses without deleting those cases from the data file.

Sequential count of matching cases in each group. Creates a variable with a sequential value from 1 to *n* for cases in each matching group. The sequence is based on the current order of cases in each group, which is either the original file order or the order determined by any specified sort variables.

Move matching cases to the top. Sorts the data file so that all groups of matching cases are at the top of the data file, making it easy to visually inspect the matching cases in the Data Editor.

Display frequencies for created variables. Frequency tables containing counts for each value of the created variables. For example, for the primary indicator variable, the table would show the number of cases with a value 0 for that variable, which indicates the number of duplicates, and the number of cases with a value of 1 for that variable, which indicates the number of unique and primary cases.

Missing Values. For numeric variables, the system-missing value is treated like any other value—cases with the system-missing value for an identifier variable are treated as having matching values for that variable. For string variables, cases with no value for an identifier variable are treated as having matching values for that variable.

Visual Binning

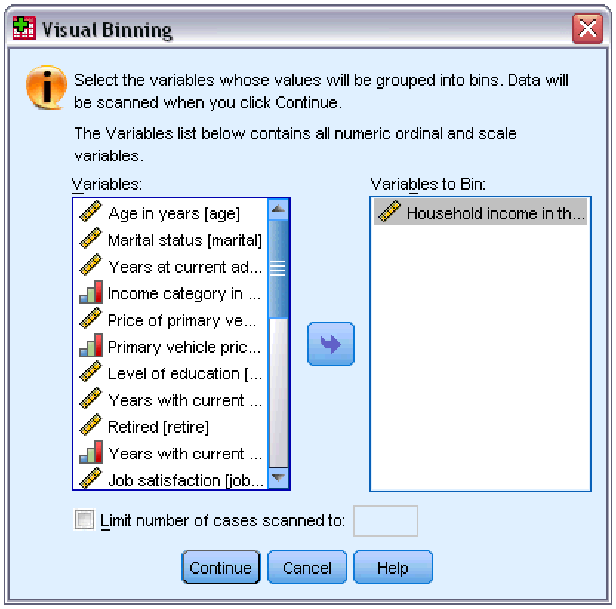
Visual Binning is designed to assist you in the process of creating new variables based on grouping contiguous values of existing variables into a limited number of distinct categories. You can use Visual Binning to:

- Create categorical variables from continuous scale variables. For example, you could use a scale income variable to create a new categorical variable that contains income ranges.
- Collapse a large number of ordinal categories into a smaller set of categories. For example, you could collapse a rating scale of nine down to three categories representing low, medium, and high.

In the first step, you:

▶ Select the numeric scale and/or ordinal variables for which you want to create new categorical (binned) variables.

Figure 7-14
Initial dialog box for selecting variables to bin


Select the variables whose values will be grouped into bins. Data will be scanned when you click Continue.

The Variables list below contains all numeric ordinal and scale variables.

Optionally, you can limit the number of cases to scan. For data files with a large number of cases, limiting the number of cases scanned can save time, but you should avoid this if possible because it will affect the distribution of values used in subsequent calculations in Visual Binning.

Note: String variables and nominal numeric variables are not displayed in the source variable list. Visual Binning requires numeric variables, measured on either a scale or ordinal level, since it assumes that the data values represent some logical order that can be used to group values in a meaningful fashion. You can change the defined measurement level of a variable in Variable View in the Data Editor. For more information, see the topic Variable Measurement Level in Chapter 5 on p. 71.

To Bin Variables

- ▶ From the menus in the Data Editor window choose:

  Transform
    Visual Binning...

- ▶ Select the numeric scale and/or ordinal variables for which you want to create new categorical (binned) variables.

- ▶ Select a variable in the Scanned Variable List.

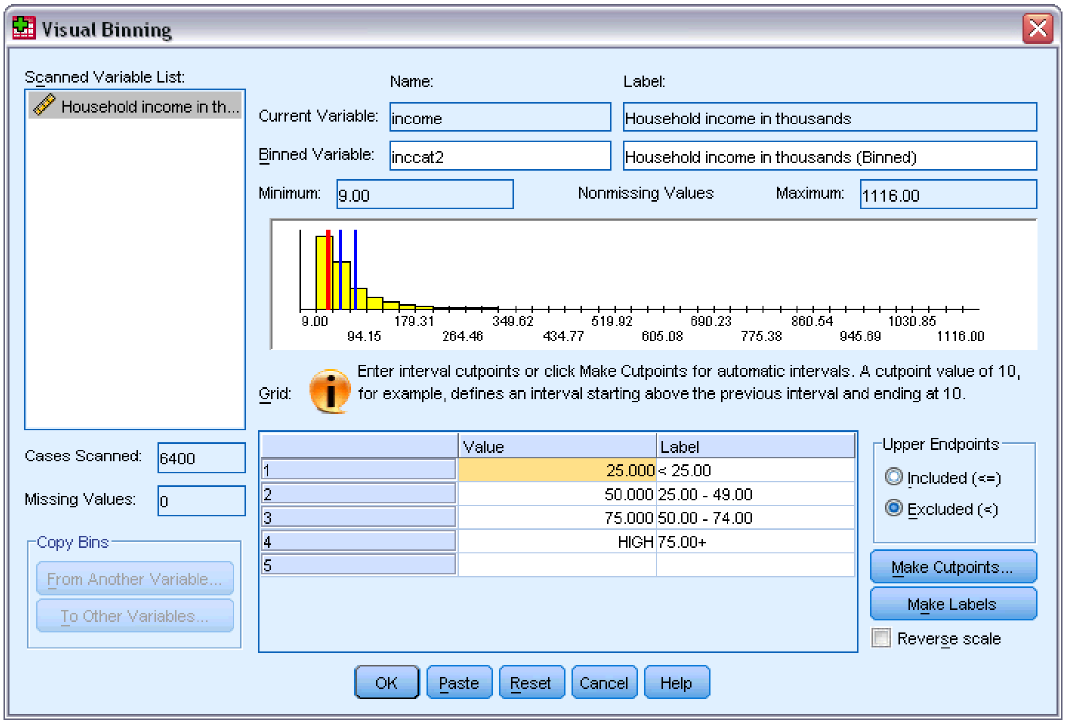
- ▶ Enter a name for the new binned variable. Variable names must be unique and must follow variable naming rules. For more information, see the topic Variable Names in Chapter 5 on p. 70.

- ▶ Define the binning criteria for the new variable. For more information, see the topic Binning Variables on p. 117.

- ▶ Click OK.

Binning Variables

Figure 7-15
Visual Binning, main dialog box



	Name:	Label:
Current Variable:	income	Household income in thousands
Binned Variable:	inccat2	Household income in thousands (Binned)

Minimum: 9.00 Nonmissing Values Maximum: 1116.00

Enter interval cutpoints or click Make Cutpoints for automatic intervals. A cutpoint value of 10, for example, defines an interval starting above the previous interval and ending at 10.

Cases Scanned: 6400

Missing Values: 0

	Value	Label
1	25.000	< 25.00
2	50.000	25.00 - 49.00
3	75.000	50.00 - 74.00
4	HIGH	75.00+
5		

Upper Endpoints
○ Included (<=)
● Excluded (<)

□ Reverse scale

The Visual Binning main dialog box provides the following information for the scanned variables:

Scanned Variable List. Displays the variables you selected in the initial dialog box. You can sort the list by measurement level (scale or ordinal) or by variable label or name by clicking on the column headings.

Cases Scanned. Indicates the number of cases scanned. All scanned cases without user-missing or system-missing values for the selected variable are used to generate the distribution of values used in calculations in Visual Binning, including the histogram displayed in the main dialog box and cutpoints based on percentiles or standard deviation units.

Missing Values. Indicates the number of scanned cases with user-missing or system-missing values. Missing values are not included in any of the binned categories. For more information, see the topic User-Missing Values in Visual Binning on p. 122.

Current Variable. The name and variable label (if any) for the currently selected variable that will be used as the basis for the new, binned variable.

Binned Variable. Name and optional variable label for the new, binned variable.

- **Name.** You must enter a name for the new variable. Variable names must be unique and must follow variable naming rules. For more information, see the topic Variable Names in Chapter 5 on p. 70.
- **Label.** You can enter a descriptive variable label up to 255 characters long. The default variable label is the variable label (if any) or variable name of the source variable with *(Binned)* appended to the end of the label.

Minimum and Maximum. Minimum and maximum values for the currently selected variable, based on the scanned cases and not including values defined as user-missing.

Nonmissing Values. The histogram displays the distribution of nonmissing values for the currently selected variable, based on the scanned cases.

- After you define bins for the new variable, vertical lines on the histogram are displayed to indicate the cutpoints that define bins.
- You can click and drag the cutpoint lines to different locations on the histogram, changing the bin ranges.
- You can remove bins by dragging cutpoint lines off the histogram.

Note: The histogram (displaying nonmissing values), the minimum, and the maximum are based on the scanned values. If you do not include all cases in the scan, the true distribution may not be accurately reflected, particularly if the data file has been sorted by the selected variable. If you scan zero cases, no information about the distribution of values is available.

Grid. Displays the values that define the upper endpoints of each bin and optional value labels for each bin.

- **Value.** The values that define the upper endpoints of each bin. You can enter values or use Make Cutpoints to automatically create bins based on selected criteria. By default, a cutpoint with a value of *HIGH* is automatically included. This bin will contain any nonmissing values above the other cutpoints. The bin defined by the lowest cutpoint will include all nonmissing values lower than or equal to that value (or simply lower than that value, depending on how you define upper endpoints).
- **Label.** Optional, descriptive labels for the values of the new, binned variable. Since the values of the new variable will simply be sequential integers from 1 to *n*, labels that describe what the values represent can be very useful. You can enter labels or use Make Labels to automatically create value labels.

To Delete a Bin from the Grid

- Right-click on the either the *Value* or *Label* cell for the bin.
- From the pop-up context menu, select Delete Row.

Note: If you delete the *HIGH* bin, any cases with values higher than the last specified cutpoint value will be assigned the system-missing value for the new variable.

To Delete All Labels or Delete All Defined Bins

- Right-click anywhere in the grid.
- From the pop-up context menu select either Delete All Labels or Delete All Cutpoints.

Upper Endpoints. Controls treatment of upper endpoint values entered in the *Value* column of the grid.

- **Included (<=).** Cases with the value specified in the *Value* cell are included in the binned category. For example, if you specify values of 25, 50, and 75, cases with a value of exactly 25 will go in the first bin, since this will include all cases with values less than or equal to 25.
- **Excluded (<).** Cases with the value specified in the *Value* cell are not included in the binned category. Instead, they are included in the next bin. For example, if you specify values of 25, 50, and 75, cases with a value of exactly 25 will go in the second bin rather than the first, since the first bin will contain only cases with values less than 25.

Make Cutpoints. Generates binned categories automatically for equal width intervals, intervals with the same number of cases, or intervals based on standard deviations. This is not available if you scanned zero cases. For more information, see the topic Automatically Generating Binned Categories on p. 119.

Make Labels. Generates descriptive labels for the sequential integer values of the new, binned variable, based on the values in the grid and the specified treatment of upper endpoints (included or excluded).

Reverse scale. By default, values of the new, binned variable are ascending sequential integers from 1 to *n*. Reversing the scale makes the values descending sequential integers from *n* to 1.

Copy Bins. You can copy the binning specifications from another variable to the currently selected variable or from the selected variable to multiple other variables. For more information, see the topic Copying Binned Categories on p. 121.

Automatically Generating Binned Categories

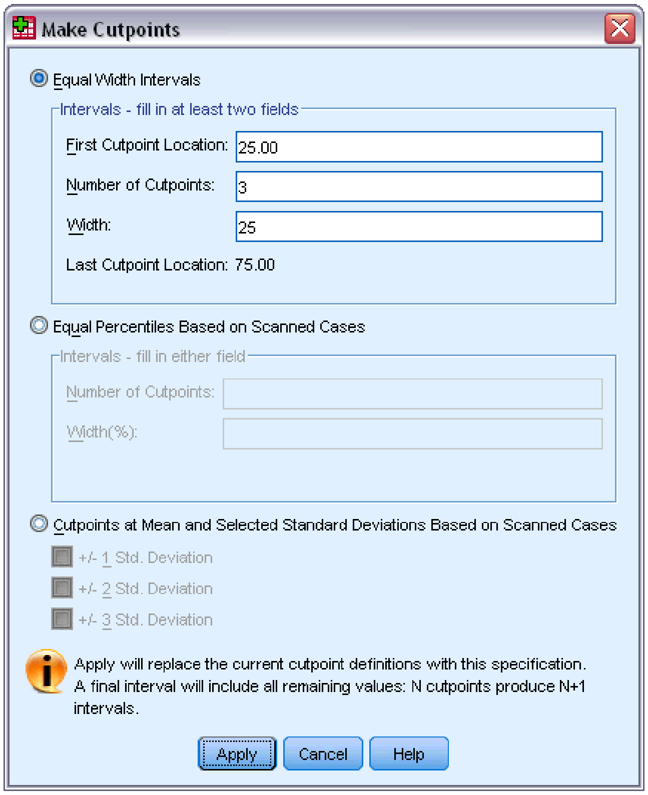
The Make Cutpoints dialog box allows you to auto-generate binned categories based on selected criteria.

To Use the Make Cutpoints Dialog Box

- Select (click) a variable in the Scanned Variable List.
- Click Make Cutpoints.

- ▶ Select the criteria for generating cutpoints that will define the binned categories.
- ▶ Click Apply.

Figure 7-16
Make Cutpoints dialog box



Make Cutpoints

● Equal Width Intervals

Intervals - fill in at least two fields

First Cutpoint Location: 25.00

Number of Cutpoints: 3

Width: 25

Last Cutpoint Location: 75.00

○ Equal Percentiles Based on Scanned Cases

Intervals - fill in either field

Number of Cutpoints:

Width(%):

○ Cutpoints at Mean and Selected Standard Deviations Based on Scanned Cases

□ +/- 1 Std. Deviation

□ +/- 2 Std. Deviation

□ +/- 3 Std. Deviation

Apply will replace the current cutpoint definitions with this specification. A final interval will include all remaining values: N cutpoints produce N+1 intervals.

Note: The Make Cutpoints dialog box is not available if you scanned zero cases.

Equal Width Intervals. Generates binned categories of equal width (for example, 1–10, 11–20, and 21–30) based on any two of the following three criteria:

- **First Cutpoint Location.** The value that defines the upper end of the lowest binned category (for example, a value of 10 indicates a range that includes all values up to 10).
- **Number of Cutpoints.** The number of binned categories is the number of cutpoints *plus one*. For example, 9 cutpoints generate 10 binned categories.
- **Width.** The width of each interval. For example, a value of 10 would bin *age in years* into 10-year intervals.

Equal Percentiles Based on Scanned Cases. Generates binned categories with an equal number of cases in each bin (using the **aempirical algorithm** for percentiles), based on either of the following criteria:

- **Number of Cutpoints.** The number of binned categories is the number of cutpoints *plus one*. For example, three cutpoints generate four percentile bins (quartiles), each containing 25% of the cases.
- **Width (%).** Width of each interval, expressed as a percentage of the total number of cases. For example, a value of 33.3 would produce three binned categories (two cutpoints), each containing 33.3% of the cases.

If the source variable contains a relatively small number of distinct values or a large number of cases with the same value, you may get fewer bins than requested. If there are multiple identical values at a cutpoint, they will all go into the same interval; so the actual percentages may not always be exactly equal.

Cutpoints at Mean and Selected Standard Deviations Based on Scanned Cases. Generates binned categories based on the values of the mean and standard deviation of the distribution of the variable.

- If you don't select any of the standard deviation intervals, two binned categories will be created, with the mean as the cutpoint dividing the bins.
- You can select any combination of standard deviation intervals based on one, two, and/or three standard deviations. For example, selecting all three would result in eight binned categories—six bins in one standard deviation intervals and two bins for cases more than three standard deviations above and below the mean.

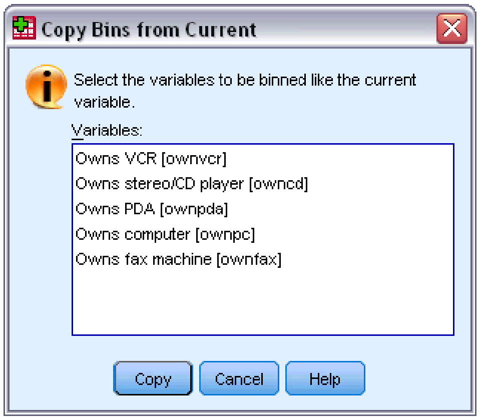
In a normal distribution, 68% of the cases fall within one standard deviation of the mean; 95%, within two standard deviations; and 99%, within three standard deviations. Creating binned categories based on standard deviations may result in some defined bins outside of the actual data range and even outside of the range of possible data values (for example, a negative salary range).

Note: Calculations of percentiles and standard deviations are based on the scanned cases. If you limit the number of cases scanned, the resulting bins may not contain the proportion of cases that you wanted in those bins, particularly if the data file is sorted by the source variable. For example, if you limit the scan to the first 100 cases of a data file with 1000 cases and the data file is sorted in ascending order of age of respondent, instead of four percentile age bins each containing 25% of the cases, you may find that the first three bins each contain only about 3.3% of the cases, and the last bin contains 90% of the cases.

Copying Binned Categories

When creating binned categories for one or more variables, you can copy the binning specifications from another variable to the currently selected variable or from the selected variable to multiple other variables.

Figure 7-17
Copying bins to or from the current variable



Select the variables to be binned like the current variable.

Variables:

Owns VCR [ownvcr]
Owns stereo/CD player [owncd]
Owns PDA [ownpda]
Owns computer [ownpc]
Owns fax machine [ownfax]

To Copy Binning Specifications

- Define binned categories for at least one variable—but do *not* click OK or Paste.
- Select (click) a variable in the Scanned Variable List for which you have defined binned categories.
- Click To Other Variables.
- Select the variables for which you want to create new variables with the same binned categories.
- Click Copy.

or

- Select (click) a variable in the Scanned Variable List to which you want to copy defined binned categories.
- Click From Another Variable.
- Select the variable with the defined binned categories that you want to copy.
- Click Copy.

If you have specified value labels for the variable from which you are copying the binning specifications, those are also copied.

Note: Once you click OK in the Visual Binning main dialog box to create new binned variables (or close the dialog box in any other way), you cannot use Visual Binning to copy those binned categories to other variables.

User-Missing Values in Visual Binning

Values defined as user-missing (values identified as codes for missing data) for the source variable are not included in the binned categories for the new variable. User-missing values for the source variables are copied as user-missing values for the new variable, and any defined value labels for missing value codes are also copied.

If a missing value code conflicts with one of the binned category values for the new variable, the missing value code for the new variable is recoded to a nonconflicting value by adding 100 to the highest binned category value. For example, if a value of 1 is defined as user-missing for the source variable and the new variable will have six binned categories, any cases with a value of 1 for the source variable will have a value of 106 for the new variable, and 106 will be defined as user-missing. If the user-missing value for the source variable had a defined value label, that label will be retained as the value label for the recoded value of the new variable.

Note: If the source variable has a defined range of user-missing values of the form *LO-n*, where *n* is a positive number, the corresponding user-missing values for the new variable will be negative numbers.

Chapter

8

Data Transformations

In an ideal situation, your raw data are perfectly suitable for the type of analysis you want to perform, and any relationships between variables are either conveniently linear or neatly orthogonal. Unfortunately, this is rarely the case. Preliminary analysis may reveal inconvenient coding schemes or coding errors, or data transformations may be required in order to expose the true relationship between variables.

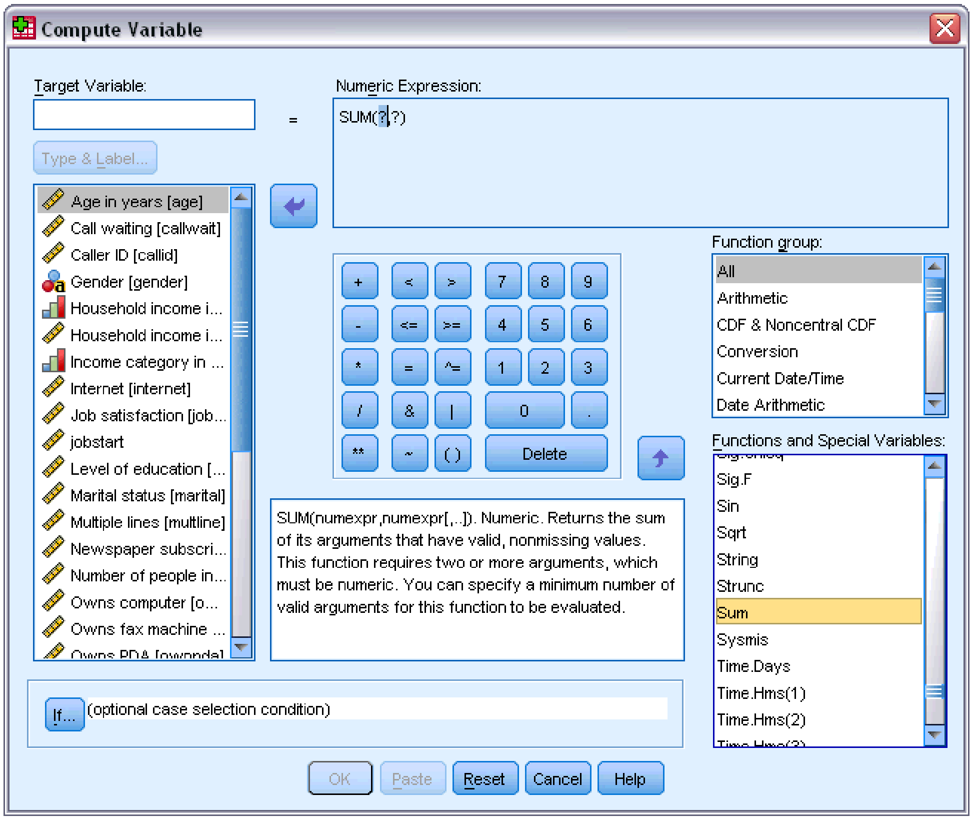
You can perform data transformations ranging from simple tasks, such as collapsing categories for analysis, to more advanced tasks, such as creating new variables based on complex equations and conditional statements.

Computing Variables

Use the Compute dialog box to compute values for a variable based on numeric transformations of other variables.

- You can compute values for numeric or string (alphanumeric) variables.
- You can create new variables or replace the values of existing variables. For new variables, you can also specify the variable type and label.
- You can compute values selectively for subsets of data based on logical conditions.
- You can use a large variety of built-in functions, including arithmetic functions, statistical functions, distribution functions, and string functions.

Figure 8-1
Compute Variable dialog box



Numeric Expression: SUM(?,?)

SUM(numexpr,numexpr[,..]). Numeric. Returns the sum of its arguments that have valid, nonmissing values. This function requires two or more arguments, which must be numeric. You can specify a minimum number of valid arguments for this function to be evaluated.

To Compute Variables

▶ From the menus choose:

Transform
  Compute Variable...

▶ Type the name of a single target variable. It can be an existing variable or a new variable to be added to the active dataset.

▶ To build an expression, either paste components into the Expression field or type directly in the Expression field.

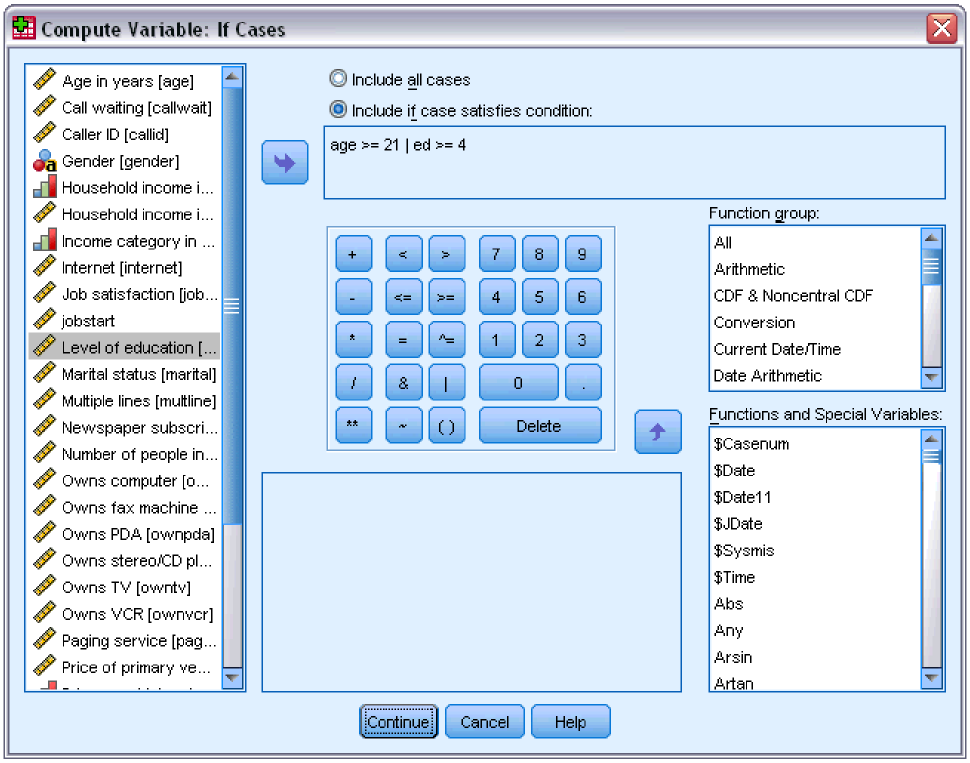
- You can paste functions or commonly used system variables by selecting a group from the Function group list and double-clicking the function or variable in the Functions and Special Variables list (or select the function or variable and click the arrow adjacent to the Function group list). Fill in any parameters indicated by question marks (only applies to functions). The function group labeled All provides a listing of all available functions and system variables. A brief description of the currently selected function or variable is displayed in a reserved area in the dialog box.
- String constants must be enclosed in quotation marks or apostrophes.

- If values contain decimals, a period (.) must be used as the decimal indicator.
- For new string variables, you must also select Type & Label to specify the data type.

Compute Variable: If Cases

The If Cases dialog box allows you to apply data transformations to selected subsets of cases, using conditional expressions. A conditional expression returns a value of *true*, *false*, or *missing* for each case.

Figure 8-2
Compute Variable If Cases dialog box



Compute Variable: If Cases

○ Include all cases

● Include if case satisfies condition:

```
age >= 21 | ed >= 4
```

- If the result of a conditional expression is *true*, the case is included in the selected subset.
- If the result of a conditional expression is *false* or *missing*, the case is not included in the selected subset.
- Most conditional expressions use one or more of the six relational operators (<, >, <=, >=, =, and ~=) on the calculator pad.
- Conditional expressions can include variable names, constants, arithmetic operators, numeric (and other) functions, logical variables, and relational operators.

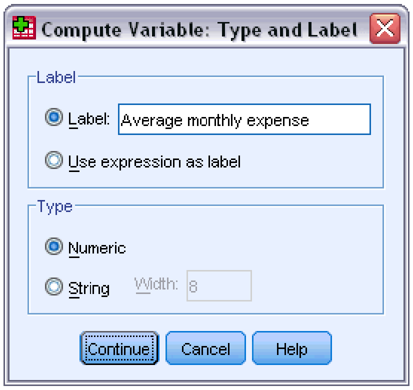
Compute Variable: Type and Label

By default, new computed variables are numeric. To compute a new string variable, you must specify the data type and width.

Label. Optional, descriptive variable label up to 255 bytes long. You can enter a label or use the first 110 characters of the compute expression as the label.

Type. Computed variables can be numeric or string (alphanumeric). String variables cannot be used in calculations.

Figure 8-3
Type and Label dialog box



Compute Variable: Type and Label

Label

● Label: Average monthly expense

○ Use expression as label

Type

● Numeric

○ String Width: 8

Functions

Many types of functions are supported, including:

- Arithmetic functions
- Statistical functions
- String functions
- Date and time functions
- Distribution functions
- Random variable functions
- Missing value functions
- Scoring functions (PASW Statistics Server only)

For more information and a detailed description of each function, type functions on the Index tab of the Help system.

■

Missing Values in Functions

Functions and simple arithmetic expressions treat missing values in different ways. In the expression:

```
(var1+var2+var3)/3
```

the result is missing if a case has a missing value for any of the three variables.

In the expression:

```
MEAN(var1, var2, var3)
```

the result is missing only if the case has missing values for all three variables.

For statistical functions, you can specify the minimum number of arguments that must have nonmissing values. To do so, type a period and the minimum number after the function name, as in:

```
MEAN.2(var1, var2, var3)
```

Random Number Generators

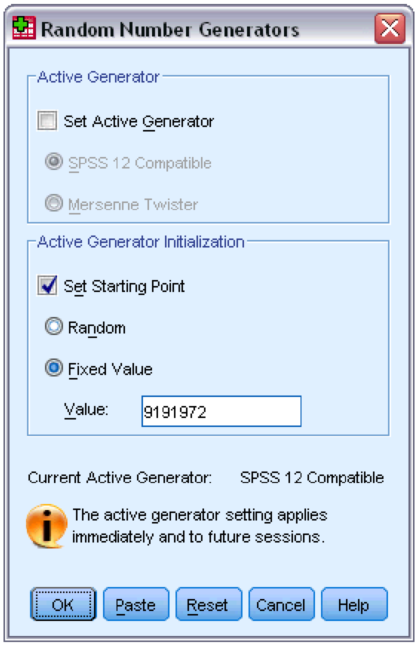
The Random Number Generators dialog box allows you to select the random number generator and set the starting sequence value so you can reproduce a sequence of random numbers.

Active Generator. Two different random number generators are available:

- **Version 12 Compatible.** The random number generator used in version 12 and previous releases. If you need to reproduce randomized results generated in previous releases based on a specified seed value, use this random number generator.
- **Mersenne Twister.** A newer random number generator that is more reliable for simulation purposes. If reproducing randomized results from version 12 or earlier is not an issue, use this random number generator.

Active Generator Initialization. The random number seed changes each time a random number is generated for use in transformations (such as random distribution functions), random sampling, or case weighting. To replicate a sequence of random numbers, set the initialization starting point value prior to each analysis that uses the random numbers. The value must be a positive integer.

Figure 8-4
Random Number Generators dialog box



Random Number Generators

Active Generator

☐ Set Active Generator

● SPSS 12 Compatible

○ Mersenne Twister

Active Generator Initialization

☑ Set Starting Point

○ Random

● Fixed Value

Value: 9191972

Current Active Generator: SPSS 12 Compatible

The active generator setting applies immediately and to future sessions.

To select the random number generator and/or set the initialization value:

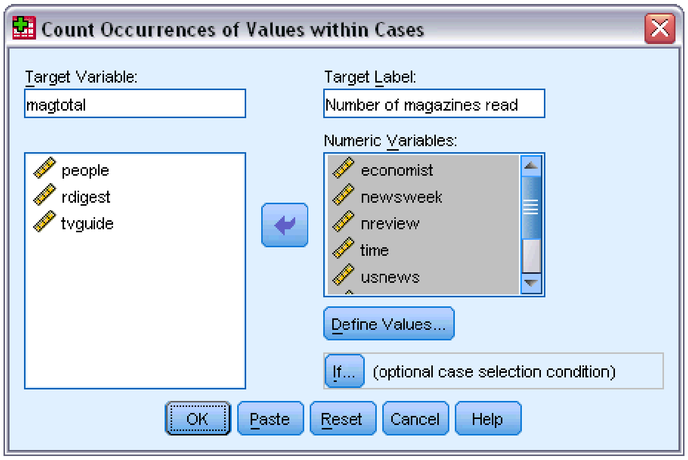
▶ From the menus choose:

Transform
  Random Number Generators

Count Occurrences of Values within Cases

This dialog box creates a variable that counts the occurrences of the same value(s) in a list of variables for each case. For example, a survey might contain a list of magazines with *yes*/*no* check boxes to indicate which magazines each respondent reads. You could count the number of *yes* responses for each respondent to create a new variable that contains the total number of magazines read.

Figure 8-5
Count Occurrences of Values within Cases dialog box



Count Occurrences of Values within Cases

Target Variable: magtotal

Target Label: Number of magazines read

people
rdigest
tvguide

Numeric Variables:
economist
newsweek
nreview
time
usnews

To Count Occurrences of Values within Cases

- From the menus choose:

  Transform
  Count Values within Cases...

- Enter a target variable name.

- Select two or more variables of the same type (numeric or string).

- Click Define Values and specify which value or values should be counted.

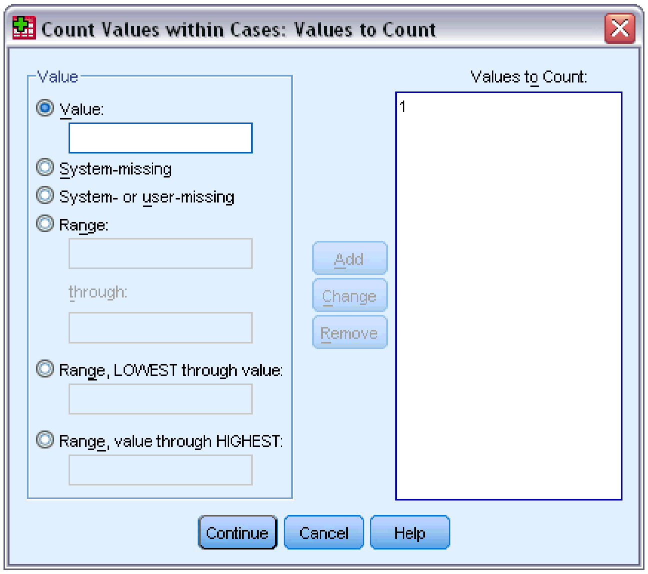
  Optionally, you can define a subset of cases for which to count occurrences of values.

Count Values within Cases: Values to Count

The value of the target variable (on the main dialog box) is incremented by 1 each time one of the selected variables matches a specification in the Values to Count list here. If a case matches several specifications for any variable, the target variable is incremented several times for that variable.

Value specifications can include individual values, missing or system-missing values, and ranges. Ranges include their endpoints and any user-missing values that fall within the range.

Figure 8-6
Values to Count dialog box



Count Values within Cases: Values to Count

Value

● Value:

○ System-missing

○ System- or user-missing

○ Range:

through:

○ Range, LOWEST through value:

○ Range, value through HIGHEST:

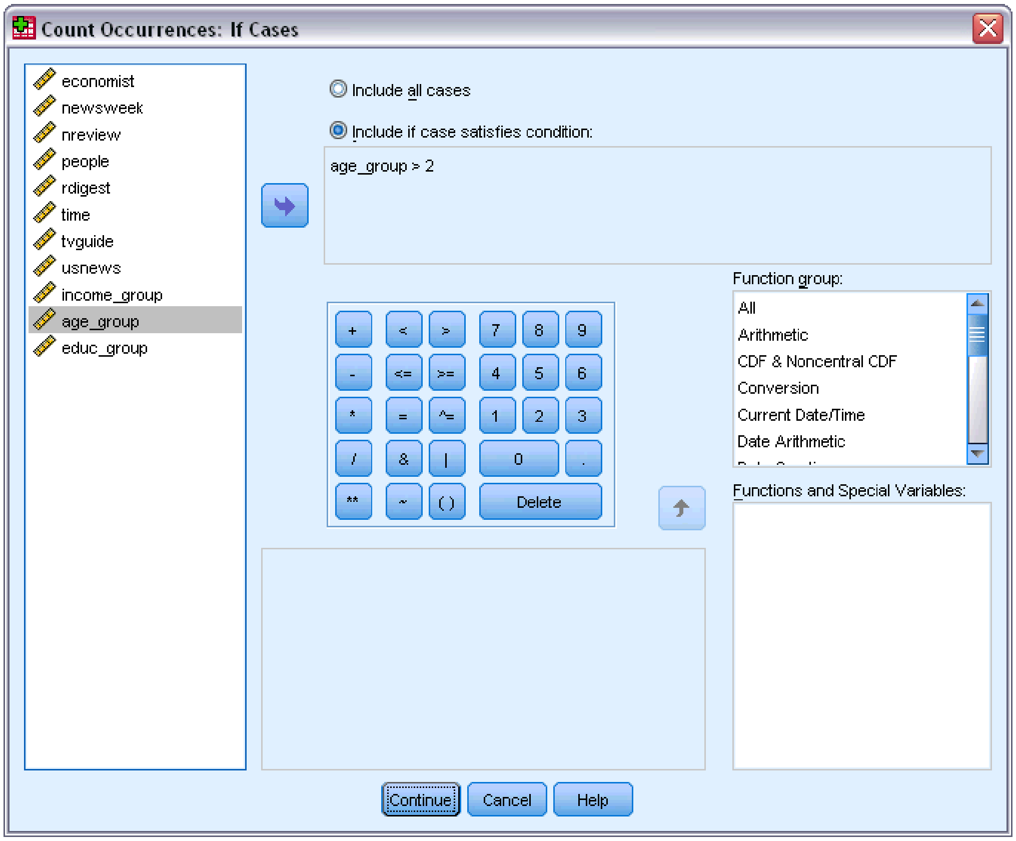
Values to Count:

1

Count Occurrences: If Cases

The If Cases dialog box allows you to count occurrences of values for a selected subset of cases, using conditional expressions. A conditional expression returns a value of *true*, *false*, or *missing* for each case.

Figure 8-7
Count Occurrences If Cases dialog box



○ Include all cases

● Include if case satisfies condition:

age_group > 2

For general considerations on using the If Cases dialog box, see Compute Variable: If Cases on p. 126.

Shift Values

Shift Values creates new variables that contain the values of existing variables from preceding or subsequent cases.

Name. Name for the new variable. This must be a name that does not already exist in the active dataset.

Get value from earlier case (lag). Get the value from a previous case in the active dataset. For example, with the default number of cases value of 1, each case for the new variable has the value of the original variable from the case that immediately precedes it.

Get value from following case (lead). Get the value from a subsequent case in the active dataset. For example, with the default number of cases value of 1, each case for the new variable has the value of the original variable from the next case.

Number of cases to shift. Get the value from the *n*th preceding or subsequent case, where *n* is the value specified. The value must be a non-negative integer.

- If split file processing is on, the scope of the shift is limited to each split group. A shift value cannot be obtained from a case in a preceding or subsequent split group.
- Filter status is ignored.
- The value of the result variable is set to system-missing for the first or last *n* cases in the dataset or split group, where *n* is the value specified for **Number of cases to shift**. For example, using the Lag method with a value of 1 would set the result variable to system-missing for the first case in the dataset (or first case in each split group).
- User-missing values are preserved.
- Dictionary information from the original variable, including defined value labels and user-missing value assignments, is applied to the new variable. (Note: Custom variable attributes are not included.)
- A variable label is automatically generated for the new variable that describes the shift operation that created the variable.

To Create a New Variable with Shifted Values

▶ From the menus, choose:

Transform
  Shift Values

▶ Select the variable to use as the source of values for the new variable.

▶ Enter a name for the new variable.

▶ Select the shift method (lag or lead) and the number of cases to shift.

▶ Click Change.

▶ Repeat for each new variable you want to create.

Recoding Values

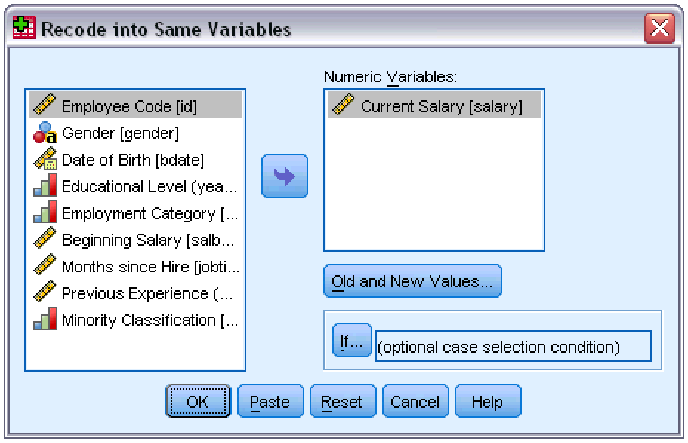
You can modify data values by recoding them. This is particularly useful for collapsing or combining categories. You can recode the values within existing variables, or you can create new variables based on the recoded values of existing variables.

Recode into Same Variables

The Recode into Same Variables dialog box allows you to reassign the values of existing variables or collapse ranges of existing values into new values. For example, you could collapse salaries into salary range categories.

You can recode numeric and string variables. If you select multiple variables, they must all be the same type. You cannot recode numeric and string variables together.

Figure 8-8
Recode into Same Variables dialog box



To Recode Values of a Variable

▶ From the menus choose:

Transform
  Recode into Same Variables...

▶ Select the variables you want to recode. If you select multiple variables, they must be the same type (numeric or string).

▶ Click Old and New Values and specify how to recode values.

Optionally, you can define a subset of cases to recode. The If Cases dialog box for doing this is the same as the one described for Count Occurrences.

Recode into Same Variables: Old and New Values

You can define values to recode in this dialog box. All value specifications must be the same data type (numeric or string) as the variables selected in the main dialog box.

Old Value. The value(s) to be recoded. You can recode single values, ranges of values, and missing values. System-missing values and ranges cannot be selected for string variables because neither concept applies to string variables. Ranges include their endpoints and any user-missing values that fall within the range.

- **Value.** Individual old value to be recoded into a new value. The value must be the same data type (numeric or string) as the variable(s) being recoded.
- **System-missing.** Values assigned by the program when values in your data are undefined according to the format type you have specified, when a numeric field is blank, or when a value resulting from a transformation command is undefined. Numeric system-missing values are displayed as periods. String variables cannot have system-missing values, since any character is legal in a string variable.

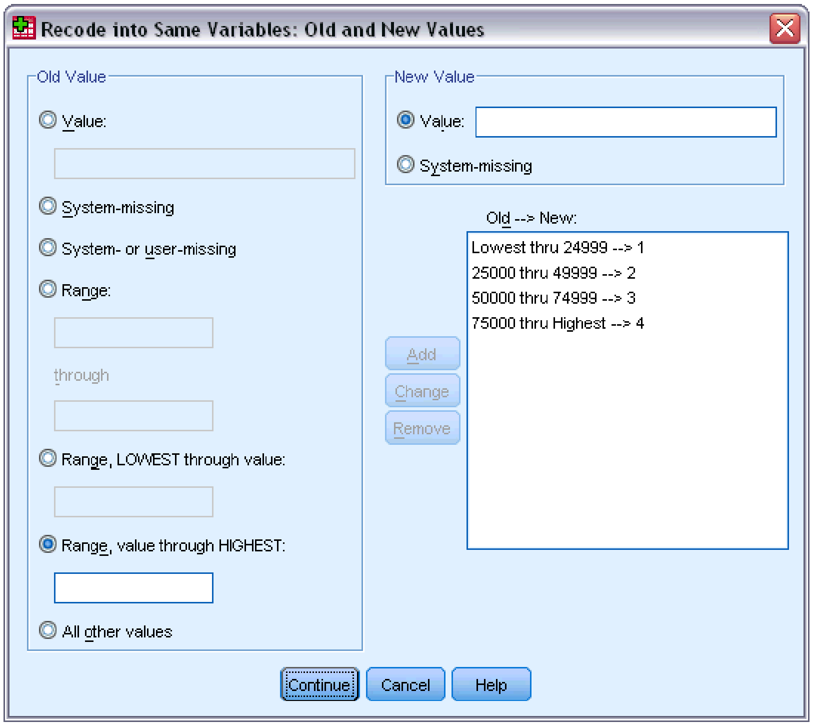
- **System- or user-missing.** Observations with values that either have been defined as user-missing values or are unknown and have been assigned the system-missing value, which is indicated with a period (.).
- **Range.** Inclusive range of values. Not available for string variables. Any user-missing values within the range are included.
- **All other values.** Any remaining values not included in one of the specifications on the Old-New list. This appears as ELSE on the Old-New list.

New Value. The single value into which each old value or range of values is recoded. You can enter a value or assign the system-missing value.

- **Value.** Value into which one or more old values will be recoded. The value must be the same data type (numeric or string) as the old value.
- **System-missing.** Recodes specified old values into the system-missing value. The system-missing value is not used in calculations, and cases with the system-missing value are excluded from many procedures. Not available for string variables.

Old–>New. The list of specifications that will be used to recode the variable(s). You can add, change, and remove specifications from the list. The list is automatically sorted, based on the old value specification, using the following order: single values, missing values, ranges, and all other values. If you change a recode specification on the list, the procedure automatically re-sorts the list, if necessary, to maintain this order.

Figure 8-9
Old and New Values dialog box



Recode into Same Variables: Old and New Values

Old Value

○ Value:

○ System-missing

○ System- or user-missing

○ Range:

through

○ Range, LOWEST through value:

● Range, value through HIGHEST:

○ All other values

New Value

● Value:

○ System-missing

Old --> New:

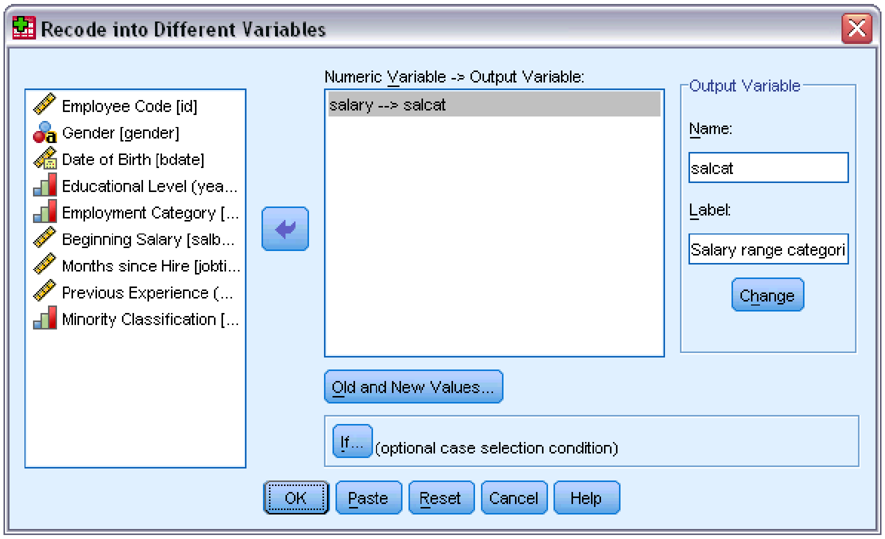
Lowest thru 24999 --> 1
25000 thru 49999 --> 2
50000 thru 74999 --> 3
75000 thru Highest --> 4

Recode into Different Variables

The Recode into Different Variables dialog box allows you to reassign the values of existing variables or collapse ranges of existing values into new values for a new variable. For example, you could collapse salaries into a new variable containing salary-range categories.

- You can recode numeric and string variables.
- You can recode numeric variables into string variables and vice versa.
- If you select multiple variables, they must all be the same type. You cannot recode numeric and string variables together.

Figure 8-10
Recode into Different Variables dialog box



Numeric Variable -> Output Variable: salary --> salcat

Output Variable

Name: salcat

Label: Salary range categori

Recode into Different Variables
Employee Code [id]
Gender [gender]
Date of Birth [bdate]
Educational Level (yea...
Employment Category [...
Beginning Salary [salb...
Months since Hire [jobti...
Previous Experience (...
Minority Classification [...
Change
Old and New Values...
If...
(optional case selection condition)
OK
Paste
Reset
Cancel
Help

To Recode Values of a Variable into a New Variable

- ▶ From the menus choose:

Transform
  Recode into Different Variables...

- ▶ Select the variables you want to recode. If you select multiple variables, they must be the same type (numeric or string).
- ▶ Enter an output (new) variable name for each new variable and click Change.
- ▶ Click Old and New Values and specify how to recode values.

Optionally, you can define a subset of cases to recode. The If Cases dialog box for doing this is the same as the one described for Count Occurrences.

Recode into Different Variables: Old and New Values

You can define values to recode in this dialog box.

Old Value. The value(s) to be recoded. You can recode single values, ranges of values, and missing values. System-missing values and ranges cannot be selected for string variables because neither concept applies to string variables. Old values must be the same data type (numeric or string) as the original variable. Ranges include their endpoints and any user-missing values that fall within the range.

- **Value.** Individual old value to be recoded into a new value. The value must be the same data type (numeric or string) as the variable(s) being recoded.
- **System-missing.** Values assigned by the program when values in your data are undefined according to the format type you have specified, when a numeric field is blank, or when a value resulting from a transformation command is undefined. Numeric system-missing values are displayed as periods. String variables cannot have system-missing values, since any character is legal in a string variable.
- **System- or user-missing.** Observations with values that either have been defined as user-missing values or are unknown and have been assigned the system-missing value, which is indicated with a period (.).
- **Range.** Inclusive range of values. Not available for string variables. Any user-missing values within the range are included.
- **All other values.** Any remaining values not included in one of the specifications on the Old-New list. This appears as ELSE on the Old-New list.

New Value. The single value into which each old value or range of values is recoded. New values can be numeric or string.

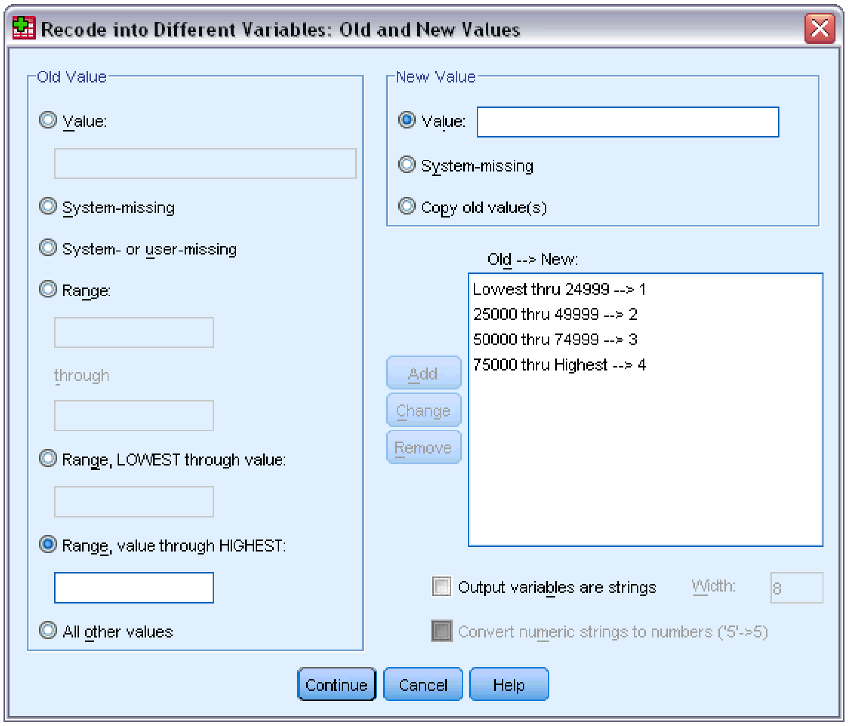
- **Value.** Value into which one or more old values will be recoded. The value must be the same data type (numeric or string) as the old value.
- **System-missing.** Recodes specified old values into the system-missing value. The system-missing value is not used in calculations, and cases with the system-missing value are excluded from many procedures. Not available for string variables.
- **Copy old values.** Retains the old value. If some values don't require recoding, use this to include the old values. Any old values that are not specified are not included in the new variable, and cases with those values will be assigned the system-missing value for the new variable.

Output variables are strings. Defines the new, recoded variable as a string (alphanumeric) variable. The old variable can be numeric or string.

Convert numeric strings to numbers. Converts string values containing numbers to numeric values. Strings containing anything other than numbers and an optional sign (+ or -) are assigned the system-missing value.

Old–>New. The list of specifications that will be used to recode the variable(s). You can add, change, and remove specifications from the list. The list is automatically sorted, based on the old value specification, using the following order: single values, missing values, ranges, and all other values. If you change a recode specification on the list, the procedure automatically re-sorts the list, if necessary, to maintain this order.

Figure 8-11
Old and New Values dialog box



Recode into Different Variables: Old and New Values

Old Value

○ Value:

○ System-missing

○ System- or user-missing

○ Range:

through

○ Range, LOWEST through value:

● Range, value through HIGHEST:

○ All other values

New Value

● Value:

○ System-missing

○ Copy old value(s)

Old --> New:

Lowest thru 24999 --> 1
25000 thru 49999 --> 2
50000 thru 74999 --> 3
75000 thru Highest --> 4

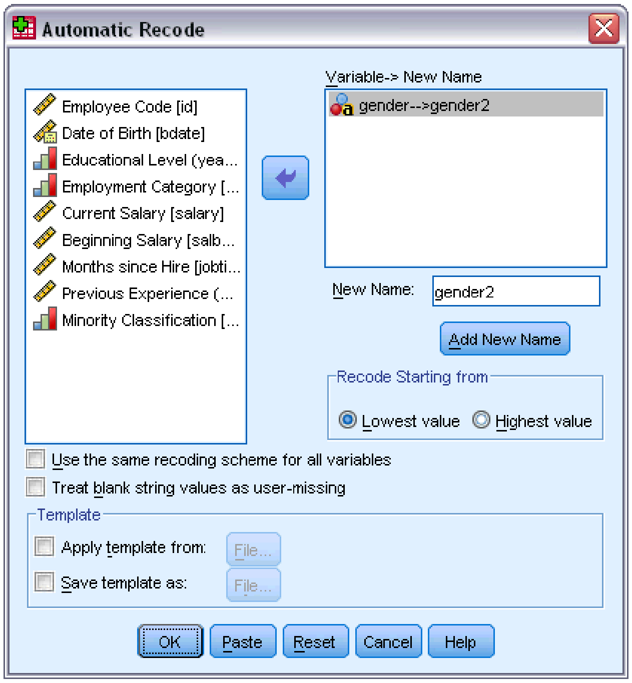
□ Output variables are strings Width: 8

□ Convert numeric strings to numbers ('5'->5)

Automatic Recode

The Automatic Recode dialog box allows you to convert string and numeric values into consecutive integers. When category codes are not sequential, the resulting empty cells reduce performance and increase memory requirements for many procedures. Additionally, some procedures cannot use string variables, and some require consecutive integer values for factor levels.

Figure 8-12
Automatic Recode dialog box



Automatic Recode

Employee Code [id]
Date of Birth [bdate]
Educational Level (yea...
Employment Category [...
Current Salary [salary]
Beginning Salary [salb...
Months since Hire [jobti...
Previous Experience (...
Minority Classification [...

Variable-> New Name
gender-->gender2

New Name: gender2

Recode Starting from
● Lowest value ○ Highest value

□ Use the same recoding scheme for all variables
□ Treat blank string values as user-missing

Template
□ Apply template from:
□ Save template as:

- The new variable(s) created by Automatic Recode retain any defined variable and value labels from the old variable. For any values without a defined value label, the original value is used as the label for the recoded value. A table displays the old and new values and value labels.
- String values are recoded in alphabetical order, with uppercase letters preceding their lowercase counterparts.
- Missing values are recoded into missing values higher than any nonmissing values, with their order preserved. For example, if the original variable has 10 nonmissing values, the lowest missing value would be recoded to 11, and the value 11 would be a missing value for the new variable.

Use the same recoding scheme for all variables. This option allows you to apply a single autorecoding scheme to all the selected variables, yielding a consistent coding scheme for all the new variables.

If you select this option, the following rules and limitations apply:

- All variables must be of the same type (numeric or string).
- All observed values for all selected variables are used to create a sorted order of values to recode into sequential integers.
- User-missing values for the new variables are based on the first variable in the list with defined user-missing values. All other values from other original variables, except for system-missing, are treated as valid.

Treat blank string values as user-missing. For string variables, blank or null values are not treated as **system-missing**. This option will autorecode blank strings into a **user-missing** value higher than the highest nonmissing value.

Templates

You can save the autorecoding scheme in a template file and then apply it to other variables and other data files.

For example, you may have a large number of alphanumeric product codes that you autorecode into integers every month, but some months new product codes are added that change the original autorecoding scheme. If you save the original scheme in a template and then apply it to the new data that contain the new set of codes, any new codes encountered in the data are autorecoded into values higher than the last value in the template, preserving the original autorecode scheme of the original product codes.

Save template as. Saves the autorecode scheme for the selected variables in an external template file.

- The template contains information that maps the original nonmissing values to the recoded values.
- Only information for nonmissing values is saved in the template. User-missing value information is not retained.
- If you have selected multiple variables for recoding but you have not selected to use the same autorecoding scheme for all variables or you are not applying an existing template as part of the autorecoding, the template will be based on the first variable in the list.
- If you have selected multiple variables for recoding and you have also selected Use the same recoding scheme for all variables and/or you have selected Apply template, then the template will contain the combined autorecoding scheme for all variables.

Apply template from. Applies a previously saved autorecode template to variables selected for recoding, appending any additional values found in the variables to the end of the scheme and preserving the relationship between the original and autorecoded values stored in the saved scheme.

- All variables selected for recoding must be the same type (numeric or string), and that type must match the type defined in the template.
- Templates do not contain any information on user-missing values. User-missing values for the target variables are based on the first variable in the original variable list with defined user-missing values. All other values from other original variables, except for system-missing, are treated as valid.
- Value mappings from the template are applied first. All remaining values are recoded into values higher than the last value in the template, with user-missing values (based on the first variable in the list with defined user-missing values) recoded into values higher than the last valid value.
- If you have selected multiple variables for autorecoding, the template is applied first, followed by a common, combined autorecoding for all additional values found in the selected variables, resulting in a single, common autorecoding scheme for all selected variables.

To Recode String or Numeric Values into Consecutive Integers

▶ From the menus choose:

Transform
  Automatic Recode...

▶ Select one or more variables to recode.

▶ For each selected variable, enter a name for the new variable and click New Name.

Rank Cases

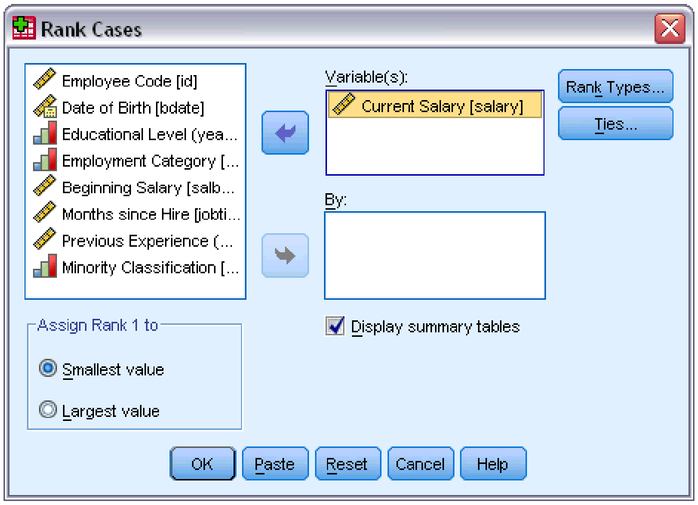
The Rank Cases dialog box allows you to create new variables containing ranks, normal and Savage scores, and percentile values for numeric variables.

New variable names and descriptive variable labels are automatically generated, based on the original variable name and the selected measure(s). A summary table lists the original variables, the new variables, and the variable labels. (Note: The automatically generated new variable names are limited to a maximum length of 8 bytes.)

Optionally, you can:

- Rank cases in ascending or descending order.
- Organize rankings into subgroups by selecting one or more grouping variables for the By list. Ranks are computed within each group. Groups are defined by the combination of values of the grouping variables. For example, if you select *gender* and *minority* as grouping variables, ranks are computed for each combination of *gender* and *minority*.

Figure 8-13
Rank Cases dialog box



Variable(s): Current Salary [salary]

Assign Rank 1 to
● Smallest value
○ Largest value

☑ Display summary tables

To Rank Cases

▶ From the menus choose:

Transform
  Rank Cases...

▶ Select one or more variables to rank. You can rank only numeric variables.

Optionally, you can rank cases in ascending or descending order and organize ranks into subgroups.

Rank Cases: Types

You can select multiple ranking methods. A separate ranking variable is created for each method. Ranking methods include simple ranks, Savage scores, fractional ranks, and percentiles. You can also create rankings based on proportion estimates and normal scores.

Rank. Simple rank. The value of the new variable equals its rank.

Savage score. The new variable contains Savage scores based on an exponential distribution.

Fractional rank. The value of the new variable equals rank divided by the sum of the weights of the nonmissing cases.

Fractional rank as percent. Each rank is divided by the number of cases with valid values and multiplied by 100.

Sum of case weights. The value of the new variable equals the sum of case weights. The new variable is a constant for all cases in the same group.

Ntiles. Ranks are based on percentile groups, with each group containing approximately the same number of cases. For example, 4 Ntiles would assign a rank of 1 to cases below the 25th percentile, 2 to cases between the 25th and 50th percentile, 3 to cases between the 50th and 75th percentile, and 4 to cases above the 75th percentile.

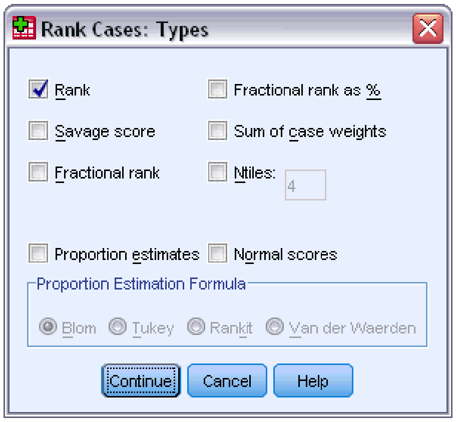
Proportion estimates. Estimates of the cumulative proportion of the distribution corresponding to a particular rank.

Normal scores. The z scores corresponding to the estimated cumulative proportion.

Proportion Estimation Formula. For proportion estimates and normal scores, you can select the proportion estimation formula: Blom, Tukey, Rankit, or Van der Waerden.

- **Blom.** Creates new ranking variable based on proportion estimates that uses the formula (r-3/8) / (w+1/4), where w is the sum of the case weights and r is the rank.
- **Tukey.** Uses the formula (r-1/3) / (w+1/3), where r is the rank and w is the sum of the case weights.
- **Rankit.** Uses the formula (r-1/2) / w, where w is the number of observations and r is the rank, ranging from 1 to w.
- **Van der Waerden.** Van der Waerden's transformation, defined by the formula r/(w+1), where w is the sum of the case weights and r is the rank, ranging from 1 to w.

Figure 8-14
Rank Cases Types dialog box



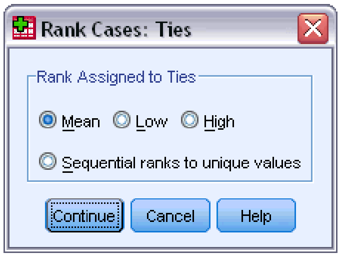
☑ Rank
□ Savage score
□ Fractional rank
□ Fractional rank as %
□ Sum of case weights
□ Ntiles: 4
□ Proportion estimates
□ Normal scores

Proportion Estimation Formula
● Blom ○ Tukey ○ Rankit ○ Van der Waerden

Rank Cases: Ties

This dialog box controls the method for assigning rankings to cases with the same value on the original variable.

Figure 8-15
Rank Cases Ties dialog box



Rank Assigned to Ties
● Mean ○ Low ○ High
○ Sequential ranks to unique values

The following table shows how the different methods assign ranks to tied values:

Value	Mean	Low	High	Sequential
10	1	1	1	1
15	3	2	4	2
15	3	2	4	2
15	3	2	4	2
16	5	5	5	3
20	6	6	6	4

Date and Time Wizard

The Date and Time Wizard simplifies a number of common tasks associated with date and time variables.

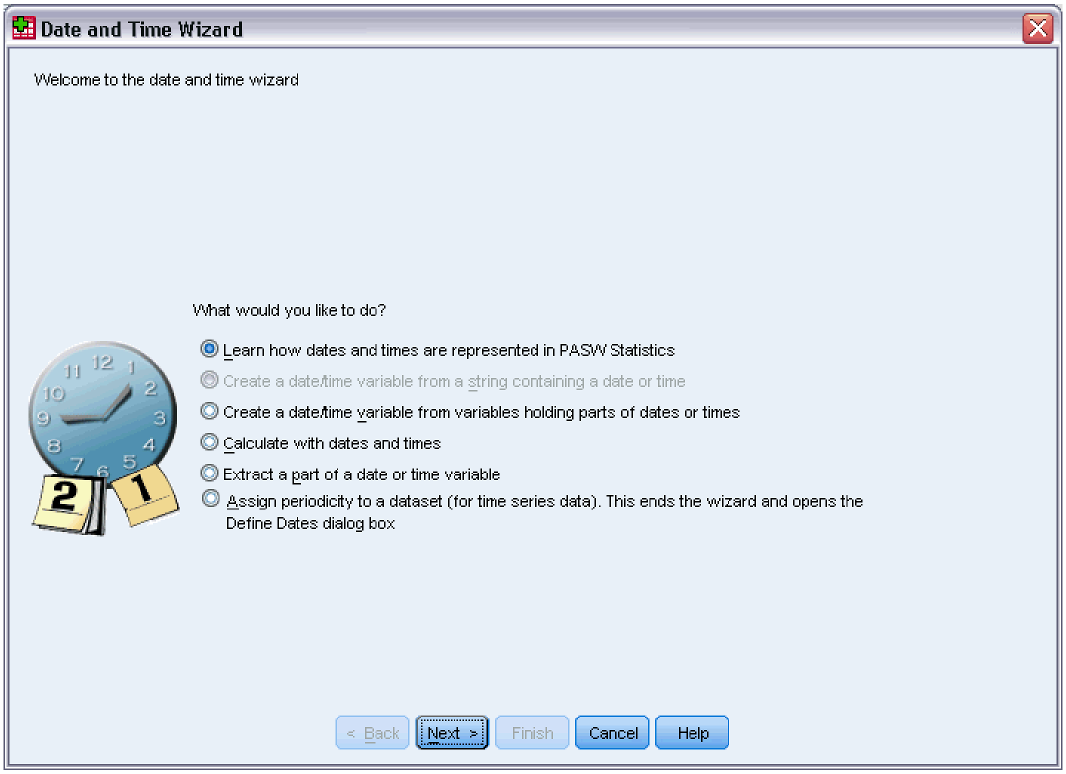
To Use the Date and Time Wizard

- ▶ From the menus choose:

  Transform
    Date and Time Wizard...

- ▶ Select the task you wish to accomplish and follow the steps to define the task.

Figure 8-16
Date and Time Wizard introduction screen



Welcome to the date and time wizard

What would you like to do?

● Learn how dates and times are represented in PASW Statistics
○ Create a date/time variable from a string containing a date or time
○ Create a date/time variable from variables holding parts of dates or times
○ Calculate with dates and times
○ Extract a part of a date or time variable
○ Assign periodicity to a dataset (for time series data). This ends the wizard and opens the Define Dates dialog box

- ■ **Learn how dates and times are represented.** This choice leads to a screen that provides a brief overview of date/time variables in PASW Statistics. By clicking on the Help button, it also provides a link to more detailed information.
- ■ **Create a date/time variable from a string containing a date or time.** Use this option to create a date/time variable from a string variable. For example, you have a string variable representing dates in the form mm/dd/yyyy and want to create a date/time variable from this.
- ■ **Create a date/time variable from variables holding parts of dates or times.** This choice allows you to construct a date/time variable from a set of existing variables. For example, you have a variable that represents the month (as an integer), a second that represents the day of the month, and a third that represents the year. You can combine these three variables into a single date/time variable.
- ■ **Calculate with dates and times.** Use this option to add or subtract values from date/time variables. For example, you can calculate the duration of a process by subtracting a variable representing the start time of the process from another variable representing the end time of the process.

- **Extract a part of a date or time variable.** This option allows you to extract part of a date/time variable, such as the day of the month from a date/time variable, which has the form mm/dd/yyyy.
- **Assign periodicity to a dataset.** This choice takes you to the Define Dates dialog box, used to create date/time variables that consist of a set of sequential dates. This feature is typically used to associate dates with time series data.

Note: Tasks are disabled when the dataset lacks the types of variables required to accomplish the task. For instance, if the dataset contains no string variables, then the task to create a date/time variable from a string does not apply and is disabled.

Dates and Times in PASW Statistics

Variables that represent dates and times in PASW Statistics have a variable type of numeric, with display formats that correspond to the specific date/time formats. These variables are generally referred to as date/time variables. Date/time variables that actually represent dates are distinguished from those that represent a time duration that is independent of any date, such as 20 hours, 10 minutes, and 15 seconds. The latter are referred to as duration variables and the former as date or date/time variables. For a complete list of display formats, see “Date and Time” in the “Universals” section of the *Command Syntax Reference*.

Date and date/time variables. Date variables have a format representing a date, such as mm/dd/yyyy. Date/time variables have a format representing a date and time, such as dd-mmm-yyyy hh:mm:ss. Internally, date and date/time variables are stored as the number of seconds from October 14, 1582. Date and date/time variables are sometimes referred to as date-format variables.

- Both two- and four-digit year specifications are recognized. By default, two-digit years represent a range beginning 69 years prior to the current date and ending 30 years after the current date. This range is determined by your Options settings and is configurable (from the Edit menu, choose Options and click the Data tab).
- Dashes, periods, commas, slashes, or blanks can be used as delimiters in day-month-year formats.
- Months can be represented in digits, Roman numerals, or three-character abbreviations, and they can be fully spelled out. Three-letter abbreviations and fully spelled-out month names must be in English; month names in other languages are not recognized.

Duration variables. Duration variables have a format representing a time duration, such as hh:mm. They are stored internally as seconds without reference to a particular date.

- In time specifications (applies to date/time and duration variables), colons can be used as delimiters between hours, minutes, and seconds. Hours and minutes are required, but seconds are optional. A period is required to separate seconds from fractional seconds. Hours can be of unlimited magnitude, but the maximum value for minutes is 59 and for seconds is 59.999....

Current date and time. The system variable *$TIME* holds the current date and time. It represents the number of seconds from October 14, 1582, to the date and time when the transformation command that uses it is executed.

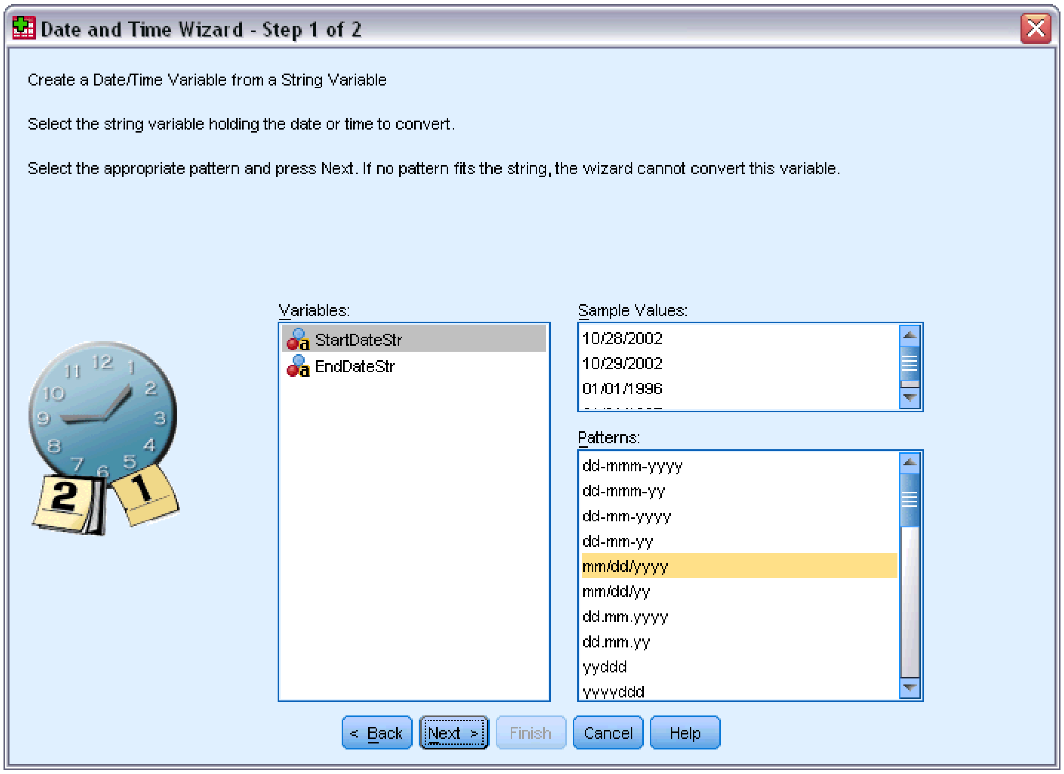
Create a Date/Time Variable from a String

To create a date/time variable from a string variable:

- Select Create a date/time variable from a string containing a date or time on the introduction screen of the Date and Time Wizard.

Select String Variable to Convert to Date/Time Variable

Figure 8-17
Create date/time variable from string variable, step 1



Date and Time Wizard - Step 1 of 2

Create a Date/Time Variable from a String Variable

Select the string variable holding the date or time to convert.

Select the appropriate pattern and press Next. If no pattern fits the string, the wizard cannot convert this variable.

Variables:
StartDateStr
EndDateStr

Sample Values:
10/28/2002
10/29/2002
01/01/1996

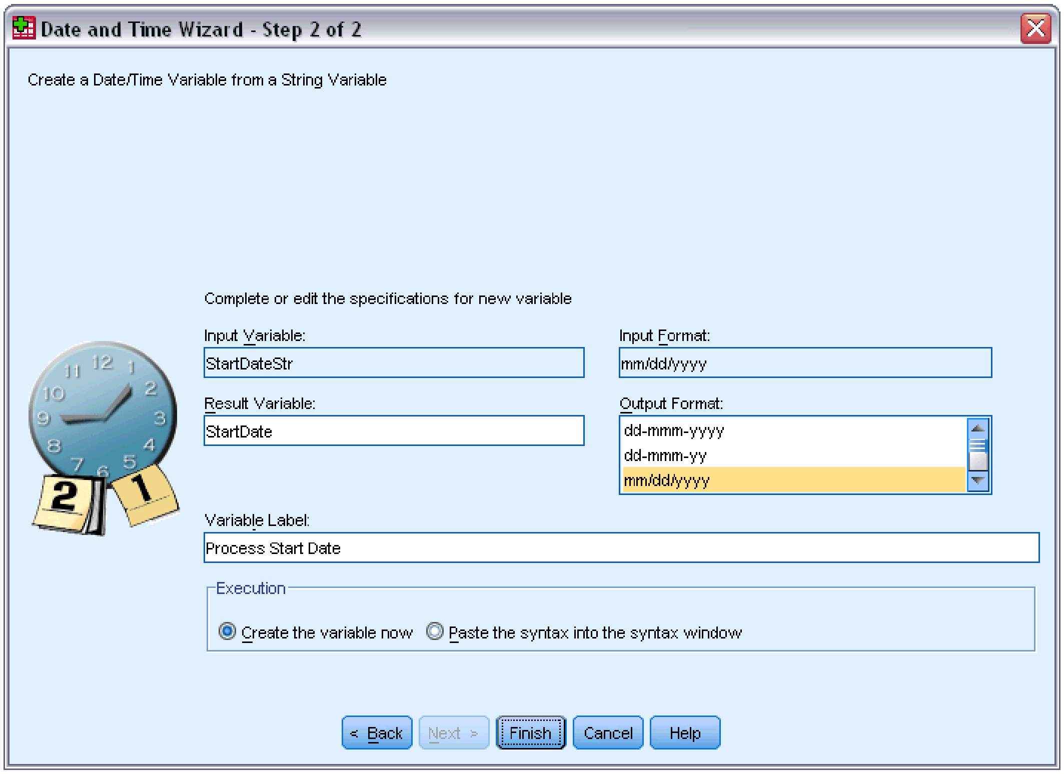
Patterns:
dd-mmm-yyyy
dd-mmm-yy
dd-mm-yyyy
dd-mm-yy
mm/dd/yyyy
mm/dd/yy
dd.mm.yyyy
dd.mm.yy
yyddd
yyyyddd

- Select the string variable to convert in the Variables list. Note that the list displays only string variables.

- Select the pattern from the Patterns list that matches how dates are represented by the string variable. The Sample Values list displays actual values of the selected variable in the data file. Values of the string variable that do not fit the selected pattern result in a value of system-missing for the new variable.

Specify Result of Converting String Variable to Date/Time Variable

Figure 8-18
Create date/time variable from string variable, step 2



Date and Time Wizard - Step 2 of 2

Create a Date/Time Variable from a String Variable

Complete or edit the specifications for new variable

Input Variable: StartDateStr

Input Format: mm/dd/yyyy

Result Variable: StartDate

Output Format: dd-mmm-yyyy, dd-mmm-yy, mm/dd/yyyy (selected)

Variable Label: Process Start Date

Execution

● Create the variable now ○ Paste the syntax into the syntax window

▶ Enter a name for the Result Variable. This cannot be the name of an existing variable.

Optionally, you can:

- Select a date/time format for the new variable from the Output Format list.
- Assign a descriptive variable label to the new variable.

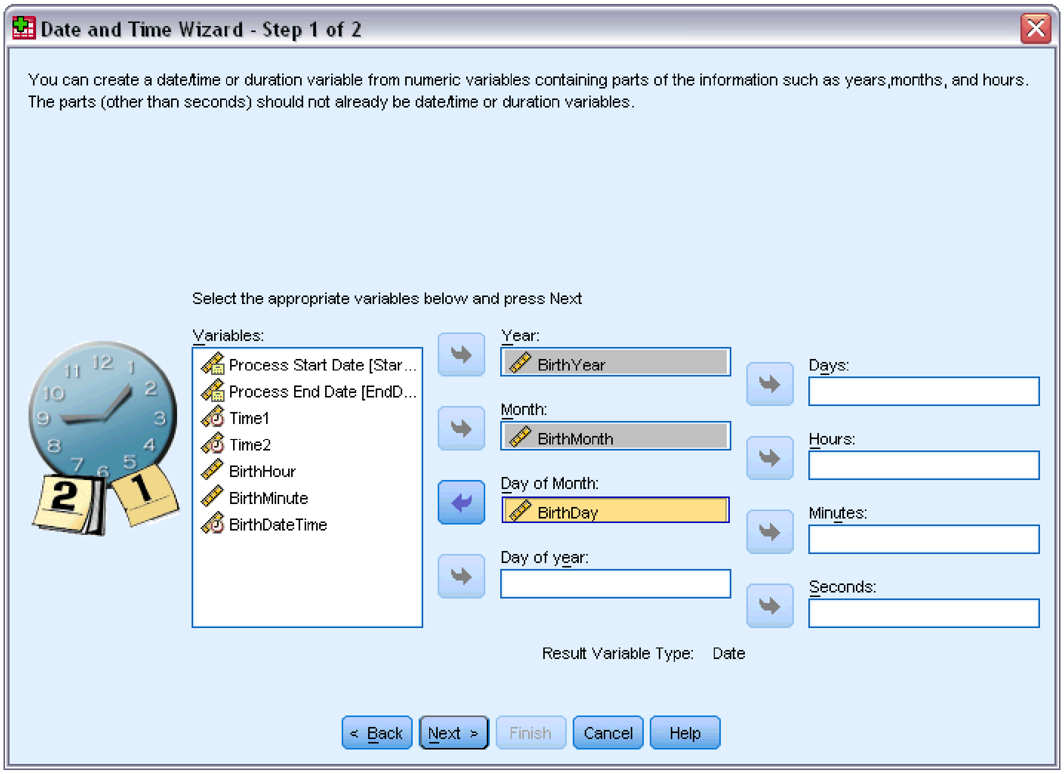
Create a Date/Time Variable from a Set of Variables

To merge a set of existing variables into a single date/time variable:

▶ Select Create a date/time variable from variables holding parts of dates or times on the introduction screen of the Date and Time Wizard.

Select Variables to Merge into Single Date/Time Variable

Figure 8-19
Create date/time variable from variable set, step 1



Date and Time Wizard - Step 1 of 2

You can create a date/time or duration variable from numeric variables containing parts of the information such as years,months, and hours. The parts (other than seconds) should not already be date/time or duration variables.

Select the appropriate variables below and press Next

Variables:
- Process Start Date [Star...
- Process End Date [EndD...
- Time1
- Time2
- BirthHour
- BirthMinute
- BirthDateTime

Year: BirthYear

Month: BirthMonth

Day of Month: BirthDay

Day of year:

Days:

Hours:

Minutes:

Seconds:

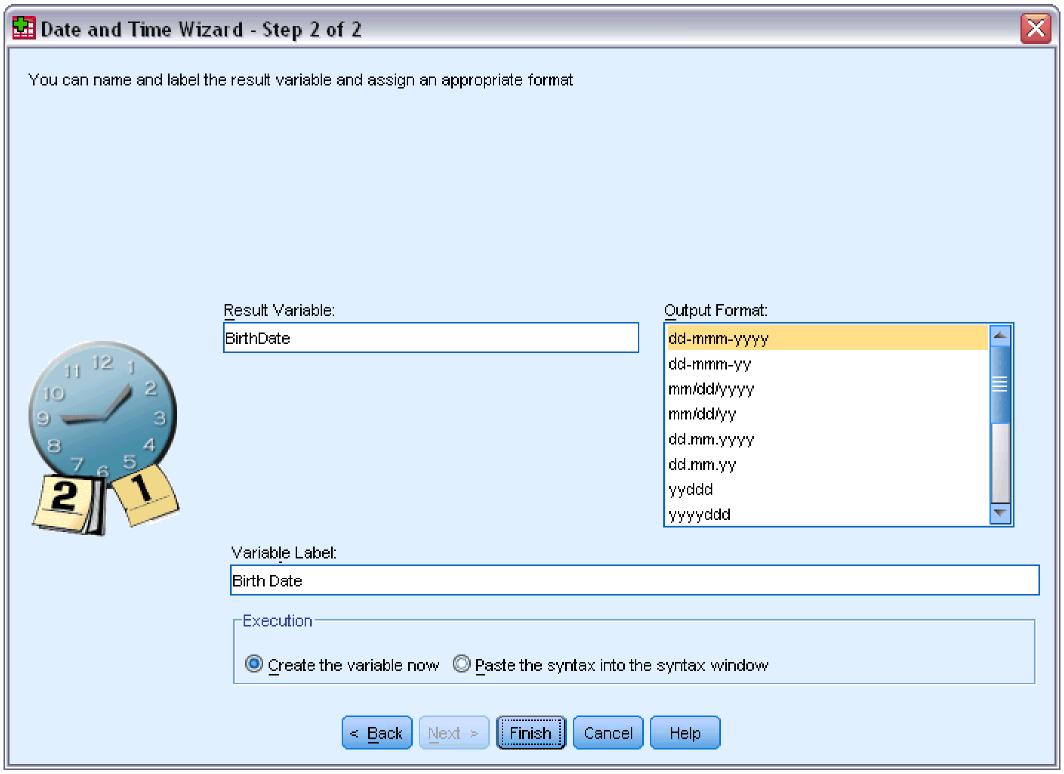
Result Variable Type: Date

▶ Select the variables that represent the different parts of the date/time.

- Some combinations of selections are not allowed. For instance, creating a date/time variable from Year and Day of Month is invalid because once Year is chosen, a full date is required.
- You cannot use an existing date/time variable as one of the parts of the final date/time variable you're creating. Variables that make up the parts of the new date/time variable must be integers. The exception is the allowed use of an existing date/time variable as the Seconds part of the new variable. Since fractional seconds are allowed, the variable used for Seconds is not required to be an integer.
- Values, for any part of the new variable, that are not within the allowed range result in a value of system-missing for the new variable. For instance, if you inadvertently use a variable representing day of month for Month, any cases with day of month values in the range 14–31 will be assigned the system-missing value for the new variable since the valid range for months in PASW Statistics is 1–13.

Specify Date/Time Variable Created by Merging Variables

Figure 8-20
Create date/time variable from variable set, step 2



Date and Time Wizard - Step 2 of 2

You can name and label the result variable and assign an appropriate format

Result Variable: BirthDate

Output Format: dd-mmm-yyyy (selected)

Variable Label: Birth Date

Execution

● Create the variable now ○ Paste the syntax into the syntax window

- ▶ Enter a name for the Result Variable. This cannot be the name of an existing variable.
- ▶ Select a date/time format from the Output Format list.

Optionally, you can:

- ■ Assign a descriptive variable label to the new variable.

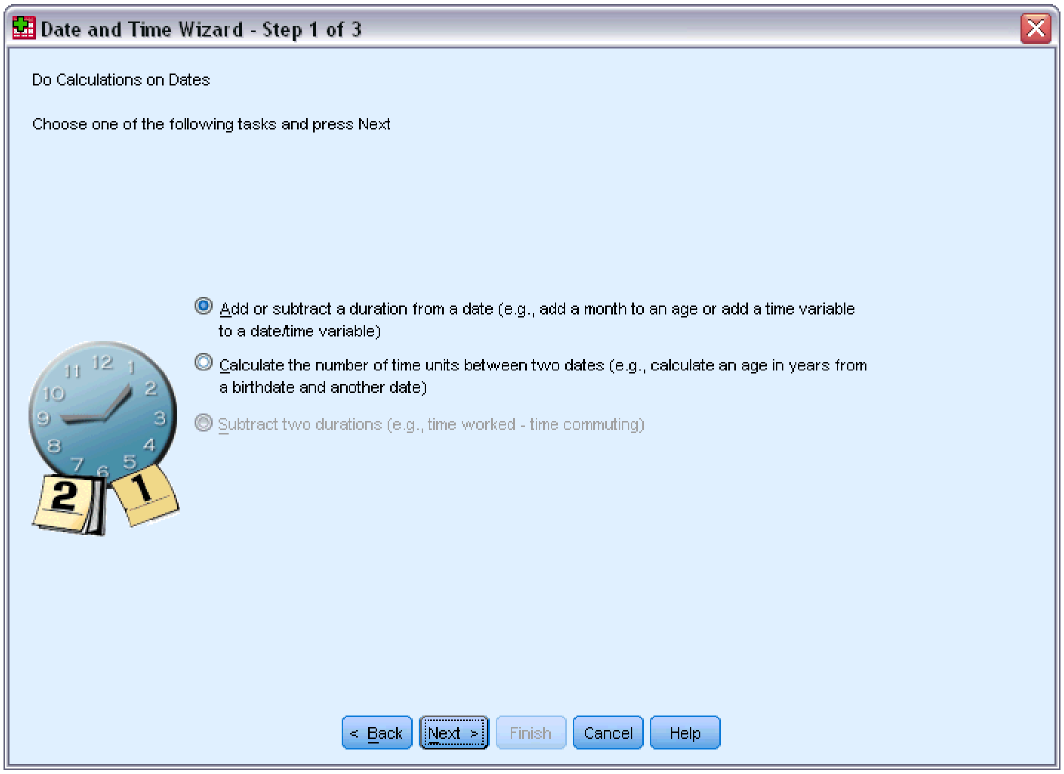
Add or Subtract Values from Date/Time Variables

To add or subtract values from date/time variables:

- ▶ Select Calculate with dates and times on the introduction screen of the Date and Time Wizard.

Select Type of Calculation to Perform with Date/Time Variables

Figure 8-21
Add or subtract values from date/time variables, step 1



Do Calculations on Dates

Choose one of the following tasks and press Next

● Add or subtract a duration from a date (e.g., add a month to an age or add a time variable to a date/time variable)

○ Calculate the number of time units between two dates (e.g., calculate an age in years from a birthdate and another date)

○ Subtract two durations (e.g., time worked - time commuting)

- **Add or subtract a duration from a date.** Use this option to add to or subtract from a date-format variable. You can add or subtract durations that are fixed values, such as 10 days, or the values from a numeric variable, such as a variable that represents years.
- **Calculate the number of time units between two dates.** Use this option to obtain the difference between two dates as measured in a chosen unit. For example, you can obtain the number of years or the number of days separating two dates.
- **Subtract two durations.** Use this option to obtain the difference between two variables that have formats of durations, such as hh:mm or hh:mm:ss.

Note: Tasks are disabled when the dataset lacks the types of variables required to accomplish the task. For instance, if the dataset lacks two variables with formats of durations, then the task to subtract two durations does not apply and is disabled.

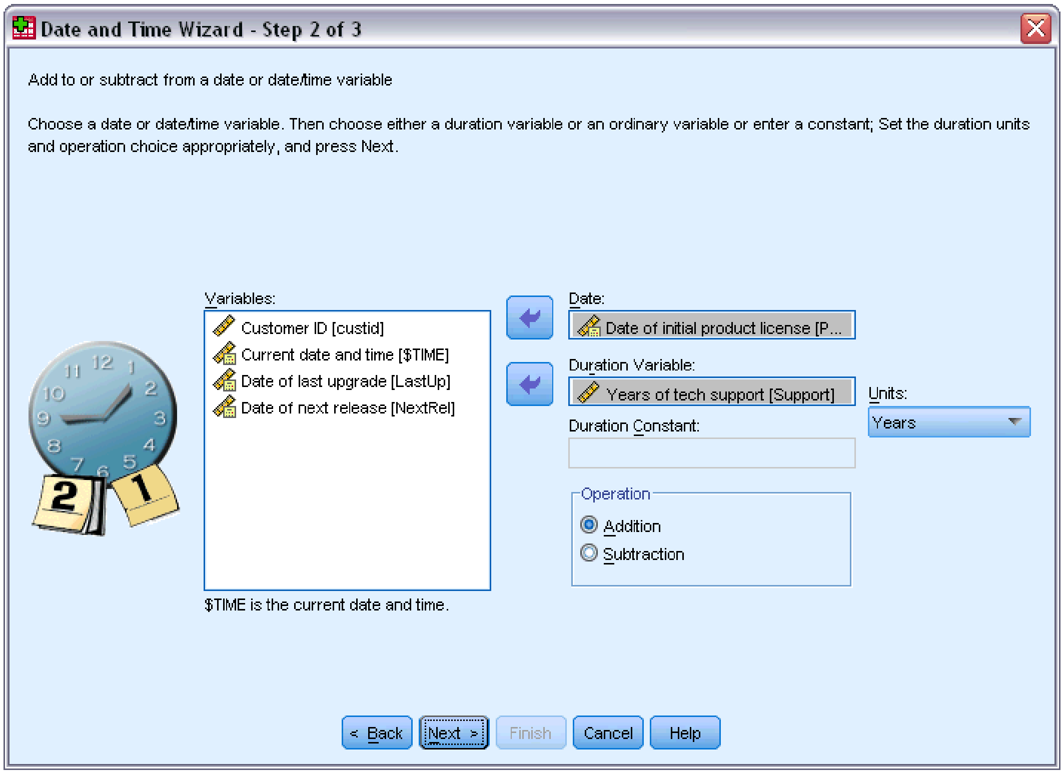
Add or Subtract a Duration from a Date

To add or subtract a duration from a date-format variable:

▶ Select Add or subtract a duration from a date on the screen of the Date and Time Wizard labeled *Do Calculations on Dates*.

Select Date/Time Variable and Duration to Add or Subtract

Figure 8-22
Add or subtract duration, step 2



Date and Time Wizard - Step 2 of 3

Add to or subtract from a date or date/time variable

Choose a date or date/time variable. Then choose either a duration variable or an ordinary variable or enter a constant; Set the duration units and operation choice appropriately, and press Next.

Variables:
- Customer ID [custid]
- Current date and time [$TIME]
- Date of last upgrade [LastUp]
- Date of next release [NextRel]

$TIME is the current date and time.

Date: Date of initial product license [P...

Duration Variable: Years of tech support [Support]

Duration Constant:

Units: Years

Operation
- ● Addition
- ○ Subtraction

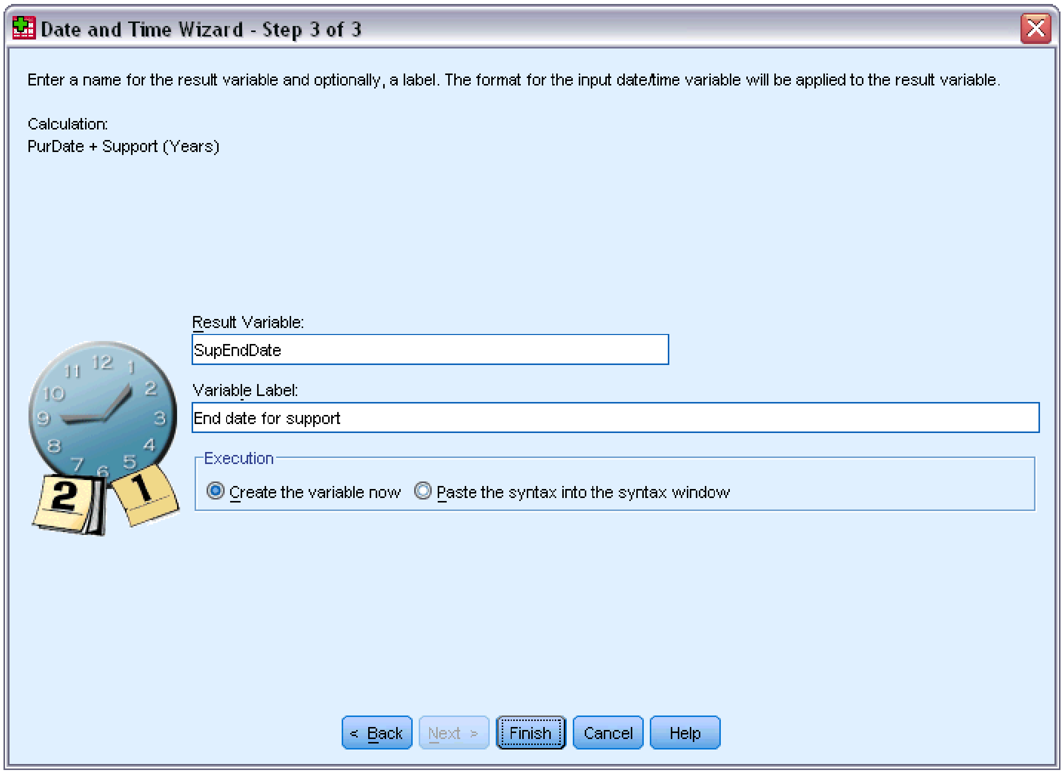
▶ Select a date (or time) variable.

▶ Select a duration variable or enter a value for Duration Constant. Variables used for durations cannot be date or date/time variables. They can be duration variables or simple numeric variables.

▶ Select the unit that the duration represents from the drop-down list. Select Duration if using a variable and the variable is in the form of a duration, such as hh:mm or hh:mm:ss.

Specify Result of Adding or Subtracting a Duration from a Date/Time Variable

Figure 8-23
Add or subtract duration, step 3



Date and Time Wizard - Step 3 of 3

Enter a name for the result variable and optionally, a label. The format for the input date/time variable will be applied to the result variable.

Calculation:
PurDate + Support (Years)

Result Variable: SupEndDate

Variable Label: End date for support

Execution

● Create the variable now ○ Paste the syntax into the syntax window

▶ Enter a name for Result Variable. This cannot be the name of an existing variable.

Optionally, you can:

- Assign a descriptive variable label to the new variable.

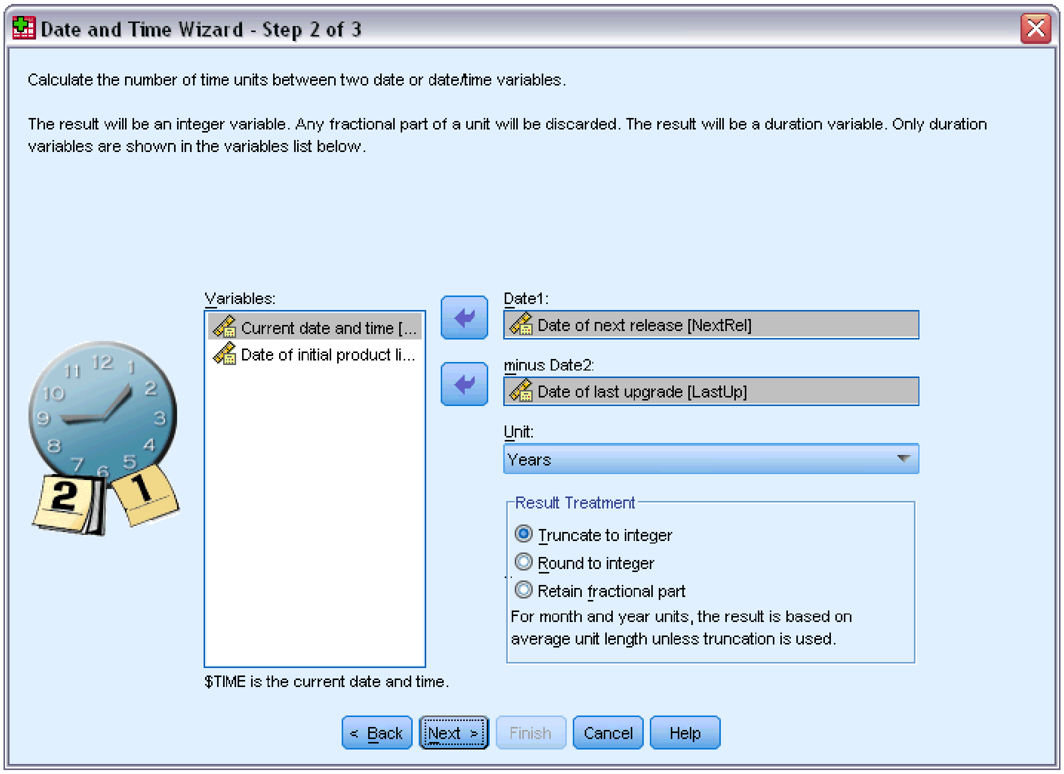
Subtract Date-Format Variables

To subtract two date-format variables:

▶ Select Calculate the number of time units between two dates on the screen of the Date and Time Wizard labeled *Do Calculations on Dates*.

Select Date-Format Variables to Subtract

Figure 8-24
Subtract dates, step 2



Date and Time Wizard - Step 2 of 3

Calculate the number of time units between two date or date/time variables.

The result will be an integer variable. Any fractional part of a unit will be discarded. The result will be a duration variable. Only duration variables are shown in the variables list below.

Variables:
- Current date and time [...
- Date of initial product li...

Date1: Date of next release [NextRel]

minus Date2: Date of last upgrade [LastUp]

Unit: Years

Result Treatment
- ● Truncate to integer
- ○ Round to integer
- ○ Retain fractional part

For month and year units, the result is based on average unit length unless truncation is used.

$TIME is the current date and time.

- ▶ Select the variables to subtract.
- ▶ Select the unit for the result from the drop-down list.
- ▶ Select how the result should be calculated (Result Treatment).

Result Treatment

The following options are available for how the result is calculated:

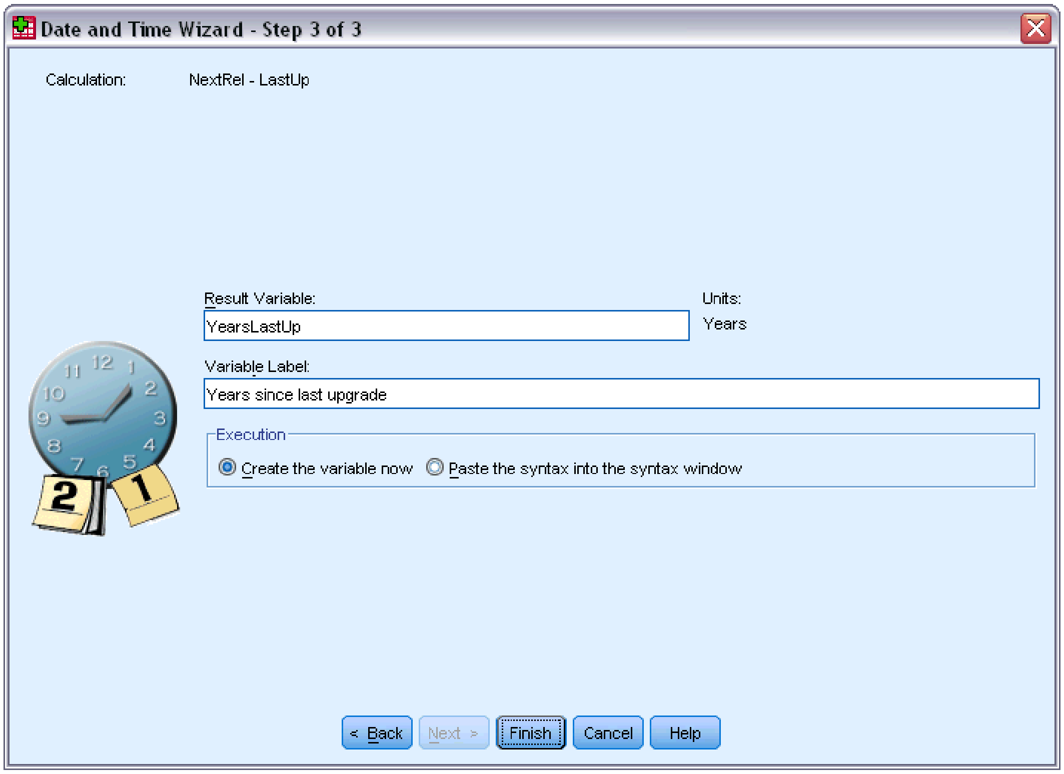
- **Truncate to integer.** Any fractional portion of the result is ignored. For example, subtracting 10/28/2006 from 10/21/2007 returns a result of 0 for years and 11 for months.
- **Round to integer.** The result is rounded to the closest integer. For example, subtracting 10/28/2006 from 10/21/2007 returns a result of 1 for years and 12 for months.
- **Retain fractional part.** The complete value is retained; no rounding or truncation is applied. For example, subtracting 10/28/2006 from 10/21/2007 returns a result of 0.98 for years and 11.76 for months.

For rounding and fractional retention, the result for years is based on average number of days in a year (365.25), and the result for months is based on the average number of days in a month (30.4375). For example, subtracting 2/1/2007 from 3/1/2007 (m/d/y format) returns a fractional result of 0.92 months, whereas subtracting 3/1/2007 from 2/1/2007 returns a fractional difference of 1.02 months. This also affects values calculated on time spans that include leap years. For example, subtracting 2/1/2008 from 3/1/2008 returns a fractional difference of 0.95 months, compared to 0.92 for the same time span in a non-leap year.

Date 1	Date 2	Years			Months		
		Truncate	Round	Fraction	Truncate	Round	Fraction
10/21/2006	10/28/2007	1	1	1.02	12	12	12.22
10/28/2006	10/21/2007	0	1	.98	11	12	11.76
2/1/2007	3/1/2007	0	0	.08	1	1	.92
2/1/2008	3/1/2008	0	0	.08	1	1	.95
3/1/2007	4/1/2007	0	0	.08	1	1	1.02
4/1/2007	5/1/2007	0	0	.08	1	1	.99

Specify Result of Subtracting Two Date-Format Variables

Figure 8-25
Subtract dates, step 3



Calculation: NextRel - LastUp

Result Variable: YearsLastUp

Units: Years

Variable Label: Years since last upgrade

Execution

● Create the variable now ○ Paste the syntax into the syntax window

▶ Enter a name for Result Variable. This cannot be the name of an existing variable.

Optionally, you can:

- Assign a descriptive variable label to the new variable.

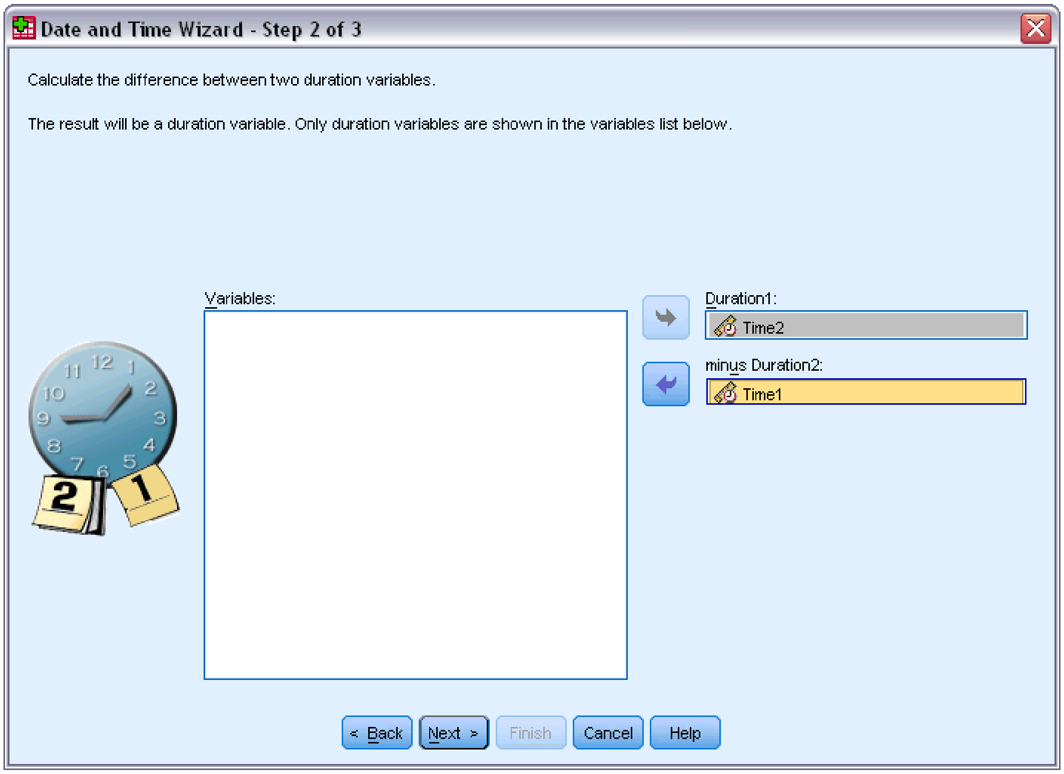
Subtract Duration Variables

To subtract two duration variables:

- ▶ Select Subtract two durations on the screen of the Date and Time Wizard labeled *Do Calculations on Dates*.

Select Duration Variables to Subtract

Figure 8-26
Subtract two durations, step 2



Date and Time Wizard - Step 2 of 3

Calculate the difference between two duration variables.

The result will be a duration variable. Only duration variables are shown in the variables list below.

Variables:

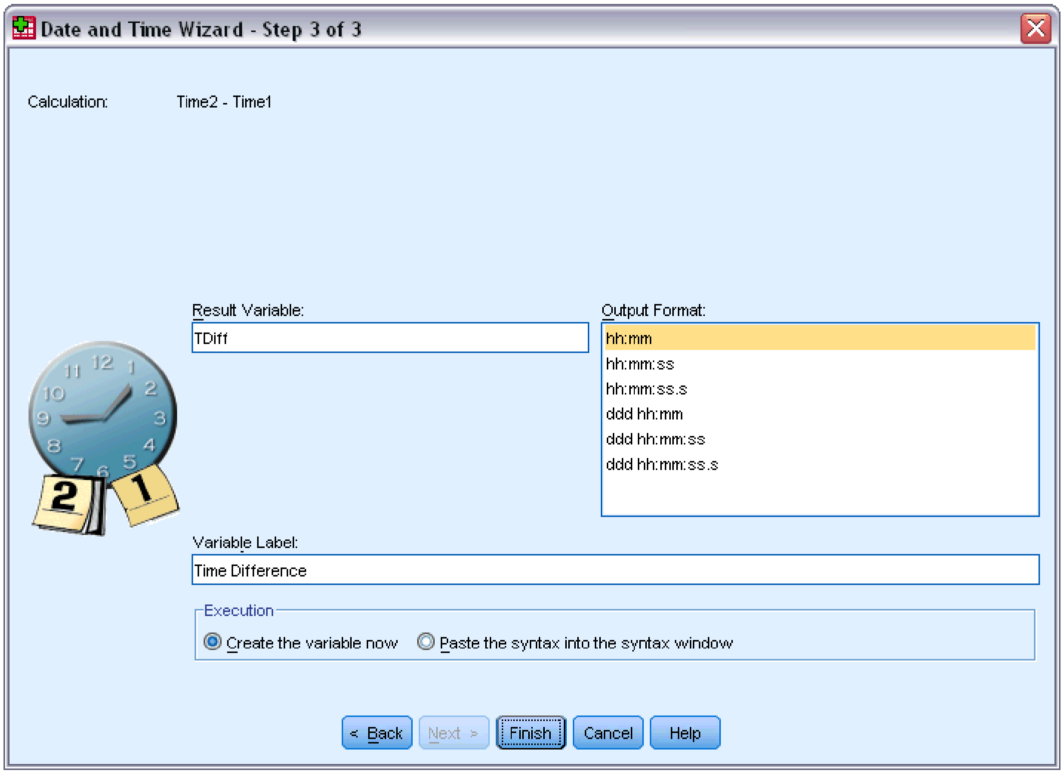
Duration1: Time2

minus Duration2: Time1

- ▶ Select the variables to subtract.

Specify Result of Subtracting Two Duration Variables

Figure 8-27
Subtract two durations, step 3



Date and Time Wizard - Step 3 of 3

Calculation: Time2 - Time1

Result Variable: TDiff

Output Format:
- hh:mm
- hh:mm:ss
- hh:mm:ss.s
- ddd hh:mm
- ddd hh:mm:ss
- ddd hh:mm:ss.s

Variable Label: Time Difference

Execution
● Create the variable now ○ Paste the syntax into the syntax window

▶ Enter a name for Result Variable. This cannot be the name of an existing variable.

▶ Select a duration format from the Output Format list.

Optionally, you can:

■ Assign a descriptive variable label to the new variable.

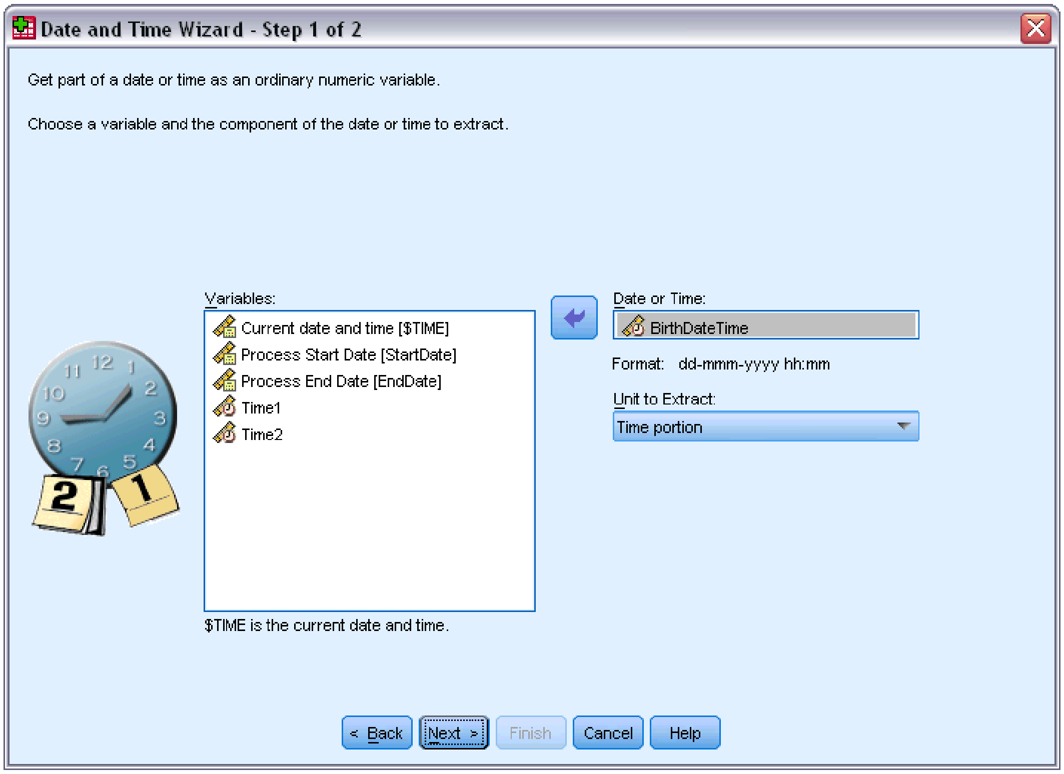
Extract Part of a Date/Time Variable

To extract a component—such as the year—from a date/time variable:

▶ Select Extract a part of a date or time variable on the introduction screen of the Date and Time Wizard.

Select Component to Extract from Date/Time Variable

Figure 8-28
Get part of a date/time variable, step 1



Get part of a date or time as an ordinary numeric variable.

Choose a variable and the component of the date or time to extract.

Variables:
- Current date and time [$TIME]
- Process Start Date [StartDate]
- Process End Date [EndDate]
- Time1
- Time2

$TIME is the current date and time.

Date or Time: BirthDateTime

Format: dd-mmm-yyyy hh:mm

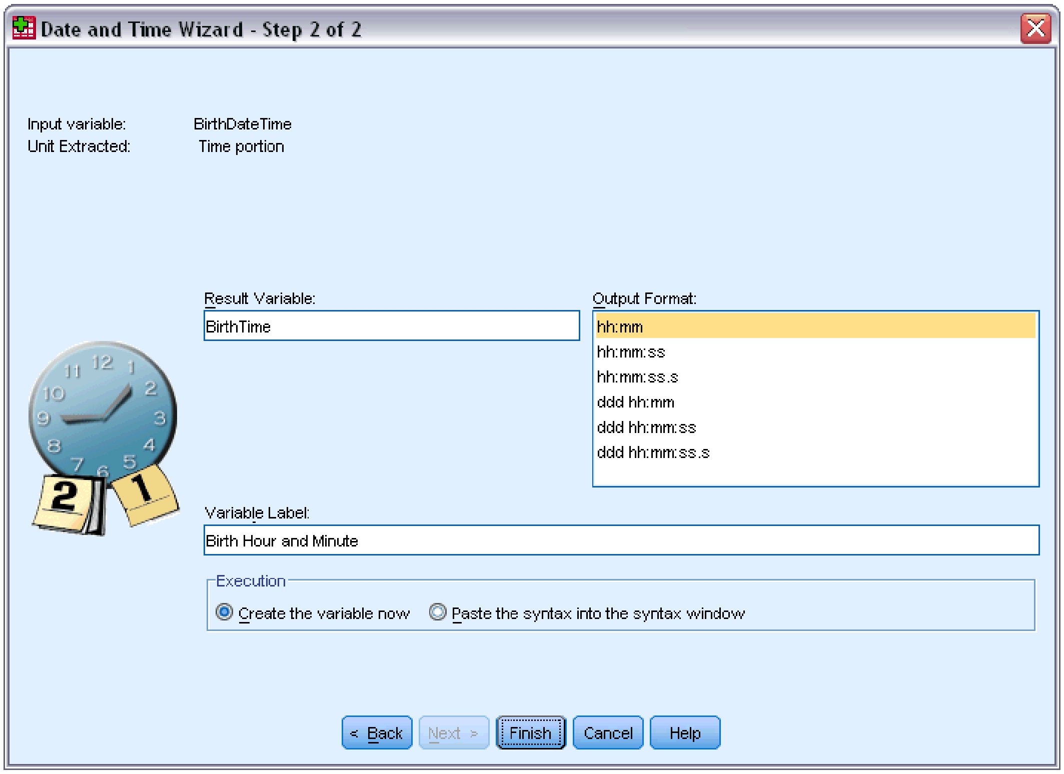
Unit to Extract: Time portion

- ▶ Select the variable containing the date or time part to extract.

- ▶ Select the part of the variable to extract, from the drop-down list. You can extract information from dates that is not explicitly part of the display date, such as day of the week.

Specify Result of Extracting Component from Date/Time Variable

Figure 8-29
Get part of a date/time variable, step 2



Date and Time Wizard - Step 2 of 2

Input variable: BirthDateTime
Unit Extracted: Time portion

Result Variable: BirthTime

Output Format:
- hh:mm
- hh:mm:ss
- hh:mm:ss.s
- ddd hh:mm
- ddd hh:mm:ss
- ddd hh:mm:ss.s

Variable Label: Birth Hour and Minute

Execution
● Create the variable now ○ Paste the syntax into the syntax window

▶ Enter a name for Result Variable. This cannot be the name of an existing variable.

▶ If you're extracting the date or time part of a date/time variable, then you must select a format from the Output Format list. In cases where the output format is not required, the Output Format list will be disabled.

Optionally, you can:

- Assign a descriptive variable label to the new variable.

Time Series Data Transformations

Several data transformations that are useful in time series analysis are provided:

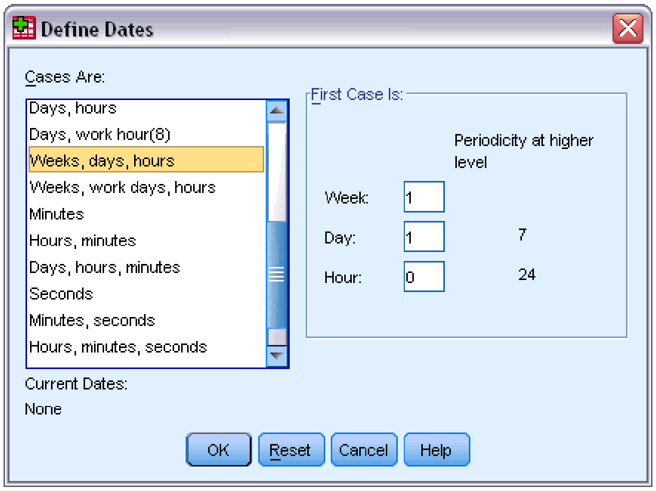
- Generate date variables to establish periodicity and to distinguish between historical, validation, and forecasting periods.
- Create new time series variables as functions of existing time series variables.
- Replace system- and user-missing values with estimates based on one of several methods.

A **time series** is obtained by measuring a variable (or set of variables) regularly over a period of time. Time series data transformations assume a data file structure in which each case (row) represents a set of observations at a different time, and the length of time between cases is uniform.

Define Dates

The Define Dates dialog box allows you to generate date variables that can be used to establish the periodicity of a **time series** and to label output from time series analysis.

Figure 8-30
Define Dates dialog box



Cases Are: Weeks, days, hours

First Case Is:

		Periodicity at higher level
Week:	1	
Day:	1	7
Hour:	0	24

Current Dates:
None

Cases Are. Defines the time interval used to generate dates.

- Not dated removes any previously defined date variables. Any variables with the following names are deleted: *year_*, *quarter_*, *month_*, *week_*, *day_*, *hour_*, *minute_*, *second_*, and *date_*.
- Custom indicates the presence of custom date variables created with command syntax (for example, a four-day workweek). This item merely reflects the current state of the active dataset. Selecting it from the list has no effect.

First Case Is. Defines the starting date value, which is assigned to the first case. Sequential values, based on the time interval, are assigned to subsequent cases.

Periodicity at higher level. Indicates the repetitive cyclical variation, such as the number of months in a year or the number of days in a week. The value displayed indicates the maximum value you can enter. For hours, minutes, and seconds the maximum is the displayed value minus one.

A new numeric variable is created for each component that is used to define the date. The new variable names end with an underscore. A descriptive string variable, *date_*, is also created from the components. For example, if you selected Weeks, days, hours, four new variables are created: *week_*, *day_*, *hour_*, and *date_*.

If date variables have already been defined, they are replaced when you define new date variables that will have the same names as the existing date variables.

To Define Dates for Time Series Data

▶ From the menus choose:

Data
  Define Dates...

- ▶ Select a time interval from the Cases Are list.
- ▶ Enter the value(s) that define the starting date for First Case Is, which determines the date assigned to the first case.

Date Variables versus Date Format Variables

Date variables created with Define Dates should not be confused with date format variables defined in the Variable View of the Data Editor. Date variables are used to establish periodicity for time series data. Date format variables represent dates and/or times displayed in various date/time formats. Date variables are simple integers representing the number of days, weeks, hours, and so on, from a user-specified starting point. Internally, most date format variables are stored as the number of seconds from October 14, 1582.

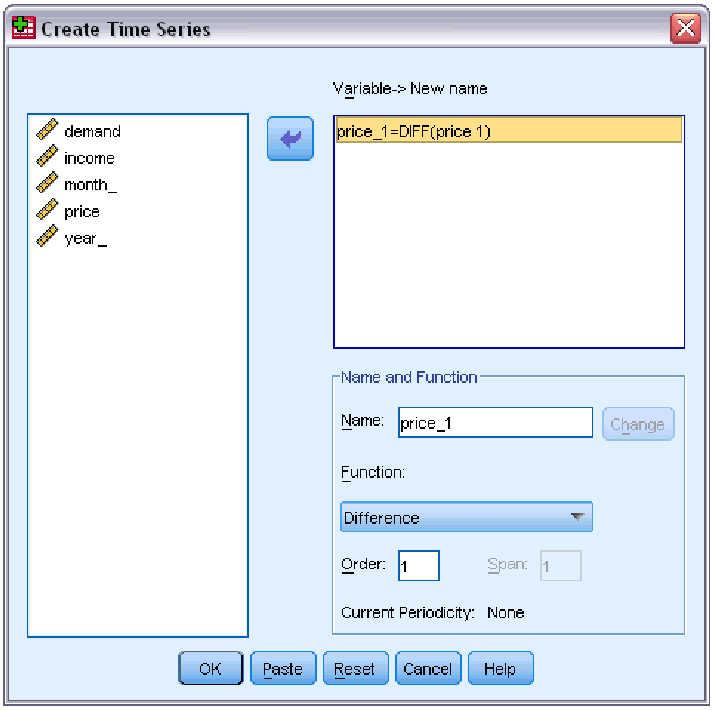
Create Time Series

The Create Time Series dialog box allows you to create new variables based on functions of existing numeric **time series** variables. These transformed values are useful in many time series analysis procedures.

Default new variable names are the first six characters of the existing variable used to create it, followed by an underscore and a sequential number. For example, for the variable *price*, the new variable name would be *price_1*. The new variables retain any defined value labels from the original variables.

Available functions for creating time series variables include differences, moving averages, running medians, lag, and lead functions.

Figure 8-31
Create Time Series dialog box



Variable-> New name: price_1=DIFF(price 1)

Name: price_1

Function: Difference

Order: 1

Span: 1

Current Periodicity: None

To Create New Time Series Variables

▶ From the menus choose:

Transform
  Create Time Series...

▶ Select the time series function that you want to use to transform the original variable(s).

▶ Select the variable(s) from which you want to create new time series variables. Only numeric variables can be used.

Optionally, you can:

- Enter variable names to override the default new variable names.
- Change the function for a selected variable.

Time Series Transformation Functions

Difference. Nonseasonal difference between successive values in the series. The order is the number of previous values used to calculate the difference. Because one observation is lost for each order of difference, system-missing values appear at the beginning of the series. For example, if the difference order is 2, the first two cases will have the system-missing value for the new variable.

Seasonal difference. Difference between series values a constant span apart. The span is based on the currently defined periodicity. To compute seasonal differences, you must have defined date variables (Data menu, Define Dates) that include a periodic component (such as months of the year). The order is the number of seasonal periods used to compute the difference. The number of cases with the system-missing value at the beginning of the series is equal to the periodicity multiplied by the order. For example, if the current periodicity is 12 and the order is 2, the first 24 cases will have the system-missing value for the new variable.

Centered moving average. Average of a span of series values surrounding and including the current value. The span is the number of series values used to compute the average. If the span is even, the moving average is computed by averaging each pair of uncentered means. The number of cases with the system-missing value at the beginning and at the end of the series for a span of *n* is equal to *n*/2 for even span values and (*n*–1)/2 for odd span values. For example, if the span is 5, the number of cases with the system-missing value at the beginning and at the end of the series is 2.

Prior moving average. Average of the span of series values preceding the current value. The span is the number of preceding series values used to compute the average. The number of cases with the system-missing value at the beginning of the series is equal to the span value.

Running medians. Median of a span of series values surrounding and including the current value. The span is the number of series values used to compute the median. If the span is even, the median is computed by averaging each pair of uncentered medians. The number of cases with the system-missing value at the beginning and at the end of the series for a span of *n* is equal to *n*/2 for even span values and (*n*–1)/2 for odd span values. For example, if the span is 5, the number of cases with the system-missing value at the beginning and at the end of the series is 2.

Cumulative sum. Cumulative sum of series values up to and including the current value.

Lag. Value of a previous case, based on the specified lag order. The order is the number of cases prior to the current case from which the value is obtained. The number of cases with the system-missing value at the beginning of the series is equal to the order value.

Lead. Value of a subsequent case, based on the specified lead order. The order is the number of cases after the current case from which the value is obtained. The number of cases with the system-missing value at the end of the series is equal to the order value.

Smoothing. New series values based on a compound data smoother. The smoother starts with a running median of 4, which is centered by a running median of 2. It then resmoothes these values by applying a running median of 5, a running median of 3, and hanning (running weighted averages). Residuals are computed by subtracting the smoothed series from the original series. This whole process is then repeated on the computed residuals. Finally, the smoothed residuals are computed by subtracting the smoothed values obtained the first time through the process. This is sometimes referred to as **T4253H smoothing**.

Replace Missing Values

Missing observations can be problematic in analysis, and some time series measures cannot be computed if there are missing values in the series. Sometimes the value for a particular observation is simply not known. In addition, missing data can result from any of the following:

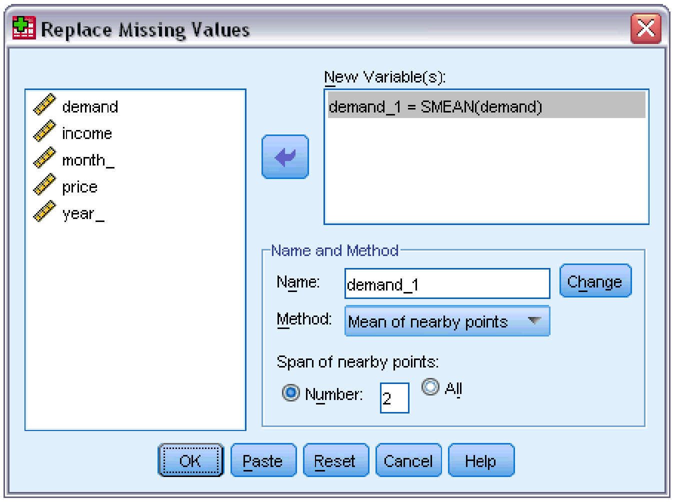
- Each degree of differencing reduces the length of a series by 1.

- Each degree of seasonal differencing reduces the length of a series by one season.
- If you create new series that contain forecasts beyond the end of the existing series (by clicking a Save button and making suitable choices), the original series and the generated residual series will have missing data for the new observations.
- Some transformations (for example, the log transformation) produce missing data for certain values of the original series.

Missing data at the beginning or end of a series pose no particular problem; they simply shorten the useful length of the series. Gaps in the middle of a series (*embedded* missing data) can be a much more serious problem. The extent of the problem depends on the analytical procedure you are using.

The Replace Missing Values dialog box allows you to create new **time series** variables from existing ones, replacing missing values with estimates computed with one of several methods. Default new variable names are the first six characters of the existing variable used to create it, followed by an underscore and a sequential number. For example, for the variable *price*, the new variable name would be *price_1*. The new variables retain any defined value labels from the original variables.

Figure 8-32
Replace Missing Values dialog box



New Variable(s): demand_1 = SMEAN(demand)

Name: demand_1

Method: Mean of nearby points

Span of nearby points: ● Number: 2 ○ All

To Replace Missing Values for Time Series Variables

▶ From the menus choose:

Transform
  Replace Missing Values...

▶ Select the estimation method you want to use to replace missing values.

▶ Select the variable(s) for which you want to replace missing values.

Optionally, you can:

- Enter variable names to override the default new variable names.
- Change the estimation method for a selected variable.

Estimation Methods for Replacing Missing Values

Series mean. Replaces missing values with the mean for the entire series.

Mean of nearby points. Replaces missing values with the mean of valid surrounding values. The span of nearby points is the number of valid values above and below the missing value used to compute the mean.

Median of nearby points. Replaces missing values with the median of valid surrounding values. The span of nearby points is the number of valid values above and below the missing value used to compute the median.

Linear interpolation. Replaces missing values using a linear interpolation. The last valid value before the missing value and the first valid value after the missing value are used for the interpolation. If the first or last case in the series has a missing value, the missing value is not replaced.

Linear trend at point. Replaces missing values with the linear trend for that point. The existing series is regressed on an index variable scaled 1 to *n*. Missing values are replaced with their predicted values.

Scoring Data with Predictive Models

The process of applying a predictive model to a set of data is referred to as **scoring** the data. PASW Statistics, PASW Modeler, and AnswerTree have procedures for building predictive models such as regression, clustering, tree, and neural network models. Once a model has been built, the model specifications can be saved as an XML file containing all of the information necessary to reconstruct the model. The PASW Statistics Server product then provides the means to read an XML model file and apply the model to a dataset.

Example. A credit application is rated for risk based on various aspects of the applicant and the loan in question. The credit score obtained from the risk model is used to accept or reject the loan application.

Scoring is treated as a transformation of the data. The model is expressed internally as a set of numeric transformations to be applied to a given set of variables—the predictor variables specified in the model—in order to obtain a predicted result. In this sense, the process of scoring data with a given model is inherently the same as applying any function, such as a square root function, to a set of data.

Scoring is only available with PASW Statistics Server and can be done interactively by users working in distributed analysis mode. For scoring large data files, you'll probably want to make use of the Batch Facility, a separate executable provided with PASW Statistics Server. For information about using the Batch Facility, see the *Batch Facility User's Guide*, provided as a PDF file on the PASW Statistics Server product CD.

The scoring process consists of the following:

- Loading a model from a file in XML (PMML) format.
- Computing your scores as a new variable, using the ApplyModel or StrApplyModel function available in the Compute Variable dialog box.

For details concerning the `ApplyModel` or `StrApplyModel` function, see Scoring Expressions in the Transformation Expressions section of the *Command Syntax Reference*.

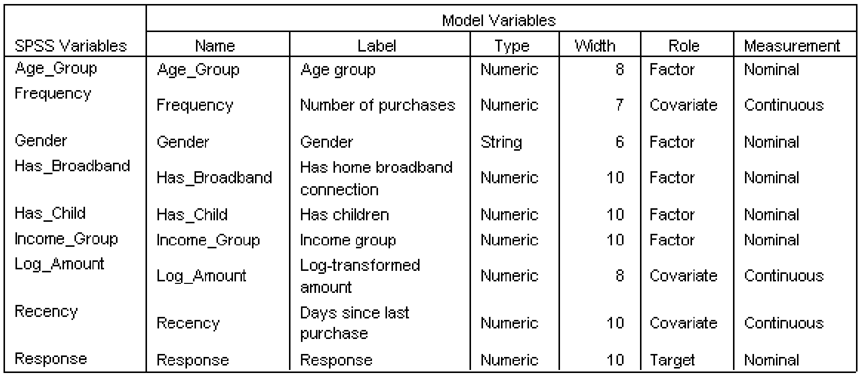
The following table lists the procedures that support the export of model specifications to XML. The exported models can be used with PASW Statistics Server to score new data as described above. The full list of model types that can be scored with PASW Statistics Server is provided in the description of the `ApplyModel` function.

Procedure name	**Command name**	**Option**
Discriminant	`DISCRIMINANT`	Base
Linear Regression	`REGRESSION`	Base
TwoStep Cluster	`TWOSTEP CLUSTER`	Base
Generalized Linear Models	`GENLIN`	Advanced Statistics
Complex Samples General Linear Model	`CSGLM`	Complex Samples
Complex Samples Logistic Regression	`CSLOGISTIC`	Complex Samples
Complex Samples Ordinal Regression	`CSORDINAL`	Complex Samples
Logistic Regression	`LOGISTIC REGRESSION`	Regression
Multinomial Logistic Regression	`NOMREG`	Regression
Decision Tree	`TREE`	Decision Tree

Loading a Saved Model

The Load Model dialog box allows you to load predictive models saved in XML (PMML) format and is only available when working in distributed analysis mode. Loading models is a necessary first step to scoring data with them.

Figure 8-33
Load Model output



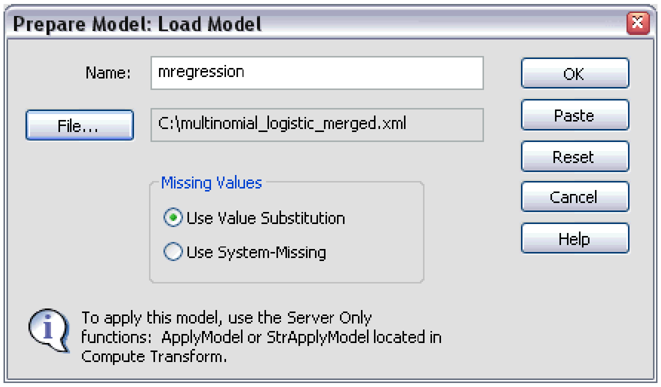
	Model Variables					
SPSS Variables	Name	Label	Type	Width	Role	Measurement
Age_Group	Age_Group	Age group	Numeric	8	Factor	Nominal
Frequency	Frequency	Number of purchases	Numeric	7	Covariate	Continuous
Gender	Gender	Gender	String	6	Factor	Nominal
Has_Broadband	Has_Broadband	Has home broadband connection	Numeric	10	Factor	Nominal
Has_Child	Has_Child	Has children	Numeric	10	Factor	Nominal
Income_Group	Income_Group	Income group	Numeric	10	Factor	Nominal
Log_Amount	Log_Amount	Log-transformed amount	Numeric	8	Covariate	Continuous
Recency	Recency	Days since last purchase	Numeric	10	Covariate	Continuous
Response	Response	Response	Numeric	10	Target	Nominal

To Load a Model

▶ From the menus choose:

Transform
  Prepare Model
    Load Model...

Figure 8-34
Prepare Model Load Model dialog box



Prepare Model: Load Model

Name: mregression

File... C:\multinomial_logistic_merged.xml

Missing Values

● Use Value Substitution

○ Use System-Missing

To apply this model, use the Server Only functions: ApplyModel or StrApplyModel located in Compute Transform.

▶ Enter a name to associate with this model. Each loaded model must have a unique name.

▶ Click File and select a model file. The resulting Open File dialog box displays the files that are available in distributed analysis mode. This includes files on the machine where PASW Statistics Server is installed and files on your local computer that reside in shared folders or on shared drives.

Note: When you score data, the model will be applied to variables in the active dataset with the same names as the variables from the model file. You can map variables in the original model to different variables in the active dataset with the use of command syntax (see the `MODEL HANDLE` command).

Name. A name used to identify this model. The rules for valid model names are the same as for variable names (see Variable Names in Chapter 5 on p. 70), with the addition of the $ character as an allowed first character. You will use this name to specify the model when scoring data with the ApplyModel or StrApplyModel functions.

File. The XML (PMML) file containing the model specifications.

Missing Values

This group of options controls the treatment of missing values, encountered during the scoring process, for the predictor variables defined in the model. A missing value in the context of scoring refers to one of the following:

- A predictor variable contains no value. For numeric variables, this means the system-missing value. For string variables, this means a null string.
- The value has been defined as user-missing, in the model, for the given predictor. Values defined as user-missing in the active dataset, but not in the model, are not treated as missing values in the scoring process.
- The predictor variable is categorical and the value is not one of the categories defined in the model.

Use Value Substitution. Attempt to use value substitution when scoring cases with missing values. The method for determining a value to substitute for a missing value depends on the type of predictive model.

- **PASW Statistics Models.** For independent variables in linear regression and discriminant models, if mean value substitution for missing values was specified when building and saving the model, then this mean value is used in place of the missing value in the scoring computation, and scoring proceeds. If the mean value is not available, then the system-missing value is returned.
- **AnswerTree models & TREE command models.** For the CHAID and Exhaustive CHAID models, the biggest child node is selected for a missing split variable. The biggest child node is the one with the largest population among the child nodes using learning sample cases. For C&RT and QUEST models, surrogate split variables (if any) are used first. (Surrogate splits are splits that attempt to match the original split as closely as possible using alternate predictors.) If no surrogate splits are specified or all surrogate split variables are missing, the biggest child node is used.
- **PASW Modeler models.** Linear regression models are handled as described under PASW Statistics models. Logistic regression models are handled as described under Logistic Regression models. C&RT Tree models are handled as described for C&RT models under AnswerTree models.
- **Logistic Regression models.** For covariates in logistic regression models, if a mean value of the predictor was included as part of the saved model, then this mean value is used in place of the missing value in the scoring computation, and scoring proceeds. If the predictor is categorical (for example, a factor in a logistic regression model), or if the mean value is not available, then the system-missing value is returned.

Use System-Missing. Return the system-missing value when scoring a case with a missing value.

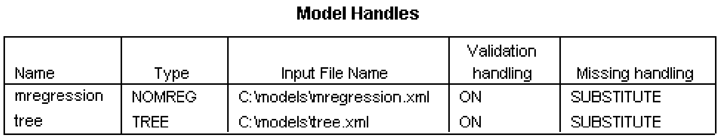
Displaying a List of Loaded Models

You can obtain a list of the currently loaded models. From the menus (only available in distributed analysis mode) choose:

Transform
  Prepare Model
    List Model(s)

This will generate a Model Handles table. The table contains a list of all currently loaded models and includes the name (referred to as the model handle) assigned to the model, the type of model, the path to the model file, and the method for handling missing values.

Figure 8-35
Listing of loaded models



Model Handles

Name	Type	Input File Name	Validation handling	Missing handling
mregression	NOMREG	C:\models\mregression.xml	ON	SUBSTITUTE
tree	TREE	C:\models\tree.xml	ON	SUBSTITUTE

Additional Features Available with Command Syntax

From the Load Model dialog box, you can paste your selections into a syntax window and edit the resulting `MODEL HANDLE` command syntax. This allows you to:

- Map variables in the original model to different variables in the active dataset (with the `MAP` subcommand). By default, the model is applied to variables in the active dataset with the same names as the variables from the model file.

See the *Command Syntax Reference* for complete syntax information.

Chapter

9

File Handling and File Transformations

Data files are not always organized in the ideal form for your specific needs. You may want to combine data files, sort the data in a different order, select a subset of cases, or change the unit of analysis by grouping cases together. A wide range of file transformation capabilities is available, including the ability to:

Sort data. You can sort cases based on the value of one or more variables.

Transpose cases and variables. The PASW Statistics data file format reads rows as cases and columns as variables. For data files in which this order is reversed, you can switch the rows and columns and read the data in the correct format.

Merge files. You can merge two or more data files. You can combine files with the same variables but different cases or the same cases but different variables.

Select subsets of cases. You can restrict your analysis to a subset of cases or perform simultaneous analyses on different subsets.

Aggregate data. You can change the unit of analysis by aggregating cases based on the value of one or more grouping variables.

Weight data. You can weight cases for analysis based on the value of a weight variable.

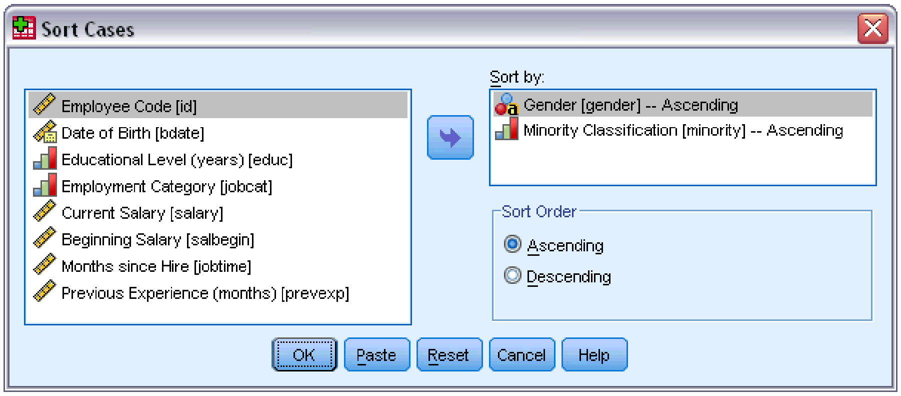
Restructure data. You can restructure data to create a single case (record) from multiple cases or create multiple cases from a single case.

Sort Cases

This dialog box sorts cases (rows) of the data file based on the values of one or more sorting variables. You can sort cases in ascending or descending order.

- If you select multiple sort variables, cases are sorted by each variable within categories of the preceding variable on the Sort list. For example, if you select *gender* as the first sorting variable and *minority* as the second sorting variable, cases will be sorted by minority classification within each gender category.
- The sort sequence is based on the locale-defined order (and is not necessarily the same as the numerical order of the character codes). The default locale is the operating system locale. You can control the locale with the Language setting on the General tab of the Options dialog box (Edit menu).

Figure 9-1
Sort Cases dialog box



Sort Cases
Employee Code [id]
Date of Birth [bdate]
Educational Level (years) [educ]
Employment Category [jobcat]
Current Salary [salary]
Beginning Salary [salbegin]
Months since Hire [jobtime]
Previous Experience (months) [prevexp]
OK
Paste
Reset
Cancel
Help

Sort by:
Gender [gender] -- Ascending
Minority Classification [minority] -- Ascending

Sort Order
● Ascending
○ Descending

To Sort Cases

▶ From the menus choose:

Data
  Sort Cases...

▶ Select one or more sorting variables.

Sort Variables

You can sort the variables in the active dataset based on the values of any of the variable attributes (e.g., variable name, data type, measurement level), including custom variable attributes.

- Values can be sorted in ascending or descending order.
- You can save the original (pre-sorted) variable order in a custom variable attribute.
- Sorting by values of custom variable attributes is limited to custom variable attributes that are currently visible in Variable View.

For more information on custom variable attributes, see Custom Variable Attributes.

To Sort Variables

In Variable View of the Data Editor:

▶ Right-click on the attribute column heading and from the context menu choose Sort Ascending or Sort Descending.

or

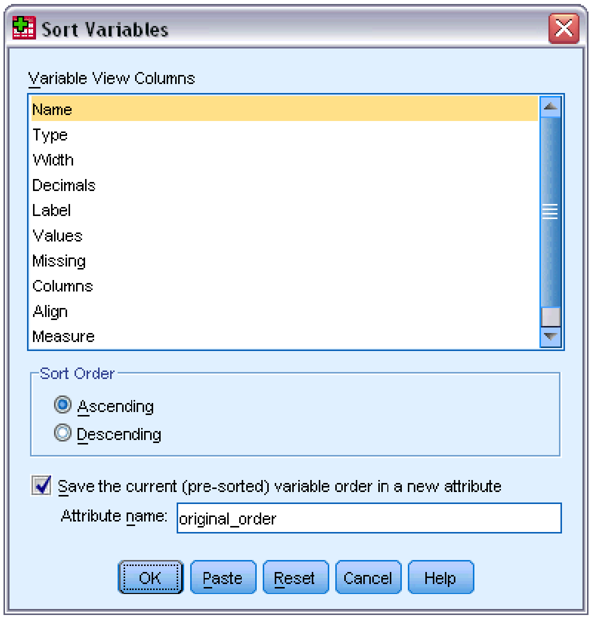
▶ From the menus in Variable View or Data View, choose:

Data
  Sort Variables

▶ Select the attribute you want to use to sort variables.

▶ Select the sort order (ascending or descending).

Figure 9-2
Sort Variables dialog



Sort Variables

Variable View Columns
- Name
- Type
- Width
- Decimals
- Label
- Values
- Missing
- Columns
- Align
- Measure

Sort Order
- ● Ascending
- ○ Descending

☑ Save the current (pre-sorted) variable order in a new attribute

Attribute name: original_order

- The list of variable attributes matches the attribute column names displayed in Variable View of the Data Editor.
- You can save the original (pre-sorted) variable order in a custom variable attribute. For each variable, the value of the attribute is an integer value indicating its position prior to sorting; so by sorting variables based on the value of that custom attribute you can restore the original variable order.

Transpose

Transpose creates a new data file in which the rows and columns in the original data file are transposed so that cases (rows) become variables and variables (columns) become cases. Transpose automatically creates new variable names and displays a list of the new variable names.

- A new string variable that contains the original variable name, *case_lbl*, is automatically created.
- If the active dataset contains an ID or name variable with unique values, you can use it as the name variable, and its values will be used as variable names in the transposed data file. If it is a numeric variable, the variable names start with the letter *V*, followed by the numeric value.
- User-missing values are converted to the system-missing value in the transposed data file. To retain any of these values, change the definition of missing values in the Variable View in the Data Editor.

To Transpose Variables and Cases

▶ From the menus choose:

Data
  Transpose...

- ▶ Select one or more variables to transpose into cases.

Merging Data Files

You can merge data from two files in two different ways. You can:

- Merge the active dataset with another open dataset or PASW Statistics data file containing the same variables but different cases.
- Merge the active dataset with another open dataset or PASW Statistics data file containing the same cases but different variables.

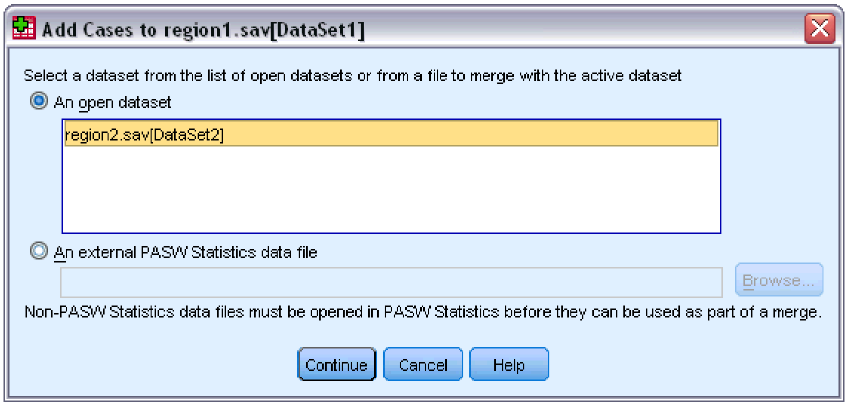
To Merge Files

- ▶ From the menus choose:

Data
  Merge Files

- ▶ Select Add Cases or Add Variables.

Figure 9-3
Selecting files to merge



Add Cases to region1.sav[DataSet1]

Select a dataset from the list of open datasets or from a file to merge with the active dataset

● An open dataset

region2.sav[DataSet2]

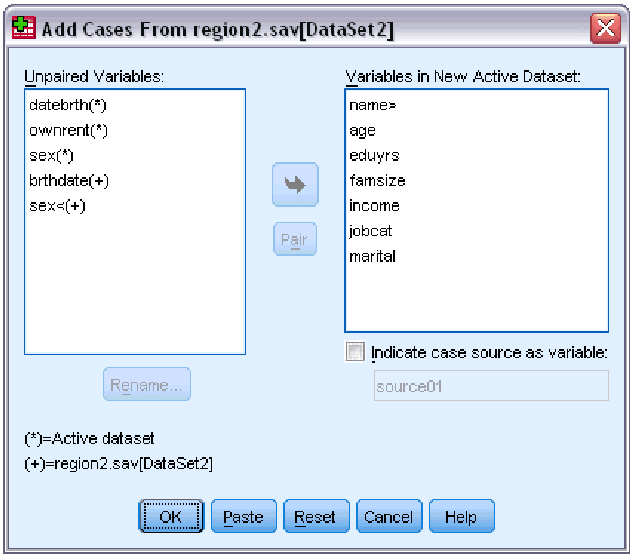
○ An external PASW Statistics data file

Non-PASW Statistics data files must be opened in PASW Statistics before they can be used as part of a merge.

Add Cases

Add Cases merges the active dataset with a second dataset or external PASW Statistics data file that contains the same variables (columns) but different cases (rows). For example, you might record the same information for customers in two different sales regions and maintain the data for each region in separate files. The second dataset can be an external PASW Statistics data file or a dataset available in the current session.

Figure 9-4
Add Cases dialog box



Add Cases From region2.sav[DataSet2]

Unpaired Variables:
datebrth(*)
ownrent(*)
sex(*)
brthdate(+)
sex<(+)

Variables in New Active Dataset:
name>
age
eduyrs
famsize
income
jobcat
marital

□ Indicate case source as variable:
source01

(*)=Active dataset
(+)=region2.sav[DataSet2]

Unpaired Variables. Variables to be excluded from the new, merged data file. Variables from the active dataset are identified with an asterisk (*). Variables from the other dataset are identified with a plus sign (+). By default, this list contains:

- Variables from either data file that do not match a variable name in the other file. You can create pairs from unpaired variables and include them in the new, merged file.
- Variables defined as numeric data in one file and string data in the other file. Numeric variables cannot be merged with string variables.
- String variables of unequal width. The defined width of a string variable must be the same in both data files.

Variables in New Active Dataset. Variables to be included in the new, merged data file. By default, all of the variables that match both the name and the data type (numeric or string) are included on the list.

- You can remove variables from the list if you do not want them to be included in the merged file.
- Any unpaired variables included in the merged file will contain missing data for cases from the file that does not contain that variable.

Indicate case source as variable. Indicates the source data file for each case. This variable has a value of 0 for cases from the active dataset and a value of 1 for cases from the external data file.

To Merge Data Files with the Same Variables and Different Cases

▶ Open at least one of the data files that you want to merge. If you have multiple datasets open, make one of the datasets that you want to merge the active dataset. The cases from this file will appear first in the new, merged data file.

▶ From the menus choose:

Data
  Merge Files
    Add Cases...

▶ Select the dataset or external PASW Statistics data file to merge with the active dataset.

▶ Remove any variables that you do not want from the Variables in New Active Dataset list.

▶ Add any variable pairs from the Unpaired Variables list that represent the same information recorded under different variable names in the two files. For example, date of birth might have the variable name *brthdate* in one file and *datebrth* in the other file.

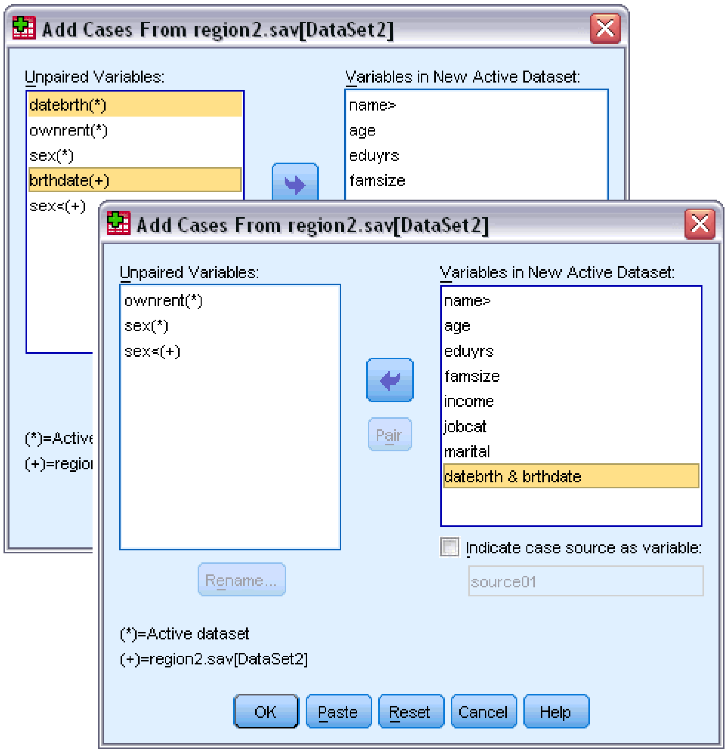
To Select a Pair of Unpaired Variables

▶ Click one of the variables on the Unpaired Variables list.

▶ Ctrl-click the other variable on the list. (Press the Ctrl key and click the left mouse button at the same time.)

▶ Click Pair to move the variable pair to the Variables in New Active Dataset list. (The variable name from the active dataset is used as the variable name in the merged file.)

Figure 9-5
Selecting pairs of variables with Ctrl-click



Add Cases: Rename

You can rename variables from either the active dataset or the other dataset before moving them from the unpaired list to the list of variables to be included in the merged data file. Renaming variables enables you to:

- Use the variable name from the other dataset rather than the name from the active dataset for variable pairs.
- Include two variables with the same name but of unmatched types or different string widths. For example, to include both the numeric variable *sex* from the active dataset and the string variable *sex* from the other dataset, one of them must be renamed first.

Add Cases: Dictionary Information

Any existing dictionary information (variable and value labels, user-missing values, display formats) in the active dataset is applied to the merged data file.

- If any dictionary information for a variable is undefined in the active dataset, dictionary information from the other dataset is used.
- If the active dataset contains any defined value labels or user-missing values for a variable, any additional value labels or user-missing values for that variable in the other dataset are ignored.

Merging More Than Two Data Sources

Using command syntax, you can merge up to 50 datasets and/or data files. For more information, see the `ADD FILES` command in the *Command Syntax Reference* (available from the Help menu).

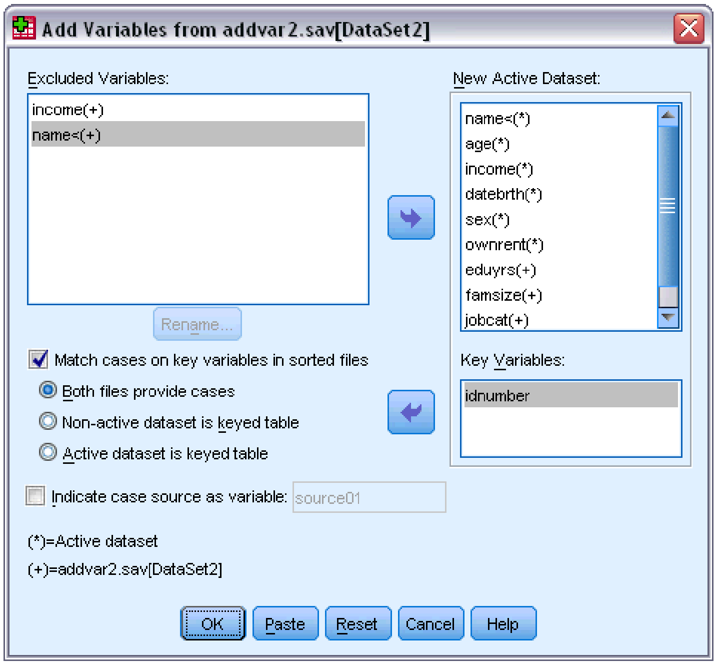
Add Variables

Add Variables merges the active dataset with another open dataset or external PASW Statistics data file that contains the same cases (rows) but different variables (columns). For example, you might want to merge a data file that contains pre-test results with one that contains post-test results.

- Cases must be sorted in the same order in both datasets.
- If one or more key variables are used to match cases, the two datasets must be sorted by ascending order of the key variable(s).
- Variable names in the second data file that duplicate variable names in the active dataset are excluded by default because Add Variables assumes that these variables contain duplicate information.

Indicate case source as variable. Indicates the source data file for each case. This variable has a value of 0 for cases from the active dataset and a value of 1 for cases from the external data file.

Figure 9-6
Add Variables dialog box



Add Variables from addvar2.sav[DataSet2]

Excluded Variables:
income(+)
name<(+)

New Active Dataset:
name<(*)
age(*)
income(*)
datebrth(*)
sex(*)
ownrent(*)
eduyrs(+)
famsize(+)
jobcat(+)

Key Variables:
idnumber

☑ Match cases on key variables in sorted files
● Both files provide cases
○ Non-active dataset is keyed table
○ Active dataset is keyed table

□ Indicate case source as variable: source01

(*)=Active dataset
(+)=addvar2.sav[DataSet2]

Excluded Variables. Variables to be excluded from the new, merged data file. By default, this list contains any variable names from the other dataset that duplicate variable names in the active dataset. Variables from the active dataset are identified with an asterisk (*). Variables from the other dataset are identified with a plus sign (+). If you want to include an excluded variable with a duplicate name in the merged file, you can rename it and add it to the list of variables to be included.

New Active Dataset. Variables to be included in the new, merged dataset. By default, all unique variable names in both datasets are included on the list.

Key Variables. If some cases in one dataset do not have matching cases in the other dataset (that is, some cases are missing in one dataset), use key variables to identify and correctly match cases from the two datasets. You can also use key variables with table lookup files.

- The key variables must have the same names in both datasets.
- Both datasets must be sorted by ascending order of the key variables, and the order of variables on the Key Variables list must be the same as their sort sequence.
- Cases that do not match on the key variables are included in the merged file but are not merged with cases from the other file. Unmatched cases contain values for only the variables in the file from which they are taken; variables from the other file contain the system-missing value.

Non-active or active dataset is keyed table. A keyed table, or **table lookup file**, is a file in which data for each "case" can be applied to multiple cases in the other data file. For example, if one file contains information on individual family members (such as sex, age, education) and the other file contains overall family information (such as total income, family size, location), you can use the file of family data as a table lookup file and apply the common family data to each individual family member in the merged data file.

To Merge Files with the Same Cases but Different Variables

- ▶ Open at least one of the data files that you want to merge. If you have multiple datasets open, make one of the datasets that you want to merge the active dataset.
- ▶ From the menus choose:

  Data
    Merge Files
      Add Variables...
- ▶ Select the dataset or external PASW Statistics data file to merge with the active dataset.

To Select Key Variables

- ▶ Select the variables from the external file variables (+) on the Excluded Variables list.
- ▶ Select Match cases on key variables in sorted files.
- ▶ Add the variables to the Key Variables list.

  The key variables must exist in both the active dataset and the other dataset. Both datasets must be sorted by ascending order of the key variables, and the order of variables on the Key Variables list must be the same as their sort sequence.

Add Variables: Rename

You can rename variables from either the active dataset or the other dataset before moving them to the list of variables to be included in the merged data file. This is primarily useful if you want to include two variables with the same name that contain different information in the two files.

Merging More Than Two Data Sources

Using command syntax, you can merge up to 50 datasets and/or data files. For more information, see the `MATCH FILES` command in the *Command Syntax Reference* (available from the Help menu).

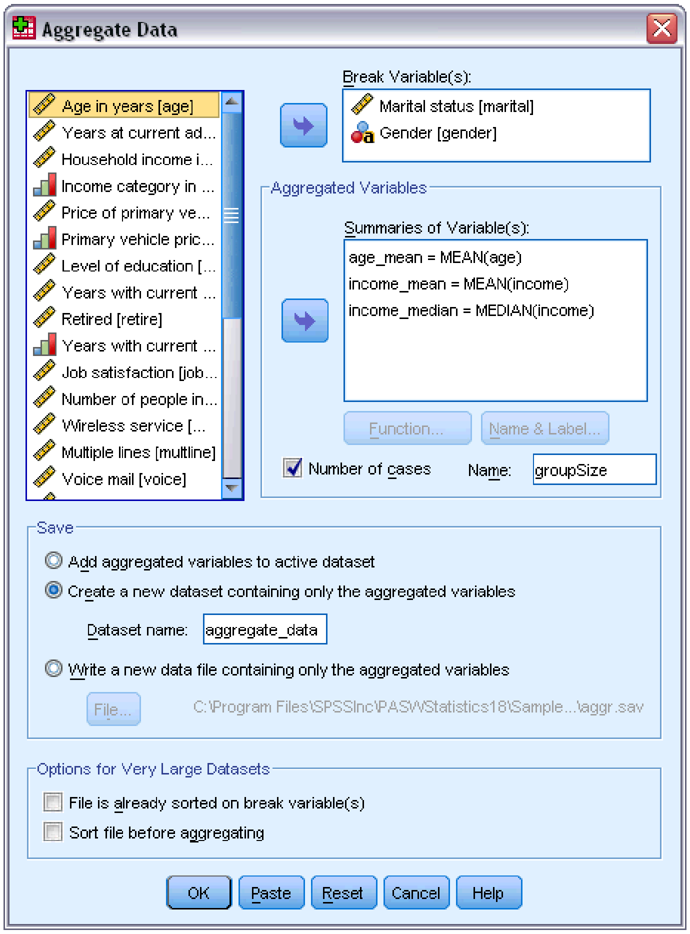
Aggregate Data

Aggregate Data aggregates groups of cases in the active dataset into single cases and creates a new, aggregated file or creates new variables in the active dataset that contain aggregated data. Cases are aggregated based on the value of *zero* or more break (grouping) variables. If no break variables are specified, then the entire dataset is a single break group.

- ■ If you create a new, aggregated data file, the new data file contains one case for each group defined by the break variables. For example, if there is one break variable with two values, the new data file will contain only two cases. If no break variable is specified, the new data file will contain one case.
- ■ If you add aggregate variables to the active dataset, the data file itself is not aggregated. Each case with the same value(s) of the break variable(s) receives the same values for the new aggregate variables. For example, if *gender* is the only break variable, all males would

receive the same value for a new aggregate variable that represents average age. If no break variable is specified, all cases would receive the same value for a new aggregate variable that represents average age.

Figure 9-7
Aggregate Data dialog box



Break Variable(s):
Marital status [marital]
Gender [gender]

Aggregated Variables

Summaries of Variable(s):

age_mean = MEAN(age)
income_mean = MEAN(income)
income_median = MEDIAN(income)

☑ Number of cases Name: groupSize

○ Add aggregated variables to active dataset
● Create a new dataset containing only the aggregated variables
Dataset name: aggregate_data
○ Write a new data file containing only the aggregated variables
C:\Program Files\SPSSInc\PASWStatistics18\Sample...\aggr.sav

□ File is already sorted on break variable(s)
□ Sort file before aggregating

Break Variable(s). Cases are grouped together based on the values of the break variables. Each unique combination of break variable values defines a group. When creating a new, aggregated data file, all break variables are saved in the new file with their existing names and dictionary information. The break variable, if specified, can be either numeric or string.

Aggregated Variables. Source variables are used with aggregate functions to create new aggregate variables. The aggregate variable name is followed by an optional variable label, the name of the aggregate function, and the source variable name in parentheses.

You can override the default aggregate variable names with new variable names, provide descriptive variable labels, and change the functions used to compute the aggregated data values. You can also create a variable that contains the number of cases in each break group.

To Aggregate a Data File

- ▶ From the menus choose:

  Data
    Aggregate...

- ▶ Optionally select break variables that define how cases are grouped to create aggregated data. If no break variables are specified, then the entire dataset is a single break group.

- ▶ Select one or more aggregate variables.

- ▶ Select an aggregate function for each aggregate variable.

Saving Aggregated Results

You can add aggregate variables to the active dataset or create a new, aggregated data file.

- **Add aggregated variables to active dataset.** New variables based on aggregate functions are added to the active dataset. The data file itself is not aggregated. Each case with the same value(s) of the break variable(s) receives the same values for the new aggregate variables.
- **Create a new dataset containing only the aggregated variables.** Saves aggregated data to a new dataset in the current session. The dataset includes the break variables that define the aggregated cases and all aggregate variables defined by aggregate functions. The active dataset is unaffected.
- **Write a new data file containing only the aggregated variables.** Saves aggregated data to an external data file. The file includes the break variables that define the aggregated cases and all aggregate variables defined by aggregate functions. The active dataset is unaffected.

Sorting Options for Large Data Files

For very large data files, it may be more efficient to aggregate presorted data.

File is already sorted on break variable(s). If the data have already been sorted by values of the break variables, this option enables the procedure to run more quickly and use less memory. Use this option with caution.

- Data must by sorted by values of the break variables in the same order as the break variables specified for the Aggregate Data procedure.
- If you are adding variables to the active dataset, select this option only if the data are sorted by ascending values of the break variables.

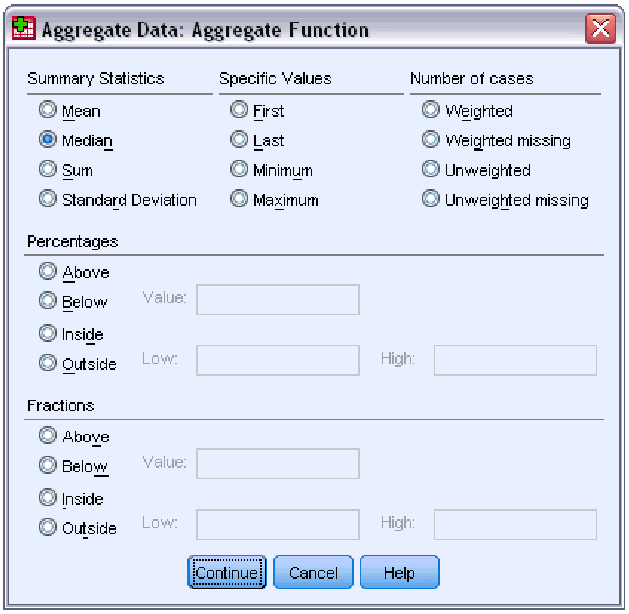
Sort file before aggregating. In very rare instances with large data files, you may find it necessary to sort the data file by values of the break variables prior to aggregating. This option is not recommended unless you encounter memory or performance problems.

Aggregate Data: Aggregate Function

This dialog box specifies the function to use to calculate aggregated data values for selected variables on the Aggregate Variables list in the Aggregate Data dialog box. Aggregate functions include:

- Summary functions for numeric variables, including mean, median, standard deviation, and sum
- Number of cases, including unweighted, weighted, nonmissing, and missing
- Percentage or fraction of values above or below a specified value
- Percentage or fraction of values inside or outside of a specified range

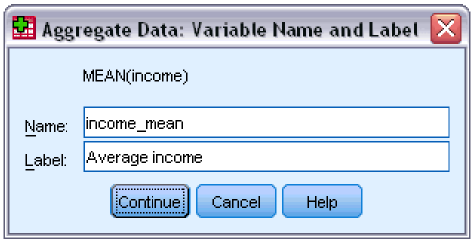
Figure 9-8
Aggregate Function dialog box



Aggregate Data: Variable Name and Label

Aggregate Data assigns default variable names for the aggregated variables in the new data file. This dialog box enables you to change the variable name for the selected variable on the Aggregate Variables list and provide a descriptive variable label. For more information, see the topic Variable Names in Chapter 5 on p. 70.

Figure 9-9
Variable Name and Label dialog box



Aggregate Data: Variable Name and Label

MEAN(income)

Name: income_mean

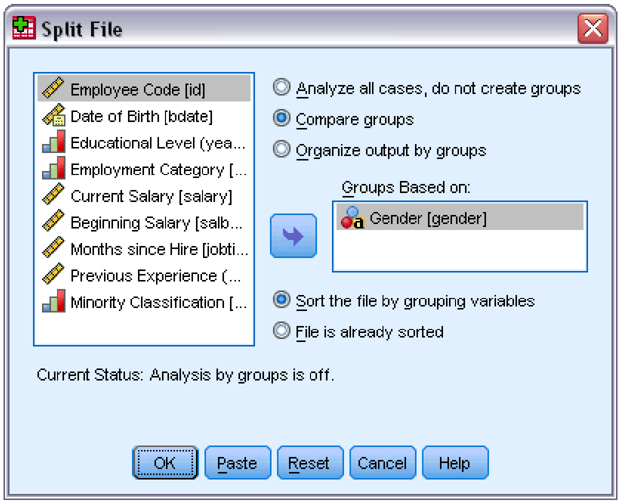
Label: Average income

Split File

Split File splits the data file into separate groups for analysis based on the values of one or more grouping variables. If you select multiple grouping variables, cases are grouped by each variable within categories of the preceding variable on the Groups Based On list. For example, if you select *gender* as the first grouping variable and *minority* as the second grouping variable, cases will be grouped by minority classification within each gender category.

- You can specify up to eight grouping variables.
- Each eight bytes of a long string variable (string variables longer than eight bytes) counts as a variable toward the limit of eight grouping variables.
- Cases should be sorted by values of the grouping variables and in the same order that variables are listed in the Groups Based On list. If the data file isn't already sorted, select Sort the file by grouping variables.

Figure 9-10
Split File dialog box



Split File

Employee Code [id]
Date of Birth [bdate]
Educational Level (yea...
Employment Category [...
Current Salary [salary]
Beginning Salary [salb...
Months since Hire [jobti...
Previous Experience (...
Minority Classification [...

○ Analyze all cases, do not create groups
● Compare groups
○ Organize output by groups

Groups Based on:
Gender [gender]

● Sort the file by grouping variables
○ File is already sorted

Current Status: Analysis by groups is off.

Compare groups. Split-file groups are presented together for comparison purposes. For pivot tables, a single pivot table is created and each split-file variable can be moved between table dimensions. For charts, a separate chart is created for each split-file group and the charts are displayed together in the Viewer.

Organize output by groups. All results from each procedure are displayed separately for each split-file group.

To Split a Data File for Analysis

▶ From the menus choose:

Data
  Split File...

▶ Select Compare groups or Organize output by groups.

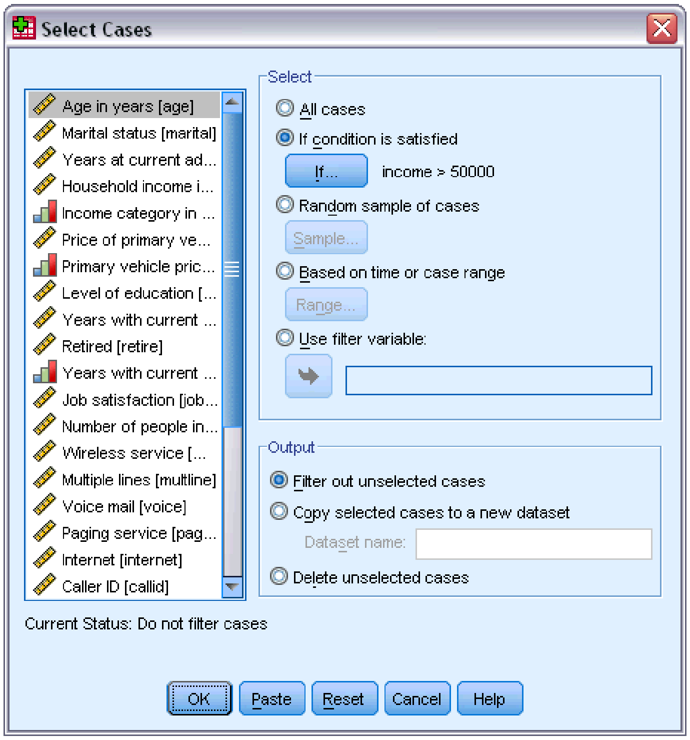
▶ Select one or more grouping variables.

Select Cases

Select Cases provides several methods for selecting a subgroup of cases based on criteria that include variables and complex expressions. You can also select a random sample of cases. The criteria used to define a subgroup can include:

- Variable values and ranges
- Date and time ranges
- Case (row) numbers
- Arithmetic expressions
- Logical expressions
- Functions

Figure 9-11
Select Cases dialog box



○ All cases
● If condition is satisfied
If... income > 50000
○ Random sample of cases
○ Based on time or case range
○ Use filter variable:

Output
● Filter out unselected cases
○ Copy selected cases to a new dataset
○ Delete unselected cases

Current Status: Do not filter cases

All cases. Turns case filtering off and uses all cases.

If condition is satisfied. Use a conditional expression to select cases. If the result of the conditional expression is true, the case is selected. If the result is false or missing, the case is not selected.

Random sample of cases. Selects a random sample based on an approximate percentage or an exact number of cases.

Based on time or case range. Selects cases based on a range of case numbers or a range of dates/times.

Use filter variable. Use the selected numeric variable from the data file as the filter variable. Cases with any value other than 0 or missing for the filter variable are selected.

Output

This section controls the treatment of unselected cases. You can choose one of the following alternatives for the treatment of unselected cases:

- **Filter out unselected cases.** Unselected cases are not included in the analysis but remain in the dataset. You can use the unselected cases later in the session if you turn filtering off. If you select a random sample or if you select cases based on a conditional expression, this generates a variable named *filter_$* with a value of 1 for selected cases and a value of 0 for unselected cases.

- **Copy selected cases to a new dataset.** Selected cases are copied to a new dataset, leaving the original dataset unaffected. Unselected cases are not included in the new dataset and are left in their original state in the original dataset.
- **Delete unselected cases.** Unselected cases are deleted from the dataset. Deleted cases can be recovered only by exiting from the file without saving any changes and then reopening the file. The deletion of cases is permanent if you save the changes to the data file.

Note: If you delete unselected cases and save the file, the cases cannot be recovered.

To Select a Subset of Cases

▶ From the menus choose:

Data
  Select Cases...

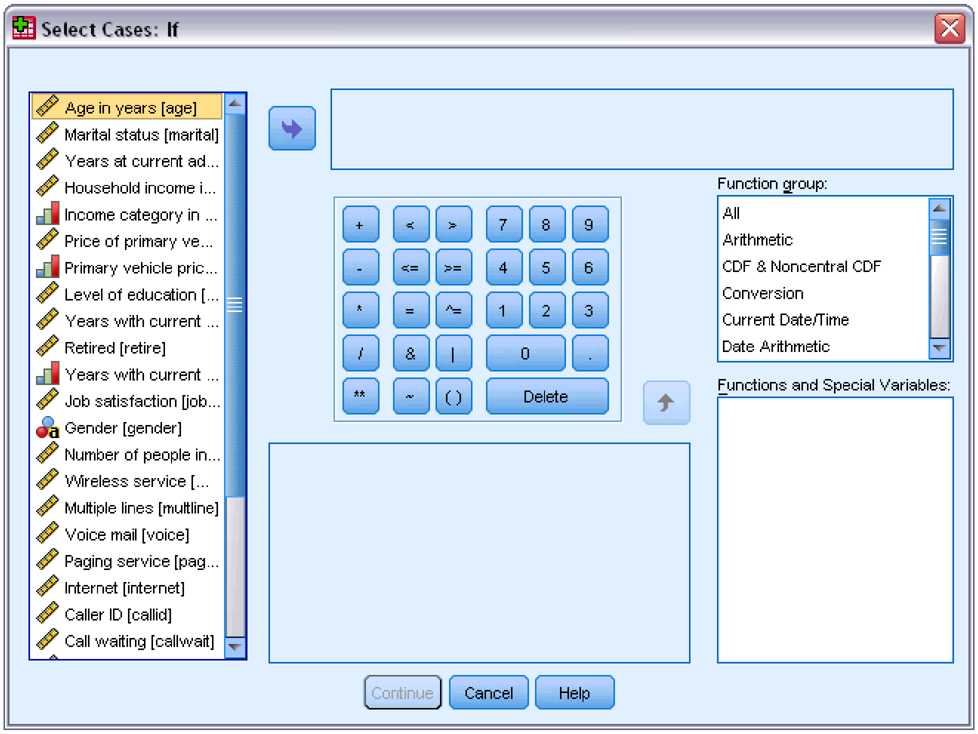
▶ Select one of the methods for selecting cases.

▶ Specify the criteria for selecting cases.

Select Cases: If

This dialog box allows you to select subsets of cases using conditional expressions. A conditional expression returns a value of *true*, *false*, or *missing* for each case.

Figure 9-12
Select Cases If dialog box



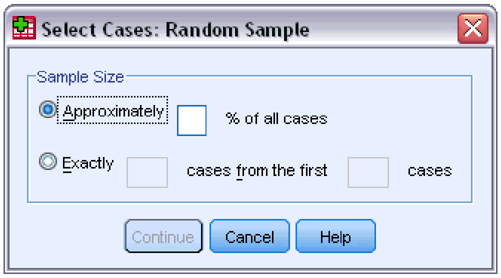
- If the result of a conditional expression is *true*, the case is included in the selected subset.

- If the result of a conditional expression is *false* or *missing*, the case is not included in the selected subset.
- Most conditional expressions use one or more of the six relational operators (<, >, <=, >=, =, and ~=) on the calculator pad.
- Conditional expressions can include variable names, constants, arithmetic operators, numeric (and other) functions, logical variables, and relational operators.

Select Cases: Random Sample

This dialog box allows you to select a random sample based on an approximate percentage or an exact number of cases. Sampling is performed without replacement; so, the same case cannot be selected more than once.

Figure 9-13
Select Cases Random Sample dialog box



Approximately. Generates a random sample of approximately the specified percentage of cases. Since this routine makes an independent pseudo-random decision for each case, the percentage of cases selected can only approximate the specified percentage. The more cases there are in the data file, the closer the percentage of cases selected is to the specified percentage.

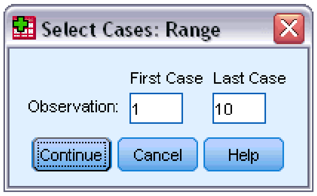
Exactly. A user-specified number of cases. You must also specify the number of cases from which to generate the sample. This second number should be less than or equal to the total number of cases in the data file. If the number exceeds the total number of cases in the data file, the sample will contain proportionally fewer cases than the requested number.

Select Cases: Range

This dialog box selects cases based on a range of case numbers or a range of dates or times.

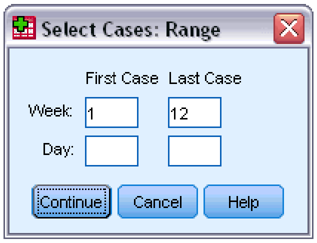
- Case ranges are based on row number as displayed in the Data Editor.
- Date and time ranges are available only for **time series data** with defined date variables (Data menu, Define Dates).

Figure 9-14
Select Cases Range dialog box for range of cases (no defined date variables)



	First Case	Last Case
Observation:	1	10

Figure 9-15
Select Cases Range dialog box for time series data with defined date variables



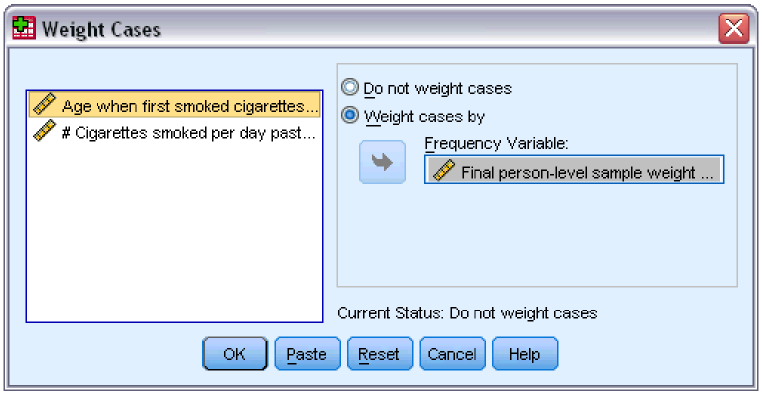
	First Case	Last Case
Week:	1	12
Day:		

Weight Cases

Weight Cases gives cases different weights (by simulated replication) for statistical analysis.

- The values of the weighting variable should indicate the number of observations represented by single cases in your data file.
- Cases with zero, negative, or missing values for the weighting variable are excluded from analysis.
- Fractional values are valid and some procedures, such as Frequencies, Crosstabs, and Custom Tables, will use fractional weight values. However, most procedures treat the weighting variable as a replication weight and will simply round fractional weights to the nearest integer. Some procedures ignore the weighting variable completely, and this limitation is noted in the procedure-specific documentation.

Figure 9-16
Weight Cases dialog box


Weight Cases
Age when first smoked cigarettes...
# Cigarettes smoked per day past...
OK
Paste
Reset
Cancel
Help

○ Do not weight cases
● Weight cases by
Frequency Variable: Final person-level sample weight ...

Current Status: Do not weight cases

Once you apply a weight variable, it remains in effect until you select another weight variable or turn off weighting. If you save a weighted data file, weighting information is saved with the data file. You can turn off weighting at any time, even after the file has been saved in weighted form.

Weights in Crosstabs. The Crosstabs procedure has several options for handling case weights.

Weights in scatterplots and histograms. Scatterplots and histograms have an option for turning case weights on and off, but this does not affect cases with a zero, negative, or missing value for the weight variable. These cases remain excluded from the chart even if you turn weighting off from within the chart.

To Weight Cases

▶ From the menus choose:

Data
  Weight Cases...

▶ Select Weight cases by.

▶ Select a frequency variable.

The values of the frequency variable are used as case weights. For example, a case with a value of 3 for the frequency variable will represent three cases in the weighted data file.

Restructuring Data

Use the Restructure Data Wizard to restructure your data for the procedure that you want to use. The wizard replaces the current file with a new, restructured file. The wizard can:

- Restructure selected variables into cases
- Restructure selected cases into variables
- Transpose all data

To Restructure Data

▶ From the menus choose:

Data
  Restructure...

▶ Select the type of restructuring that you want to do.

▶ Select the data to restructure.

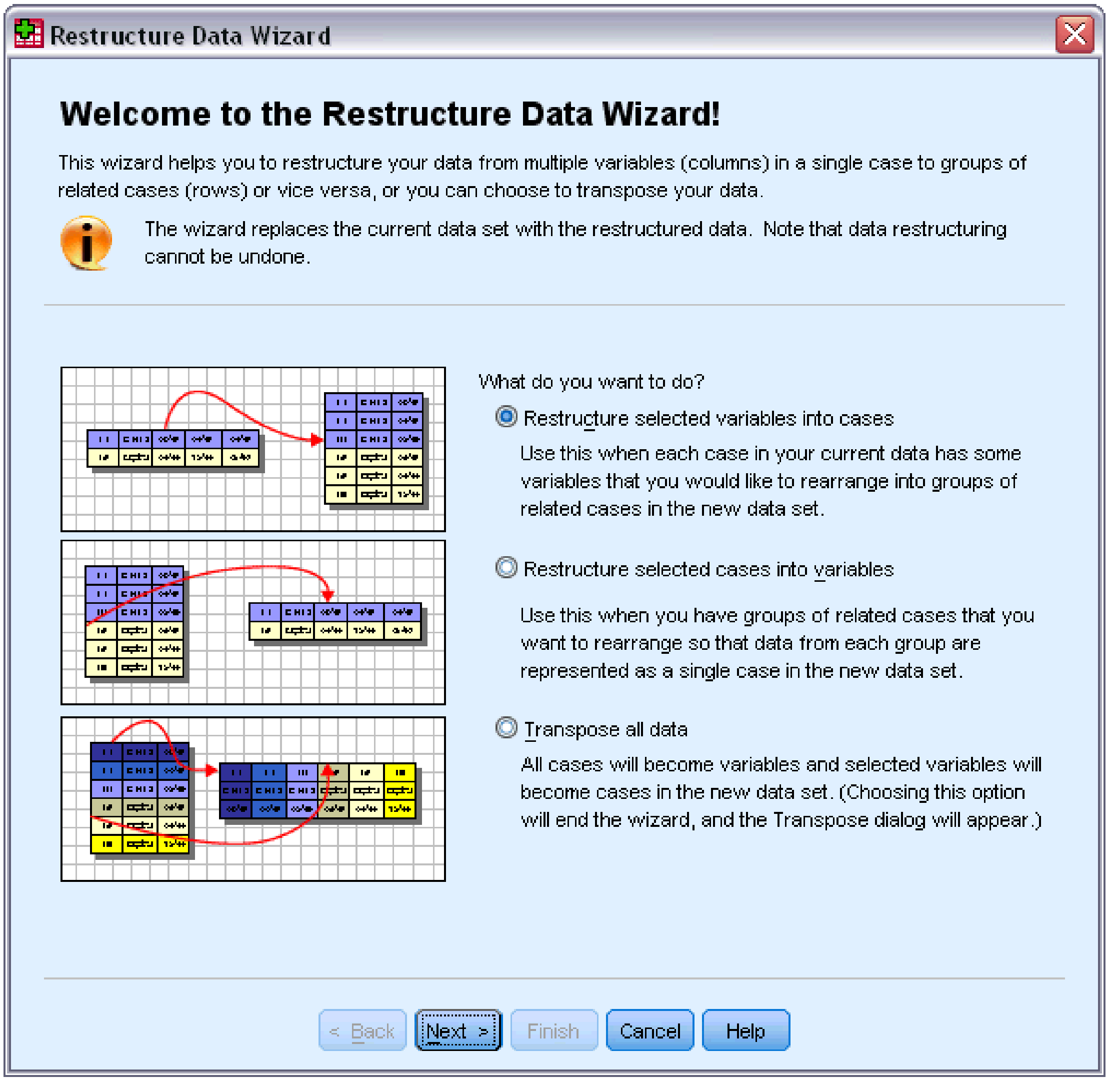
Optionally, you can:

- Create identification variables, which allow you to trace a value in the new file back to a value in the original file
- Sort the data prior to restructuring
- Define options for the new file
- Paste the command syntax into a syntax window

Restructure Data Wizard: Select Type

Use the Restructure Data Wizard to restructure your data. In the first dialog box, select the type of restructuring that you want to do.

Figure 9-17
Restructure Data Wizard



Restructure Data Wizard

Welcome to the Restructure Data Wizard!

This wizard helps you to restructure your data from multiple variables (columns) in a single case to groups of related cases (rows) or vice versa, or you can choose to transpose your data.

The wizard replaces the current data set with the restructured data. Note that data restructuring cannot be undone.

What do you want to do?

● Restructure selected variables into cases

Use this when each case in your current data has some variables that you would like to rearrange into groups of related cases in the new data set.

○ Restructure selected cases into variables

Use this when you have groups of related cases that you want to rearrange so that data from each group are represented as a single case in the new data set.

○ Transpose all data

All cases will become variables and selected variables will become cases in the new data set. (Choosing this option will end the wizard, and the Transpose dialog will appear.)

- **Restructure selected variables into cases.** Choose this when you have groups of related columns in your data and you want them to appear in groups of rows in the new data file. If you choose this, the wizard will display the steps for **Variables to Cases**.
- **Restructure selected cases into variables.** Choose this when you have groups of related rows in your data and you want them to appear in groups of columns in the new data file. If you choose this, the wizard will display the steps for **Cases to Variables**.
- **Transpose all data.** Choose this when you want to transpose your data. All rows will become columns and all columns will become rows in the new data. This choice closes the Restructure Data Wizard and opens the Transpose Data dialog box.

Deciding How to Restructure the Data

A **variable** contains information that you want to analyze—for example, a measurement or a score. A **case** is an observation—for example, an individual. In a *simple* data structure, each variable is a single column in your data and each case is a single row. So, for example, if you were measuring test scores for all students in a class, all score values would appear in only one column, and there would be a row for each student.

When you analyze data, you are often analyzing how a variable varies according to some condition. The condition can be a specific experimental treatment, a demographic, a point in time, or something else. In data analysis, conditions of interest are often referred to as **factors**. When you analyze factors, you have a *complex* data structure. You may have information about a variable in more than one column in your data (for example, a column for each level of a factor), or you may have information about a case in more than one row (for example, a row for each level of a factor). The Restructure Data Wizard helps you to restructure files with a complex data structure.

The structure of the current file and the structure that you want in the new file determine the choices that you make in the wizard.

How are the data arranged in the current file? The current data may be arranged so that factors are recorded in a *separate* variable (in groups of cases) or *with* the variable (in groups of variables).

- **Groups of cases.** Does the current file have variables and conditions recorded in separate columns? For example:

var	factor
8	1
9	1
3	2
1	2

In this example, the first two rows are a **case group** because they are related. They contain data for the same factor level. In PASW Statistics data analysis, the factor is often referred to as a **grouping variable** when the data are structured this way.

- **Groups of columns.** Does the current file have variables and conditions recorded in the same column? For example:

var_1	var_2
8	3
9	1

In this example, the two columns are a **variable group** because they are related. They contain data for the same variable—*var_1* for factor level 1 and *var_2* for factor level 2. In PASW Statistics data analysis, the factor is often referred to as a **repeated measure** when the data are structured this way.

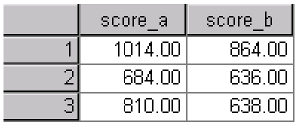
How should the data be arranged in the new file? This is usually determined by the procedure that you want to use to analyze your data.

- **Procedures that require groups of cases.** Your data must be structured in case groups to do analyses that require a grouping variable. Examples are *univariate*, *multivariate*, and *variance components* with General Linear Model, Mixed Models, and OLAP Cubes and *independent samples* with T Test or Nonparametric Tests. If your current data structure is variable groups and you want to do these analyses, select Restructure selected variables into cases.
- **Procedures that require groups of variables.** Your data must be structured in variable groups to analyze repeated measures. Examples are *repeated measures* with General Linear Model, *time-dependent covariate* analysis with Cox Regression Analysis, *paired samples* with T Test, or *related samples* with Nonparametric Tests. If your current data structure is case groups and you want to do these analyses, select Restructure selected cases into variables.

Example of Variables to Cases

In this example, test scores are recorded in separate columns for each factor, *A* and *B*.

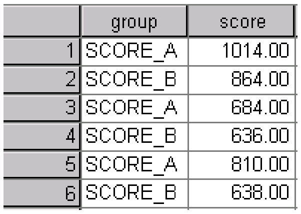
Figure 9-18
Current data for variables to cases



	score_a	score_b
1	1014.00	864.00
2	684.00	636.00
3	810.00	638.00

You want to do an independent-samples *t* test. You have a column group consisting of *score_a* and *score_b*, but you don't have the **grouping variable** that the procedure requires. Select Restructure selected variables into cases in the Restructure Data Wizard, restructure one variable group into a new variable named *score*, and create an index named *group*. The new data file is shown in the following figure.

Figure 9-19
New, restructured data for variables to cases



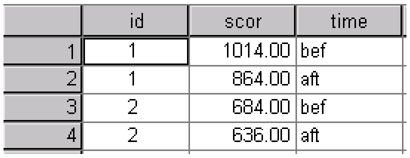
	group	score
1	SCORE_A	1014.00
2	SCORE_B	864.00
3	SCORE_A	684.00
4	SCORE_B	636.00
5	SCORE_A	810.00
6	SCORE_B	638.00

When you run the independent-samples *t* test, you can now use *group* as the grouping variable.

Example of Cases to Variables

In this example, test scores are recorded twice for each subject—before and after a treatment.

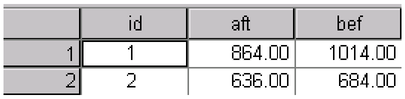
Figure 9-20
Current data for cases to variables



	id	scor	time
1	1	1014.00	bef
2	1	864.00	aft
3	2	684.00	bef
4	2	636.00	aft

You want to do a paired-samples *t* test. Your data structure is case groups, but you don't have the **repeated measures** for the paired variables that the procedure requires. Select Restructure selected cases into variables in the Restructure Data Wizard, use *id* to identify the row groups in the current data, and use *time* to create the variable group in the new file.

Figure 9-21
New, restructured data for cases to variables



	id	aft	bef
1	1	864.00	1014.00
2	2	636.00	684.00

When you run the paired-samples *t* test, you can now use *bef* and *aft* as the variable pair.

Restructure Data Wizard (Variables to Cases): Number of Variable Groups

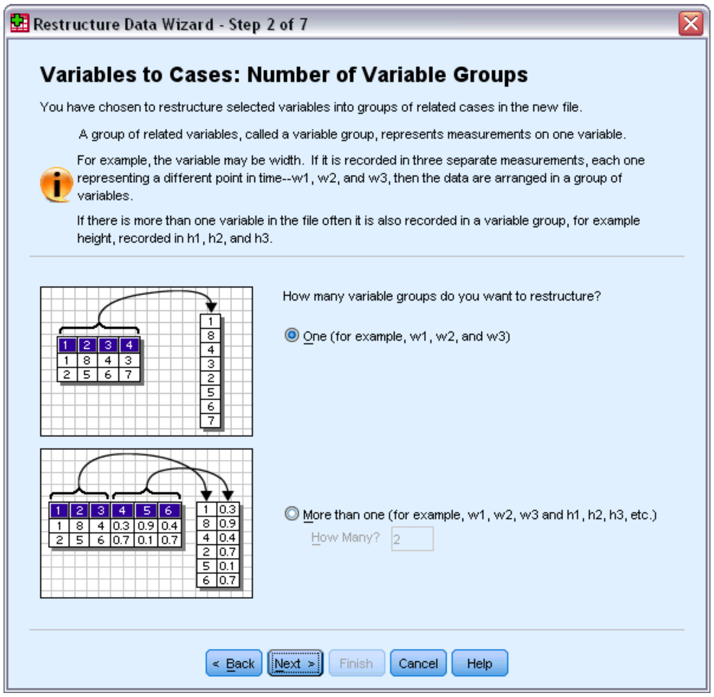
Note: The wizard presents this step if you choose to restructure variable groups into rows.

In this step, decide how many variable groups in the current file that you want to restructure in the new file.

How many variable groups are in the current file? Think about how many variable groups exist in the current data. A group of related columns, called a **variable group**, records repeated measures of the same variable in separate columns. For example, if you have three columns in the current data—*w1*, *w2*, and *w3*—that record **width**, you have one variable group. If you have an additional three columns—*h1*, *h2*, and *h3*—that record **height**, you have two variable groups.

How many variable groups should be in the new file? Consider how many variable groups you want to have represented in the new data file. You do not have to restructure all variable groups into the new file.

Figure 9-22
Restructure Data Wizard: Number of Variable Groups, Step 2



Restructure Data Wizard - Step 2 of 7

Variables to Cases: Number of Variable Groups

You have chosen to restructure selected variables into groups of related cases in the new file.

A group of related variables, called a variable group, represents measurements on one variable.

For example, the variable may be width. If it is recorded in three separate measurements, each one representing a different point in time--w1, w2, and w3, then the data are arranged in a group of variables.

If there is more than one variable in the file often it is also recorded in a variable group, for example height, recorded in h1, h2, and h3.

How many variable groups do you want to restructure?

● One (for example, w1, w2, and w3)

○ More than one (for example, w1, w2, w3 and h1, h2, h3, etc.)

How Many? 2

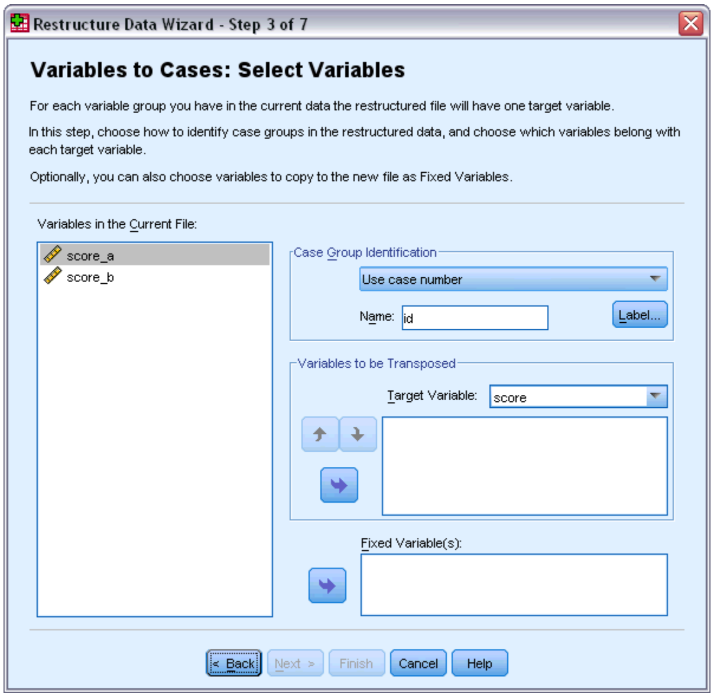
- **One.** The wizard will create a single restructured variable in the new file from one variable group in the current file.
- **More than one.** The wizard will create multiple restructured variables in the new file. The number that you specify affects the next step, in which the wizard automatically creates the specified number of new variables.

Restructure Data Wizard (Variables to Cases): Select Variables

Note: The wizard presents this step if you choose to restructure variable groups into rows.

In this step, provide information about how the variables in the current file should be used in the new file. You can also create a variable that identifies the rows in the new file.

Figure 9-23
Restructure Data Wizard: Select Variables, Step 3



Restructure Data Wizard - Step 3 of 7

Variables to Cases: Select Variables

For each variable group you have in the current data the restructured file will have one target variable.

In this step, choose how to identify case groups in the restructured data, and choose which variables belong with each target variable.

Optionally, you can also choose variables to copy to the new file as Fixed Variables.

Variables in the Current File:
score_a
score_b

Case Group Identification: Use case number
Name: id

Variables to be Transposed
Target Variable: score

Fixed Variable(s):

How should the new rows be identified? You can create a variable in the new data file that identifies the row in the current data file that was used to create a group of new rows. The identifier can be a sequential case number or it can be the values of the variable. Use the controls in Case Group Identification to define the identification variable in the new file. Click a cell to change the default variable name and provide a descriptive variable label for the identification variable.

What should be restructured in the new file? In the previous step, you told the wizard how many variable groups you want to restructure. The wizard created one new variable for each group. The values for the variable group will appear in that variable in the new file. Use the controls in Variable to be Transposed to define the restructured variable in the new file.

To Specify One Restructured Variable

▶ Put the variables that make up the variable group that you want to transform into the Variable to be Transposed list. All of the variables in the group must be of the same type (numeric or string).

You can include the same variable more than once in the variable group (variables are copied rather than moved from the source variable list); its values are repeated in the new file.

To Specify Multiple Restructured Variables

▶ Select the first target variable that you want to define from the Target Variable drop-down list.

▶ Put the variables that make up the variable group that you want to transform into the Variable to be Transposed list. All of the variables in the group must be of the same type (numeric or string).

You can include the same variable more than once in the variable group. (A variable is copied rather than moved from the source variable list, and its values are repeated in the new file.)

▶ Select the next target variable that you want to define, and repeat the variable selection process for all available target variables.

- Although you can include the same variable more than once in the same target variable group, you cannot include the same variable in more than one target variable group.
- Each target variable group list must contain the same number of variables. (Variables that are listed more than once are included in the count.)
- The number of target variable groups is determined by the number of variable groups that you specified in the previous step. You can change the default variable names here, but you must return to the previous step to change the number of variable groups to restructure.
- You must define variable groups (by selecting variables in the source list) for all available target variables before you can proceed to the next step.

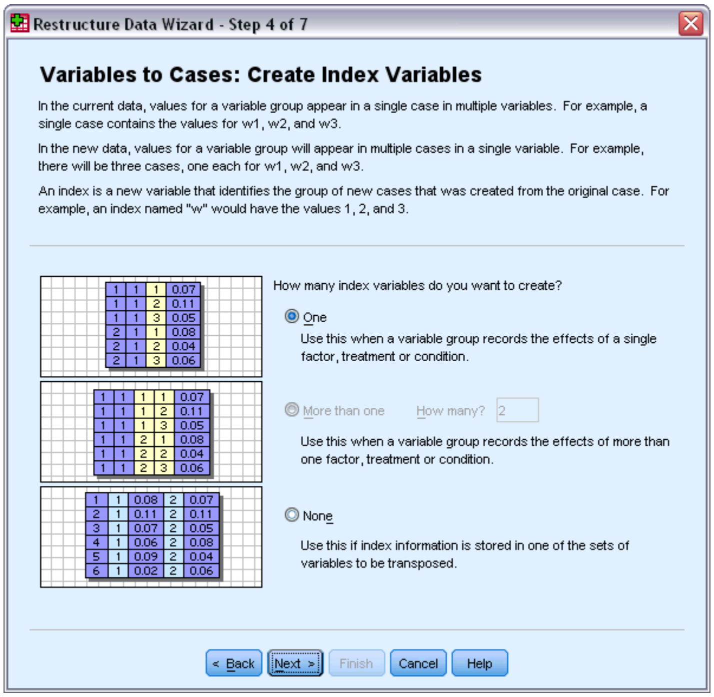
What should be copied into the new file? Variables that aren't restructured can be copied into the new file. Their values will be propagated in the new rows. Move variables that you want to copy into the new file into the Fixed Variable(s) list.

Restructure Data Wizard (Variables to Cases): Create Index Variables

Note: The wizard presents this step if you choose to restructure variable groups into rows.

In this step, decide whether to create index variables. An index is a new variable that sequentially identifies a row group based on the original variable from which the new row was created.

Figure 9-24
Restructure Data Wizard: Create Index Variables, Step 4


Restructure Data Wizard - Step 4 of 7

Variables to Cases: Create Index Variables

In the current data, values for a variable group appear in a single case in multiple variables. For example, a single case contains the values for w1, w2, and w3.

In the new data, values for a variable group will appear in multiple cases in a single variable. For example, there will be three cases, one each for w1, w2, and w3.

An index is a new variable that identifies the group of new cases that was created from the original case. For example, an index named "w" would have the values 1, 2, and 3.

1	1	1	0.07
1	1	2	0.11
1	1	3	0.05
2	1	1	0.08
2	1	2	0.04
2	1	3	0.06

1	1	1	1	0.07
1	1	1	2	0.11
1	1	1	3	0.05
1	1	2	1	0.08
1	1	2	2	0.04
1	1	2	3	0.06

1	1	0.08	2	0.07
2	1	0.11	2	0.11
3	1	0.07	2	0.05
4	1	0.06	2	0.08
5	1	0.09	2	0.04
6	1	0.02	2	0.06

How many index variables do you want to create?

● One
Use this when a variable group records the effects of a single factor, treatment or condition.

○ More than one   How many? 2
Use this when a variable group records the effects of more than one factor, treatment or condition.

○ None
Use this if index information is stored in one of the sets of variables to be transposed.

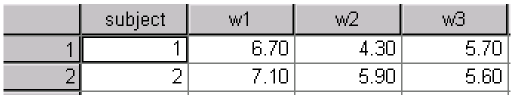
How many index variables should be in the new file? Index variables can be used as grouping variables in procedures. In most cases, a single index variable is sufficient; however, if the variable groups in your current file reflect multiple factor levels, multiple indices may be appropriate.

- **One.** The wizard will create a single index variable.
- **More than one.** The wizard will create multiple indices and enter the number of indices that you want to create. The number that you specify affects the next step, in which the wizard automatically creates the specified number of indices.
- **None.** Select this if you do not want to create index variables in the new file.

Example of One Index for Variables to Cases

In the current data, there is one variable group, *width*, and one factor, *time*. Width was measured three times and recorded in *w1*, *w2*, and *w3*.

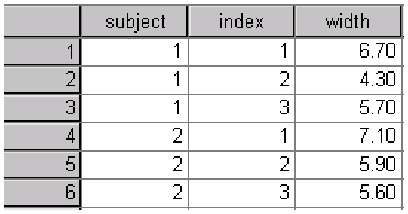
Figure 9-25
Current data for one index



	subject	w1	w2	w3
1	1	6.70	4.30	5.70
2	2	7.10	5.90	5.60

We'll restructure the variable group into a single variable, *width*, and create a single numeric index. The new data are shown in the following table.

Figure 9-26
New, restructured data with one index



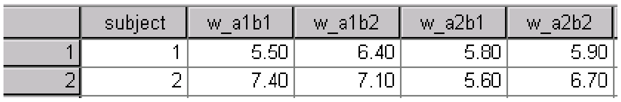
	subject	index	width
1	1	1	6.70
2	1	2	4.30
3	1	3	5.70
4	2	1	7.10
5	2	2	5.90
6	2	3	5.60

Index starts with 1 and increments for each variable in the group. It restarts each time a new row is encountered in the original file. We can now use *index* in procedures that require a grouping variable.

Example of Two Indices for Variables to Cases

When a variable group records more than one factor, you can create more than one index; however, the current data must be arranged so that the levels of the first factor are a primary index within which the levels of subsequent factors cycle. In the current data, there is one variable group, *width*, and two factors, *A* and *B*. The data are arranged so that levels of factor *B* cycle within levels of factor *A*.

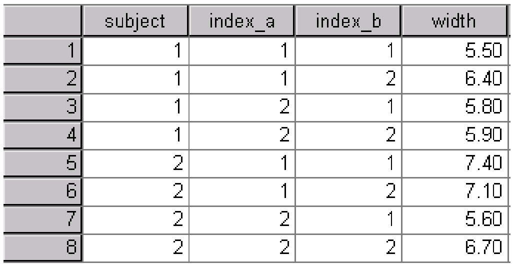
Figure 9-27
Current data for two indices



	subject	w_a1b1	w_a1b2	w_a2b1	w_a2b2
1	1	5.50	6.40	5.80	5.90
2	2	7.40	7.10	5.60	6.70

We'll restructure the variable group into a single variable, *width*, and create two indices. The new data are shown in the following table.

Figure 9-28
New, restructured data with two indices



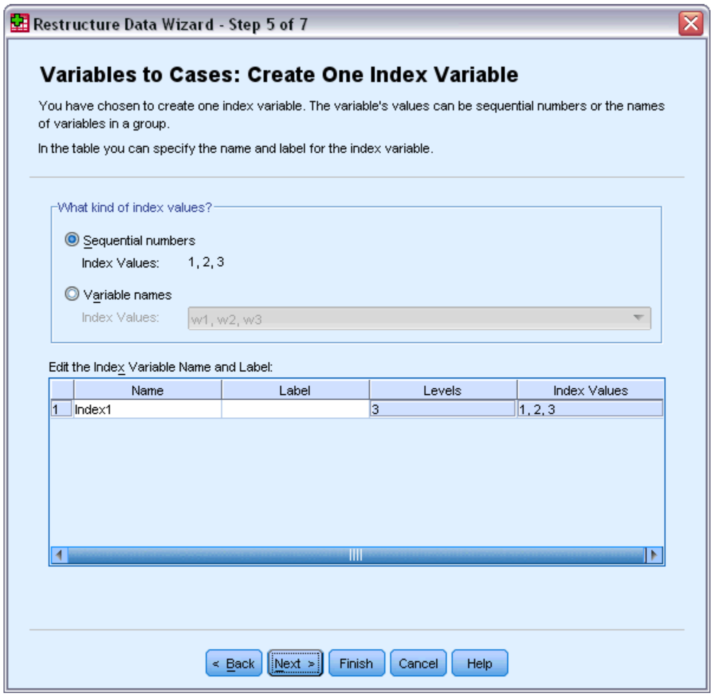
	subject	index_a	index_b	width
1	1	1	1	5.50
2	1	1	2	6.40
3	1	2	1	5.80
4	1	2	2	5.90
5	2	1	1	7.40
6	2	1	2	7.10
7	2	2	1	5.60
8	2	2	2	6.70

Restructure Data Wizard (Variables to Cases): Create One Index Variable

Note: The wizard presents this step if you choose to restructure variable groups into rows and create one index variable.

In this step, decide what values you want for the index variable. The values can be sequential numbers or the names of the variables in an original variable group. You can also specify a name and a label for the new index variable.

Figure 9-29
Restructure Data Wizard: Create One Index Variable, Step 5



Variables to Cases: Create One Index Variable

You have chosen to create one index variable. The variable's values can be sequential numbers or the names of variables in a group.

In the table you can specify the name and label for the index variable.

What kind of index values?

● Sequential numbers
Index Values: 1, 2, 3

○ Variable names
Index Values: w1, w2, w3

Edit the Index Variable Name and Label:

	Name	Label	Levels	Index Values
1	Index1		3	1, 2, 3

For more information, see the topic Example of One Index for Variables to Cases on p. 195.

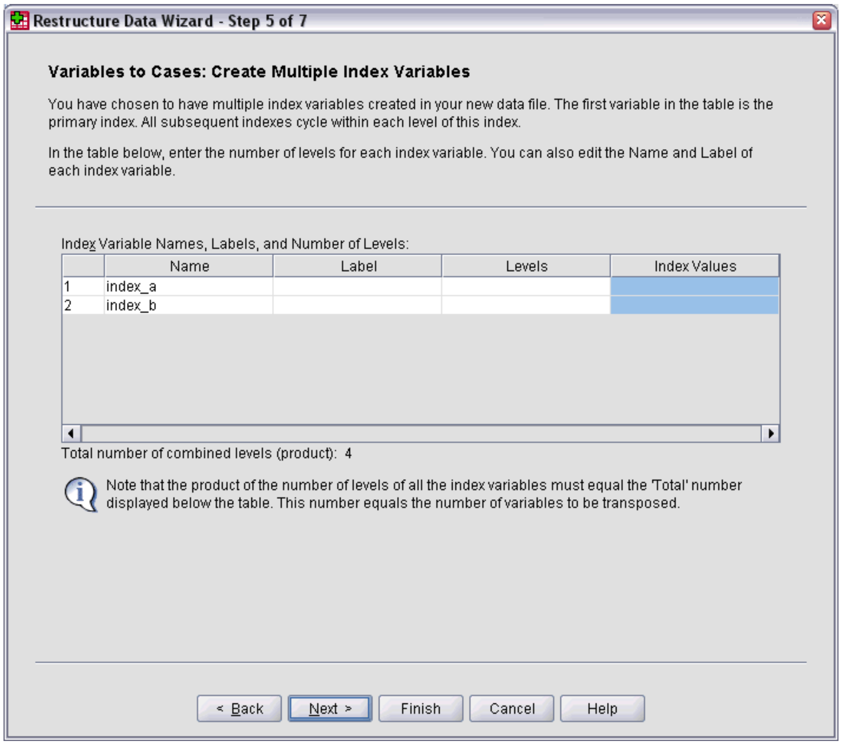
- **Sequential numbers.** The wizard will automatically assign sequential numbers as index values.
- **Variable names.** The wizard will use the names of the selected variable group as index values. Choose a variable group from the list.
- **Names and labels.** Click a cell to change the default variable name and provide a descriptive variable label for the index variable.

Restructure Data Wizard (Variables to Cases): Create Multiple Index Variables

Note: The wizard presents this step if you choose to restructure variable groups into rows and create multiple index variables.

In this step, specify the number of levels for each index variable. You can also specify a name and a label for the new index variable.

Figure 9-30
Restructure Data Wizard: Create Multiple Index Variables, Step 5



Variables to Cases: Create Multiple Index Variables

You have chosen to have multiple index variables created in your new data file. The first variable in the table is the primary index. All subsequent indexes cycle within each level of this index.

In the table below, enter the number of levels for each index variable. You can also edit the Name and Label of each index variable.

Index Variable Names, Labels, and Number of Levels:

	Name	Label	Levels	Index Values
1	index_a			
2	index_b			

Total number of combined levels (product): 4

Note that the product of the number of levels of all the index variables must equal the 'Total' number displayed below the table. This number equals the number of variables to be transposed.

For more information, see the topic Example of Two Indices for Variables to Cases on p. 196.

How many levels are recorded in the current file? Consider how many factor levels are recorded in the current data. A **level** defines a group of cases that experienced identical conditions. If there are multiple factors, the current data must be arranged so that the levels of the first factor are a primary index within which the levels of subsequent factors cycle.

How many levels should be in the new file? Enter the number of levels for each index. The values for multiple index variables are always sequential numbers. The values start at 1 and increment for each level. The first index increments the slowest, and the last index increments the fastest.

Total combined levels. You cannot create more levels than exist in the current data. Because the restructured data will contain one row for each combination of treatments, the wizard checks the number of levels that you create. It will compare the product of the levels that you create to the number of variables in your variable groups. They must match.

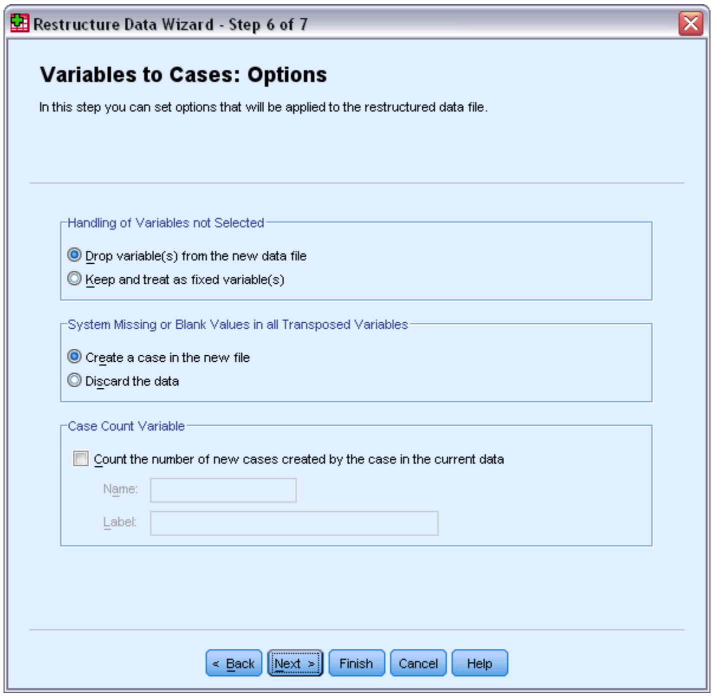
Names and labels. Click a cell to change the default variable name and provide a descriptive variable label for the index variables.

Restructure Data Wizard (Variables to Cases): Options

Note: The wizard presents this step if you choose to restructure variable groups into rows.

In this step, specify options for the new, restructured file.

Figure 9-31
Restructure Data Wizard: Options, Step 6



Variables to Cases: Options

In this step you can set options that will be applied to the restructured data file.

Handling of Variables not Selected
● Drop variable(s) from the new data file
○ Keep and treat as fixed variable(s)

System Missing or Blank Values in all Transposed Variables
● Create a case in the new file
○ Discard the data

Case Count Variable
□ Count the number of new cases created by the case in the current data
Name:
Label:

Drop unselected variables? In the Select Variables step (step 3), you selected variable groups to be restructured, variables to be copied, and an identification variable from the current data. The data from the selected variables will appear in the new file. If there are other variables in the current data, you can choose to discard or keep them.

Keep missing data? The wizard checks each potential new row for null values. A **null value** is a system-missing or blank value. You can choose to keep or discard rows that contain only null values.

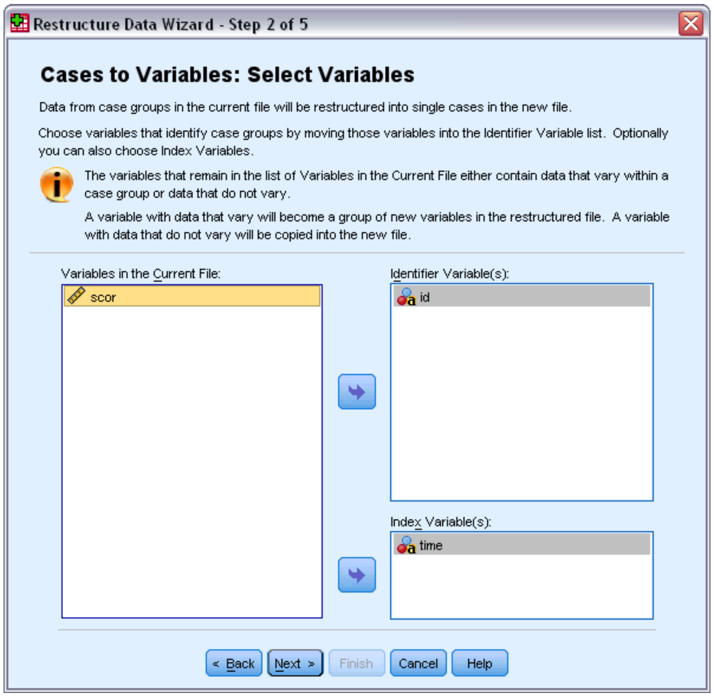
Create a count variable? The wizard can create a **count variable** in the new file. It contains the number of new rows generated by a row in the current data. A count variable may be useful if you choose to discard null values from the new file because that makes it possible to generate a different number of new rows for a given row in the current data. Click a cell to change the default variable name and provide a descriptive variable label for the count variable.

Restructure Data Wizard (Cases to Variables): Select Variables

Note: The wizard presents this step if you choose to restructure case groups into columns.

In this step, provide information about how the variables in the current file should be used in the new file.

Figure 9-32
Restructure Data Wizard: Select Variables, Step 2



Cases to Variables: Select Variables

Data from case groups in the current file will be restructured into single cases in the new file.

Choose variables that identify case groups by moving those variables into the Identifier Variable list. Optionally you can also choose Index Variables.

The variables that remain in the list of Variables in the Current File either contain data that vary within a case group or data that do not vary.

A variable with data that vary will become a group of new variables in the restructured file. A variable with data that do not vary will be copied into the new file.

Variables in the Current File: scor

Identifier Variable(s): id

Index Variable(s): time

What identifies case groups in the current data? A **case group** is a group of rows that are related because they measure the same observational unit—for example, an individual or an institution. The wizard needs to know which variables in the current file identify the case groups so that it can consolidate each group into a single row in the new file. Move variables that identify case groups in the current file into the Identifier Variable(s) list. Variables that are used to split the current data file are automatically used to identify case groups. Each time a new combination of identification values is encountered, the wizard will create a new row, so cases in the current file should be sorted by values of the identification variables in the same order that variables are listed in the Identifier Variable(s) list. If the current data file isn't already sorted, you can sort it in the next step.

How should the new variable groups be created in the new file? In the original data, a variable appears in a single column. In the new data file, that variable will appear in multiple new columns. **Index variables** are variables in the current data that the wizard should use to create the new columns. The restructured data will contain one new variable for each unique value in these columns. Move the variables that should be used to form the new variable groups to the Index Variable(s) list. When the wizard presents options, you can also choose to order the new columns by index.

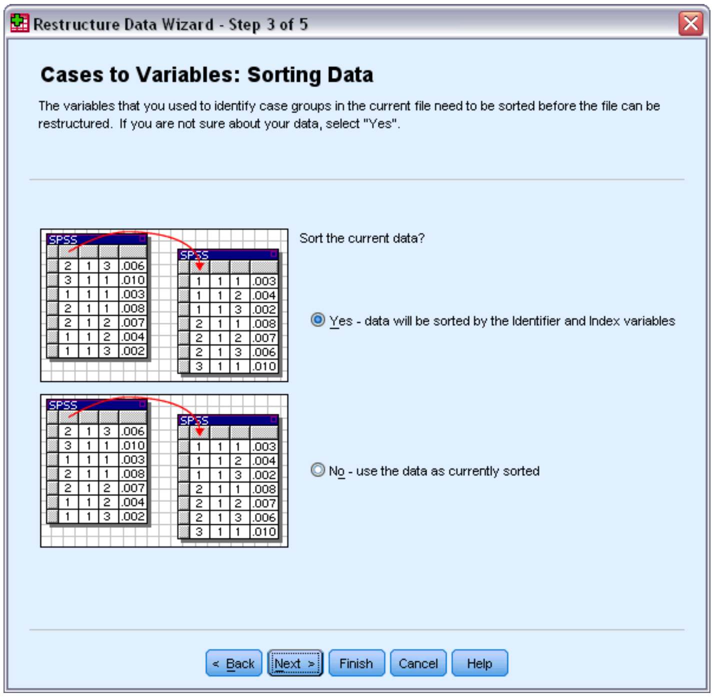
What happens to the other columns? The wizard automatically decides what to do with the variables that remain in the Current File list. It checks each variable to see if the data values vary within a case group. If they do, the wizard restructures the values into a variable group in the new file. If they don't, the wizard copies the values into the new file.

Restructure Data Wizard (Cases to Variables): Sort Data

Note: The wizard presents this step if you choose to restructure case groups into columns.

In this step, decide whether to sort the current file before restructuring it. Each time the wizard encounters a new combination of identification values, a new row is created, so it is important that the data are sorted by the variables that identify case groups.

Figure 9-33
Restructure Data Wizard: Sorting Data, Step 3



Cases to Variables: Sorting Data

The variables that you used to identify case groups in the current file need to be sorted before the file can be restructured. If you are not sure about your data, select "Yes".

2	1	3	.006
3	1	1	.010
1	1	1	.003
2	1	1	.008
2	1	2	.007
1	1	2	.004
1	1	3	.002

1	1	1	.003
1	1	2	.004
1	1	3	.002
2	1	1	.008
2	1	2	.007
2	1	3	.006
3	1	1	.010

Sort the current data?

● Yes - data will be sorted by the Identifier and Index variables

2	1	3	.006
3	1	1	.010
1	1	1	.003
2	1	1	.008
2	1	2	.007
1	1	2	.004
1	1	3	.002

1	1	1	.003
1	1	2	.004
1	1	3	.002
2	1	1	.008
2	1	2	.007
2	1	3	.006
3	1	1	.010

○ No - use the data as currently sorted

How are the rows ordered in the current file? Consider how the current data are sorted and which variables you are using to identify case groups (specified in the previous step).

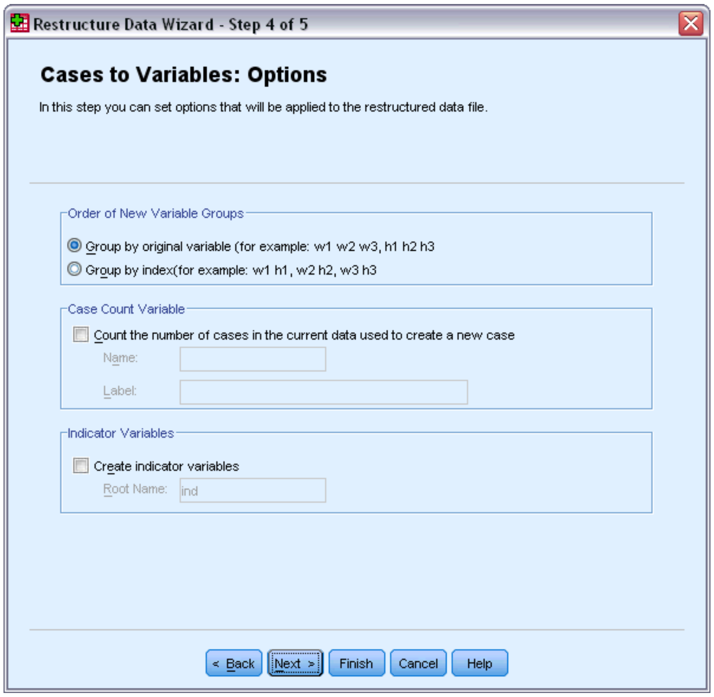
- **Yes.** The wizard will automatically sort the current data by the identification variables in the same order that variables are listed in the Identifier Variable(s) list in the previous step. Choose this when the data aren't sorted by the identification variables or when you aren't sure. This choice requires a separate pass of the data, but it guarantees that the rows are correctly ordered for restructuring.
- **No.** The wizard will not sort the current data. Choose this when you are sure that the current data are sorted by the variables that identify case groups.

Restructure Data Wizard (Cases to Variables): Options

Note: The wizard presents this step if you choose to restructure case groups into columns.

In this step, specify options for the new, restructured file.

Figure 9-34
Restructure Data Wizard: Options, Step 4



Restructure Data Wizard - Step 4 of 5

Cases to Variables: Options

In this step you can set options that will be applied to the restructured data file.

Order of New Variable Groups

● Group by original variable (for example: w1 w2 w3, h1 h2 h3

○ Group by index(for example: w1 h1, w2 h2, w3 h3

Case Count Variable

□ Count the number of cases in the current data used to create a new case

Name:

Label:

Indicator Variables

□ Create indicator variables

Root Name: ind

How should the new variable groups be ordered in the new file?

- **By variable.** The wizard groups the new variables created from an original variable together.
- **By index.** The wizard groups the variables according to the values of the index variables.

Example. The variables to be restructured are *w* and *h*, and the index is *month*:

w	h	month

Grouping by variable results in:

w.jan	w.feb	h.jan

Grouping by index results in:

w.jan	h.jan	w.feb

Create a count variable? The wizard can create a count variable in the new file. It contains the number of rows in the current data that were used to create a row in the new data file.

Create indicator variables? The wizard can use the index variables to create **indicator variables** in the new data file. It creates one new variable for each unique value of the index variable. The indicator variables signal the presence or absence of a value for a case. An indicator variable has the value of 1 if the case has a value; otherwise, it is 0.

Example. The index variable is *product*. It records the products that a customer purchased. The original data are:

customer	product
1	chick
1	eggs
2	eggs
3	chick

Creating an indicator variable results in one new variable for each unique value of *product*. The restructured data are:

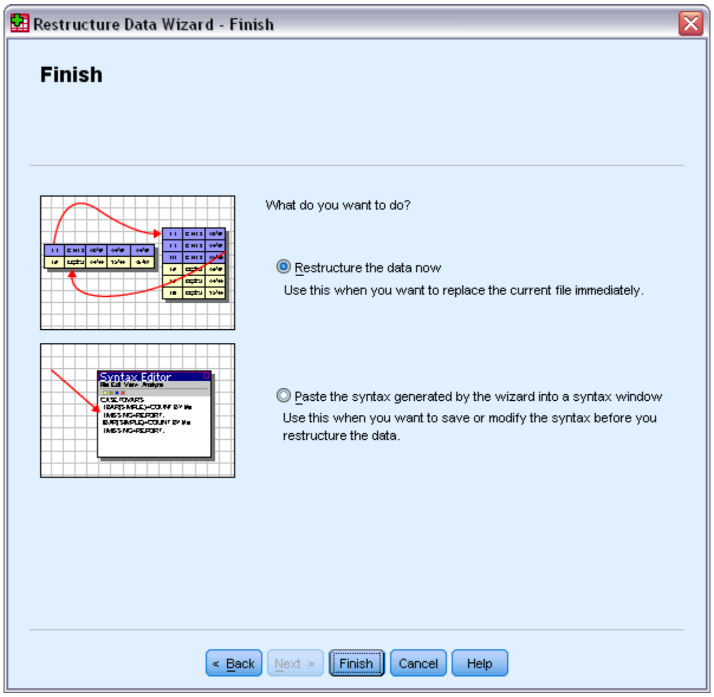
customer	indchick	indeggs
1	1	1
2	0	1
3	1	0

In this example, the restructured data could be used to get frequency counts of the products that customers buy.

Restructure Data Wizard: Finish

This is the final step of the Restructure Data Wizard. Decide what to do with your specifications.

Figure 9-35
Restructure Data Wizard: Finish



Restructure Data Wizard - Finish

Finish

What do you want to do?

● Restructure the data now
Use this when you want to replace the current file immediately.

○ Paste the syntax generated by the wizard into a syntax window
Use this when you want to save or modify the syntax before you restructure the data.

- **Restructure now.** The wizard will create the new, restructured file. Choose this if you want to replace the current file immediately.

Note: If original data are weighted, the new data will be weighted unless the variable that is used as the weight is restructured or dropped from the new file.

- **Paste syntax**. The wizard will paste the syntax it generates into a syntax window. Choose this when you are not ready to replace the current file, when you want to modify the syntax, or when you want to save it for future use.

Chapter

10

Working with Output

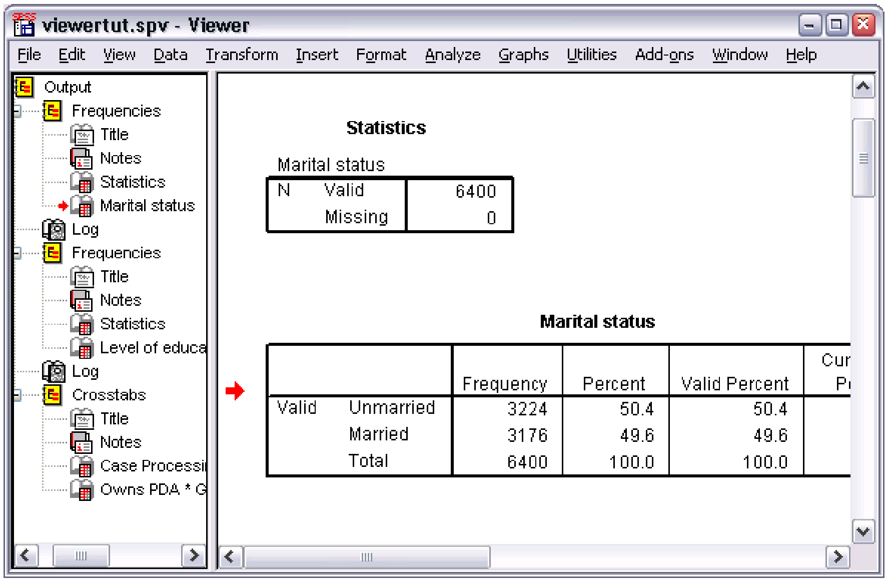
When you run a procedure, the results are displayed in a window called the Viewer. In this window, you can easily navigate to the output that you want to see. You can also manipulate the output and create a document that contains precisely the output that you want.

Viewer

Results are displayed in the Viewer. You can use the Viewer to:

- Browse results
- Show or hide selected tables and charts
- Change the display order of results by moving selected items
- Move items between the Viewer and other applications

Figure 10-1
Viewer



Statistics

Marital status

N	Valid	6400
	Missing	0

Marital status

		Frequency	Percent	Valid Percent	Cur P
Valid	Unmarried	3224	50.4	50.4	
	Married	3176	49.6	49.6	
	Total	6400	100.0	100.0	

The Viewer is divided into two panes:

- The left pane contains an outline view of the contents.
- The right pane contains statistical tables, charts, and text output.

You can click an item in the outline to go directly to the corresponding table or chart. You can click and drag the right border of the outline pane to change the width of the outline pane.

Showing and Hiding Results

In the Viewer, you can selectively show and hide individual tables or results from an entire procedure. This process is useful when you want to shorten the amount of visible output in the contents pane.

To Hide Tables and Charts

▶ Double-click the item's book icon in the outline pane of the Viewer.

or

▶ Click the item to select it.

▶ From the menus choose:

View
  Hide

or

▶ Click the closed book (Hide) icon on the Outlining toolbar.

The open book (Show) icon becomes the active icon, indicating that the item is now hidden.

To Hide Procedure Results

▶ Click the box to the left of the procedure name in the outline pane.

This hides all results from the procedure and collapses the outline view.

Moving, Deleting, and Copying Output

You can rearrange the results by copying, moving, or deleting an item or a group of items.

To Move Output in the Viewer

▶ Select the items in the outline or contents pane.

▶ Drag and drop the selected items into a different location.

To Delete Output in the Viewer

▶ Select the items in the outline or contents pane.

▶ Press the Delete key.

or

▶ From the menus choose:

Edit
  Delete

Changing Initial Alignment

By default, all results are initially left-aligned. To change the initial alignment of new output items:

▶ From the menus choose:

Edit
  Options

▶ Click the Viewer tab.

▶ In the Initial Output State group, select the item type (for example, pivot table, chart, text output).

▶ Select the alignment option you want.

Changing Alignment of Output Items

▶ In the outline or contents pane, select the items that you want to align.

▶ From the menus choose:

Format
  Align Left

or

Format
  Center

or

Format
  Align Right

Viewer Outline

The outline pane provides a table of contents of the Viewer document. You can use the outline pane to navigate through your results and control the display. Most actions in the outline pane have a corresponding effect on the contents pane.

- Selecting an item in the outline pane displays the corresponding item in the contents pane.
- Moving an item in the outline pane moves the corresponding item in the contents pane.
- Collapsing the outline view hides the results from all items in the collapsed levels.

Controlling the outline display. To control the outline display, you can:

- Expand and collapse the outline view
- Change the outline level for selected items
- Change the size of items in the outline display
- Change the font that is used in the outline display

To Collapse and Expand the Outline View

▶ Click the box to the left of the outline item that you want to collapse or expand.

or

▶ Click the item in the outline.

▶ From the menus choose:

View
  Collapse

or

View
  Expand

To Change the Outline Level

▶ Click the item in the outline pane.

▶ Click the left arrow on the Outlining toolbar to promote the item (move the item to the left).

or

Click the right arrow on the Outlining toolbar to demote the item (move the item to the right).

or

▶ From the menus choose:

Edit
  Outline
    Promote

or

Edit
  Outline
    Demote

Changing the outline level is particularly useful after you move items in the outline level. Moving items can change the outline level of the items, and you can use the left- and right-arrow buttons on the Outlining toolbar to restore the original outline level.

To Change the Size of Outline Items

▶ From the menus choose:

View
  Outline Size

▶ Select the outline size (Small, Medium, or Large).

To Change the Font in the Outline

- ▶ From the menus choose:

  View
    Outline Font...

- ▶ Select a font.

Adding Items to the Viewer

In the Viewer, you can add items such as titles, new text, charts, or material from other applications.

To Add a Title or Text

Text items that are not connected to a table or chart can be added to the Viewer.

- ▶ Click the table, chart, or other object that will precede the title or text.

- ▶ From the menus choose:

  Insert
    New Title

  or

  Insert
    New Text

- ▶ Double-click the new object.

- ▶ Enter the text.

To Add a Text File

- ▶ In the outline pane or contents pane of the Viewer, click the table, chart, or other object that will precede the text.

- ▶ From the menus choose:

  Insert
    Text File...

- ▶ Select a text file.

  To edit the text, double-click it.

Pasting Objects into the Viewer

Objects from other applications can be pasted into the Viewer. You can use either Paste After or Paste Special. Either type of pasting puts the new object after the currently selected object in the Viewer. Use Paste Special when you want to choose the format of the pasted object.

Finding and Replacing Information in the Viewer

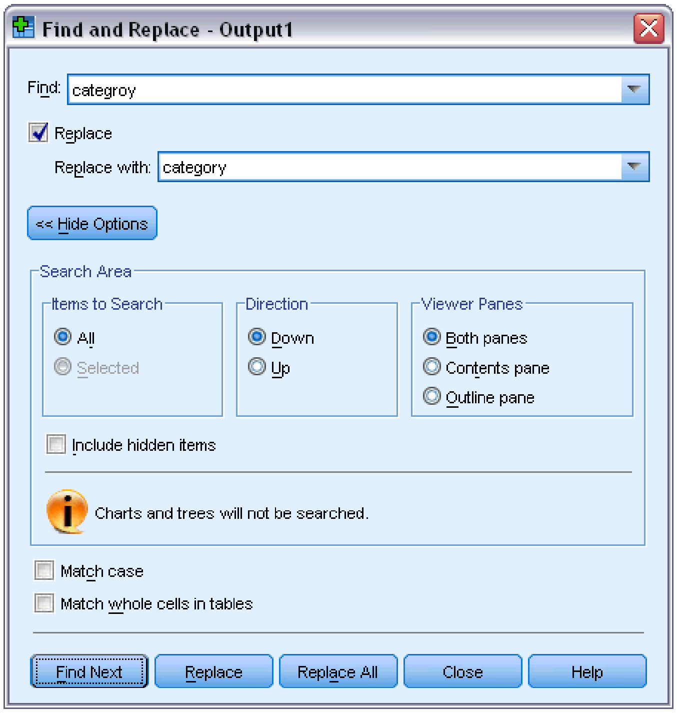
▶ To find or replace information in the Viewer, from the menus choose:

Edit
  Find

or

Edit
  Replace

Figure 10-2
Find and Replace dialog box



Find and Replace - Output1

Find: categroy

☑ Replace

Replace with: category

Search Area

Items to Search: ● All ○ Selected

Direction: ● Down ○ Up

Viewer Panes: ● Both panes ○ Contents pane ○ Outline pane

□ Include hidden items

Charts and trees will not be searched.

□ Match case

□ Match whole cells in tables

You can use Find and Replace to:

- Search the entire document or just the selected items.
- Search down or up from the current location.
- Search both panes or restrict the search to the contents or outline pane.
- Search for hidden items. These include any items hidden in the contents pane (for example, Notes tables, which are hidden by default) and hidden rows and columns in pivot tables.
- Restrict the search criteria to case-sensitive matches.
- Restrict the search criteria in pivot tables to matches of the entire cell contents.

Hidden Items and Pivot Table Layers

- Layers beneath the currently visible layer of a multidimensional pivot table are not considered hidden and will be included in the search area even when hidden items are not included in the search.

- Hidden items include hidden items in the contents pane (items with closed book icons in the outline pane or included within collapsed blocks of the outline pane) and rows and columns in pivot tables either hidden by default (for example, empty rows and columns are hidden by default) or manually hidden by editing the table and selectively hiding specific rows or columns. Hidden items are only included in the search if you explicitly select Include hidden items.
- In both cases, the hidden or nonvisible element that contains the search text or value is displayed when it is found, but the item is returned to its original state afterward.

Copying Output into Other Applications

Output objects can be copied and pasted into other applications, such as a word-processing program or a spreadsheet. You can paste output in a variety of formats. Depending on the target application, some or all of the following formats may be available:

Picture (metafile). Pivot tables and charts can be pasted as metafile pictures. The picture format can be resized in the other application, and sometimes a limited amount of editing can be done with the facilities of the other application. Pivot tables that are pasted as pictures retain all borders and font characteristics. For pivot tables, select Files from the Paste Special menu. This format is available only on Windows operating systems.

RTF (rich text format). Pivot tables can be pasted into other applications in RTF format. In most applications, this means that the pivot table is pasted as a table that can then be edited in the other application.

Note: Microsoft Word may not display extremely wide tables properly.

Bitmap. Charts and other graphics can be pasted into other applications as bitmaps.

BIFF. The contents of a table can be pasted into a spreadsheet and retain numeric precision. This format is available only on Windows operating systems.

Text. The contents of a table can be copied and pasted as text. This process can be useful for applications such as e-mail, where the application can accept or transmit only text.

If the target application supports multiple available formats, it may have a Paste Special menu item that allows you to select the format, or it may automatically display a list of available formats.

To Copy and Paste Output Items into Another Application

▶ Select the objects in either the outline or contents pane of the Viewer.

▶ From the Viewer menus choose:

Edit
  Copy

▶ From the menus in the target application choose:

Edit
  Paste

or

Edit
  Paste Special...

- **Paste.** Output is copied to the Clipboard in a number of formats. Each application determines the “best” format to use for Paste.
- **Paste Special.** Results are copied to the Clipboard in multiple formats. Paste Special allows you to select the format that you want from the list of formats that are available to the target application.

Copying and pasting model views. All model views, including tables, are pasted as graphic images.

Copying and pasting pivot tables. When pivot tables are pasted in Word/RTF format, tables that are too wide for the document width will either be wrapped, scaled down to fit the document width, or left unchanged, depending on the pivot table options settings. For more information, see the topic Pivot Table Options in Chapter 16 on p. 303.

Export Output

Export Output saves Viewer output in HTML, text, Word/RTF, Excel, PowerPoint (requires PowerPoint 97 or later), and PDF formats. Charts can also be exported in a number of different graphics formats.

Note: Export to PowerPoint is available only on Windows operating systems and is not available in the Student Version.

To Export Output

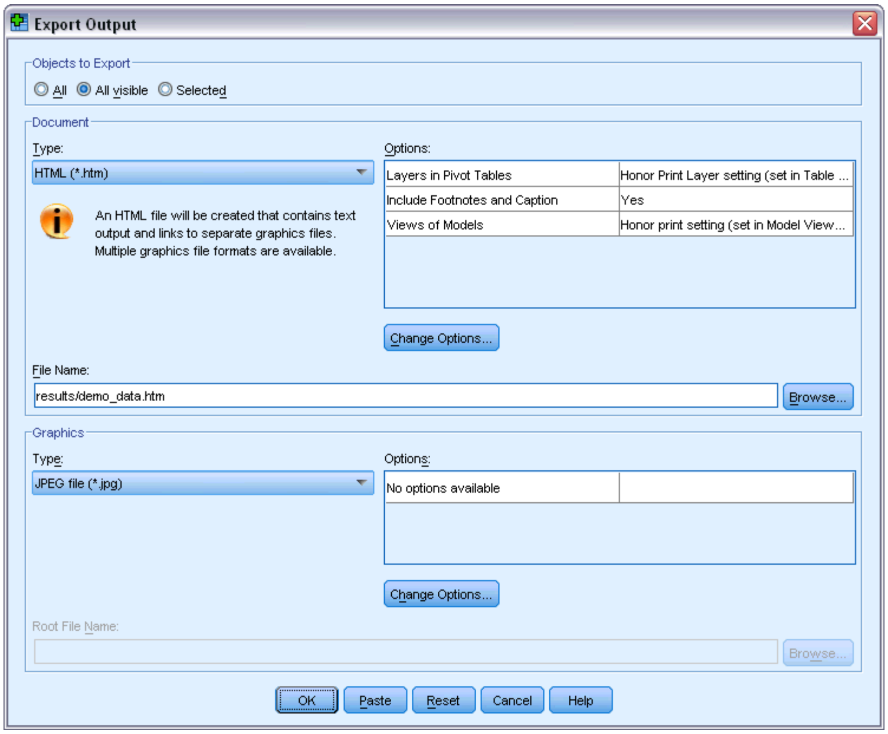
▶ Make the Viewer the active window (click anywhere in the window).

▶ From the menus choose:

File
  Export...

▶ Enter a filename (or prefix for charts) and select an export format.

Figure 10-3
Export Output dialog box



Export Output

Objects to Export

○ All ● All visible ○ Selected

Document

Type: HTML (*.htm)

An HTML file will be created that contains text output and links to separate graphics files. Multiple graphics file formats are available.

Options:

Layers in Pivot Tables	Honor Print Layer setting (set in Table ...
Include Footnotes and Caption	Yes
Views of Models	Honor print setting (set in Model View...

File Name: results/demo_data.htm

Graphics

Type: JPEG file (*.jpg)

Options: No options available

Root File Name:

Objects to Export. You can export all objects in the Viewer, all visible objects, or only selected objects.

Document Type. The available options are:

- **Word/RTF (*.doc).** Pivot tables are exported as Word tables with all formatting attributes intact—for example, cell borders, font styles, and background colors. Text output is exported as formatted RTF. Charts, tree diagrams, and model views are included in PNG format.

  Note: Microsoft Word may not display extremely wide tables properly.

- **Excel (*.xls).** Pivot table rows, columns, and cells are exported as Excel rows, columns, and cells, with all formatting attributes intact—for example, cell borders, font styles, and background colors. Text output is exported with all font attributes intact. Each line in the text output is a row in the Excel file, with the entire contents of the line contained in a single cell. Charts, tree diagrams, and model views are included in PNG format.

- **HTML (*.htm).** Pivot tables are exported as HTML tables. Text output is exported as preformatted HTML. Charts, tree diagrams, and model views are embedded by reference, and you should export charts in a suitable format for inclusion in HTML documents (for example, PNG and JPEG).

- **Portable Document Format (*.pdf).** All output is exported as it appears in Print Preview, with all formatting attributes intact.
- **PowerPoint file (*.ppt).** Pivot tables are exported as Word tables and are embedded on separate slides in the PowerPoint file, with one slide for each pivot table. All formatting attributes of the pivot table are retained—for example, cell borders, font styles, and background colors. Charts, tree diagrams, and model views are exported in TIFF format. Text output is not included.

  Export to PowerPoint is available only on Windows operating systems.
- **Text (*.txt).** Text output formats include plain text, UTF-8, and UTF-16. Pivot tables can be exported in tab-separated or space-separated format. All text output is exported in space-separated format. For charts, tree diagrams, and model views, a line is inserted in the text file for each graphic, indicating the image filename.
- **None (Graphics Only).** Available export formats include: EPS, JPEG, TIFF, PNG, and BMP. On Windows operating systems, EMF (enhanced metafile) format is also available.

Output Management System. You can also automatically export all output or user-specified types of output as Word, Excel, PDF, HTML, text or PASW Statistics-format data files. For more information, see the topic Output Management System in Chapter 20 on p. 353.

HTML Options

The following options are available for exporting output in HTML format:

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

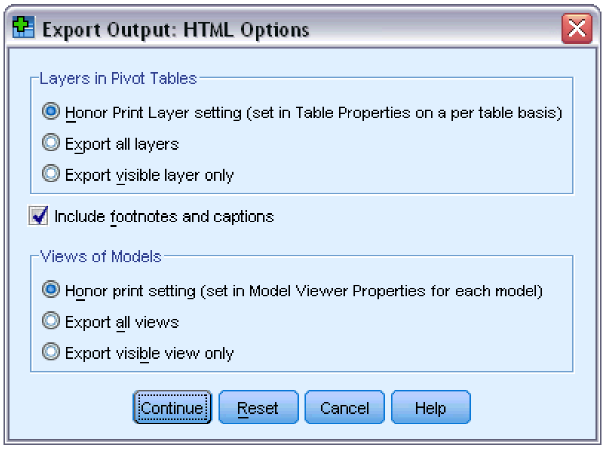
Note: For HTML, you can also control the image file format for exported charts. For more information, see the topic Graphics Format Options on p. 222.

To Set HTML Export Options

▶ Select HTML as the export format.

▶ Click Change Options.

Figure 10-4
HTML Export Output Options



Export Output: HTML Options

Layers in Pivot Tables

● Honor Print Layer setting (set in Table Properties on a per table basis)
○ Export all layers
○ Export visible layer only

☑ Include footnotes and captions

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)
○ Export all views
○ Export visible view only

Word/RTF Options

The following options are available for exporting output in Word format:

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

Wide Pivot Tables. Controls the treatment of tables that are too wide for the defined document width. By default, the table is wrapped to fit. The table is divided into sections, and row labels are repeated for each section of the table. Alternatively, you can shrink wide tables or make no changes to wide tables and allow them to extend beyond the defined document width.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

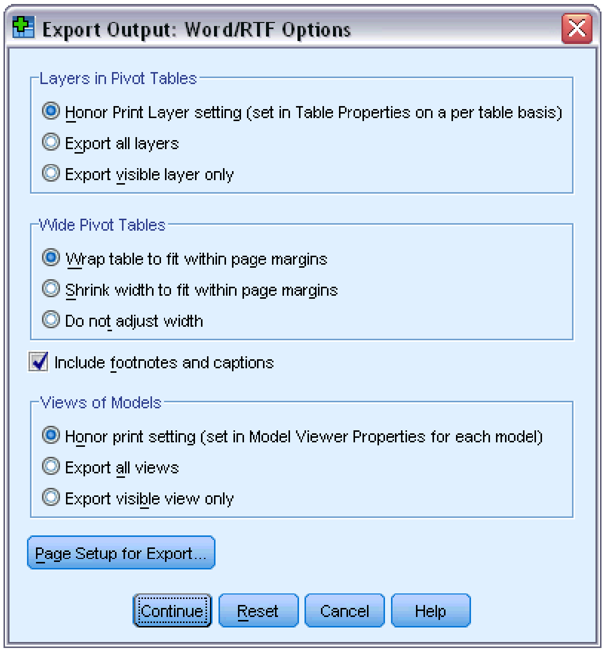
Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

Page Setup for Export. This opens a dialog where you can define the page size and margins for the exported document. The document width used to determine wrapping and shrinking behavior is the page width minus the left and right margins.

To Set Word Export Options

- Select Word/RTF as the export format.
- Click Change Options.

Figure 10-5
Word Export Output Options



Export Output: Word/RTF Options

Layers in Pivot Tables

● Honor Print Layer setting (set in Table Properties on a per table basis)
○ Export all layers
○ Export visible layer only

Wide Pivot Tables

● Wrap table to fit within page margins
○ Shrink width to fit within page margins
○ Do not adjust width

☑ Include footnotes and captions

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)
○ Export all views
○ Export visible view only

Excel Options

The following options are available for exporting output in Excel format:

Create a worksheet or workbook or modify an existing worksheet. By default, a new workbook is created. If a file with the specified name already exists, it will be overwritten. If you select the option to create a worksheet, if a worksheet with the specified name already exists in the specified file, it will be overwritten. If you select the option to modify an existing worksheet, you must also specify the worksheet name. (This is optional for creating a worksheet.) Worksheet names cannot exceed 31 characters and cannot contain forward or back slashes, square brackets, question marks, or asterisks.

If you modify an existing worksheet, charts, model views, and tree diagrams are not included in the exported output.

Location in worksheet. Controls the location within the worksheet for the exported output. By default, exported output will be added after the last column that has any content, starting in the first row, without modifying any existing contents. This is a good choice for adding new columns to an existing worksheet. Adding exported output after the last row is a good choice for adding new rows to an existing worksheet. Adding exported output starting at a specific cell location will overwrite any existing content in the area where the exported output is added.

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

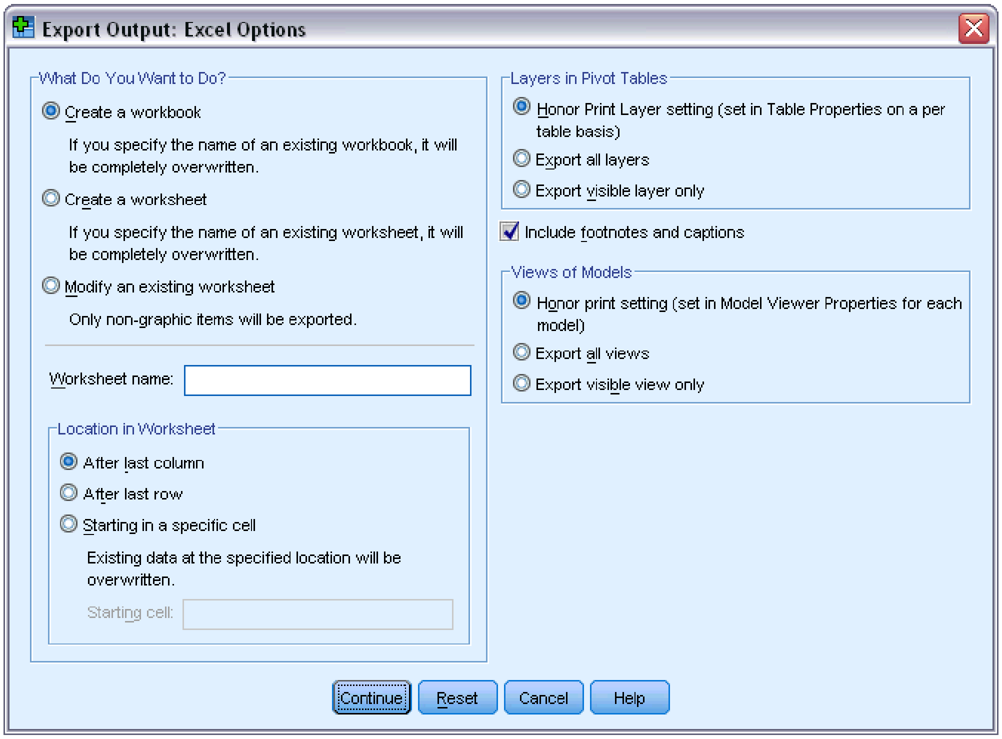
Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

To Set Excel Export Options

▶ Select Excel as the export format.

▶ Click Change Options.

Figure 10-6
Excel Export Output Options



Export Output: Excel Options

What Do You Want to Do?

● Create a workbook

If you specify the name of an existing workbook, it will be completely overwritten.

○ Create a worksheet

If you specify the name of an existing worksheet, it will be completely overwritten.

○ Modify an existing worksheet

Only non-graphic items will be exported.

Worksheet name:

Location in Worksheet

● After last column
○ After last row
○ Starting in a specific cell

Existing data at the specified location will be overwritten.

Starting cell:

Layers in Pivot Tables

● Honor Print Layer setting (set in Table Properties on a per table basis)
○ Export all layers
○ Export visible layer only

☑ Include footnotes and captions

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)
○ Export all views
○ Export visible view only

PowerPoint Options

The following options are available for PowerPoint:

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

Wide Pivot Tables. Controls the treatment of tables that are too wide for the defined document width. By default, the table is wrapped to fit. The table is divided into sections, and row labels are repeated for each section of the table. Alternatively, you can shrink wide tables or make no changes to wide tables and allow them to extend beyond the defined document width.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Use Viewer outline entries as slide titles. Includes a title on each slide that is created by the export. Each slide contains a single item that is exported from the Viewer. The title is formed from the outline entry for the item in the outline pane of the Viewer.

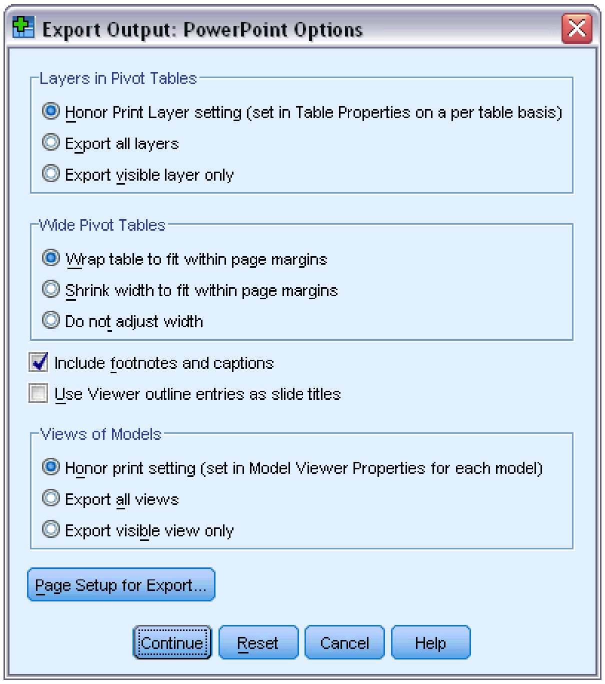
Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

Page Setup for Export. This opens a dialog where you can define the page size and margins for the exported document. The document width used to determine wrapping and shrinking behavior is the page width minus the left and right margins.

To Set PowerPoint Export Options

- ▶ Select PowerPoint as the export format.
- ▶ Click Change Options.

Figure 10-7
Powerpoint Export Options dialog



Export Output: PowerPoint Options

Layers in Pivot Tables

● Honor Print Layer setting (set in Table Properties on a per table basis)
○ Export all layers
○ Export visible layer only

Wide Pivot Tables

● Wrap table to fit within page margins
○ Shrink width to fit within page margins
○ Do not adjust width

☑ Include footnotes and captions
□ Use Viewer outline entries as slide titles

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)
○ Export all views
○ Export visible view only

Note: Export to PowerPoint is available only on Windows operating systems.

PDF Options

The following options are available for PDF:

Embed bookmarks. This option includes bookmarks in the PDF document that correspond to the Viewer outline entries. Like the Viewer outline pane, bookmarks can make it much easier to navigate documents with a large number of output objects.

Embed fonts. Embedding fonts ensures that the PDF document will look the same on all computers. Otherwise, if some fonts used in the document are not available on the computer being used to view (or print) the PDF document, font substitution may yield suboptimal results.

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

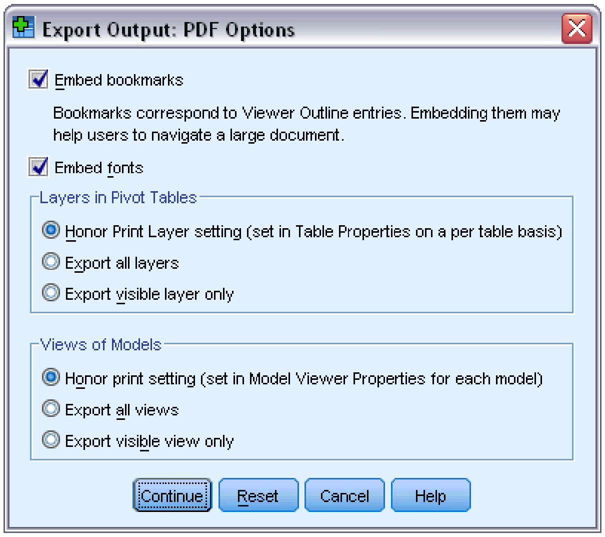
Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

To Set PDF Export Options

▶ Select Portable Document Format as the export format.

▶ Click Change Options.

Figure 10-8
PDF Options dialog box



☑ Embed bookmarks

Bookmarks correspond to Viewer Outline entries. Embedding them may help users to navigate a large document.

☑ Embed fonts

Layers in Pivot Tables

● Honor Print Layer setting (set in Table Properties on a per table basis)
○ Export all layers
○ Export visible layer only

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)
○ Export all views
○ Export visible view only

Other Settings That Affect PDF Output

Page Setup/Page Attributes. Page size, orientation, margins, content and display of page headers and footers, and printed chart size in PDF documents are controlled by page setup and page attribute options.

Table Properties/TableLooks. Scaling of wide and/or long tables and printing of table layers are controlled by table properties for each table. These properties can also be saved in TableLooks. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

Default/Current Printer. The resolution (DPI) of the PDF document is the current resolution setting for the default or currently selected printer (which can be changed using Page Setup). The maximum resolution is 1200 DPI. If the printer setting is higher, the PDF document resolution will be 1200 DPI.

Note: High-resolution documents may yield poor results when printed on lower-resolution printers.

Text Options

The following options are available for text export:

Pivot Table Format. Pivot tables can be exported in tab-separated or space-separated format. For space-separated format, you can also control:

- **Column Width.** Autofit does not wrap any column contents, and each column is as wide as the widest label or value in that column. Custom sets a maximum column width that is applied to all columns in the table, and values that exceed that width wrap onto the next line in that column.
- **Row/Column Border Character.** Controls the characters used to create row and column borders. To suppress display of row and column borders, enter blank spaces for the values.

Layers in pivot tables. By default, inclusion or exclusion of pivot table layers is controlled by the table properties for each pivot table. You can override this setting and include all layers or exclude all but the currently visible layer. For more information, see the topic Table Properties: Printing in Chapter 11 on p. 243.

Include footnotes and captions. Controls the inclusion or exclusion of all pivot table footnotes and captions.

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

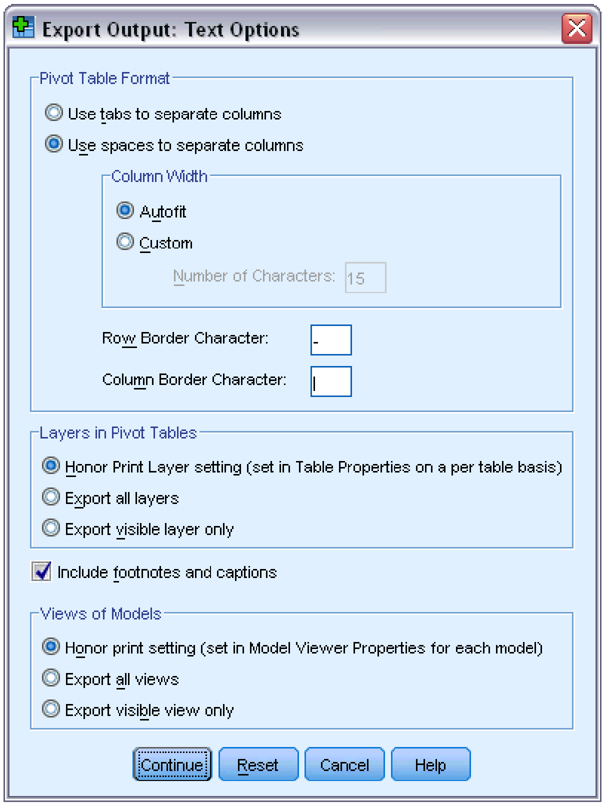
To Set Text Export Options

▶ Select Text as the export format.

▶ Click Change Options.

Figure 10-9

Text Options dialog box



Export Output: Text Options

Pivot Table Format

○ Use tabs to separate columns

● Use spaces to separate columns

Column Width

● Autofit

○ Custom

Number of Characters: 15

Row Border Character: -

Column Border Character: |

Layers in Pivot Tables

● Honor Print Layer setting (set in Table Properties on a per table basis)

○ Export all layers

○ Export visible layer only

☑ Include footnotes and captions

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)

○ Export all views

○ Export visible view only

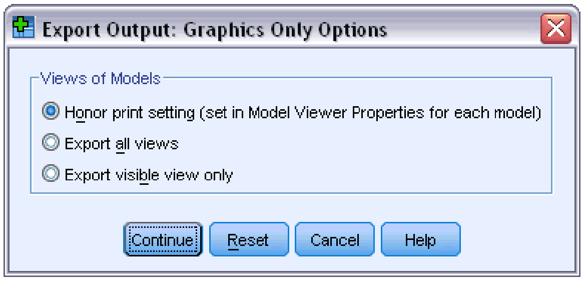
Graphics Only Options

The following options are available for exporting graphics only:

Views of Models. By default, inclusion or exclusion of model views is controlled by the model properties for each model. You can override this setting and include all views or exclude all but the currently visible view. For more information, see the topic Model Properties in Chapter 12 on p. 254. (Note: all model views, including tables, are exported as graphics.)

Figure 10-10

Graphics Only Options



Export Output: Graphics Only Options

Views of Models

● Honor print setting (set in Model Viewer Properties for each model)

○ Export all views

○ Export visible view only

Graphics Format Options

For HTML and text documents and for exporting charts only, you can select the graphic format, and for each graphic format you can control various optional settings.

To select the graphic format and options for exported charts:

- ▶ Select HTML, Text, or None (Graphics only) as the document type.
- ▶ Select the graphic file format from the drop-down list.
- ▶ Click Change Options to change the options for the selected graphic file format.

JPEG Chart Export Options

- **Image size.** Percentage of original chart size, up to 200 percent.
- **Convert to grayscale.** Converts colors to shades of gray.

BMP Chart Export Options

- **Image size.** Percentage of original chart size, up to 200 percent.
- **Compress image to reduce file size.** A lossless compression technique that creates smaller files without affecting image quality.

PNG Chart Export Options

Image size. Percentage of original chart size, up to 200 percent.

Color Depth. Determines the number of colors in the exported chart. A chart that is saved under any depth will have a minimum of the number of colors that are actually used and a maximum of the number of colors that are allowed by the depth. For example, if the chart contains three colors—red, white, and black—and you save it as 16 colors, the chart will remain as three colors.

- If the number of colors in the chart exceeds the number of colors for that depth, the colors will be dithered to replicate the colors in the chart.
- Current screen depth is the number of colors currently displayed on your computer monitor.

EMF and TIFF Chart Export Options

Image size. Percentage of original chart size, up to 200 percent.

Note: EMF (enhanced metafile) format is available only on Windows operating systems.

EPS Chart Export Options

Image size. You can specify the size as a percentage of the original image size (up to 200 percent), or you can specify an image width in pixels (with height determined by the width value and the aspect ratio). The exported image is always proportional to the original.

Include TIFF preview image. Saves a preview with the EPS image in TIFF format for display in applications that cannot display EPS images on screen.

Fonts. Controls the treatment of fonts in EPS images.

- **Use font references.** If the fonts that are used in the chart are available on the output device, the fonts are used. Otherwise, the output device uses alternate fonts.
- **Replace fonts with curves.** Turns fonts into PostScript curve data. The text itself is no longer editable as text in applications that can edit EPS graphics. This option is useful if the fonts that are used in the chart are not available on the output device.

Viewer Printing

There are two options for printing the contents of the Viewer window:

All visible output. Prints only items that are currently displayed in the contents pane. Hidden items (items with a closed book icon in the outline pane or hidden in collapsed outline layers) are not printed.

Selection. Prints only items that are currently selected in the outline and/or contents panes.

To Print Output and Charts

▶ Make the Viewer the active window (click anywhere in the window).

▶ From the menus choose:

File
  Print...

▶ Select the print settings that you want.

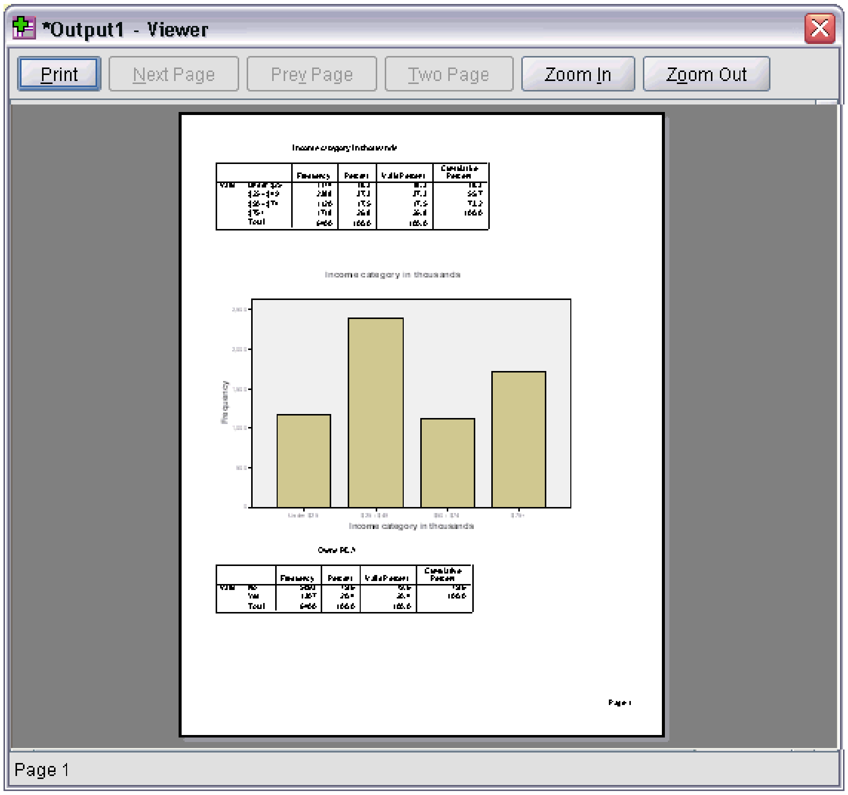
▶ Click OK to print.

Print Preview

Print Preview shows you what will print on each page for Viewer documents. It is a good idea to check Print Preview before actually printing a Viewer document, because Print Preview shows you items that may not be visible by looking at the contents pane of the Viewer, including:

- Page breaks
- Hidden layers of pivot tables
- Breaks in wide tables
- Headers and footers that are printed on each page

Figure 10-11
Print Preview



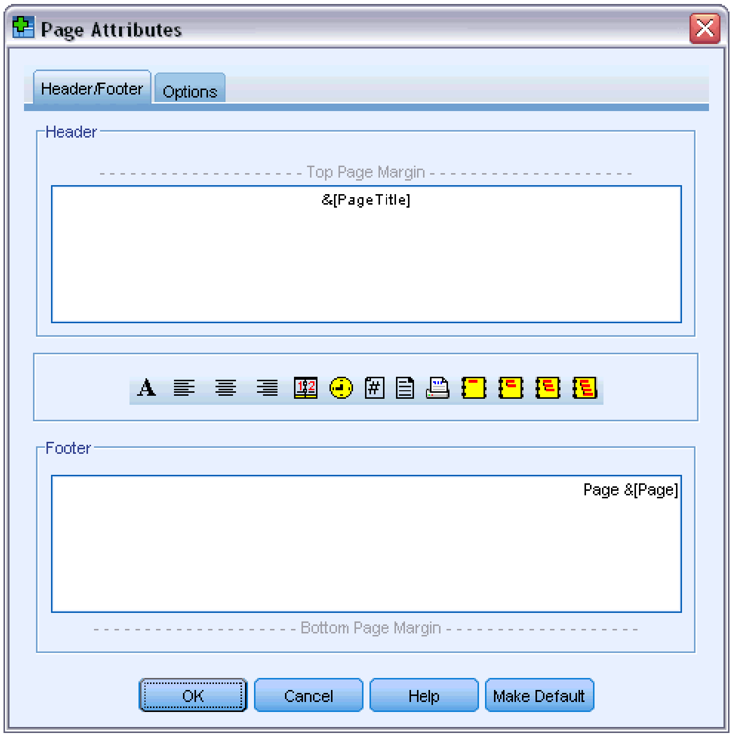
If any output is currently selected in the Viewer, the preview displays only the selected output. To view a preview for all output, make sure nothing is selected in the Viewer.

Page Attributes: Headers and Footers

Headers and footers are the information that is printed at the top and bottom of each page. You can enter any text that you want to use as headers and footers. You can also use the toolbar in the middle of the dialog box to insert:

- Date and time
- Page numbers
- Viewer filename
- Outline heading labels
- Page titles and subtitles

Figure 10-12
Page Attributes dialog box, Header/Footer tab



- Make Default uses the settings specified here as the default settings for new Viewer documents. (Note: this makes the current settings on both the Header/Footer tab and the Options tab the default settings.)
- Outline heading labels indicate the first-, second-, third-, and/or fourth-level outline heading for the first item on each page.
- Page titles and subtitles print the current page titles and subtitles. These can be created with New Page Title on the Viewer Insert menu or with the `TITLE` and `SUBTITLE` commands. If you have not specified any page titles or subtitles, this setting is ignored.

  Note: Font characteristics for new page titles and subtitles are controlled on the Viewer tab of the Options dialog box (accessed by choosing Options on the Edit menu). Font characteristics for existing page titles and subtitles can be changed by editing the titles in the Viewer.

To see how your headers and footers will look on the printed page, choose Print Preview from the File menu.

To Insert Page Headers and Footers

▶ Make the Viewer the active window (click anywhere in the window).

▶ From the menus choose:

File
  Page Attributes...

▶ Click the Header/Footer tab.

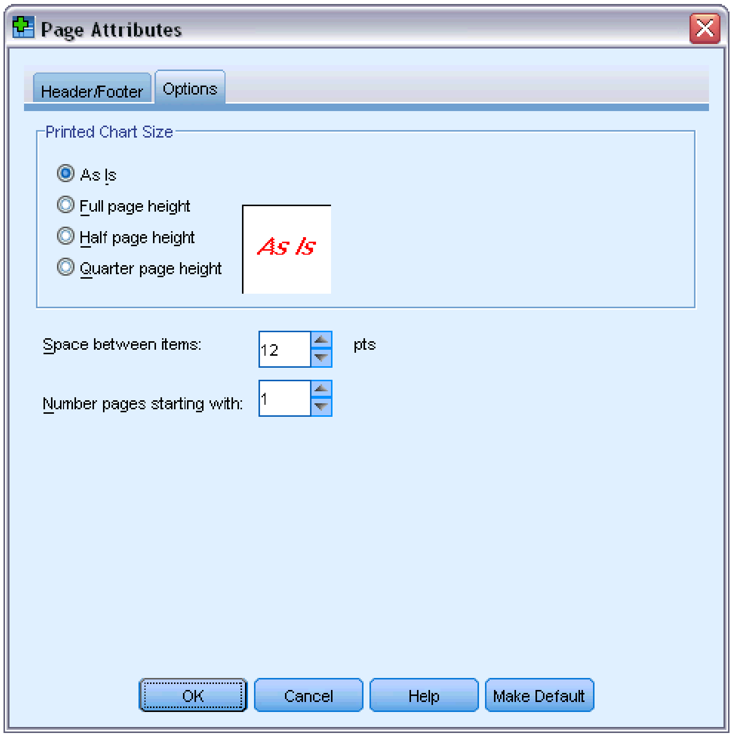
▶ Enter the header and/or footer that you want to appear on each page.

Page Attributes: Options

This dialog box controls the printed chart size, the space between printed output items, and page numbering.

- **Printed Chart Size.** Controls the size of the printed chart relative to the defined page size. The chart's aspect ratio (width-to-height ratio) is not affected by the printed chart size. The overall printed size of a chart is limited by both its height and width. When the outer borders of a chart reach the left and right borders of the page, the chart size cannot increase further to fill additional page height.
- **Space between items.** Controls the space between printed items. Each pivot table, chart, and text object is a separate item. This setting does not affect the display of items in the Viewer.
- **Number pages starting with.** Numbers pages sequentially, starting with the specified number.
- **Make Default.** This option uses the settings specified here as the default settings for new Viewer documents. (Note: this makes the current settings on both the Header/Footer tab and the Options tab the default settings.)

Figure 10-13
Page Attributes dialog box, Options tab



Page Attributes

Printed Chart Size

● As Is
○ Full page height
○ Half page height
○ Quarter page height

Space between items: 12 pts

Number pages starting with: 1

To Change Printed Chart Size, Page Numbering, and Space between Printed Items

▶ Make the Viewer the active window (click anywhere in the window).

▶ From the menus choose:

File
  Page Attributes...

▶ Click the Options tab.

▶ Change the settings and click OK.

Saving Output

The contents of the Viewer can be saved to a Viewer document. The saved document includes both panes of the Viewer window (the outline and the contents).

To Save a Viewer Document

▶ From the Viewer window menus choose:

File
  Save

▶ Enter the name of the document, and then click Save.

Optionally, you can lock files to prevent editing in Smartreader (a separate product for working with Viewer documents). If a Viewer document is locked, you can manipulate pivot tables (swap rows and columns, change the displayed layer, etc.) but you cannot edit any output or save any changes to the Viewer document in Smartreader. This setting has no effect on Viewer documents opened in PASW Statistics.

To save results in external formats (for example, HTML or text), use Export on the File menu.

Chapter

11

Pivot Tables

Many results are presented in tables that can be pivoted interactively. That is, you can rearrange the rows, columns, and layers.

Manipulating a Pivot Table

Options for manipulating a pivot table include:

- Transposing rows and columns
- Moving rows and columns
- Creating multidimensional layers
- Grouping and ungrouping rows and columns
- Showing and hiding rows, columns, and other information
- Rotating row and column labels
- Finding definitions of terms

Activating a Pivot Table

Before you can manipulate or modify a pivot table, you need to **activate** the table. To activate a table:

▶ Double-click the table.

or

▶ Right-click the table and from the context menu choose Edit Content.

▶ From the sub-menu choose either In Viewer or In Separate Window.

- By default, activating the table by double-clicking will activate all but very large tables in the Viewer window. For more information, see the topic Pivot Table Options in Chapter 16 on p. 303.
- If you want to have more than one pivot table activated at the same time, you must activate the tables in separate windows.

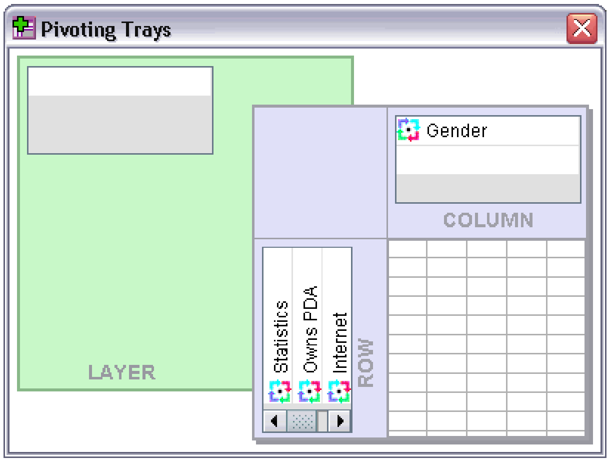
Pivoting a Table

▶ Activate the pivot table.

▶ From the menus choose:

Pivot
  Pivoting Trays

Figure 11-1
Pivoting trays



A table has three dimensions: rows, columns, and layers. A dimension can contain multiple elements (or none at all). You can change the organization of the table by moving elements between or within dimensions. To move an element, just drag and drop it where you want it.

Changing Display Order of Elements within a Dimension

To change the display order of elements within a table dimension (row, column, or layer):

▶ If pivoting trays are not already on, from the Pivot Table menu choose:

Pivot
  Pivoting Trays

▶ Drag and drop the elements within the dimension in the pivoting tray.

Moving Rows and Columns within a Dimension Element

▶ In the table itself (not the pivoting trays), click the label for the row or column you want to move.

▶ Drag the label to the new position.

▶ From the context menu choose Insert Before or Swap.

Note: Make sure that Drag to Copy on the Edit menu is *not* enabled (checked). If Drag to Copy is enabled, deselect it.

Transposing Rows and Columns

If you just want to flip the rows and columns, there's a simple alternative to using the pivoting trays:

- From the menus choose:

Pivot
  Transpose Rows and Columns

This has the same effect as dragging all of the row elements into the column dimension and dragging all of the column elements into the row dimension.

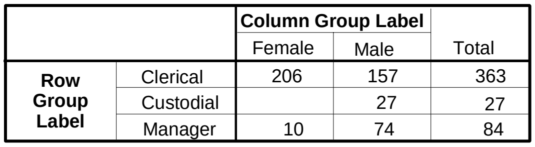
Grouping Rows or Columns

- Select the labels for the rows or columns that you want to group together (click and drag or Shift+click to select multiple labels).
- From the menus choose:

Edit
  Group

A group label is automatically inserted. Double-click the group label to edit the label text.

Figure 11-2
Row and column groups and labels



		Column Group Label		
		Female	Male	Total
Row Group Label	Clerical	206	157	363
	Custodial		27	27
	Manager	10	74	84

Note: To add rows or columns to an existing group, you must first ungroup the items that are currently in the group. Then you can create a new group that includes the additional items.

Ungrouping Rows or Columns

- Click anywhere in the group label for the rows or columns that you want to ungroup.
- From the menus choose:

Edit
  Ungroup

Ungrouping automatically deletes the group label.

Rotating Row or Column Labels

You can rotate labels between horizontal and vertical display for the innermost column labels and the outermost row labels in a table.

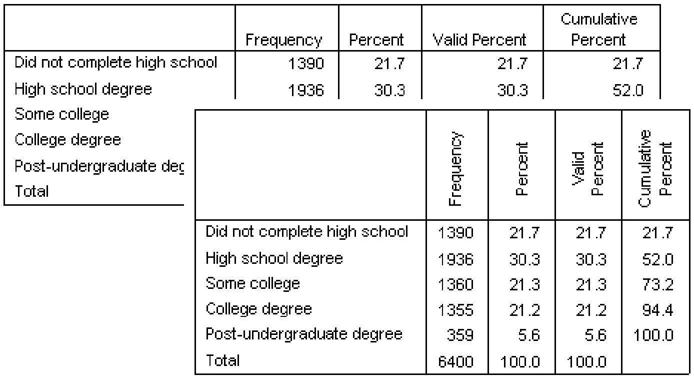
- From the menus choose:

Format
  Rotate Inner Column Labels

or

Format
  Rotate Outer Row Labels

Figure 11-3
Rotated column labels



	Frequency	Percent	Valid Percent	Cumulative Percent
Did not complete high school	1390	21.7	21.7	21.7
High school degree	1936	30.3	30.3	52.0
Some college				
College degree				
Post-undergraduate deg				
Total				

	Frequency	Percent	Valid Percent	Cumulative Percent
Did not complete high school	1390	21.7	21.7	21.7
High school degree	1936	30.3	30.3	52.0
Some college	1360	21.3	21.3	73.2
College degree	1355	21.2	21.2	94.4
Post-undergraduate degree	359	5.6	5.6	100.0
Total	6400	100.0	100.0	

Only the innermost column labels and the outermost row labels can be rotated.

Working with Layers

You can display a separate two-dimensional table for each category or combination of categories. The table can be thought of as stacked in layers, with only the top layer visible.

Creating and Displaying Layers

To create layers:

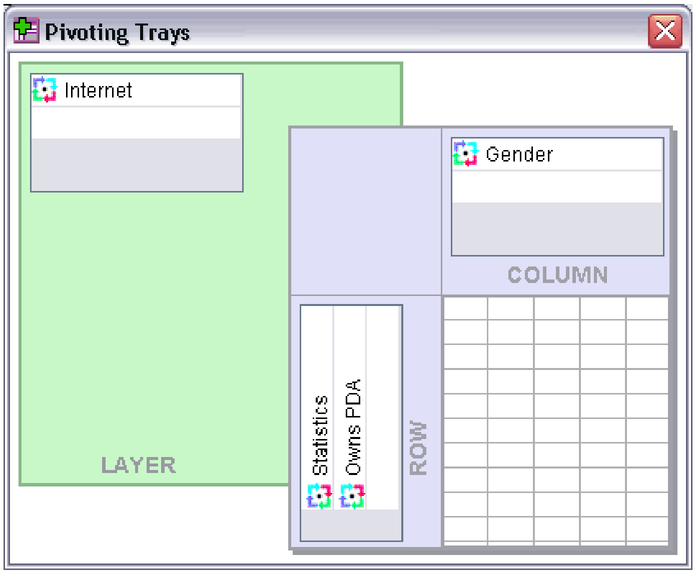
▶ Activate the pivot table.

▶ If pivoting trays are not already on, from the Pivot Table menu choose:

Pivot
  Pivoting Trays

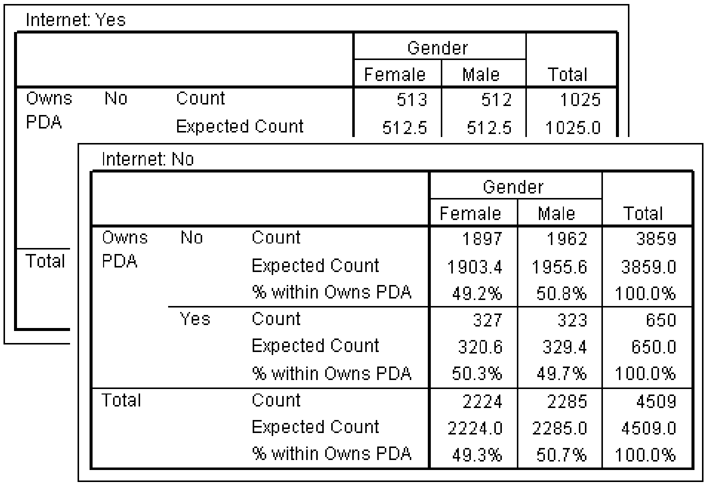
▶ Drag an element from the row or column dimension into the layer dimension.

Figure 11-4
Moving categories into layers



Moving elements to the layer dimension creates a multidimensional table, but only a single two-dimensional “slice” is displayed. The visible table is the table for the top layer. For example, if a yes/no categorical variable is in the layer dimension, then the multidimensional table has two layers: one for the *yes* category and one for the *no* category.

Figure 11-5
Categories in separate layers



Internet: Yes

			Gender		
			Female	Male	Total
Owns PDA	No	Count	513	512	1025
		Expected Count	512.5	512.5	1025.0
Total					

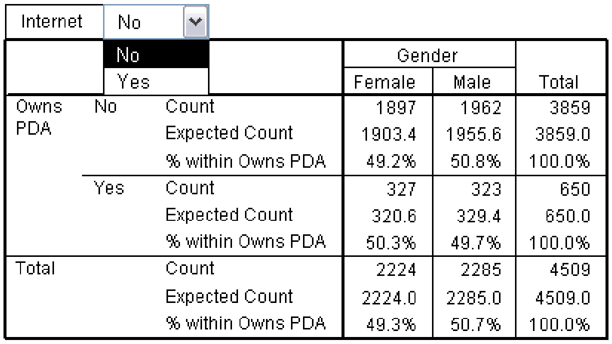
Internet: No

			Gender		
			Female	Male	Total
Owns PDA	No	Count	1897	1962	3859
		Expected Count	1903.4	1955.6	3859.0
		% within Owns PDA	49.2%	50.8%	100.0%
	Yes	Count	327	323	650
		Expected Count	320.6	329.4	650.0
		% within Owns PDA	50.3%	49.7%	100.0%
Total		Count	2224	2285	4509
		Expected Count	2224.0	2285.0	4509.0
		% within Owns PDA	49.3%	50.7%	100.0%

Changing the Displayed Layer

▶ Choose a category from the drop-down list of layers (in the pivot table itself, not the pivoting tray).

Figure 11-6
Selecting layers from drop-down lists



			Gender		
			Female	Male	Total
Owns PDA	No	Count	1897	1962	3859
		Expected Count	1903.4	1955.6	3859.0
		% within Owns PDA	49.2%	50.8%	100.0%
	Yes	Count	327	323	650
		Expected Count	320.6	329.4	650.0
		% within Owns PDA	50.3%	49.7%	100.0%
Total		Count	2224	2285	4509
		Expected Count	2224.0	2285.0	4509.0
		% within Owns PDA	49.3%	50.7%	100.0%

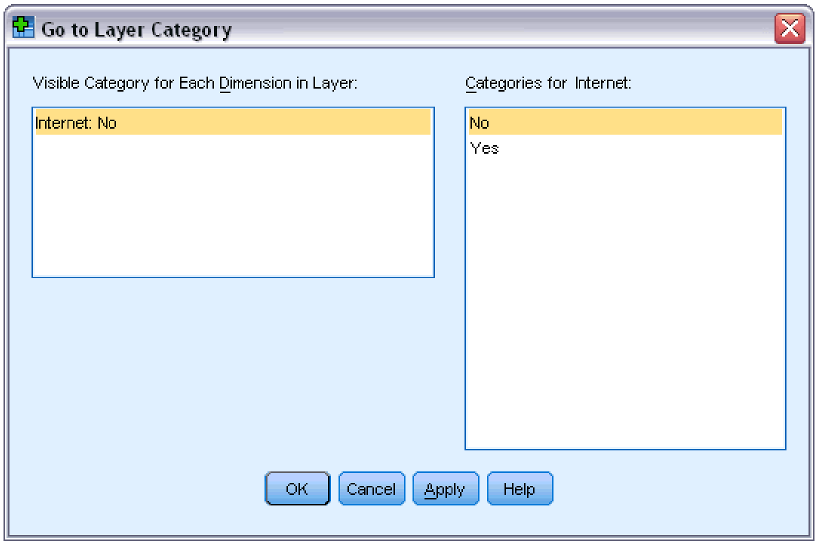
Go to Layer Category

Go to Layer Category allows you to change layers in a pivot table. This dialog box is particularly useful when there are many layers or the selected layer has many categories.

▶ From the menus choose:

Pivot
  Go to Layer...

Figure 11-7
Go to Layer Category dialog box



▶ In the Visible Category list, select a layer dimension. The Categories list will display all categories for the selected dimension.

▶ In the Categories list, select the category that you want, and then click OK or Apply.

Showing and Hiding Items

Many types of cells can be hidden, including:

- Dimension labels
- Categories, including the label cell and data cells in a row or column
- Category labels (without hiding the data cells)
- Footnotes, titles, and captions

Hiding Rows and Columns in a Table

▶ Right-click the category label for the row or column you want to hide.

▶ From the context menu choose:

Select
  Data and Label Cells

▶ Right-click the category label again and from the context menu choose Hide Category.

or

▶ From the View menu choose Hide.

Showing Hidden Rows and Columns in a Table

▶ Right-click another row or column label in the same dimension as the hidden row or column.

▶ From the context menu choose:

Select
  Data and Label Cells

▶ From the menus choose:

View
  Show All Categories in [dimension name]

or

▶ To display all hidden rows and columns in an activated pivot table, from the menus choose:

View
  Show All

This displays all hidden rows and columns in the table. (If Hide empty rows and columns is selected in Table Properties for this table, a completely empty row or column remains hidden.)

Hiding and Showing Dimension Labels

▶ Select the dimension label or any category label within the dimension.

▶ From the View menu or the context menu choose Hide Dimension Label or Show Dimension Label.

Hiding and Showing Table Titles

To hide a title:

▶ Select the title.

▶ From the View menu choose Hide.

To show hidden titles:

▶ From the View menu choose Show All.

TableLooks

A TableLook is a set of properties that define the appearance of a table. You can select a previously defined TableLook or create your own TableLook.

- Before or after a TableLook is applied, you can change cell formats for individual cells or groups of cells by using cell properties. The edited cell formats will remain intact, even when you apply a new TableLook. For more information, see the topic Cell Properties on p. 245.
- Optionally, you can reset all cells to the cell formats that are defined by the current TableLook. This resets any cells that have been edited. If As Displayed is selected in the TableLook Files list, any edited cells are reset to the current table properties.

Note: TableLooks created in earlier versions of PASW Statistics cannot be used in version 16.0 or later.

To Apply or Save a TableLook

▶ Activate a pivot table.

▶ From the menus choose:

Format
  TableLooks...

Figure 11-8
TableLooks dialog box

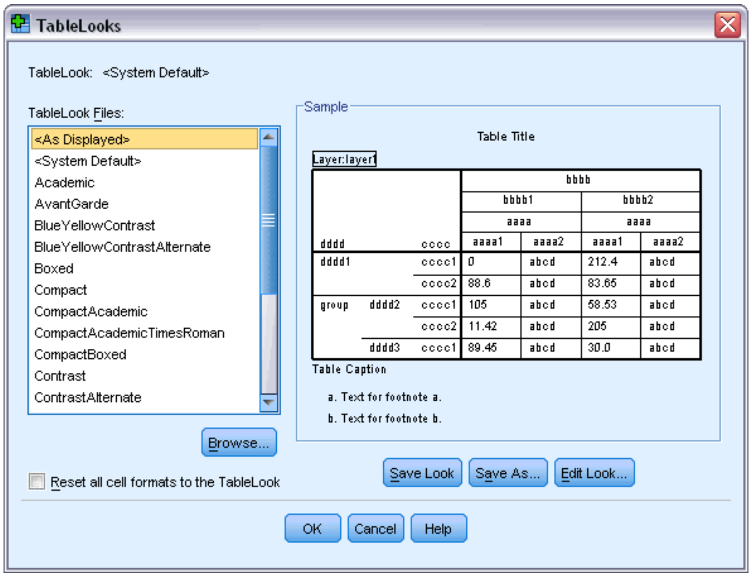


Table Title

Layer:layer1

			bbbb			
			bbbb1		bbbb2	
			aaaa		aaaa	
dddd		cccc	aaaa1	aaaa2	aaaa1	aaaa2
dddd1		cccc1	0	abcd	212.4	abcd
		cccc2	88.6	abcd	83.65	abcd
group	dddd2	cccc1	105	abcd	58.53	abcd
		cccc2	11.42	abcd	205	abcd
	dddd3	cccc1	89.45	abcd	30.0	abcd

Table Caption

a. Text for footnote a.

b. Text for footnote b.

- ▶ Select a TableLook from the list of files. To select a file from another directory, click Browse.
- ▶ Click OK to apply the TableLook to the selected pivot table.

To Edit or Create a TableLook

- ▶ In the TableLooks dialog box, select a TableLook from the list of files.
- ▶ Click Edit Look.
- ▶ Adjust the table properties for the attributes that you want, and then click OK.
- ▶ Click Save Look to save the edited TableLook, or click Save As to save it as a new TableLook.

Editing a TableLook affects only the selected pivot table. An edited TableLook is not applied to any other tables that uses that TableLook unless you select those tables and reapply the TableLook.

Table Properties

Table Properties allows you to set general properties of a table, set cell styles for various parts of a table, and save a set of those properties as a TableLook. You can:

- Control general properties, such as hiding empty rows or columns and adjusting printing properties.
- Control the format and position of footnote markers.
- Determine specific formats for cells in the data area, for row and column labels, and for other areas of the table.
- Control the width and color of the lines that form the borders of each area of the table.

To Change Pivot Table Properties

- ▶ Activate the pivot table.
- ▶ From the menus choose:

  Format

    Table Properties...
- ▶ Select a tab (General, Footnotes, Cell Formats, Borders, or Printing).
- ▶ Select the options that you want.
- ▶ Click OK or Apply.

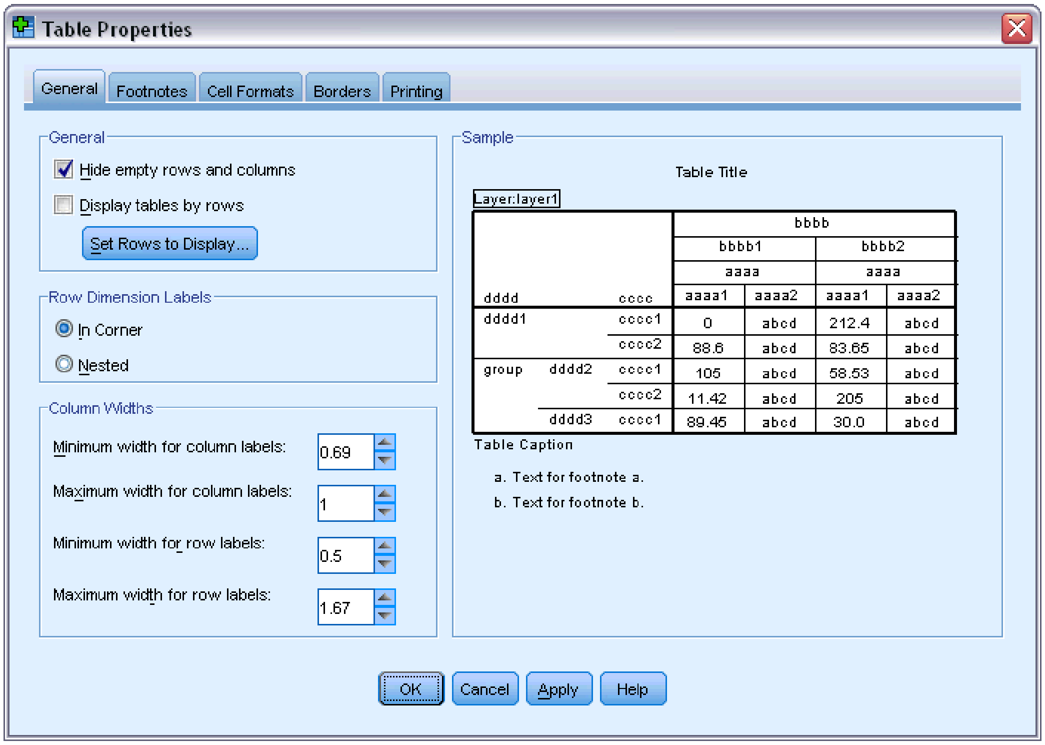
  The new properties are applied to the selected pivot table. To apply new table properties to a TableLook instead of just the selected table, edit the TableLook (Format menu, TableLooks).

Table Properties: General

Several properties apply to the table as a whole. You can:

- Show or hide empty rows and columns. (An empty row or column has nothing in any of the data cells.)
- Control the default number of rows to display in long tables. To display all the rows in a table, regardless of how long it is, deselect (uncheck) Display table by rows.
- Control the placement of row labels, which can be in the upper left corner or nested.
- Control maximum and minimum column width (expressed in points).

Figure 11-9
Table Properties dialog box, General tab



General

☑ Hide empty rows and columns

□ Display tables by rows

Row Dimension Labels

● In Corner

○ Nested

Column Widths

Minimum width for column labels: 0.69

Maximum width for column labels: 1

Minimum width for row labels: 0.5

Maximum width for row labels: 1.67

Sample

Table Title

Layer:layer1

			bbbb			
			bbbb1		bbbb2	
			aaaa		aaaa	
dddd		cccc	aaaa1	aaaa2	aaaa1	aaaa2
dddd1		cccc1	0	abcd	212.4	abcd
		cccc2	88.6	abcd	83.65	abcd
group	dddd2	cccc1	105	abcd	58.53	abcd
		cccc2	11.42	abcd	205	abcd
	dddd3	cccc1	89.45	abcd	30.0	abcd

Table Caption

a. Text for footnote a.

b. Text for footnote b.

To change general table properties:

▶ Click the General tab.

▶ Select the options that you want.

▶ Click OK or Apply.

Set Rows to Display

By default, tables with many rows are displayed in sections of 100 rows. To control the number of rows displayed in a table:

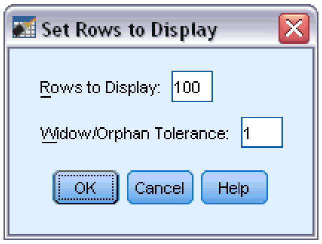
▶ Select Display table by rows.

▶ Click Set Rows to Display.

or

▶ From the View menu of an activated pivot table, choose Display table by rows and Set Rows to Display.

Figure 11-10
Set Rows to Display dialog


Set Rows to Display

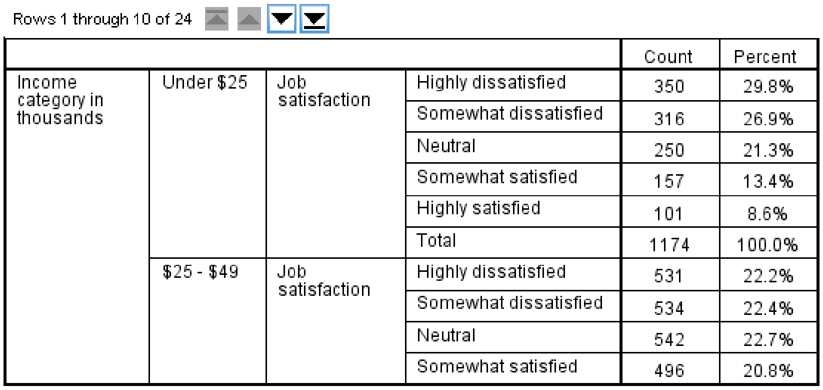
Rows to Display: 100

Widow/Orphan Tolerance: 1

Rows to display. Controls the maximum number of rows to display at one time. Navigation controls allow you move to different sections of the table. The minimum value is 10. The default is 100.

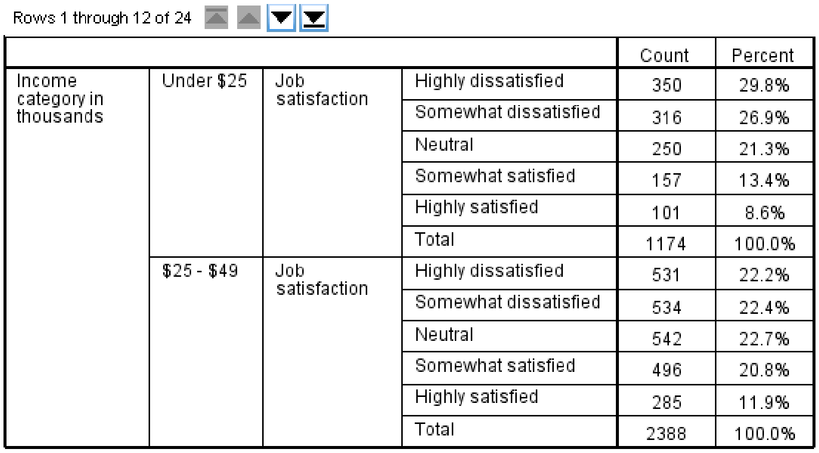
Widow/orphan tolerance. Controls the maximum number of rows of the inner most row dimension of the table to split across displayed views of the table. For example, if there are six categories in each group of the inner most row dimension, specifying a value of six would prevent any group from splitting across displayed views. This setting can cause the total number of rows in a displayed view to exceed the specified maximum number of rows to display.

Figure 11-11
Displayed rows with default tolerance


Rows 1 through 10 of 24

				Count	Percent
Income category in thousands	Under $25	Job satisfaction	Highly dissatisfied	350	29.8%
			Somewhat dissatisfied	316	26.9%
			Neutral	250	21.3%
			Somewhat satisfied	157	13.4%
			Highly satisfied	101	8.6%
			Total	1174	100.0%
	$25 - $49	Job satisfaction	Highly dissatisfied	531	22.2%
			Somewhat dissatisfied	534	22.4%
			Neutral	542	22.7%
			Somewhat satisfied	496	20.8%

Figure 11-12
Tolerance set based on number of rows in inner row dimension group


Rows 1 through 12 of 24

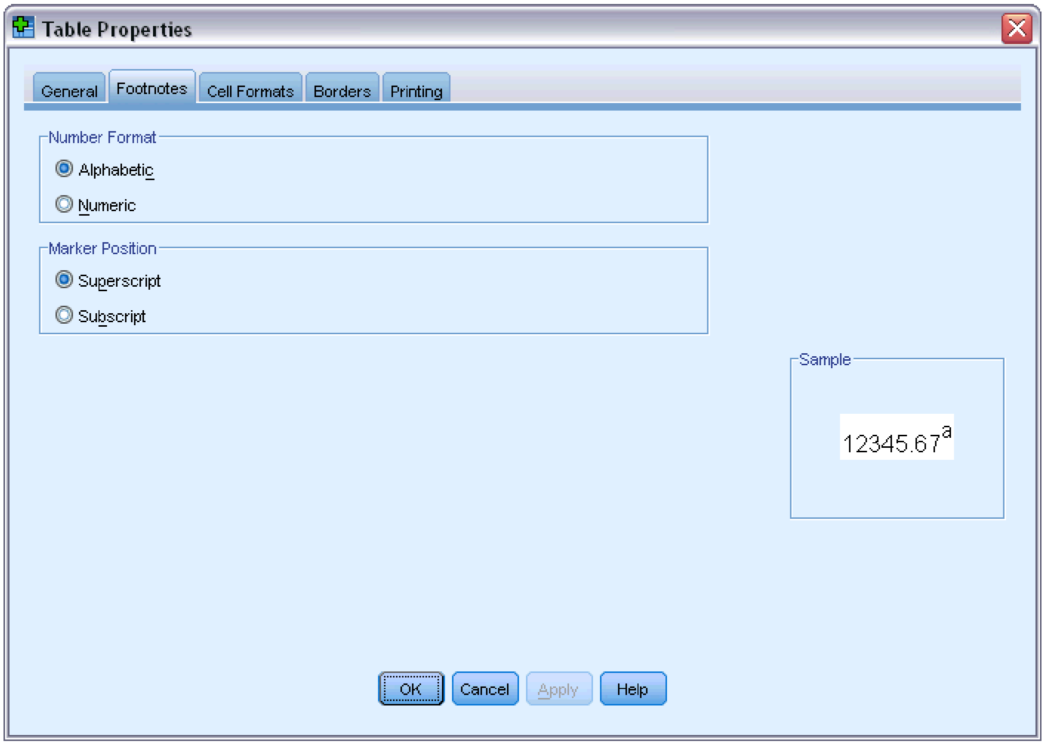
				Count	Percent
Income category in thousands	Under $25	Job satisfaction	Highly dissatisfied	350	29.8%
			Somewhat dissatisfied	316	26.9%
			Neutral	250	21.3%
			Somewhat satisfied	157	13.4%
			Highly satisfied	101	8.6%
			Total	1174	100.0%
	$25 - $49	Job satisfaction	Highly dissatisfied	531	22.2%
			Somewhat dissatisfied	534	22.4%
			Neutral	542	22.7%
			Somewhat satisfied	496	20.8%
			Highly satisfied	285	11.9%
			Total	2388	100.0%

Table Properties: Footnotes

The properties of footnote markers include style and position in relation to text.

- The style of footnote markers is either numbers (1, 2, 3, ...) or letters (a, b, c, ...).
- The footnote markers can be attached to text as superscripts or subscripts.

Figure 11-13
Table Properties dialog box, Footnotes tab



Number Format
● Alphabetic
○ Numeric

Marker Position
● Superscript
○ Subscript

Sample
12345.67[a]

To change footnote marker properties:

▶ Click the Footnotes tab.

▶ Select a footnote number format.

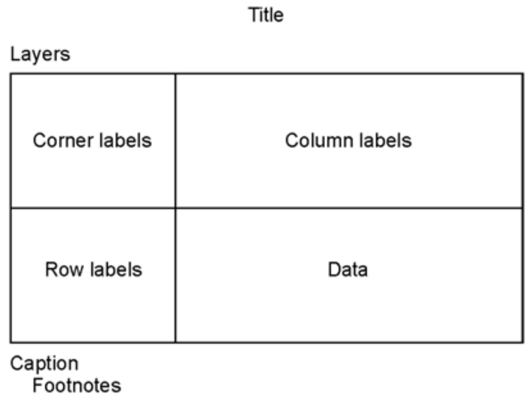
▶ Select a marker position.

▶ Click OK or Apply.

Table Properties: Cell Formats

For formatting, a table is divided into areas: title, layers, corner labels, row labels, column labels, data, caption, and footnotes. For each area of a table, you can modify the associated cell formats. Cell formats include text characteristics (such as font, size, color, and style), horizontal and vertical alignment, background colors, and inner cell margins.

Figure 11-14
Areas of a table

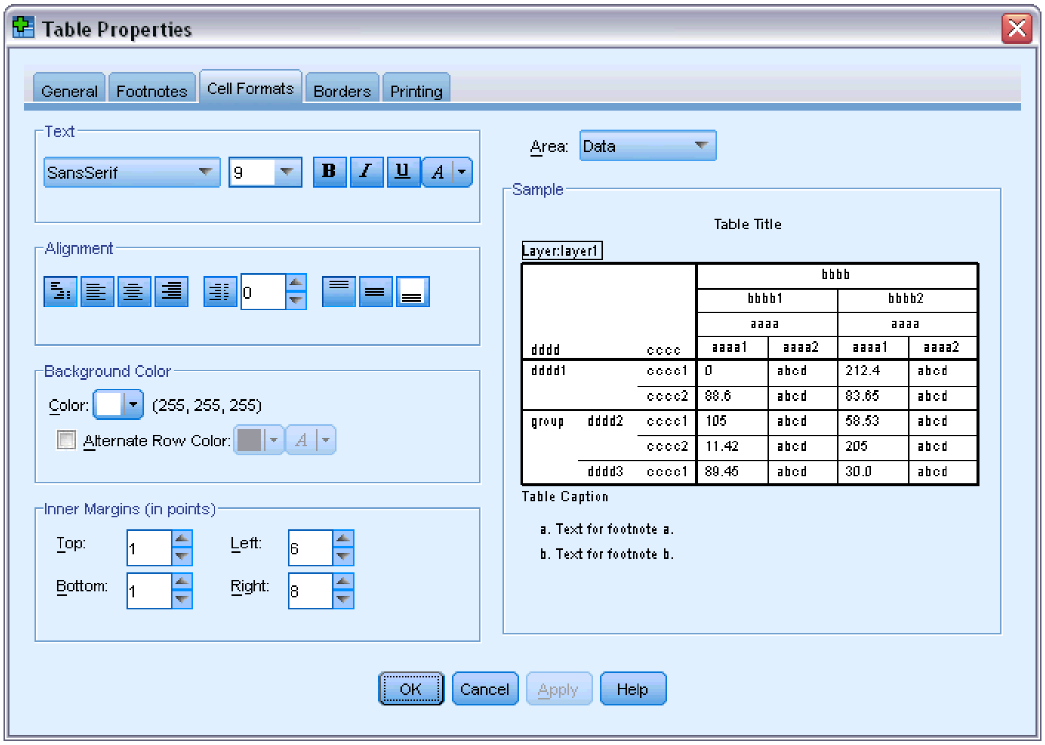


Cell formats are applied to areas (categories of information). They are not characteristics of individual cells. This distinction is an important consideration when pivoting a table.

For example,

- If you specify a bold font as a cell format of column labels, the column labels will appear bold no matter what information is currently displayed in the column dimension. If you move an item from the column dimension to another dimension, it does not retain the bold characteristic of the column labels.
- If you make column labels bold simply by highlighting the cells in an activated pivot table and clicking the Bold button on the toolbar, the contents of those cells will remain bold no matter what dimension you move them to, and the column labels will not retain the bold characteristic for other items moved into the column dimension.

Figure 11-15
Table Properties dialog box, Cell Formats tab



Text: SansSerif, 9

Alignment: 0

Background Color

Color: (255, 255, 255)

□ Alternate Row Color:

Inner Margins (in points)

Top: 1 Left: 6

Bottom: 1 Right: 8

Area: Data

Sample

Table Title

Layer:layer1

			bbbb			
			bbbb1		bbbb2	
			aaaa		aaaa	
dddd		cccc	aaaa1	aaaa2	aaaa1	aaaa2
dddd1		cccc1	0	abcd	212.4	abcd
		cccc2	88.6	abcd	83.65	abcd
group	dddd2	cccc1	105	abcd	58.53	abcd
		cccc2	11.42	abcd	205	abcd
	dddd3	cccc1	89.45	abcd	30.0	abcd

Table Caption

a. Text for footnote a.

b. Text for footnote b.

To change cell formats:

▶ Select the Cell Formats tab.

▶ Select an Area from the drop-down list or click an area of the sample.

▶ Select characteristics for the area. Your selections are reflected in the sample.

▶ Click OK or Apply.

Alternating Row Colors

To apply a different background and/or text color to alternate rows in the Data area of the table:

▶ Select Data from the Area drop-down list.

▶ Select (check) Alternate row color in the Background Color group.

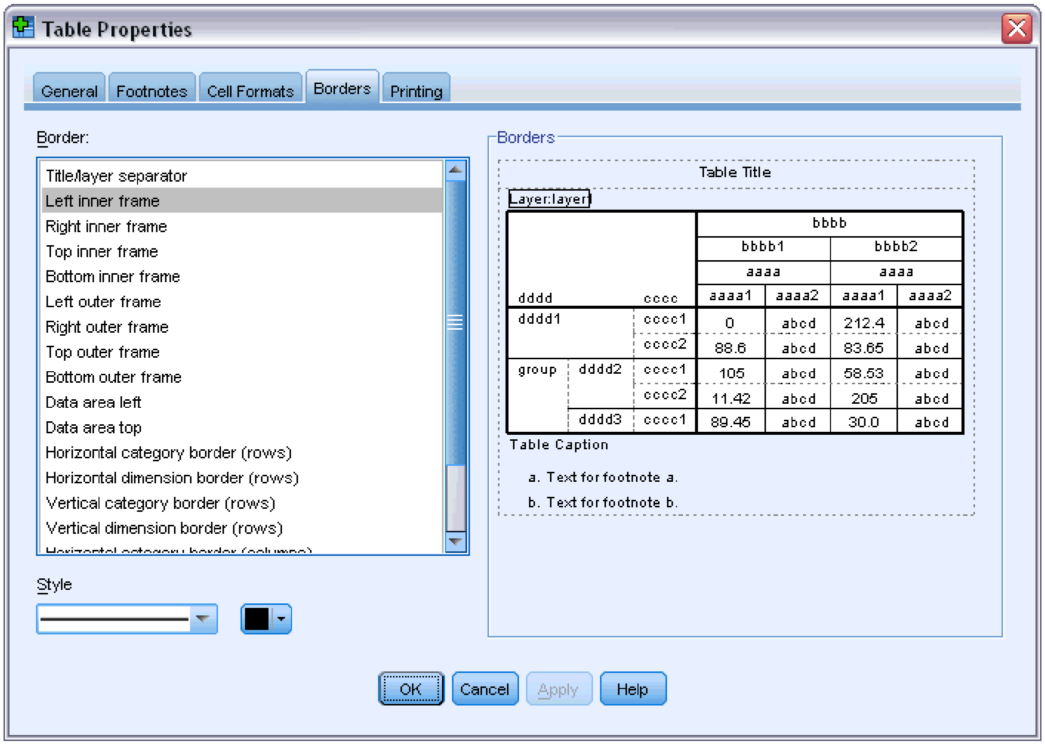
▶ Select the colors to use for the alternate row background and text.

Alternate row colors affect only the Data area of the table. They do not affect row or column label areas.

Table Properties: Borders

For each border location in a table, you can select a line style and a color. If you select None as the style, there will be no line at the selected location.

Figure 11-16
Table Properties dialog box, Borders tab



Border:
- Title/layer separator
- Left inner frame
- Right inner frame
- Top inner frame
- Bottom inner frame
- Left outer frame
- Right outer frame
- Top outer frame
- Bottom outer frame
- Data area left
- Data area top
- Horizontal category border (rows)
- Horizontal dimension border (rows)
- Vertical category border (rows)
- Vertical dimension border (rows)
- Horizontal category border (columns)

Table Title

Layer:layer

			bbbb			
			bbbb1		bbbb2	
			aaaa		aaaa	
dddd		cccc	aaaa1	aaaa2	aaaa1	aaaa2
dddd1		cccc1	0	abcd	212.4	abcd
		cccc2	88.6	abcd	83.65	abcd
group	dddd2	cccc1	105	abcd	58.53	abcd
		cccc2	11.42	abcd	205	abcd
	dddd3	cccc1	89.45	abcd	30.0	abcd

Table Caption

a. Text for footnote a.

b. Text for footnote b.

To change table borders:

- ▶ Click the Borders tab.
- ▶ Select a border location, either by clicking its name in the list or by clicking a line in the Sample area.
- ▶ Select a line style or select None.
- ▶ Select a color.
- ▶ Click OK or Apply.

Table Properties: Printing

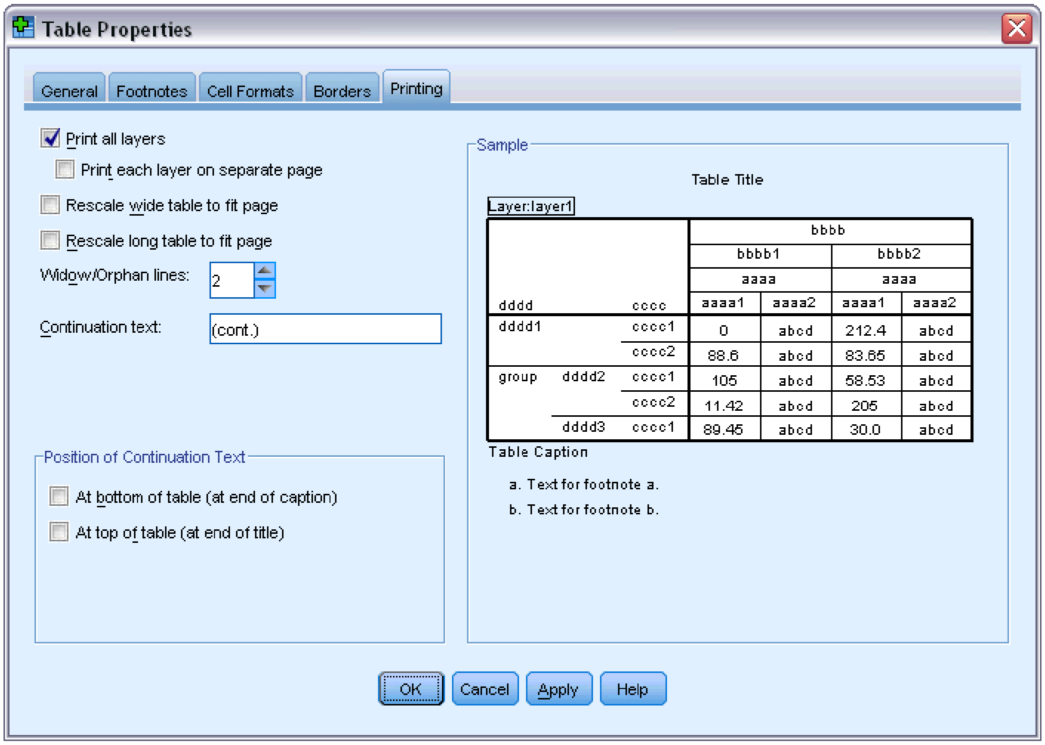
You can control the following properties for printed pivot tables:

- ■ Print all layers or only the top layer of the table, and print each layer on a separate page.

- Shrink a table horizontally or vertically to fit the page for printing.
- Control widow/orphan lines by controlling the minimum number of rows and columns that will be contained in any printed section of a table if the table is too wide and/or too long for the defined page size.

  Note: If a table is too long to fit on the current page because there is other output above it, but it will fit within the defined page length, the table is automatically printed on a new page, regardless of the widow/orphan setting.
- Include continuation text for tables that don't fit on a single page. You can display continuation text at the bottom of each page and at the top of each page. If neither option is selected, the continuation text will not be displayed.

Figure 11-17
Table Properties dialog box, Printing tab



☑ Print all layers

☐ Print each layer on separate page

☐ Rescale wide table to fit page

☐ Rescale long table to fit page

Widow/Orphan lines: 2

Continuation text: (cont.)

Position of Continuation Text

☐ At bottom of table (at end of caption)

☐ At top of table (at end of title)

Sample

Table Title

Layer:layer1

			bbbb			
			bbbb1		bbbb2	
			aaaa		aaaa	
	dddd	cccc	aaaa1	aaaa2	aaaa1	aaaa2
	dddd1	cccc1	0	abcd	212.4	abcd
		cccc2	88.6	abcd	83.65	abcd
group	dddd2	cccc1	105	abcd	58.53	abcd
		cccc2	11.42	abcd	205	abcd
	dddd3	cccc1	89.45	abcd	30.0	abcd

Table Caption

a. Text for footnote a.

b. Text for footnote b.

To control pivot table printing properties:

▶ Click the Printing tab.

▶ Select the printing options that you want.

▶ Click OK or Apply.

Cell Properties

Cell properties are applied to a selected cell. You can change the font, value format, alignment, margins, and colors. Cell properties override table properties; therefore, if you change table properties, you do not change any individually applied cell properties.

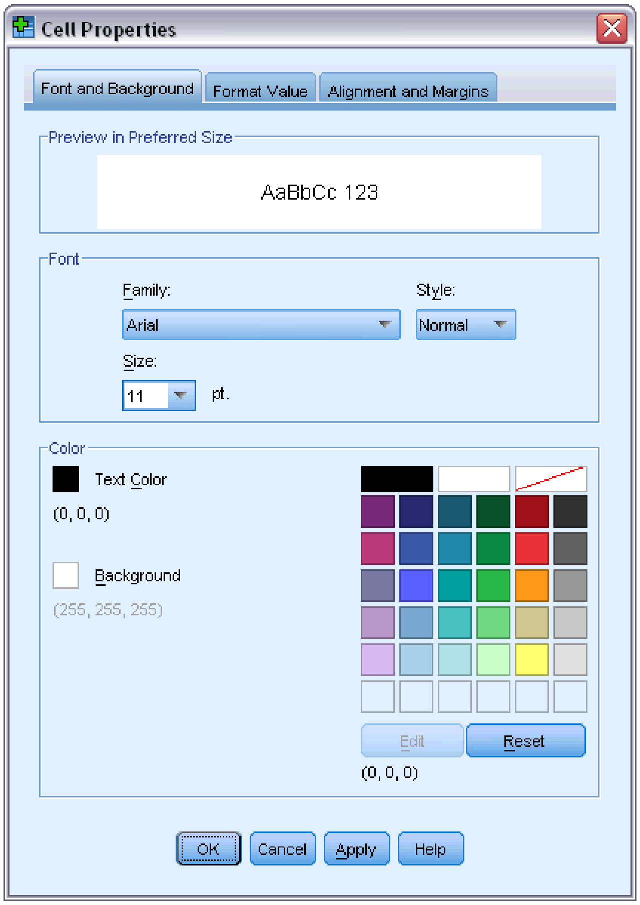
To change cell properties:

- ▶ Activate a table and select the cell(s) in the table.
- ▶ From the Format menu or the context menu choose Cell Properties.

Font and Background

The Font and Background tab controls the font style and color and background color for the selected cells in the table.

Figure 11-18
Cell Properties dialog box, Font and Background tab



Cell Properties

Font and Background | Format Value | Alignment and Margins

Preview in Preferred Size

AaBbCc 123

Font

Family: Arial

Style: Normal

Size: 11 pt.

Color

Text Color (0, 0, 0)

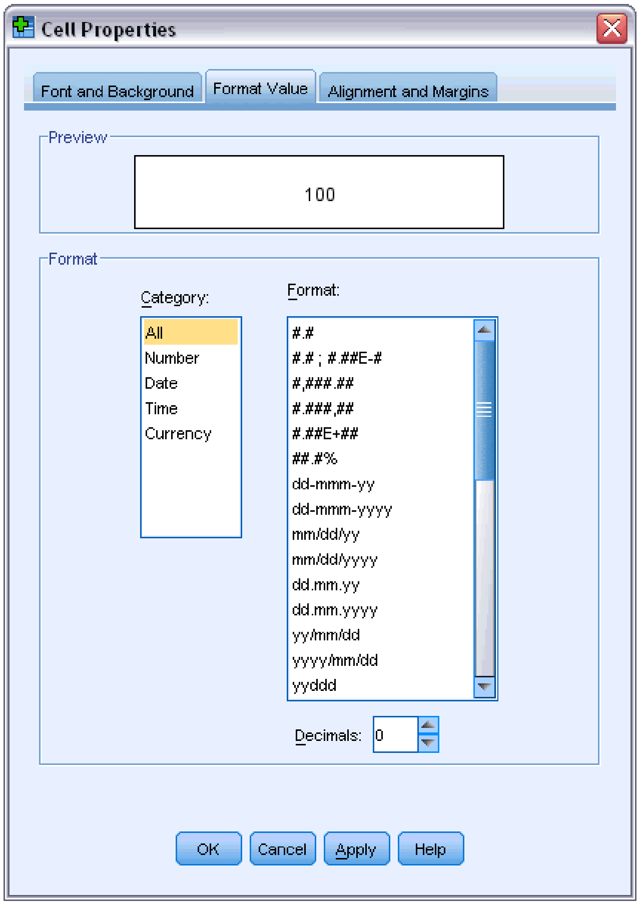
Background (255, 255, 255)

(0, 0, 0)

Format Value

The Format Value tab controls value formats for the selected cells. You can select formats for numbers, dates, time, or currencies, and you can adjust the number of decimal digits that are displayed.

Figure 11-19
Cell Properties dialog box, Format Value tab



Preview: 100

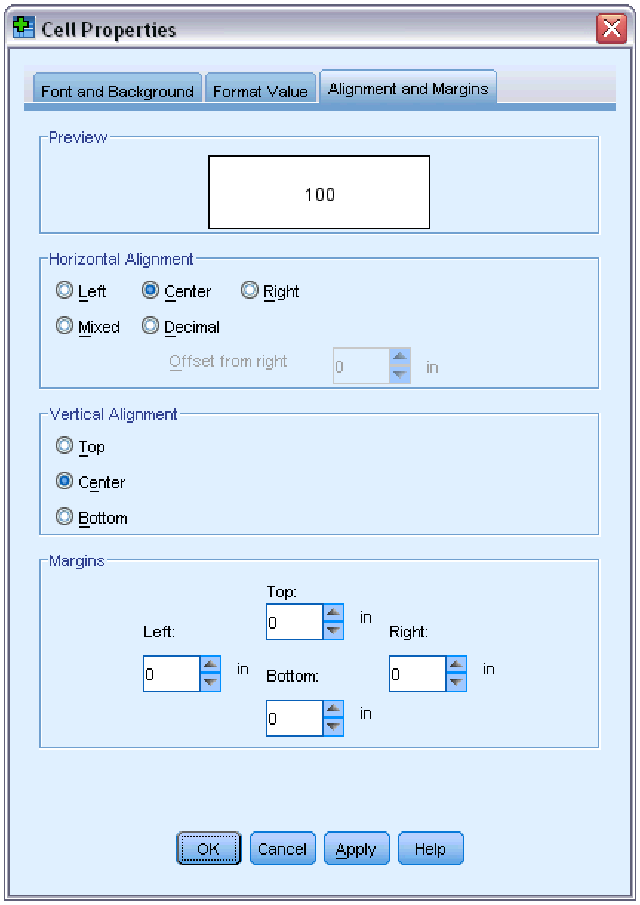
Category: All

Decimals: 0

Alignment and Margins

The Alignment and Margins tab controls horizontal and vertical alignment of values and top, bottom, left, and right margins for the selected cells. Mixed horizontal alignment aligns the content of each cell according to its type. For example, dates are right-aligned and text values are left-aligned.

Figure 11-20
Cell Properties dialog box, Alignment and Margins tab



Cell Properties

Preview: 100

Horizontal Alignment: ○ Left ● Center ○ Right ○ Mixed ○ Decimal

Offset from right: 0 in

Vertical Alignment: ○ Top ● Center ○ Bottom

Margins: Top: 0 in, Left: 0 in, Right: 0 in, Bottom: 0 in

Footnotes and Captions

You can add footnotes and captions to a table. You can also hide footnotes or captions, change footnote markers, and renumber footnotes.

Some footnote attributes are controlled by table properties. For more information, see the topic Table Properties: Footnotes on p. 240.

Adding Footnotes and Captions

To add a caption to a table:

- ▶ From the Insert menu choose Caption.

A footnote can be attached to any item in a table. To add a footnote:

- ▶ Click a title, cell, or caption within an activated pivot table.
- ▶ From the Insert menu choose Footnote.

To Hide or Show a Caption

To hide a caption:

- Select the caption.
- From the View menu choose Hide.

To show hidden captions:

- From the View menu choose Show All.

To Hide or Show a Footnote in a Table

To hide a footnote:

- Select the footnote.
- From the View menu choose Hide or from the context menu choose Hide Footnote.

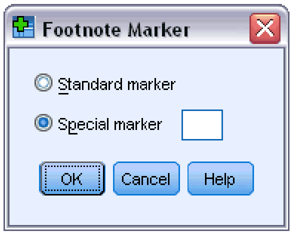
To show hidden footnotes:

- From the View menu choose Show All Footnotes.

Footnote Marker

Footnote Marker changes the character(s) that can be used to mark a footnote.

Figure 11-21
Footnote Marker dialog box



To change footnote markers:

- Select a footnote.
- From the Format menu choose Footnote Marker.
- Enter one or two characters.

Renumbering Footnotes

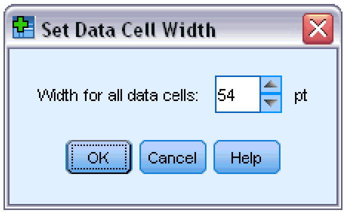
When you have pivoted a table by switching rows, columns, and layers, the footnotes may be out of order. To renumber the footnotes:

- From the Format menu choose Renumber Footnotes.

Data Cell Widths

Set Data Cell Width is used to set all data cells to the same width.

Figure 11-22
Set Data Cell Width dialog box



To set the width for all data cells:

▶ From the menus choose:

Format
  Set Data Cell Widths...

▶ Enter a value for the cell width.

Changing Column Width

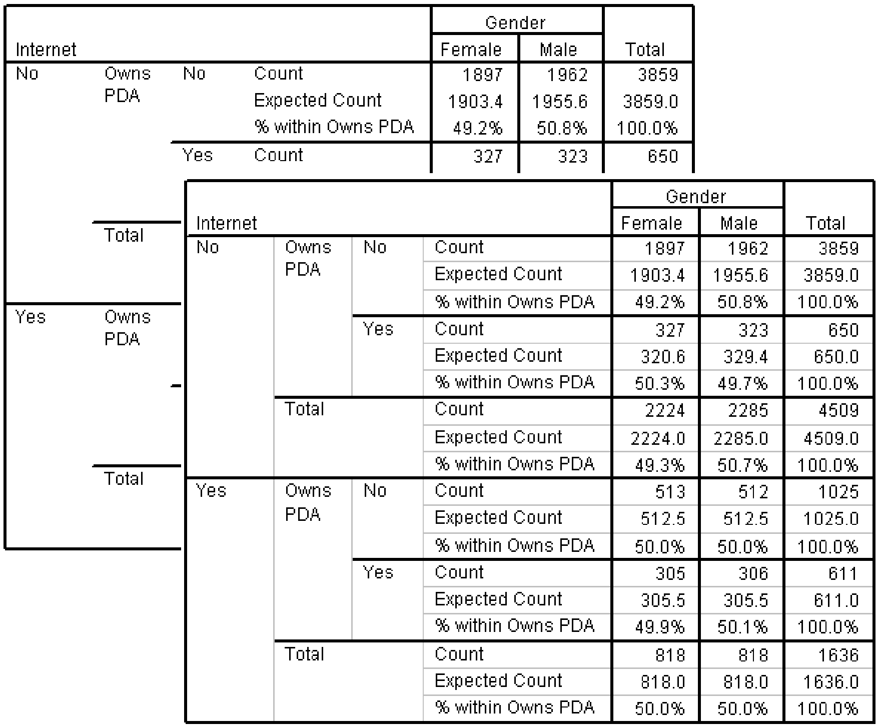
▶ Click and drag the column border.

Displaying Hidden Borders in a Pivot Table

For tables without many visible borders, you can display the hidden borders. This can simplify tasks like changing column widths.

▶ From the View menu choose Gridlines.

Figure 11-23
Gridlines displayed for hidden borders



Internet				Gender		Total
				Female	Male	
No	Owns PDA	No	Count	1897	1962	3859
			Expected Count	1903.4	1955.6	3859.0
			% within Owns PDA	49.2%	50.8%	100.0%
		Yes	Count	327	323	650
	Total					
Yes	Owns PDA					
	Total					

Internet				Gender		Total
				Female	Male	
No	Owns PDA	No	Count	1897	1962	3859
			Expected Count	1903.4	1955.6	3859.0
			% within Owns PDA	49.2%	50.8%	100.0%
		Yes	Count	327	323	650
			Expected Count	320.6	329.4	650.0
			% within Owns PDA	50.3%	49.7%	100.0%
	Total		Count	2224	2285	4509
			Expected Count	2224.0	2285.0	4509.0
			% within Owns PDA	49.3%	50.7%	100.0%
Yes	Owns PDA	No	Count	513	512	1025
			Expected Count	512.5	512.5	1025.0
			% within Owns PDA	50.0%	50.0%	100.0%
		Yes	Count	305	306	611
			Expected Count	305.5	305.5	611.0
			% within Owns PDA	49.9%	50.1%	100.0%
	Total		Count	818	818	1636
			Expected Count	818.0	818.0	1636.0
			% within Owns PDA	50.0%	50.0%	100.0%

Selecting Rows and Columns in a Pivot Table

In pivot tables, there are some constraints on how you select entire rows and columns, and the visual highlight that indicates the selected row or column may span noncontiguous areas of the table. To select an entire row or column:

▶ Click a row or column label.

▶ From the menus choose:

Edit
  Select
    Data and Label Cells

or

▶ Right-click the category label for the row or column.

▶ From the context menu choose:

Select
  Data and Label Cells

or

▶ Ctrl+Alt+click the row or column label.

Printing Pivot Tables

Several factors can affect the way that printed pivot tables look, and these factors can be controlled by changing pivot table attributes.

- For multidimensional pivot tables (tables with layers), you can either print all layers or print only the top (visible) layer. For more information, see the topic Table Properties: Printing on p. 243.
- For long or wide pivot tables, you can automatically resize the table to fit the page or control the location of table breaks and page breaks. For more information, see the topic Table Properties: Printing on p. 243.
- For tables that are too wide or too long for a single page, you can control the location of table breaks between pages.

Use Print Preview on the File menu to see how printed pivot tables will look.

Controlling Table Breaks for Wide and Long Tables

Pivot tables that are either too wide or too long to print within the defined page size are automatically split and printed in multiple sections. (For wide tables, multiple sections will print on the same page if there is room.) You can:

- Control the row and column locations where large tables are split.
- Specify rows and columns that should be kept together when tables are split.
- Rescale large tables to fit the defined page size.

To Specify Row and Column Breaks for Printed Pivot Tables

▶ Click the column label to the left of where you want to insert the break, or click the row label above where you want to insert the break.

▶ From the menus choose:

Format
  Break Here

To Specify Rows or Columns to Keep Together

▶ Select the labels of the rows or columns that you want to keep together. (Click and drag or Shift+click to select multiple row or column labels.)

▶ From the menus choose:

Format
  Keep Together

Creating a Chart from a Pivot Table

▶ Double-click the pivot table to activate it.

▶ Select the rows, columns, or cells you want to display in the chart.

- Right-click anywhere in the selected area.
- Choose Create Graph from the context menu and select a chart type.

Chapter

12

Models

Some results are presented as models, which appear in the output Viewer as a special type of visualization. The visualization displayed in the output Viewer is not the only view of the model that is available. A single model contains many different views. You can activate the model in the Model Viewer and interact with the model directly to display the available model views. You can also choose to print and export all the views in the model.

Interacting with a Model

To interact with a model, you first activate it:

▶ Double-click the model.

or

▶ Right-click the model and from the context menu choose Edit Content.

▶ From the submenu choose In Separate Window.

Activating the model displays the model in the Model Viewer. For more information, see the topic Working with the Model Viewer on p. 253.

Working with the Model Viewer

The Model Viewer is an interactive tool for displaying the available model views and editing the look of the model views. (For information about displaying the Model Viewer, see Interacting with a Model on p. 253.) The Model Viewer is split into two parts:

- **Main view**. The main view appears in the left part of the Model Viewer. The main view displays some general visualization (for example, a network graph) for the model. The main view itself may have more than one model view. The drop-down list below the main view allows you to choose from the available main views.
- **Auxiliary view**. The auxiliary view appears in the right part of the Model Viewer. The auxiliary view typically displays a more detailed visualization (including tables) of the model compared to the general visualization in the main view. Like the main view, the auxiliary view may have more than one model view. The drop-down list below the auxiliary view allows you to choose from the available main views. The auxiliary can also display specific visualizations for elements that are selected in the main view. For example, depending on the type of model, you may be able to select a variable node in the main view to display a table for that variable in the auxiliary view.

The specific visualizations that are displayed depend on the procedure that created the model. For information about working with specific models, refer to the documentation for the procedure that created the model.

Model View Tables

Tables displayed in the Model Viewer are not pivot tables. You cannot manipulate these tables as you can manipulate pivot tables.

Setting Model Properties

Within the Model Viewer, you can set specific properties for the model. For more information, see the topic Model Properties on p. 254.

Copying Model Views

You can also copy individual model views within the Model Viewer. For more information, see the topic Copying Model Views on p. 254.

Model Properties

From within the Model Viewer, choose from the menus:

File
  Properties

Each model has associated properties that let you specify which views are printed from the output Viewer. By default, only the view that is visible in the output Viewer is printed. This is always a main view, and only one main view. You can also specify that all available model views are printed. These include all the main views and all the auxiliary views (except for auxiliary views based on selection in the main view; these are not printed). Note that you can also print individual model views within the Model Viewer itself. For more information, see the topic Printing a Model on p. 254.

Copying Model Views

From the Edit menu within the Model Viewer, you can copy the currently displayed main view or the currently display auxiliary view. Only one model view is copied. You can paste the model view into the output Viewer, where the individual model view is subsequently rendered as a visualization that can be edited in the Graphboard Editor. Pasting into the output Viewer allows you to display multiple model views simultaneously. You can also paste into other applications, where the view may appear as an image or a table depending on the target application.

Printing a Model

Printing from the Model Viewer

You can print a single model view within the Model Viewer itself.

- ▶ Activate the model in the Model Viewer. For more information, see the topic Interacting with a Model on p. 253.

- ▶ From the menus choose:

  View
  Edit Mode

- ▶ On the General toolbar palette in the main or auxiliary view (depending on which one you want to print), click the print icon. (If this palette is not displayed, choose Palettes>General from the View menu.)

Printing from the Output Viewer

When you print from the output Viewer, the number of views that are printed for a specific model depend on the model's properties. The model can be set to print only the displayed view or all of the available model views. For more information, see the topic Model Properties on p. 254.

Exporting a Model

By default, when you export models from the output Viewer, inclusion or exclusion of model views is controlled by the model properties for each model. For more information about model properties, see Model Properties on p. 254. On export, you can override this setting and include all model views or only the currently visible model view. In the Export Output dialog box, click Change Options... in the Document group. For more information about exporting and this dialog box, see Export Output on p. 212. Note that all model views, including tables, are exported as graphics. Also note that auxiliary views based on selections in the main view are never exported.

Chapter

13

Working with Command Syntax

The powerful command language allows you to save and automate many common tasks. It also provides some functionality not found in the menus and dialog boxes.

Most commands are accessible from the menus and dialog boxes. However, some commands and options are available only by using the command language. The command language also allows you to save your jobs in a syntax file so that you can repeat your analysis at a later date or run it in an automated job with the a production job.

A syntax file is simply a text file that contains commands. While it is possible to open a syntax window and type in commands, it is often easier if you let the software help you build your syntax file using one of the following methods:

- Pasting command syntax from dialog boxes
- Copying syntax from the output log
- Copying syntax from the journal file

Detailed command syntax reference information is available in two forms: integrated into the overall Help system and as a separate PDF file, called the *Command Syntax Reference*, also available from the Help menu. Context-sensitive Help for the current command in a syntax window is available by pressing the F1 key.

Syntax Rules

When you run commands from a command syntax window during a session, you are running commands in **interactive** mode.

The following rules apply to command specifications in interactive mode:

- Each command must start on a new line. Commands can begin in any column of a command line and continue for as many lines as needed. The exception is the `END DATA` command, which must begin in the first column of the first line after the end of data.
- Each command should end with a period as a command terminator. It is best to omit the terminator on `BEGIN DATA`, however, so that inline data are treated as one continuous specification.
- The command terminator must be the last nonblank character in a command.
- In the absence of a period as the command terminator, a blank line is interpreted as a command terminator.

Note: For compatibility with other modes of command execution (including command files run with `INSERT` or `INCLUDE` commands in an interactive session), each line of command syntax should not exceed 256 bytes.

- Most subcommands are separated by slashes (/). The slash before the first subcommand on a command is usually optional.
- Variable names must be spelled out fully.
- Text included within apostrophes or quotation marks must be contained on a single line.
- A period (.) must be used to indicate decimals, regardless of your regional or locale settings.
- Variable names ending in a period can cause errors in commands created by the dialog boxes. You cannot create such variable names in the dialog boxes, and you should generally avoid them.

Command syntax is case insensitive, and three- or four-letter abbreviations can be used for many command specifications. You can use as many lines as you want to specify a single command. You can add space or break lines at almost any point where a single blank is allowed, such as around slashes, parentheses, arithmetic operators, or between variable names. For example,

```
FREQUENCIES
  VARIABLES=JOBCAT GENDER
  /PERCENTILES=25 50 75
  /BARCHART.
```

and

```
freq var=jobcat gender /percent=25 50 75 /bar.
```

are both acceptable alternatives that generate the same results.

INCLUDE Files

For command files run via the `INCLUDE` command, batch mode syntax rules apply.

The following rules apply to command specifications in batch mode:

- All commands in the command file must begin in column 1. You can use plus (+) or minus (–) signs in the first column if you want to indent the command specification to make the command file more readable.
- If multiple lines are used for a command, column 1 of each continuation line must be blank.
- Command terminators are optional.
- A line cannot exceed 256 bytes; any additional characters are truncated.

Unless you have existing command files that already use the `INCLUDE` command, you should probably use the `INSERT` command instead, since it can accommodate command files that conform to either set of rules. If you generate command syntax by pasting dialog box choices into a syntax window, the format of the commands is suitable for any mode of operation. See the *Command Syntax Reference* (available in PDF format from the Help menu) for more information.

Pasting Syntax from Dialog Boxes

The easiest way to build a command syntax file is to make selections in dialog boxes and paste the syntax for the selections into a syntax window. By pasting the syntax at each step of a lengthy analysis, you can build a job file that allows you to repeat the analysis at a later date or run an automated job with the Production Facility.

In the syntax window, you can run the pasted syntax, edit it, and save it in a syntax file.

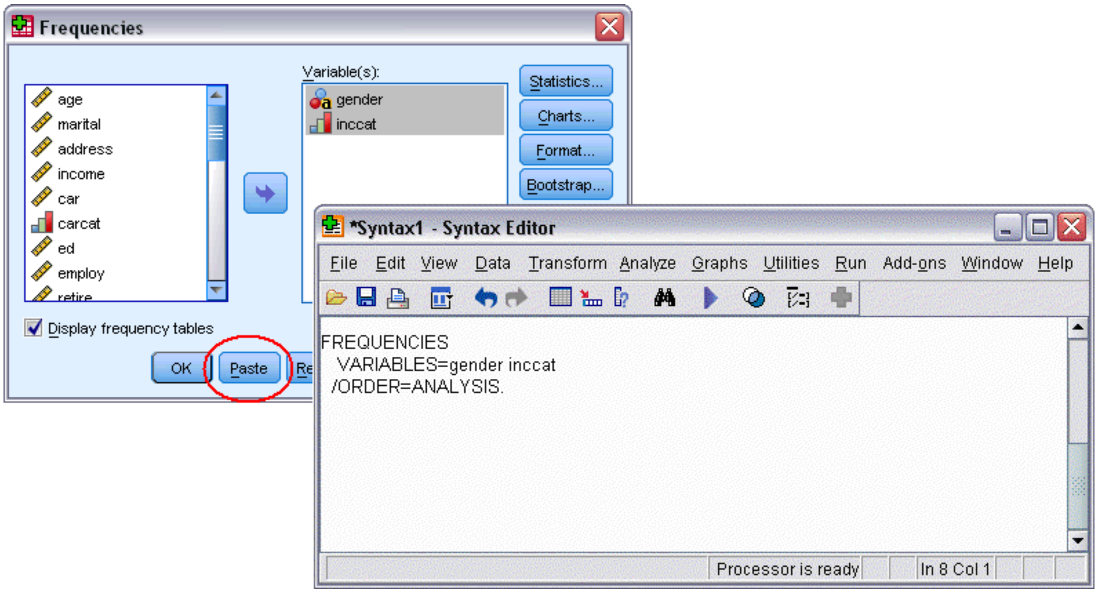
To Paste Syntax from Dialog Boxes

- Open the dialog box and make the selections that you want.

- Click Paste.

The command syntax is pasted to the designated syntax window. If you do not have an open syntax window, a new syntax window opens automatically, and the syntax is pasted there.

Figure 13-1
Command syntax pasted from a dialog box



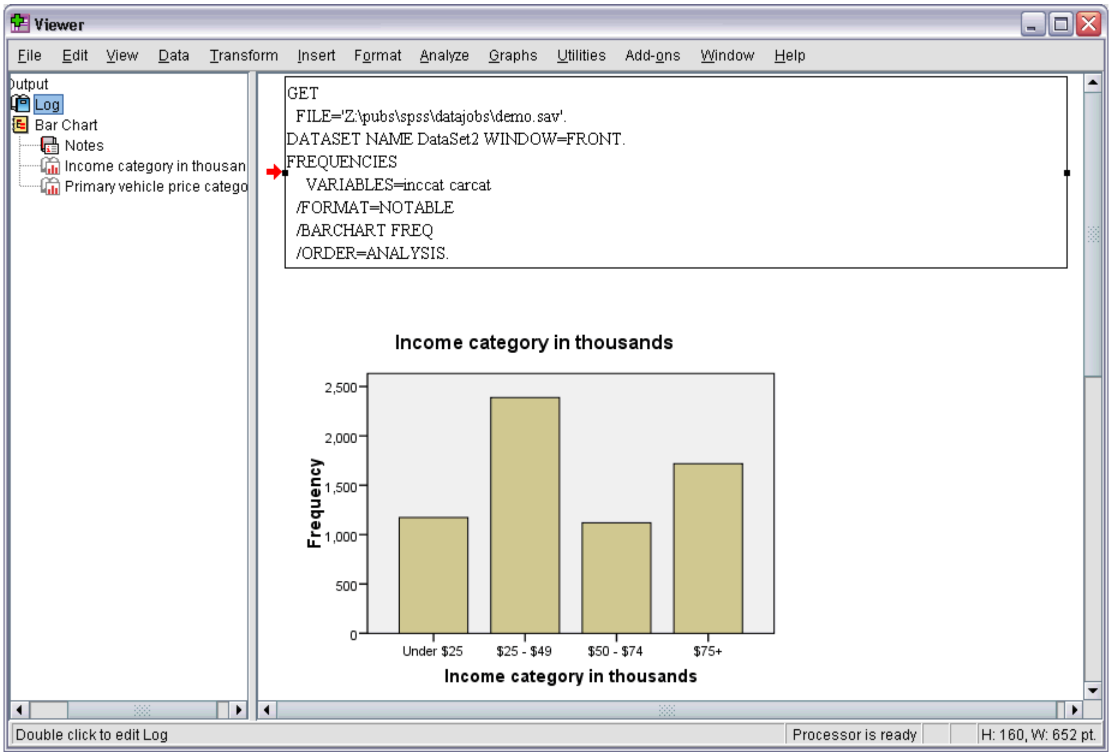
```
FREQUENCIES
  VARIABLES=gender inccat
 /ORDER=ANALYSIS.
```

Copying Syntax from the Output Log

You can build a syntax file by copying command syntax from the log that appears in the Viewer. To use this method, you must select Display commands in the log in the Viewer settings (Edit menu, Options, Viewer tab) before running the analysis. Each command will then appear in the Viewer along with the output from the analysis.

In the syntax window, you can run the pasted syntax, edit it, and save it in a syntax file.

Figure 13-2
Command syntax in the log


Viewer
File Edit View Data Transform Insert Format Analyze Graphs Utilities Add-ons Window Help
Output
Log
Bar Chart
Notes
Income category in thousan
Primary vehicle price catego
Income category in thousands
2,500
2,000
1,500
1,000
500
0
Frequency
Under $25
$25 - $49
$50 - $74
$75+
Income category in thousands
Double click to edit Log
Processor is ready
H: 160, W: 652 pt.

```
GET
  FILE='Z:\pubs\spss\datajobs\demo.sav'.
DATASET NAME DataSet2 WINDOW=FRONT.
FREQUENCIES
    VARIABLES=inccat carcat
  /FORMAT=NOTABLE
  /BARCHART FREQ
  /ORDER=ANALYSIS.
```

To Copy Syntax from the Output Log

▶ Before running the analysis, from the menus choose:

Edit
  Options...

▶ On the Viewer tab, select Display commands in the log.

As you run analyses, the commands for your dialog box selections are recorded in the log.

▶ Open a previously saved syntax file or create a new one. To create a new syntax file, from the menus choose:

File
  New
    Syntax

▶ In the Viewer, double-click a log item to activate it.

▶ Select the text that you want to copy.

▶ From the Viewer menus choose:

Edit
  Copy

▶ In a syntax window, from the menus choose:

Edit
  Paste

Using the Syntax Editor

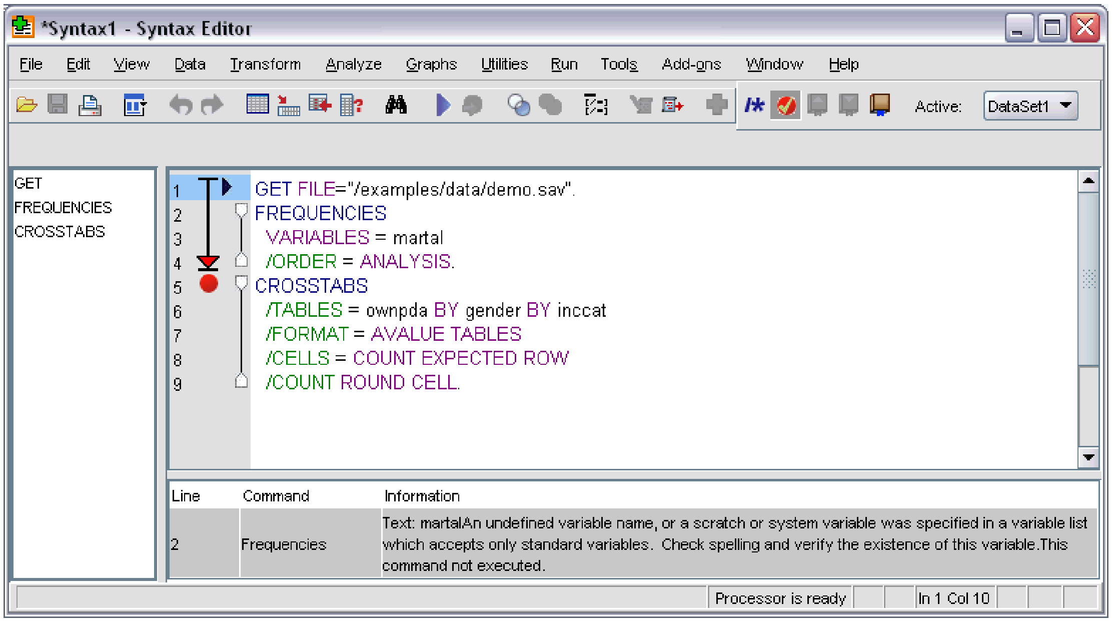
The Syntax Editor provides an environment specifically designed for creating, editing, and running command syntax. The Syntax Editor features:

- **Auto-Completion.** As you type, you can select commands, subcommands, keywords, and keyword values from a context-sensitive list. You can choose to be prompted automatically with the list or display the list on demand.
- **Color Coding.** Recognized elements of command syntax (commands, subcommands, keywords, and keyword values) are color coded so, at a glance, you can spot unrecognized terms. Also, a number of common syntactical errors—such as unmatched quotes—are color coded for quick identification.
- **Breakpoints.** You can stop execution of command syntax at specified points, allowing you to inspect the data or output before proceeding.
- **Bookmarks.** You can set bookmarks that allow you to quickly navigate large command syntax files.
- **Step Through.** You can step through command syntax one command at a time, advancing to the next command with a single click.

Note: When working with right to left languages, it is recommended to check the Optimize for right to left languages box on the Syntax Editor tab in the Options dialog box.

Syntax Editor Window

Figure 13-3
Syntax Editor



GET
FREQUENCIES
CROSSTABS

```
GET FILE="/examples/data/demo.sav".
FREQUENCIES
  VARIABLES = martal
  /ORDER = ANALYSIS.
CROSSTABS
  /TABLES = ownpda BY gender BY inccat
  /FORMAT = AVALUE TABLES
  /CELLS = COUNT EXPECTED ROW
  /COUNT ROUND CELL.
```

Line	Command	Information
2	Frequencies	Text: martalAn undefined variable name, or a scratch or system variable was specified in a variable list which accepts only standard variables. Check spelling and verify the existence of this variable.This command not executed.

Processor is ready
In 1 Col 10

The Syntax Editor window is divided into four areas:

- The editor pane is the main part of the Syntax Editor window and is where you enter and edit command syntax.
- The gutter is adjacent to the editor pane and displays information such as line numbers and breakpoint positions.
- The navigation pane is to the left of the gutter and editor pane and displays a list of all commands in the Syntax Editor window and provides single click navigation to any command.
- The error pane is below the editor pane and displays runtime errors.

Gutter Contents

Line numbers, breakpoints, bookmarks, command spans, and a progress indicator are displayed in the gutter to the left of the editor pane in the syntax window.

- Line numbers do not account for any external files referenced in INSERT and INCLUDE commands. You can show or hide line numbers by choosing View > Show Line Numbers from the menus.
- Breakpoints stop execution at specified points and are represented as a red circle adjacent to the command on which the breakpoint is set.
- Bookmarks mark specific lines in a command syntax file and are represented as a square enclosing the number (1-9) assigned to the bookmark. Hovering over the icon for a bookmark displays the number of the bookmark and the name, if any, assigned to the bookmark.

- Command spans are icons that provide visual indicators of the start and end of a command. You can show or hide command spans by choosing View > Show Command Spans from the menus.
- The progress of a given syntax run is indicated with a downward pointing arrow in the gutter, stretching from the first command run to the last command run. This is most useful when running command syntax containing breakpoints and when stepping through command syntax. For more information, see the topic Running Command Syntax on p. 267.

Navigation Pane

The navigation pane contains a list of all recognized commands in the syntax window, displayed in the order in which they occur in the window. Clicking on a command in the navigation pane positions the cursor at the start of the command.

- You can use the Up and Down arrow keys to move through the list of commands or click on a command to navigate to it. A double click will select the command.
- Command names for commands containing certain types of syntactical errors—such as unmatched quotes—are colored red and in bold text by default. For more information, see the topic Color Coding on p. 263.
- The first word of each line of unrecognized text is shown in gray.
- You can show or hide the navigation pane by choosing View > Show Navigation Pane from the menus.

Error Pane

The error pane displays runtime errors from the most previous run.

- The information for each error contains the line number on which the error occurred.
- You can use the Up and Down arrow keys to move through the list of errors.
- Clicking on an entry in the list will position the cursor on the line that generated the error.
- You can show or hide the error pane by choosing View > Show Error Pane from the menus.

Terminology

Commands. The basic unit of syntax is the command. Each command begins with the command name, which consists of one, two, or three words—for example, `DESCRIPTIVES`, `SORT CASES`, or `ADD VALUE LABELS`.

Subcommands. Most commands contain subcommands. Subcommands provide for additional specifications and begin with a forward slash followed by the name of the subcommand.

Keywords. Keywords are fixed terms that are typically used within a subcommand to specify options available for the subcommand.

Keyword Values. Keywords can have values such as a fixed term that specifies an option or a numeric value.

Example

```
CODEBOOK gender jobcat salary
  /VARINFO VALUELABELS MISSING
  /OPTIONS VARORDER=MEASURE.
```

- The name of the command is `CODEBOOK`.
- `VARINFO` and `OPTIONS` are subcommands.
- `VALUELABELS`, `MISSING`, and `VARORDER` are keywords.
- `MEASURE` is a keyword value associated with `VARORDER`.

Auto-Completion

The Syntax Editor provides assistance in the form of auto-completion of commands, subcommands, keywords, and keyword values. By default, you are prompted with a context-sensitive list of available terms. You can display the list on demand by pressing Ctrl+Spacebar and you can close the list by pressing the Esc key.

The Auto Complete menu item on the Tools menu toggles the automatic display of the auto-complete list on or off. You can also enable or disable automatic display of the list from the Syntax Editor tab in the Options dialog box. Toggling the Auto Complete menu item overrides the setting on the Options dialog but does not persist across sessions.

Note: The auto-completion list will close if a space is entered. For commands consisting of multiple words—such as `ADD FILES`—select the desired command before entering any spaces.

Color Coding

The Syntax Editor color codes recognized elements of command syntax, such as commands and subcommands, as well as various syntactical errors like unmatched quotes or parentheses. Unrecognized text is not color coded.

Commands. By default, recognized commands are colored blue and in bold text. If, however, there is a recognized syntactical error within the command—such as a missing parenthesis—the command name is colored red and in bold text by default.

Note: Abbreviations of command names—such as `FREQ` for `FREQUENCIES`—are not colored, but such abbreviations are valid.

Subcommands. Recognized subcommands are colored green by default. If, however, the subcommand is missing a required equals sign or an invalid equals sign follows it, the subcommand name is colored red by default.

Keywords. Recognized keywords are colored maroon by default. If, however, the keyword is missing a required equals sign or an invalid equals sign follows it, the keyword is colored red by default.

Keyword values. Recognized keyword values are colored pink by default. User-specified values of keywords such as integers, real numbers, and quoted strings are not color coded.

Comments. Text within a comment is colored gray.

Syntactical Errors. Text associated with the following syntactical errors is colored red by default.

- **Unmatched Parentheses, Brackets, and Quotes.** Unmatched parentheses and brackets within comments and quoted strings are not detected. Unmatched single or double quotes within quoted strings are syntactically valid.

  Certain commands contain blocks of text that are not command syntax—such as `BEGIN DATA-END DATA`, `BEGIN GPL-END GPL`, and `BEGIN PROGRAM-END PROGRAM`. Unmatched values are not detected within such blocks.
- **Long lines.** Long lines are lines containing more than 251 characters.
- **End statements.** Several commands require either an `END` statement prior to the command terminator (for example, `BEGIN DATA-END DATA`) or require a matching `END` command at some point later in the command stream (for example, `LOOP-END LOOP`). In both cases, the command will be colored red, by default, until the required `END` statement is added.

From the Syntax Editor tab in the Options dialog box, you can change default colors and text styles and you can turn color coding off or on. You can also turn color coding of commands, subcommands, keywords, and keyword values off or on by choosing Tools > Color Coding from the menus. You can turn color coding of syntactical errors off or on by choosing Tools > Validation. Choices made on the Tools menu override settings in the Options dialog box but do not persist across sessions.

Note: Color coding of command syntax within macros is not supported.

Breakpoints

Breakpoints allow you to stop execution of command syntax at specified points within the syntax window and continue execution when ready.

- Breakpoints are set at the level of a command and stop execution prior to running the command.
- Breakpoints cannot occur within `LOOP-END LOOP`, `DO IF-END IF`, `DO REPEAT-END REPEAT`, `INPUT PROGRAM-END INPUT PROGRAM`, and `MATRIX-END MATRIX` blocks. They can, however, be set at the beginning of such blocks and will stop execution prior to running the block.
- Breakpoints cannot be set on lines containing non-PASW Statistics command syntax, such as occur within `BEGIN PROGRAM-END PROGRAM`, `BEGIN DATA-END DATA`, and `BEGIN GPL-END GPL` blocks.
- Breakpoints are not saved with the command syntax file and are not included in copied text.
- By default, breakpoints are honored during execution. You can toggle whether breakpoints are honored or not from Tools > Honor Breakpoints.

To Insert a Breakpoint

▶ Click anywhere in the gutter to the left of the command text.

or

▶ Position the cursor within the desired command.

▶ From the menus choose:

Tools
  Toggle Breakpoint

The breakpoint is represented as a red circle in the gutter to the left of the command text and on the same line as the command name.

Clearing Breakpoints

To clear a single breakpoint:

▶ Click on the icon representing the breakpoint in the gutter to the left of the command text.

or

▶ Position the cursor within the desired command.

▶ From the menus choose:

Tools
  Toggle Breakpoint

To clear all breakpoints:

▶ From the menus choose:

Tools
  Clear All Breakpoints

See Running Command Syntax on p. 267 for information about the run-time behavior in the presence of breakpoints.

Bookmarks

Bookmarks allow you to quickly navigate to specified positions in a command syntax file. You can have up to 9 bookmarks in a given file. Bookmarks are saved with the file, but are not included when copying text.

To Insert a Bookmark

▶ Position the cursor on the line where you want to insert the bookmark.

▶ From the menus choose:

Tools
  Toggle Bookmark

The new bookmark is assigned the next available number, from 1 to 9. It is represented as a square enclosing the assigned number and displayed in the gutter to the left of the command text.

Clearing Bookmarks

To clear a single bookmark:

▶ Position the cursor on the line containing the bookmark.

▶ From the menus choose:

Tools
  Toggle Bookmark

To clear all bookmarks:

▶ From the menus choose:

Tools
  Clear All Bookmarks

Renaming a Bookmark

You can associate a name with a bookmark. This is in addition to the number (1-9) assigned to the bookmark when it was created.

▶ From the menus choose:

Tools
  Rename Bookmark

▶ Enter a name for the bookmark and click OK.

The specified name replaces any existing name for the bookmark.

Navigating with Bookmarks

To navigate to the next or previous bookmark:

▶ From the menus choose:

Tools
  Next Bookmark

or

Tools
  Previous Bookmark

To navigate to a specific bookmark:

▶ From the menus choose:

Tools
  Go To Bookmark

▶ Select the desired bookmark.

Commenting Out Text

You can comment out entire commands as well as text that is not recognized as command syntax.

▶ Select the desired text. When selecting multiple commands, a command will be commented out if any part of it is selected.

▶ From the menus choose:

Tools
  Comment Selection

You can comment out a single command by positioning the cursor anywhere within the command and clicking the Comment Selection button on the toolbar.

Running Command Syntax

▶ Highlight the commands that you want to run in the syntax window.

▶ Click the Run button (the right-pointing triangle) on the Syntax Editor toolbar. It runs the selected commands or the command where the cursor is located if there is no selection.

or

▶ Choose one of the items from the Run menu.

- **All.** Runs all commands in the syntax window, honoring any breakpoints.
- **Selection.** Runs the currently selected commands, honoring any breakpoints. This includes any partially highlighted commands. If there is no selection, the command where the cursor is positioned is run.
- **To End.** Runs all commands starting from the first command in the current selection to the last command in the syntax window, honoring any breakpoints. If nothing is selected, the run starts from the command where the cursor is positioned.
- **Step Through.** Runs the command syntax one command at a time starting from the first command in the syntax window (Step Through From Start) or from the command where the cursor is positioned (Step Through From Current). If there is selected text, the run starts from the first command in the selection. After a given command has run, the cursor advances to the next command and you continue the step through sequence by choosing Continue.

  `LOOP-END LOOP`, `DO IF-END IF`, `DO REPEAT-END REPEAT`, `INPUT PROGRAM-END INPUT PROGRAM`, and `MATRIX-END MATRIX` blocks are treated as single commands when using Step Through. You can not step into one of these blocks.
- **Continue.** Continues a run stopped by a breakpoint or Step Through.

Progress Indicator

The progress of a given syntax run is indicated with a downward pointing arrow in the gutter, spanning the last set of commands run. For instance, you choose to run all commands in a syntax window that contains breakpoints. At the first breakpoint, the arrow will span the region from the first command in the window to the command prior to the one containing the breakpoint. At the second breakpoint, the arrow will stretch from the command containing the first breakpoint to the command prior to the one containing the second breakpoint.

Run-time Behavior with Breakpoints

- When running command syntax containing breakpoints, execution stops at each breakpoint. Specifically, the block of command syntax from a given breakpoint (or beginning of the run) to the next breakpoint (or end of the run) is submitted for execution exactly as if you had selected that syntax and chosen Run > Selection.

- You can work with multiple syntax windows, each with its own set of breakpoints, but there is only one queue for executing command syntax. Once a block of command syntax has been submitted—such as the block of command syntax up to the first breakpoint—no other block of command syntax will be executed until the previous block has completed, regardless of whether the blocks are in the same or different syntax windows.
- With execution stopped at a breakpoint, you can run command syntax in other syntax windows, and inspect Data Editor or Viewer windows. However, modifying the contents of the syntax window containing the breakpoint or changing the cursor position in that window will cancel the run.

Unicode Syntax Files

In Unicode mode, the default format for saving command syntax files created or modified during the session is also Unicode (UTF-8). Unicode-format command syntax files cannot be read by versions of PASW Statistics prior to 16.0. For more information on Unicode mode, see General Options on p. 291.

To save a syntax file in a format compatible with earlier releases:

▶ From the syntax window menus, choose:

File
  Save As

▶ In the Save As dialog, from the Encoding drop-down list, choose Local Encoding. The local encoding is determined by the current locale.

Multiple Execute Commands

Syntax pasted from dialog boxes or copied from the log or the journal may contain `EXECUTE` commands. When you run commands from a syntax window, `EXECUTE` commands are generally unnecessary and may slow performance, particularly with larger data files, because each `EXECUTE` command reads the entire data file. For more information, see the `EXECUTE` command in the *Command Syntax Reference* (available from the Help menu in any PASW Statistics window).

Lag Functions

One notable exception is transformation commands that contain lag functions. In a series of transformation commands without any intervening `EXECUTE` commands or other commands that read the data, lag functions are calculated after all other transformations, regardless of command order. For example,

```
COMPUTE lagvar=LAG(var1).
COMPUTE var1=var1*2.
```

and

```
COMPUTE lagvar=LAG(var1).
```

```
EXECUTE.
COMPUTE var1=var1*2.
```

yield very different results for the value of *lagvar*, since the former uses the transformed value of *var1* while the latter uses the original value.

Chapter

14

Overview of the Chart Facility

High-resolution charts and plots are created by the procedures on the Graphs menu and by many of the procedures on the Analyze menu. This chapter provides an overview of the chart facility.

Building and Editing a Chart

Before you can create a chart, you need to have your data in the Data Editor. You can enter the data directly into the Data Editor; open a previously saved data file; or read a spreadsheet, tab-delimited data file, or database file. The Tutorial selection on the Help menu has online examples of creating and modifying a chart, and the online Help system provides information about creating and modifying all chart types.

Building Charts

The Chart Builder allows you to build charts from predefined gallery charts or from the individual parts (for example, axes and bars). You build a chart by dragging and dropping the gallery charts or basic elements onto the **canvas**, which is the large area to the right of the Variables list in the Chart Builder dialog box.

As you are building the chart, the canvas displays a preview of the chart. Although the preview uses defined variable labels and measurement levels, it does not display your actual data. Instead, it uses randomly generated data to provide a rough sketch of how the chart will look.

Using the gallery is the preferred method for new users. For information about using the gallery, see Building a Chart from the Gallery on p. 271.

How to Start the Chart Builder

▶ From the menus choose:

Graphs
  Chart Builder

This opens the Chart Builder dialog box.

Figure 14-1
Chart Builder dialog box

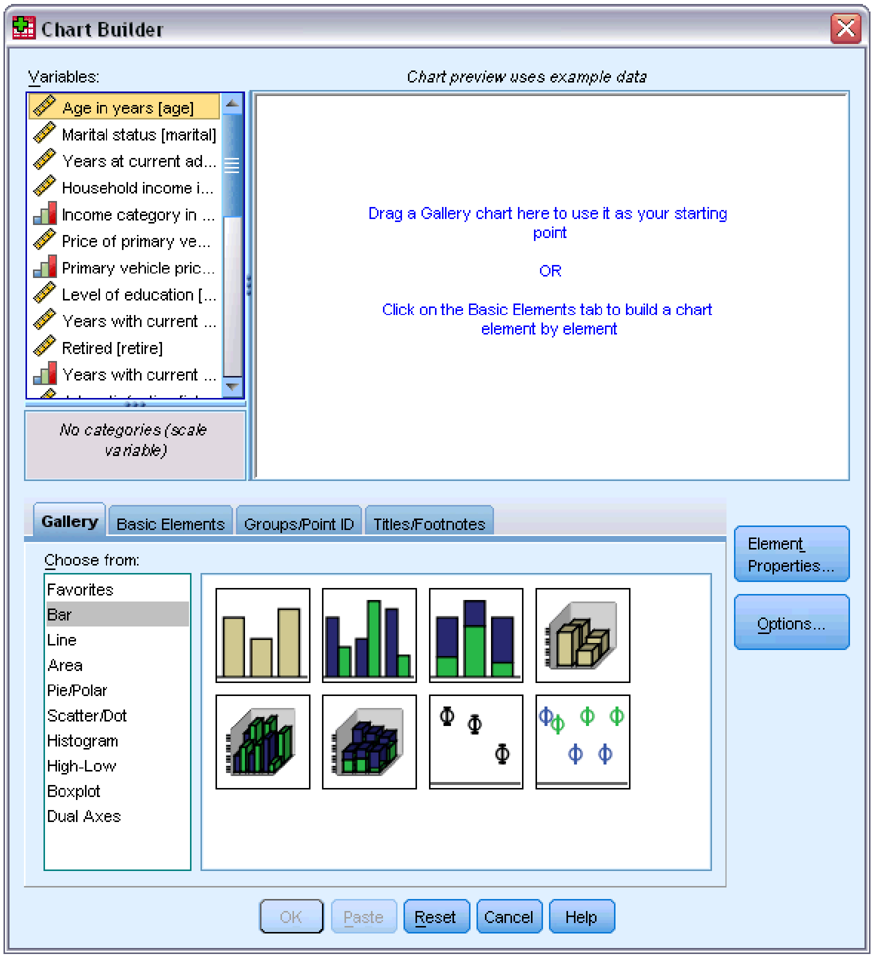


Chart preview uses example data

Drag a Gallery chart here to use it as your starting point

OR

Click on the Basic Elements tab to build a chart element by element

No categories (scale variable)

Building a Chart from the Gallery

The easiest method for building charts is to use the gallery. Following are general steps for building a chart from the gallery.

- Click the Gallery tab if it is not already displayed.

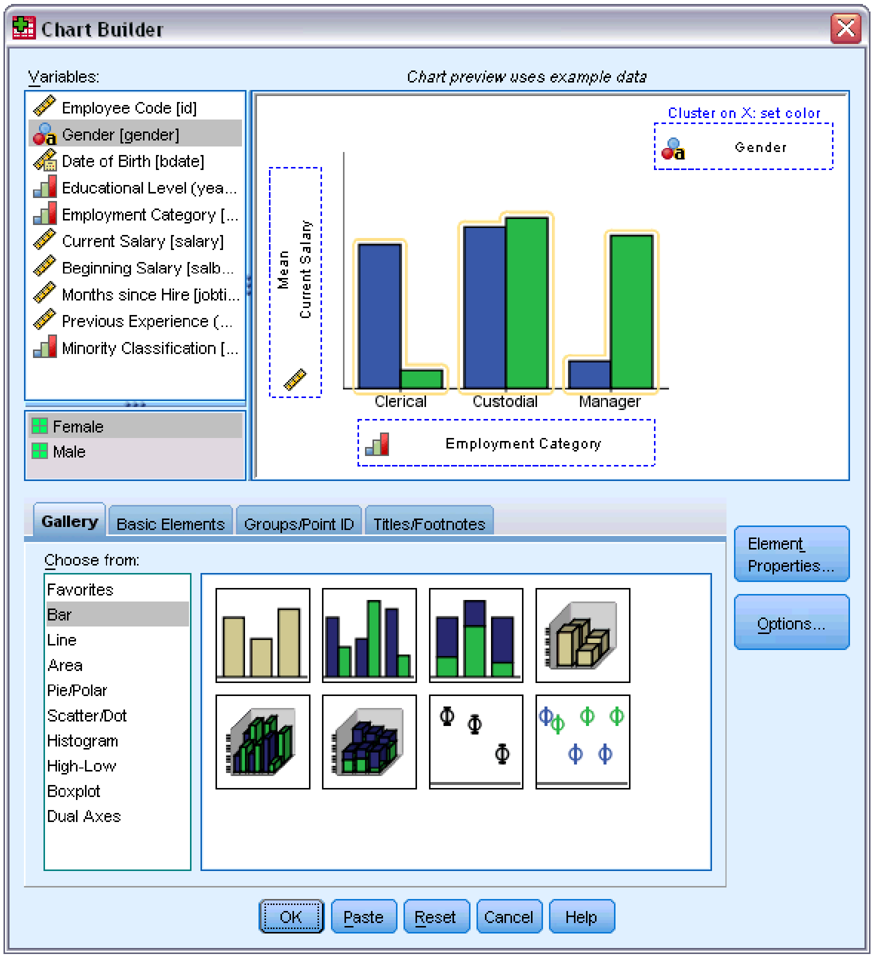
- In the Choose From list, select a category of charts. Each category offers several types.

- Drag the picture of the desired chart onto the canvas. You can also double-click the picture. If the canvas already displays a chart, the gallery chart replaces the axis set and graphic elements on the chart.

- Drag variables from the Variables list and drop them into the axis drop zones and, if available, the grouping drop zone. If an axis drop zone already displays a statistic and you want to use that statistic, you do not have to drag a variable into the drop zone. You need to add a variable to a zone only when the text in the zone is blue. If the text is black, the zone already contains a variable or statistic.

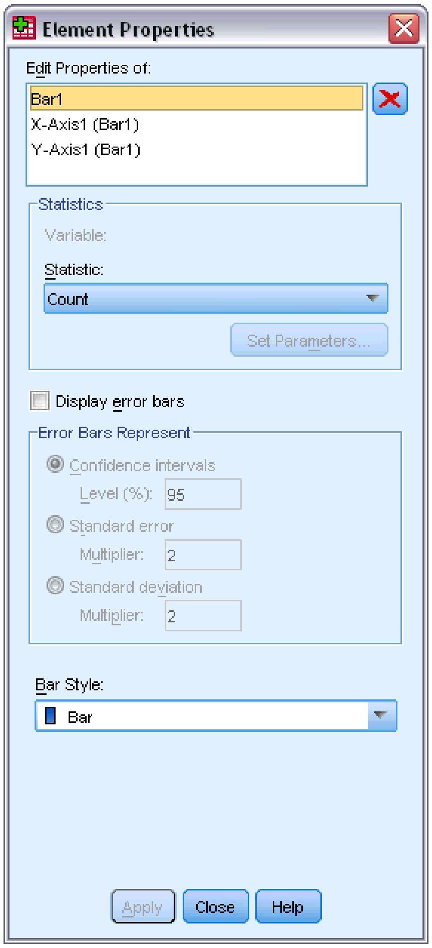
Note: The measurement level of your variables is important. The Chart Builder sets defaults based on the measurement level while you are building the chart. Furthermore, the resulting chart may also look different for different measurement levels. You can temporarily change a variable's measurement level by right-clicking the variable and choosing an option.

Figure 14-2
Chart Builder dialog box with completed drop zones



▶ If you need to change statistics or modify attributes of the axes or legends (such as the scale range), click Element Properties.

Figure 14-3
Element Properties dialog box



Edit Properties of:
Bar1
X-Axis1 (Bar1)
Y-Axis1 (Bar1)

Variable:
Statistic: Count

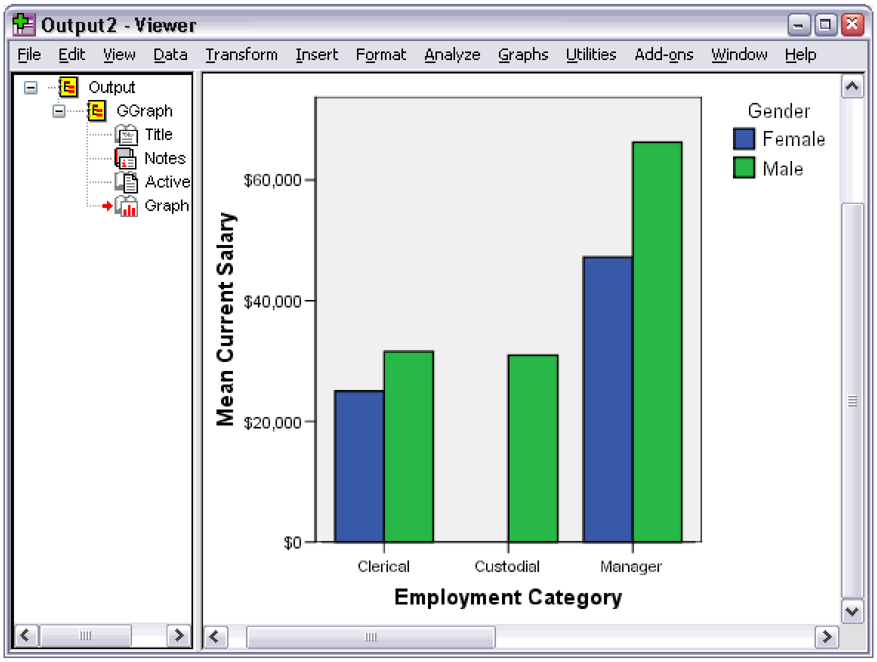
□ Display error bars

● Confidence intervals
Level (%): 95
○ Standard error
Multiplier: 2
○ Standard deviation
Multiplier: 2

Bar Style: Bar

- ▶ In the Edit Properties Of list, select the item you want to change. (For information about the specific properties, click Help.)
- ▶ After making any changes, click Apply.
- ▶ If you need to add more variables to the chart (for example, for clustering or paneling), click the Groups/Point ID tab in the Chart Builder dialog box and select one or more options. Then drag categorical variables to the new drop zones that appear on the canvas.
- ▶ If you want to transpose the chart (for example, to make the bars horizontal), click the Basic Elements tab and then click Transpose.
- ▶ Click OK to create the chart. The chart is displayed in the Viewer.

Figure 14-4
Bar chart displayed in Viewer window



Editing Charts

The Chart Editor provides a powerful, easy-to-use environment where you can customize your charts and explore your data. The Chart Editor features:

- **Simple, intuitive user interface.** You can quickly select and edit parts of the chart using menus, context menus, and toolbars. You can also enter text directly on a chart.
- **Wide range of formatting and statistical options.** You can choose from a full range of styles and statistical options.
- **Powerful exploratory tools.** You can explore your data in various ways, such as by labeling, reordering, and rotating it. You can change chart types and the roles of variables in the chart. You can also add distribution curves and fit, interpolation, and reference lines.
- **Flexible templates for consistent look and behavior.** You can create customized templates and use them to easily create charts with the look and options that you want. For example, if you always want a specific orientation for axis labels, you can specify the orientation in a template and apply the template to other charts.

How to View the Chart Editor

▶ Create a chart in PASW Statistics, or open a Viewer file with charts.

▶ Double-click a chart in the Viewer.

The chart is displayed in the Chart Editor.

Figure 14-5
Chart displayed in the Chart Editor

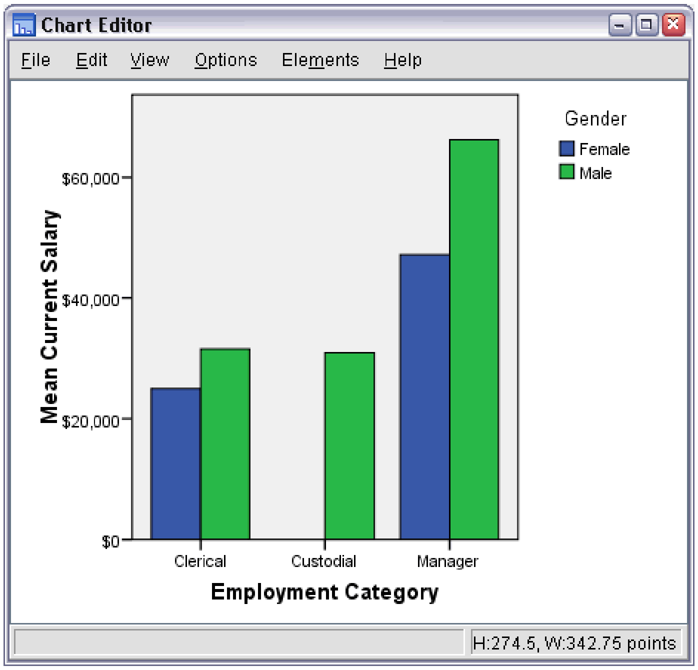


Chart Editor Fundamentals

The Chart Editor provides various methods for manipulating charts.

Menus

Many actions that you can perform in the Chart Editor are done with the menus, especially when you are adding an item to the chart. For example, you use the menus to add a fit line to a scatterplot. After adding an item to the chart, you often use the Properties dialog box to specify options for the added item.

Properties Dialog Box

Options for the chart and its chart elements can be found in the Properties dialog box.

To view the Properties dialog box, you can:

▶ Double-click a chart element.

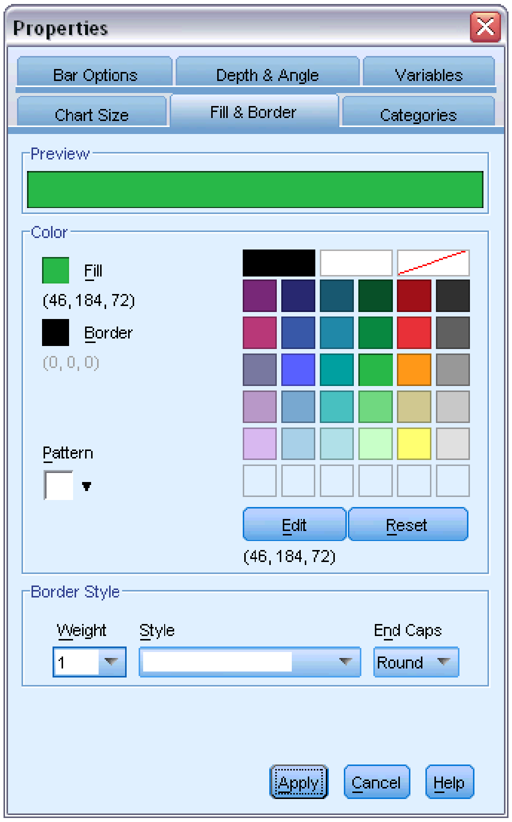
or

▶ Select a chart element, and then from the menus choose:

Edit
  Properties

Additionally, the Properties dialog box automatically appears when you add an item to the chart.

Figure 14-6
Properties dialog box, Fill & Border tab



Fill (46, 184, 72)
Border (0, 0, 0)
(46, 184, 72)

Weight: 1
Style:
End Caps: Round

The Properties dialog box has tabs that allow you to set the options and make other changes to a chart. The tabs that you see in the Properties dialog box are based on your current selection.

Some tabs include a preview to give you an idea of how the changes will affect the selection when you apply them. However, the chart itself does not reflect your changes until you click Apply. You can make changes on more than one tab before you click Apply. If you have to change the selection to modify a different element on the chart, click Apply before changing the selection. If you do not click Apply before changing the selection, clicking Apply at a later point will apply changes only to the element or elements currently selected.

Depending on your selection, only certain settings will be available. The help for the individual tabs specifies what you need to select to view the tabs. If multiple elements are selected, you can change only those settings that are common to all the elements.

Toolbars

The toolbars provide a shortcut for some of the functionality in the Properties dialog box. For example, instead of using the Text tab in the Properties dialog box, you can use the Edit toolbar to change the font and style of the text.

Saving the Changes

Chart modifications are saved when you close the Chart Editor. The modified chart is subsequently displayed in the Viewer.

Chart Definition Options

When you are defining a chart in the Chart Builder, you can add titles and change options for the chart creation.

Adding and Editing Titles and Footnotes

You can add titles and footnotes to the chart to help a viewer interpret it. The Chart Builder also automatically displays error bar information in the footnotes.

How to Add Titles and Footnotes

- ▶ Click the Titles/Footnotes tab.
- ▶ Select one or more titles and footnotes. The canvas displays some text to indicate that these were added to the chart.
- ▶ Use the Element Properties dialog box to edit the title/footnote text.

How to Remove a Title or Footnote

- ▶ Click the Titles/Footnotes tab.
- ▶ Deselect the title or footnote that you want to remove.

How to Edit the Title or Footnote Text

When you add titles and footnotes, you cannot edit their associated text directly on the chart. As with other items in the Chart Builder, you edit them using the Element Properties dialog box.

- ▶ Click Element Properties if the Element Properties dialog box is not displayed.
- ▶ In the Edit Properties Of list, select a title, subtitle, or footnote (for example, Title 1).
- ▶ In the content box, type the text associated with the title, subtitle, or footnote.
- ▶ Click Apply.

Setting General Options

The Chart Builder offers general options for the chart. These are options that apply to the overall chart, rather than a specific item on the chart. General options include missing value handling, templates, chart size, and panel wrapping.

- ▶ Click Options.

▶ Modify the general options. Details about these follow.

▶ Click Apply.

User-Missing Values

Break Variables. If there are missing values for the variables used to define categories or subgroups, select Include so that the category or categories of user-missing values (values identified as missing by the user) are included in the chart. These "missing" categories also act as break variables in calculating the statistic. The "missing" category or categories are displayed on the category axis or in the legend, adding, for example, an extra bar or a slice to a pie chart. If there are no missing values, the "missing" categories are not displayed.

If you select this option and want to suppress display after the chart is drawn, open the chart in the Chart Editor and choose Properties from the Edit menu. Use the Categories tab to move the categories that you want to suppress to the Excluded list. Note, however, that the statistics are not recalculated if you hide the "missing" categories. Therefore, something like a percent statistic will still take the "missing" categories into account.

Note: This control does not affect system-missing values. These are always excluded from the chart.

Summary Statistics and Case Values. You can choose one of the following alternatives for exclusion of cases having missing values:

- **Exclude listwise to obtain a consistent base for the chart.** If any of the variables in the chart has a missing value for a given case, the whole case is excluded from the chart.
- **Exclude variable-by-variable to maximize the use of the data.** If a selected variable has any missing values, the cases having those missing values are excluded when the variable is analyzed.

To see the difference between listwise and variable-by-variable exclusion of missing values, consider the following figures, which show a bar chart for each of the two options.

Figure 14-7
Listwise exclusion of missing values

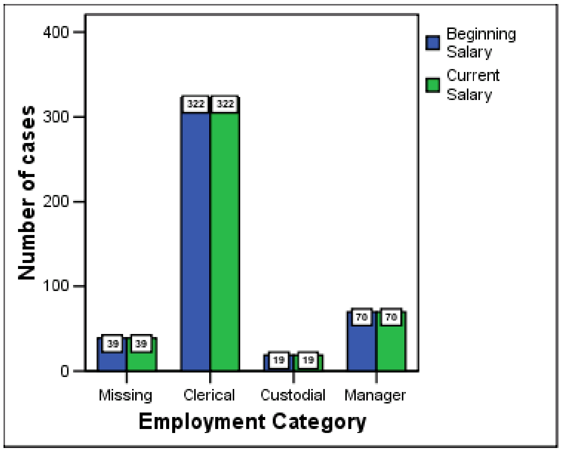
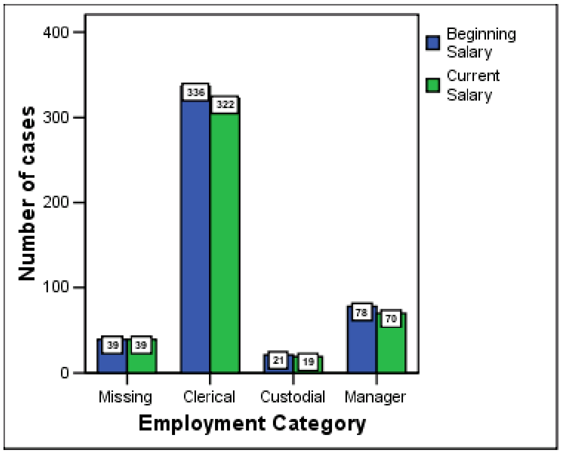


Figure 14-8
Variable-by-variable exclusion of missing values



The data include some system-missing (blank) values in the variables for *Current Salary* and *Employment Category*. In some other cases, the value 0 was entered and defined as missing. For both charts, the option Display groups defined by missing values is selected, which adds the category *Missing* to the other displayed job categories. In each chart, the values of the summary function, *Number of cases*, are displayed in the bar labels.

In both charts, 26 cases have a system-missing value for the job category, and 13 cases have the user-missing value (0). In the listwise chart, the number of cases is the same for both variables in each bar cluster because whenever a value was missing, the case was excluded for all variables. In the variable-by-variable chart, the number of nonmissing cases for each variable in a category is plotted without regard to missing values in other variables.

Templates

A chart template allows you to apply the attributes of one chart to another. When you open a chart in the Chart Editor, you can save it as a template. You can then apply that template by specifying it at creation or later by applying it in the Chart Editor.

Default Template. This is the template specified by the Options. You can access these by choosing Options from the Edit menu in the Data Editor and then clicking the Charts tab. The default template is applied first, which means that the other templates can override it.

Template Files. Click Add to specify one or more templates with the standard file selection dialog box. These are applied in the order in which they appear. Therefore, templates at the end of the list can override templates at the beginning of the list.

Chart Size and Panels

Chart Size. Specify a percentage greater than 100 to enlarge the chart or less than 100 to shrink it. The percentage is relative to the default chart size.

Panels. When there are many panel columns, select Wrap Panels to allow panels to wrap across rows rather than being forced to fit in a specific row. Unless this option is selected, the panels are shrunk to force them to fit in a row.

Chapter

15

Utilities

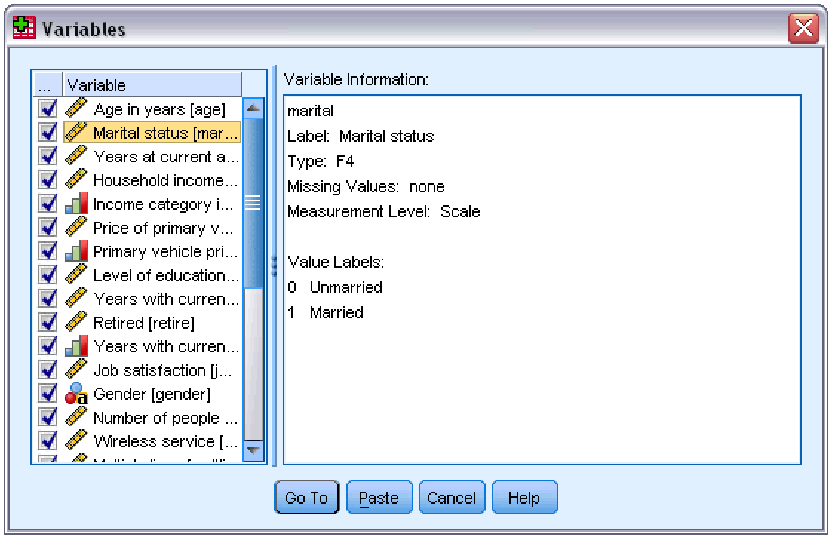
This chapter describes the functions found on the Utilities menu and the ability to reorder target variable lists.

Variable Information

The Variables dialog box displays variable definition information for the currently selected variable, including:

- Variable label
- Data format
- User-missing values
- Value labels
- Measurement level

Figure 15-1
Variables dialog box



marital
Label: Marital status
Type: F4
Missing Values: none
Measurement Level: Scale

Value Labels:
0 Unmarried
1 Married

Visible. The Visible column in the variable list indicates if the variable is currently visible in the Data Editor and in dialog box variable lists. Visibility is controlled by variable sets. For more information, see the topic Variable Sets on p. 281.

Go To. Goes to the selected variable in the Data Editor window.

Paste. Pastes the selected variables into the designated syntax window at the cursor location.

To modify variable definitions, use the Variable view in the Data Editor.

To Obtain Variable Information

- From the menus choose:

  Utilities
    Variables...

- Select the variable for which you want to display variable definition information.

Data File Comments

You can include descriptive comments with a data file. For PASW Statistics data files, these comments are saved with the data file.

To Add, Modify, Delete, or Display Data File Comments

- From the menus choose:

  Utilities
    Data File Comments...

- To display the comments in the Viewer, select Display comments in output.

Comments can be any length but are limited to 80 bytes (typically 80 characters in single-byte languages) per line; lines will automatically wrap at 80 characters. Comments are displayed in the same font as text output to accurately reflect how they will appear when displayed in the Viewer.

A date stamp (the current date in parentheses) is automatically appended to the end of the list of comments whenever you add or modify comments. This may lead to some ambiguity concerning the dates associated with comments if you modify an existing comment or insert a new comment between existing comments.

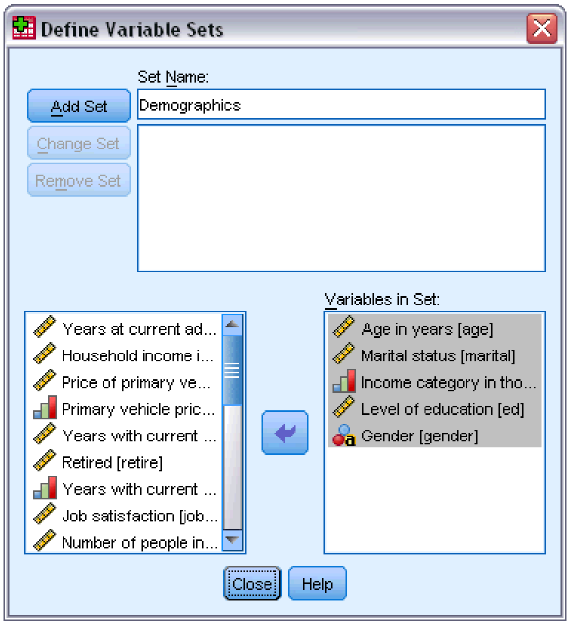
Variable Sets

You can restrict the variables that are displayed in the Data Editor and in dialog box variable lists by defining and using variable sets. This is particularly useful for data files with a large number of variables. Small variable sets make it easier to find and select the variables for your analysis.

Define Variable Sets

Define Variable Sets creates subsets of variables to display in the Data Editor and in dialog box variable lists. Defined variable sets are saved with PASW Statistics data files.

Figure 15-2
Define Variable Sets dialog box



Set Name: Demographics

Years at current ad...
Household income i...
Price of primary ve...
Primary vehicle pric...
Years with current ...
Retired [retire]
Years with current ...
Job satisfaction [job...
Number of people in...

Variables in Set:
Age in years [age]
Marital status [marital]
Income category in tho...
Level of education [ed]
Gender [gender]

Set Name. Set names can be up to 64 bytes. Any characters, including blanks, can be used.

Variables in Set. Any combination of numeric and string variables can be included in a set. The order of variables in the set has no effect on the display order of the variables in the Data Editor or in dialog box variable lists. A variable can belong to multiple sets.

To Define Variable Sets

- From the menus choose:

  Utilities
    Define Variable Sets...

- Select the variables that you want to include in the set.

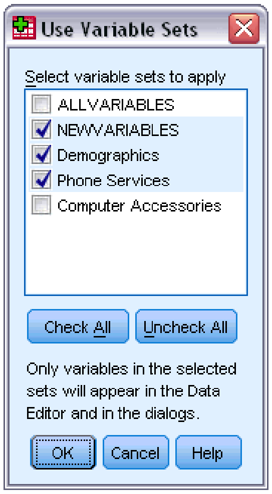
- Enter a name for the set (up to 64 bytes).

- Click Add Set.

Using Variable Sets to Show and Hide Variables

Use Variable Sets restricts the variables displayed in the Data Editor and in dialog box variable lists to the variables in the selected (checked) sets.

Figure 15-3
Use Variable Sets dialog box



Select variable sets to apply

- ☐ ALLVARIABLES
- ☑ NEWVARIABLES
- ☑ Demographics
- ☑ Phone Services
- ☐ Computer Accessories

Only variables in the selected sets will appear in the Data Editor and in the dialogs.

- The set of variables displayed in the Data Editor and in dialog box variable lists is the union of all selected sets.
- A variable can be included in multiple selected sets.
- The order of variables in the selected sets and the order of selected sets have no effect on the display order of variables in the Data Editor or in dialog box variable lists.
- Although the defined variable sets are saved with PASW Statistics data files, the list of currently selected sets is reset to the default, built-in sets each time you open the data file.

The list of available variable sets includes any variable sets defined for the active dataset, plus two built-in sets:

- **ALLVARIABLES.** This set contains all variables in the data file, including new variables created during a session.
- **NEWVARIABLES.** This set contains only new variables created during the session.

  Note: Even if you save the data file after creating new variables, the new variables are still included in the *NEWVARIABLES* set until you close and reopen the data file.

At least one variable set must be selected. If *ALLVARIABLES* is selected, any other selected sets will not have any visible effect, since this set contains all variables.

To Select Variable Sets to Display

▶ From the menus choose:

Utilities
  Use Variable Sets...

▶ Select the defined variable sets that contain the variables that you want to appear in the Data Editor and in dialog box variable lists.

To Display All Variables

▶ From the menus choose:

Utilities
  Show All Variables

Reordering Target Variable Lists

Variables appear on dialog box target lists in the order in which they are selected from the source list. If you want to change the order of variables on a target list—but you don't want to deselect all of the variables and reselect them in the new order—you can move variables up and down on the target list using the Ctrl key (Macintosh: Command key) with the up and down arrow keys. You can move multiple variables simultaneously if they are contiguous (grouped together). You cannot move noncontiguous groups of variables.

Working with Extension Bundles

Extension bundles provide the ability to package custom components, such as custom dialogs and extension commands, so they can be easily installed by end users. For example, if you have created a custom dialog for an extension command and wish to share this with other users, then you'll want to create an extension bundle that contains the custom dialog package (*.spd*) file, and the files associated with the extension command (the XML file that specifies the syntax of the extension command and the implementation code file(s) written in Python or R). The extension bundle is then what you share with other users.

Creating Extension Bundles

How to Create an Extension Bundle

▶ From the menus choose:

Utilities
  Extension Bundles
    Create Extension Bundle...

▶ Enter values for all fields on the Required tab.

▶ Enter values for any fields on the Optional tab that are needed for your extension bundle.

▶ Specify a target file for the extension bundle.

▶ Click Save to save the extension bundle to the specified location. This closes the Create Extension Bundle dialog box.

Required Fields for Extension Bundles

Name. A unique name to associate with the extension bundle. It can consist of up to three words and is not case sensitive. Characters are restricted to seven-bit ASCII. To minimize the possibility of name conflicts, you may want to use a multi-word name, where the first word is an identifier for your organization, such as a URL.

Files. Click Add to add the files associated with the extension bundle. An extension bundle must at least include a custom dialog specification (*.spd*) file, or an XML specification file for an extension command. If an XML specification file is included then the bundle must include at least one python or R code file—specifically, a file of type .py, pyc, .pyo, or .R.

Summary. A short description of the extension bundle, intended to be displayed as a single line.

Description. A more detailed description of the extension bundle than that provided for the Summary field. For example, you might list the major features available with the extension bundle. If the extension bundle provides a wrapper for a function from an R package, then that should be mentioned here.

Author. The author of the extension bundle. You may wish to include an email address.

Version. A version identifier of the form x.x.x, where each component of the identifier must be an integer, as in 1.0.0. Zeros are implied if not provided. For example, a version identifier of 3.1 implies 3.1.0.

Minimum PASW Statistics Version. The minimum version of PASW Statistics required to run the extension bundle.

Optional Fields for Extension Bundles

Release Date. An optional release date. No formatting is provided.

Links. An optional set of URL's to associate with the extension bundle—for example, the author's home page. The format of this field is arbitrary so be sure to delimit multiple URL's with spaces, commas, or some other reasonable delimiter.

Categories. An optional set of categories with which to associate the extension bundle when authoring an extension bundle for posting to Developer Central (*http://www.spss.com/devcentral*). Enter one or more of the values that appear in the Filter Downloads drop-down list on the Download Library page on Developer Central. For example, you might enter *Extension Commands*, *Statistics*, and *R*.

Required Plug-ins. Check the boxes for any Plug-ins (Python or R) that are required in order to run the custom components associated with the extension bundle. Users will be alerted at install time if they don't have the required Plug-ins.

Required R Packages. Enter the names of any R packages, from the CRAN package repository, that are required for the extension bundle. Names are case sensitive. To add the first package, click anywhere in the Required R Packages control to highlight the entry field. Pressing Enter, with the cursor in a given row, will create a new row. You delete a row by selecting it and pressing Delete. When the extension bundle is installed, PASW Statistics will check if the required R packages exist on the end user's machine and attempt to download and install any that are missing.

Required Python Modules. Enter the names of any Python modules, other than those added to the extension bundle, that are required for the extension bundle. Any such modules should be posted to Developer Central (*http://www.spss.com/devcentral*). To add the first module, click anywhere in the Required Python Modules control to highlight the entry field. Pressing Enter, with the cursor in a given row, will create a new row. You delete a row by selecting it and pressing Delete. The end user is responsible for downloading any required Python modules and copying them to the *extensions* directory under the PASW Statistics installation directory, or to an alternate location specified by their *SPSS_EXTENSIONS_PATH* environment variable.

Translation Catalogues Folder. You can include a folder containing translation catalogues created with the GNU gettext utilities (*http://www.gnu.org/software/gettext/*). This allows you to provide localized messages and localized output to end users of your extension bundle. For example, if you have stored translation catalogues for your extension bundle under the */myextbundle/po* folder, then browse to that folder and select it. Translation catalogues are for use in localizing output from Python and R programs. For more information, see the topics on the R Integration Plug-in and the Python Integration Plug-in, in the Help system.

Installing Extension Bundles

To install an extension bundle:

▶ From the menus choose:

Utilities
  Extension Bundles
    Install Extension Bundle...

▶ Select the desired extension bundle. Extension bundles have a file type of *spe*.

If the extension bundle contains a custom dialog, you will be alerted that you need to restart PASW Statistics to use the dialog. A restart is also required to use any extension commands included with the bundle. To view details for the currently installed extension bundles, choose Utilities>Extension Bundles>View Installed Extension Bundles.

You can also install extension bundles with a command line utility that allows you to install multiple bundles at once. For more information, see the topic Batch Installation of Extension Bundles on p. 288.

Installation Locations for Extension Bundles

For Windows and Linux, and by default, installing an extension bundle requires write permission to the PASW Statistics installation directory. If you do not have write permission to the required location or would like to store extension bundles elsewhere, you can specify one or more alternate locations by defining the *SPSS_EXTENSIONS_PATH* environment variable. When present, the paths specified in *SPSS_EXTENSIONS_PATH* take precedence over the default location. Extension bundles will be installed to the first writable location. For multiple locations, separate each with a semicolon on Windows and a colon on Linux. The specified locations must exist on the target machine. After setting *SPSS_EXTENSIONS_PATH* , you will need to restart PASW Statistics for the changes to take effect.

If the extension bundle contains a custom dialog and you don't have write permission to the PASW Statistics installation directory (for Windows and Linux) then you will also need to set the *SPSS_CDIALOGS_PATH* environment variable. The rules for *SPSS_CDIALOGS_PATH* are the same as for *SPSS_EXTENSIONS_PATH*. If you don't know whether the extension bundles you are installing contain a custom dialog, and you don't have write access to the default location, then it is best to specify an alternate location in *SPSS_CDIALOGS_PATH*.

To create the *SPSS_EXTENSIONS_PATH* or *SPSS_CDIALOGS_PATH* environment variable on Windows, from the Control Panel:

Windows XP

- ▶ Select System.
- ▶ Select the Advanced tab and click Environment Variables.
- ▶ In the User variables section, click New, enter the name of the environment variable—for instance, *SPSS_EXTENSIONS_PATH* —in the Variable name field and the desired path(s) in the Variable value field.

Windows Vista or Windows 7

- ▶ Select User Accounts.
- ▶ Select Change my environment variables.
- ▶ Click New, enter the name of the environment variable—for instance, *SPSS_EXTENSIONS_PATH* —in the Variable name field and the desired path(s) in the Variable value field.

To view the current locations for extension bundles (the same locations as for extension commands) and custom dialogs, run the following command syntax: `SHOW EXTPATHS`.

Required R Packages

The extension bundle installer attempts to download and install any R packages that are required by the extension bundle and not found on your machine. If you do not have internet access, you will need to obtain the necessary packages from someone who does. Packages can be downloaded from *http://www.r-project.org/* and then installed from within R. For details, see the *R Installation and Administration* guide, distributed with R. If installation of the packages fails, you will be alerted with the list of required packages. You can also view the list from the Extension Bundle Details dialog box, once the bundle is installed. For more information, see the topic Viewing Installed Extension Bundles on p. 288.

Note: For UNIX (including Linux) users, packages are downloaded in source form and then compiled. This requires that you have the appropriate tools installed on your machine. See the *R Installation and Administration* guide for details. In particular, Debian users should install the `r-base-dev` package from `apt-get install r-base-dev`.

Permissions

By default, required R packages are installed to the *library* folder under the location where R is installed—for example, *C:\Program Files\R\R-2.8.1\library* on Windows. If you do not have write permission to this location or would like to store R packages—installed for extension bundles—elsewhere, you can specify one or more alternate locations by defining the *PASW_RPACKAGES_PATH* environment variable. When present, the paths specified in *PASW_RPACKAGES_PATH* take precedence over the default location. R packages will be installed to the first writable location. For multiple locations, separate each with a semicolon on Windows and a colon on Linux and Mac. The specified locations must exist on the target machine. After setting *PASW_RPACKAGES_PATH*, you will need to restart PASW Statistics for the changes to take effect. For information on how to set an environment variable on Windows, see Installation Locations for Extension Bundles on p. 286.

Batch Installation of Extension Bundles

You can install multiple extension bundles at once using the batch utility *installextbundles.bat* (*installextbundles.sh* for Mac and UNIX systems) located in the PASW Statistics installation directory. For Windows and Mac the utility is located at the root of the installation directory. For Linux, the utility is located in the *bin* subdirectory of the installation directory. The utility is designed to run from a command prompt and must be run from its installed location. The form of the command line is:

```
installextbundles [-statssrv] [-download no|yes ] -source <folder> | <filename>...
```

-statssrv. Specifies that you are running the utility on a PASW Statistics Server. You should also install the same extension bundles on client machines that connect to the server.

-download no|yes. Specifies whether the utility has permission to access the internet in order to download any R packages required by the specified extension bundles. The default is `no`. If you keep the default or do not have internet access then you will need to manually install any required R packages. For more information, see the topic Required R Packages on p. 287.

–source <folder> | <filename>... Specifies the extension bundles to install. You can specify the path to a folder containing extension bundles or you can specify a list of filenames of extension bundles. If you specify a folder, all extension bundles (files of type *.spe*) found in the folder will be installed. Separate multiple filenames with one or more spaces. Enclose paths with double quotes if they contain spaces.

Viewing Installed Extension Bundles

To view details for the extension bundles installed on your machine:

▶ From the menus choose:

Utilities
  Extension Bundles
    View Installed Extension Bundles...

▶ Click on the highlighted text in the *Summary* column for the desired extension bundle.

The Extension Bundle Details dialog box displays the information provided by the author of the extension bundle. In addition to required information, such as Summary, Description, and Version, the author may have included URL's to locations of relevance, such as the author's home page. An extension bundle may or may not include a Readme file.

Components. The Components group lists the custom dialog, if any, and the names of any extension commands included in the extension bundle. Extension commands included with the bundle can be run from the syntax editor in the same manner as built-in PASW Statistics commands. Help for an extension command may be available by running `CommandName /HELP` in the syntax editor. Help for a custom dialog is available if the dialog contains a Help button.

Note: Installing an extension bundle that contains a custom dialog requires a restart of PASW Statistics in order to see the entry for the dialog in the Components group.

Dependencies. The Dependencies group lists add-ons that are required to run the components included in the extension bundle.

- **Python and R Plug-ins.** The components for an extension bundle may require the Python Plug-in and/or the R Plug-in, both of which are freely available. For details, see How to Get Integration Plug-ins, under the Python Integration Plug-in or R Integration Plug-in topic in the Help system.
- **R Packages.** Lists any R packages required by the extension bundle. During installation of the extension bundle, the installer attempts to download and install the necessary packages on your machine. If this process fails, you will be alerted and will need to manually install the packages. For more information, see the topic Required R Packages on p. 287.
- **Python Modules.** Lists any Python modules required by the extension bundle. Any such modules should be available from Developer Central at *http://www.spss.com/devcentral*. The modules should be copied to the *extensions* directory under the PASW Statistics installation directory. For Mac, the installation directory refers to the *Contents* directory in the PASW Statistics application bundle. If you have specified alternate locations for extension bundles with the *SPSS_EXTENSIONS_PATH* environment variable then copy the Python modules to one of those locations. For more information, see the topic Installation Locations for Extension Bundles on p. 286. Alternatively, you can copy the modules to a location on the Python search path, such as the Python *site-packages* directory.

Chapter

16

Options

Options control a wide variety of settings, including:

- Session journal, which keeps a record of all commands run in every session
- Display order for variables in dialog box source lists
- Items displayed and hidden in new output results
- TableLook for new pivot tables
- Custom currency formats

To Change Options Settings

▶ From the menus choose:

Edit
  Options...

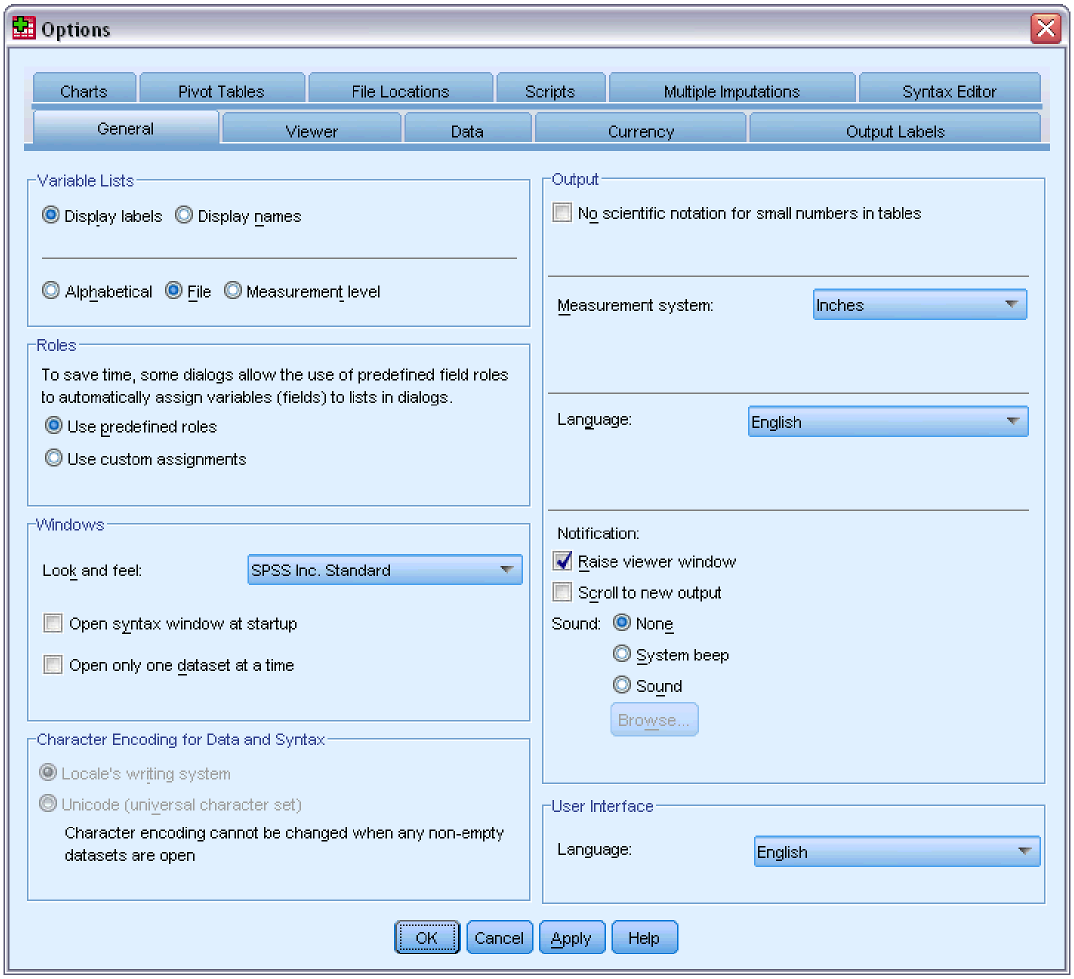
▶ Click the tabs for the settings that you want to change.

▶ Change the settings.

▶ Click OK or Apply.

General Options

Figure 16-1
Options dialog box: General tab



Variable Lists
● Display labels ○ Display names
○ Alphabetical ● File ○ Measurement level

Roles
To save time, some dialogs allow the use of predefined field roles to automatically assign variables (fields) to lists in dialogs.
● Use predefined roles
○ Use custom assignments

Windows
Look and feel: SPSS Inc. Standard
□ Open syntax window at startup
□ Open only one dataset at a time

Character Encoding for Data and Syntax
● Locale's writing system
○ Unicode (universal character set)
Character encoding cannot be changed when any non-empty datasets are open

Output
□ No scientific notation for small numbers in tables
Measurement system: Inches
Language: English
Notification:
☑ Raise viewer window
□ Scroll to new output
Sound: ● None
○ System beep
○ Sound

User Interface
Language: English

Variable Lists

These settings control the display of variables in dialog box lists. You can display variable names or variable labels. Names or labels can be displayed in alphabetical or file order or grouped by measurement level. Display order affects only source variable lists. Target variable lists always reflect the order in which variables were selected.

Roles

Some dialogs support the ability to pre-select variables for analysis based on defined roles. For more information, see the topic Roles in Chapter 5 on p. 76.

- **Use predefined roles.** By default, pre-select variables based on defined roles.
- **Use custom assignments.** By default, do *not* use roles to pre-select variables.

You can also switch between predefined roles and custom assignment within the dialogs that support this functionality. The setting here controls only the initial default behavior in effect for each dataset.

Windows

Look and feel. Controls the basic appearance of windows and dialog boxes. If you notice any display issues after changing the look and feel, try shutting down and restarting the application.

Open syntax window at startup. Syntax windows are text file windows used to enter, edit, and run commands. If you frequently work with command syntax, select this option to automatically open a syntax window at the beginning of each session. This is useful primarily for experienced users who prefer to work with command syntax instead of dialog boxes.

Open only one dataset at a time. Closes the currently open data source each time you open a different data source using the menus and dialog boxes. By default, every time you use the menus and dialog boxes to open a new data source, that data source opens in a new Data Editor window, and any other data sources open in other Data Editor windows remain open and available during the session until explicitly closed.

When you select this option, it takes effect immediately but does not close any datasets that were open at the time the setting was changed. This setting has no effect on data sources opened using command syntax, which relies on `DATASET` commands to control multiple datasets.For more information, see the topic Working with Multiple Data Sources in Chapter 6 on p. 92.

Character Encoding for Data Files and Syntax Files

This controls the default behavior for determining the encoding for reading and writing data files and syntax files. You can change this setting only when there are no open data sources, and the setting remains in effect for subsequent sessions until explicitly changed.

- **Locale's writing system.** Use the current locale setting to determine the encoding for reading and writing files. This is referred to as **code page mode**.
- **Unicode (universal character set).** Use Unicode encoding (UTF-8) for reading and writing files. This is referred to as **Unicode mode**.

There are a number of important implications regarding Unicode mode and Unicode files:

- PASW Statistics data files and syntax files saved in Unicode encoding should not be used in releases of PASW Statistics prior to 16.0. For syntax files, you can specify the encoding when you save the file. For data files, you should open the data file in code page mode and then resave it if you want to read the file with earlier versions.
- When code page data files are read in Unicode mode, the defined width of all string variables is tripled. To automatically set the width of each string variable to the longest observed value for that variable, select Minimize string widths based on observed values in the Open Data dialog box.

Output

No scientific notation for small numbers in tables. Suppresses the display of scientific notation for small decimal values in output. Very small decimal values will be displayed as 0 (or 0.000).

Measurement System. The measurement system used (points, inches, or centimeters) for specifying attributes such as pivot table cell margins, cell widths, and space between tables for printing.

Language. Controls the language used in the output. Does not apply to simple text output. The list of available languages depends on the currently installed language files. (Note: This does *not* affect the user interface language.) Depending on the language, you may also need to use Unicode character encoding for characters to render to properly.

Note: Custom scripts that rely on language-specific text strings in the output may not run correctly when you change the output language. For more information, see the topic Script Options on p. 306.

Notification. Controls the manner in which the program notifies you that it has finished running a procedure and that the results are available in the Viewer.

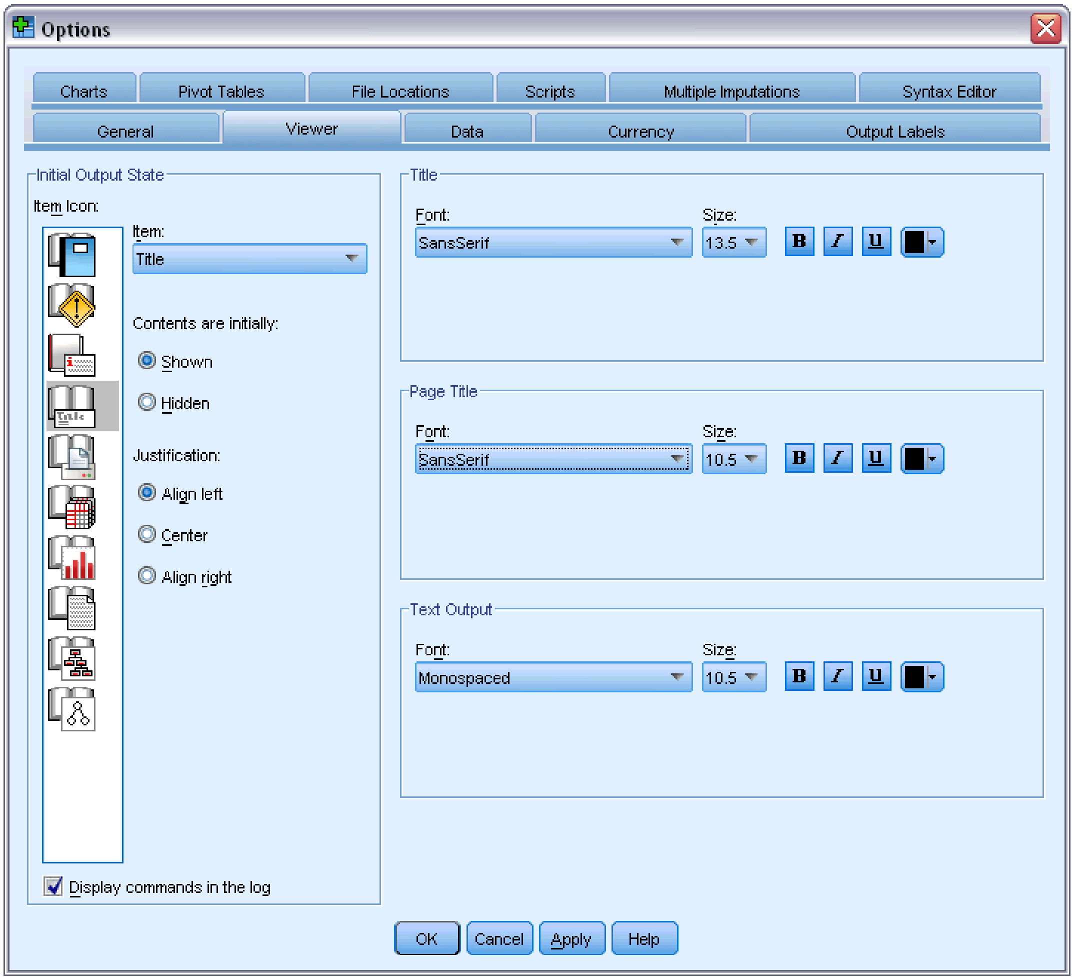
User Interface

This controls the language used in menus, dialogs, and other user interface features. (Note: This does *not* affect the output language.)

Viewer Options

Viewer output display options affect only new output produced after you change the settings. Output already displayed in the Viewer is not affected by changes in these settings.

Figure 16-2
Options dialog box: Viewer tab



Initial Output State

Item Icon:

Item: Title

Contents are initially:

● Shown

○ Hidden

Justification:

● Align left

○ Center

○ Align right

☑ Display commands in the log

Title

Font: SansSerif Size: 13.5

Page Title

Font: SansSerif Size: 10.5

Text Output

Font: Monospaced Size: 10.5

Initial Output State. Controls which items are automatically displayed or hidden each time you run a procedure and how items are initially aligned. You can control the display of the following items: log, warnings, notes, titles, pivot tables, charts, tree diagrams, and text output. You can also turn the display of commands in the log on or off. You can copy command syntax from the log and save it in a syntax file.

Note: All output items are displayed left-aligned in the Viewer. Only the alignment of printed output is affected by the justification settings. Centered and right-aligned items are identified by a small symbol above and to the left of the item.

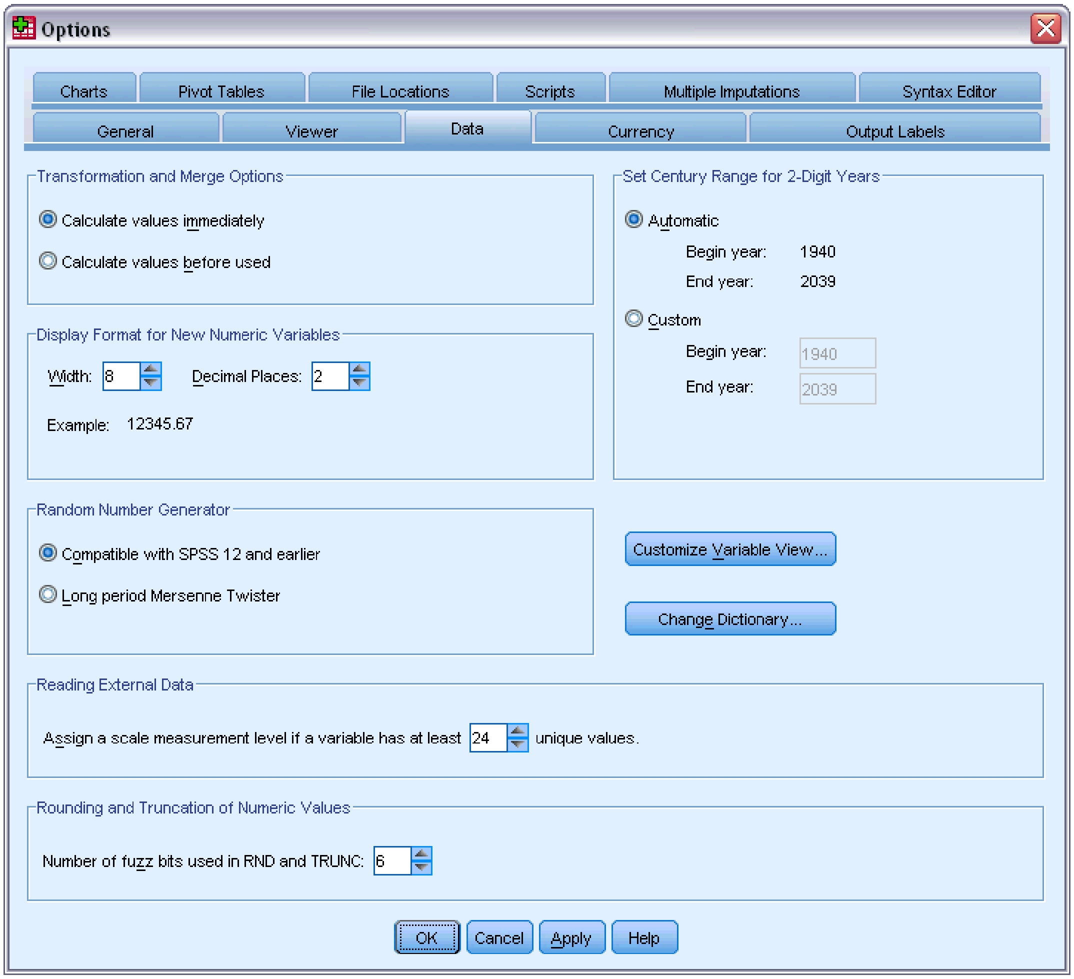
Title. Controls the font style, size, and color for new output titles.

Page Title. Controls the font style, size, and color for new page titles and page titles generated by `TITLE` and `SUBTITLE` command syntax or created by New Page Title on the Insert menu.

Text Output Font used for text output. Text output is designed for use with a monospaced (fixed-pitch) font. If you select a proportional font, tabular output will not align properly.

Data Options

Figure 16-3
Options dialog box: Data tab



Transformation and Merge Options

● Calculate values immediately

○ Calculate values before used

Display Format for New Numeric Variables

Width: 8 Decimal Places: 2

Example: 12345.67

Set Century Range for 2-Digit Years

● Automatic

Begin year: 1940

End year: 2039

○ Custom

Begin year: 1940

End year: 2039

Random Number Generator

● Compatible with SPSS 12 and earlier

○ Long period Mersenne Twister

Reading External Data

Assign a scale measurement level if a variable has at least 24 unique values.

Rounding and Truncation of Numeric Values

Number of fuzz bits used in RND and TRUNC: 6

Transformation and Merge Options. Each time the program executes a command, it reads the data file. Some data transformations (such as Compute and Recode) and file transformations (such as Add Variables and Add Cases) do not require a separate pass of the data, and execution of these commands can be delayed until the program reads the data to execute another command, such as a statistical or charting procedure.

- For large data files, where reading the data can take some time, you may want to select Calculate values before used to delay execution and save processing time. When this option is selected, the results of transformations you make using dialog boxes such as Compute Variable will not appear immediately in the Data Editor; new variables created by transformations will be displayed without any data values; and data values in the Data Editor cannot be changed while there are pending transformations. Any command that reads the data, such as a statistical or charting procedure, will execute the pending transformations and update the data displayed in the Data Editor. Alternatively, you can use Run Pending Transforms on the Transform menu.

- With the default setting of Calculate values immediately, when you paste command syntax from dialogs, an EXECUTE command is pasted after each transformation command. For more information, see the topic Multiple Execute Commands in Chapter 13 on p. 268.

Display Format for New Numeric Variables. Controls the default display width and number of decimal places for new numeric variables. There is no default display format for new string variables. If a value is too large for the specified display format, first decimal places are rounded and then values are converted to scientific notation. Display formats do not affect internal data values. For example, the value 123456.78 may be rounded to 123457 for display, but the original unrounded value is used in any calculations.

Set Century Range for 2-Digit Years. Defines the range of years for date-format variables entered and/or displayed with a two-digit year (for example, 10/28/86, 29-OCT-87). The automatic range setting is based on the current year, beginning 69 years prior to and ending 30 years after the current year (adding the current year makes a total range of 100 years). For a custom range, the ending year is automatically determined based on the value that you enter for the beginning year.

Random Number Generator. Two different random number generators are available:

- **Version 12 Compatible.** The random number generator used in version 12 and previous releases. If you need to reproduce randomized results generated in previous releases based on a specified seed value, use this random number generator.
- **Mersenne Twister.** A newer random number generator that is more reliable for simulation purposes. If reproducing randomized results from version 12 or earlier is not an issue, use this random number generator.

Reading External Data. For data read from external file formats and older PASW Statistics data files (prior to release 8.0), you can specify the minimum number of data values for a numeric variable used to classify the variable as scale or nominal. Variables with fewer than the specified number of unique values are classified as nominal.

Rounding and Truncation of Numeric Values. For the RND and TRUNC functions, this setting controls the default threshold for rounding up values that are very close to a rounding boundary. The setting is specified as a number of bits and is set to 6 at install time, which should be sufficient for most applications. Setting the number of bits to 0 produces the same results as in release 10. Setting the number of bits to 10 produces the same results as in releases 11 and 12.

- For the RND function, this setting specifies the number of least-significant bits by which the value to be rounded may fall short of the threshold for rounding up but still be rounded up. For example, when rounding a value between 1.0 and 2.0 to the nearest integer this setting specifies how much the value can fall short of 1.5 (the threshold for rounding up to 2.0) and still be rounded up to 2.0.
- For the TRUNC function, this setting specifies the number of least-significant bits by which the value to be truncated may fall short of the nearest rounding boundary and be rounded up before truncating. For example, when truncating a value between 1.0 and 2.0 to the nearest integer this setting specifies how much the value can fall short of 2.0 and be rounded up to 2.0.

Customize Variable View. Controls the default display and order of attributes in Variable View. For more information, see the topic Changing the Default Variable View on p. 297.

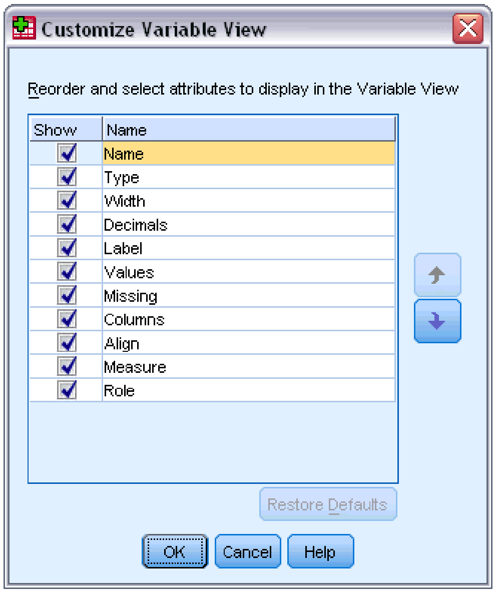
Change Dictionary. Controls the language version of the dictionary used for checking the spelling of items in Variable View. For more information, see the topic Spell Checking in Chapter 5 on p. 82.

Changing the Default Variable View

You can use Customize Variable View to control which attributes are displayed by default in Variable View (for example, name, type, label) and the order in which they are displayed.

Click Customize Variable View.

Figure 16-4
Customize Variable View (default)



Customize Variable View

Reorder and select attributes to display in the Variable View

Show	Name
☑	Name
☑	Type
☑	Width
☑	Decimals
☑	Label
☑	Values
☑	Missing
☑	Columns
☑	Align
☑	Measure
☑	Role

- Select (check) the variable attributes you want to display.

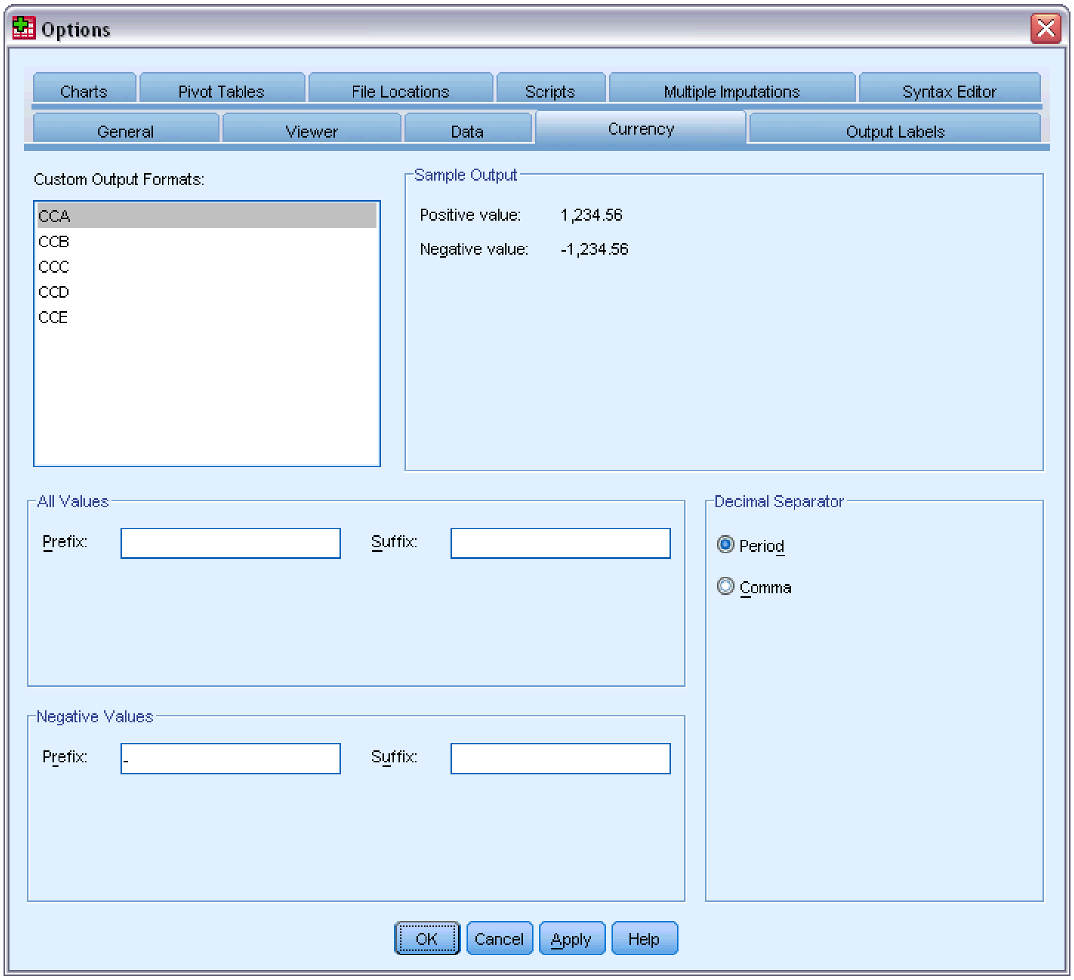
- Use the up and down arrow buttons to change the display order of the attributes.

Currency Options

You can create up to five custom currency display formats that can include special prefix and suffix characters and special treatment for negative values.

The five custom currency format names are CCA, CCB, CCC, CCD, and CCE. You cannot change the format names or add new ones. To modify a custom currency format, select the format name from the source list and make the changes that you want.

Figure 16-5
Options dialog box: Currency tab



Custom Output Formats:

- CCA
- CCB
- CCC
- CCD
- CCE

Sample Output

Positive value: 1,234.56

Negative value: -1,234.56

All Values

Prefix:

Suffix:

Negative Values

Prefix: -

Suffix:

Decimal Separator

● Period

○ Comma

Prefixes, suffixes, and decimal indicators defined for custom currency formats are for display purposes only. You cannot enter values in the Data Editor using custom currency characters.

To Create Custom Currency Formats

▶ Click the Currency tab.

▶ Select one of the currency formats from the list (CCA, CCB, CCC, CCD, and CCE).

▶ Enter the prefix, suffix, and decimal indicator values.

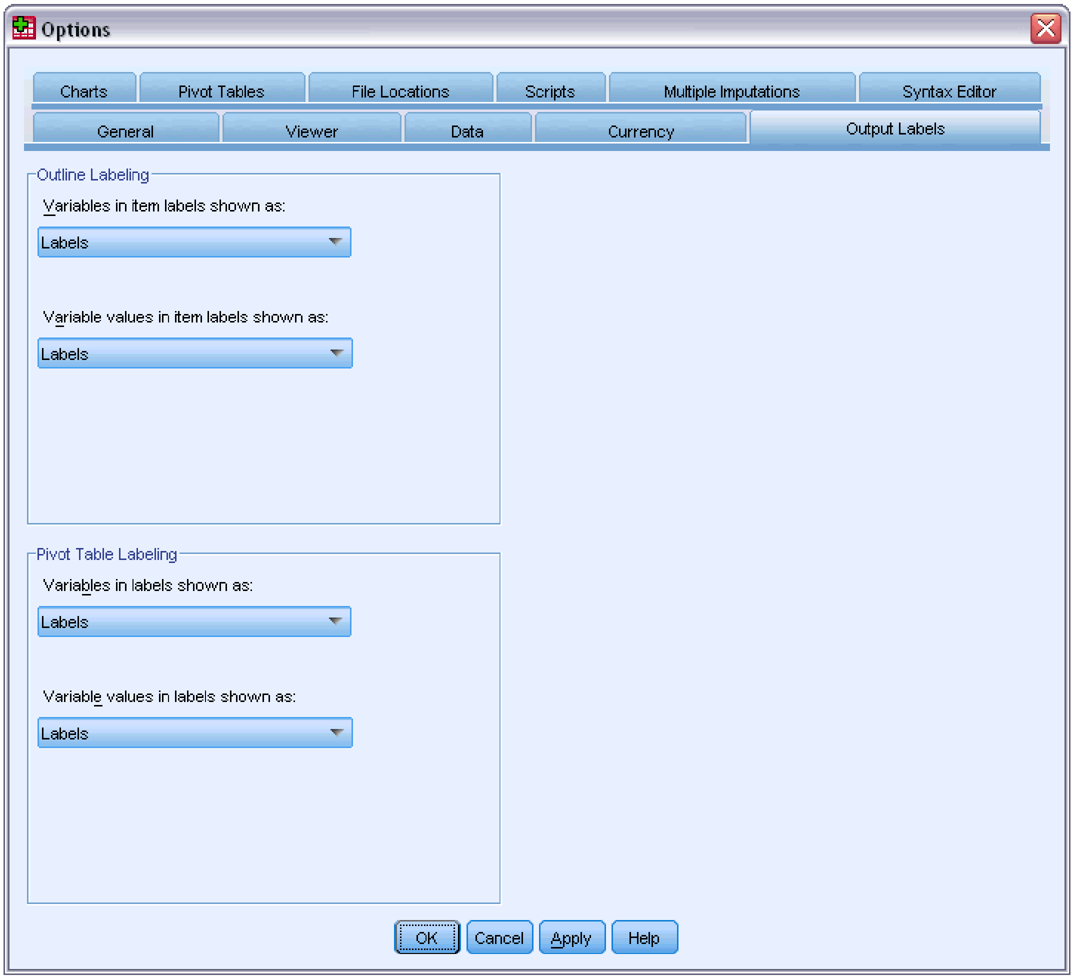
▶ Click OK or Apply.

Output Label Options

Output label options control the display of variable and data value information in the outline and pivot tables. You can display variable names, defined variable labels and actual data values, defined value labels, or a combination.

Descriptive variable and value labels (Variable view in the Data Editor, *Label* and *Values* columns) often make it easier to interpret your results. However, long labels can be awkward in some tables.

Figure 16-6
Options dialog box: Output Labels tab



Outline Labeling

Variables in item labels shown as: Labels

Variable values in item labels shown as: Labels

Pivot Table Labeling

Variables in labels shown as: Labels

Variable values in labels shown as: Labels

Output label options affect only new output produced after you change the settings. Output already displayed in the Viewer is not affected by changes in these settings. These settings affect only pivot table output. Text output is not affected by these settings.

Chart Options

Figure 16-7
Options dialog box: Charts tab

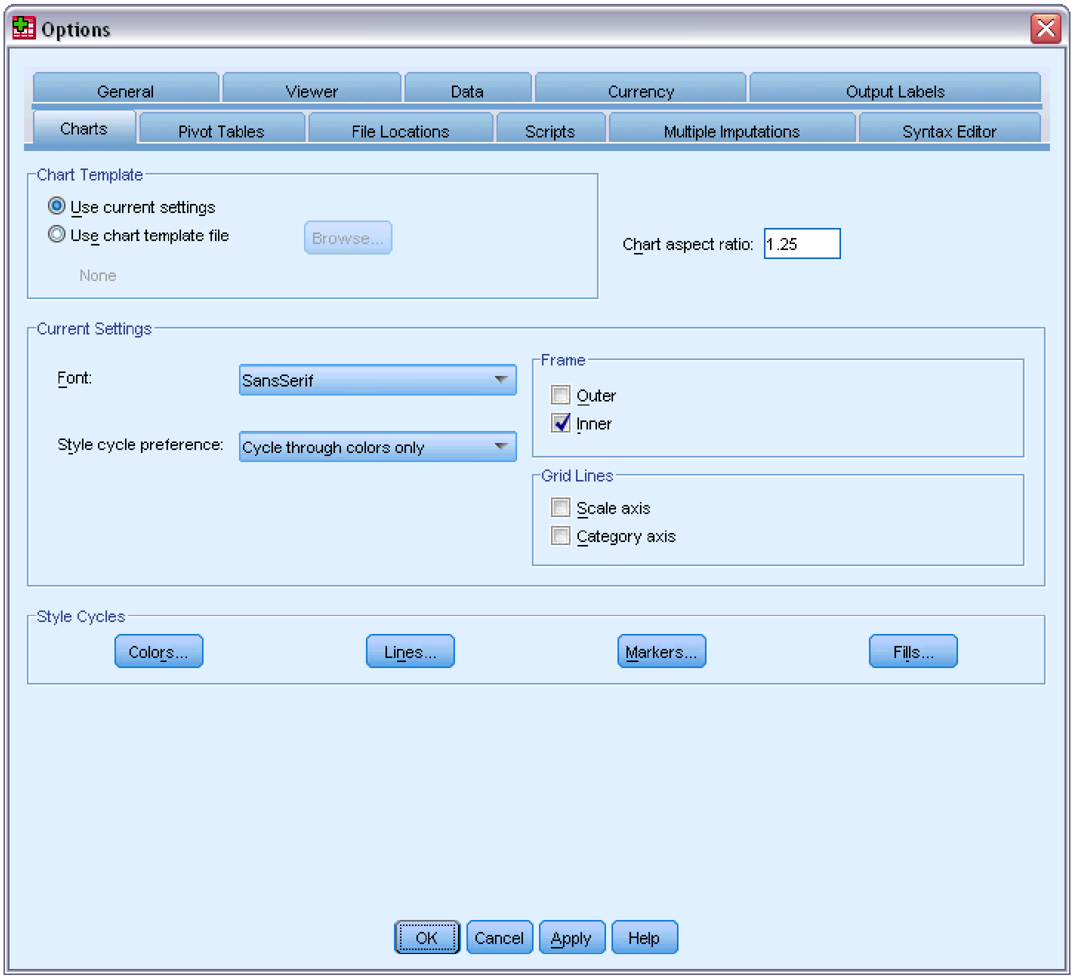


Chart Template

● Use current settings

○ Use chart template file

None

Chart aspect ratio: 1.25

Current Settings

Font: SansSerif

Style cycle preference: Cycle through colors only

Frame

□ Outer

☑ Inner

Grid Lines

□ Scale axis

□ Category axis

Chart Template. New charts can use either the settings selected here or the settings from a chart template file. Click Browse to select a chart template file. To create a chart template file, create a chart with the attributes that you want and save it as a template (choose Save Chart Template from the File menu).

Chart Aspect Ratio. The width-to-height ratio of the outer frame of new charts. You can specify a width-to-height ratio from 0.1 to 10.0. Values less than 1 make charts that are taller than they are wide. Values greater than 1 make charts that are wider than they are tall. A value of 1 produces a square chart. Once a chart is created, its aspect ratio cannot be changed.

Current Settings. Available settings include:

- **Font.** Font used for all text in new charts.

- **Style Cycle Preference.** The initial assignment of colors and patterns for new charts. Cycle through colors only uses only colors to differentiate chart elements and does not use patterns. Cycle through patterns only uses only line styles, marker symbols, or fill patterns to differentiate chart elements and does not use color.
- **Frame.** Controls the display of inner and outer frames on new charts.
- **Grid Lines.** Controls the display of scale and category axis grid lines on new charts.

Style Cycles. Customizes the colors, line styles, marker symbols, and fill patterns for new charts. You can change the order of the colors and patterns that are used when a new chart is created.

Data Element Colors

Specify the order in which colors should be used for the data elements (such as bars and markers) in your new chart. Colors are used whenever you select a choice that includes *color* in the Style Cycle Preference group in the main Chart Options dialog box.

For example, if you create a clustered bar chart with two groups and you select Cycle through colors, then patterns in the main Chart Options dialog box, the first two colors in the Grouped Charts list are used as the bar colors on the new chart.

To Change the Order in Which Colors Are Used

▶ Select Simple Charts and then select a color that is used for charts without categories.

▶ Select Grouped Charts to change the color cycle for charts with categories. To change a category's color, select a category and then select a color for that category from the palette.

Optionally, you can:

- Insert a new category above the selected category.
- Move a selected category.
- Remove a selected category.
- Reset the sequence to the default sequence.
- Edit a color by selecting its well and then clicking Edit.

Data Element Lines

Specify the order in which styles should be used for the line data elements in your new chart. Line styles are used whenever your chart includes line data elements and you select a choice that includes *patterns* in the Style Cycle Preference group in the main Chart Options dialog box.

For example, if you create a line chart with two groups and you select Cycle through patterns only in the main Chart Options dialog box, the first two styles in the Grouped Charts list are used as the line patterns on the new chart.

To Change the Order in Which Line Styles Are Used

▶ Select Simple Charts and then select a line style that is used for line charts without categories.

▶ Select Grouped Charts to change the pattern cycle for line charts with categories. To change a category's line style, select a category and then select a line style for that category from the palette.

Optionally, you can:

- Insert a new category above the selected category.
- Move a selected category.
- Remove a selected category.
- Reset the sequence to the default sequence.

Data Element Markers

Specify the order in which symbols should be used for the marker data elements in your new chart. Marker styles are used whenever your chart includes marker data elements and you select a choice that includes *patterns* in the Style Cycle Preference group in the main Chart Options dialog box.

For example, if you create a scatterplot chart with two groups and you select Cycle through patterns only in the main Chart Options dialog box, the first two symbols in the Grouped Charts list are used as the markers on the new chart.

To Change the Order in Which Marker Styles Are Used

▶ Select Simple Charts and then select a marker symbol that is used for charts without categories.

▶ Select Grouped Charts to change the pattern cycle for charts with categories. To change a category's marker symbol, select a category and then select a symbol for that category from the palette.

Optionally, you can:

- Insert a new category above the selected category.
- Move a selected category.
- Remove a selected category.
- Reset the sequence to the default sequence.

Data Element Fills

Specify the order in which fill styles should be used for the bar and area data elements in your new chart. Fill styles are used whenever your chart includes bar or area data elements, and you select a choice that includes *patterns* in the Style Cycle Preference group in the main Chart Options dialog box.

For example, if you create a clustered bar chart with two groups and you select Cycle through patterns only in the main Chart Options dialog box, the first two styles in the Grouped Charts list are used as the bar fill patterns on the new chart.

To Change the Order in Which Fill Styles Are Used

▶ Select Simple Charts and then select a fill pattern that is used for charts without categories.

▶ Select Grouped Charts to change the pattern cycle for charts with categories. To change a category's fill pattern, select a category and then select a fill pattern for that category from the palette.

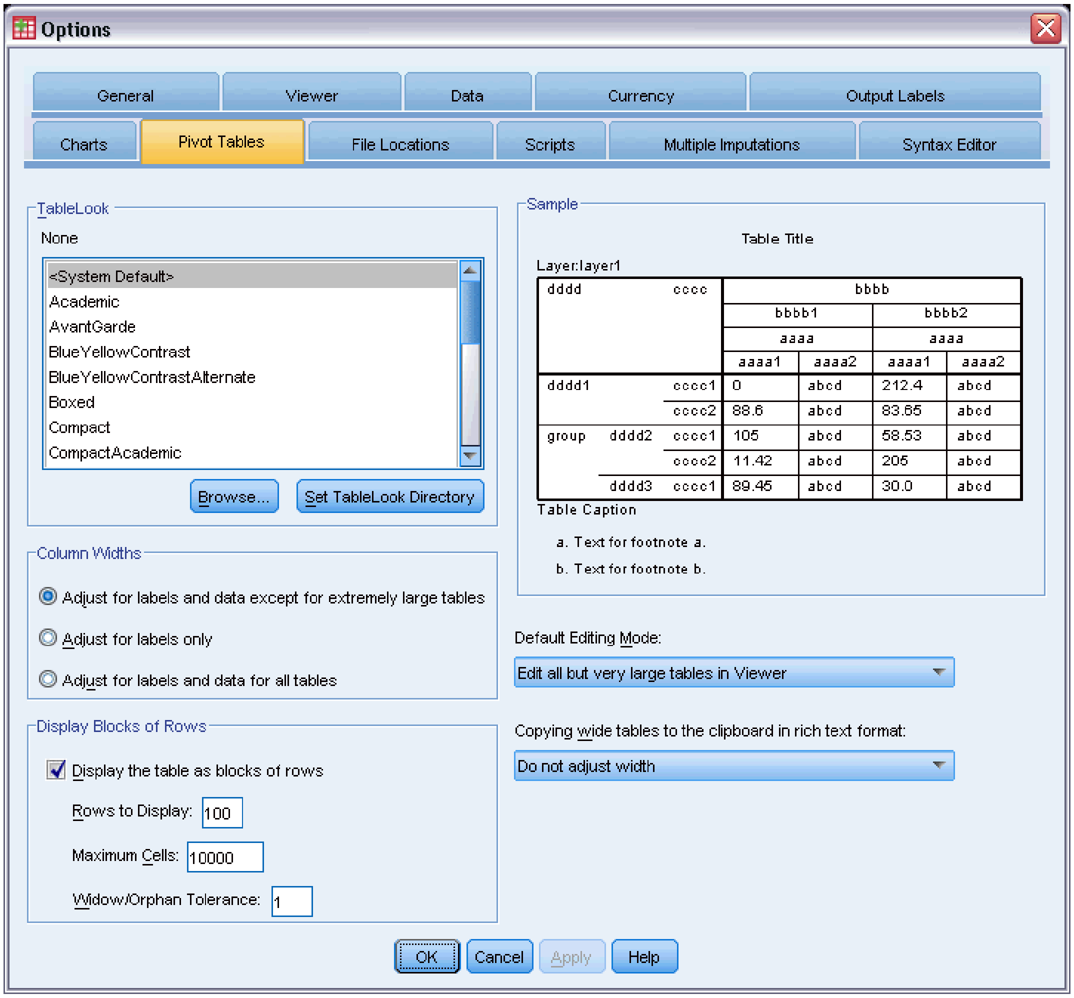
Optionally, you can:

- Insert a new category above the selected category.
- Move a selected category.
- Remove a selected category.
- Reset the sequence to the default sequence.

Pivot Table Options

Pivot Table options set the default TableLook used for new pivot table output. TableLooks can control a variety of pivot table attributes, including the display and width of grid lines; font style, size, and color; and background colors.

Figure 16-8
Options dialog box: Pivot Tables tab



TableLook

None

- <System Default>
- Academic
- AvantGarde
- BlueYellowContrast
- BlueYellowContrastAlternate
- Boxed
- Compact
- CompactAcademic

Column Widths

● Adjust for labels and data except for extremely large tables

○ Adjust for labels only

○ Adjust for labels and data for all tables

Display Blocks of Rows

☑ Display the table as blocks of rows

Rows to Display: 100

Maximum Cells: 10000

Widow/Orphan Tolerance: 1

Sample

Table Title

Layer:layer1

dddd		cccc	bbbb			
			bbbb1		bbbb2	
			aaaa		aaaa	
			aaaa1	aaaa2	aaaa1	aaaa2
dddd1		cccc1	0	abcd	212.4	abcd
		cccc2	88.6	abcd	83.65	abcd
group	dddd2	cccc1	105	abcd	58.53	abcd
		cccc2	11.42	abcd	205	abcd
	dddd3	cccc1	89.45	abcd	30.0	abcd

Table Caption

a. Text for footnote a.

b. Text for footnote b.

Default Editing Mode: Edit all but very large tables in Viewer

Copying wide tables to the clipboard in rich text format: Do not adjust width

TableLook. Select a TableLook from the list of files and click OK or Apply. You can use one of the TableLooks provided with PASW Statistics, or you can create your own in the Pivot Table Editor (choose TableLooks from the Format menu).

- **Browse.** Allows you to select a TableLook from another directory.
- **Set TableLook Directory.** Allows you to change the default TableLook directory. Use Browse to navigate to the directory you want to use, select a TableLook in that directory, and then select Set TableLook Directory.

Note: TableLooks created in earlier versions of PASW Statistics cannot be used in version 16.0 or later.

Column Widths. Controls the automatic adjustment of column widths in pivot tables.

- **Adjust for labels and data except for extremely large tables.** For tables that don't exceed 10,000 cells, adjusts column width to whichever is larger—the column label or the largest data value. For tables with more than 10,000 cells, adjusts column width to the width of the column label.
- **Adjust for labels only.** Adjusts column width to the width of the column label. This produces more compact tables, but data values wider than the label may be truncated.
- **Adjust for labels and data.** Adjusts column width to whichever is larger: the column label or the largest data value. This produces wider tables, but it ensures that all values will be displayed.

Display Blocks of Rows. These settings control the display of large pivot tables in the Viewer. These settings have no effect on printing large pivot tables or exporting output to external formats.

- **Display the table as blocks of rows.** Tables are displayed in the Viewer in sections, and navigation controls allow you to view different sections of the table.
- **Rows to display.** Sets the number of rows to display in each section. The value must be an integer between 10 and 1000.
- **Maximum cells.** Sets the maximum number of cells to display in each section. The value must an integer between 1000 and 100000.
- **Widow/orphan tolerance.** Controls the maximum number of rows of the inner most row dimension of the table to split across displayed views of the table. For example, if there are six categories in each group of the inner most row dimension, specifying a value of six would prevent any group from splitting across displayed views. The value must be an integer and cannot be greater than the Rows to display value.

Several factors can affect the actual number of rows displayed, including:

- If the Maximum cells value is reached before the Rows to display value, then the table is split at that point. For example, if Rows to display is 1000 and Maximum cells is 10000, a table with 20 columns will be displayed in blocks of 500 rows.
- The Widow/orphan tolerance setting can cause fewer or more rows to be displayed than the value specified for Rows to display.

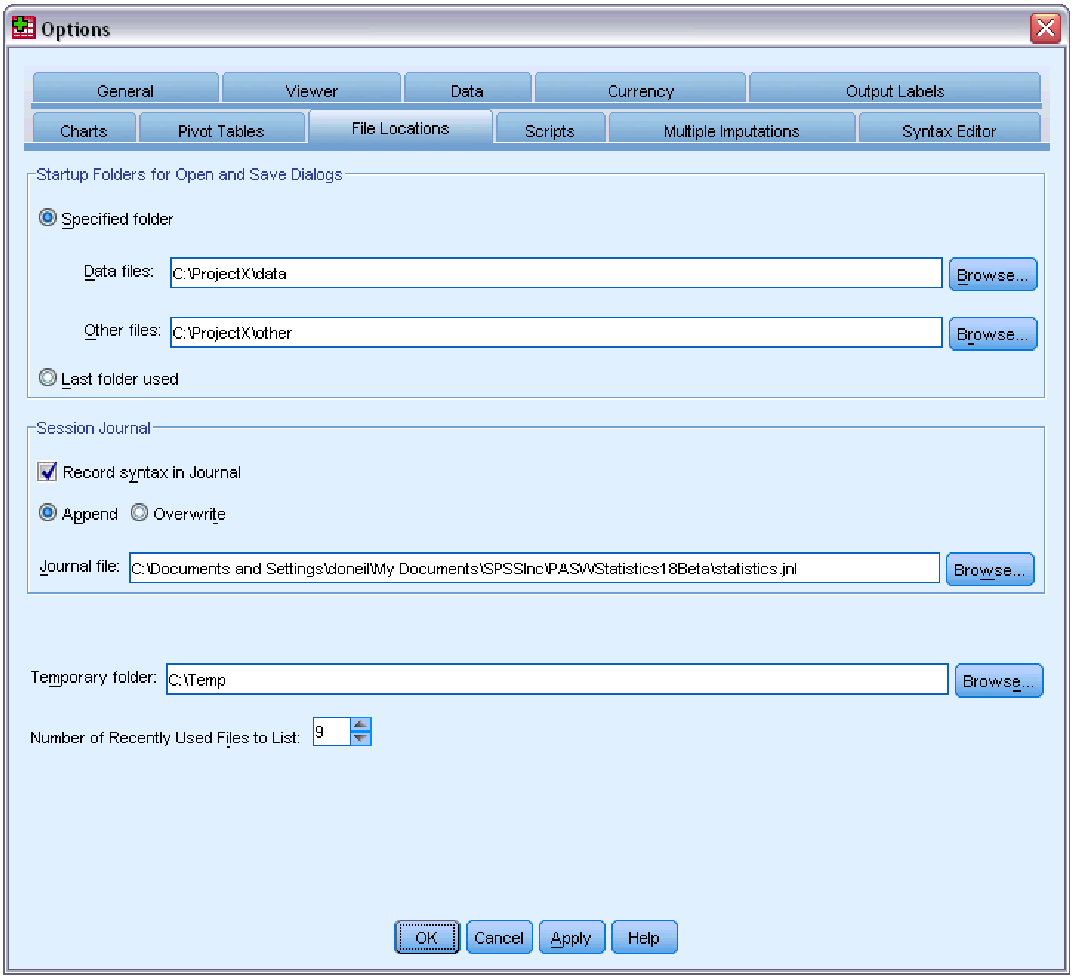
Default Editing Mode. Controls activation of pivot tables in the Viewer window or in a separate window. By default, double-clicking a pivot table activates all but very large tables in the Viewer window. You can choose to activate pivot tables in a separate window or select a size setting that will open smaller pivot tables in the Viewer window and larger pivot tables in a separate window.

Copying wide tables to the clipboard in rich text format. When pivot tables are pasted in Word/RTF format, tables that are too wide for the document width will either be wrapped, scaled down to fit the document width, or left unchanged.

File Locations Options

Options on the File Locations tab control the default location that the application will use for opening and saving files at the start of each session, the location of the journal file, the location of the temporary folder, and the number of files that appear in the recently used file list.

Figure 16-9
Options dialog box: FIle Locations tab



Startup Folders for Open and Save Dialogs

● Specified folder

Data files: C:\ProjectX\data

Other files: C:\ProjectX\other

○ Last folder used

Session Journal

☑ Record syntax in Journal

● Append ○ Overwrite

Journal file: C:\Documents and Settings\doneil\My Documents\SPSSInc\PASWStatistics18Beta\statistics.jnl

Temporary folder: C:\Temp

Number of Recently Used Files to List: 9

Startup Folders for Open and Save Dialog Boxes

- **Specified folder.** The specified folder is used as the default location at the start of each session. You can specify different default locations for data and other (non-data) files.
- **Last folder used.** The last folder used to either open or save files in the previous session is used as the default at the start of the next session. This applies to both data and other (non-data) files.

These settings apply only to dialog boxes for opening and saving files—and the "last folder used" is determined by the last dialog box used to open or save a file. Files opened or saved via command syntax have no effect on—and are not affected by—these settings. These settings are only available in local analysis mode. In distributed analysis mode connected to a remote server (requires PASW Statistics Server), you cannot control the startup folder locations.

Session Journal

You can use the session journal to automatically record commands run in a session. This includes commands entered and run in syntax windows and commands generated by dialog box choices. You can edit the journal file and use the commands again in other sessions. You can turn journaling off and on, append or overwrite the journal file, and select the journal filename and location. You can copy command syntax from the journal file and save it in a syntax file.

Temporary Folder

This specifies the location of temporary files created during a session. In distributed mode (available with the server version), this does not affect the location of temporary data files. In distributed mode, the location of temporary data files is controlled by the environment variable *SPSSTMPDIR*, which can be set only on the computer running the server version of the software. If you need to change the location of the temporary directory, contact your system administrator.

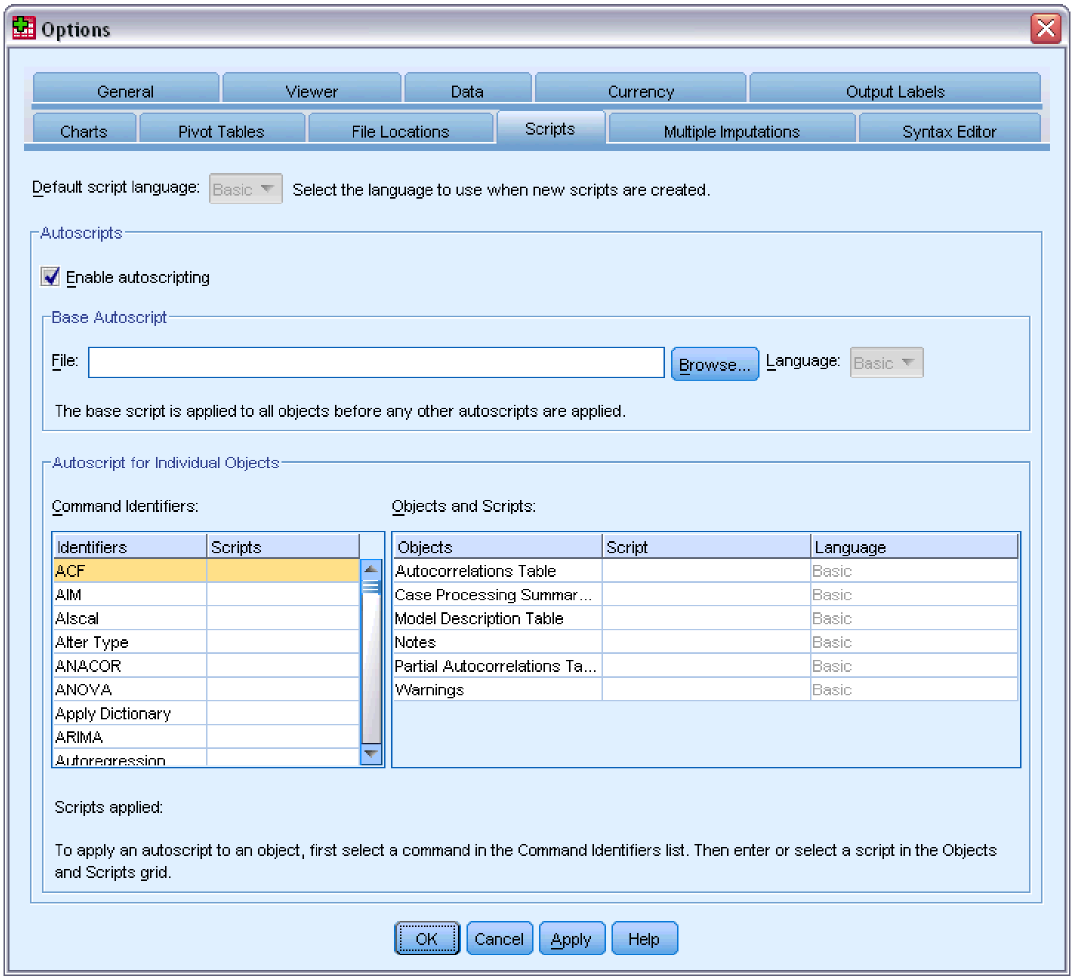
Recently Used File List

This controls the number of recently used files that appear on the File menu.

Script Options

Use the Scripts tab to specify the default script language and any autoscripts you want to use. You can use scripts to automate many functions, including customizing pivot tables.

Figure 16-10
Options dialog box: Scripts tab



Default script language: Basic Select the language to use when new scripts are created.

Autoscripts

☑ Enable autoscripting

Base Autoscript

File: Language: Basic

The base script is applied to all objects before any other autoscripts are applied.

Autoscript for Individual Objects

Command Identifiers:

Identifiers	Scripts
ACF	
AIM	
Alscal	
Alter Type	
ANACOR	
ANOVA	
Apply Dictionary	
ARIMA	
Autoregression	

Objects and Scripts:

Objects	Script	Language
Autocorrelations Table		Basic
Case Processing Summar...		Basic
Model Description Table		Basic
Notes		Basic
Partial Autocorrelations Ta...		Basic
Warnings		Basic

Scripts applied:

To apply an autoscript to an object, first select a command in the Command Identifiers list. Then enter or select a script in the Objects and Scripts grid.

Note: Legacy Sax Basic users must manually convert any custom autoscripts. The autoscripts installed with pre-16.0 versions are available as a set of separate script files located in the *Samples* subdirectory of the directory where PASW Statistics is installed. By default, no output items are associated with autoscripts. You must manually associate all autoscripts with the desired output items, as described below. For information on converting legacy autoscripts, see Compatibility with Versions Prior to 16.0 on p. 390.

Default script language. The default script language determines the script editor that is launched when new scripts are created. It also specifies the default language whose executable will be used to run autoscripts. The available scripting languages depend on your platform. For Windows, the available scripting languages are Basic, which is installed with the Core system, and the Python programming language. For all other platforms, scripting is available with the Python programming language.

To enable scripting with the Python programming language, you must have Python and the PASW Statistics-Python Integration Plug-in installed. For information, see How to Get Integration Plug-ins, available from Core System>Frequently Asked Questions in the Help system.

Enable Autoscripting. This check box allows you to enable or disable autoscripting. By default, autoscripting is enabled.

Base Autoscript. An optional script that is applied to all new Viewer objects before any other autoscripts are applied. Specify the script file to be used as the base autoscript as well as the language whose executable will be used to run the script.

To Apply Autoscripts to Output Items

▶ In the Command Identifiers grid, select a command that generates output items to which autoscripts will be applied.

The *Objects* column, in the Objects and Scripts grid, displays a list of the objects associated with the selected command. The *Script* column displays any existing script for the selected command.

▶ Specify a script for any of the items shown in the *Objects* column. Click on the corresponding *Script* cell. Enter the path to the script or click the ellipsis (...) button to browse for the script.

▶ Specify the language whose executable will be used to run the script. *Note*: The selected language is not affected by changing the default script language.

▶ Click Apply or OK.

To Remove Autoscript Associations

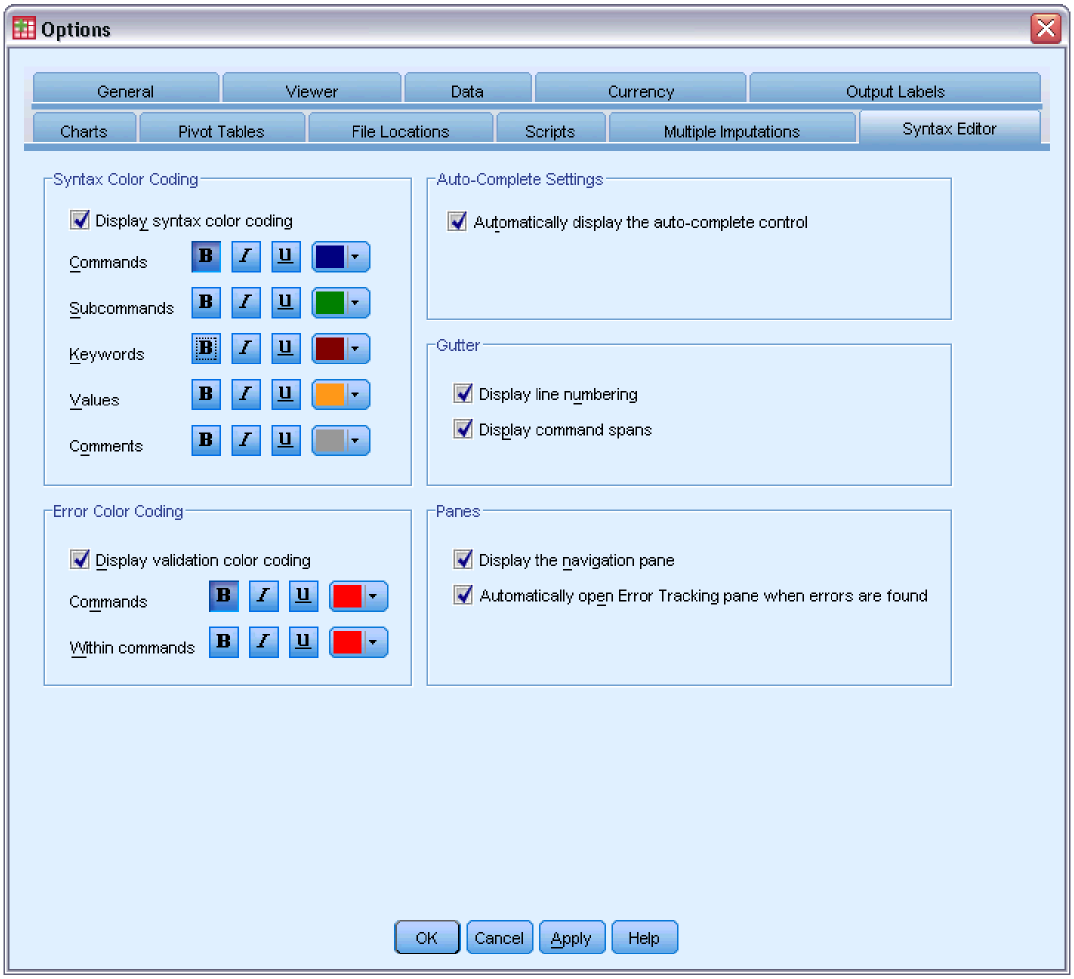
▶ In the Objects and Scripts grid, click on the cell in the Script column corresponding to the script to dissociate.

▶ Delete the path to the script and then click on any other cell in the Objects and Scripts grid.

▶ Click Apply or OK.

Syntax Editor Options

Figure 16-11
Options dialog box: Syntax Editor tab



Syntax Color Coding

☑ Display syntax color coding

Commands B I U 🟦

Subcommands B I U 🟩

Keywords B I U 🟥

Values B I U 🟧

Comments B I U ⬜

Auto-Complete Settings

☑ Automatically display the auto-complete control

Gutter

☑ Display line numbering

☑ Display command spans

Error Color Coding

☑ Display validation color coding

Commands B I U 🟥

Within commands B I U 🟥

Panes

☑ Display the navigation pane

☑ Automatically open Error Tracking pane when errors are found

Syntax Color Coding

You can turn color coding of commands, subcommands, keywords, keyword values, and comments off or on and you can specify the font style and color for each.

Error Color Coding

You can turn color coding of certain syntactical errors off or on and you can specify the font style and color used. Both the command name and the text (within the command) containing the error are color coded, and you can choose different styles for each. For more information, see the topic Color Coding in Chapter 13 on p. 263.

Auto-Complete Settings

You can turn automatic display of the auto-complete control off or on. The auto-complete control can always be displayed by pressing Ctrl+Spacebar. For more information, see the topic Auto-Completion in Chapter 13 on p. 263.

Gutter

These options specify the default behavior for showing or hiding line numbers and command spans within the Syntax Editor's gutter—the region to the left of the text pane that is reserved for line numbers, bookmarks, breakpoints, and command spans. Command spans are icons that provide visual indicators of the start and end of a command.

Panes

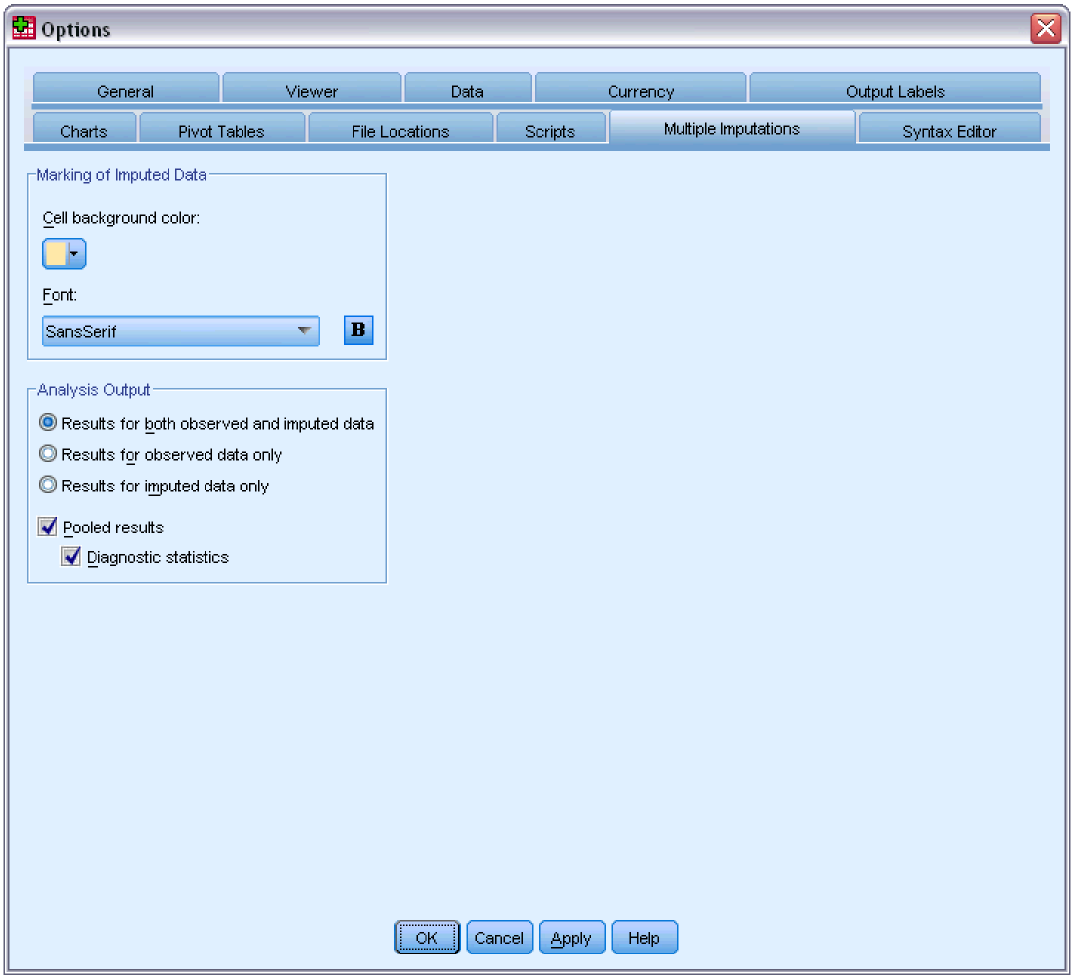
Display the navigation pane. This specifies the default for showing or hiding the navigation pane. The navigation pane contains a list of all recognized commands in the syntax window, displayed in the order in which they occur. Clicking on a command in the navigation pane positions the cursor at the start of the command.

Automatically open Error Tracking pane when errors are found. This specifies the default for showing or hiding the error tracking pane when run-time errors are found.

Optimize for right to left languages. Check this box for the best user experience when working with right to left languages.

Multiple Imputations Options

Figure 16-12
Options dialog box: Multiple Imputations tab



Marking of Imputed Data

Cell background color:

Font: SansSerif

Analysis Output

● Results for both observed and imputed data
○ Results for observed data only
○ Results for imputed data only

☑ Pooled results
☑ Diagnostic statistics

The Multiple Imputations tab controls two kinds of preferences related to Multiple Imputations:

Appearance of Imputed Data. By default, cells containing imputed data will have a different background color than cells containing nonimputed data. The distinctive appearance of the imputed data should make it easy for you to scroll through a dataset and locate those cells. You can change the default cell background color, the font, and make the imputed data display in bold type.

Analysis Output. This group controls the type of Viewer output produced whenever a multiply imputed dataset is analyzed. By default, output will be produced for the original (pre-imputation) dataset and for each of the imputed datasets. In addition, for those procedures that support pooling of imputed data, final pooled results will be generated. When univariate pooling is performed, pooling diagnostics will also display. However, you can suppress any output you do not want to see.

To Set Multiple Imputation Options

From the menus, choose:

Edit
  Options

Click the Multiple Imputation tab.

Chapter

17

Customizing Menus and Toolbars

Menu Editor

You can use the Menu Editor to customize your menus. With the Menu Editor you can:

- Add menu items that run customized scripts.
- Add menu items that run command syntax files.
- Add menu items that launch other applications and automatically send data to other applications.

You can send data to other applications in the following formats: PASW Statistics, Excel, Lotus 1-2-3, tab-delimited, and dBASE IV.

To Add Items to Menus

▶ From the menus choose:

View
  Menu Editor...

▶ In the Menu Editor dialog box, double-click the menu (or click the plus sign icon) to which you want to add a new item.

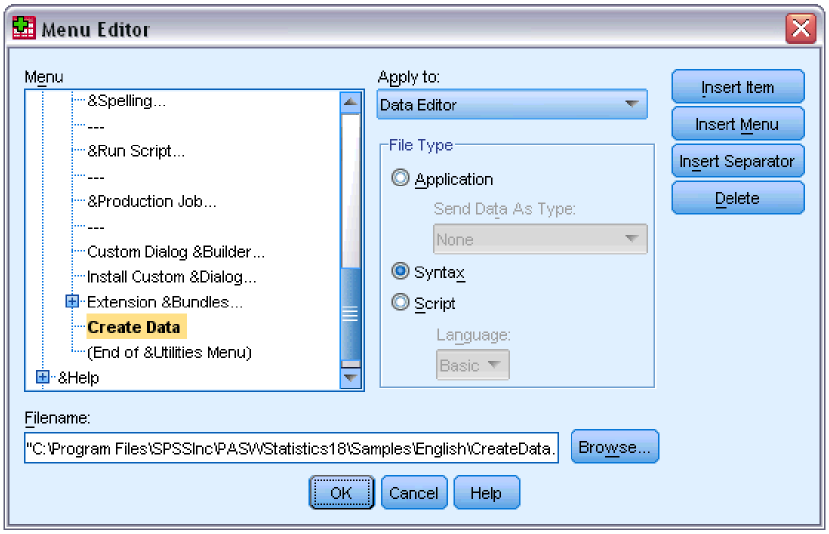
▶ Select the menu item above which you want the new item to appear.

▶ Click Insert Item to insert a new menu item.

▶ Select the file type for the new item (script file, command syntax file, or external application).

▶ Click Browse to select a file to attach to the menu item.

Figure 17-1
Menu Editor dialog box



Apply to: Data Editor

● Syntax

Filename: "C:\Program Files\SPSSInc\PASWStatistics18\Samples\English\CreateData.

You can also add entirely new menus and separators between menu items. Optionally, you can automatically send the contents of the Data Editor to another application when you select that application on the menu.

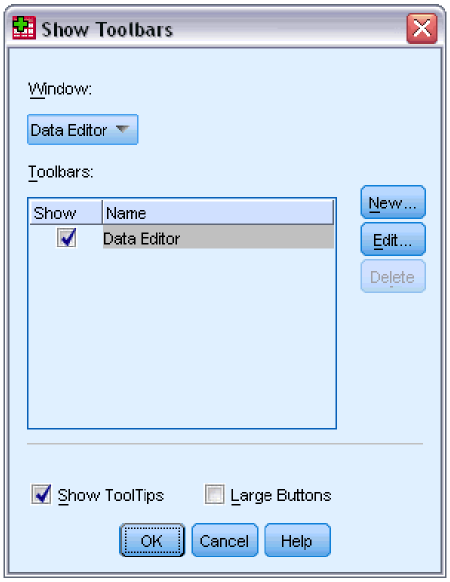
Customizing Toolbars

You can customize toolbars and create new toolbars. Toolbars can contain any of the available tools, including tools for all menu actions. They can also contain custom tools that launch other applications, run command syntax files, or run script files.

Show Toolbars

Use Show Toolbars to show or hide toolbars, customize toolbars, and create new toolbars. Toolbars can contain any of the available tools, including tools for all menu actions. They can also contain custom tools that launch other applications, run command syntax files, or run script files.

Figure 17-2
Show Toolbars dialog box



Window: Data Editor

Toolbars:

Show	Name
☑	Data Editor

☑ Show ToolTips □ Large Buttons

To Customize Toolbars

▶ From the menus choose:

View
  Toolbars
    Customize

▶ Select the toolbar you want to customize and click Edit, or click New to create a new toolbar.

▶ For new toolbars, enter a name for the toolbar, select the windows in which you want the toolbar to appear, and click Edit.

▶ Select an item in the Categories list to display available tools in that category.

▶ Drag and drop the tools you want onto the toolbar displayed in the dialog box.

▶ To remove a tool from the toolbar, drag it anywhere off the toolbar displayed in the dialog box.

To create a custom tool to open a file, to run a command syntax file, or to run a script:

▶ Click New Tool in the Edit Toolbar dialog box.

▶ Enter a descriptive label for the tool.

▶ Select the action you want for the tool (open a file, run a command syntax file, or run a script).

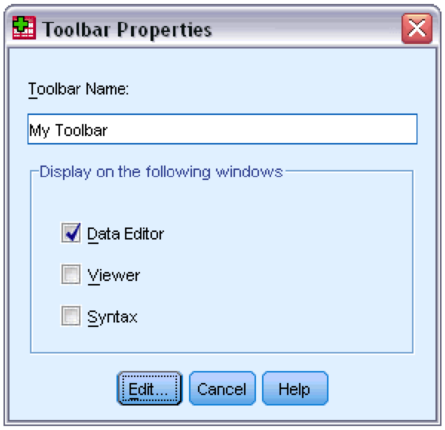
▶ Click Browse to select a file or application to associate with the tool.

New tools are displayed in the User-Defined category, which also contains user-defined menu items.

Toolbar Properties

Use Toolbar Properties to select the window types in which you want the selected toolbar to appear. This dialog box is also used for creating names for new toolbars.

Figure 17-3
Toolbar Properties dialog box



Toolbar Name:

My Toolbar

Display on the following windows

☑ Data Editor

□ Viewer

□ Syntax

To Set Toolbar Properties

- From the menus choose:

View
  Toolbars
    Customize

- For existing toolbars, click Edit, and then click Properties in the Edit Toolbar dialog box.

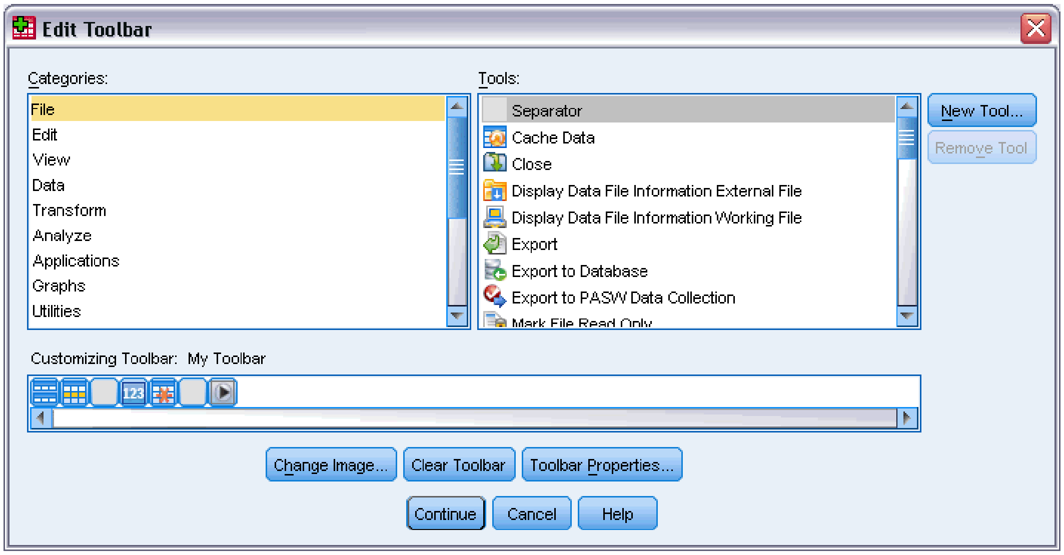
- For new toolbars, click New Tool.

- Select the window types in which you want the toolbar to appear. For new toolbars, also enter a toolbar name.

Edit Toolbar

Use the Edit Toolbar dialog box to customize existing toolbars and create new toolbars. Toolbars can contain any of the available tools, including tools for all menu actions. They can also contain custom tools that launch other applications, run command syntax files, or run script files.

Figure 17-4
Customize Toolbar dialog box



To Change Toolbar Images

▶ Select the tool for which you want to change the image on the toolbar display.

▶ Click Change Image.

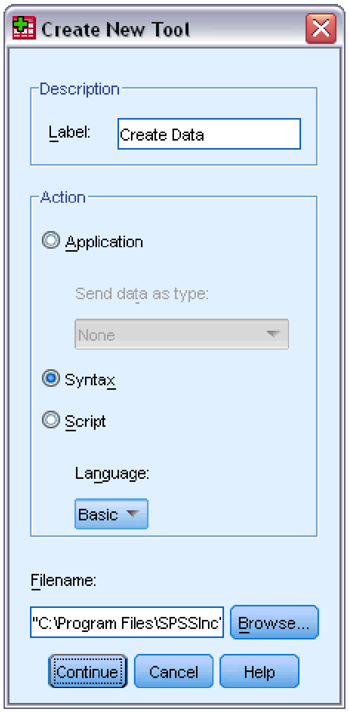
▶ Select the image file that you want to use for the tool. The following image formats are supported: BMP, PNG, GIF, JPG.

- Images should be square. Images that are not square are clipped to a square area.
- Images are automatically scaled to fit. For optimal display, use 16x16 pixel images for small toolbar images or 32x32 pixel images for large toolbar images.

Create New Tool

Use the Create New Tool dialog box to create custom tools to launch other applications, run command syntax files, and run script files.

Figure 17-5
Create New Tool dialog box



Create New Tool

Description

Label: Create Data

Action

○ Application

Send data as type: None

● Syntax

○ Script

Language: Basic

Filename: "C:\Program Files\SPSSInc'

Chapter

18

Creating and Managing Custom Dialogs

The Custom Dialog Builder allows you to create and manage custom dialogs for generating command syntax. Using the Custom Dialog Builder you can:

- Create your own version of a dialog for a built-in PASW Statistics procedure. For example, you can create a dialog for the Frequencies procedure that only allows the user to select the set of variables and then generates command syntax with pre-set options that standardize the output.
- Create a user interface that generates command syntax for an extension command. Extension commands are user-defined PASW Statistics commands that are implemented in either the Python programming language or R. For more information, see the topic Custom Dialogs for Extension Commands on p. 341.
- Open a file containing the specification for a custom dialog—perhaps created by another user—and add the dialog to your installation of PASW Statistics, optionally making your own modifications.
- Save the specification for a custom dialog so that other users can add it to their installations of PASW Statistics.

How to Start the Custom Dialog Builder

▶ From the menus choose:

Utilities
  Custom Dialogs
    Custom Dialog Builder...

What's New in Version 18?

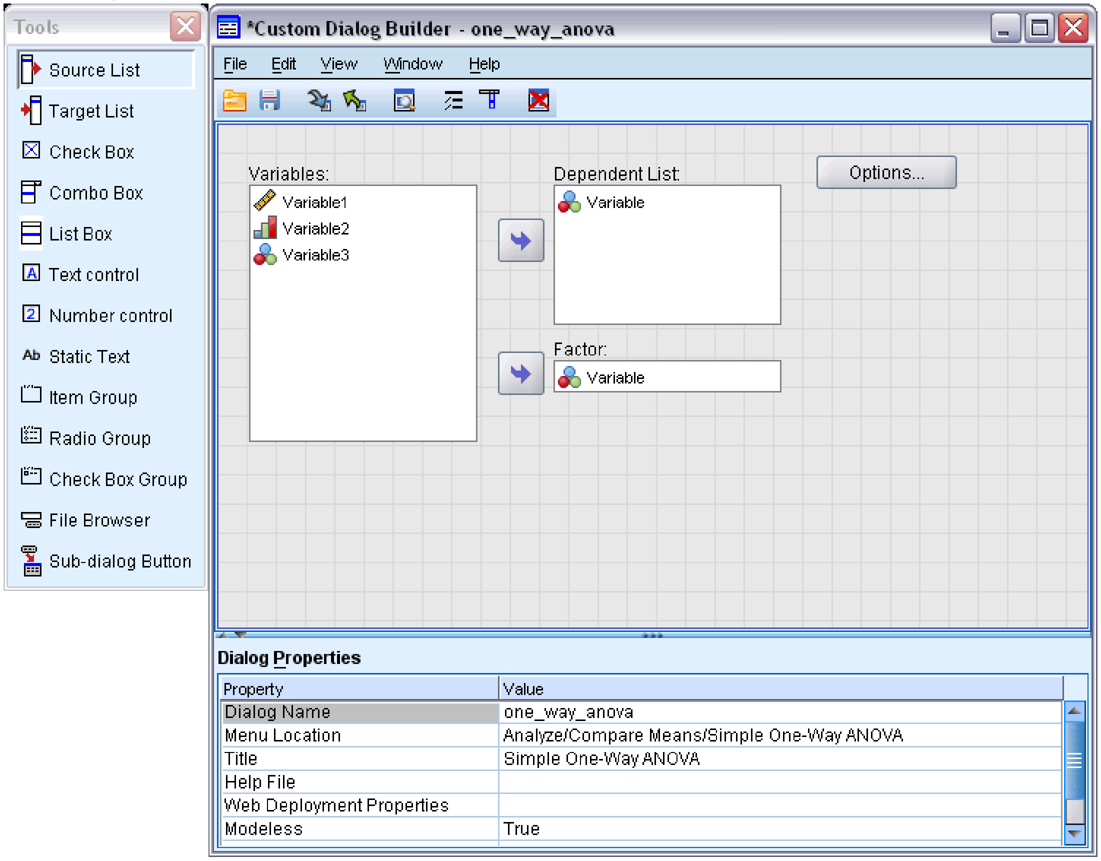
List box control. The Custom Dialog Builder now has a list box control that supports single or multiple selection. For more information, see the topic Combo Box and List Box Controls on p. 333.

Dynamically populate list items. List items for combo box controls and the new list box control can be dynamically populated with values associated with the variables in a specified target list. For more information, see the topic Specifying List Items for Combo Boxes and List Boxes on p. 334.

Nest controls under radio buttons. Radio buttons can now contain a set of nested controls. For more information, see the topic Defining Radio Buttons on p. 337.

Required Add-Ons property. The new Required Add-Ons dialog property allows you to specify one or more add-ons—such as the Python or R Integration Plug-in—that are required in order to run the command syntax generated by the dialog. For more information, see the topic Dialog Properties on p. 321.

Custom Dialog Builder Layout



Property	Value
Dialog Name	one_way_anova
Menu Location	Analyze/Compare Means/Simple One-Way ANOVA
Title	Simple One-Way ANOVA
Help File	
Web Deployment Properties	
Modeless	True

Canvas

The canvas is the area of the Custom Dialog Builder where you design the layout of your dialog.

Properties Pane

The properties pane is the area of the Custom Dialog Builder where you specify properties of the controls that make up the dialog as well as properties of the dialog itself, such as the menu location.

Tools Palette

The tools palette provides the set of controls that can be included in a custom dialog. You can show or hide the Tools Palette by choosing Tools Palette from the View menu.

Building a Custom Dialog

The basic steps involved in building a custom dialog are:

- Specify the properties of the dialog itself, such as the title that appears when the dialog is launched and the location of the new menu item for the dialog within the PASW Statistics menus. For more information, see the topic Dialog Properties on p. 321.

- Specify the controls, such as source and target variable lists, that make up the dialog and any sub-dialogs. For more information, see the topic Control Types on p. 330.

- Create the syntax template that specifies the command syntax to be generated by the dialog. For more information, see the topic Building the Syntax Template on p. 325.

- Install the dialog to PASW Statistics and/or save the specification for the dialog to a custom dialog package (*.spd*) file. For more information, see the topic Managing Custom Dialogs on p. 328.

You can preview your dialog as you're building it. For more information, see the topic Previewing a Custom Dialog on p. 327.

Dialog Properties

To view and set Dialog Properties:

- Click on the canvas in an area outside of any controls. With no controls on the canvas, Dialog Properties are always visible.

Dialog Name. The Dialog Name property is required and specifies a unique name to associate with the dialog. This is the name used to identify the dialog when installing or uninstalling it. To minimize the possibility of name conflicts, you may want to prefix the dialog name with an identifier for your organization, such as a URL.

Menu Location. Click the ellipsis (...) button to open the Menu Location dialog box, which allows you to specify the name and location of the menu item for the custom dialog. Adding a menu location for a new custom dialog or changing the menu location of an installed custom dialog requires a restart of PASW Statistics.

Title. The Title property specifies the text to be displayed in the title bar of the dialog box.

Help File. The Help File property is optional and specifies the path to a help file for the dialog. This is the file that will be launched when the user clicks the Help button on the dialog. Help files must be in HTML format. A copy of the specified help file is included with the specifications for the dialog when the dialog is installed or saved to a custom dialog package file. The Help button on the run-time dialog is hidden if there is no associated help file.

Any supporting files, such as image files and style sheets, must be stored along with the main help file once the custom dialog has been installed. By default, the specifications for an installed custom dialog are stored in the *ext/lib/<Dialog Name>* folder of the installation directory for Windows and Linux. For Mac, specifications are stored under the */Library/Application Support/SPSSInc/PASWStatistics/<version>/CustomDialogs/<Dialog Name>* folder, where <version> is the two digit PASW Statistics version—for example, 18. Supporting files should be located at the root of the folder and not in sub-folders. They must be manually added to any custom dialog package files you create for the dialog.

If you have specified alternate locations for installed dialogs—using the *SPSS_CDIALOGS_PATH* environment variable—then store any supporting files under the *<Dialog Name>* folder at the appropriate alternate location. For more information, see the topic Managing Custom Dialogs on p. 328.

Note: When working with a dialog opened from a custom dialog package (*.spd*) file, the Help File property points to a temporary folder associated with the *.spd* file. Any modifications to the help file should be made to the copy in the temporary folder.

Web Deployment Properties. Allows you to associate a properties file with this dialog for use in building thin client applications that are deployed over the web.

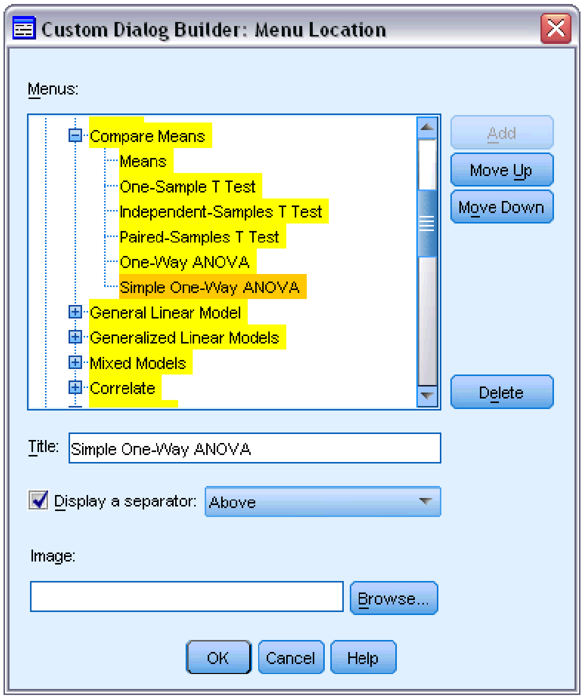
Modeless. Specifies whether the dialog is modal or modeless. When a dialog is modal, it must be closed before the user can interact with the main application windows (Data, Output, and Syntax) or with other open dialogs. Modeless dialogs do not have that constraint. The default is modeless.

Syntax. The Syntax property specifies the syntax template, used to create the command syntax generated by the dialog at run-time. Click the ellipsis (...) button to open the Syntax Template. For more information, see the topic Building the Syntax Template on p. 325.

Required Add-Ons. Specifies one or more add-ons—such as the Python or R Integration Plug-in—that are required in order to run the command syntax generated by the dialog. For example, if the dialog generates command syntax for an extension command implemented in R, then check the box for the R Integration Plug-in. Users will be warned about required add-ons that are missing when they try to install or run the dialog.

Specifying the Menu Location for a Custom Dialog

Figure 18-1
Custom Dialog Builder Menu Location dialog box



Title: Simple One-Way ANOVA

☑ Display a separator: Above

The Menu Location dialog box allows you to specify the name and location of the menu item for a custom dialog. Adding a menu location for a new custom dialog or changing the menu location of an installed custom dialog requires a restart of PASW Statistics. Menu items for custom dialogs do not appear in the Menu Editor within PASW Statistics.

▶ Double-click the menu (or click the plus sign icon) to which you want to add the item for the new dialog. You can also add items to the top-level menu labelled Custom (located between the Graphs and Utilities items), which is only available for menu items associated with custom dialogs.

If you want to create custom menus or sub-menus, use the Menu Editor. For more information, see the topic Menu Editor in Chapter 17 on p. 313. Note, however, that other users of your dialog will have to manually create the same menu or sub-menu from their Menu Editor; otherwise, the dialog will be added to their Custom menu.

Note: The Menu Location dialog box displays all menus, including those for all add-on modules. Be sure to add the menu item for your custom dialog to a menu that will be available to you or end users of your dialog.

▶ Select the menu item above which you want the item for the new dialog to appear. Once the item is added you can use the Move Up and Move Down buttons to reposition it.

▶ Enter a title for the menu item. Titles within a given menu or sub-menu must be unique.

▶ Click Add.

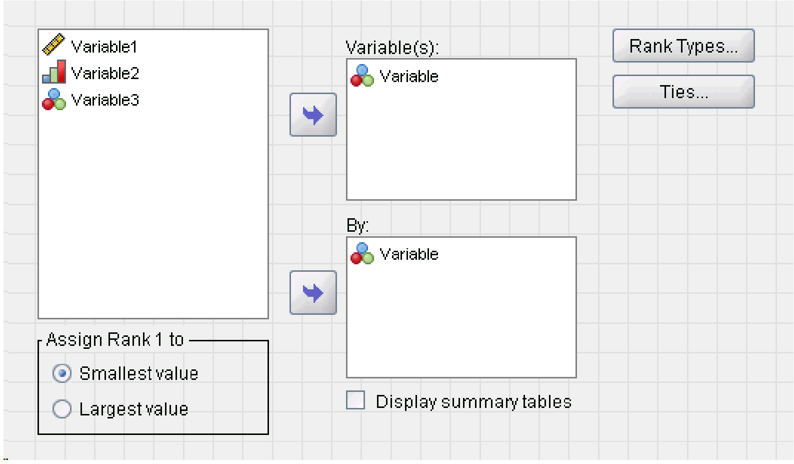
Optionally, you can:

- Add a separator above or below the new menu item.
- Specify the path to an image that will appear next to the menu item for the custom dialog. The supported image types are *gif* and *png*. The image cannot be larger than 16 x 16 pixels.

Laying Out Controls on the Canvas

You add controls to a custom dialog by dragging them from the tools palette onto the canvas. To ensure consistency with built-in dialogs, the canvas is divided into three functional columns in which you can place controls.

Figure 18-2
Canvas structure



Left Column. The left column is primarily intended for a source list control. All controls other than target lists and sub-dialog buttons can be placed in the left column.

Center Column. The center column can contain any control other than source lists and sub-dialog buttons.

Right Column. The right column can only contain sub-dialog buttons.

Although not shown on the canvas, each custom dialog contains OK, Paste, Cancel, and Help buttons positioned across the bottom of the dialog. The presence and locations of these buttons is automatic, however, the Help button is hidden if there is no help file associated with the dialog (as specified by the Help File property in Dialog Properties).

You can change the vertical order of the controls within a column by dragging them up or down, but the exact position of the controls will be determined automatically for you. At run-time, controls will resize in appropriate ways when the dialog itself is resized. Controls such as source and target lists automatically expand to fill the available space below them.

Building the Syntax Template

The syntax template specifies the command syntax that will be generated by the custom dialog. A single custom dialog can generate command syntax for any number of built-in PASW Statistics commands or extension commands.

The syntax template may consist of static text that is always generated and control identifiers that are replaced at run-time with the values of the associated custom dialog controls. For example, command names and subcommand specifications that don't depend on user input would be static text, whereas the set of variables specified in a target list would be represented with the control identifier for the target list control.

To Build the Syntax Template

▶ From the menus in the Custom Dialog Builder choose:

Edit
  Syntax Template

(Or click the ellipsis (...) button in the Syntax property field in Dialog Properties)

▶ For static command syntax that does not depend on user-specified values, enter the syntax as you would in the Syntax Editor.

▶ Add control identifiers of the form `%%Identifier%%` at the locations where you want to insert command syntax generated by controls, where `Identifier` is the value of the Identifier property for the control. You can select from a list of available control identifiers by pressing Ctrl+Spacebar. If you manually enter identifiers, retain any spaces, since all spaces in identifiers are significant.

At run-time, and for all controls other than check boxes and check box groups, each identifier is replaced with the current value of the Syntax property of the associated control. For check boxes and check box groups, the identifier is replaced by the current value of the Checked Syntax or Unchecked Syntax property of the associated control, depending on the current state of the control—checked or unchecked. For more information, see the topic Control Types on p. 330.

Note: The syntax generated at run-time automatically includes a command terminator (period) as the very last character if one is not present.

Example: Including Run-time Values in the Syntax Template

Consider a simplified version of the Frequencies dialog that only contains a source list control and a target list control, and generates command syntax of the following form:

```
FREQUENCIES VARIABLES=var1 var2...
  /FORMAT = NOTABLE
  /BARCHART.
```

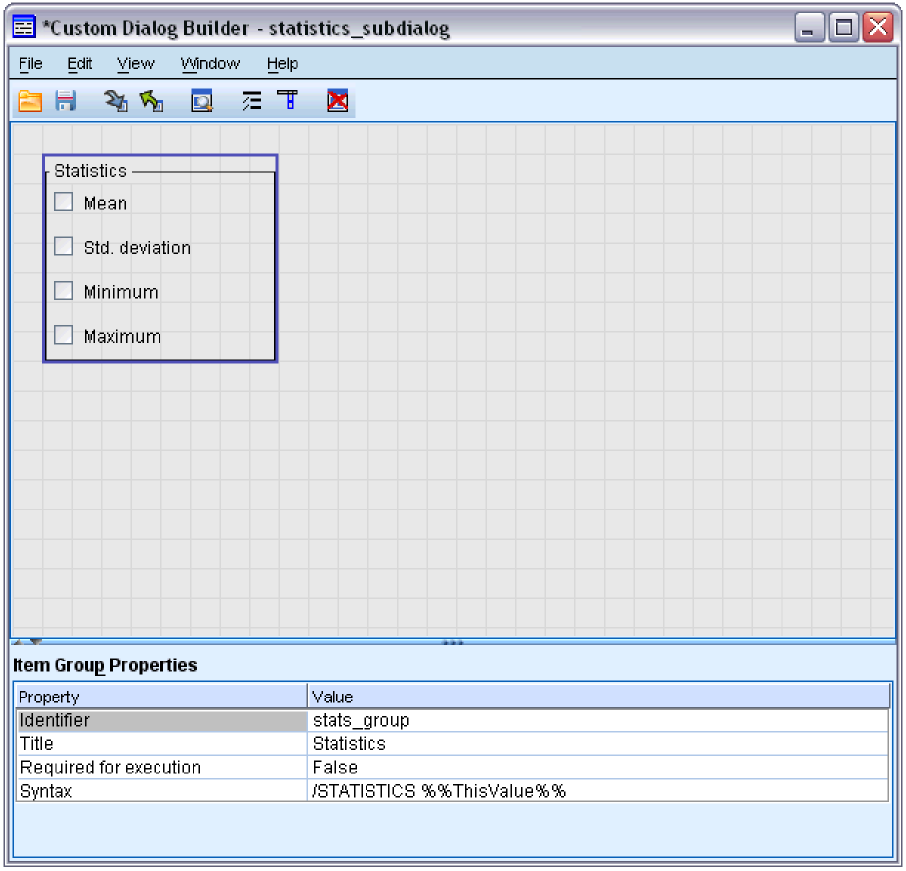
The syntax template to generate this might look like:

```
FREQUENCIES VARIABLES=%%target_list%%
  /FORMAT = NOTABLE
  /BARCHART.
```

- %%target_list%% is the value of the Identifier property for the target list control. At run-time it will be replaced by the current value of the Syntax property of the control.
- Defining the Syntax property of the target list control to be %%ThisValue%% specifies that at run-time, the current value of the property will be the value of the control, which is the set of variables in the target list.

Example: Including Command Syntax from Container Controls

Building on the previous example, consider adding a Statistics sub-dialog that contains a single group of check boxes that allow a user to specify mean, standard deviation, minimum, and maximum. Assume the check boxes are contained in an item group control, as shown in the following figure.



Item Group Properties

Property	Value
Identifier	stats_group
Title	Statistics
Required for execution	False
Syntax	/STATISTICS %%ThisValue%%

An example of the generated command syntax would be:

```
FREQUENCIES VARIABLES=var1 var2...
  /FORMAT = NOTABLE
  /STATISTICS MEAN STDDEV
  /BARCHART.
```

The syntax template to generate this might look like:

```
FREQUENCIES VARIABLES=%%target_list%%
  /FORMAT = NOTABLE
  %%stats_group%%
  /BARCHART.
```

- `%%target_list%%` is the value of the Identifier property for the target list control and `%%stats_group%%` is the value of the Identifier property for the item group control.

The following table shows one way to specify the Syntax properties of the item group and the check boxes contained within it in order to generate the desired result. The Syntax property of the target list would be set to `%%ThisValue%%`, as described in the previous example.

Syntax property of item group	`/STATISTICS %%ThisValue%%`
Checked Syntax property of mean check box	`MEAN`
Checked Syntax property of stddev check box	`STDDEV`
Checked Syntax property of min check box	`MINIMUM`
Checked Syntax property of max check box	`MAXIMUM`

At run-time, `%%stats_group%%` will be replaced by the current value of the Syntax property for the item group control. Specifically, `%%ThisValue%%` will be replaced by a blank-separated list of the values of the Checked or Unchecked Syntax property of each check box, depending on its state—checked or unchecked. Since we only specified values for the Checked Syntax property, only checked boxes will contribute to `%%ThisValue%%`. For example, if the user checks the mean and standard deviation boxes, the run-time value of the Syntax property for the item group will be `/STATISTICS MEAN STDDEV`.

If no boxes are checked, then the Syntax property for the item group control will be empty, and the generated command syntax will not contain any reference to `%%stats_group%%`. This may or may not be desirable. For example, even with no boxes checked you may still want to generate the `STATISTICS` subcommand. This can be accomplished by referencing the identifiers for the check boxes directly in the syntax template, as in:

```
FREQUENCIES VARIABLES=%%target_list%%
  /FORMAT = NOTABLE
  /STATISTICS %%stats_mean%% %%stats_stddev%% %%stats_min%% %%stats_max%%
  /BARCHART.
```

Previewing a Custom Dialog

You can preview the dialog that is currently open in the Custom Dialog Builder. The dialog appears and functions as it would when run from the menus within PASW Statistics.

- Source variable lists are populated with dummy variables that can be moved to target lists.
- The Paste button pastes command syntax into the designated syntax window.
- The OK button closes the preview.
- If a help file is specified, the Help button is enabled and will open the specified file. If no help file is specified, the help button is disabled when previewing, and hidden when the actual dialog is run.

To preview a custom dialog:

▶ From the menus in the Custom Dialog Builder choose:

File
  Preview Dialog

Managing Custom Dialogs

The Custom Dialog Builder allows you to manage custom dialogs created by you or by other users. You can install, uninstall, or modify installed dialogs; and you can save specifications for a custom dialog to an external file or open a file containing the specifications for a custom dialog in order to modify it. Custom dialogs must be installed before they can be used.

Installing a Custom Dialog

You can install the dialog that is currently open in the Custom Dialog Builder or you can install a dialog from a custom dialog package (*.spd*) file. Re-installing an existing dialog will replace the existing version.

To install the currently open dialog:

▶ From the menus in the Custom Dialog Builder choose:

File
  Install

To install from a custom dialog package file:

▶ From the menus choose:

Utilities
  Custom Dialogs
    Install Custom Dialog...

To use a new custom dialog, you will need to restart PASW Statistics. Re-installing an existing dialog does not require a restart unless the menu location changes.

For Windows and Linux, and by default, installing a dialog requires write permission to the PASW Statistics installation directory. If you do not have write permissions to the required location or would like to store installed dialogs elsewhere, you can specify one or more alternate locations by defining the *SPSS_CDIALOGS_PATH* environment variable. When present, the paths specified in *SPSS_CDIALOGS_PATH* take precedence over the default location. Custom dialogs will be installed to the first writable location. Note that Mac users may also utilize the *SPSS_CDIALOGS_PATH* environment variable. For multiple locations, separate each with a semicolon on Windows and a colon on Linux and Mac. The specified locations must exist on the target machine. After setting *SPSS_CDIALOGS_PATH* , you will need to restart PASW Statistics for the changes to take effect.

To create the *SPSS_CDIALOGS_PATH* environment variable on Windows, from the Control Panel:

Windows XP

▶ Select System.

- ▶ Select the Advanced tab and click Environment Variables.
- ▶ In the User variables section, click New, enter *SPSS_CDIALOGS_PATH* in the Variable name field and the desired path(s) in the Variable value field.

Windows Vista or Windows 7

- ▶ Select User Accounts.
- ▶ Select Change my environment variables.
- ▶ Click New, enter *SPSS_CDIALOGS_PATH* in the Variable name field and the desired path(s) in the Variable value field.

To view the current locations for custom dialogs, run the following command syntax: `SHOW EXTPATHS`.

Opening an Installed Custom Dialog

You can open a custom dialog that you have already installed, allowing you to modify it and/or save it externally so that it can be distributed to other users.

- ▶ From the menus in the Custom Dialog Builder choose:

File
  Open Installed

Note: If you are opening an installed dialog in order to modify it, choosing File > Install will re-install it, replacing the existing version.

Uninstalling a Custom Dialog

- ▶ From the menus in the Custom Dialog Builder choose:

File
  Uninstall

The menu item for the custom dialog will be removed the next time you start PASW Statistics.

Saving to a Custom Dialog Package File

You can save the specifications for a custom dialog to an external file, allowing you to distribute the dialog to other users or save specifications for a dialog that is not yet installed. Specifications are saved to a custom dialog package (*.spd*) file.

- ▶ From the menus in the Custom Dialog Builder choose:

File
  Save

Opening a Custom Dialog Package File

You can open a custom dialog package file containing the specifications for a custom dialog, allowing you to modify the dialog and re-save it or install it.

▶ From the menus in the Custom Dialog Builder choose:

File
  Open

Manually Copying an Installed Custom Dialog

By default, the specifications for an installed custom dialog are stored in the *ext/lib/<Dialog Name>* folder of the installation directory for Windows and Linux. For Mac, specifications are stored under the */Library/Application Support/SPSSInc/PASWStatistics/<version>/CustomDialogs/<Dialog Name>* folder, where <version> is the two digit PASW Statistics version—for example, 18. You can copy this folder to the same relative location in another instance of PASW Statistics and it will be recognized as an installed dialog the next time that instance is launched.

- If you have specified alternate locations for installed dialogs—using the *SPSS_CDIALOGS_PATH* environment variable—then copy the *<Dialog Name>* folder from the appropriate alternate location.
- If alternate locations for installed dialogs have been defined for the instance of PASW Statistics you are copying to, then you can copy to any one of the specified locations and the dialog will be recognized as an installed dialog the next time that instance is launched.

Control Types

The tools palette provides the controls that can be added to a custom dialog.

- **Source List:** A list of source variables from the active dataset. For more information, see the topic Source List on p. 331.
- **Target List:** A target for variables transferred from the source list. For more information, see the topic Target List on p. 331.
- **Check Box:** A single check box. For more information, see the topic Check Box on p. 333.
- **Combo Box:** A combo box for creating drop-down lists. For more information, see the topic Combo Box and List Box Controls on p. 333.
- **List Box:** A list box for creating single selection or multiple selection lists. For more information, see the topic Combo Box and List Box Controls on p. 333.
- **Text control:** A text box that accepts arbitrary text as input. For more information, see the topic Text Control on p. 335.
- **Number control:** A text box that is restricted to numeric values as input. For more information, see the topic Number Control on p. 335.
- **Static Text control:** A control for displaying static text. For more information, see the topic Static Text Control on p. 336.
- **Item Group:** A container for grouping a set of controls, such as a set of check boxes. For more information, see the topic Item Group on p. 336.
- **Radio Group:** A group of radio buttons. For more information, see the topic Radio Group on p. 337.
- **Check Box Group:** A container for a set of controls that are enabled or disabled as a group, by a single check box. For more information, see the topic Check Box Group on p. 338.

- **File Browser:** A control for browsing the file system to open or save a file. For more information, see the topic File Browser on p. 339.
- **Sub-dialog Button:** A button for launching a sub-dialog. For more information, see the topic Sub-dialog Button on p. 341.

Source List

The Source Variable List control displays the list of variables from the active dataset that are available to the end user of the dialog. You can display all variables from the active dataset (the default) or you can filter the list based on type and measurement level—for instance, numeric variables that have a measurement level of scale. Use of a Source List control implies the use of one or more Target List controls. The Source List control has the following properties:

Identifier. The unique identifier for the control.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control. The specified text only appears when hovering over the title area of the control. Hovering over one of the listed variables will display the variable name and label.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Variable Transfers. Specifies whether variables transferred from the source list to a target list remain in the source list (Copy Variables), or are removed from the source list (Move Variables).

Variable Filter. Allows you to filter the set of variables displayed in the control. You can filter on variable type and measurement level, and you can specify that multiple response sets are included in the variable list. Click the ellipsis (...) button to open the Filter dialog. You can also open the Filter dialog by double-clicking the Source List control on the canvas. For more information, see the topic Filtering Variable Lists on p. 332.

Note: The Source List control cannot be added to a sub-dialog.

Target List

The Target List control provides a target for variables that are transferred from the source list. Use of the Target List control assumes the presence of a Source List control. You can specify that only a single variable can be transferred to the control or that multiple variables can be transferred to it, and you can constrain which types of variables can be transferred to the control—for instance, only numeric variables with a measurement level of nominal or ordinal. The Target List control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control. The specified text only appears when hovering over the title area of the control. Hovering over one of the listed variables will display the variable name and label.

Target list type. Specifies whether multiple variables or only a single variable can be transferred to the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the OK and Paste buttons will be disabled until a value is specified for this control. If False is specified, the absence of a value in this control has no effect on the state of the OK and Paste buttons. The default is True.

Variable Filter. Allows you to constrain the types of variables that can be transferred to the control. You can constrain by variable type and measurement level, and you can specify whether multiple response sets can be transferred to the control. Click the ellipsis (...) button to open the Filter dialog. You can also open the Filter dialog by double-clicking the Target List control on the canvas. For more information, see the topic Filtering Variable Lists on p. 332.

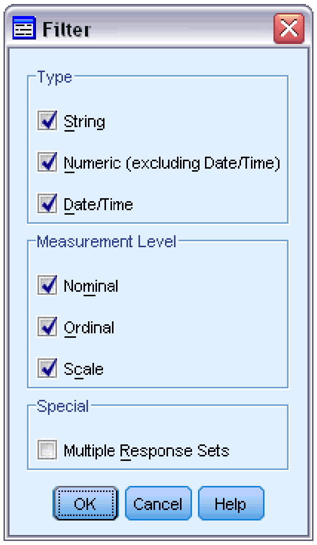
Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- The value `%%ThisValue%%` specifies the run-time value of the control, which is the list of variables transferred to the control. This is the default.

Note: The Target List control cannot be added to a sub-dialog.

Filtering Variable Lists

Figure 18-3
Filter dialog box



Type

☑ String

☑ Numeric (excluding Date/Time)

☑ Date/Time

Measurement Level

☑ Nominal

☑ Ordinal

☑ Scale

Special

☐ Multiple Response Sets

The Filter dialog box, associated with source and target variable lists, allows you to filter the types of variables from the active dataset that can appear in the lists. You can also specify whether multiple response sets associated with the active dataset are included. Numeric variables include all numeric formats except date and time formats.

Check Box

The Check Box control is a simple check box that can generate different command syntax for the checked versus the unchecked state. The Check Box control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. The label that is displayed with the check box. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Default Value. The default state of the check box—checked or unchecked.

Checked/Unchecked Syntax. Specifies the command syntax that is generated when the control is checked and when it is unchecked. To include the command syntax in the syntax template, use the value of the Identifier property. The generated syntax, whether from the Checked Syntax or Unchecked Syntax property, will be inserted at the specified position(s) of the identifier. For example, if the identifier is *checkbox1*, then at run-time, instances of `%%checkbox1%%` in the syntax template will be replaced by the value of the Checked Syntax property when the box is checked and the Unchecked Syntax property when the box is unchecked.

- You can specify any valid command syntax and you can use `\n` for line breaks.

Combo Box and List Box Controls

The Combo Box control allows you to create a drop-down list that can generate command syntax specific to the selected list item. It is limited to single selection. The List Box control allows you to display a list of items that support single or multiple selection and generate command syntax specific to the selected item(s). The Combo Box and List Box controls have the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

List Items. Click the ellipsis (...) button to open the List Item Properties dialog box, which allows you to specify the list items of the control. You can also open the List Item Properties dialog by double-clicking the Combo Box or List Box control on the canvas.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

List Box Type (List Box only). Specifies whether the list box supports single selection only or multiple selection. You can also specify that items are displayed as a list of check boxes.

Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- The value `%%ThisValue%%` specifies the run-time value of the control and is the default. If the list items are manually defined, the run-time value is the value of the Syntax property for the selected list item. If the list items are based on a target list control, the run-time value is the value of the selected list item. For multiple selection list box controls, the run-time value is a blank-separated list of the selected items. For more information, see the topic Specifying List Items for Combo Boxes and List Boxes on p. 334.
- You can specify any valid command syntax and you can use `\n` for line breaks.

Specifying List Items for Combo Boxes and List Boxes

The List Item Properties dialog box allows you to specify the list items of a combo box or list box control.

Manually defined values. Allows you to explicitly specify each of the list items.

- **Identifier.** A unique identifier for the list item.
- **Name.** The name that appears in the list for this item. The name is a required field.
- **Default.** For a combo box, specifies whether the list item is the default item displayed in the combo box. For a list box, specifies whether the list item is selected by default.
- **Syntax.** Specifies the command syntax that is generated when the list item is selected.
- You can specify any valid command syntax and you can use `\n` for line breaks.

Note: You can add a new list item in the blank line at the bottom of the existing list. Entering any of the properties other than the identifier will generate a unique identifier, which you can keep or modify. You can delete a list item by clicking on the *Identifier* cell for the item and pressing delete.

Values based on the contents of a target list control. Specifies that the list items are dynamically populated with values associated with the variables in a selected target list control. Select an existing target list control as the source of the list items, or enter the value of the Identifier property for a target list control into the text area of the Target List combo box. The latter approach allows you to enter the Identifier for a target list control that you plan to add later.

- **Variable Names.** Populate the list items with the names of the variables in the specified target list control.
- **Value Labels.** Populate the list items with the union of the value labels associated with the variables in the specified target list control. You can choose whether the command syntax generated by the associated combo box or list box control contains the selected value label or its associated value.

- **Custom Attribute.** Populate the list items with the union of the attribute values associated with variables in the target list control that contain the specified custom attribute.
- **Syntax.** Displays the Syntax property of the associated combo box or list box control, allowing you to make changes to the property. For more information, see the topic Combo Box and List Box Controls on p. 333.

Text Control

The Text control is a simple text box that can accept arbitrary input, and has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Text Content. Specifies whether the contents are arbitrary or whether the text box must contain a string that adheres to rules for PASW Statistics variable names.

Default Value. The default contents of the text box.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the OK and Paste buttons will be disabled until a value is specified for this control. If False is specified, the absence of a value in this control has no effect on the state of the OK and Paste buttons. The default is False.

Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- The value `%%ThisValue%%` specifies the run-time value of the control, which is the content of the text box. This is the default.
- If the Syntax property includes `%%ThisValue%%` and the run-time value of the text box is empty, then the text box control does not generate any command syntax.

Number Control

The Number control is a text box for entering a numeric value, and has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Numeric Type. Specifies any limitations on what can be entered. A value of Real specifies that there are no restrictions on the entered value, other than it be numeric. A value of Integer specifies that the value must be an integer.

Default Value. The default value, if any.

Minimum Value. The minimum allowed value, if any.

Maximum Value. The maximum allowed value, if any.

Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the OK and Paste buttons will be disabled until a value is specified for this control. If False is specified, the absence of a value in this control has no effect on the state of the OK and Paste buttons. The default is False.

Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- The value `%%ThisValue%%` specifies the run-time value of the control, which is the numeric value. This is the default.
- If the Syntax property includes `%%ThisValue%%` and the run-time value of the number control is empty, then the number control does not generate any command syntax.

Static Text Control

The Static Text control allows you to add a block of text to your dialog, and has the following properties:

Identifier. The unique identifier for the control.

Title. The content of the text block. For multi-line content, use `\n` to specify line breaks.

Item Group

The Item Group control is a container for other controls, allowing you to group and control the syntax generated from multiple controls. For example, you have a set of check boxes that specify optional settings for a subcommand, but only want to generate the syntax for the subcommand if at least one box is checked. This is accomplished by using an Item Group control as a container for the check box controls. The following types of controls can be contained in an Item Group: check box, combo box, text control, number control, static text, radio group, and file browser. The Item Group control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title for the group. For multi-line titles, use `\n` to specify line breaks.

Required for execution. True specifies that the OK and Paste buttons will be disabled until at least one control in the group has a value. The default is False.

For example, the group consists of a set of check boxes. If Required for execution is set to True and all of the boxes are unchecked, then OK and Paste will be disabled.

Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- You can include identifiers for any controls contained in the item group. At run-time the identifiers are replaced with the syntax generated by the controls.
- The value `%%ThisValue%%` generates a blank-separated list of the syntax generated by each control in the item group, in the order in which they appear in the group (top to bottom). This is the default. If the Syntax property includes `%%ThisValue%%` and no syntax is generated by any of the controls in the item group, then the item group as a whole does not generate any command syntax.

Radio Group

The Radio Group control is a container for a set of radio buttons, each of which can contain a set of nested controls. The Radio Group control has the following properties:

Identifier. The unique identifier for the control.This is the identifier to use when referencing the control in the syntax template.

Title. An optional title for the group. If omitted, the group border is not displayed. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Radio Buttons. Click the ellipsis (...) button to open the Radio Group Properties dialog box, which allows you to specify the properties of the radio buttons as well as to add or remove buttons from the group. The ability to nest controls under a given radio button is a property of the radio button and is set in the Radio Group Properties dialog box. *Note*: You can also open the Radio Group Properties dialog by double-clicking the Radio Group control on the canvas.

Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- The value `%%ThisValue%%` specifies the run-time value of the radio button group, which is the value of the Syntax property for the selected radio button. This is the default. If the Syntax property includes `%%ThisValue%%` and no syntax is generated by the selected radio button, then the radio button group does not generate any command syntax.

Defining Radio Buttons

The Radio Button Group Properties dialog box allows you to specify a group of radio buttons.

Identifier. A unique identifier for the radio button.

Name. The name that appears next to the radio button. The name is a required field.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the name to use as a mnemonic. The specified character must exist in the name.

Nested Group. Specifies whether other controls can be nested under this radio button. The default is false. When the nested group property is set to true, a rectangular drop zone is displayed, nested and indented, under the associated radio button. The following controls can be nested under a radio button: check box, text, static text, number, combo box, list box, and file browser.

Default. Specifies whether the radio button is the default selection.

Syntax. Specifies the command syntax that is generated when the radio button is selected.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- For radio buttons containing nested controls, the value `%%ThisValue%%` generates a blank-separated list of the syntax generated by each nested control, in the order in which they appear under the radio button (top to bottom).

You can add a new radio button in the blank line at the bottom of the existing list. Entering any of the properties other than the identifier will generate a unique identifier, which you can keep or modify. You can delete a radio button by clicking on the *Identifier* cell for the button and pressing delete.

Check Box Group

The Check Box Group control is a container for a set of controls that are enabled or disabled as a group, by a single check box. The following types of controls can be contained in a Check Box Group: check box, combo box, text control, number control, static text, radio group, and file browser. The Check Box Group control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title for the group. If omitted, the group border is not displayed. For multi-line titles, use `\n` to specify line breaks.

Checkbox Title. An optional label that is displayed with the controlling check box. Supports `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Default Value. The default state of the controlling check box—checked or unchecked.

Checked/Unchecked Syntax. Specifies the command syntax that is generated when the control is checked and when it is unchecked. To include the command syntax in the syntax template, use the value of the Identifier property. The generated syntax, whether from the Checked Syntax or Unchecked Syntax property, will be inserted at the specified position(s) of the identifier. For example, if the identifier is *checkboxgroup1*, then at run-time, instances of

`%%checkboxgroup1%%` in the syntax template will be replaced by the value of the Checked Syntax property when the box is checked and the Unchecked Syntax property when the box is unchecked.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- You can include identifiers for any controls contained in the check box group. At run-time the identifiers are replaced with the syntax generated by the controls.
- The value `%%ThisValue%%` can be used in either the Checked Syntax or Unchecked Syntax property. It generates a blank-separated list of the syntax generated by each control in the check box group, in the order in which they appear in the group (top to bottom).
- By default, the Checked Syntax property has a value of `%%ThisValue%%` and the Unchecked Syntax property is blank.

File Browser

The File Browser control consists of a text box for a file path and a browse button that opens a standard PASW Statistics dialog to open or save a file. The File Browser control has the following properties:

Identifier. The unique identifier for the control. This is the identifier to use when referencing the control in the syntax template.

Title. An optional title that appears above the control. For multi-line titles, use `\n` to specify line breaks.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

File System Operation. Specifies whether the dialog launched by the browse button is appropriate for opening files or for saving files. A value of Open indicates that the browse dialog validates the existence of the specified file. A value of Save indicates that the browse dialog does not validate the existence of the specified file.

Browser Type. Specifies whether the browse dialog is used to select a file (Locate File) or to select a folder (Locate Folder).

File Filter. Click the ellipsis (...) button to open the File Filter dialog box, which allows you to specify the available file types for the open or save dialog. By default, all file types are allowed. *Note*: You can also open the File Filter dialog by double-clicking the File Browser control on the canvas.

File System Type. In distributed analysis mode, this specifies whether the open or save dialog browses the file system on which PASW Statistics Server is running or the file system of your local computer. Select Server to browse the file system of the server or Client to browse the file system of your local computer. The property has no effect in local analysis mode.

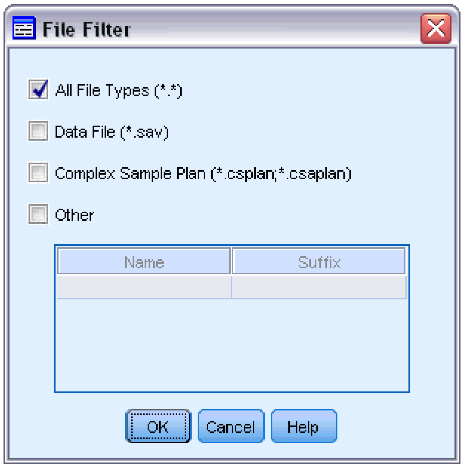
Required for execution. Specifies whether a value is required in this control in order for execution to proceed. If True is specified, the OK and Paste buttons will be disabled until a value is specified for this control. If False is specified, the absence of a value in this control has no effect on the state of the OK and Paste buttons. The default is False.

Syntax. Specifies command syntax that is generated by this control at run-time and can be inserted in the syntax template.

- You can specify any valid command syntax and you can use `\n` for line breaks.
- The value `%%ThisValue%%` specifies the run-time value of the text box, which is the file path, specified manually or populated by the browse dialog. This is the default.
- If the Syntax property includes `%%ThisValue%%` and the run-time value of the text box is empty, then the file browser control does not generate any command syntax.

File Type Filter

Figure 18-4
File Filter dialog box



File Filter

☑ All File Types (*.*)

□ Data File (*.sav)

□ Complex Sample Plan (*.csplan;*.csaplan)

□ Other

Name	Suffix

The File Filter dialog box allows you to specify the file types displayed in the Files of type and Save as type drop-down lists for open and save dialogs accessed from a File System Browser control. By default, all file types are allowed.

To specify file types not explicitly listed in the dialog box:

▶ Select Other.

▶ Enter a name for the file type.

▶ Enter a file type using the form *.suffix—for example, *.xls. You can specify multiple file types, each separated by a semicolon.

Sub-dialog Button

The Sub-dialog Button control specifies a button for launching a sub-dialog and provides access to the Dialog Builder for the sub-dialog. The Sub-dialog Button has the following properties:

Identifier. The unique identifier for the control.

Title. The text that is displayed in the button.

ToolTip. Optional ToolTip text that appears when the user hovers over the control.

Sub-dialog. Click the ellipsis (...) button to open the Custom Dialog Builder for the sub-dialog. You can also open the builder by double-clicking on the Sub-dialog button.

Mnemonic Key. An optional character in the title to use as a keyboard shortcut to the control. The character appears underlined in the title. The shortcut is activated by pressing Alt+[mnemonic key]. The Mnemonic Key property is not supported on Mac.

Note: The Sub-dialog Button control cannot be added to a sub-dialog.

Dialog Properties for a Sub-dialog

To view and set properties for a sub-dialog:

▶ Open the sub-dialog by double-clicking on the button for the sub-dialog in the main dialog, or single-click the sub-dialog button and click the ellipsis (...) button for the Sub-dialog property.

▶ In the sub-dialog, click on the canvas in an area outside of any controls. With no controls on the canvas, the properties for a sub-dialog are always visible.

Sub-dialog Name. The unique identifier for the sub-dialog. The Sub-dialog Name property is required.

Note: If you specify the Sub-dialog Name as an identifier in the Syntax Template—as in `%%My Sub-dialog Name%%`—it will be replaced at run-time with a blank-separated list of the syntax generated by each control in the sub-dialog, in the order in which they appear (top to bottom and left to right).

Title. Specifies the text to be displayed in the title bar of the sub-dialog box. The Title property is optional but recommended.

Help File. Specifies the path to an optional help file for the sub-dialog. This is the file that will be launched when the user clicks the Help button on the sub-dialog, and may be the same help file specified for the main dialog. Help files must be in HTML format. See the description of the Help File property for Dialog Properties for more information.

Syntax. Click the ellipsis (...) button to open the Syntax Template. For more information, see the topic Building the Syntax Template on p. 325.

Custom Dialogs for Extension Commands

Extension commands are user-defined PASW Statistics commands that are implemented in either the Python programming language or R. Once deployed to an instance of PASW Statistics, an extension command is run in the same manner as any built-in PASW Statistics command.

You can use the Custom Dialog Builder to create dialogs for extension commands, and you can install custom dialogs for extension commands created by other users.

Creating Custom Dialogs for Extension Commands

Whether an extension command was written by you or another user, you can create a custom dialog for it. The syntax template for the dialog should generate the command syntax for the extension command. If the custom dialog is only for your use, install the dialog. Assuming the extension command is already deployed on your system, you will be able to run the command from the installed dialog.

If you are creating a custom dialog for an extension command and wish to share it with other users, you should first save the specifications for the dialog to a custom dialog package (*.spd*) file. You'll then want to create an extension bundle containing the custom dialog package file, the XML file that specifies the syntax of the extension command, and the implementation code file(s) written in Python or R. The extension bundle is then what you share with other users. For more information, see the topic Working with Extension Bundles in Chapter 15 on p. 284.

Installing Extension Commands with Associated Custom Dialogs

An extension command with an associated custom dialog consists of three pieces: an XML file that specifies the syntax of the command; one or more code files (Python or R) that implement the command; and a custom dialog package file that contains the specifications for the custom dialog. If the extension command and its associated custom dialog are distributed in an extension bundle (*.spe*) file, then you can simply install the bundle from Utilities>Extension Bundles>Install Extension Bundle. Otherwise, you need to install the custom dialog and the extension command files separately as follows:

Custom Dialog Package File. Install the custom dialog package file from Utilities>Custom Dialogs>Install Custom Dialog.

XML Syntax Specification File and Implementation Code. For Windows and Linux, the XML file specifying the syntax of the extension command and the implementation code (Python or R) should be placed in the *extensions* directory under the PASW Statistics installation directory. For Mac, the XML and code files should be placed in the */Library/Application Support/SPSSInc/PASWStatistics/18/extensions* directory.

- For Windows and Linux, If you do not have write permissions to the PASW Statistics installation directory or would like to store the XML file and the implementation code elsewhere, you can specify one or more alternate locations by defining the *SPSS_EXTENSIONS_PATH* environment variable. When present, the paths specified in *SPSS_EXTENSIONS_PATH* take precedence over the PASW Statistics installation directory. For multiple locations, separate each with a semicolon on Windows and a colon on Linux.

  To create the *SPSS_EXTENSIONS_PATH* environment variable on Windows, follow the same general steps used to create the *SPSS_CDIALOGS_PATH* variable. See the section on Installing a Custom Dialog in Managing Custom Dialogs on p. 328.

To view the current locations for custom dialogs, run the following command syntax: `SHOW EXTPATHS`.

- For an extension command implemented in Python, you can always store the associated Python module(s) to a location on the Python search path, such as the Python *site-packages* directory.

Creating Localized Versions of Custom Dialogs

You can create localized versions of custom dialogs for any of the languages supported by PASW Statistics. You can localize any string that appears in a custom dialog and you can localize the optional help file.

To Localize Dialog Strings

▶ Make a copy of the properties file associated with the dialog. The properties file contains all of the localizable strings associated with the dialog. By default, it is located in the *ext/lib/<Dialog Name>* folder of the PASW Statistics installation directory for Windows and Linux, and under the */Library/Application Support/SPSSInc/PASWStatistics/18/CustomDialogs/<Dialog Name>* folder for Mac. The copy should reside in that same folder and not in a sub-folder.

If you have specified alternate locations for installed dialogs—using the *SPSS_CDIALOGS_PATH* environment variable—then the copy should reside in the *<Dialog Name>* folder at the appropriate alternate location. For more information, see the topic Managing Custom Dialogs on p. 328.

▶ Rename the copy to *<Dialog Name>_<language identifier>.properties*, using the language identifiers in the table below. For example, if the dialog name is *mydialog* and you want to create a Japanese version of the dialog, then the localized properties file should be named *mydialog_ja.properties*. Localized properties files must be manually added to any custom dialog package files you create for the dialog. A custom dialog package file is simply a ZIP file that can be opened and modified with an application such as WinZip on Windows.

▶ Open the new properties file with a text editor that supports UTF-8, such as Notepad on Windows, or the TextEdit application on Mac. Modify the values associated with any properties that need to be localized, but do not modify the names of the properties. Properties associated with a specific control are prefixed with the identifier for the control. For example, the ToolTip property for a control with the identifier *options_button* is *options_button_tooltip_LABEL*. Title properties are simply named *<identifier>_LABEL*, as in *options_button_LABEL*.

When the dialog is launched, PASW Statistics will search for a properties file whose language identifier matches the current language, as specified by the Language drop-down on the General tab in the Options dialog box. If no such properties file is found, the default file *<Dialog Name>.properties* will be used.

To Localize the Help File

▶ Make a copy of the help file associated with the dialog and localize the text for the desired language. The copy must reside in the same folder as the help file and not in a sub-folder. The help file should reside in the *ext/lib/<Dialog Name>* folder of the PASW Statistics installation directory for Windows and Linux, and under the */Library/Application Support/SPSSInc/PASWStatistics/18/CustomDialogs/<Dialog Name>* folder for Mac.

If you have specified alternate locations for installed dialogs—using the *SPSS_CDIALOGS_PATH* environment variable—then the copy should reside in the *<Dialog Name>* folder at the appropriate alternate location. For more information, see the topic Managing Custom Dialogs on p. 328.

- Rename the copy to *<Help File>_<language identifier>*, using the language identifiers in the table below. For example, if the help file is *myhelp.htm* and you want to create a German version of the file, then the localized help file should be named *myhelp_de.htm*. Localized help files must be manually added to any custom dialog package files you create for the dialog. A custom dialog package file is simply a ZIP file that can be opened and modified with an application such as WinZip on Windows.

If there are supplementary files such as image files that also need to be localized, you will need to manually modify the appropriate paths in the main help file to point to the localized versions, which should be stored along with the original versions.

When the dialog is launched, PASW Statistics will search for a help file whose language identifier matches the current language, as specified by the Language drop-down on the General tab in the Options dialog box. If no such help file is found, the help file specified for the dialog (the file specified in the Help File property of Dialog Properties) is used.

Language Identifiers

de	German
en	English
es	Spanish
fr	French
it	Italian
ja	Japanese
ko	Korean
pl	Polish
ru	Russian
zh_CN	Simplified Chinese
zh_TW	Traditional Chinese

Note: Text in custom dialogs and associated help files is not limited to the languages supported by PASW Statistics. You are free to write the dialog and help text in any language without creating language-specific properties and help files. All users of your dialog will then see the text in that language.

Chapter

19

Production Jobs

Production jobs provide the ability to run PASW Statistics in an automated fashion. The program runs unattended and terminates after executing the last command, so you can perform other tasks while it runs or schedule the production job to run automatically at scheduled times. Production jobs are useful if you often run the same set of time-consuming analyses, such as weekly reports.

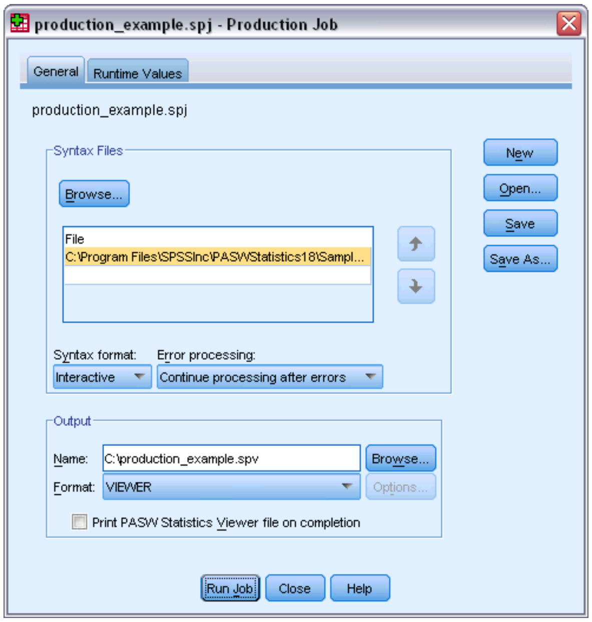
Production jobs use command syntax files to tell PASW Statistics what to do. A command syntax file is a simple text file containing command syntax. You can use any text editor to create the file. You can also generate command syntax by pasting dialog box selections into a syntax window or by editing the journal file. For more information, see the topic Working with Command Syntax in Chapter 13 on p. 256.

To create a production job:

▶ From the menus in any window choose:

Utilities
  Production Job

Figure 19-1
Production Job dialog box



production_example.spj

File
C:\Program Files\SPSSInc\PASWStatistics18\Sampl...

Syntax format: Interactive
Error processing: Continue processing after errors

Name: C:\production_example.spv
Format: VIEWER

□ Print PASW Statistics Viewer file on completion

▶ Click New to create a new production job or Open to open an existing production job.

Note: Production Facility job files (.spp) created in releases prior to 16.0 will not run in release 16.0 or later. A conversion utility is available to convert Windows and Macintosh Production Facility job files to production jobs (.spj). For more information, see the topic Converting Production Facility Files on p. 352.

▶ Select one or more command syntax files.

▶ Select the output file name, location, and format.

▶ Click Save or Save As to save the production job.

Syntax format. Controls the form of the syntax rules used for the job.

- **Interactive.** Each command must end with a period. Periods can appear anywhere within the command, and commands can continue on multiple lines, but a period as the last nonblank character on a line is interpreted as the end of the command. Continuation lines and new commands can start anywhere on a new line. These are the "interactive" rules in effect when you select and run commands in a syntax window.
- **Batch.** Each command must start at the beginning of a new line (no blank spaces before the start of the command), and continuation lines must be indented at least one space. If you want to indent new commands, you can use a plus sign, dash, or period as the first character at the start of the line and then indent the actual command. The period at the end of the command is optional. This setting is compatible with the syntax rules for command files included with the `INCLUDE` command.

  Note: Do *not* use the Batch option if your syntax files contain `GGRAPH` command syntax that includes `GPL` statements. `GPL` statements will only run under interactive rules.

Error Processing. Controls the treatment of error conditions in the job.

- **Continue processing after errors.** Errors in the job do not automatically stop command processing. The commands in the production job files are treated as part of the normal command stream, and command processing continues in the normal fashion.
- **Stop processing immediately.** Command processing stops when the first error in a production job file is encountered. This is compatible with the behavior of command files included with the `INCLUDE` command.

Output. Controls the name, location, and format of the production job results. The following format options are available:

- **Viewer file to disk (.spv).** Results are saved in PASW Statistics Viewer format in the specified file location.
- **Viewer file (.spv) to PES Repository.** This requires the Adaptor for Enterprise Services option.
- **Web Reports (.spw) to PES Repository.** This requires the Adaptor for Enterprise Services option.
- **Word/RTF (*.doc).** Pivot tables are exported as Word tables with all formatting attributes intact—for example, cell borders, font styles, and background colors. Text output is exported as formatted RTF. Charts, tree diagrams, and model views are included in PNG format.

  Note: Microsoft Word may not display extremely wide tables properly.
- **Excel (*.xls).** Pivot table rows, columns, and cells are exported as Excel rows, columns, and cells, with all formatting attributes intact—for example, cell borders, font styles, and background colors. Text output is exported with all font attributes intact. Each line in the text

output is a row in the Excel file, with the entire contents of the line contained in a single cell. Charts, tree diagrams, and model views are included in PNG format.

- **HTML (*.htm).** Pivot tables are exported as HTML tables. Text output is exported as preformatted HTML. Charts, tree diagrams, and model views are embedded by reference, and you should export charts in a suitable format for inclusion in HTML documents (for example, PNG and JPEG).
- **Portable Document Format (*.pdf).** All output is exported as it appears in Print Preview, with all formatting attributes intact.
- **PowerPoint file (*.ppt).** Pivot tables are exported as Word tables and are embedded on separate slides in the PowerPoint file, with one slide for each pivot table. All formatting attributes of the pivot table are retained—for example, cell borders, font styles, and background colors. Charts, tree diagrams, and model views are exported in TIFF format. Text output is not included.

  Export to PowerPoint is available only on Windows operating systems.
- **Text (*.txt).** Text output formats include plain text, UTF-8, and UTF-16. Pivot tables can be exported in tab-separated or space-separated format. All text output is exported in space-separated format. For charts, tree diagrams, and model views, a line is inserted in the text file for each graphic, indicating the image filename.

Print Viewer file on completion. Sends the final Viewer output file to the printer on completion of the production job.

Run Job. This runs the production job in a separate session. This is particularly useful for testing new production jobs before deploying them.

Production Jobs with OUTPUT Commands

Production jobs honor `OUTPUT` commands, such as `OUTPUT SAVE`, `OUTPUT ACTIVATE`, and `OUTPUT NEW`. `OUTPUT SAVE` commands executed during the course of a production job will write the contents of the specified output documents to the specified locations. This is in addition to the output file created by the production job. When using `OUTPUT NEW` to create a new output document, it is recommended that you explicitly save it with the `OUTPUT SAVE` command.

A production job output file consists of the contents of the active output document as of the end of the job. For jobs containing `OUTPUT` commands, the output file may not contain all output created in the session. For example, suppose the production job consists of a number of procedures followed by an `OUTPUT NEW` command, followed by more procedures but no more `OUTPUT` commands. The `OUTPUT NEW` command defines a new active output document. At the end of the production job, it will contain output from only the procedures executed after the `OUTPUT NEW` command.

HTML Options

Table Options. No table options are available for HTML format. All pivot tables are converted to HTML tables.

Image Options. The available image types are: EPS, JPEG, TIFF, PNG, and BMP. On Windows operating systems, EMF (enhanced metafile) format is also available. You can also scale the image size from 1% to 200%.

PowerPoint Options

Table Options. You can use the Viewer outline entries as slide titles. Each slide contains a single output item. The title is formed from the outline entry for the item in the outline pane of the Viewer.

Image Options. You can scale the image size from 1% to 200%. (All images are exported to PowerPoint in TIFF format.)

Note: PowerPoint format is only available on Windows operating systems and requires PowerPoint 97 or later.

PDF Options

Embed bookmarks. This option includes bookmarks in the PDF document that correspond to the Viewer outline entries. Like the Viewer outline pane, bookmarks can make it much easier to navigate documents with a large number of output objects.

Embed fonts. Embedding fonts ensures that the PDF document will look the same on all computers. Otherwise, if some fonts used in the document are not available on the computer being used to view (or print) the PDF document, font substitution may yield suboptimal results.

Text Options

Table Options. Pivot tables can be exported in tab-separated or space-separated format. For space-separated format, you can also control:

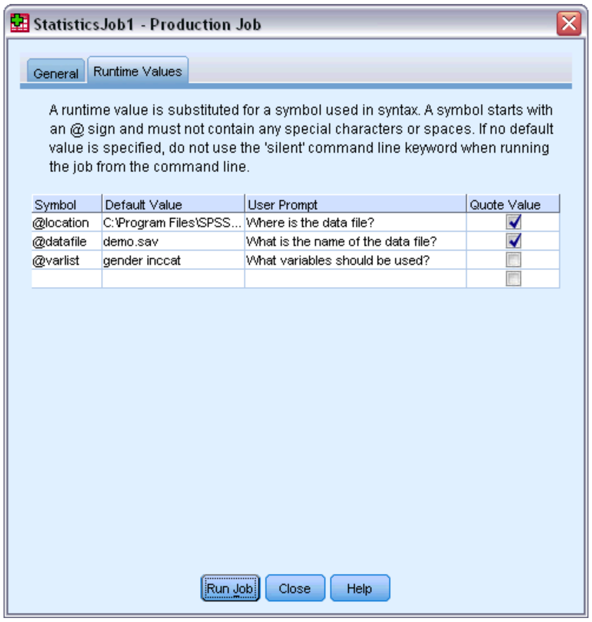
- **Column Width.** Autofit does not wrap any column contents, and each column is as wide as the widest label or value in that column. Custom sets a maximum column width that is applied to all columns in the table, and values that exceed that width wrap onto the next line in that column.
- **Row/Column Border Character.** Controls the characters used to create row and column borders. To suppress display of row and column borders, enter blank spaces for the values.

Image Options. The available image types are: EPS, JPEG, TIFF, PNG, and BMP. On Windows operating systems, EMF (enhanced metafile) format is also available. You can also scale the image size from 1% to 200%.

Runtime Values

Runtime values defined in a production job file and used in a command syntax file simplify tasks such as running the same analysis for different data files or running the same set of commands for different sets of variables. For example, you could define the runtime value *@datafile* to prompt you for a data filename each time you run a production job that uses the string *@datafile* in place of a filename in the command syntax file.

Figure 19-2
Production Job dialog box, Runtime Values tab



A runtime value is substituted for a symbol used in syntax. A symbol starts with an @ sign and must not contain any special characters or spaces. If no default value is specified, do not use the 'silent' command line keyword when running the job from the command line.

Symbol	Default Value	User Prompt	Quote Value
@location	C:\Program Files\SPSS...	Where is the data file?	☑
@datafile	demo.sav	What is the name of the data file?	☑
@varlist	gender inccat	What variables should be used?	☐
			☐

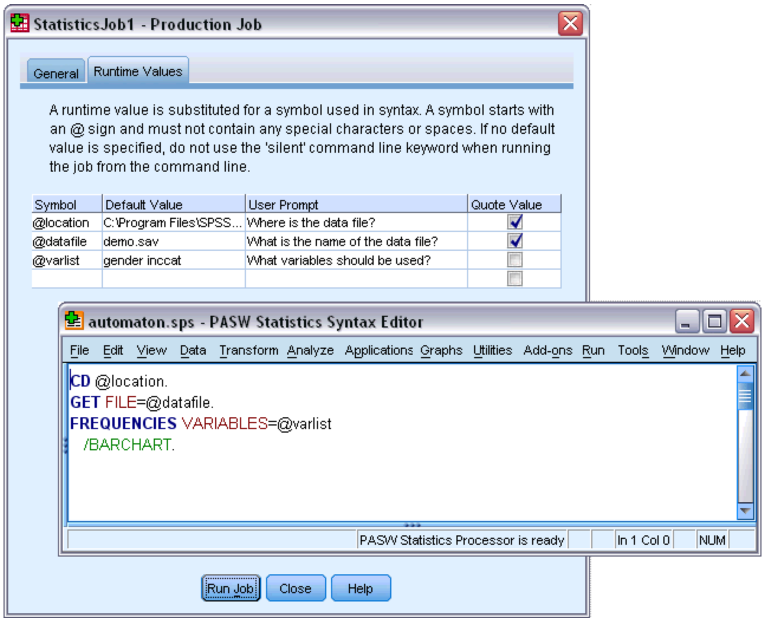
Symbol. The string in the command syntax file that triggers the production job to prompt the user for a value. The symbol name must begin with an @ sign and must conform to variable naming rules. For more information, see the topic Variable Names in Chapter 5 on p. 70.

Default Value. The value that the production job supplies by default if you don't enter a different value. This value is displayed when the production job prompts you for information. You can replace or modify the value at runtime. If you don't provide a default value, don't use the `silent` keyword when running the production job with command line switches, unless you also use the `-symbol` switch to specify runtime values. For more information, see the topic Running Production Jobs from a Command Line on p. 350.

User Prompt. The descriptive label that is displayed when the production job prompts you to enter information. For example, you could use the phrase "What data file do you want to use?" to identify a field that requires a data filename.

Quote Value. Encloses the default value or the value entered by the user in quotes. For example, file specifications should be enclosed in quotes.

Figure 19-3
Runtime symbols in a command syntax file


StatisticsJob1 - Production Job
General
Runtime Values
automaton.sps - PASW Statistics Syntax Editor
File Edit View Data Transform Analyze Applications Graphs Utilities Add-ons Run Tools Window Help
PASW Statistics Processor is ready
In 1 Col 0
NUM
Run Job
Close
Help

A runtime value is substituted for a symbol used in syntax. A symbol starts with an @ sign and must not contain any special characters or spaces. If no default value is specified, do not use the 'silent' command line keyword when running the job from the command line.

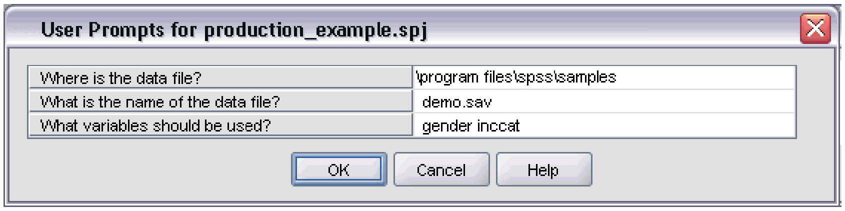
Symbol	Default Value	User Prompt	Quote Value
@location	C:\Program Files\SPSS...	Where is the data file?	☑
@datafile	demo.sav	What is the name of the data file?	☑
@varlist	gender inccat	What variables should be used?	□
			□

```
CD @location.
GET FILE=@datafile.
FREQUENCIES VARIABLES=@varlist
  /BARCHART.
```

User Prompts

A production job prompts you for values whenever you run a production job that contains defined runtime symbols. You can replace or modify the default values that are displayed. Those values are then substituted for the runtime symbols in all command syntax files associated with the production job.

Figure 19-4
Production User Prompts dialog box



Where is the data file?	\program files\spss\samples
What is the name of the data file?	demo.sav
What variables should be used?	gender inccat

Running Production Jobs from a Command Line

Command line switches enable you to schedule production jobs to run automatically at certain times, using scheduling utilities available on your operating system. The basic form of the command line argument is:

```
paswstat filename.spj -production
```

Depending on how you invoke the production job, you may need to include directory paths for the *paswstat* executable file (located in the directory in which the application is installed) and/or the production job file.

You can run production jobs from a command line with the following switches:

-production [prompt|silent]. Start the application in production mode. The `prompt` and `silent` keywords specify whether to display the dialog box that prompts for runtime values if they are specified in the job. The `prompt` keyword is the default and shows the dialog box. The `silent` keyword suppresses the dialog box. If you use the `silent` keyword, you can define the runtime symbols with the `-symbol` switch. Otherwise, the default value is used. The `-switchserver` and `-singleseat` switches are ignored when using the `-production` switch.

-symbol <values>. List of symbol-value pairs used in the production job. Each symbol name starts with @. Values that contain spaces should be enclosed in quotes. Rules for including quotes or apostrophes in string literals may vary across operating systems, but enclosing a string that includes single quotes or apostrophes in double quotes usually works (for example, `“'a quoted value'”`).

To run production jobs on a remote server in distributed analysis mode, you also need to specify the server login information:

-server <inet:hostname:port>. The name or IP address and port number of the server. Windows only.

-user <name>. A valid user name. If a domain name is required, precede the user name with the domain name and a backslash (\). Windows only.

-password <password>. The user’s password. Windows only.

Example

```
paswstat \production_jobs\prodjob1.spj -production silent -symbol @datafile /data/July_data.sav
```

- This example assumes that you are running the command line from the installation directory, so no path is required for the *paswstat* executable file.
- This example also assumes that the production job specifies that the value for *@datafile* should be quoted (Quote Value checkbox on the Runtime Values tab), so no quotes are necessary when specifying the data file on the command line. Otherwise, you would need to specify something like "'/data/July_data.sav'" to include quotes with the data file specification, since file specifications should be quoted in command syntax.
- The directory path for the location of the production job uses the Windows back slash convention. On Macintosh and Linux, use forward slashes. The forward slashes in the quoted data file specification will work on all operating systems since this quoted string is inserted into the command syntax file and forward slashes are acceptable in commands that include file specifications (for example, `GET FILE`, `GET DATA`, `SAVE`) on all operating systems.
- The `silent` keyword suppresses any user prompts in the production job, and the `-symbol` switch inserts the quoted data file name and location wherever the runtime symbol *@datafile* appears in the command syntax files included in the production job.

Converting Production Facility Files

Production Facility job files (.spp) created in releases prior to 16.0 will not run in release 16.0 or later. For Windows and Macintosh Production Facility job files created in earlier releases, you can use *prodconvert*, located in the installation directory, to convert those files to new production job files (.spj). Run *prodconvert* from a command window using the following specifications:

```
[installpath]\prodconvert [filepath]\filename.spp
```

where *[installpath]* is the location of the folder in which PASW Statistics is installed and *[filepath]* is the location of the folder where the original production job file. A new file with the same name but with the extension *.spj* is created in the same folder as the original file. (Note: If the path contains spaces, enclose each path and file specification in double quotes. On Macintosh operating systems, use forward slashes instead on back slashes.)

Limitations

- WMF and EMF chart formats are not supported. PNG format is used in place of these formats.
- The export options Output Document (No Charts), Charts Only, and Nothing are not supported. All output objects supported by the selected format are included.
- Remote server settings are ignored. To specify remote server settings for distributed analysis, you need to run the production job from a command line, using command line switches to specify the server settings. For more information, see the topic Running Production Jobs from a Command Line on p. 350.
- Publish to Web settings are ignored.

Chapter

20

Output Management System

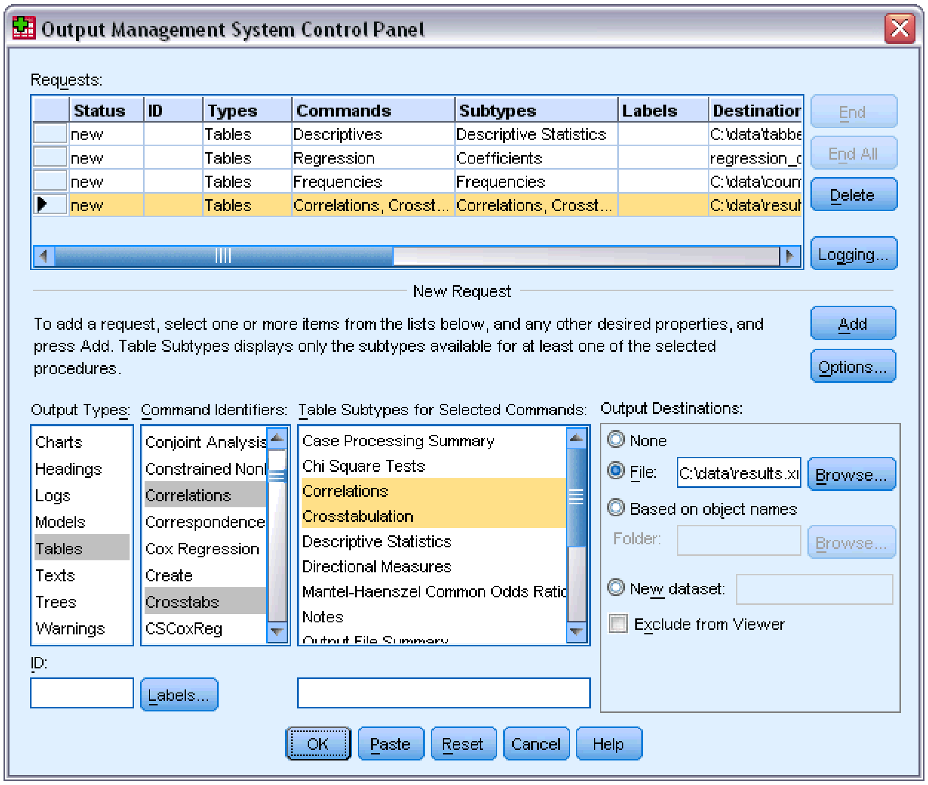
The Output Management System (OMS) provides the ability to automatically write selected categories of output to different output files in different formats. Formats include: Word, Excel, PDF, PASW Statistics data file format (.sav), Viewer file format (.spv), web report format (.spw), XML, HTML, and text. For more information, see the topic OMS Options on p. 359.

To Use the Output Management System Control Panel

▶ From the menus choose:

Utilities
  OMS Control Panel...

Figure 20-1
Output Management System Control Panel



Output Management System Control Panel

Requests:

	Status	ID	Types	Commands	Subtypes	Labels	Destinatior
	new		Tables	Descriptives	Descriptive Statistics		C:\data\tabbe
	new		Tables	Regression	Coefficients		regression_c
	new		Tables	Frequencies	Frequencies		C:\data\coun
▶	new		Tables	Correlations, Crosst...	Correlations, Crosst...		C:\data\resul

New Request

To add a request, select one or more items from the lists below, and any other desired properties, and press Add. Table Subtypes displays only the subtypes available for at least one of the selected procedures.

Output Types: Charts, Headings, Logs, Models, Tables, Texts, Trees, Warnings

Command Identifiers: Conjoint Analysis, Constrained Nonl, Correlations, Correspondence, Cox Regression, Create, Crosstabs, CSCoxReg

Table Subtypes for Selected Commands: Case Processing Summary, Chi Square Tests, Correlations, Crosstabulation, Descriptive Statistics, Directional Measures, Mantel-Haenszel Common Odds Ratio, Notes, Output File Summary

Output Destinations:

○ None
● File: C:\data\results.xı
○ Based on object names
Folder:
○ New dataset:
□ Exclude from Viewer

ID:

You can use the control panel to start and stop the routing of output to various destinations.

- Each OMS request remains active until explicitly ended or until the end of the session.

- A destination file that is specified on an OMS request is unavailable to other procedures and other applications until the OMS request is ended.
- While an OMS request is active, the specified destination files are stored in memory (RAM), so active OMS requests that write a large amount of output to external files may consume a large amount of memory.
- Multiple OMS requests are independent of each other. The same output can be routed to different locations in different formats, based on the specifications in different OMS requests.
- The order of the output objects in any particular destination is the order in which they were created, which is determined by the order and operation of the procedures that generate the output.

To Add New OMS Requests

▶ Select the output types (tables, charts, etc.) that you want to include. For more information, see the topic Output Object Types on p. 355.

▶ Select the commands to include. If you want to include all output, select all items in the list. For more information, see the topic Command Identifiers and Table Subtypes on p. 357.

▶ For commands that produce pivot table output, select the specific table types to include.

The list displays only the tables that are available in the selected commands; any table type that is available in one or more of the selected commands is displayed in the list. If no commands are selected, all table types are displayed. For more information, see the topic Command Identifiers and Table Subtypes on p. 357.

▶ To select tables based on text labels instead of subtypes, click Labels. For more information, see the topic Labels on p. 358.

▶ Click Options to specify the output format (for example, PASW Statistics data file, XML, or HTML). By default, Output XML format is used. For more information, see the topic OMS Options on p. 359.

▶ Specify an output destination:

- **File.** All selected output is routed to a single file.
- **Based on object names.** Output is routed to multiple destination files based on object names. A separate file is created for each output object, with a filename based on either table subtype names or table labels. Enter the destination folder name.
- **New dataset.** For PASW Statistics data file format output, you can route the output to a dataset. The dataset is available for subsequent use in the same session but is not saved unless you explicitly save it as a file prior to the end of the session. This option is available only for PASW Statistics data file format output. Dataset names must conform to variable-naming rules. For more information, see the topic Variable Names in Chapter 5 on p. 70.

▶ Optionally:

- Exclude the selected output from the Viewer. If you select Exclude from Viewer, the output types in the OMS request will not be displayed in the Viewer window. If multiple active OMS requests include the same output types, the display of those output types in the Viewer is determined by the most recent OMS request that contains those output types. For more information, see the topic Excluding Output Display from the Viewer on p. 364.

- Assign an ID string to the request. All requests are automatically assigned an ID value, and you can override the system default ID string with a descriptive ID, which can be useful if you have multiple active requests that you want to identify easily. ID values that you assign cannot start with a dollar sign ($).

The following tips are for selecting multiple items in a list:

- Press Ctrl+A to select all items in a list.
- Use Shift+click to select multiple contiguous items.
- Use Ctrl+click to select multiple noncontiguous items.

To End and Delete OMS Requests

Active and new OMS requests are displayed in the Requests list, with the most recent request at the top. You can change the widths of the information columns by clicking and dragging the borders, and you can scroll the list horizontally to see more information about a particular request.

An asterisk (*) after the word Active in the *Status* column indicates an OMS request that was created with command syntax that includes features that are not available in the Control Panel.

To end a specific, active OMS request:

▶ In the Requests list, click any cell in the row for the request.

▶ Click End.

To end all active OMS requests:

▶ Click End All.

To delete a new request (a request that has been added but is not yet active):

▶ In the Requests list, click any cell in the row for the request.

▶ Click Delete.

Note: Active OMS requests are not ended until you click OK.

Output Object Types

There are different types of output objects:

Charts. This includes charts created with the Chart Builder, charting procedures, and charts created by statistical procedures (for example, a bar chart created by the Frequencies procedure).

Headings. Text objects that are labeled *Title* in the outline pane of the Viewer. Heading text objects are not included with Output XML format.

Logs. Log text objects. Log objects contain certain types of error and warning messages. Depending on your Options settings (Edit menu, Options, Viewer tab), log objects may also contain the command syntax that is executed during the session. Log objects are labeled *Log* in the outline pane of the Viewer.

Models. Output objects displayed in the Model Viewer. A single model object can contain multiple views of the model, including both tables and charts.

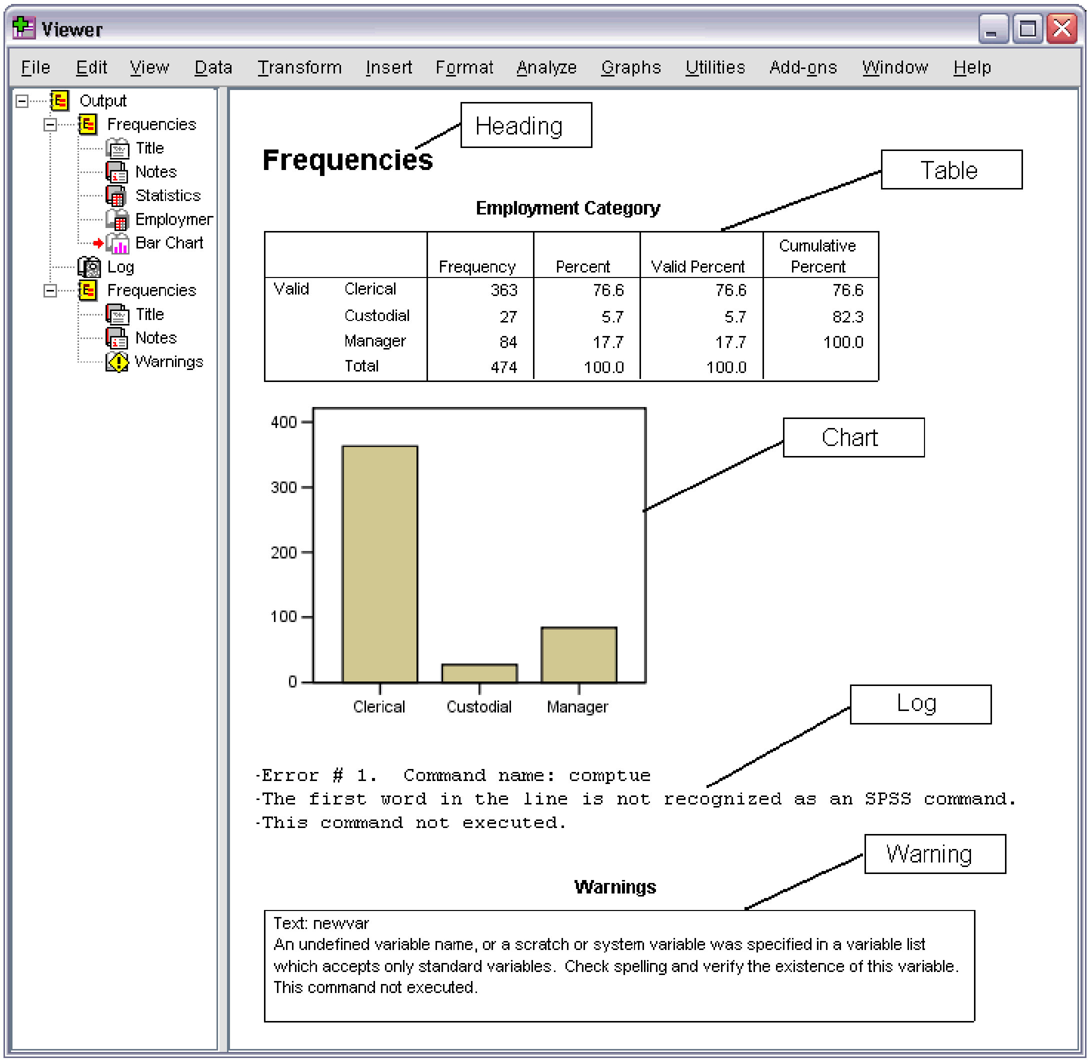
Tables. Output objects that are pivot tables in the Viewer (includes Notes tables). Tables are the only output objects that can be routed to PASW Statistics data file (.sav) format.

Texts. Text objects that aren't logs or headings (includes objects labeled *Text Output* in the outline pane of the Viewer).

Trees. Tree model diagrams that are produced by the Decision Tree option.

Warnings. Warning objects contain certain types of error and warning messages.

Figure 20-2
Output object types



Frequencies

Employment Category

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Clerical	363	76.6	76.6	76.6
	Custodial	27	5.7	5.7	82.3
	Manager	84	17.7	17.7	100.0
	Total	474	100.0	100.0	

```
-Error # 1.  Command name: comptue
-The first word in the line is not recognized as an SPSS command.
-This command not executed.
```

Warnings

Text: newvar
An undefined variable name, or a scratch or system variable was specified in a variable list which accepts only standard variables. Check spelling and verify the existence of this variable.
This command not executed.

Command Identifiers and Table Subtypes

Command Identifiers

Command identifiers are available for all statistical and charting procedures and any other commands that produce blocks of output with their own identifiable heading in the outline pane of the Viewer. These identifiers are usually (but not always) the same or similar to the procedure names on the menus and dialog box titles, which are usually (but not always) similar to the underlying command names. For example, the command identifier for the Frequencies procedure is “Frequencies,” and the underlying command name is also the same.

There are, however, some cases where the procedure name and the command identifier and/or the command name are not all that similar. For example, all of the procedures on the Nonparametric Tests submenu (from the Analyze menu) use the same underlying command, and the command identifier is the same as the underlying command name: Npar Tests.

Table Subtypes

Table subtypes are the different types of pivot tables that can be produced. Some subtypes are produced by only one command; other subtypes can be produced by multiple commands (although the tables may not look similar). Although table subtype names are generally descriptive, there can be many names to choose from (particularly if you have selected a large number of commands); also, two subtypes may have very similar names.

To Find Command Identifiers and Table Subtypes

When in doubt, you can find command identifiers and table subtype names in the Viewer window:

▶ Run the procedure to generate some output in the Viewer.

▶ Right-click the item in the outline pane of the Viewer.

▶ Choose Copy OMS Command Identifier or Copy OMS Table Subtype.

▶ Paste the copied command identifier or table subtype name into any text editor (such as a Syntax Editor window).

Labels

As an alternative to table subtype names, you can select tables based on the text that is displayed in the outline pane of the Viewer. You can also select other object types based on their labels. Labels are useful for differentiating between multiple tables of the same type in which the outline text reflects some attribute of the particular output object, such as the variable names or labels. There are, however, a number of factors that can affect the label text:

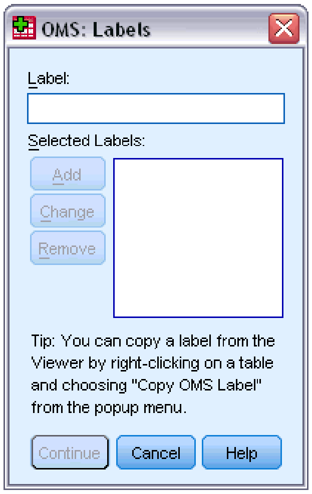
- If split-file processing is on, split-file group identification may be appended to the label.
- Labels that include information about variables or values are affected by your current output label options settings (Edit menu, Options, Output Labels tab).
- Labels are affected by the current output language setting (Edit menu, Options, General tab).

To Specify Labels to Use to Identify Output Objects

▶ In the Output Management System Control Panel, select one or more output types and then select one or more commands.

▶ Click Labels.

Figure 20-3
OMS Labels dialog box



Tip: You can copy a label from the Viewer by right-clicking on a table and choosing "Copy OMS Label" from the popup menu.

- ▶ Enter the label *exactly* as it appears in the outline pane of the Viewer window. (You can also right-click the item in the outline, choose Copy OMS Label, and paste the copied label into the Label text field.)
- ▶ Click Add.
- ▶ Repeat the process for each label that you want to include.
- ▶ Click Continue.

Wildcards

You can use an asterisk (*) as the last character of the label string as a wildcard character. All labels that begin with the specified string (except for the asterisk) will be selected. This process works only when the asterisk is the last character, because asterisks can appear as valid characters inside a label.

OMS Options

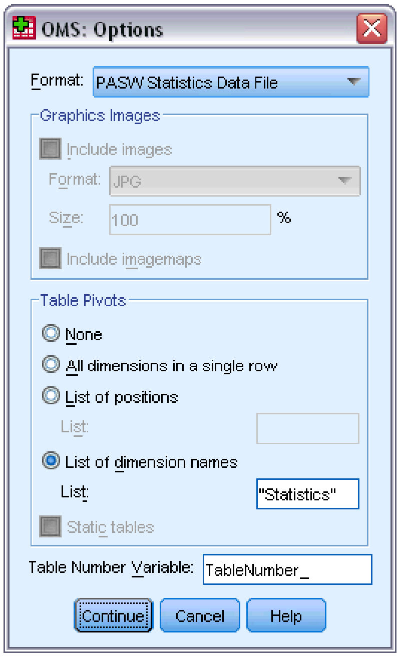
You can use the OMS Options dialog box to:

- Specify the output format.
- Specify the image format (for HTML and Output XML output formats).
- Specify what table dimension elements should go into the row dimension.
- Include a variable that identifies the sequential table number that is the source for each case (for PASW Statistics data file format).

To Specify OMS Options

- ▶ Click Options in the Output Management System Control Panel.

Figure 20-4
OMS Options dialog box



OMS: Options

Format: PASW Statistics Data File

Graphics Images

☐ Include images

Format: JPG

Size: 100 %

☐ Include imagemaps

Table Pivots

○ None

○ All dimensions in a single row

○ List of positions

List:

● List of dimension names

List: "Statistics"

☐ Static tables

Table Number Variable: TableNumber_

Format

Excel. Excel 97-2003 format. Pivot table rows, columns, and cells are exported as Excel rows, columns, and cells, with all formatting attributes intact — for example, cell borders, font styles, and background colors. Text output is exported with all font attributes intact. Each line in the text output is a row in the Excel file, with the entire contents of the line contained in a single cell. Charts, tree diagrams, and model views are included in PNG format.

HTML. Output objects that would be pivot tables in the Viewer are converted to simple HTML tables. No TableLook attributes (font characteristics, border styles, colors, etc.) are supported. Text output objects are tagged <PRE> in the HTML. Charts, tree diagrams, and model views are exported as separate files in the selected graphics format and are embedded by reference. Image file names use the HTML file name as the root name, followed by a sequential integer, starting with 0.

Output XML. XML that conforms to the *spss-output* schema.

PDF. Output is exported as it would appear in Print Preview, with all formatting attributes intact. The PDF file includes bookmarks that correspond to the entries in the Viewer outline pane.

PASW Statistics Data File. This format is a binary file format. All output object types other than tables are excluded. Each column of a table becomes a variable in the data file. To use a data file that is created with OMS in the same session, you must end the active OMS request before you can open the data file. For more information, see the topic Routing Output to PASW Statistics Data Files on p. 365.

Text. Space-separated text. Output is written as text, with tabular output aligned with spaces for fixed-pitch fonts. Charts, tree diagrams, and model views are excluded.

Tabbed Text. Tab-delimited text. For output that is displayed as pivot tables in the Viewer, tabs delimit table column elements. Text block lines are written as is; no attempt is made to divide them with tabs at useful places. Charts, tree diagrams, and model views are excluded.

Viewer File. This is the same format used when you save the contents of a Viewer window.

Web Report File. This output file format is designed for use with Predictive Enterprise Services. It is essentially the same as the PASW Statistics Viewer format except that tree diagrams are saved as static images.

Word/RTF. Pivot tables are exported as Word tables with all formatting attributes intact—for example, cell borders, font styles, and background colors. Text output is exported as formatted RTF. Charts, tree diagrams, and model views are included in PNG format.

Graphics Images

For HTML and Output XML formats, you can include charts, tree diagrams, and model views as image files. A separate image file is created for each chart and/or tree.

- For HTML document format, standard `<IMG SRC='filename'>` tags are included in the HTML document for each image file.
- For Output XML document format, the XML file contains a `chart` element with an `ImageFile` attribute of the general form `<chart imageFile="filepath/filename"/>` for each image file.
- Image files are saved in a separate subdirectory (folder). The subdirectory name is the name of the destination file, without any extension and with *_files* appended to the end. For example, if the destination file is *julydata.htm*, the images subdirectory will be named *julydata_files*.

Format. The available image formats are PNG, JPG, EMF, BMP, and VML.

- EMF (enhanced metafile) format is available only on Windows operating systems.
- VML image format is available only for HTML document format.
- VML image format does not create separate image files. The VML code that renders the image is embedded in the HTML.
- VML image format does not include tree diagrams.

Size. You can scale the image size from 10% to 200%.

Include Imagemaps. For HTML document format, this option creates image map ToolTips that display information for some chart elements, such as the value of the selected point on a line chart or bar on a bar chart.

Table Pivots

For pivot table output, you can specify the dimension element(s) that should appear in the columns. All other dimension elements appear in the rows. For PASW Statistics data file format, table columns become variables, and rows become cases.

- If you specify multiple dimension elements for the columns, they are nested in the columns in the order in which they are listed. For PASW Statistics data file format, variable names are constructed by nested column elements. For more information, see the topic Variable Names in OMS-Generated Data Files on p. 372.
- If a table doesn't contain any of the listed dimension elements, all dimension elements for that table will appear in the rows.
- Table pivots that are specified here have no effect on tables that are displayed in the Viewer.

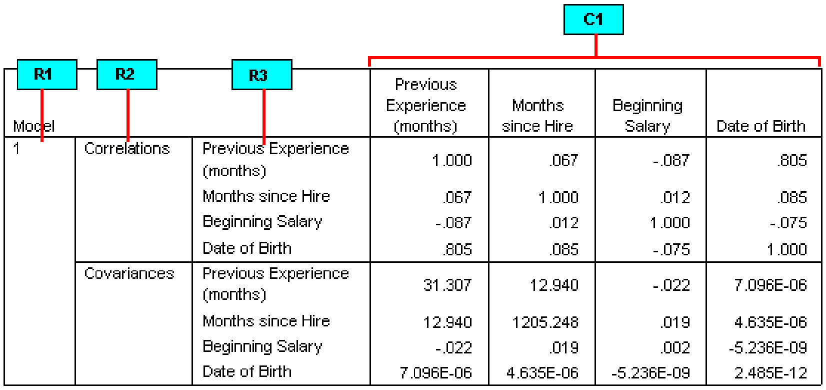
Each dimension of a table—row, column, layer—may contain zero or more elements. For example, a simple two-dimensional crosstabulation contains a single row dimension element and a single column dimension element, each of which contains one of the variables that are used in the table. You can use either positional arguments or dimension element "names" to specify the dimension elements that you want to put into the column dimension.

All dimensions in rows. Creates a single row for each table. For PASW Statistics format data files, this means each table is a single case, and all the table elements are variables.

List of positions. The general form of a positional argument is a letter indicating the default position of the element—C for column, R for row, or L for layer—followed by a positive integer indicating the default position within that dimension. For example, R1 would indicate the outermost row dimension element.

- To specify multiple elements from multiple dimensions, separate each dimension with a space—for example, R1 C2.
- The dimension letter followed by ALL indicates all elements in that dimension in their default order. For example, CALL is the same as the default behavior (using all column elements in their default order to create columns).
- CALL RALL LALL (or RALL CALL LALL, and so on) will put all dimension elements into the columns. For PASW Statistics data file format, this creates one row/case per table in the data file.

Figure 20-5
Row and column positional arguments



Model			Previous Experience (months)	Months since Hire	Beginning Salary	Date of Birth
1	Correlations	Previous Experience (months)	1.000	.067	-.087	.805
		Months since Hire	.067	1.000	.012	.085
		Beginning Salary	-.087	.012	1.000	-.075
		Date of Birth	.805	.085	-.075	1.000
	Covariances	Previous Experience (months)	31.307	12.940	-.022	7.096E-06
		Months since Hire	12.940	1205.248	.019	4.635E-06
		Beginning Salary	-.022	.019	.002	-5.236E-09
		Date of Birth	7.096E-06	4.635E-06	-5.236E-09	2.485E-12

List of dimension names. As an alternative to positional arguments, you can use dimension element "names," which are the text labels that appear in the table. For example, a simple two-dimensional crosstabulation contains a single row dimension element and a single column dimension element, each with labels based on the variables in those dimensions, plus a single layer dimension element labeled *Statistics* (if English is the output language).

- Dimension element names may vary, based on the output language and/or settings that affect the display of variable names and/or labels in tables.
- Each dimension element name must be enclosed in single or double quotation marks. To specify multiple dimension element names, include a space between each quoted name.

The labels that are associated with the dimension elements may not always be obvious.

To See All Dimension Elements and Their Labels for a Pivot Table

▶ Activate (double-click) the table in the Viewer.

▶ From the menus choose:

View
  Show All

and/or

▶ If the pivoting trays aren’t displayed, from the menus choose:

Pivot
  Pivoting Trays

The element labels are dispalyed in the pivoting trays.

Figure 20-6
Dimension element names displayed in table and pivoting trays

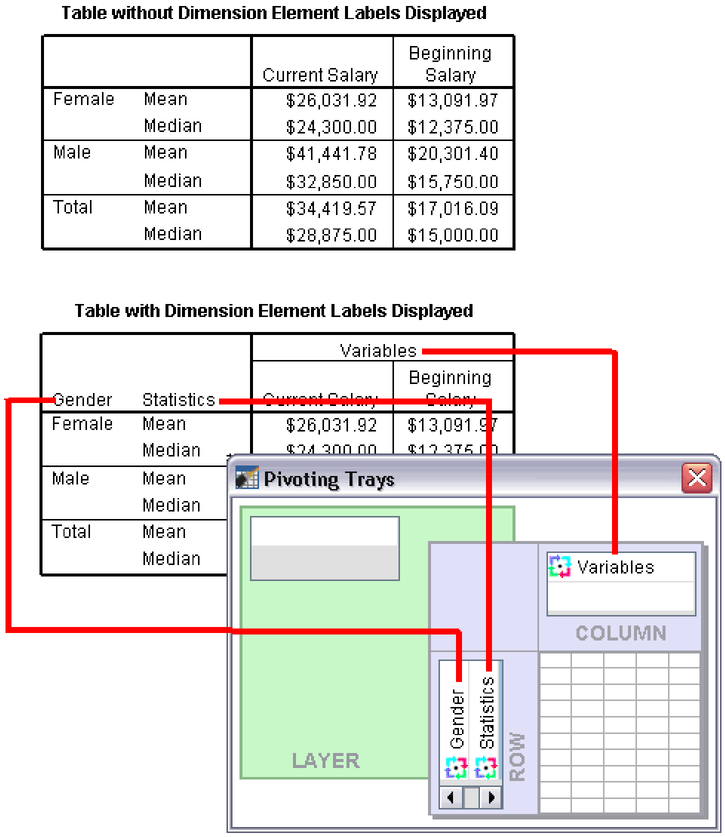


Table without Dimension Element Labels Displayed

		Current Salary	Beginning Salary
Female	Mean	$26,031.92	$13,091.97
	Median	$24,300.00	$12,375.00
Male	Mean	$41,441.78	$20,301.40
	Median	$32,850.00	$15,750.00
Total	Mean	$34,419.57	$17,016.09
	Median	$28,875.00	$15,000.00

Table with Dimension Element Labels Displayed

		Variables	
Gender	Statistics	Current Salary	Beginning Salary
Female	Mean	$26,031.92	$13,091.97
	Median		
Male	Mean		
	Median		
Total	Mean		
	Median		

Logging

You can record OMS activity in a log in XML or text format.

- The log tracks all new OMS requests for the session but does not include OMS requests that were already active before you requested a log.
- The current log file ends if you specify a new log file or if you deselect (uncheck) Log OMS activity.

To specify OMS logging:

▶ Click Logging in the Output Management System Control Panel.

Excluding Output Display from the Viewer

The Exclude from Viewer check box affects all output that is selected in the OMS request by suppressing the display of that output in the Viewer window. This process is often useful for production jobs that generate a lot of output and when you don't need the results in the form of a Viewer document (.spv file). You can also use this functionality to suppress the display of particular output objects that you simply never want to see, without routing any other output to some external file and format.

To suppress the display of certain output objects without routing other output to an external file:

▶ Create an OMS request that identifies the unwanted output.

▶ Select Exclude from Viewer.

▶ For the output destination, select File—but leave the File field blank.

▶ Click Add.

The selected output will be excluded from the Viewer while all other output will be displayed in the Viewer in the normal fashion.

Note: This setting has no effect on OMS output saved to external formats or files, including the Viewer SPV and SPW formats. It also has no effect on output saved to SPV format in a batch job executed with the Batch Facility (available with PASW Statistics Server).

Routing Output to PASW Statistics Data Files

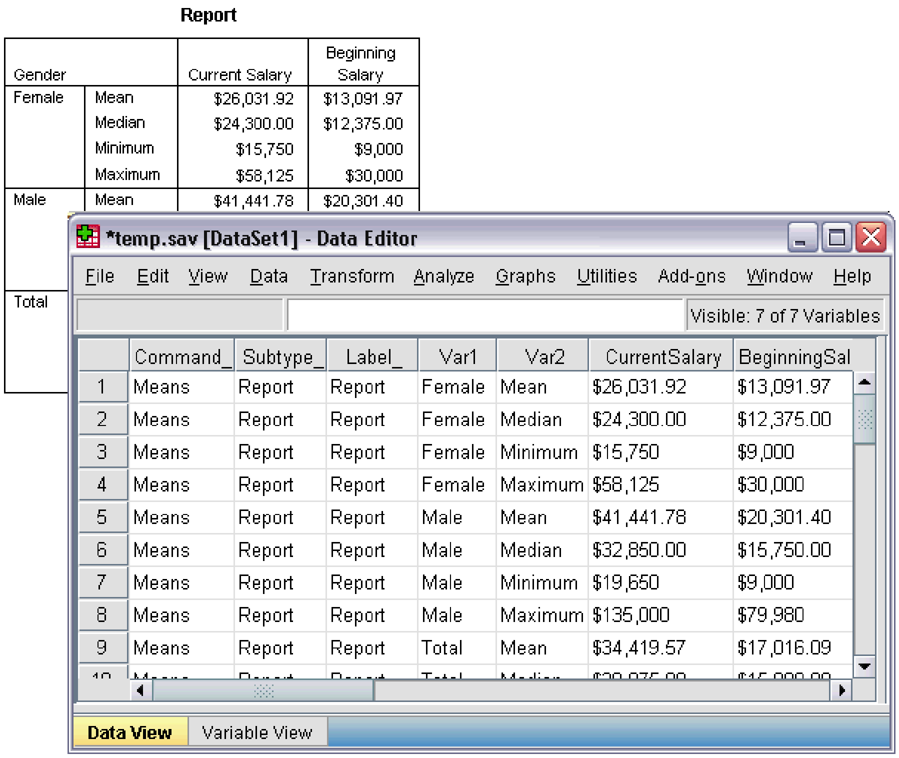
A data file in PASW Statistics format consists of variables in the columns and cases in the rows, which is essentially the format in which pivot tables are converted to data files:

- Columns in the table are variables in the data file. Valid variable names are constructed from the column labels.
- Row labels in the table become variables with generic variable names (*Var1*, *Var2*, *Var3*, and so on) in the data file. The values of these variables are the row labels in the table.
- Three table-identifier variables are automatically included in the data file: *Command_*, *Subtype_*, and *Label_*. All three are string variables. The first two variables correspond to the command and subtype identifiers. For more information, see the topic Command Identifiers and Table Subtypes on p. 357. *Label_* contains the table title text.
- Rows in the table become cases in the data file.

Example: Single Two-Dimensional Table

In the simplest case—a single two-dimensional table—the table columns become variables, and the rows become cases in the data file.

Figure 20-7
Single two-dimensional table



Report

Gender		Current Salary	Beginning Salary
Female	Mean	$26,031.92	$13,091.97
	Median	$24,300.00	$12,375.00
	Minimum	$15,750	$9,000
	Maximum	$58,125	$30,000
Male	Mean	$41,441.78	$20,301.40
Total			

*temp.sav [DataSet1] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Visible: 7 of 7 Variables
Data View
Variable View

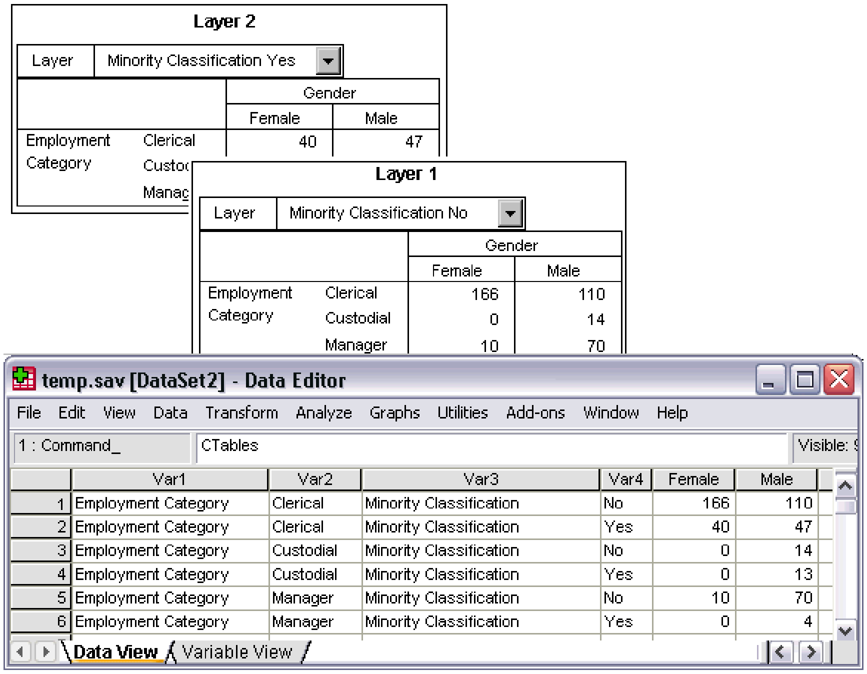
	Command_	Subtype_	Label_	Var1	Var2	CurrentSalary	BeginningSal
1	Means	Report	Report	Female	Mean	$26,031.92	$13,091.97
2	Means	Report	Report	Female	Median	$24,300.00	$12,375.00
3	Means	Report	Report	Female	Minimum	$15,750	$9,000
4	Means	Report	Report	Female	Maximum	$58,125	$30,000
5	Means	Report	Report	Male	Mean	$41,441.78	$20,301.40
6	Means	Report	Report	Male	Median	$32,850.00	$15,750.00
7	Means	Report	Report	Male	Minimum	$19,650	$9,000
8	Means	Report	Report	Male	Maximum	$135,000	$79,980
9	Means	Report	Report	Total	Mean	$34,419.57	$17,016.09

- The first three variables identify the source table by command, subtype, and label.
- The two elements that defined the rows in the table—values of the variable *Gender* and statistical measures—are assigned the generic variable names *Var1* and *Var2*. These variables are both string variables.
- The column labels from the table are used to create valid variable names. In this case, those variable names are based on the variable labels of the three scale variables that are summarized in the table. If the variables didn't have defined variable labels, or you chose to display variable names instead of variable labels as the column labels in the table, the variable names in the new data file would be the same as in the source data file.

Example: Tables with Layers

In addition to rows and columns, a table can also contain a third dimension: the layer dimension.

Figure 20-8
Table with layers



Layer 2

Layer: Minority Classification Yes

		Gender	
		Female	Male
Employment Category	Clerical	40	47
	Custoc		
	Manac		

Layer 1

Layer: Minority Classification No

		Gender	
		Female	Male
Employment Category	Clerical	166	110
	Custodial	0	14
	Manager	10	70

temp.sav [DataSet2] - Data Editor

1 : Command_ CTables

	Var1	Var2	Var3	Var4	Female	Male
1	Employment Category	Clerical	Minority Classification	No	166	110
2	Employment Category	Clerical	Minority Classification	Yes	40	47
3	Employment Category	Custodial	Minority Classification	No	0	14
4	Employment Category	Custodial	Minority Classification	Yes	0	13
5	Employment Category	Manager	Minority Classification	No	10	70
6	Employment Category	Manager	Minority Classification	Yes	0	4

- In the table, the variable labeled *Minority Classification* defines the layers. In the data file, two additional variables are created: one variable that identifies the layer element and one variable that identifies the categories of the layer element.
- As with the variables that are created from the row elements, the variables that are created from the layer elements are string variables with generic variable names (the prefix *Var* followed by a sequential number).

Data Files Created from Multiple Tables

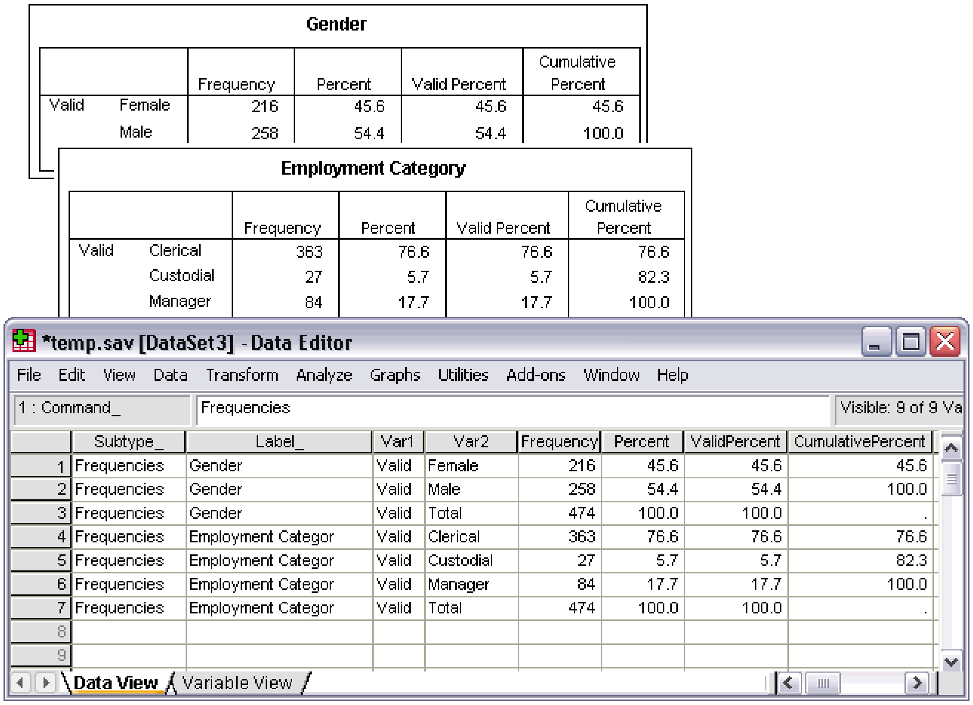
When multiple tables are routed to the same data file, each table is added to the data file in a fashion that is similar to merging data files by adding cases from one data file to another data file (Data menu, Merge Files, Add Cases).

- Each subsequent table will always add cases to the data file.
- If column labels in the tables differ, each table may also add variables to the data file, with missing values for cases from other tables that don't have an identically labeled column.

Example: Multiple Tables with the Same Column Labels

Multiple tables that contain the same column labels will typically produce the most immediately useful data files (data files that don't require additional manipulation). For example, two or more frequency tables from the Frequencies procedure will all have identical column labels.

Figure 20-9
Two tables with identical column labels



Gender

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Female	216	45.6	45.6	45.6
	Male	258	54.4	54.4	100.0

Employment Category

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Clerical	363	76.6	76.6	76.6
	Custodial	27	5.7	5.7	82.3
	Manager	84	17.7	17.7	100.0

1 : Command_ Frequencies

	Subtype_	Label_	Var1	Var2	Frequency	Percent	ValidPercent	CumulativePercent
1	Frequencies	Gender	Valid	Female	216	45.6	45.6	45.6
2	Frequencies	Gender	Valid	Male	258	54.4	54.4	100.0
3	Frequencies	Gender	Valid	Total	474	100.0	100.0	.
4	Frequencies	Employment Categor	Valid	Clerical	363	76.6	76.6	76.6
5	Frequencies	Employment Categor	Valid	Custodial	27	5.7	5.7	82.3
6	Frequencies	Employment Categor	Valid	Manager	84	17.7	17.7	100.0
7	Frequencies	Employment Categor	Valid	Total	474	100.0	100.0	.
8								
9								

- The second table contributes additional cases (rows) to the data file but contributes no new variables because the column labels are exactly the same, so there are no large patches of missing data.
- Although the values for *Command_* and *Subtype_* are the same, the *Label_* value identifies the source table for each group of cases because the two frequency tables have different titles.

Example: Multiple Tables with Different Column Labels

A new variable is created in the data file for each unique column label in the tables that are routed to the data file. This process results in blocks of missing values if the tables contain different column labels.

Figure 20-10
Two tables with different column labels

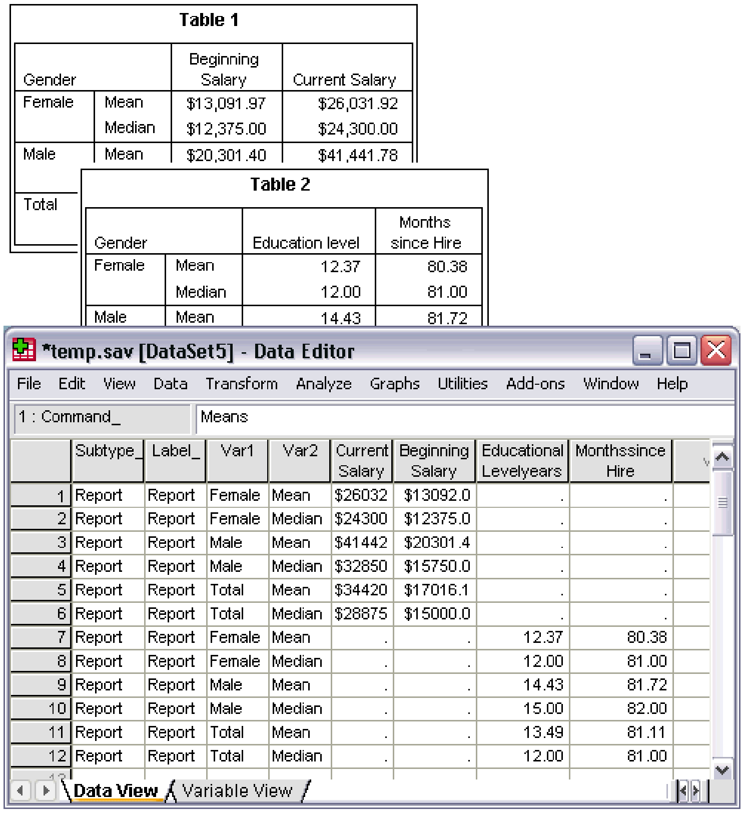


Table 1

Gender		Beginning Salary	Current Salary
Female	Mean	$13,091.97	$26,031.92
	Median	$12,375.00	$24,300.00
Male	Mean	$20,301.40	$41,441.78
Total			

Table 2

Gender		Education level	Months since Hire
Female	Mean	12.37	80.38
	Median	12.00	81.00
Male	Mean	14.43	81.72

*temp.sav [DataSet5] - Data Editor
File Edit View Data Transform Analyze Graphs Utilities Add-ons Window Help
Data View
Variable View

1 : Command_ Means

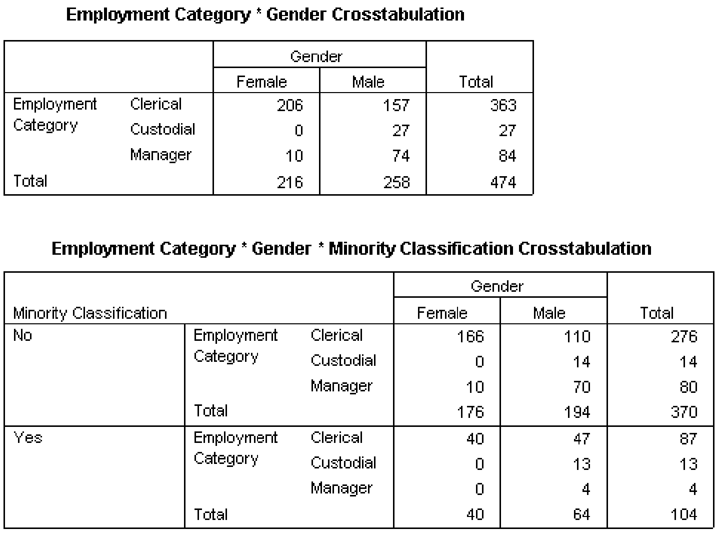
	Subtype_	Label_	Var1	Var2	Current Salary	Beginning Salary	Educational Levelyears	Monthssince Hire
1	Report	Report	Female	Mean	$26032	$13092.0	.	.
2	Report	Report	Female	Median	$24300	$12375.0	.	.
3	Report	Report	Male	Mean	$41442	$20301.4	.	.
4	Report	Report	Male	Median	$32850	$15750.0	.	.
5	Report	Report	Total	Mean	$34420	$17016.1	.	.
6	Report	Report	Total	Median	$28875	$15000.0	.	.
7	Report	Report	Female	Mean	.	.	12.37	80.38
8	Report	Report	Female	Median	.	.	12.00	81.00
9	Report	Report	Male	Mean	.	.	14.43	81.72
10	Report	Report	Male	Median	.	.	15.00	82.00
11	Report	Report	Total	Mean	.	.	13.49	81.11
12	Report	Report	Total	Median	.	.	12.00	81.00

- The first table has columns labeled *Beginning Salary* and *Current Salary*, which are not present in the second table, resulting in missing values for those variables for cases from the second table.
- Conversely, the second table has columns labeled *Education level* and *Months since Hire*, which are not present in the first table, resulting in missing values for those variables for cases from the first table.
- Mismatched variables like the variables in this example can occur even with tables of the same subtype. In this example, both tables are the same subtype.

Example: Data Files Not Created from Multiple Tables

If any tables do not have the same number of row elements as the other tables, no data file will be created. The number of rows doesn’t have to be the same; the number of row *elements* that become variables in the data file must be the same. For example, a two-variable crosstabulation and a three-variable crosstabulation contain different numbers of row elements, because the “layer” variable is actually nested within the row variable in the default three-variable crosstabulation display.

Figure 20-11
Tables with different numbers of row elements



Employment Category * Gender Crosstabulation

		Gender		Total
		Female	Male	
Employment Category	Clerical	206	157	363
	Custodial	0	27	27
	Manager	10	74	84
Total		216	258	474

Employment Category * Gender * Minority Classification Crosstabulation

			Gender		Total
Minority Classification			Female	Male	
No	Employment Category	Clerical	166	110	276
		Custodial	0	14	14
		Manager	10	70	80
	Total		176	194	370
Yes	Employment Category	Clerical	40	47	87
		Custodial	0	13	13
		Manager	0	4	4
	Total		40	64	104

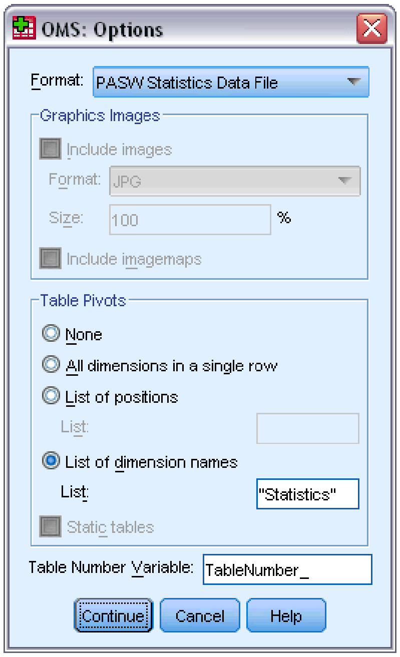
Controlling Column Elements to Control Variables in the Data File

In the Options dialog box of the Output Management System Control Panel, you can specify which dimension elements should be in the columns and therefore will be used to create variables in the generated data file. This process is equivalent to pivoting the table in the Viewer.

For example, the Frequencies procedure produces a descriptive statistics table with statistics in the rows, while the Descriptives procedure produces a descriptive statistics table with statistics in the columns. To include both table types in the same data file in a meaningful fashion, you need to change the column dimension for one of the table types.

Because both table types use the element name “Statistics” for the statistics dimension, we can put the statistics from the Frequencies statistics table in the columns simply by specifying “Statistics” (in quotation marks) in the list of dimension names in the OMS Options dialog box.

Figure 20-12
OMS Options dialog box



Format: PASW Statistics Data File

Graphics Images

☐ Include images

Format: JPG

Size: 100 %

☐ Include imagemaps

Table Pivots

○ None

○ All dimensions in a single row

○ List of positions

List:

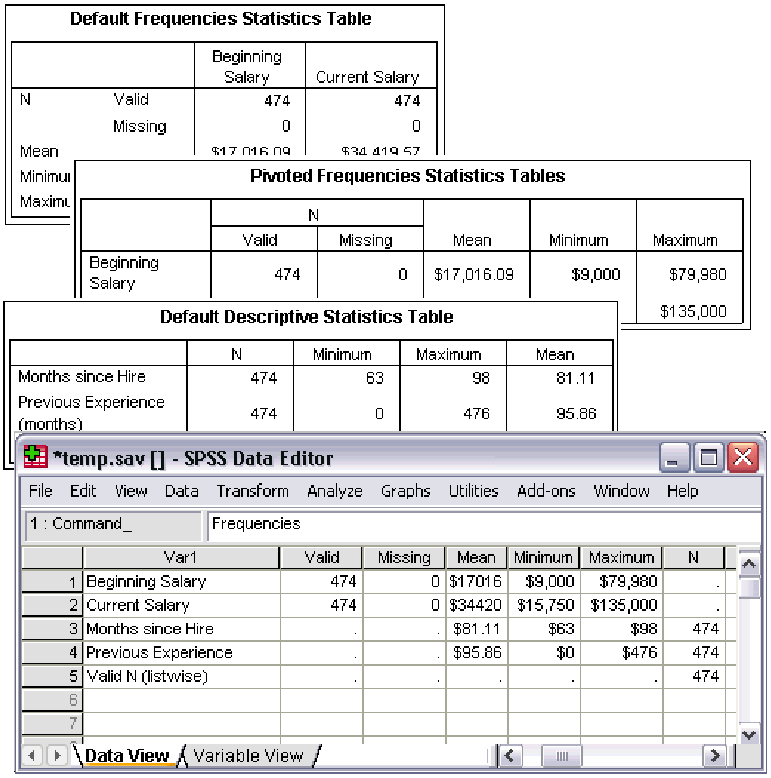
● List of dimension names

List: "Statistics"

☐ Static tables

Table Number Variable: TableNumber_

Figure 20-13
Combining different table types in a data file by pivoting dimension elements



Default Frequencies Statistics Table

		Beginning Salary	Current Salary
N	Valid	474	474
	Missing	0	0
Mean		$17,016.09	$34,419.57
Minimu			
Maximu			

Pivoted Frequencies Statistics Tables

	N		Mean	Minimum	Maximum
	Valid	Missing			
Beginning Salary	474	0	$17,016.09	$9,000	$79,980
					$135,000

Default Descriptive Statistics Table

	N	Minimum	Maximum	Mean
Months since Hire	474	63	98	81.11
Previous Experience (months)	474	0	476	95.86

1 : Command_ Frequencies

	Var1	Valid	Missing	Mean	Minimum	Maximum	N
1	Beginning Salary	474	0	$17016	$9,000	$79,980	.
2	Current Salary	474	0	$34420	$15,750	$135,000	.
3	Months since Hire	.	.	$81.11	$63	$98	474
4	Previous Experience	.	.	$95.86	$0	$476	474
5	Valid N (listwise)	.	.	.	.	.	474
6							
7							

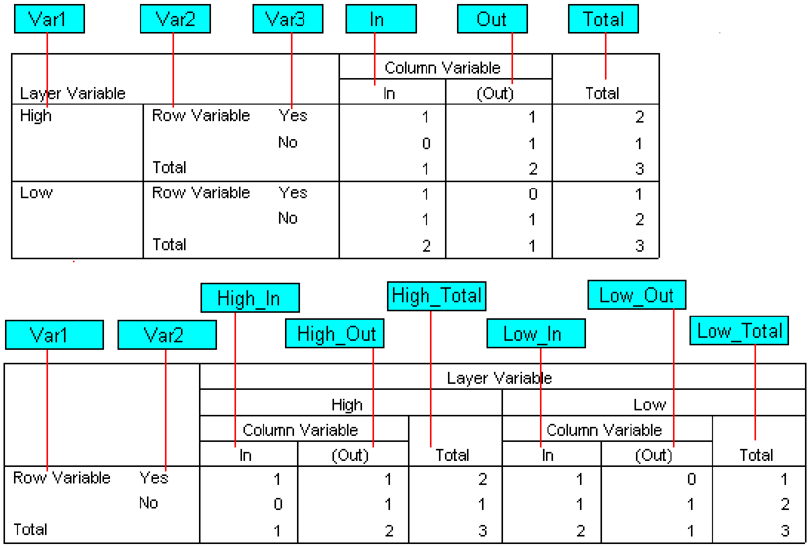
Some of the variables will have missing values, because the table structures still aren't exactly the same with statistics in the columns.

Variable Names in OMS-Generated Data Files

OMS constructs valid, unique variable names from column labels:

- Row and layer elements are assigned generic variable names—the prefix *Var* followed by a sequential number.
- Characters that aren't allowed in variable names (spaces, parentheses, etc.) are removed. For example, "This (Column) Label" would become a variable named *ThisColumnLabel*.
- If the label begins with a character that is allowed in variable names but not allowed as the first character (for example, a number), "@" is inserted as a prefix. For example, "2nd" would become a variable named *@2nd*.
- Underscores or periods at the end of labels are removed from the resulting variable names. The underscores at the end of the automatically generated variables *Command_*, *Subtype_*, and *Label_* are not removed.
- If more than one element is in the column dimension, variable names are constructed by combining category labels with underscores between category labels. Group labels are not included. For example, if *VarB* is nested under *VarA* in the columns, you would get variables like *CatA1_CatB1*, not *VarA_CatA1_VarB_CatB1*.

Figure 20-14
Variable names constructed from table elements



Layer Variable			Column Variable		
			In	(Out)	Total
High	Row Variable	Yes	1	1	2
		No	0	1	1
	Total		1	2	3
Low	Row Variable	Yes	1	0	1
		No	1	1	2
	Total		2	1	3

		Layer Variable					
		High			Low		
		Column Variable			Column Variable		
		In	(Out)	Total	In	(Out)	Total
Row Variable	Yes	1	1	2	1	0	1
	No	0	1	1	1	1	2
Total		1	2	3	2	1	3

OXML Table Structure

Output XML (OXML) is XML that conforms to the *spss-output* schema. For a detailed description of the schema, see the Output Schema section of the Help system.

- OMS command and subtype identifiers are used as values of the command and subType attributes in OXML. An example is as follows:

```
<command text="Frequencies" command="Frequencies"...>
 <pivotTable text="Gender" label="Gender" subType="Frequencies"...>
```

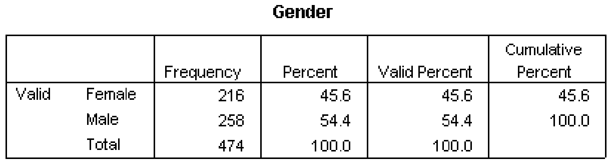
- OMS command and subType attribute values are not affected by output language or display settings for variable names/labels or values/value labels.
- XML is case sensitive. A subType attribute value of “frequencies” is *not* the same as a subType attribute value of “Frequencies.”
- All information that is displayed in a table is contained in attribute values in OXML. At the individual cell level, OXML consists of “empty” elements that contain attributes but no “content” other than the content that is contained in attribute values.
- Table structure in OXML is represented row by row; elements that represent columns are nested within the rows, and individual cells are nested within the column elements:

```
<pivotTable...>
 <dimension axis='row'...>
  <dimension axis='column'...>
   <category...>
    <cell text='...' number='...' decimals='...'/>
   </category>
   <category...>
    <cell text='...' number='...' decimals='...'/>
   </category>
  </dimension>
 </dimension>
   ...
 </pivotTable>
```

The preceding example is a simplified representation of the structure that shows the descendant/ancestor relationships of these elements. However, the example does not necessarily show the parent/child relationships, because there are typically intervening nested element levels.

The following figure shows a simple frequency table and the complete output XML representation of that table.

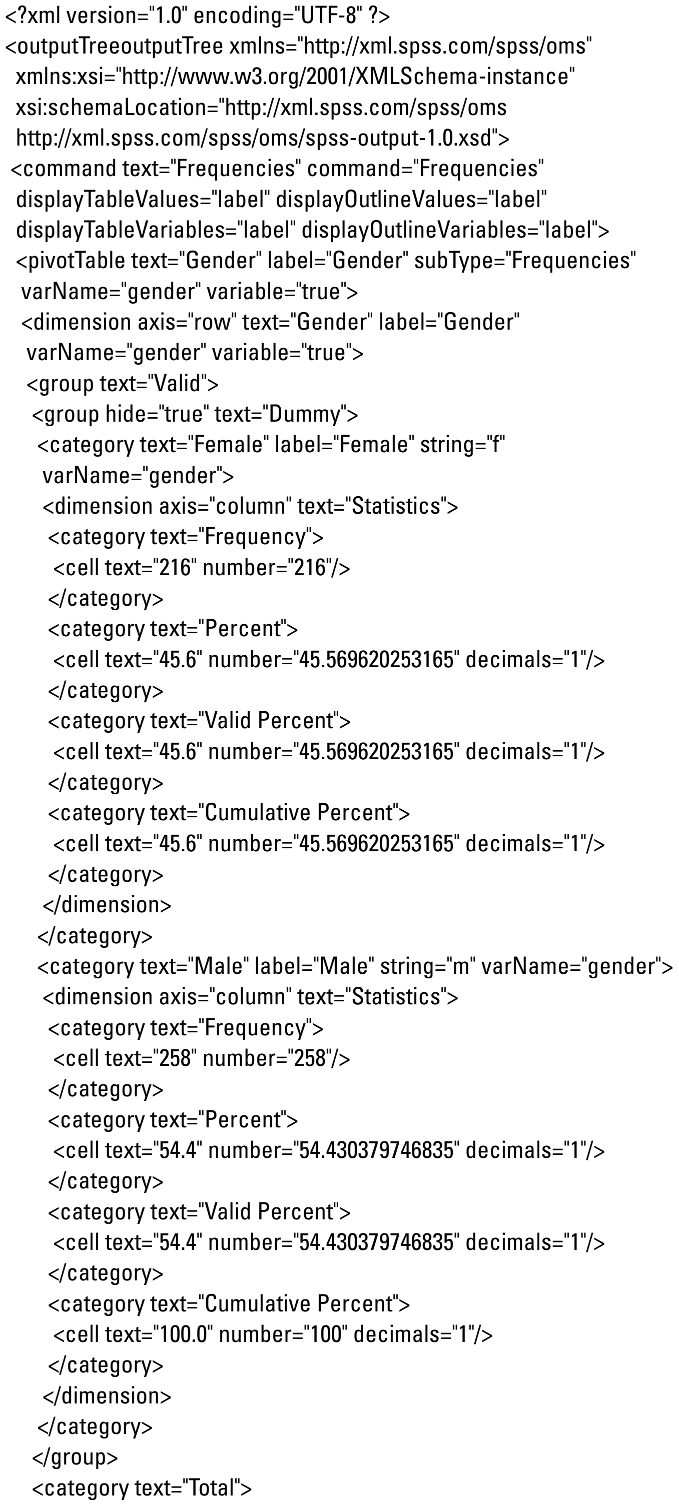
Figure 20-15
Simple frequency table



Gender

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Female	216	45.6	45.6	45.6
	Male	258	54.4	54.4	100.0
	Total	474	100.0	100.0	

Figure 20-16
Output XML for the simple frequency table



```
<?xml version="1.0" encoding="UTF-8" ?>
<outputTreeoutputTree xmlns="http://xml.spss.com/spss/oms"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xml.spss.com/spss/oms
  http://xml.spss.com/spss/oms/spss-output-1.0.xsd">
 <command text="Frequencies" command="Frequencies"
  displayTableValues="label" displayOutlineValues="label"
  displayTableVariables="label" displayOutlineVariables="label">
  <pivotTable text="Gender" label="Gender" subType="Frequencies"
   varName="gender" variable="true">
   <dimension axis="row" text="Gender" label="Gender"
    varName="gender" variable="true">
    <group text="Valid">
     <group hide="true" text="Dummy">
      <category text="Female" label="Female" string="f"
       varName="gender">
       <dimension axis="column" text="Statistics">
        <category text="Frequency">
         <cell text="216" number="216"/>
        </category>
        <category text="Percent">
         <cell text="45.6" number="45.569620253165" decimals="1"/>
        </category>
        <category text="Valid Percent">
         <cell text="45.6" number="45.569620253165" decimals="1"/>
        </category>
        <category text="Cumulative Percent">
         <cell text="45.6" number="45.569620253165" decimals="1"/>
        </category>
       </dimension>
      </category>
      <category text="Male" label="Male" string="m" varName="gender">
       <dimension axis="column" text="Statistics">
        <category text="Frequency">
         <cell text="258" number="258"/>
        </category>
        <category text="Percent">
         <cell text="54.4" number="54.430379746835" decimals="1"/>
        </category>
        <category text="Valid Percent">
         <cell text="54.4" number="54.430379746835" decimals="1"/>
        </category>
        <category text="Cumulative Percent">
         <cell text="100.0" number="100" decimals="1"/>
        </category>
       </dimension>
      </category>
     </group>
     <category text="Total">
```

```
      <dimension axis="column" text="Statistics">
       <category text="Frequency">
        <cell text="474" number="474"/>
       </category>
       <category text="Percent">
        <cell text="100.0" number="100" decimals="1"/>
       </category>
       <category text="Valid Percent">
        <cell text="100.0" number="100" decimals="1"/>
       </category>
      </dimension>
     </category>
    </group>
   </dimension>
  </pivotTable>
 </command>
</outputTree>
```

As you may notice, a simple, small table produces a substantial amount of XML. That's partly because the XML contains some information that is not readily apparent in the original table, some information that might not even be available in the original table, and a certain amount of redundancy.

- The table contents as they are (or would be) displayed in a pivot table in the Viewer are contained in text attributes. An example is as follows:

```
<command text="Frequencies" command="Frequencies"...>
```

- Text attributes can be affected by both output language and settings that affect the display of variable names/labels and values/value labels. In this example, the text attribute value will differ, depending on the output language, whereas the command attribute value remains the same, regardless of output language.
- Wherever variables or values of variables are used in row or column labels, the XML will contain a text attribute and one or more additional attribute values. An example is as follows:

```
<dimension axis="row" text="Gender" label="Gender" varName="gender">
    ...<category text="Female" label="Female" string="f" varName="gender">
```

- For a numeric variable, there would be a number attribute instead of a string attribute. The label attribute is present only if the variable or values have defined labels.
- The <cell> elements that contain cell values for numbers will contain the text attribute and one or more additional attribute values. An example is as follows:

```
<cell text="45.6" number="45.569620253165" decimals="1"/>
```

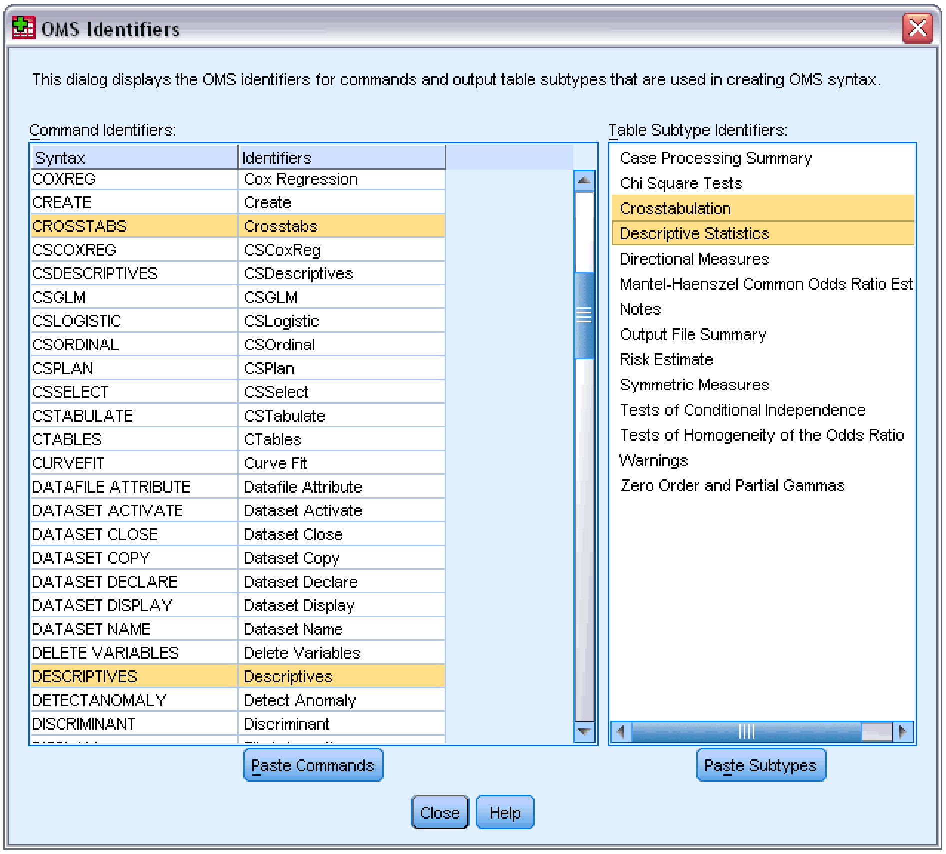
The number attribute is the actual, unrounded numeric value, and the decimals attribute indicates the number of decimal positions that are displayed in the table.

- Because columns are nested within rows, the category element that identifies each column is repeated for each row. For example, because the statistics are displayed in the columns, the element <category text="Frequency"> appears three times in the XML: once for the male row, once for the female row, and once for the total row.

OMS Identifiers

The OMS Identifiers dialog box is designed to assist you in writing OMS command syntax. You can use this dialog box to paste selected command and subtype identifiers into a command syntax window.

Figure 20-17
OMS Identifiers dialog box



This dialog displays the OMS identifiers for commands and output table subtypes that are used in creating OMS syntax.

Command Identifiers:

Syntax	Identifiers
COXREG	Cox Regression
CREATE	Create
CROSSTABS	Crosstabs
CSCOXREG	CSCoxReg
CSDESCRIPTIVES	CSDescriptives
CSGLM	CSGLM
CSLOGISTIC	CSLogistic
CSORDINAL	CSOrdinal
CSPLAN	CSPlan
CSSELECT	CSSelect
CSTABULATE	CSTabulate
CTABLES	CTables
CURVEFIT	Curve Fit
DATAFILE ATTRIBUTE	Datafile Attribute
DATASET ACTIVATE	Dataset Activate
DATASET CLOSE	Dataset Close
DATASET COPY	Dataset Copy
DATASET DECLARE	Dataset Declare
DATASET DISPLAY	Dataset Display
DATASET NAME	Dataset Name
DELETE VARIABLES	Delete Variables
DESCRIPTIVES	Descriptives
DETECTANOMALY	Detect Anomaly
DISCRIMINANT	Discriminant

Table Subtype Identifiers:

Case Processing Summary
Chi Square Tests
Crosstabulation
Descriptive Statistics
Directional Measures
Mantel-Haenszel Common Odds Ratio Est
Notes
Output File Summary
Risk Estimate
Symmetric Measures
Tests of Conditional Independence
Tests of Homogeneity of the Odds Ratio
Warnings
Zero Order and Partial Gammas

To Use the OMS Identifiers Dialog Box

▶ From the menus choose:

Utilities
  OMS Identifiers...

▶ Select one or more command or subtype identifiers. (Use Ctrl+click to select multiple identifiers in each list.)

▶ Click Paste Commands and/or Paste Subtypes.

- The list of available subtypes is based on the currently selected command(s). If multiple commands are selected, the list of available subtypes is the union of all subtypes that are available for any of the selected commands. If no commands are selected, all subtypes are listed.
- The identifiers are pasted into the designated command syntax window at the current cursor location. If there are no open command syntax windows, a new syntax window is automatically opened.
- Each command and/or subtype identifier is enclosed in quotation marks when pasted, because `OMS` command syntax requires these quotation marks.
- Identifier lists for the `COMMANDS` and `SUBTYPES` keywords must be enclosed in brackets, as in:

```
/IF COMMANDS=['Crosstabs' 'Descriptives']
    SUBTYPES=['Crosstabulation' 'Descriptive Statistics']
```

Copying OMS Identifiers from the Viewer Outline

You can copy and paste OMS command and subtype identifiers from the Viewer outline pane.

▶ In the outline pane, right-click the outline entry for the item.

▶ Choose Copy OMS Command Identifier or Copy OMS Table Subtype.

This method differs from the OMS Identifiers dialog box method in one respect: The copied identifier is not automatically pasted into a command syntax window. The identifier is simply copied to the clipboard, and you can then paste it anywhere you want. Because command and subtype identifier values are identical to the corresponding command and subtype attribute values in Output XML format (OXML), you might find this copy/paste method useful if you write XSLT transformations.

Copying OMS Labels

Instead of identifiers, you can copy labels for use with the `LABELS` keyword. Labels can be used to differentiate between multiple graphs or multiple tables of the same type in which the outline text reflects some attribute of the particular output object, such as the variable names or labels. There are, however, a number of factors that can affect the label text:

- If split-file processing is on, split-file group identification may be appended to the label.
- Labels that include information about variables or values are affected by the settings for the display of variable names/labels and values/value labels in the outline pane (Edit menu, Options, Output Labels tab).
- Labels are affected by the current output language setting (Edit menu, Options, General tab).

To copy OMS labels

- ▶ In the outline pane, right-click the outline entry for the item.
- ▶ Choose Copy OMS Label.

As with command and subtype identifiers, the labels must be in quotation marks, and the entire list must be enclosed in square brackets, as in:

```
/IF LABELS=['Employment Category' 'Education Level']
```

Chapter

21

Scripting Facility

The scripting facility allows you to automate tasks, including:

- Opening and saving data files.
- Exporting charts as graphic files in a variety of formats.
- Customizing output in the Viewer.

Scripting Languages

The available scripting languages depend on your platform. For Windows, the available scripting languages are Basic, which is installed with the Core system, and the Python programming language. For all other platforms, scripting is available with the Python programming language.

To enable scripting with the Python programming language, you must have Python and the PASW Statistics-Python Integration Plug-in installed. For information, see How to Get Integration Plug-ins, available from Core System>Frequently Asked Questions in the Help system.

Note: SPSS Inc. is not the owner or licensor of the Python software. Any user of Python must agree to the terms of the Python license agreement located on the Python Web site. SPSS Inc. does not make any statement about the quality of the Python program. SPSS Inc. fully disclaims all liability associated with your use of the Python program.

Default Script Language

The default script language determines the script editor that is launched when new scripts are created. It also specifies the default language whose executable will be used to run autoscripts. On Windows, the default script language is Basic. You can change the default language from the Scripts tab in the Options dialog. For more information, see the topic Script Options in Chapter 16 on p. 306.

Sample Scripts

A number of scripts are included with the software, in the *Samples* subdirectory of the directory where PASW Statistics is installed. You can use these scripts as they are or you can customize them to your needs.

To Create a New Script

- From the menus choose:

File
  New
    Script

The editor associated with the default script language opens.

To Run a Script

- From the menus choose:

Utilities
  Run Script...

- Select the script you want.
- Click Run.

Python scripts can be run in a number of ways, other than from Utilities>Run Script. For more information, see the topic Scripting with the Python Programming Language on p. 383.

To Edit a Script

- From the menus choose:

File
  Open
    Script...

- Select the script you want.
- Click Open.

The script is opened in the editor associated with the language in which the script is written.

Autoscripts

Autoscripts are scripts that run automatically when triggered by the creation of specific pieces of output from selected procedures. For example, you can use an autoscript to automatically remove the upper diagonal and highlight correlation coefficients below a certain significance whenever a Correlations table is produced by the Bivariate Correlations procedure.

Autoscripts can be specific to a given procedure and output type or apply to specific output types from different procedures. For example, you might have an autoscript that formats the ANOVA tables produced by One-Way ANOVA as well as ANOVA tables produced by other statistical procedures. On the other hand, Frequencies produces both a frequency table and a table of statistics, and you might choose to have a different autoscript for each.

Each output type for a given procedure can only be associated with a single autoscript. You can, however, create a base autoscript that is applied to all new Viewer items prior to the application of any autoscripts for specific output types. For more information, see the topic Script Options in Chapter 16 on p. 306.

The Scripts tab in the Options dialog box (accessed from the Edit menu) displays the autoscripts that have been configured on your system and allows you to set up new autoscripts or modify the settings for existing ones. Optionally, you can create and configure autoscripts for output items directly from the Viewer.

Events that Trigger Autoscripts

The following events can trigger autoscripts:

- Creation of a pivot table
- Creation of a Notes object
- Creation of a Warnings object

You can also use a script to trigger an autoscript indirectly. For example, you could write a script that invokes the Correlations procedure, which in turn triggers an autoscript registered to the resulting Correlations table.

Creating Autoscripts

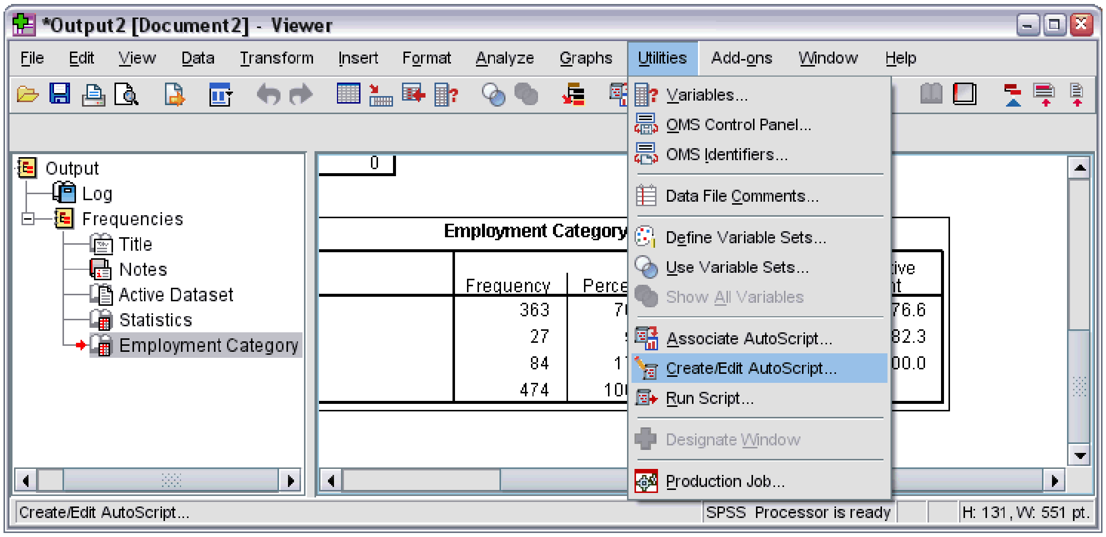
You can create an autoscript by starting with the output object that you want to serve as the trigger—for instance, a frequency table.

▶ In the Viewer, select the object that will trigger the autoscript.

▶ From the menus choose:

Utilities
  Create/Edit AutoScript...

Figure 21-1
Creating a new autoscript



If the selected object does not have an associated autoscript, an Open dialog prompts you for the location and name of a new script.

- ▶ Browse to the location where the new script will be stored, enter a file name and click Open. The editor for the default script language opens. You can change the default script language from the Scripts tab on the Options dialog. For more information, see the topic Script Options in Chapter 16 on p. 306.

- ▶ Type the code.

  For help with coding autoscripts in Python, see Getting Started with Autoscripts in Python on p. 386. For help with converting custom Sax Basic autoscripts used in pre-16.0 versions, see Compatibility with Versions Prior to 16.0 on p. 390.

  Note: By default, the executable associated with the default script language will be used to run the autoscript. You can change the executable from the Scripts tab in the Options dialog.

If the selected object is already associated with an autoscript, the script is opened in the script editor associated with the language in which the script is written.

Associating Existing Scripts with Viewer Objects

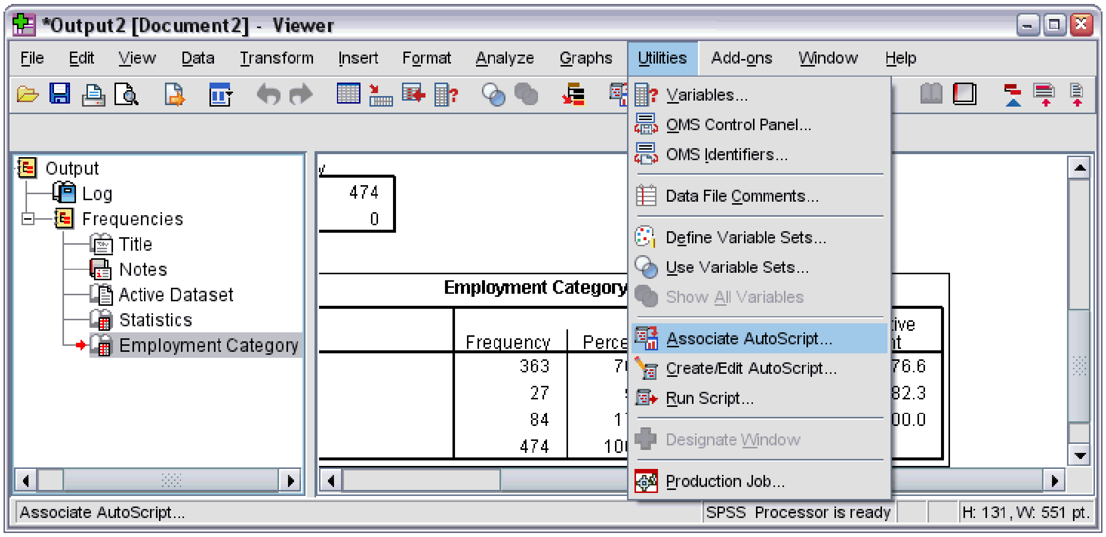
You can use existing scripts as autoscripts by associating them with a selected object in the Viewer—for instance, a frequency table.

- ▶ In the Viewer, select an object to associate with an autoscript (multiple Viewer objects can trigger the same autoscript, but each object can only be associated with a single autoscript).

- ▶ From the menus choose:

  Utilities
    Associate AutoScript...

Figure 21-2
Associating an autoscript



If the selected object does not have an associated autoscript, the Select Autoscript dialog opens.

▶ Browse for the script you want and select it.

▶ Click Apply.

If the selected object is already associated with an autoscript, you are prompted to verify that you want to change the association. Clicking OK opens the Select Autoscript dialog.

Optionally, you can configure an existing script as an autoscript from the Scripts tab in the Options dialog box. The autoscript can be applied to a selected set of output types or specified as the base autoscript that is applied to all new Viewer items. For more information, see the topic Script Options in Chapter 16 on p. 306.

Scripting with the Python Programming Language

PASW Statistics provides two separate interfaces for programming with the Python language on Windows, Linux, and Mac OS. Use of these interfaces requires the PASW Statistics-Python Integration Plug-in, which is provided with your PASW Statistics product. For information, see How to Get Integration Plug-ins, available from Core System>Frequently Asked Questions in the Help system. For help getting started with the Python programming language, see the Python tutorial, available at *http://docs.python.org/tut/tut.html*.

Python Scripts

Python scripts make use of the interface exposed by the Python `SpssClient` module. They operate on user interface and output objects and can also run command syntax. For instance, you would use a Python script to customize a pivot table.

- Python scripts are run from Utilities>Run Script, from the Python editor launched from PASW Statistics (accessed from File>Open>Script), or from an external Python process, such as a Python IDE or the Python interpreter.
- Python scripts can be run as autoscripts.
- Python scripts run on the machine where the PASW Statistics client is running.

Complete documentation of the PASW Statistics classes and methods available for Python scripts can be found in the PASW Statistics Scripting Guide, available under Python Integration Plug-in in the Help system.

Python Programs

Python programs make use of the interface exposed by the Python `spss` module. They operate on the PASW Statistics processor and are used to control the flow of a command syntax job, read from and write to the active dataset, create new datasets, and create custom procedures that generate their own pivot table output.

- Python programs are run from command syntax within `BEGIN PROGRAM-END PROGRAM` blocks, or from an external Python process, such as a Python IDE or the Python interpreter.
- Python programs cannot be run as autoscripts.
- In distributed analysis mode (available with PASW Statistics Server), Python programs execute on the computer where PASW Statistics Server is running.

More information about Python programs, including complete documentation of the PASW Statistics functions and classes available for them, can be found in the documentation for the PASW Statistics-Python Integration Package, available under Python Integration Plug-in in the Help system.

Running Python Scripts and Python programs

Both Python scripts and Python programs can be run from within PASW Statistics or from an external Python process, such as a Python IDE or the Python interpreter.

Python Scripts

Python Script Run from PASW Statistics. You can run a Python script from Utilities>Run Script or from the Python script editor which is launched when opening a Python file (.py) from File>Open>Script. Scripts run from the Python editor that is launched from PASW Statistics operate on the PASW Statistics client that launched the editor. This allows you to debug your Python code from a Python editor.

Python Script Run from an External Python Process. You can run a Python script from any external Python process, such as a Python IDE that is not launched from PASW Statistics, or the Python interpreter. The script will attempt to connect to an existing PASW Statistics client. If more than one client is found, a connection is made to the most recently launched one. If an existing client is not found, the Python script starts up a new instance of the PASW Statistics client. By default, the Data Editor and Viewer are invisible for the new client. You can choose to make them visible or work in invisible mode with datasets and output documents.

Python Programs

Python Program Run from Command Syntax. You can run a Python program by embedding Python code within a BEGIN PROGRAM-END PROGRAM block in command syntax. The command syntax can be run from the PASW Statistics client or from the PASW Statistics Batch Facility—a separate executable provided with PASW Statistics Server.

Python Program Run from an External Python Process. You can run a Python program from any external Python process, such as a Python IDE or the Python interpreter. In this mode, the Python program starts up a new instance of the PASW Statistics processor without an associated instance of the PASW Statistics client. You can use this mode to debug your Python programs using the Python IDE of your choice.

Invoking Python Scripts from Python Programs and Vice Versa

Python Script Run from Python Program. You can run a Python script from a Python program by importing the Python module containing the script and calling the function in the module that implements the script. You can also call Python script methods directly from within a Python program. For more information, see the topic Running Python Scripts from Python Programs on p. 387. These features are not available in distributed mode or when running a Python program from an external Python process. To programmatically invoke a Python script in distributed mode, use the SCRIPT command.

Python Autoscript Triggered from Python Program. A Python script specified as an autoscript will be triggered when a Python program executes the procedure containing the output item associated with the autoscript. For example, you associate an autoscript with the Descriptive Statistics table generated by the Descriptives procedure. You then run a Python program that executes the Descriptives procedure. The Python autoscript will be executed.

Python Program Run from Python Script. Python scripts can run command syntax, which means they can run command syntax containing Python programs.

Limitations and Warnings

- Running a Python program from the Python editor launched by PASW Statistics will start up a new instance of the PASW Statistics processor and will not interact with the instance of PASW Statistics that launched the editor.
- Python programs are not intended to be run from Utilities>Run Script.
- Python programs cannot be run as autoscripts.
- The interfaces exposed by the spss module cannot be used in a Python script.

Getting Started with Python Scripts

The basic structure of a Python script is:

```
import SpssClient
SpssClient.StartClient()
<Python language statements>
SpssClient.StopClient()
```

- The `import SpssClient` statement imports the Python module containing the PASW Statistics classes and methods available in the Python scripting interface.
- `SpssClient.StartClient()` provides a connection to the associated PASW Statistics client, enabling the script to retrieve information from the client and to perform operations on objects managed by the client, such as pivot tables. Whether the script connects to an existing client or starts up a new client depends on how the script was invoked. For more information, see the topic Running Python Scripts and Python programs on p. 384.
- `SpssClient.StopClient()` terminates the connection to the PASW Statistics client and should be called at the completion of each Python script.

  Note: If you're running a Python script from an external Python process that starts up a new client, call `SpssClient.Exit()` before `SpssClient.StopClient()`.

Example

This script accesses the designated output document and sets each of the pivot tables as selected.

```
import SpssClient
SpssClient.StartClient()

OutputDoc = SpssClient.GetDesignatedOutputDoc()
OutputItems = OutputDoc.GetOutputItems()

for index in range(OutputItems.Size()):
   OutputItem = OutputItems.GetItemAt(index)
   if OutputItem.GetType() == SpssClient.OutputItemType.PIVOT:
        OutputItem.SetSelected(True)
SpssClient.StopClient()
```

Target for Standard output

The Python `print` statement writes output to Python's standard output. When you run a Python script from Utilities>Run Script, Python's standard output is directed to a log item in the PASW Statistics Viewer.

Getting Started with Autoscripts in Python

Autoscripts typically require a reference to the object that triggered the script. They may also require a reference to the associated output document and possibly the index of the output item in the output document. These values are obtained from the `SpssScriptContext` object, as shown in this example of an autoscript that transposes the rows and columns of a pivot table.

```
import SpssClient
SpssClient.StartClient()

SpssScriptContext = SpssClient.GetScriptContext()
SpssOutputItem = SpssScriptContext.GetOutputItem()
SpssPivotTable = SpssOutputItem.GetSpecificType()
SpssPivotMgr = SpssPivotTable.PivotManager()
SpssPivotMgr.TransposeRowsWithColumns()

SpssClient.StopClient()
```

- SpssClient.GetScriptContext returns an SpssScriptContext object that provides values for use by the autoscript.
- The GetOutputItem method of the SpssScriptContext object returns the output item that triggered the current autoscript—in this example, the pivot table whose rows and columns are to be transposed.

Although not used in this example, the GetOutputDoc method of the SpssScriptContext object returns the associated output document, and the GetOutputItemIndex method returns the index (in the associated output document) of the output item that triggered the autoscript.

Detecting When a Script is Run as an Autoscript

Using the GetScriptContext method, you can detect when a script is being run as an autoscript. This allows you to code a script so that it functions in either context (autoscript or not). This trivial script illustrates the approach.

```
import SpssClient
SpssClient.StartClient()

SpssScriptContext = SpssClient.GetScriptContext()
if SpssScriptContext == None:
   print "I'm not an autoscript"
else:
   print "I'm an autoscript"

SpssClient.StopClient()
```

- When a script is not run as an autoscript, the GetScriptContext method will return a value of None.
- Given the if-else logic in this example, you would include your autoscript-specific code in the else clause. Any code that is not to be run in the context of an autoscript would be included in the if clause. Of course you can also include code that is to be run in either context.

Running Python Scripts from Python Programs

You can run Python scripts from Python programs and you can call Python script methods from within a Python program. This allows you to write Python programs that operate on user interface and output objects.

Note: This feature is only available when running a Python program from the PASW Statistics client—within a BEGIN PROGRAM-END PROGRAM block in command syntax or within an extension command. It is not available in distributed mode or when running a Python program from an external Python process. To programmatically invoke a Python script in distributed mode, use the SCRIPT command.

Example: Calling a Python Script from a Python Program

This example shows a Python program that creates a custom pivot table and calls a Python script to make the column labels of the table bold.

```
BEGIN PROGRAM.
import spss, MakeColsBold
spss.StartProcedure("Demo")
table = spss.BasePivotTable("Sample Table","OMS subtype")
table.SimplePivotTable(rowlabels = ["1","2"],
                       collabels = ["A","B"],
                       cells = ["1A","1B","2A","2B"])
spss.EndProcedure()
MakeColsBold.Run("Sample Table")
END PROGRAM.
```

- Python programs use the interface exposed by the Python `spss` module, so the first line of the program contains an `import` statement for that module. The Python script is assumed to be contained in a Python module named `MakeColsBold`, so the `import` statement also includes that module.
- The code from `spss.StartProcedure` to `spss.EndProcedure` creates a pivot table titled “Sample Table”.
- `MakeColsBold.Run("Sample Table")` calls the `Run` function in the `MakeColsBold` module and passes the value “Sample Table” as the argument. The `Run` function implements the Python script to make the column labels of the specified table bold.

The content of the `MakeColsBold` module is as follows:

```
import SpssClient

def Run(tableName):
   SpssClient.StartClient()
   OutputDoc = SpssClient.GetDesignatedOutputDoc()
   OutputItems = OutputDoc.GetOutputItems()
   for index in range(OutputItems.Size()):
      OutputItem = OutputItems.GetItemAt(index)
      if OutputItem.GetType() == SpssClient.OutputItemType.PIVOT \
        and OutputItem.GetDescription() == tableName:
         PivotTable = OutputItem.GetSpecificType()
         ColumnLabels = PivotTable.ColumnLabelArray()
         for i in range(ColumnLabels.GetNumColumns()):
            ColumnLabels.SelectLabelAt(1,i)
         PivotTable.SetTextStyle(SpssClient.SpssTextStyleTypes.SpssTSBold)
   SpssClient.StopClient()
```

- The `import SpssClient` statement is needed to access the classes and methods available in the Python scripting interface.
- The module contains a single function named `Run`, which implements the script. It takes a single argument that specifies the name of the table to modify. There is nothing special about the name *Run* and the module is not limited to a single function. You can create a module that contains many functions, each of which implements a different script.
- The `Run` function calls `SpssClient.StartClient()` to provide a connection to the associated PASW Statistics client and `SpssClient.StopClient()` to terminate the connection at the completion of the script.

Example: Calling Python Scripting Methods Directly from a Python Program

This example shows a Python program that creates a custom pivot table and makes direct calls to Python scripting methods to make the title of the table italic.

```
BEGIN PROGRAM.
import spss, SpssClient
spss.StartProcedure("Demo")
table = spss.BasePivotTable("Sample Table","OMS subtype")
table.SimplePivotTable(cells = ["A","B","C","D"])
spss.EndProcedure()

SpssClient.StartClient()
OutputDoc = SpssClient.GetDesignatedOutputDoc()
OutputItems = OutputDoc.GetOutputItems()
OutputItem = OutputItems.GetItemAt(OutputItems.Size()-1)
PivotTable = OutputItem.GetSpecificType()
PivotTable.SelectTitle()
PivotTable.SetTextStyle(SpssClient.SpssTextStyleTypes.SpssTSItalic)
SpssClient.StopClient()
END PROGRAM.
```

- The `import spss, SpssClient` statement provides access to the classes and methods available for Python programs (`spss`) as well as those for Python scripts (`SpssClient`).
- The code from `spss.StartProcedure` to `spss.EndProcedure` is the Python program code that creates the pivot table.
- The code from `SpssClient.StartClient()` to `SpssClient.StopClient()` is the Python script code that makes the title italic.

Script Editor for the Python Programming Language

For the Python programming language, the default editor is IDLE, which is provided with Python. IDLE provides an integrated development environment (IDE) with a limited set of features. Many IDE's are available for the Python programming language. For instance, on Windows you may choose to use the freely available PythonWin IDE.

To change the script editor for the Python programming language:

▶ Open the file *clientscriptingcfg.ini*, located in the directory where PASW Statistics is installed. *Note*: *clientscriptingcfg.ini* must be edited with a UTF-16 aware editor, such as SciTE on Windows or the TextEdit application on Mac.

▶ Under the section labeled [Python], change the value of EDITOR_PATH to point to the executable for the desired editor.

▶ In that same section, change the value of EDITOR_ARGS to handle any arguments that need to be passed to the editor. If no arguments are required, remove any existing values.

Scripting in Basic

Scripting in Basic is available on Windows only and is installed with the Core system. Extensive online help for scripting in Basic is available from the PASW Statistics Basic Script Editor. The editor can be accessed from File>New>Script when the default script language (set from the Scripts tab on the Options dialog) is set to Basic (the system default on Windows). It is also accessed from File>Open>Script, by choosing Basic (wwd;sbs) in the Files of type list.

Compatibility with Versions Prior to 16.0

Obsolete Methods and Properties

A number of automation methods and properties are obsolete for version 16.0 and above. In terms of general features, this includes all objects associated with interactive graphs, the Draft Document object, and methods and properties associated with maps. For additional information, see “Release Notes for Version 16.0” in the help system provided with the PASW Statistics Basic Script Editor. The PASW Statistics-specific help is accessed from Help>PASW Statistics Objects Help, in the script editor.

Global Procedures

Prior to version 16.0, the scripting facility included a global procedures file. For version 16.0 and above the scripting facility does not use a global procedures file, although the pre-16.0 version of *Global.sbs* (renamed *Global.wwd*) is installed for backwards compatibility.

To migrate a pre-16.0 version of a script that called functions in the global procedures file, add the statement `'#Uses "<install dir>\Samples\Global.wwd"` to the declarations section of the script, where `<install dir>` is the directory where PASW Statistics is installed. `'#Uses` is a special comment recognized by the Basic script processor. If you’re not sure if a script uses the global procedures file, you should add the `'#Uses` statement. You can also use `'$Include:` instead of `'#Uses`.

Legacy Autoscripts

Prior to version 16.0, the scripting facility included a single autoscript file containing all autoscripts. For version 16.0 and above there is no single autoscript file. Each autoscript is now stored in a separate file and can be applied to one or more output items, in contrast to pre-16.0 versions where each autoscript was specific to a particular output item.

Some of the autoscripts installed with pre-16.0 versions are available as a set of separate script files located in the *Samples* subdirectory of the directory where PASW Statistics is installed. They are identified by a filename ending in *Autoscript*, with a file type of *wwd*. By default, they are not associated with any output items. The association is done from the Scripts tab of the Options dialog. For more information, see the topic Script Options in Chapter 16 on p. 306.

Any custom autoscripts used in pre-16.0 versions must be manually converted and associated with one or more output items, from the Scripts tab of the Options dialog. The conversion process involves the following steps:

▶ Extract the subroutine specifying the autoscript from the legacy *Autoscript.sbs* file and save it as a new file with an extension of *wwd* or *sbs*. The name of the file is arbitrary.

▶ Change the name of the subroutine to `Main` and remove the parameter specification, keeping track of which parameters are required by the script, such as a pivot table that triggers the autoscript.

▶ Use the `scriptContext` object (always available) to get the values required by the autoscript, such as the output item that triggered the autoscript.

▶ From the Scripts tab of the Options dialog, associate the script file with the desired output object.

To illustrate the converted code, consider the autoscript *Descriptives_Table_DescriptiveStatistics_Create* from the legacy *Autoscript.sbs* file.

```
Sub Descriptives_Table_DescriptiveStatistics_Create(objPivotTable As Object,objOutputDoc As
'Autoscript
'Trigger Event: DescriptiveStatistics Table Creation after running Descriptives procedure.

'Purpose: Swaps the Rows and Columns in the currently active pivot table.
'Assumptions: Selected Pivot Table is already activated.
'Effects: Swaps the Rows and Columns in the output
'Inputs: Pivot Table, OutputDoc, Item Index

    Dim objPivotManager As ISpssPivotMgr
    Set objPivotManager=objPivotTable.PivotManager
    objPivotManager.TransposeRowsWithColumns

End Sub
```

Following is the converted script:

```
Sub Main

'Purpose: Swaps the Rows and Columns in the currently active pivot table.
'Effects: Swaps the Rows and Columns in the output

    Dim objOutputItem As ISpssItem
    Dim objPivotTable as PivotTable
    Set objOutputItem = scriptContext.GetOutputItem()
    Set objPivotTable = objOutputItem.ActivateTable

    Dim objPivotManager As ISpssPivotMgr
    Set objPivotManager = objPivotTable.PivotManager
    objPivotManager.TransposeRowsWithColumns
    objOutputItem.Deactivate
End Sub
```

- Notice that nothing in the converted script indicates which object the script is to be applied to. The association between an output item and an autoscript is set from the Scripts tab of the Options dialog and maintained across sessions.
- `scriptContext.GetOutputItem` gets the output item (an `ISpssItem` object) that triggered the autoscript.
- The object returned by `scriptContext.GetOutputItem` is not activated. If your script requires an activated object, you'll need to activate it, as done in this example with the `ActivateTable` method. When you're finished with any table manipulations, call the `Deactivate` method.

For version 16.0, there is no distinction between scripts that are run as autoscripts and scripts that aren't run as autoscripts. Any script, appropriately coded, can be used in either context. For more information, see the topic The scriptContext Object on p. 392.

Note: To trigger a script from the application creation event, see Startup Scripts.

Script Editor

For version 16.0 and above the script editor for Basic no longer supports the following pre-16.0 features:

- The Script, Analyze, Graph, Utilities, and Add-Ons menus.
- The ability to paste command syntax into a script window.

The PASW Statistics Basic Script Editor is a standalone application that is launched from within PASW Statistics via File>New>Script, File>Open>Script, or Utilities>Create/Edit AutoScript (from a Viewer window). It allows you to run scripts against the instance of PASW Statistics from which it was launched. Once opened, the editor will remain open after exiting PASW Statistics, but scripts that use PASW Statistics objects will no longer run.

File Types

For version 16.0 and above, the scripting facility will continue to support running and editing scripts with a file type of *sbs*. By default, new Basic scripts created with the PASW Statistics Basic Script Editor have a file type of *wwd*.

Using External COM Clients

For version 16.0 and above, the program identifier for instantiating PASW Statistics from an external COM client is `SPSS.Application16`. Application objects should be declared as spsswinLib.Application16. For example:

```
Dim objSpssApp As spsswinLib.Application16
Set objSpssApp=CreateObject("SPSS.Application16")
```

To connect to a running instance of the PASW Statistics client from an external COM client, use:

```
Dim objSpssApp As spsswinLib.Application16
Set objSpssApp=GetObject("","SPSS.Application16")
```

If more than one client is running, `GetObject` will connect to the most recently launched one.

Note: For post-16.0 versions, the identifier is still `Application16`.

The scriptContext Object

Detecting When a Script is Run as an Autoscript

Using the `scriptContext` object, you can detect when a script is being run as an autoscript. This allows you to code a script so that it functions in either context (autoscript or not). This trivial script illustrates the approach.

```
Sub Main
    If scriptContext Is Nothing Then
     MsgBox "I'm not an autoscript"
    Else
     MsgBox "I'm an autoscript"
    End If
End Sub
```

- When a script is not run as an autoscript, the `scriptContext` object will have a value of `Nothing`.
- Given the `If-Else` logic in this example, you would include your autoscript-specific code in the `Else` clause. Any code that is not to be run in the context of an autoscript would be included in the `If` clause. Of course you can also include code that is to be run in either context.

Getting Values Required by Autoscripts

The `scriptContext` object provides access to values required by an autoscript, such as the output item that triggered the current autoscript.

- The `scriptContext.GetOutputItem` method returns the output item (an `ISpssItem` object) that triggered the current autoscript.
- The `scriptContext.GetOutputDoc` method returns the output document (an `ISpssOutputDoc` object) associated with the current autoscript.
- The `scriptContext.GetOutputItemIndex` method returns the index, in the associated output document, of the output item that triggered the current autoscript.

Note: The object returned by `scriptContext.GetOutputItem` is not activated. If your script requires an activated object, you'll need to activate it—for instance, with the `ActivateTable` method. When you're finished with any manipulations, call the `Deactivate` method.

Startup Scripts

You can create a script that runs at the start of each session and a separate script that runs each time you switch servers. For Windows you can have versions of these scripts in both Python and Basic. For all other platforms the scripts can only be in Python.

- The startup script must be named *StartClient_.py* for Python or *StartClient_.wwd* for Basic.
- The script that runs when switching servers must be named *StartServer_.py* for Python or *StartServer_.wwd* for Basic.
- The scripts must be located in the *scripts* directory of the installation directory—located at the root of the installation directory for Windows and Linux, and under the *Contents* directory in the application bundle for Mac.
- On Windows, if the *scripts* directory contains both a Python and a Basic version of *StartClient_* or *StartServer_* then both versions are executed. The order of execution is the Python version followed by the Basic version.
- If your system is configured to start up in distributed mode, then at the start of each session any *StartClient_* scripts are run followed by any *StartServer_* scripts. *Note*: The *StartServer_* scripts also run each time you switch servers, but the *StartClient_* scripts only run at the start of a session.

Appendix

A

TABLES and IGRAPH Command Syntax Converter

If you have command syntax files that contain TABLES syntax that you want to convert to CTABLES syntax and/or IGRAPH syntax that you want to convert to GGRAPH syntax, a simple utility program is provided to help you get started with the conversion process. There are, however, significant functionality differences between TABLES and CTABLES and between IGRAPH and GGRAPH. It is likely that you will find that the utility program cannot convert some of your TABLES and IGRAPH syntax jobs or may generate CTABLES and GGRAPH syntax that produces tables and graphs that do not closely resemble the originals produced by the TABLES and IGRAPH commands. For most tables, you can edit the converted syntax to produce a table closely resembling the original.

The utility program is designed to:

- Create a new syntax file from an existing syntax file. The original syntax file is not altered.
- Convert only TABLES and IGRAPH commands in the syntax file. Other commands in the file are not altered.
- Retain the original TABLES and IGRAPH syntax in commented form.
- Identify the beginning and end of each conversion block with comments.
- Identify TABLES and IGRAPH syntax commands that could not be converted.
- Convert command syntax files that follow either interactive or production mode syntax rules.

This utility cannot convert commands that contain errors. The following other limitations also apply.

TABLES Limitations

The utility program may convert TABLES commands incorrectly under some circumstances, including TABLES commands that contain:

- Parenthesized variable names with the initial letters “sta” or “lab” in the TABLES subcommand if the variable is parenthesized by itself—for example, var1 by (statvar) by (labvar). These will be interpreted as the (STATISTICS) and (LABELS) keywords.
- SORT subcommands that use the abbreviations A or D to indicate ascending or descending sort order. These will be interpreted as variable names.

The utility program cannot convert TABLES commands that contain:

- Syntax errors.

- `OBSERVATION` subcommands that refer to a range of variables using the `TO` keyword (for example, `var01 TO var05`).
- String literals broken into segments separated by plus signs (for example, `TITLE "My" + "Title"`).
- Macro calls that, in the absence of macro expansion, would be invalid `TABLES` syntax. Since the converter does not expand the macro calls, it treats them as if they were simply part of the standard `TABLES` syntax.

The utility program will not convert `TABLES` commands contained in macros. All macros are unaffected by the conversion process.

IGRAPH Limitations

`IGRAPH` changed significantly in release 16. Because of these changes, some subcommands and keywords in `IGRAPH` syntax created before that release may not be honored. See the `IGRAPH` section in the *Command Syntax Reference* for the complete revision history.

The conversion utility program may generate additional syntax that it stores in the `INLINETEMPLATE` keyword within the `GGRAPH` syntax. This keyword is created only by the conversion program. Its syntax is not intended to be user-editable.

Using the Conversion Utility Program

The conversion utility program, *SyntaxConverter.exe*, can be found in the installation directory. It is designed to run from a command prompt. The general form of the command is:

```
syntaxconverter.exe [path]/inputfilename.sps [path]/outputfilename.sps
```

You must run this command from the installation directory.

If any directory names contain spaces, enclose the entire path and filename in quotation marks, as in:

```
syntaxconverter.exe /myfiles/oldfile.sps "/new files/newfile.sps"
```

Interactive versus Production Mode Command Syntax Rules

The conversion utility program can convert command files that use interactive or production mode syntax rules.

Interactive. The interactive syntax rules are:

- Each command begins on a new line.
- Each command ends with a period (.).

Production mode. The Production Facility and commands in files accessed via the `INCLUDE` command in a different command file use production mode syntax rules:

- Each command must begin in the first column of a new line.
- Continuation lines must be indented at least one space.
- The period at the end of the command is optional.

If your command files use production mode syntax rules and don't contain periods at the end of each command, you need to include the command line switch `-b` (or `/b`) when you run *SyntaxConverter.exe*, as in:

```
syntaxconverter.exe -b /myfiles/oldfile.sps /myfiles/newfile.sps
```

SyntaxConverter Script (Windows Only)

On Windows, you can also run the syntax converter with the script *SyntaxConverter.wwd*, located in the *Samples* directory of the installation directory.

▶ From the menus choose:

Utilities
  Run Script...

▶ Navigate to the *Samples* directory and select *SyntaxConverter.wwd*.

This will open a simple dialog box where you can specify the names and locations of the old and new command syntax files.

Index