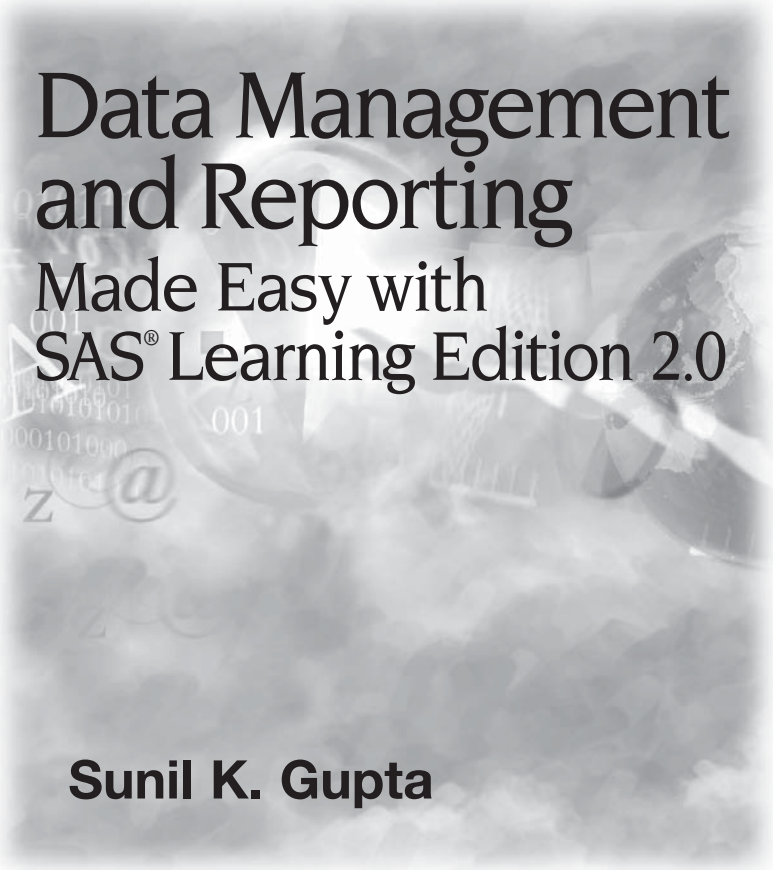
Data Management and Reporting

Made Easy with SAS® Learning Edition 2.0

Sunil K. Gupta

sas.

SAS Publishing



Data Management and Reporting

Made Easy with SAS® Learning Edition 2.0

Sunil K. Gupta

The Power to Know®

The correct bibliographic citation for this manual is as follows: Gupta, Sunil K. 2006. *Data Management and Reporting Made Easy with SAS® Learning Edition 2.0.* Cary, NC: SAS Institute Inc.

Data Management and Reporting Made Easy with SAS® Learning Edition 2.0

ISBN-13: 978-1-59047-577-5
ISBN-10: 1-59047-577-1

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, May 2006

SAS Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/pubs** or call 1-800-727-3228.

Contents

Acknowledgments

As I come to the end of writing this book, I want to thank my family, friends, and associates. Specifically, I want to thank my wife, Bindiya, for letting me work those long, long days; and my daughters, Aarti and Anupama, for always being excited about the book. I want to thank Stephenie Joyner and Julie Platt at SAS Press for giving me the opportunity to write this book and for keeping the book focused. I also want to thank Mike Boyd, who copyedited the book and Candy Farrell, who was the production specialist for the book. I want to thank Bernard Penney for his assistance in making this book useful as a practical guide for all new SAS users. I want to give a special thanks to Beth Hardin for her dedicated attention to all of the technical details and Jeanette Drisco for her business knowledge expertise in the retail industry. Finally, I want to thank my peer technical review team members, Dan Hempy, Parul Bhargava, Kirk Paul Lafler, Alex Dmitrienko, Mo Khan, and Deborah Testa for sharing their suggestions and experience.

Overview

Using This Book

This book shows how you can quickly and easily prepare your data to create simple-to-complex reports and analyses using SAS Learning Edition. The book's numerous examples demonstrate how a point-and-click approach generates and executes all the SAS code required to perform tasks. Once the data are selected or prepared in SAS Learning Edition, you can perform tasks for data management, analysis, and reporting without any programming time or effort.

Non-programmers: You can access the data management and analytical power of SAS. In addition, SAS Learning Edition simplifies a variety of frequently performed tasks; for example, enabling you to create not only PDF files but also HTML files that can be published to the Web by clicking one check box option. A key benefit of using SAS Learning Edition is that no programming is required.

Programmers: You have a tool for quickly developing prototype reports or getting assistance with the syntax of SAS procedures through a visual interface without writing any SAS code. In addition, you can include existing SAS code in your project without having to recreate it in SAS Learning Edition.

Each chapter provides step-by-step instructions and basic information on SAS and SAS Learning Edition concepts and tasks. All the analyses and reports generated in the examples use the queries created from the sample Sales data set. You can find information about the Sales data set in Chapter 2 and in the appendix.

A business model is used to help explain the types of analysis performed in a typical business environment. Once the business model is introduced in Chapter 2, you are ready to start using the point-and-click interface to create queries and tables in Chapter 3, and visualize the sales data in Chapter 4. Once you have learned about the tasks and gained a basic understanding of SAS programming, you can modify and create your own SAS code to create custom tables or graphs in Chapter 5.

Chapter 1 introduces the SAS Learning Edition basics. This chapter explains the SAS Learning Edition Online Tutorial and the types of tasks for performing data management, analysis, and reporting. The chapter also includes a review of data sets and variable types.

Chapter 2 introduces the business model used by all the examples in the book and describes the Sales data set. With SAS Learning Edition, you can answer the who, what, and when components of your business questions. To answer typical business questions, you will want to use the decision-tree tables to help identify the appropriate task to run for the identified query, table, or graph. You can then find the instructions for these tasks in chapters 3, 4, and 5.

Chapter 3 shows how to prepare your data and to produce reports for distribution. The chapter also illustrates how to create complex queries and tables that answer business-related questions. Once a table is created, you can modify the task to analyze a different viewpoint of the information.

Chapter 4 shows how to easily create complex charts and graphs, including pie charts, horizontal bar charts, and vertical bar charts. Graphs generated from SAS Learning Edition can be used in Microsoft Word or PowerPoint.

Chapter 5 discusses alternative methods for creating reports and graphs. With some SAS programming knowledge, you can modify or write your own SAS code to create your output. This chapter shows how to modify the SAS code that SAS Learning Edition generates to create custom tables and graphs.

For ease of reference, the following conventions are used in this book:

Name or Key SAS Term	Example
Format name	MARKETV.
SAS code in text	LIBNAME, value district
Text that you are to enter	Enter **`Sales`** in the field.
Variable names	PRODUCT variable

Introduction to SAS Learning Edition

The SAS Learning Edition is a personal learning version of the world's leading business intelligence and analytical software. SAS skills are in demand and hands-on experience is vital. The SAS Learning Edition is an excellent tool to discover the full power of SAS. Not only can you view SAS data sets, you can also create publication-quality reports and graphs.

SAS Learning Edition makes it easy to query data, filter reports, select variables, and control the grouping and sorting of tables. Once you activate a data set and select a task, all you need to do is drag and drop the variables into their roles and select the options you want for each task. SAS Learning Edition requires very little training to get started.

SAS Learning Edition uses the SAS Enterprise Guide point-and-click interface—the same interface used in SAS®9—to perform tasks. In addition, the SAS Program Editor is available for experienced SAS users who want to modify code generated by SAS Learning Edition. You'll see examples of both interfaces in this book, but most examples illustrate how to perform tasks using SAS Enterprise Guide windows.

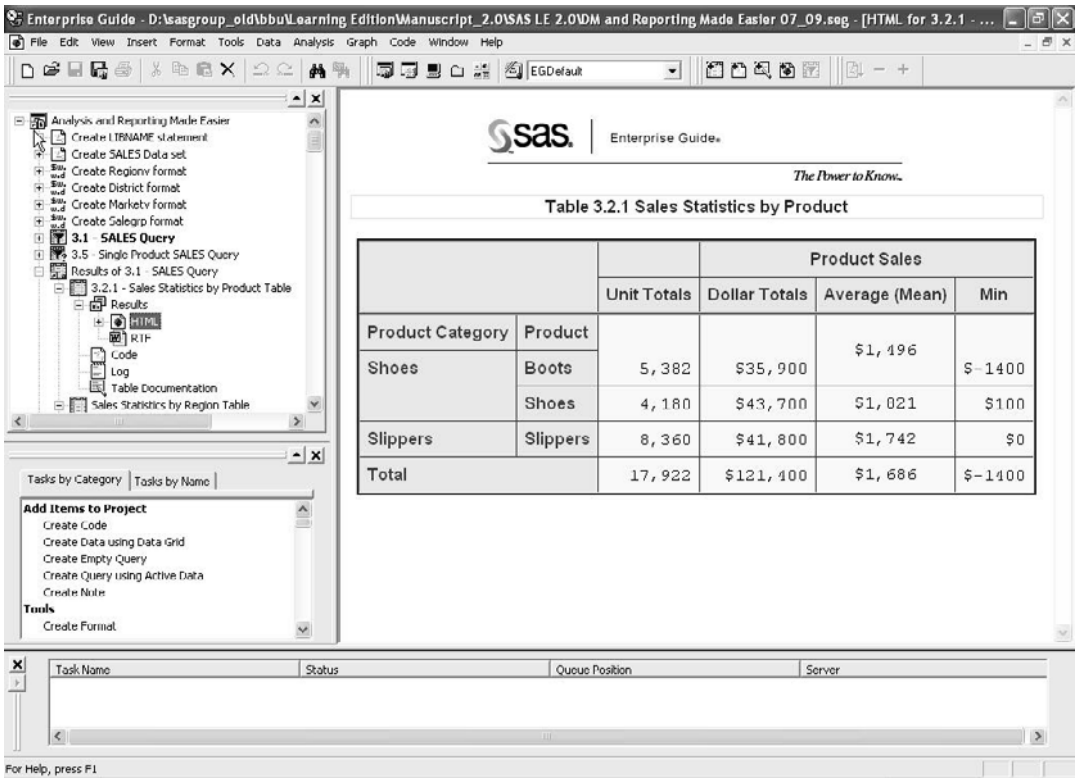
All SAS Learning Edition tasks require you to first create or open a project. The project organizes and stores all tasks, data, code, and results. Programmers will appreciate that the SAS code and log are also saved with the project. This enables programmers to continue developing SAS code just like in any other programming environment or operating system.

Projects created from SAS Learning Edition have the same file name extension as SAS Enterprise Guide. SAS Learning Edition 2.0 uses SAS Enterprise Guide 2.0.

The benefits of using SAS Learning Edition include being able to work directly with SAS data sets as well as create complex reports and apply statistical analysis. As a business analyst, you can take advantage of SAS Learning Edition to answer key critical sales-related questions.

As shown in the following figure, the SAS Learning Edition interface has four windows: Project, Task List, Workspace, and Task Status. Each window can be re-arranged and resized for better viewing. In addition to these windows, the toolbar provides an alternative method for selecting tasks.

SAS Learning Edition 2.0 Interface



Desktop or Window	Description
Project window	Displays the data sets, code, tasks, and results of the active project as a hierarchy of icons.
Task List window	Displays a list of all the available tasks. You can select tasks by category or by the SAS procedure name.
Task Status window	Indicates the task name along with the status of the code or the task it is processing.
Toolbar	Accesses the menu to select tasks, and provide other options for software operations such as opening and saving a project.
Workspace window	Space to view your data sets or results from the selected task. For programmers, this is also the space to edit your code and view the log file.

In the following chapters, you will learn more about these windows and the tasks you can perform in each. As you become more familiar with SAS Learning Edition, you will be able to manage your data efficiently and produce effective reports and graphs.

Chapter 1

IN THIS CHAPTER

Using the Online Tutorial and Help

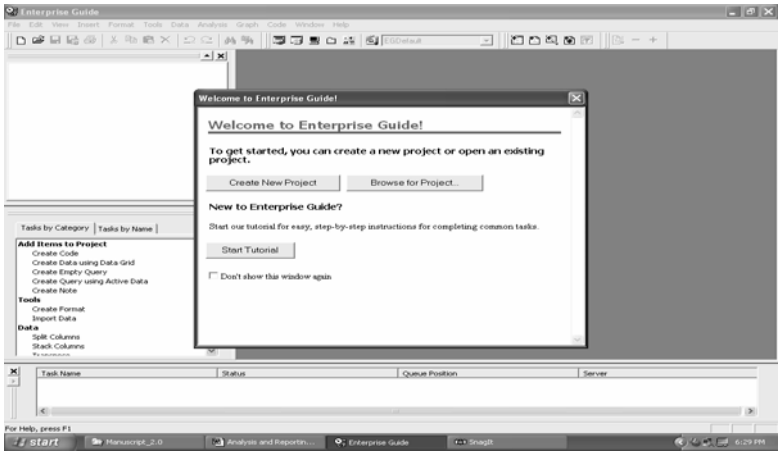
Performing Typical Tasks Using SAS Learning Edition

Understanding SAS Data Sets and Variables

Creating Publication-Quality Reports

SAS Learning Edition: The Basics

This chapter provides an introduction to some of the primary windows in SAS Learning Edition. In addition, it describes the basic tasks you'll perform using SAS Learning Edition and provides a brief overview of SAS data sets and variables.

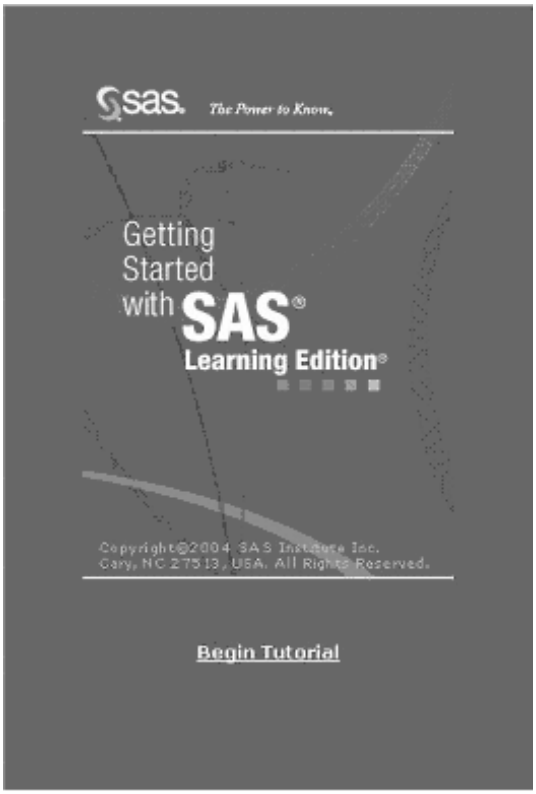


Using the Online Tutorial and Help

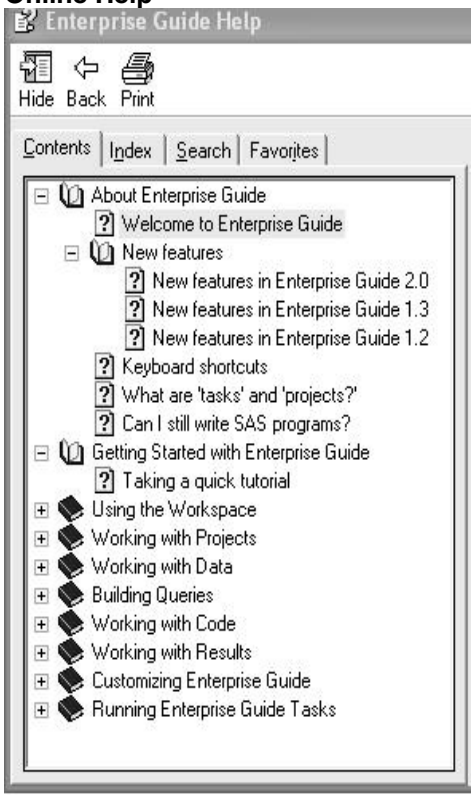
If you are new to SAS Learning Edition, you can get started quickly by going through the online tutorial. At any time, select the Help option from within SAS Learning Edition's main toolbar to access the online tutorial. The examples in the tutorial serve as an excellent introduction to the SAS Enterprise Guide tasks. For example, you can learn how to set up a project, create a basic report, and modify data for reports.

Getting assistance within SAS Learning Edition is easy because you can get help on each task and you can select the Help option from the main toolbar. In addition, you can get context-sensitive Help topics with the F1 key. Because each keyword is hyperlinked within the Help options, you can drill down for more detailed information.

Online Tutorial



Online Help



Performing Typical Tasks Using SAS Learning Edition

Whether you are in an academic or a business environment, when you want to make decisions based on data, you typically need to perform four general categories of tasks related to your data:

- access
- manage
- analyze
- present

Table 1.1 lists examples of tasks in these categories.

Table 1.1 Examples of Tasks, by Category

Category	Example Task
Data Access	View Sales data set and Sales Representative Excel file.
Data Management	Query and display sales for the past month.
Data Analysis	Summarize total sales by product.
Data Presentation	Create a pie chart of percent sales in PDF.

Because SAS/GRAPH maps and SAS/MDDB Server are not included in SAS Learning Edition, you will not be able to work with multidimensional databases (MDDBs) or run Graph Map Chart tasks in SAS Enterprise Guide.

SAS Learning Edition makes it easy for you to perform these tasks and get the results you need. Before you begin, you will need to get your data into a form that SAS Learning Edition can use. The following section describes SAS data sets and what they contain.

Understanding SAS Data Sets and Variables

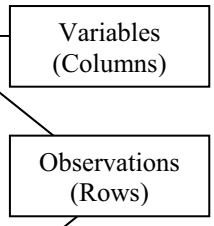
Different software packages access data in different forms. SAS Learning Edition accesses SAS data sets and their variables rather than raw data files to perform tasks.

A SAS data set is a file storing a collection of data. A SAS data set is referenced by a name such as Sales. If your data are stored in multiple SAS data sets, you can merge or join them to combine variables into a single data set. This is important to remember because tasks can access only a single data set. (See Chapter 2 for more details on tasks.) Also, SAS Learning Edition is designed for learning purposes rather than for production applications, so there is a limit of processing only the first 1,000 observations in your data set.

The contents in a SAS data set can be visualized as a table of information organized in rows and columns (like a Microsoft Excel spreadsheet). SAS uses the term *variables* to refer to columns and *observations* to refer to rows. In Figure 1.1, PRODUCT is the first variable (column) and **`Shoes`** is the first value in the first observation (row).

Figure 1.1 Sales Data Set (Partial)

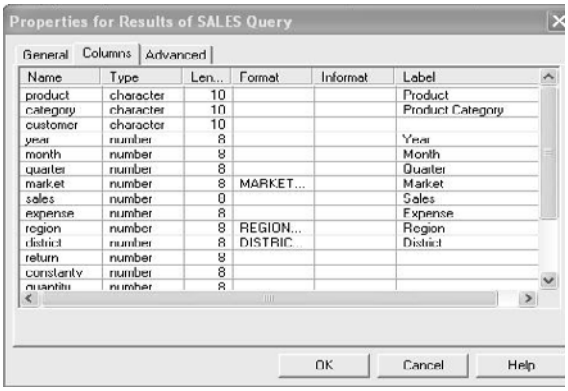
	product	category	customer	year	month	quarter	market	sales
1	Shoes	Shoes	Acme	2001	1	1	1	300
2	Boots	Shoes	Acme	2001	1	1	1	2200
3	Slippers	Slippers	Acme	2001	1	1	1	900
4	Shoes	Shoes	Acme	2001	2	1	1	100
5	Boots	Shoes	Acme	2001	2	1	1	1400
6	Slippers	Slippers	Acme	2001	2	1	1	0
7	Shoes	Shoes	Acme	2001	3	1	1	600
8	Boots	Shoes	Acme	2001	3	1	1	0
9	Slippers	Slippers	Acme	2001	3	1	1	1400
10	Shoes	Shoes	Acme	2001	4	2	1	2600



When referring to specific data values in SAS, you refer to the variables in the data set. Each variable has a set of properties associated with it: name, type (either character or numeric), length, format, informat, and label attributes. It is important to know the variable type since some tasks require numeric variables. Variables that uniquely identify records are called key variables. Every data set should have key variables. Common key variables should be specified when joining SAS data sets. This ensures that records are correctly linked together so that information between the two files is not mismatched. The example in Chapter 3 shows how easy it is to join files.

To obtain information on variable properties, as shown in Figure 1.2, select the data set and then right-click to select properties. (For programmers, the results are similar to running the CONTENTS procedure.)

Figure 1.2 Sales Data Set—Variable Properties



In SAS Learning Edition, variable types are represented as symbols. SAS Learning Edition displays the icon when assigning variable roles within tasks. In Figure 1.1, product, category, and customer are character variables because they contain alphanumeric data while the other variables (year, month, quarter, market, and sales) are numeric variables.

Often you need to display data values in a more meaningful representation to make reports and analyses easier to read. SAS Learning Edition uses *formats* to do this. Examples of common formats include DOLLARw.d, used to add a dollar sign ($) and decimal places to numbers when they are displayed and COMMAw.d, used to add commas to numbers when they are displayed. In addition to all the predefined formats available from SAS, you can create your own formats to display data values.

Both character and numeric variables can have formats as variable properties. For example, the MARKET variable uses the MARKETV. format to display **`Catalog`** for the data value of **`1`** and to display **`Retail`** for the data value of **`2`**, and so on. Note that format names end with a period. In addition, you can use formats to group a range of continuous values to obtain frequency of continuous variables. See Chapter 3 for information on formats and how to create the SALESGRP variable and format.

This table is an example of output from using the MARKETV. format with the Market value.

Formatted Value	Data Value
Catalog	1
Retail	2
Wholesale	3
Other	4

Depending on the interactive dialog box that is displayed when selecting tasks, the required variable type for each type of task role is indicated by the colored icons. This ensures that, for example, only numeric variables are used for analysis. The table below contains data value examples for each variable type.

Variable Type	Example Data Value	Example Task
Character	Shoes	Categorical Variable
Numeric	300	Analysis Variable
Numeric (Date)	01JAN60	Categorical Variable

A brief explanation of each variable type and purpose is provided below:

Character Variables

Character variables contain alphanumeric data that is case sensitive and can have embedded blanks. Make sure that you have the correct spelling of the text when performing queries containing character variables. For example: **`Shoes`** is not the same as **`shoes`**. In addition, embedded blanks can exist in variables, for example **`South East`** has a blank between South and East. Note that although character variables can store numeric values, this is not recommended because analysis can be performed only on numeric variables.

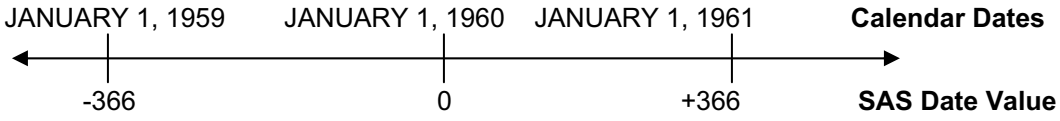
Numeric Variables

Numeric variables contain numeric values that can be used in calculations. Numeric data can be numbers, currency, date, or time data. It is best to store all sales and volume data in numeric variables. SAS Learning Edition performs analysis only on numeric variables.

Numeric (Date) Variables

Date variables are numeric variables. SAS handles dates by converting dates to SAS date values. As seen in Figure 1.3, a SAS date value is the number of days from January 1, 1960, to the given date. Formats are used to display dates in a readable form, e.g., WORDDATE12. to display dates as SEP 12, 1990. Because the Sales data set does not have a DATE variable, you can create it from the MONTH and YEAR variables, if needed. See Chapter 3 for information on creating a DATE variable.

Figure 1.3 SAS Date Value



When using dates within queries, you need to specify dates as constants. Date constants must be assigned in the format 'DDMMMYY'D or 'DDMMMYYYY'D, where the text is enclosed in single or double quotes, and followed with the letter 'D'.

Any mathematical operation can be applied to SAS date values to perform date calculations. The two examples below are used to create a date constant and apply a date condition.

Purpose	Example	Date Interger Value
Create Date Constant	`dob = '01JAN1960'D;`	0
Apply Date Condition	`where dob lt '01JAN61'D;`	366

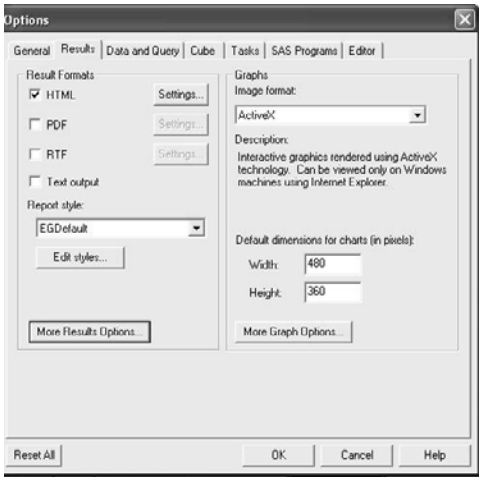
For more information about SAS data sets and variables, see the *SAS Language Guide*.

Creating Publication-Quality Reports

SAS Learning Edition uses the SAS Output Delivery System (ODS) to create publication-quality output files. This facilitates the distribution of information without manually transferring results to Microsoft Word or Microsoft Excel. Any of the analysis, tables, or graphs generated can be saved as one of the following formats: Microsoft Excel, HTML, PDF, RTF, SAS List, or SAS data set. HTML files generated from SAS Learning Edition can be shared and viewed once published on the web. You can create HTML files without having to know how to program in HTML.

SAS Learning Edition allows you to create more than one file type with a single task. As seen in Figure 1.4, select `Tools` ► `Options` from the toolbar and then the Results tab. There you can click on PDF, RTF or Text output as result formats. By default, the results are saved as HTML files. This default setting can be changed to another file type, if needed.

Figure 1.4 Options for Results Formats



In addition to these file types, you can export your data to Microsoft Word or Microsoft Excel. Once your data set is selected, you can create a Microsoft Word or Microsoft Excel file by selecting `File` ▶ `Send To` from the toolbar. Figure 1.5 shows the Sales data set saved as an Excel file.

Figure 1.5 Sales Data Set as an Excel File (Partial)

	A	B	C	D	E	F	G	H	I	J	K
1	product	category	customer	year	month	quarter	market	sales	expense	region	district
2	Shoes	Shoes	Acme	2001	1	1	Catalog	$300.00	240	Northern	NE
3	Boots	Shoes	Acme	2001	1	1	Catalog	$2,200.00	1540	Northern	NE
4	Slippers	Slippers	Acme	2001	1	1	Catalog	$900.00	540	Northern	NE
5	Shoes	Shoes	Acme	2001	2	1	Catalog	$100.00	80	Northern	NE
6	Boots	Shoes	Acme	2001	2	1	Catalog	$1,400.00	980	Northern	NE
7	Slippers	Slippers	Acme	2001	2	1	Catalog	$0.00	0	Northern	NE
8	Shoes	Shoes	Acme	2001	3	1	Catalog	$600.00	480	Northern	NE
9	Boots	Shoes	Acme	2001	3	1	Catalog	$0.00	0	Northern	NE
10	Slippers	Slippers	Acme	2001	3	1	Catalog	$1,400.00	840	Northern	NE

For more information on the Output Delivery System (ODS), see the References section for a list of SAS papers on ODS, and visit the SAS ODS web site.

Chapter 2

IN THIS CHAPTER

The Business Model and Project Setup; Selecting Analysis and Reporting Tasks

In this chapter, you will learn how to understand your business model by answering a set of typical questions. Based on the answers, you can decide which analysis tasks you need to perform and how to report the results.

Understanding the Business Model

All of the examples in this book are based on the sales analysis of a hypothetical shoe business. This section discusses a typical business model where a product such as shoes is exchanged for credit in a transaction that is recorded. All of the SAS Enterprise Guide tasks described in this book use this business model. The Sales data set and the Microsoft Excel file Repnames.xls contain all of the variables used in the examples.

Typically, companies that are in business to sell products or services need to collect similar data over time, as represented in the example Sales data set. Using this Sales data and SAS Learning Edition, a business can better understand current sales trends in order to meet customer demands. This is useful to evaluate the financial health of the business, manage money, and make effective critical business decisions. For example: When sales start to increase, you want to be sure that manufacturing and inventory have the capacity to handle the demand.

SAS Learning Edition and the Business Model

Understanding the business model is the first step in specifying the sales requests of an organization. SAS Learning Edition can be used to generate complex reports to help address the market-analysis requirements of an organization. SAS Learning Edition software has the tools for building the business by monitoring product sales. It can help access past performance to enable more intelligent planning. By establishing measurable marketing objectives, the strength and the weakness of the business can be evaluated and compared with the competition.

A Basic Example

As a Business Analyst for a shoe manufacturer, you are responsible for monitoring the sales of your company's products (shoes, boots, slippers) by all customers (Acme, BigX, TwoFeet) through all distribution channels (Catalog, Retail, Wholesale). Because you have access to the sales data, you have internal requests as well as requests from outside departments for this information. You need a software tool to quickly answer their questions without requiring the skills of a SAS programmer. In addition, because many requests are similar, you realize that these tasks can be standardized to improve efficiency.

Defining and Using Business Questions

Before using SAS Learning Edition to analyze and report on your data, it is helpful to define the questions that you want to ask about your data. The basic question that you want to answer contains these elements**:** Who, What, and When.

The three components of the ***Who*** did ***What,*** and ***When*** question in the typical core business question can be broken down into several options:

- ***Who*** questions focus on the *viewpoint* of the data.
- ***What*** questions address the *details* of the sales transaction.
- ***When*** questions explore the *time period* of the sales transaction.

Let's explore in detail the options for each question.

Question	Options
Who?	• Customer Viewpoint • Product Viewpoint • Organization Viewpoint
What?	• Product Sales Transaction Information
When?	• Past Sales • Current Sales • Future Sales

Viewpoint Categories Address the *Who* Question

One of the most important questions to answer first is "*Who* is the subject of the report?" Getting the knowledge and understanding of your customers, products, and organization is vital for your business' success. By looking at different viewpoints of the data (for example, customer, product, business unit), you can achieve a more comprehensive understanding of product sales.

Answering the *Who* question defines the dimensions of the report and helps to identify your most valuable customers, products, and business units. Once an analysis or report is created for one viewpoint, the analysis or table can easily be modified for another viewpoint. Answering the *Who* question provides insight into the type of business questions specific to that viewpoint.

Answers to the *Who* Question

Consider three viewpoints of the *Who* question:

Viewpoint	Example Business Questions
Customer viewpoint	• Who are the customers and where are they located? Examples in this book are Acme, BigX, and TwoFeet. • Who are the most profitable customers? • What are the bill-to and ship-to locations? • Are all of the customers' purchase needs being met?
Product viewpoint	• Which products are available for sale? Is it possible to categorize them? Examples in this book are shoes, boots, and slippers. • What are the market segments available for the product—retail/catalog? • Do the products have a life cycle—seasonal, shelf-life duration? • What is the business information on each product—price, promotion code, inventory status?

(*continued on next page*)

Viewpoint	Example Business Questions
Organization viewpoint	• How is the organization structured: Region, District, Branch, Territory, Business unit? Examples in this book are Northern, Southern, and Western. • How successful are the marketing and sales campaign promotions?

Product Transaction Information Addresses *What* and *When* Questions

Information about product sales (sales/return, dollar/units, retail/catalog price) is generally stored in product transaction data sets. For any organization, managing inventory control and product distribution are essential to satisfy customers' expectations. For products in demand, customers want easy access to the purchasing and to the delivery of the product.

Answers to the *What* and *When* Questions

Answers to the *What* and *When* questions help to define the quantitative variables to use for analyzing the customer's needs, behavior, and profitability. Statistical analyses are generally based on variables such as sales, quantity, and month. Based on the answers to the *Who* question, conditions can be applied to focus the analysis for specific customers, products, or business units. Tracking products from inventory to shipment becomes critical to not only managing cash flow but also monitoring delivery time and location of large shipments.

Answers to the *What* Questions

Consider these fundamental questions:

Transaction	Example Business Questions
Inventory	• Which products are currently available to be shipped? • How long have the products been on the shelf? How long does it take for products to move from inventory to retail shops to consumers?
Shipment	• What are the ship-to locations? • Which products have been shipped? Weekly? Monthly? Quarterly? Yearly? • What is the sales information: Sales/Return? Dollars/Units? Retail price?

Snapshots in Time Analyze *When* Questions

Because it is important to sell products as soon as they are available, time periods are an important dimension to analyze. The data stored in the product transaction data set should contain the dates of all relevant events such as inventory date, shelf date, ship date and return date. This information is vital for understanding where your product is in the sales process and how long it took to complete the sales transaction.

By analyzing the sales growth from previous years and current sales distribution, you are better able to forecast future sales projections. This is important for planning manufacturing requirements and product distribution schedules.

Answers to the *When* Questions

Consider these fundamental questions:

Time	Example Business Questions
Past Sales	What is the percent increase in product sales from last year?
Current Sales	What is the percent distribution of product sales during the time period (weekly, monthly, quarterly, yearly, from beginning)?
Future Sales	Analyze the percent growth to capture new market share.

The overall objective for business analysts is to use software tools to help monitor and develop programs to retain customers, sell focused products (cross-selling), and sell profitable products (up-selling). SAS Learning Edition can be used to create customer profiles, measure return on investment, and gauge the success of marketing activities. Note that information collected on your competition can also be considered another viewpoint to answer the ***Who*** question. By analyzing this data, you can identify your biggest competitor.

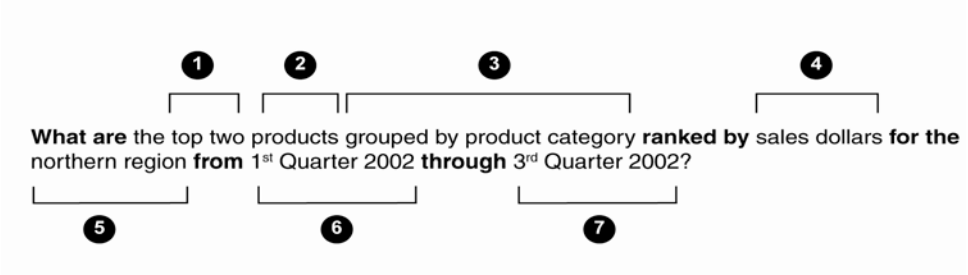
In this book, the examples ask *Who*, *What*, and *When* questions about the sales data from a shoe business. Queries created from the Sales data set will be accessed and analyzed.

Typical Core Business Questions

SAS Learning Edition has the advanced software tools to answer analytical questions. Before selecting a task, it is first helpful to identify the typical core business question which can be expressed into any combination of the following seven options:

What are ❶ All or the top <number> of ❷ Customers, Products or Sales Representative ❸ Grouped by Product Category, Customer Size or Region	Who?
Ranked by ❹ Sales/Return, Dollars/Units, etc. **For a given** ❺ Customer, Region, Market, Channel, Product, or Price Structure	What?
From ❻ Beginning or Release Date **Through** ❼ End or Current Date?	When?

An example business question might be the following:



To answer this type of typical business question, you need to perform the following SAS Enterprise Guide tasks:

1. Access the Sales Query.
2. Create a new Query to restrict the data for northern region and sales from 1st Quarter 2002 to 3rd Quarter 2002.

3. Run the Summary Table task to summarize data by Product sales and save the results to a data set.
4. Access the Summary Table result data set.
5. Run the Rank task to rank by total product sales and group by product category.

Table 3.3 in Chapter 3 shows the steps required for this type of business-related question.

Project Setup and Overview of Tasks

The first requirement in using SAS Learning Edition is to set up a *project*. When you start SAS Learning Edition, the application asks you to create a new project or open an existing project. Once a project is opened, tasks can be selected and executed. The results of tasks are stored in these projects.

For each task, there are nodes for each part of the Results, a Code icon, and a Log icon:

- The Results icon can be expanded to display versions of the output file as HTML or RTF for example.
- The Code icon contains the SAS code generated for the task. You can also view the SAS code by selecting the Preview Task Code option after selecting the task.
- The Log icon contains the results from running the task.

This approach organizes all results, SAS code, and log file for better management. See the Create Note task at the results level later in this section for a screenshot of all the task's nodes.

In this book, you review the following main categories of tasks:

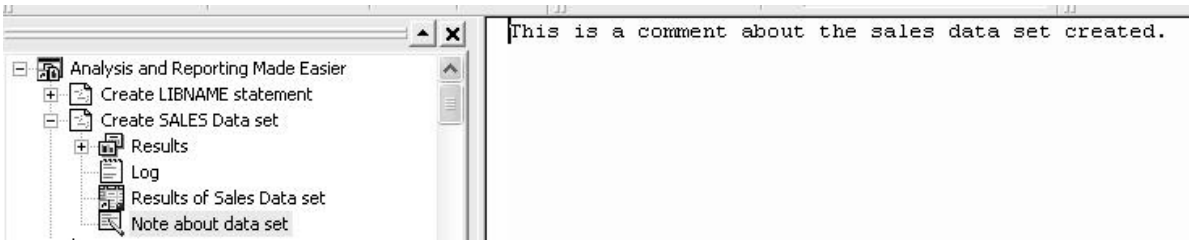
Task Category	Description
Project Setup	Create LIBREF, data set, and formats.
Data Management	Join data sets, add variables, and subset data.
Analysis and Reporting	Summarize, list, and plot data.

Project setup tasks ensure that the correct data sets are accessed or created and that formats are created and assigned to variables. These are important prerequisites before running any of the data management or analysis and reporting tasks. By including data sets in your project using the point-and-click approach, SAS Learning Edition automatically assigns a LIBREF. Programmers might want to see Chapter 5 for instructions to create a LIBNAME statement. See Chapter 3 to create and assign formats.

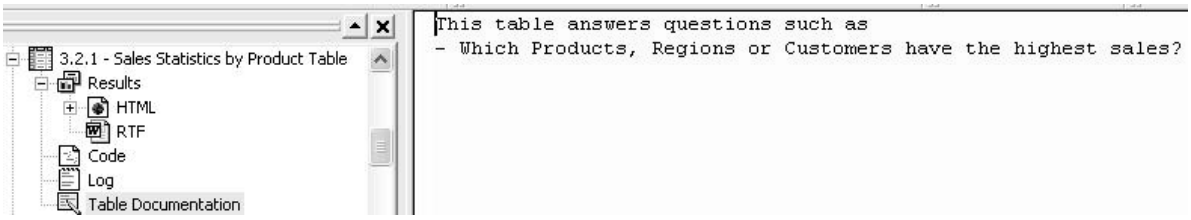
Project Setup Tasks	Description	Example In
Create Code	Assign LIBREF to point to data set directory.	Chapter 5
Create Code	Create permanent data set.	Chapter 2
Create Formats	Assign display labels.	Chapter 3
Create Note	Add your own notes to task.	Chapter 3

Although it is not required, you might want to rename your tasks with more meaningful descriptions. You can do this by selecting the task and clicking the right mouse button and then selecting the rename option. In addition, you might want to select the Create Note task to improve documentation.

Use the Create Note task at the Task level to document the purpose of the task.



Use the Create Note task at the Results level to explain the results.



Data Management Tasks

Data management tasks enable you to add data, join data sets or files together, add variables and formats, and filter the data set. This flexibility allows you, for example, to create a customized data set. Alternatively, you can use this task to create data sets from a text file containing raw data.

Note that because SAS Learning Edition Version 2.0 does not allow you to subset your data within tasks, you must create a query and apply a filter first. The subset data set is then used in the analysis and reporting tasks. See Chapter 3 for creating queries.

Each example in this book accesses one of the queries created from the Sales data set and the Excel file Repnames.xls based on the conditions below. A description of the Sales data set is provided at the end of this section. The Repnames.xls file contains data that assigns representative names to each district.

As discussed, the benefit of this approach is to take advantage of the Query window's feature to combine data, create and assign variables, and assign formats. In addition, in some cases, creating a query data set is required to first subset the data set before running the analysis or reporting task. See Chapter 3 for instructions on creating the Sales, Single Product Sales, and 2002 Shoes and Slippers Sales Queries. In general, examples in this book access the Sales query unless indicated otherwise.

Query Name	Content
Sales Query	All data (Sales, data set and Excel file)
Single Products Sales Query	Boot sales: Sales Query where `product = 'Boots'`
Shoes and Slipper 2002 Sales Query	2002 Shoes and Slipper sales: Sales Query where `year = 2002 and product = 'Shoes' or 'Slipper'`

Analysis and Reporting Tasks

Once the data set is prepared and is selected in the project, then *analysis* and *reporting tasks* can be used to create complex tables and lists. For tables using all records and existing variables, the actual Sales data set can be selected. For tables based on a subset condition or requiring new variables, a query data set should be accessed since tasks do not support subset conditions or create variables. Note that the results of all tasks are grouped by the active data set and positioned in the order the task was executed. SAS Learning Edition has this convention to organize the analysis performed. In addition, this makes it easier to identify related tables from the same source data set or query.

Sales Data Set

The Sales data set used in this book contains hypothetical shoe sales data.

The structure of the Sales data set is such that a record is created for each customer purchase of products in a given month. As shown in Figure 1.4, for example, Acme had shoe sales of $300 from the catalog market during the first month of 2001 from the North Eastern district. Thus, there will be no more than one observation for a given customer, product, market, year, and month to record that sales transaction.

The Sales data set consists of the variables below which represent what a typical business would require. The values and ranges of each variable are listed along with their formats. In addition, each variable is categorized as **categorical** or **continuous**. Categorical variables have discrete values such as shoes and slippers while continuous variables have minimum and maximum values. In general, categorical variables are character variables and continuous variables are numeric variables. As discussed, the variable type is useful to determine which specific roles within tasks the variable can be used for since roles can be restricted to character or numeric variables. Note that when formats are applied to variables, the variable icon can change to represent that format. Note also that variables DATE and SALESGRP are created using the Query window in Chapter 3.

Variable Name	Value/Range	Classification	Format
CATEGORY	Shoes, Slippers	Categorical	
CUSTOMER	Acme, BigX, TwoFeet	Categorical	
*DATE	Min – Max	Continuous	
DISTRICT	1. NE, 3. SW, 5. CW	Categorical	DISTRICT.
EXPENSE	Min – Max	Continuous	
MARKET	1. Catalog, 2. Retail, 3. Wholesale	Categorical	MARKETV.
MONTH	1 – 12	Categorical	
PRODUCT	Boots, Shoes, Slippers	Categorical	
QUANTITY	Min – Max	Continuous	COMMA6.0
QUARTER	1, 2, 3, 4	Categorical	
REGION	1. Northern, 2. Southern, 4. Western	Categorical	RETGIONV.
**SALES	Min – Max	Continuous	DOLLAR6.
*SALESGRP	Low, Moderate, High	Categorical	SALEGRP.
YEAR	2001, 2002	Categorical	

* DATE and SALESGRP are new variables created using the Query window. See Chapter 3 for more information.

** Products returned are considered as return sales and are stored as negative numbers in the SALES variable.

Creating the Sales Data Set

See the appendix for the SAS code to create the Sales data set. You can also download the SAS code to create the Sales data set from the SAS companion web site for this book. Visit the www.sas.support.com Web site and drill down to the SAS Press site.

Follow these steps to create the Sales data set.

1. Start SAS Learning Edition.
2. Create a new project or access an existing project.
3. Create or assign a directory (such as **`c:\SAS_LE_Data`**) and a libref (such as mydata) to store the Sales data set.

   As an alternative, you can save the Sales data set in the SASUSER library. The SASUSER library is a reserved libref for accessing permanent data sets. In general, the SASUSER libref references a directory such as **`C:\Documents and Settings\username\My Documents\My SAS Files\V8\`**. See Chapter 5 for instructions to create a LIBNAME statement.
4. Select the Create Code task under the Add Items to Project task.
5. Copy the SAS code to create the Sales data set from the SAS companion web site and then paste it in the Create Code task window.
6. Select `Code` ► `Run on Local` from the toolbar to submit your program.

This creates the Sales data set as a permanent data set in the assigned directory or SASUSER library.

You can now access the Sales data set by clicking on the Results of the Sales data set icon from within the Create Code task. Once the Sales data set is accessed, you can select and run tasks. To create queries based on the Sales data set, access the Sales data set by browsing to the location of the assigned directory or the SASUSER directory. See Chapter 3 for instructions to create a query using the Sales data set.

Using Tasks to Answer Your Business Questions

Using tasks in SAS Learning Edition is easy because you can select tasks by category (Figure 2.1), by name (Figure 2.2), or from the toolbar (Figure 2.3). As a non-programmer, you can first identify the category such as Data or Descriptive and then select the task within that category. As a programmer, you have the option to directly select the SAS procedure you want to access. The task associated with that SAS procedure is selected. Finally, you can open the tasks by using the menus from the toolbar to produce the same results. Select the `Data` ►, `Analysis` ►, `or Graph`► from the main menu to perform similar tasks. Remember to first open or create a project and select a data set or query before performing any task.

Figure 2.1 Select Tasks by Category

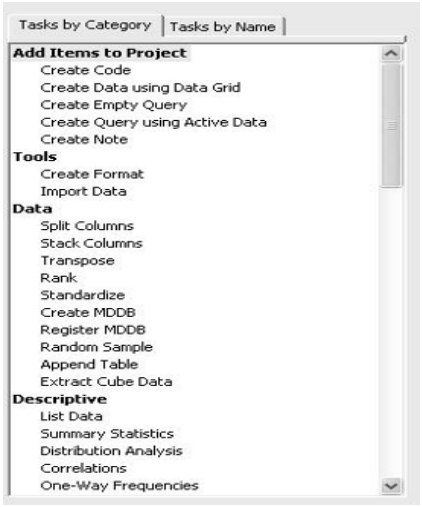


Figure 2.2 Select Tasks by Name

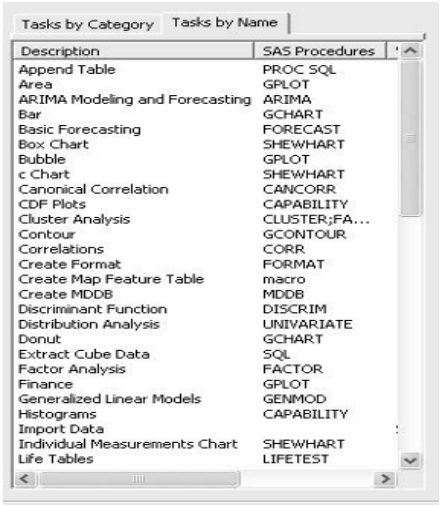
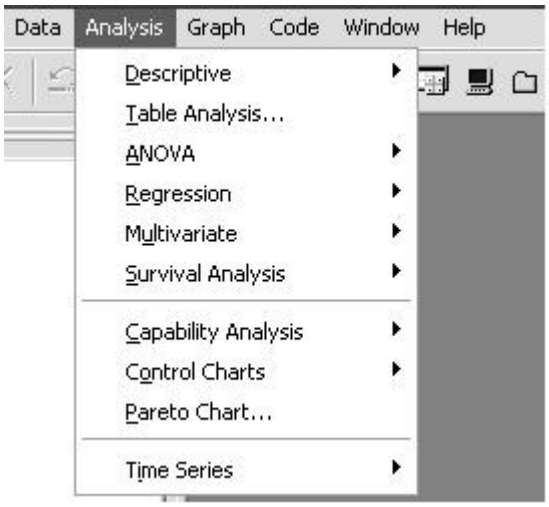


Figure 2.3 Select Tasks by Toolbar Menu



You can use the point-and-click tasks in SAS Learning Edition to prepare the data, run analyses, and create reports and graphs. It is important to note that a data set must first be selected to activate it before selecting an analysis or reporting task. All analysis and reporting tasks must access and process a data set for the dialog boxes to appear.

Programmers can recognize that each task is different because it is designed to accept input to meet the requirements of the specified SAS procedure. This enables SAS Learning Edition to effectively use dialog boxes to construct the SAS code in the background based on your selections. In general, each task is a collection of tabs. The tabs represent different categories of options.

Using the point-and-click method to make a data set active and select a task, you can use the drag-and-drop approach to assign variables and to select options. For example, "Example Instructions for Summary Table task" shows the steps for selecting the Summary Table Task. The data set should be selected from the Project Window, and the task should be selected from the Task List Window. Detail instructions are provided for each tab.

Three tabs—Columns, Options, and Titles—are common across tasks.

- The **Columns** tab assigns variables from the active data set to roles. SAS Learning Edition makes it easy to assign the correct type of variable to the tab roles by using the numeric icon and character icon for each role.
- The **Options** tab assigns statistics and table groups.
- The **Titles** tab assigns the table title.

In addition, there might be task-specific tabs such as the Results tab in the Summary Table task to assign the output format. All of the point-and-click dialog boxes are similar for each task. You can advance through the tabs by selecting the tab name or by selecting the **Back** or **Next** buttons. Selecting the **Finish** button runs the task and generates the results. All of the examples in this book show each tab's settings to create the report or graph.

Example Instructions for Summary Table Task

Click the **Sales Query** to activate it.

Click the **Summary Tables task** under the Descriptive Category.

Review the tab selections in the Summary Tables task:

Columns tab Drag and drop the Sales Query variables from the **Variables to assign** list on the left to the **Summary Tables roles** as follows:

Table tab Assign table rows, columns, and statistics.

Results tab Assign the output format.

Titles tab Add a descriptive title and footnote.

SAS Code Automatically Created

Using SAS Enterprise Guide tasks is a good starting point for creating customized reports, if needed. Since SAS Learning Edition automatically creates SAS code, you do not need to remember the SAS syntax for each SAS procedure. Customization of the code is required only to specify SAS procedure options that are not available through the task. Note that once the generated code is customized, the code can no longer be accessed by the task. See Chapter 5 for more information on modifying and writing SAS code.

Selecting the Right Task

Before selecting the SAS Enterprise Guide tasks, it is helpful to first review the specifications of the request. Typically, the request is to answer business questions such as: Which products, regions, or customers have the highest sales? The message of the table layout addresses this request. For example, a table such as "Sales Statistics by Product" should clearly show which products have the highest sales.

How well do you know your business? SAS Enterprise Guide tasks can assist you in getting to know your business better at any given moment. You can quickly determine which is best; for example:

- Which is the best-selling product? Shoes?
- Which customer buys the most? Acme?
- Which region has the best growth in sales? The Northern region?
- What market segment is performing the best? Retail sales?
- What time period has shown the best growth? Fourth quarter of 2001?

This section has three topics:

- Data Management and Querying Requests
- Reporting Writing Requests
- Graph Requests

Within each subsection, a decision tree table is provided to identity the SAS Enterprise Guide task that best answers the business question. A series of typical business questions is provided for each type of report or graph.

Answering Data Management and Querying Requests

Often before analysis can be performed, there is a need to add data, join data sets, create variables, subset data sets, or sort data sets. You might even need to view the variable attributes and assign formats. SAS Learning Edition's Query window can handle all of these requests. Because the focus of this book is on creating reports and graphs, examples related to data management and querying are limited. Answers to selected business-related questions are displayed.

When accessing data sets within SAS Learning Edition, it is important to realize that a copy of the data set is not saved with the project. SAS Learning Edition creates an associated link to the data set to access the data set. This becomes useful because if the source data set is updated; then, the updated information is automatically accessed. Data sets can also be created within SAS Learning Edition.

The decision tree Table 2.1 contains typical business questions and how SAS Learning Edition can be used to answer these requests. The business questions are grouped by Data Management categories. Each category indicates specific tabs within the Query window to perform the request.

See Figures 3.1 to 3.5 in Chapter 3 for examples to answer each of these business questions. For additional information on data management and querying, see the examples in the online tutorial.

Table 2.1 Typical Questions Relating to Data Preparation and Data Queries

Data Management Categories	SAS Enterprise Guide Task	Figure #
Combining data sets or files together to create new data sets	Query window (Tables Tab)	
How do you add data to your project? How do you combine sales data with customer information?		3.1
Data set properties	Query window (Select and Sort Tab, Properties)	
How can you view variable attributes and create and assign formats?		3.2
Creating variables – numeric, character, dates	Query window (Select and Sort Tab)	
How can you create new variables such as salesgrp and date?		3.3
Querying your data	Query window (Filter Data Tab)	
How can you select sales records for a specific customer, product and month?		3.4
Selecting observations to subset data: numeric, character, dates	Query window (Filter Data Tab, Parameter Tab)	
How do you focus your analysis on a specific product and allow the user to select the product each time?		3.5

Answering Report Writing Requests

In general, reports can be categorized by the type of information requested using one of the following five basic kinds of reports: Sales Statistics, Percent Distribution, Percent Change, Current Year Sales, and Detail List Comparison.

The decision tree Table 2.2 lists typical business questions and how SAS Learning Edition can be used to answer these requests. The business questions are grouped by the type of table requested. Each table type has a specific task to create the table. By reviewing the list of typical business questions below, you can identify the specific task that will best convey your message. All tasks and SAS code to answer these questions are in chapters 3 and 5.

Note that all examples in Chapter 5 require some basic knowledge of SAS programming because you will work directly with the SAS code.

Table 2.2 Typical Business Questions That Can Be Answered by Sales Reports

Table Type	SAS Enterprise Guide Task	Table Format / Table #
Sales Statistics	Summary Tables	Mean Min
Which products, regions or customers had the highest sales?		3.2 – 3.4
How did your products rank with other similar products?		3.5
What were the descriptive sales statistics, such as total, mean, median, and standard deviation by product?		5.8
Percent Distribution	Summary Tables	N %
Which products had the largest volume? What was the percentage distribution by market segment?		3.6, 3.7
What was the sales percent distribution by product category and market segment across time periods?		3.8, 3.9
Which product by year had the highest sales?		5.10, 5.11
Percent Change	Create Code	N % Change
Based on the percent sales change, where is your product in the product life cycle (introduction, growth, maturity, decline)? Which products had the best percent sales change by year?		5.12
Which products had the best percent sales change by quarter?		5.13, 5.14

(*continued on next page*)

Table 2.2 (*continued*)

Table Type	SAS Enterprise Guide Task	Table Format / Table #
Current Year Sales	Create Code	Quarter Year
What was the overall sales performance by quarter and product, customer, or region? How did a customer's product sale compare with other customers' product sales? Was there a pattern from quarter to quarter or from year to year?		5.15 – 5.20
Detail List Comparison	List Data	Product Sales
How do you identify monthly transactions of products that are grouped by product sales of low, average, and high?		3.4
Which were the top customers for a selected product?		5.9

Answering Graph Requests

In general, graphs can be categorized by the type of comparison requested using one of the following four basic kinds of comparisons: Percent Distribution, Product Sales, Product Group, and Time Period.

The decision tree Table 2.3 lists typical business questions and how SAS Learning Edition can be used to answer these requests. The business questions are grouped by the type of chart requested. Each chart type has a specific task to generate the chart. By reviewing the list of typical business questions below, you can identify the specific task that will best convey your message. All tasks to answer these questions are in Chapter 4. Note that Chapter 5 contains one example to customize your graph.

Note that the icon for horizontal and vertical bars is the same because both types of graphs are created from the same SAS Enterprise Guide task.

Table 2.3 Typical Business Questions That Can Be Answered by Sales Charts

Chart Type	SAS Enterprise Guide Task	Figure #
Comparing Percent Distribution	Pie	
Which products accounted for the largest share of total company sales?		4.3, 4.5
Comparing Product Sales	Horizontal Bars	
Which products were top performers based on total sales?		4.5
Comparing Product Group Sales	Horizontal Bars Vertical Bars	
When comparing shoes and slipper sales, which customer had higher sales?		4.6
When comparing shoes and slipper sales for selected customers, which customer had higher sales?		5.2
Time Period Comparison	Vertical Bars Line	
How much have sales risen this year?		4.7 – 4.9
Over a period of time, how do shoe sales compare with slipper sales?		4.10

Chapter 3

Preparing the Data and Producing Reports for Distribution

IN THIS CHAPTER

Preparing reports that are easy to understand is most helpful. It's also important that reports can be prepared quickly. In this chapter, you complete the steps to prepare your data and produce a variety of reports.

	product	category	customer	year	month	quarter	market	sales	expense	region
1	Boots	Shoes	Acme	2002	1	1	Catalog	$-1400	-980	Northern
2	Boots	Shoes	Acme	2002	2	1	Catalog	$700	490	Northern
3	Boots	Shoes	Acme	2002	3	1	Catalog	$-900	-630	Northern
4	Boots	Shoes	Acme	2002	4	2	Catalog	$1,400	980	Northern
5	Boots	Shoes	Acme	2002	5	2	Retail	$200	140	Northern
6	Boots	Shoes	Acme	2002	6	2	Retail	$1,800	1260	Northern
7	Boots	Shoes	Acme	2002	7	3	Retail	$-900	-630	Northern
8	Boots	Shoes	TwoFeet	2002	8	3	Retail	$3,500	2450	Southern
9	Boots	Shoes	Acme	2002	9	3	Retail	$3,000	2100	Northern
10	Boots	Shoes	Acme	2002	10	4	Retail	$3,000	2100	Northern
11	Boots	Shoes	BigX	2002	11	4	Retail	$3,100	2170	Western
12	Boots	Shoes	Acme	2002	12	4	Retail	$2,300	1610	Northern

In Chapter 2, you learned how to create the Sales data set and access the Repnames.xls Microsoft Excel file that you'll use to create reports like the one below:

Table 3.1 Sales Statistics by Product

			Product Sales		
		Unit Totals	Dollar Totals	Average (Mean)	Min
Product Category	Product			$1,496	
Shoes	Boots	5,382	$35,900		$-1400
	Shoes	4,180	$43,700	$1,821	$100
Slippers	Slippers	8,360	$41,800	$1,742	$0
Total		17,922	$121,400	$1,686	$-1400

In Chapter 2, you also learned about a series of business-related questions to use when you need to analyze data and produce results. The examples in this chapter review and illustrate the SAS Enterprise Guide tasks used to answer many of those questions. Data Management tasks are required to add data, and combine data sets, as well as add variables and formats, and subset data sets. The first section of this chapter shows how to prepare the data to be used in your project.

To create reports, you make a data set active and then select the specified SAS Enterprise Guide task along with the variables and options you need. The last section of this chapter shows how to create reports using the Sales query.

Combining Data to Create a Query

In this section, you use the Query window to add the Sales data set to your project, combine sales data with customer information, view, and assign variable attributes, create new variables and formats, and query the data based on specific conditions. The result of the query is a temporary data set that can be used by the analysis and reporting tasks.

Add Data to Your Project

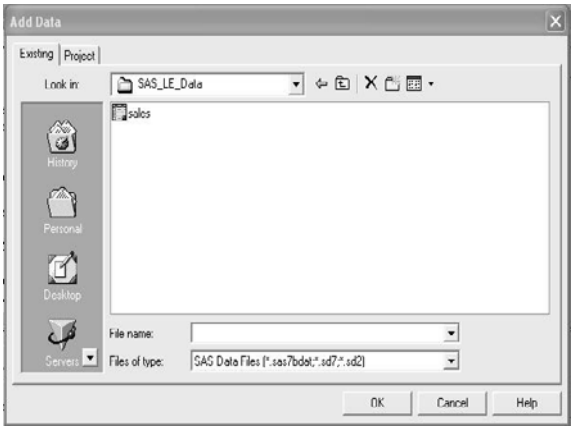
You use the Query window to add the Sales data set to your project. SAS Learning Edition enables you to access data across multiple platforms, operating systems, and databases. You can access and combine data from any of the following sources:

- SAS data sets
- Microsoft Access files
- Microsoft Excel files, including individual worksheets
- Dbase files
- Lotus and Paradox files
- Text and HTML files

Select the Create Empty Query task from under the Add Items to Project category. In this book, we use the term Query window to refer to the Create Empty Query task or the Create Query using Active Data task.

Once the Query window is opened, select the Add Data option to add data to your project. Next, browse to the Sales data set and then select OK to access the data set. In the example window, the Sales data set is available in the SAS_LE_Data directory. If you used the SASUSER libref, then you can locate the Sales data set in the directory such as `C:\Documents and Settings\username\My Documents\My SAS Files\V8\`. See Chapter 2 for instructions on creating the Sales data set. This step can be repeated for any SAS data set that you want to add to your query.

Query Window: Add Data Option



More about Your Data

Instead of making a copy of the data in the project, SAS Learning Edition sets a pointer to the source data; this pointer is a link to the data wherever they are stored. This is very useful because any updates to the source data will automatically be accessed. After selecting the data to activate them, SAS Learning Edition uses this pointer when processing tasks. Programmers can see instructions in Chapter 5 to create a LIBNAME statement. Once executed, the LIBNAME statement can be used to reference data sets.

Once the data are in SAS Learning Edition, you can view the data set without having to get into the SAS Viewer. The default setting when opening data sets is read only. As discussed in Chapter 2, adding all of the data to your project is an important first step.

Combining Sales Data with Customer Information

Once you have a data set in the project, you usually need to add and combine more data files. In this section, you add the Excel file Repnames.xls—containing the representative names for each district—to your project, and then combine this file with the Sales data set.

The power of the Query window is in its ability to construct simple and complex queries. Through a very intuitive visual drag-and-link process, the data files can be linked together. When the data files are brought into the Query window, links are automatically created for variables with similar names and types. If needed, you can manually create a link between any two variables if SAS Learning Edition does not automatically detect a join condition.

Once you establish the link, you can select the Run Query option to create a query that contains the Sales data set along with the Excel file joined by the common variable DISTRICT.

Figure 3.1 shows the results of joining the Sales data set with the Microsoft Excel file Repnames.xls, creating new variables (SALESGRP, DATE), and creating and assigning formats (SALEGRP., MARKETV., REGIONV., and DISTRICT.). The steps below show the details to create this query.

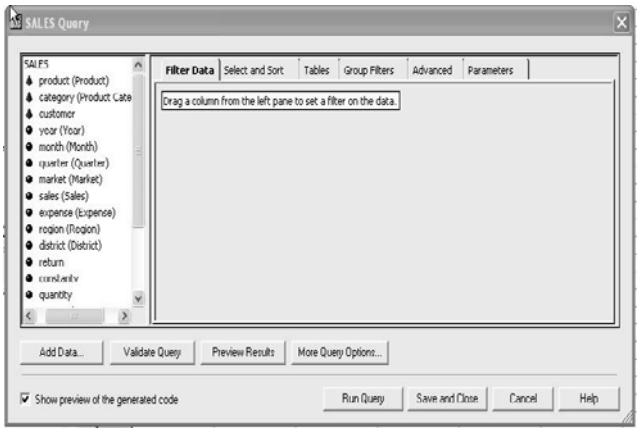
Figure 3.1 Sales Data Set Joined with an Excel File (Partial)

	expense	region	district	return	constantv	quantity	salesgrp	date	district	repname2
1	240	Northern	NE	0	1	30	Average Sales	01JAN01	1	Sara
2	1540	Northern	NE	0	1	275	Average Sales	01JAN01	1	Sara
3	540	Northern	NE	0	1	180	Average Sales	01JAN01	1	Sara
4	80	Northern	NE	0	1	10	Average Sales	01FEB01	1	Sara
5	980	Northern	NE	0	1	175	Average Sales	01FEB01	1	Sara
6	0	Northern	NE	0	1	0	Average Sales	01FEB01	1	Sara
7	480	Northern	NE	0	1	60	Average Sales	01MAR01	1	Sara
8	0	Northern	NE	0	1	0	Average Sales	01MAR01	1	Sara
9	840	Northern	NE	0	1	280	Average Sales	01MAR01	1	Sara
10	2080	Northern	NE	0	1	260	High Sales	01APR01	1	Sara

In the Project window, click the Sales data set to activate it.

In the **Add Items to Project** category, click the **Query** window.

In the Query window, click the **Add Data** option.

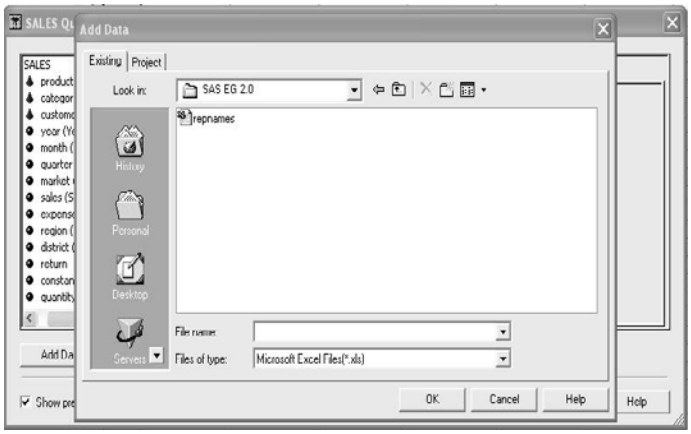


More about Adding Data

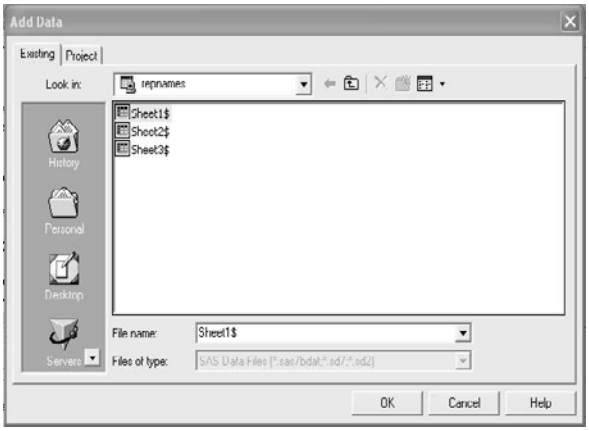
Use the Add Data button to include other data files in the query. Alternatively, you can include the Excel file using the Insert Data task. This makes the data file available in the current query.

Browse to file list.

Select the Excel file Repnames.xls to include in the project.



Select the Excel worksheet file Sheet1.

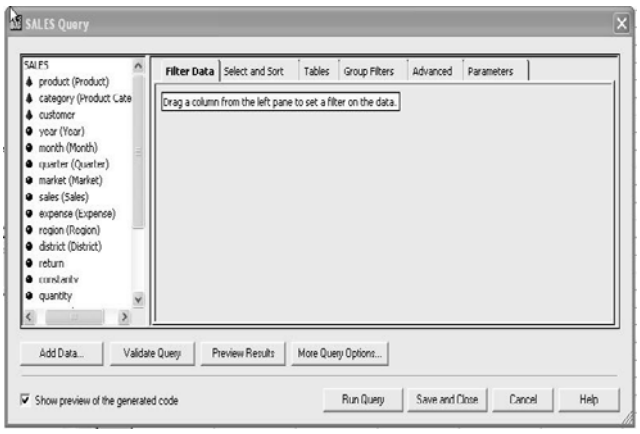


Because SAS Learning Edition automatically determines the variable type, it is recommended that you confirm this. Confirm that DISTRICT is a numeric variable and REPNAME is a character variable.

The contents of the Excel file are displayed below. After joining the Excel file with the Sales data set, you should, for example, see the sales representative Sara for District 1.

	A	B
1	district	repname
2	1	Sara
3	2	John
4	3	Tom
5	4	Steve
6	5	Bella

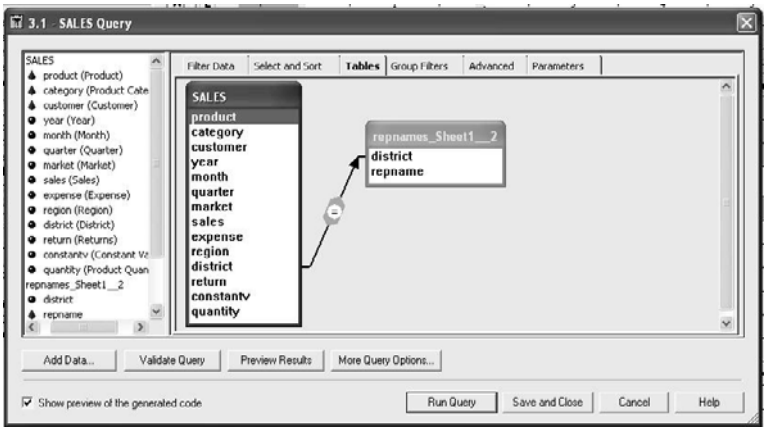
This is an example of the Query window:



The Query window has the following tabs:

Tab	Description
Filter Data	Select variables and values to be used in the query
Select and Sort	Select the variables to be saved in the query and their sort order
Tables	Drag and drop files to be included in the query
Group Filters	Apply conditions to variables; used to group the data (similar to the Filter Data tab)
Advanced	Access files located on other servers
Parameters	Create user-specified queries

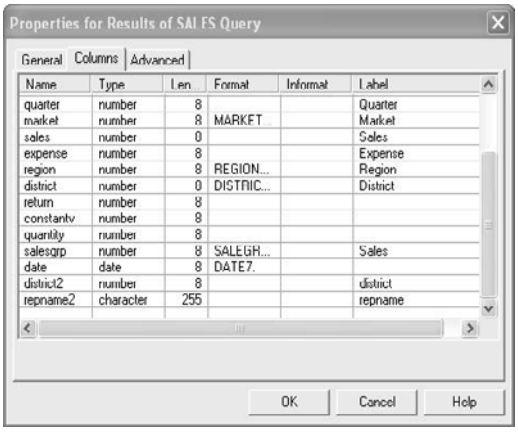
In this example, select the **Tables** tab to drag and drop the Excel file Repnames.xls from the file list to the tables area. Since the DISTRICT variable is common in both files, SAS Learning Edition automatically links the two files by this variable. Make sure that the variable type is the same for both variables.



More about Joining Files

As discussed in Chapter 1, there should be key common variables between files that you want to join. If the common variables have different names, SAS Learning Edition will display the message, "A suitable join could not be determined for the new table. You might need to join tables manually." You will then need to manually link the two files by clicking and dragging the variable in the first file over the corresponding variable in the other file.

Once completed, you can select the query, right-click it and then select Properties and then the Columns tab to view the variable attributes.

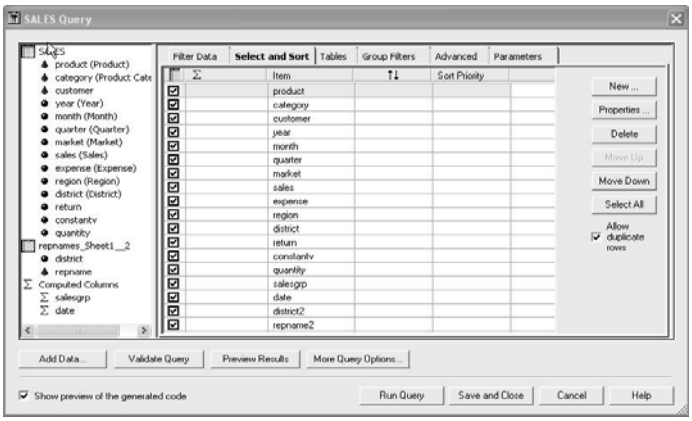


Viewing Variable Attributes to Create and Assign Formats

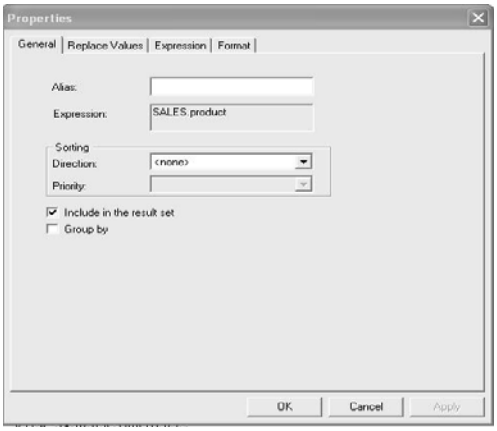
Once you have your data in SAS Learning Edition, you might want to view the variable attributes to assign and confirm them. You can view, assign, and change variable attributes in the Query window.

In the Query window Select and Sort tab, select a variable.

Figure 3.2 Query Window: View Variable Attributes



Select the Properties button to view the variable attributes.



Working with Formats

Introduction

As discussed in Chapter 1, formats are useful to display values in a more user-friendly fashion. SAS Learning Edition supplies formats that allow you to display variables in a more recognizable form. Also, you can create user-defined formats that provide formats that meet your specific needs and are separate from the formats supplied by SAS Learning Edition. A user-defined format such as MARKETV. will display 'Catalog' for the value 1, 'Retail' for the value 2, and so on. This helps make the report or analysis more meaningful. Note that the actual data values are not changed.

In this section, you use the Create Format task to create the following user-defined formats: MARKETV., REGIONV., DISTRICT., and SALEGRP. These formats are assigned to the corresponding variables: MARKET, REGION, DISTRICT, and SALESGRP. This example shows how to create the REGIONV. and SALEGRP. user-defined formats.

In addition, you should assign formats to the SALES (DOLLAR6.) and QUANTITY (COMMA6.0) variables to improve readability. Using these formats, for example, will display SALES as $35,900 instead of 35900 and QUANTITY as 5,382 instead of 5382. The DOLLAR6. and COMMA6.0 formats, along with the SAS-supplied format, are in SAS Learning Edition.

Note: Experienced SAS programmers might find it easier to create formats by copying the PROC FORMAT code from the SAS program in the Appendix and insert it in the Create Code task. See Chapter 5 for more information on using the Create Code task.

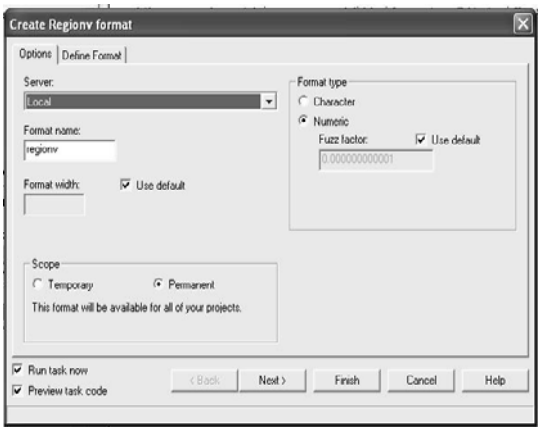
Format Names

Note that format names can be up to eight characters long for numeric variables and up to seven characters long for character variables. This is because for character formats, SAS places the character '$' as the start of the format name. As discussed, because SAS automatically places a period at the end of each format name, you should not enter a period as part of the format name when creating the format. When creating user-defined formats, make them permanent so that they will be available in all of your projects. In addition, once formats are assigned to variables, they are always used when running tasks.

Create a Format

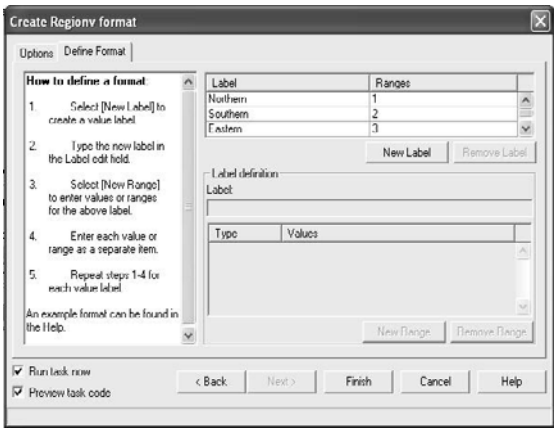
Select the **Create Format** task under the **Tools** category.

In the **Options** tab, enter **`regionv`** for **Format name** and select Numeric for **Format type**. Note that the period is not entered in the format name because SAS automatically does this. You can keep the value Local for **Server** and Permanent for **Scope**.



In the Create format window, select the **Define Format** tab. Follow the instructions to enter the labels **Northern**, **Southern**, **Eastern**, and **Western** along with the corresponding Range values **1**, **2**, **3**, and **4**.

In this example, select **New Label** and enter the label **`Northern`** in the **Label** field. Then, select **New Range** and enter the value **`1`** in the **Ranges** box.



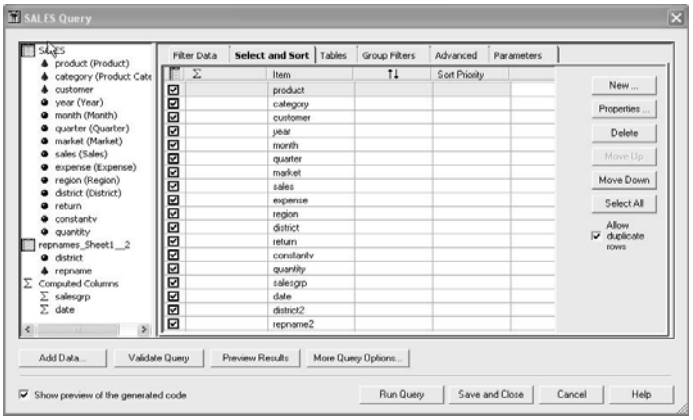
Repeat this process for the MARKETV. and DISTRICT. formats by specifying the following values:

Format	Formatted Value	Data Value
MARKETV.	Catalog	1
	Retail	2
	Wholesale	3
	Other	4
DISTRICT.	Northeast	1
	South East	2
	South West	3
	North West	4
	Central West	5

When all formats are created, you can assign formats to each variable. Although you can assign formats to multiple variables, you will assign (in this book) each format to one variable as follows: MARKETV. (MARKET), REGIONV. (REGION), DISTRICT. (DISTRICT), and SALEGRP. (SALESGRP). Note that the SALEGRP. format can be assigned once the SALESGRP variable is created in the next section.

Assign Formats

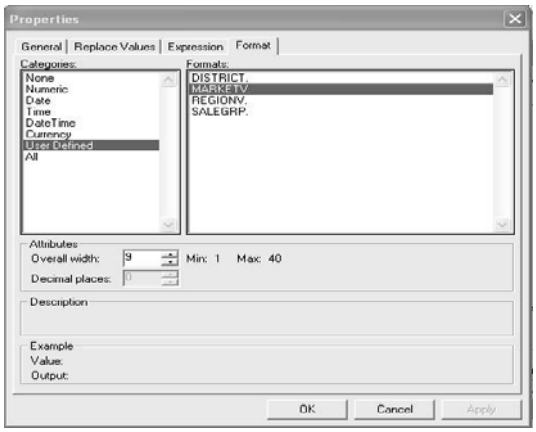
From the **Select and Sort** tab, select the specific variable and select **Properties**.



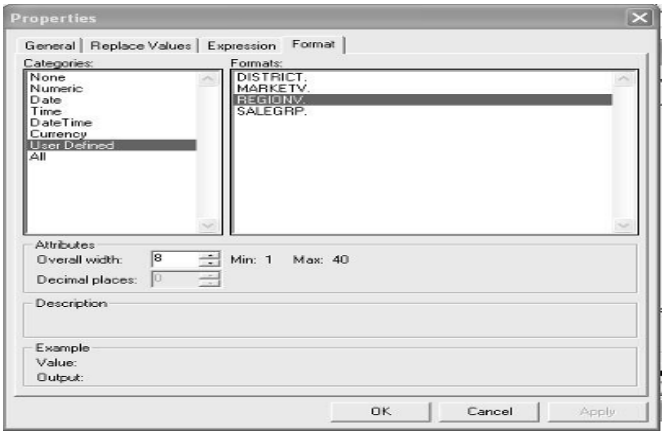
In the Properties window, select the **Format** tab.

Format tab

Select **User Defined** under **Categories**. From this selection, select the corresponding appropriate format. The following window shows the MARKETV. format being assigned.



The following window shows the REGIONV. format being assigned.



Once formats are assigned to variables, they become permanent and are used in reporting and analysis tasks.

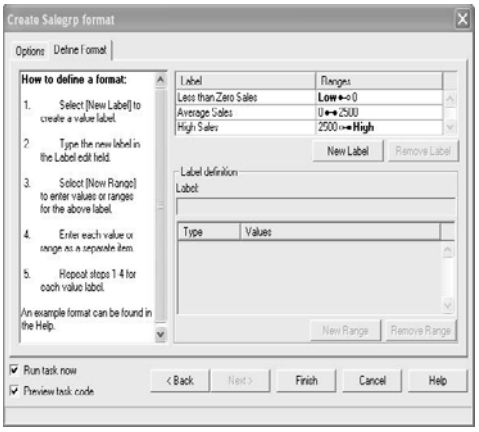
More about Using Formats

As discussed, formats can be used to group a range of continuous values to classify continuous variables. When using the SALEGRP. format, for example, instead of displaying the actual sales amount, you can display "Less than Zero Sales", "Average Sales", or "High Sales" depending on the sales value. See Figure 3.6 (page 81) for a task using this format. Note that the SALEGRP. format is being assigned to the SALESGRP variable which is created in the next section. The reason for assigning this format to a new variable and not the SALES variable is to be able to display the actual sales value if needed.

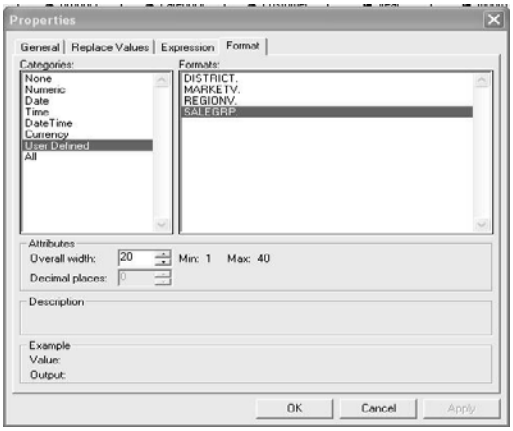
Now you can create the SALEGRP. format for the SALESGRP variable. Select the **Create Format** task, and enter the values shown below:

Format	Formatted Value	Data Value
SALEGRP.	Less than zero sales	<0
	Average Sales	0 – 2500
	High Sales	>2500

Be careful to select the correct ranges for each label. Enter the value `2500` and inclusive for **Average Sales** and exclusive for **High Sales**. This includes the value of 2500 in the **Average Sales** and not in the **High Sales**. Notice the open and closed circles for exclusive and inclusive respectively. In addition, SAS Learning Edition uses the keywords Low and High to represent missing and infinite respectively.



The following window shows the SALEGRP. format being assigned.

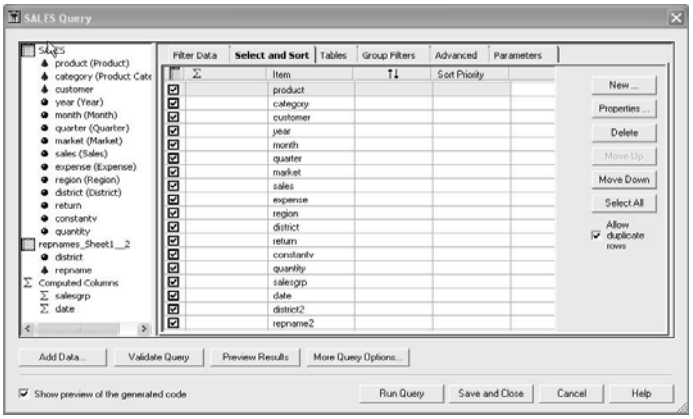


Creating New Variables

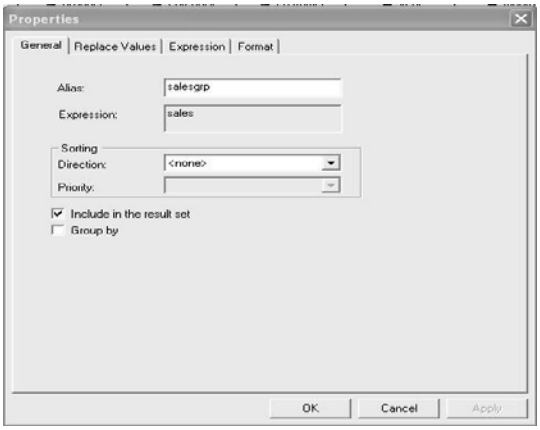
You frequently need to create new variables because they can be helpful in the reporting and analysis process and they do not exist in the source data sets. Typically, you create new variables from existing variables. In this section, you create the SALESGRP variable so that you can group sales by the ranges low, average, and high. You assign the low, average, and high expressions to equal values of the SALES variable and the SALEGRP. format. In addition, you create the DATE variable from the expression MDY (MONTH, 1, YEAR).

From the Select and Sort tab, select the New option to create a variable.

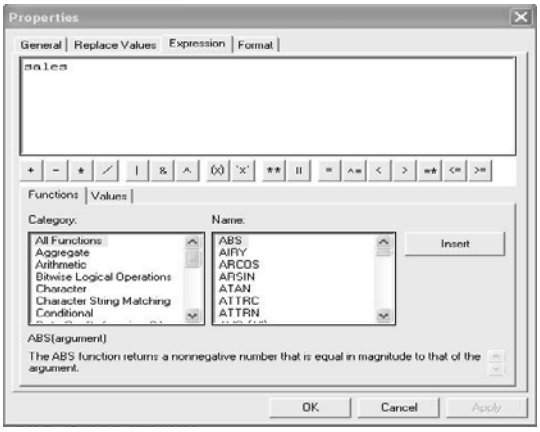
Figure 3.3 Creating SALESGRP and DATE Variables



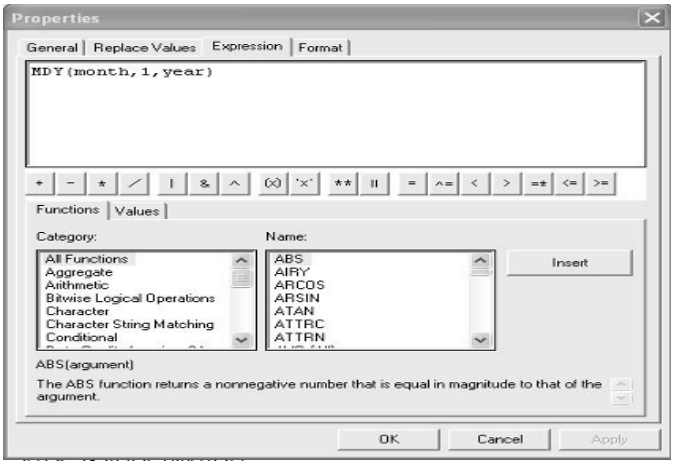
Enter **`salesgrp`** as the Alias.



Next, enter the name of the expression to be used for the SALESGRP variable. Because you want this variable to be equal to the SALES variable, enter **`SALES`** in the **Expression** box and select **OK** to open the **Expression** tab in the Properties window.



Enter the expression for the DATE variable. Because this is a combination of the MONTH and YEAR variables, you can apply the MDY() function and assign the day to be the first day of the month as follows:



The end result of the sales query is shown below. In the salesgrp column, you can now see sales grouped by low, average, and high. In the date column, you can see the date of the sales transaction. Note that the SALEGRP. format was assigned to the SALESGRP variable in the previous section. In addition, the SALESGRP values will be consistent with the sales value as specified by the SALEGRP. format.

	expense	region	district	return	constanty	quantity	salesgrp	date	district	repname2
1	240	Northern	NE	0	1	30	Average Sales	01JAN01	1	Sara
2	1540	Northern	NE	0	1	275	Average Sales	01JAN01	1	Sara
3	540	Northern	NE	0	1	180	Average Sales	01JAN01	1	Sara
4	80	Northern	NE	0	1	10	Average Sales	01FEB01	1	Sara
5	980	Northern	NE	0	1	175	Average Sales	01FEB01	1	Sara
6	0	Northern	NE	0	1	0	Average Sales	01FEB01	1	Sara
7	480	Northern	NE	0	1	60	Average Sales	01MAR01	1	Sara
8	0	Northern	NE	0	1	0	Average Sales	01MAR01	1	Sara
9	840	Northern	NE	0	1	280	Average Sales	01MAR01	1	Sara
10	2080	Northern	NE	0	1	260	High Sales	01APR01	1	Sara

Selecting Sales Records for a Specific Customer, Product, and Month

Now that you have added and combined all of the data and you have created variables and assigned attributes (such as formats) to them, you are ready to query your data.

The Query window makes it easy to select the data that you want to analyze. Perhaps you are interested in only a particular sales representative, product, customer, or time period. The Query window allows you to filter data by selecting variables and variable values using the AND and OR operators.

Note: To perform any analyses or to create tables based on a subset of a data set, you need to query the data set first. This query data set then becomes the active data set for the selected task.

Figure 3.4 shows the results of a query based on selecting the year 2002 and a product (Shoes or Slippers) from the Sales data set. The example query shows the sales information based on these conditions. Once a query has been created, you can perform any analysis or reporting task by accessing the data set created by the query. See Figure 4.6 for an example showing a task using this example query.

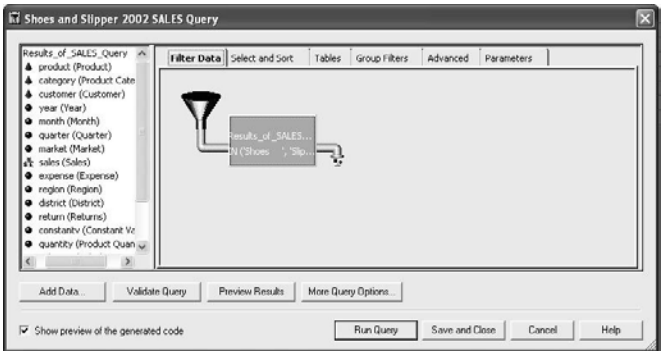
Figure 3.4 Shoes and Slipper 2002 Sales Query (Partial)

	product	category	customer	year	month	quarter	market	sales	expense	region
1	Shoes	Shoes	Acme	2002	1	1	Catalog	1000	800	Northern
2	Slippers	Slippers	Acme	2002	1	1	Catalog	1100	660	Northern
3	Shoes	Shoes	Acme	2002	2	1	Catalog	1000	800	Northern
4	Slippers	Slippers	Acme	2002	2	1	Catalog	2300	1380	Northern
5	Shoes	Shoes	Acme	2002	3	1	Catalog	400	320	Northern
6	Slippers	Slippers	Acme	2002	3	1	Catalog	1100	660	Northern
7	Shoes	Shoes	Acme	2002	4	2	Catalog	2100	1680	Northern
8	Slippers	Slippers	Acme	2002	4	2	Catalog	500	300	Northern
9	Shoes	Shoes	Acme	2002	5	2	Retail	1100	880	Northern
10	Slippers	Slippers	Acme	2002	5	2	Retail	1300	780	Northern

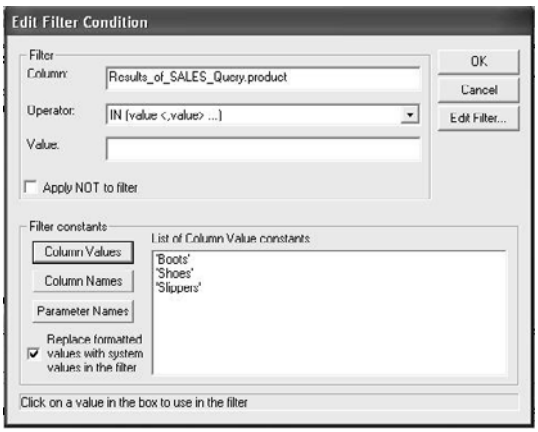
To create the query data set, click the Sales Query to activate it. Next, click the Query window under the Add Items to Project Category.

In the Query window, select the Filter Data tab. Then, select and drag the product (Product) variable from the Variables list to the open window area.

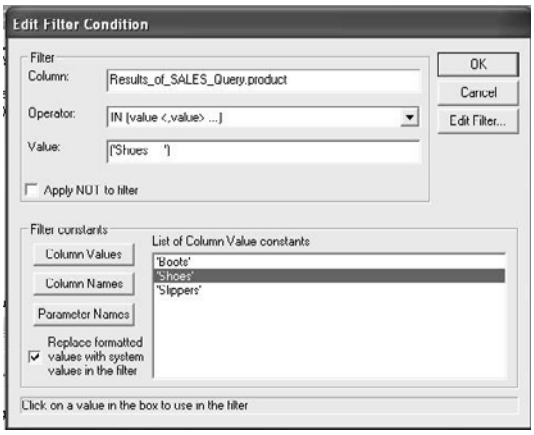
This opens the Edit Filter Condition window, where you can specify the subset condition.



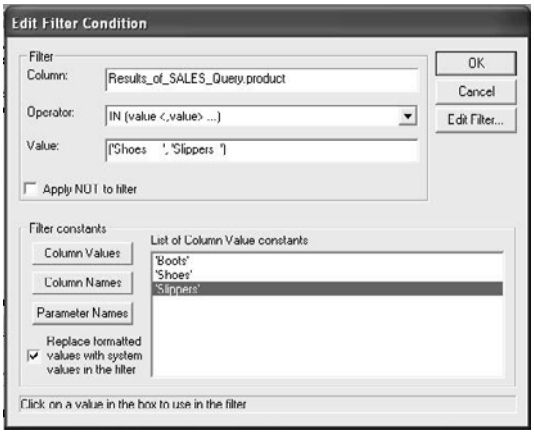
In the **Edit Filter Condition** window, click in the **Operator** field and select the **`IN (value <,value ..>)`** option to include multiple products in the query. In the Filter Constants area, click the **Column Values** button to display the values of PRODUCT.



After you select the value `Shoes` from the **List of Column Value constants**, SAS Learning Edition automatically inserts `Shoes` in the **Value** box of the **Filter** area.



After you select the value `Slippers` from the **List of Column Value constants**, SAS Learning Edition automatically inserts `Slippers` after `Shoes` in the **Value** box of the **Filter constraints** area. Next, select **OK** to close this window. This completes the subset condition for the PRODUCT variable.

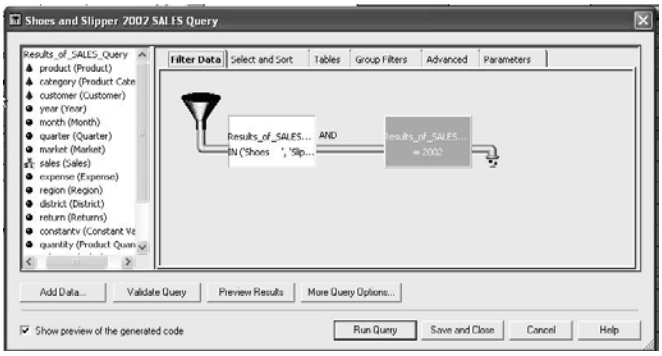


Drag and drop the YEAR variable from the left side to the right side of PRODUCT in the window to apply the following subset condition:

Select YEAR = 2002.

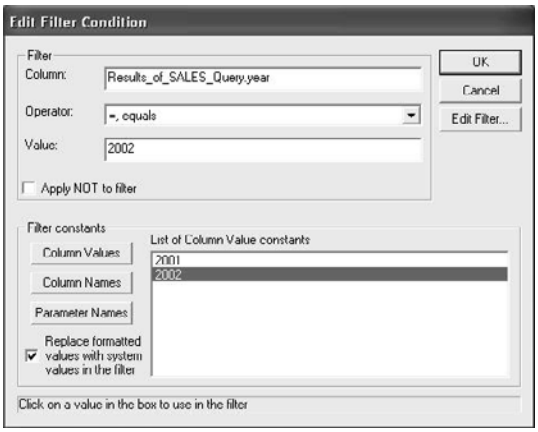
This opens another Edit Filter condition dialog box to specify subset conditions for the YEAR variable.

The condition operator AND should appear as you drag the YEAR variable to the right of PRODUCT. This is the operator you want instead of the OR operator which is specified if you positioned the YEAR variable under the PRODUCT variable. Be careful to construct your subset condition correctly.

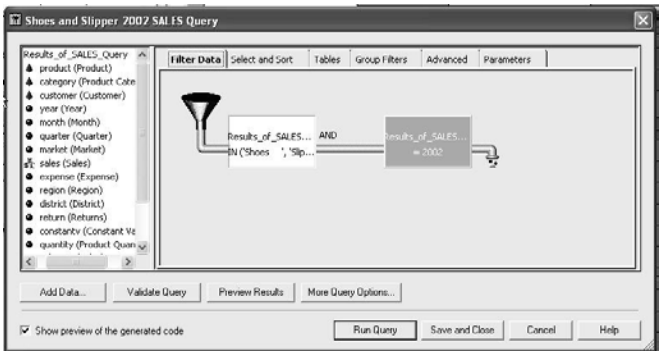


Click the Operator drop-down button to select the **`=, equals`** option. This allows only one year in the query.

Select the Column Values button to display the values of YEAR. Next, select one of the valid values for this variable. Repeat the process as before to display the values for YEAR and then select 2002 to be included in the Value window of the Filter section. Next, select OK to close the dialog box.



The final query condition for both PRODUCT and YEAR should look like the window below. The open funnel symbol on the left side of the window represents all data values going through this pipe and only the records that meet the subset conditions pass through as the query result. Note that **all** subset conditions must be met before records are included in the query result.



The table below shows the sales data that will pass the subset condition specified in this query:

	YEAR	
PRODUCT	2001	2002
Boots	Exclude Data	Exclude Data
Shoes	Exclude Data	Include Data
Slippers	Exclude Data	Include Data

Select and Sort tab

There is no need to change the default settings, which is to include all variables. If needed, you can control the variables that are saved in the query along with the sort order of selected variables.

Creating Parameter Queries

Overview

Instead of always creating a new temporary data set when changing the query condition to select another product, you can create a parameter query to re-create the temporary data set based on a different product selected from the dialog box. This is helpful for ad-hoc-type requests to query the data set.

While parameter queries can consist of multiple user selection variables such as PRODUCT, YEAR, and CUSTOMER, you will create a simple parameter query using the PRODUCT variable in this example. This allows you to select a product when you run this query instead of having the product value already specified in the query. Because of this, you have greater flexibility to create different query results. Figure 3.5 shows you the results of selecting “boots.”

In this example, follow these steps to create the query below:

- Access the **Sales Query**.
- Create a parameter query based on the PRODUCT variable.
- Subset the **Sales Query** for the specified product: Boots.

See Figure 3.6 for an example of a task using this query.

Figure 3.5 Single Product Sales Query: Boots (Partial)

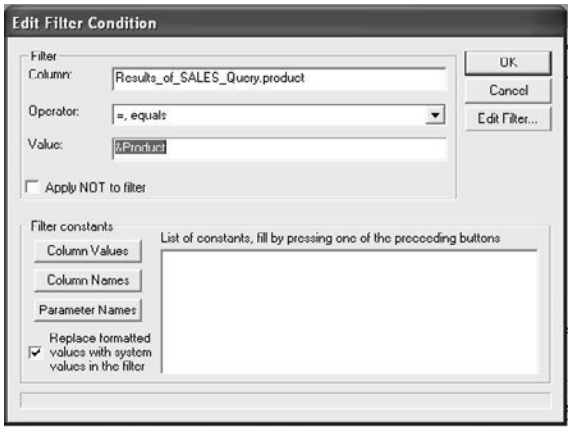
	product	category	customer	year	month	quarter	market	sales	expense	region
1	Boots	Shoes	Acme	2002	1	1	Catalog	$-1400	-980	Northern
2	Boots	Shoes	Acme	2002	2	1	Catalog	$700	490	Northern
3	Boots	Shoes	Acme	2002	3	1	Catalog	$-900	-630	Northern
4	Boots	Shoes	Acme	2002	4	2	Catalog	$1,400	980	Northern
5	Boots	Shoes	Acme	2002	5	2	Retail	$200	140	Northern
6	Boots	Shoes	Acme	2002	6	2	Retail	$1,800	1260	Northern
7	Boots	Shoes	Acme	2002	7	3	Retail	$-900	-630	Northern
8	Boots	Shoes	TwoFeet	2002	8	3	Retail	$3,500	2450	Southern
9	Boots	Shoes	Acme	2002	9	3	Retail	$3,000	2100	Northern
10	Boots	Shoes	Acme	2002	10	4	Retail	$3,000	2100	Northern
11	Boots	Shoes	BigX	2002	11	4	Retail	$3,100	2170	Western
12	Boots	Shoes	Acme	2002	12	4	Retail	$2,300	1610	Northern

Click the **Sales Query** to activate it.

Click the Query window under the Add Items to Project Category.

Filter tab

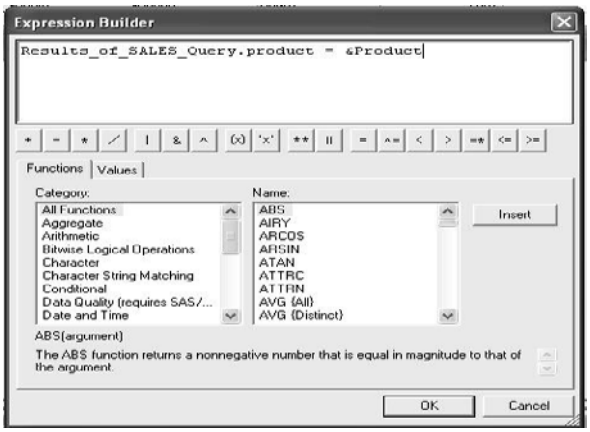
Drag and drop the PRODUCT variable from the left side to the filter data window. This opens the **Edit Filter Condition** dialog box to specify the PRODUCT variable subset condition.



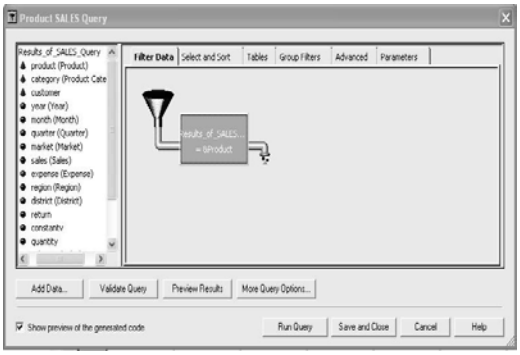
Click the **Operator** button to select the **=, equals** option. This allows only one product in the query.

Enter **`&Product`** in the Value window in the Filter section. This parameter passes the PRODUCT value selected from the user prompt when running this task. That PRODUCT value is then applied in this query condition to subset the data set. Note that this name must be the same name specified in the **Parameter** tab. Note also that **`&Product`** can be specified in titles or footnotes for tasks using this query. This automatically updates the title or footnote based on the user's selection of products.

To completely specify the query expression, select **`Edit Filter`** from the right side of the dialog box to activate the Expression Builder. Notice that the expression in the window represents the previous display.

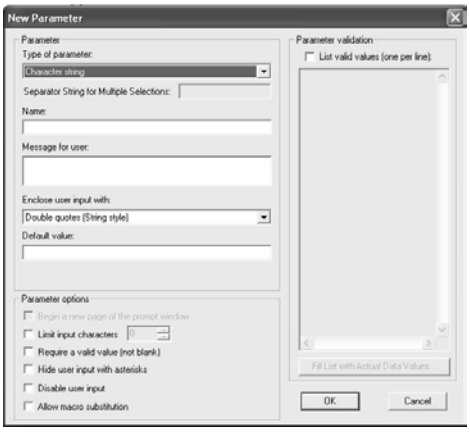


The final query condition for PRODUCT should look like this display. As discussed, the open funnel symbol represents all data values going through this pipe. Only the records that meet the subset conditions will pass through as a result of the query.



Parameters tab

Create the &PRODUCT parameter to define the query based on your selection of the product. Note that this name must be the same name specified in the Filter Data tab.

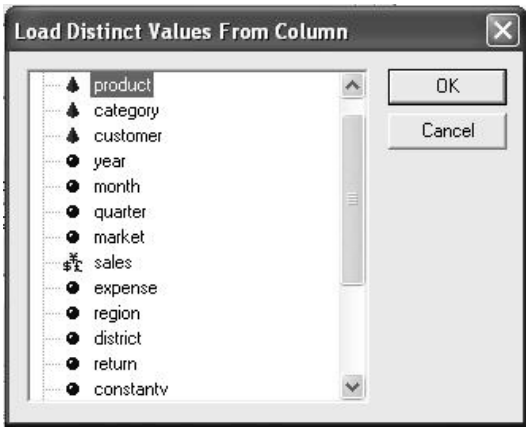


Click on the **New** button on the top right side of the dialog box. This brings up a new dialog box to specify the parameter properties.

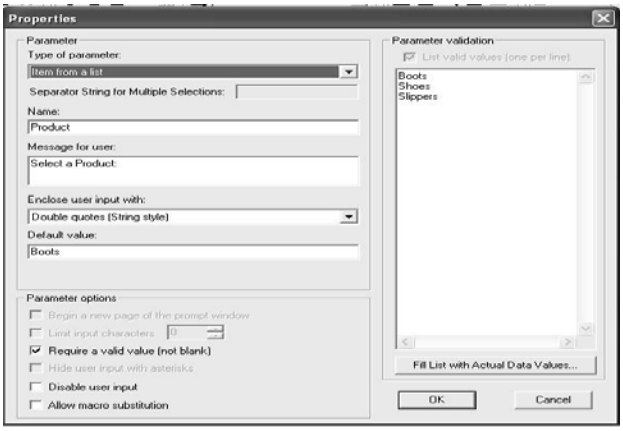
- Keep **`Item from a list`** as the type of parameter.
- Enter **`Product`** as the name of the parameter.
- Enter **`Select a Product:`** as the Message for user.
- Keep **`Double quotes (String style)`** as Enclose user input with.

You can specify a default product such as Boots in the Default value window.

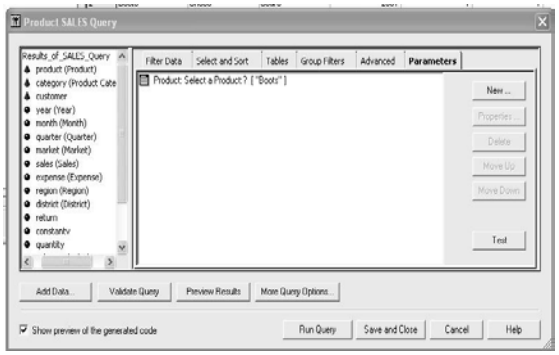
Check the List valid value option (one per line) and then the Fill list with Actual Data Values to display all valid PRODUCT values. Click **OK**.



Once all entries are made, you should see this display.



The final result should look like this display. Click **Test** to test the parameter query.



When you run this task, you are prompted with this dialog box to select a product. Although boots are selected in this example, you can select any product when running this parameter query.

Creating Summary Reports with Sales Statistics

Answer questions such as: Which products, regions, or customers had the highest sales?

Using the Summary Tables task, you can easily create complicated summary tables by basically drawing the table's shell. For example, as a business analyst, you can display descriptive sales statistics in tabular format for some or all of the columns in a table. With this report, you can get information about the top products, regions, or customers based on total profit.

In this section, you look at the overall summary sales statistics (2001 and 2002 data) by each viewpoint (product, region, customer). This report will not only give you the insight to the best products, regions, and customers, but also to the **size** of the market. You will create summary reports by product, region, and by customers with the following statistics: unit total, total sales, mean sales, and minimum sales. Programmers can especially appreciate that SAS Learning Edition automatically creates the TABULATE procedure code for these tables. As an alternative to the Summary Tables task, you can use the Summary Statistics task to obtain descriptive sales statistics.

Once the basic table is constructed, you can modify the task to create any of the other tables in this section. Note that because SAS Learning Edition is flexible, the rows and columns within the Summary Tables task are interchangeable. You can pivot any table generated from the Summary Tables task to switch between rows and columns for a different focus.

From the results in Table 3.2, you can see that for the 2001 and 2002 data, Shoes had the highest sale of $43,700 and volume close to 4,200 units.

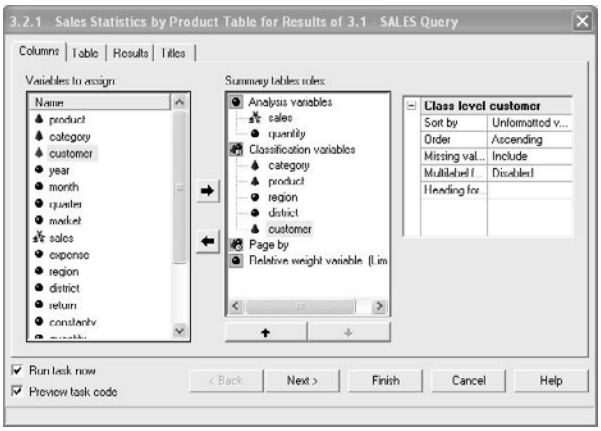
Table 3.2 Sales Statistics by Product

			Product Sales		
		Unit Totals	Dollar Totals	Average (Mean)	Min
Product Category	Product				
Shoes	Boots	5,382	$35,900	$1,496	$-1400
	Shoes	4,180	$43,700	$1,821	$100
Slippers	Slippers	8,360	$41,800	$1,742	$0
Total		17,922	$121,400	$1,686	$-1400

Click the **Sales Query** to activate it.

Click the **Summary Tables task** under the Descriptive Category.

Columns tab



Drag and drop the SALES, QUANTITY, CATEGORY, PRODUCT, REGION, DISTRICT, and CUSTOMER variables from the **Variables to assign** list on the left to the **Summary tables roles** as follows:

- Assign SALES and QUANTITY as Analysis variables.
- Assign CATEGORY, PRODUCT, REGION, DISTRICT, and CUSTOMER as Classification variables.

This step restricts the variables and classifies the **type** of variables used in this table. With the inclusion of variables REGION, DISTRICT, and CUSTOMER in the Columns tab, you have greater flexibility in creating different viewpoints of the same table. This saves you time in recreating the table layout of the variables in the Table tab.

Table tab

The Table tab has three steps:

- Drag and drop the variables and statistics to the table layout positions; discussion of this step starts on page 56.
- Specify Column Headers: Label, Font, Format; discussion of this step starts on page 57.
- Specify Column Value Formats: Label, Font, Format; discussion of this step starts on page 58.

Drag and drop variables and statistics to table layout positions

This first step can be tricky because the variables and statistics must be dropped in specific locations as represented by a line or a box in the preview window. Avoid the symbol ⃠ when locating the table layout position. The key is to place the cursor *just on*, or *just under*, the existing column or row variable. To make it easier to see all table columns, maximize this dialog box with the **Expand** button. If you make a mistake, click on the undo last action button to remove the most recent change.

Drag and drop the SALES, QUANTITY, CATEGORY, and PRODUCT variables from the **Available Variables** on the left side to the Preview window as follows:

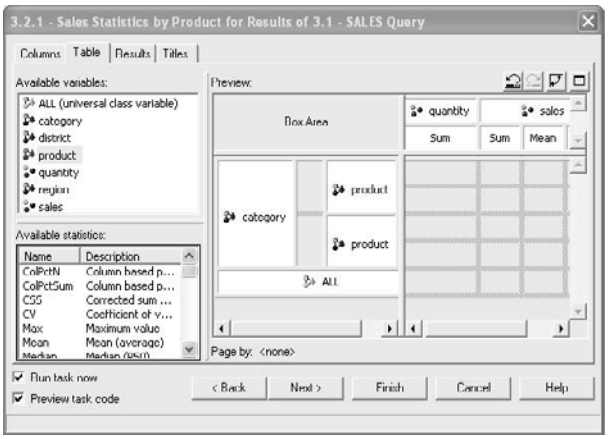
- Assign QUANTITY as the first column variable.
- Assign SALES as the second column variable *just right* of QUANTITY.
- Assign CATEGORY as the first row variable.
- Assign PRODUCT as the second nested row variable *just right* of CATEGORY.

Once the column variables are positioned, drag and drop the following statistics from the Available Statistics on the bottom left side to the Preview window on the right side as follows:

- Move SUM statistic to the column position *just under* the QUANTITY variables
- Move SUM statistics to the column position *just under* the SALES variables
- Move MEAN statistics to the column position *just under* the SALES variables and *just right* of the SUM
- Move MIN statistics to the column position *just under* the SALES variables and *just right* of the MEAN
- Move ALL keyword to the last row *just under* CATEGORY and PRODUCT. Make sure that you place the cursor between the CATEGORY and PRODUCT variables. This will calculate total sales.

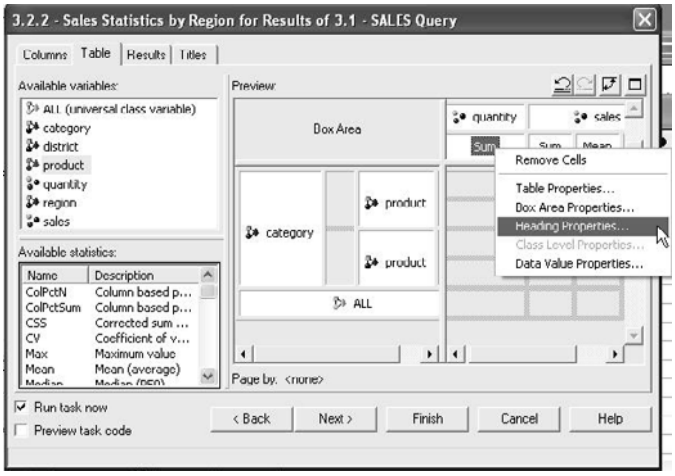
The statistics will appear as columns, and the grouping variables will appear as rows. As needed, you can change the statistics or grouping variables. In this example, the final layout position of each statistic and variable on the table is as follows:

		QUANTITY	SALES		
		SUM	SUM	MEAN	MIN
CATEGORY	PRODUCT				
	PRODUCT				
ALL					



Specify Column Headers: Label, Font, and Format

For each variable and statistic, specify the column header's label, font, and format by right-clicking the statistic and then selecting **Heading Properties**.



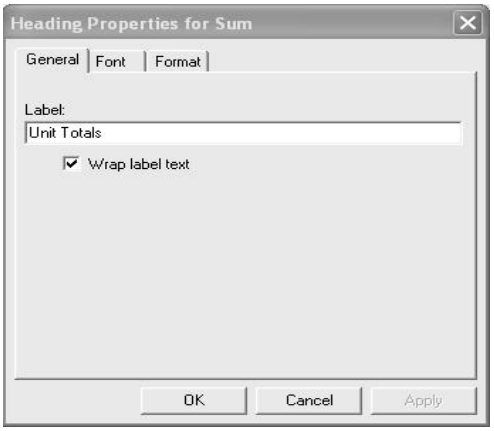
Enter **`Product Category`** for the CATEGORY column.

Enter **`Product`** for the PRODUCT column.

Enter the following labels for the listed statistics:

- Enter **`Unit Totals`** for the first SUM column.
- Enter **`Dollar Totals`** for the second SUM column.
- Enter **`Average (Mean)`** for the MEAN column.
- Enter **`Min`** for the MIN column.

The window below displays the label for the first SUM column.



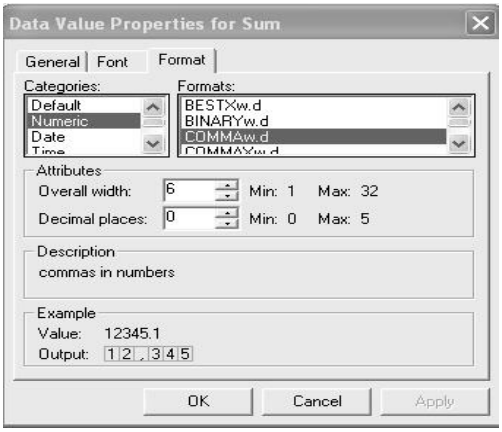
Specify Column Value Formats: Label, Font, and Format

For each statistic, specify the column value's label, font, and format by right-clicking the statistic and then selecting **Data Value Properties**.

Enter the following formats for the listed statistics:

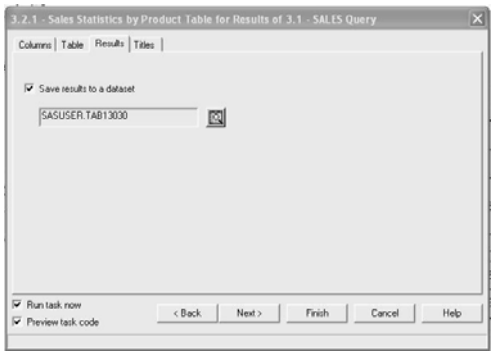
- Within the numeric category, select **`COMMAw.d`** with an overall width of 6 and decimal place of 0 for the first SUM column.
- Within the currency category, select **`DOLLARw.d`** with an overall width of 8 and decimal place of 0 for the second SUM.
- Repeat the **`DOLLARw.d`** instructions for the MEAN and MIN columns.

The window below displays the format for the first SUM column.



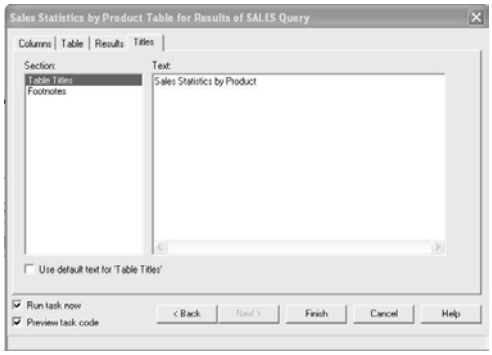
Results tab

Since you will use the data set created from this example for Table 3.5, select the **Save results to a data set** option. The data set will contain all the results from this task so that you can select this result data set for further analysis using other tasks.



Titles tab

Select **Table Titles** on the left side and enter **`Sales Statistics by Product`**. You can also specify a footnote using the Footnotes option.



This is a good time to document the results by selecting the Create Note task and then entering this text:

```
This table answers questions such as
- Which Products, Regions or Customers have the highest sales?
```

SAS Learning Edition makes it easy to switch between columns with rows to **pivot** the results. The same results or same data are displayed, just in a different layout. This structure makes it much easier to analyze all sales information by specific statistics. The pivot feature applies to any table generated from the Summary Tables task.

The only change required to pivot the table is to select the pivot icon  in the upper-right corner of the dialog box of the Table tab. This switches the table layout as follows:

Product Sales by Statistics

		Product Category			Total
		Shoes		Slippers	
		Product		Product	
		Boots	Shoes	Slippers	
Product Sales	Unit Totals	5,382	4,180	8,360	17,922
	Dollar Totals	$35,900	$43,700	$41,800	$121,400
	Average (Mean)	$1,496	$1,821	$1,742	$1,686
	Min	$-1400	$100	$0	$-1400

Different Viewpoint: Sales Statistics by Region

Once the report is created, you can easily modify the task to analyze different viewpoints of the same information. By changing the viewpoint of the same data to be grouped and sorted by region and district, you can see in Table 3.3 that the northern region has the highest sales of $91,300.

Table 3.3 Sales Statistics by Region

			Product Sales		
		Unit Totals	Dollar Totals	Average (Mean)	Min
Region	District				
Northern	NE	13,666	$91,300	$1,522	$-1400
Southern	SW	2,057	$14,700	$2,450	$1,100
Western	CW	2,199	$15,400	$2,567	$1,400
Total		17,922	$121,400	$1,686	$-1400

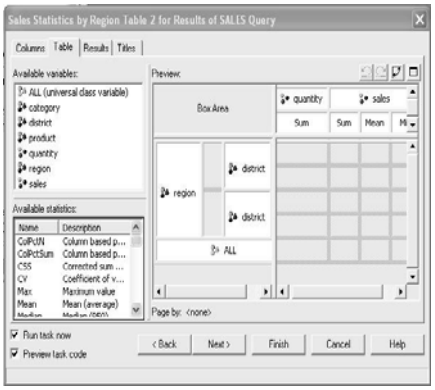
The only change required to create this table is to modify the Summary Tables task from the previous example. You can copy the task to save the original task or overwrite the current task.

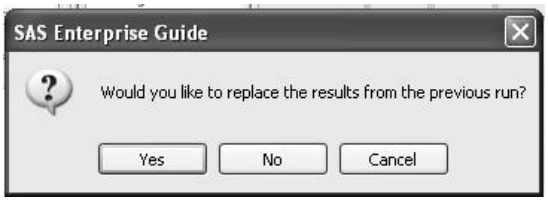
- To copy the task, right-click the task, select copy, and then paste.
- To overwrite the current task, double-click the task.

Select the **Table** tab.

- Drag and drop the variables REGION and DISTRICT from the left window to replace the table layout positions of CATEGORY and PRODUCT respectively. No other change is required.
- Click the **Yes** button to replace the results from the previous run.

Note that this becomes easily possible because the REGION and DISTRICT variables were also selected as classification variables in the previous example. If these variables were not selected, then you would need to spend time to re-create the table layout positions from scratch.





Different Viewpoint: Sales Statistics by Customer

You can modify the task for other reports. By changing the viewpoint of the same data to be grouped and sorted by customer, you can see in Table 3.4 that Acme is the best customer, with sales of $91,300.

Table 3.4 Sales Statistics by Customer

		Product Sales		
	Unit Totals	Dollar Totals	Average (Mean)	Min
Customer				
TwoFeet	2,057	$14,700	$2,450	$1,100
Acme	13,666	$91,300	$1,522	$-1400
BigX	2,199	$15,400	$2,567	$1,400
Total	17,922	$121,400	$1,686	$-1400

The only change required to create this table is to modify the Summary Tables task from the previous example. You can copy the task to save the original task or overwrite the current task.

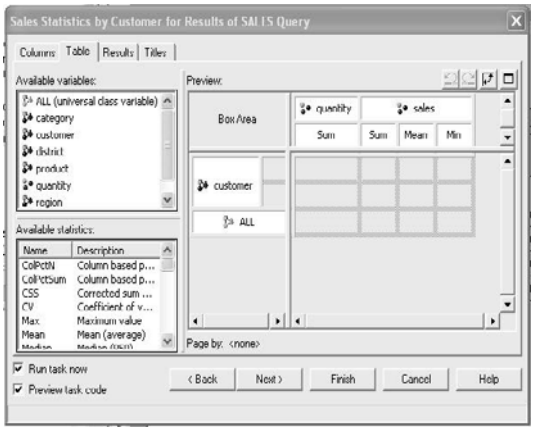
- To copy the task, right-click the task, select copy, and then paste.
- To overwrite the current task, double-click the task.

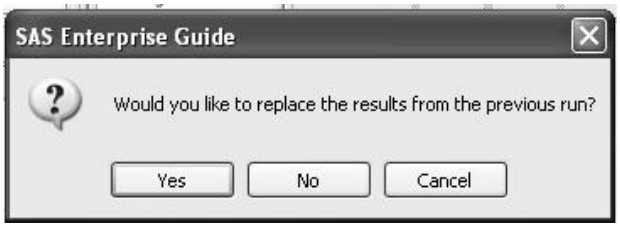
Select the **Table** tab.

- Drag and drop the variable CUSTOMER from the left window to replace the table layout position of REGION.
- Drag and drop the DISTRICT variable from the table layout to the left window to remove this variable. No other change is required.

Click **Yes** to replace the results from the previous run.

Note that this becomes easily possible because the CUSTOMER variable was also selected as a classification variable in the previous example. If this variable was not selected, then you would need to spend time to re-create the table layout positions from scratch.





Ranking Products by Sales and Volume

Answers questions such as: How did your products rank with other similar products?

To determine how your product ranks with other similar products, you can rank products by total sales **dollars** and by total sales **units**. In this example, you rank the products (Boots, Shoes, Slippers) against each other.

In this section, you analyze the sales statistics tabulated across product categories and names. This report will help you to better understand and identify which product generates the highest sales and which product generates the highest volume. You will access the summary table data set created from Table 3.2 that displays sales statistics by product. Once you access this information, run the Rank task to rank the highest sales and volume.

Depending on the rank, if for example, your product ranks third in sales dollar and ranks seventh in sales units, then your brand is more expensive than the industry average. If, however, your product ranks sixth in sales dollar and ranks fourth in sales units, then your brand is less expensive than the industry average. For your product to be about the same price as the industry average, your product would be ranked fourth in both sales dollar and sales units (Driscoll, 2003).

When you finish this section, you will have the following table.

Table 3.5 Product Rank by Total Sales and Total Volume

Product	Rank for Variable sales_Sum	sales_Sum	Rank for Variable quantity_Sum	quantity_Sum
Boots	4	35900	3	5382
Shoes	2	43700	4	4180
Slippers	3	41800	2	8360
	1	121400	1	17922

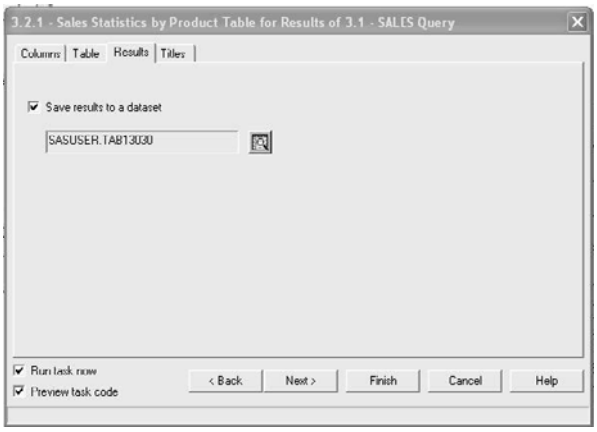
Generating this table requires the following steps:

- Select the result data set from table 3.2 to access product sales statistics.
- Run the Rank task to save the results in a data set.
- Select the result data set from the Rank task.
- Run the List Data task to display the product ranks.

Select the result data from Table 3.2 to access product sales statistics

Click the data set containing the results from **Table 3.2 Sales Statistics by Product** to activate it.

When creating Table 3.2, you selected the option to save results to a data set. This created the result data set within that task. In addition, SAS Learning Edition automatically named the data set, so all you need to do is to click the result data set to activate it.



The data set should contain all of the information in Table 3.2. Only a portion of the data set is displayed below. Note that the data set might contain extra variables that will not be used by the next task.

	category	product	_TYPE_	_PAGE_	_TABLE_	quantity_Sum	sales_Sum
1	Shoes	Boots	11	1	1	5382	35900
2	Shoes	Shoes	11	1	1	4180	43700
3	Slippers	Slippers	11	1	1	8360	41800
4			00	1	1	17922	121400

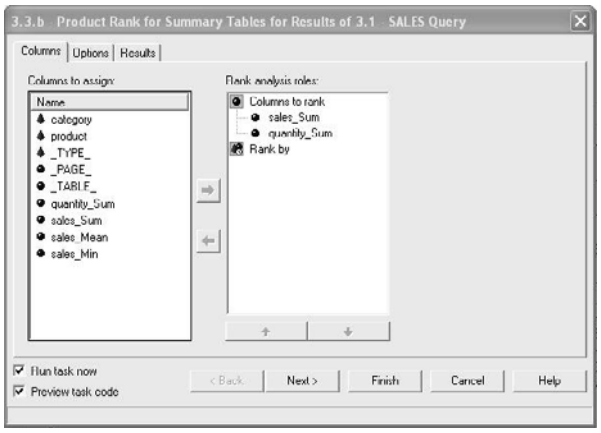
Run the Rank task to save the results in a data set

Click the Rank task in the Data Category.

Columns tab

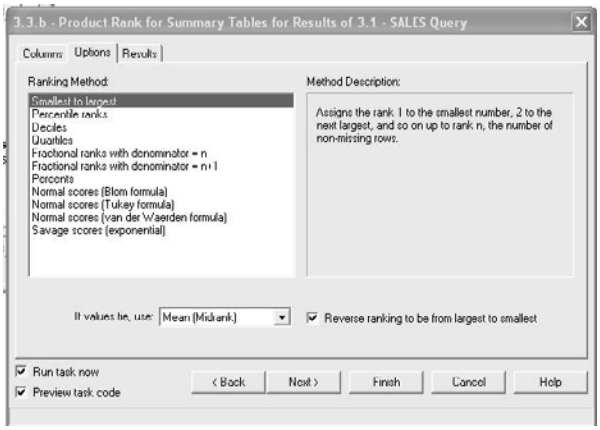
Drag and drop the SALES_SUM and QUANTITY_SUM variables from **Columns to assign** list on the left to **Rank analysis roles** as follows:

- Assign SALES_SUM and QUANTITY_SUM as **Columns to rank**.



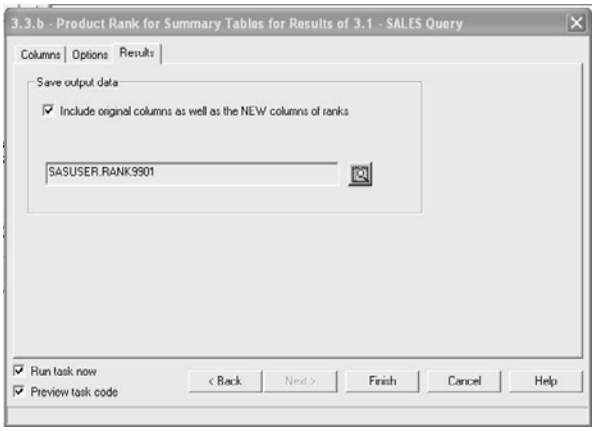
Options tab

Under Ranking Method, confirm that **Smallest to largest** is selected, unless you want to change the ranking method.



Results tab

Since you will use the data set created from this task for the next task, select the Save results to a data set option. The data set created will contain all results from this task so that you can select this result data set for further analysis using other tasks.



The result data set now contains ranked information for the two variables QUANTITY_SUM and SALES_SUM in the variables RANK_QUANTITY_SUM and RANK_SALES_SUM respectively. The range of values for these variables is 1 to 4 where the first rank can be ignored since it is associated with the total. Using these variables, you can determine how your products were ranked in volume and sales.

	category	product	_TYPE_	_PAGE_	_TABLE_	quantity_Sum	sales_Sum
1	Shoes	Boots	11	1	1	5382	35900
2	Shoes	Shoes	11	1	1	4180	43700
3	Slippers	Slippers	11	1	1	8360	41800
4			00	1	1	17922	121400

sales_Mean	sales_Min	rank_sales_Sum	rank_quantity_Sum
1495.8333333	-1400	4	3
1820.8333333	100	2	4
1741.6666667	0	3	2
1686.1111111	-1400	1	1

Select the result data set from the Rank task

Click the data set containing the results from Rank Task.

When you ran the Rank task and saved the results to a data set, SAS Learning Edition automatically named the data set. Before selecting the List Data task, you need to click this result data set to activate it.

Run the List Data task to display the product ranks

Click the List Data task under the Descriptive Category.

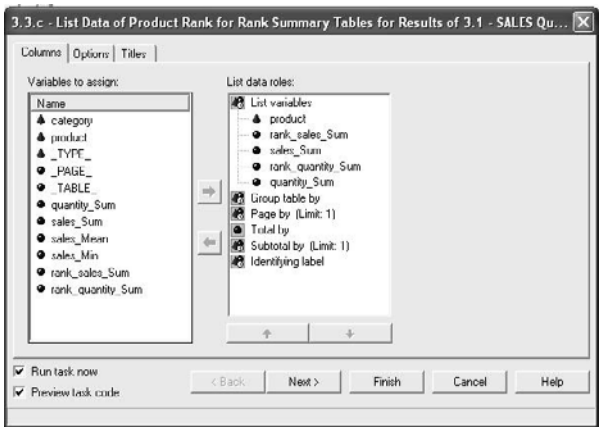
Columns tab

Drag and drop the following variables from **Variables to assign** list on the left to **List data roles** as follows:

Assign the following as List variables:

- PRODUCT
- RANK_SALES_SUM
- SALES_SUM, RANK_QUANTITY_SUM
- QUANTITY_SUM

This not only restricts the variables in the list but also specifies the sort order of the list.

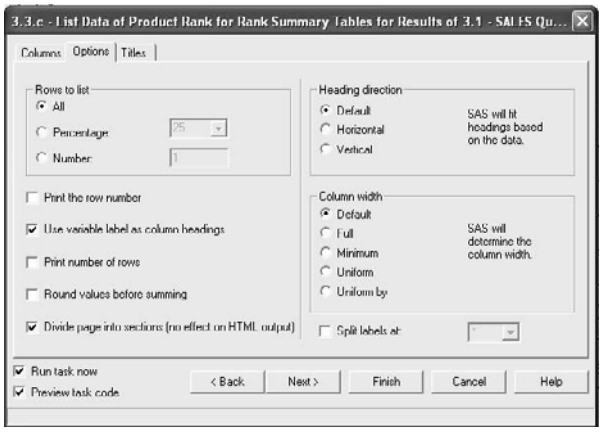


Options tab

Make the following selections to enhance the listing:

- All Rows
- Row Number
- Variable Label

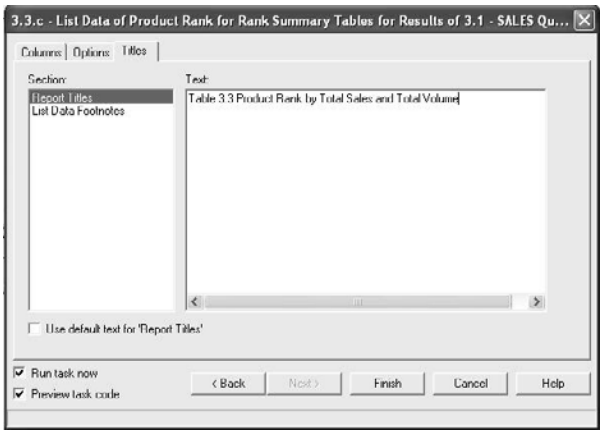
Specify other settings to change your output.



Titles tab

Select Report Titles and enter **`Table 3.3 Product Rank by Total Sales and Total Volume`** as the title in the Text field.

You can enter a footnote using the List Data Footnotes option.



The final result looks like this:

Product	Rank for Variable sales_Sum	sales_Sum	Rank for Variable quantity_Sum	quantity_Sum
Boots	4	35900	3	5382
Shoes	2	43700	4	4180
Slippers	3	41800	2	8360
	1	121400	1	17922

Creating Percent Distribution Reports

Answer questions such as:

- Which products had the largest volume?
- What was the sales percentage distribution by market segment?

Using the Summary Tables task, you can include percentages in tabular format to get frequency distribution of any variable. This becomes useful to analyze the volume of products shipped from inventory to customer sites. With this report, you not only get information about product volume, but you can also identify the strongest market segments by product or by region. The frequency values are tabulated across each product so that the sum of % catalog and % retail values equal 100%.

From the results of Table 3.6, you can see that for 2001 and 2002 data, slippers had the largest volume of 8,360 (1,660 + 6,700), and retail sales had the strongest market segment with 80% of the market. Retail sales is the dominant market segment for your products.

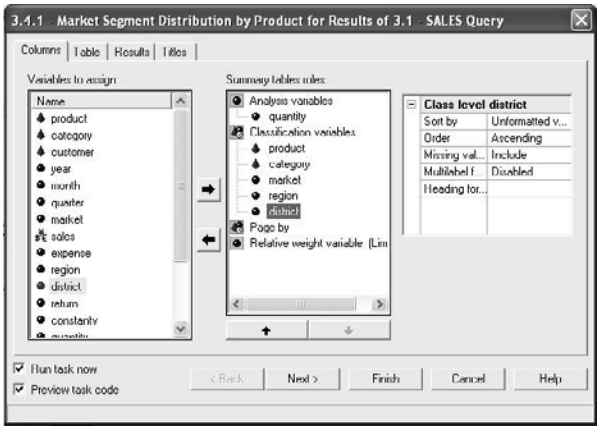
Table 3.6 Market Segment Distribution by Product

		Market Segment			
		Catalog		Retail	
		N	%	N	%
Product Category	Product				
Shoes	Boots	1,186	22	4,196	78
	Shoes	810	19	3,370	81
Slippers	Slippers	1,660	20	6,700	80
Total		3,656	20	14,266	80

Click the Sales Query to activate it.

Click the Summary Tables Task under the Descriptive Category.

Columns tab



Drag and drop the QUANTITY, PRODUCT, CATEGORY, MARKET, REGION, and DISTRICT variables from **Variables to assign** on the left to **Summary tables roles** on the right as follows:

- Assign QUANTITY as Analysis variables.
- Assign PRODUCT, CATEGORY, MARKET, REGION, and DISTRICT as Classification variables.

This step not only restricts the variables, but also classifies the type of variables used in this table. Note that by including variables REGION and DISTRICT in the Columns tab, you have greater flexibility in creating different viewpoints of the same table. This saves you time in re-creating the table layout of the variables in the Table tab.

Table tab

The Table tab requires three steps:

- Drag and drop variables and statistics to table layout positions; discussion of this step starts on page 69.
- Specify Column Headers: Label, Font, Format; discussion of this step starts on page 71.
- Specify Column Value Formats: Label, Font, Format; discussion of this step starts on page 72.

Drag and drop variables and statistics to table layout positions

As discussed, this first step might be a little tricky because the variables and statistics must be dropped in specific locations as represented by a line or a box in the preview window. Remember to place the cursor *just* to the *right* of, or *just under*, the existing column or row variable.

Drag and drop the QUANTITY, PRODUCT, CATEGORY, and MARKET variables from the **Available Variables** on the left side to the Preview window on the right side as follows:

- Assign MARKET as the first column variable.
- Assign QUANTITY as the second nested column variable under MARKET.
- Assign CATEGORY as the first row variable.
- Assign PRODUCT as the second nested row variable *just right* of CATEGORY.

When the column variables are positioned, drag and drop the following statistics from the **Available Statistics** on the bottom left to the Preview window on the right as follows:

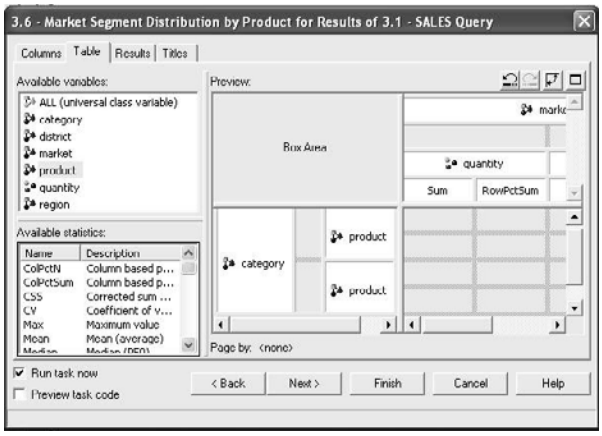
- Move SUM and ROWPCTSUM statistics to the column positions under the first QUANTITY variable.

  Note that this automatically creates the SUM and ROWPCTSUM statistics in the other column positions. The ROWPCTSUM statistics feature is used to calculate row percentages across columns. Use the COLPCTSUM statistics feature to calculate column percentages across rows. See the Glossary for a brief description of selected statistics within the Summary Tables Task.

- Move the ALL keyword to the last row *just under* CATEGORY and PRODUCT. Be sure that you place the cursor between the CATEGORY and PRODUCT variables. This calculates total sales.

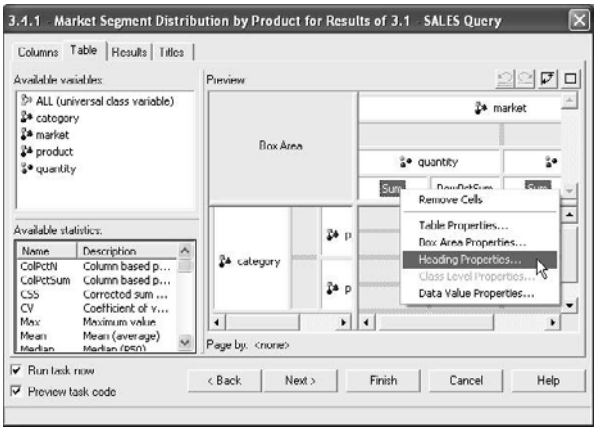
The statistics will appear as columns, and the grouping variables will appear as rows. As discussed, you can change the statistics or grouping variables. In this example, the final layout position of each statistic and variable on the table is as follows:

		MARKET			
		QUANTITY		QUANTITY	
		SUM	ROWPCTSUM	SUM	ROWPCTSUM
CATEGORY	PRODUCT				
	PRODUCT				
ALL					



Specify Column Headers: Label, Font, Format

For each variable and statistic, specify the column header's label, font, and format by right-clicking the statistic and then select **Heading Properties**.



Enter **`Product Category`** for the first column.

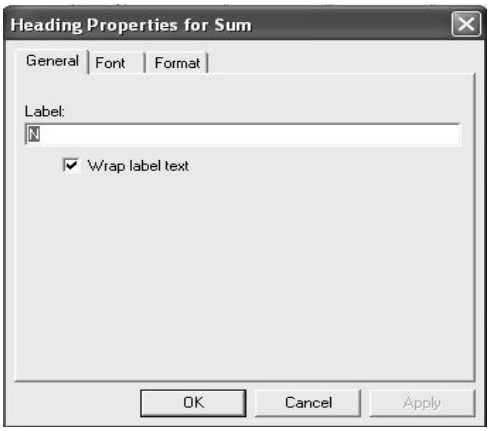
Enter **`Product`** for the second column.

Enter **`N`** for the SUM columns.

Enter **`%`** for the ROWPCTSUM columns.

Enter **`Market Segment`** for the MARKET column.

This window displays the label for the first SUM column.



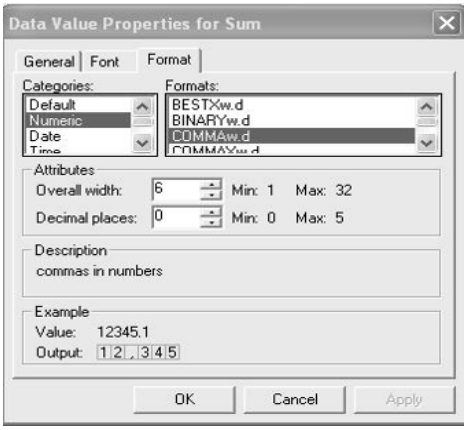
Specify Column Value Formats: Label, Font, Format

For each statistic, specify the column value's label, font, and format by right-clicking the statistic and then selecting **Data Value Properties**.

Enter the following formats for the listed statistics:

- In the numeric category, select **COMMAw.d** with an overall width of 6 and decimal place of 0 for both SUM columns. Since all SUM columns need to be consistent, SAS Learning Edition automatically selects all SUM columns when specifying column attributes.
- In the numeric category, select **BESTw.d** with an overall width of 3 and decimal place of 0 for both ROWPCTSUM columns. Since all ROWPCTSUM columns need to be consistent, SAS Learning Edition automatically selects all ROWPCTSUM columns when specifying column attributes.

This window displays the format for the first SUM column.



Titles tab

Select **Table Titles** on the left and enter **`Table 3.7 Market Segment Distribution by Product`** as the table title. You can also specify a footnote.

Different Viewpoint: Market Segment Distribution by Region

Once the report is created, you can easily modify the task to analyze different viewpoints of the same information. By changing the viewpoint of the same data to be sorted by region and district, you can see in Table 3.7 that all regions accounted for retail sales but only the northern region had catalog sales. There were no catalog sales in the southern and western regions. Note that SAS Learning Edition uses dots to represent missing values.

Table 3.7 Market Segment Distribution by Region

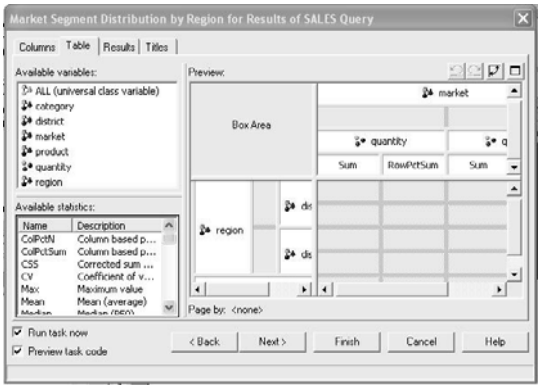
		Market Segment			
		Catalog		Retail	
		N	%	N	%
Region	District				
Northern	NE	3,656	27	10,010	73
Southern	SW	.	.	2,057	100
Western	CW	.	.	2,199	100
Total		3,656	20	14,266	80

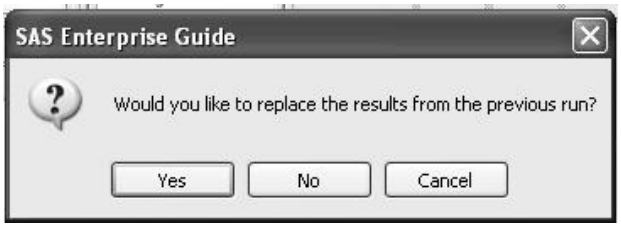
The only change required to create this table is to modify the Summary Tables task from the previous example. You can copy the task to save the original task or overwrite the current task:

- To copy the task, right-click the task and then select copy and then paste.
- To overwrite the current task, double-click the task.

Select the **Table** tab**.**

- Drag and drop the variables REGION and DISTRICT from the left window to replace the table layout positions of CATEGORY and PRODUCT respectively. No other change is required.
- Click the **Yes** button to replace the results from the previous run.





To answer the question of what was the sales percent from the previous year, you can further modify the task to identify the breakdown of the percent distribution by year and product category.

Answer questions such as: What is the sales percent distribution by product category and market segment across time periods?

Table 3.8 shows that overall slipper sales are strong considering it's the only product in its category in contrast to the shoe category; the shoe category has shoes and boots.

Table 3.8 Market Segment Distribution by Product Category and Year

		Product Category							
		Shoes				Slippers			
		Market Segment				Market Segment			
		Catalog		Retail		Catalog		Retail	
Year	Quarter	Sum	%	Sum	%	Sum	%	Sum	%
2001	1 ❹	550	54	.	.	460	46	.	.
	2	447	33	527	38	200	15	200	15
	3 ❺	.	.	1,677	60	.	.	1,140	40
	4 ❺	.	.	1,679	46	.	.	1,940	54
2002	1	614	41	.	.	900	59	.	.
	2 ❷	385	25	530	34	100	6.4	540	35
	3 ❸	.	.	1,544	57	.	.	1,160	43
	4 ❸	.	.	1,609	48	.	.	1,720	52
Total ❶		1,996	11	7,566	42	1,660	9.3	6,700	37

You can make the following conclusions from reviewing Table 3.8:

❶ Overall retail sales (42% + 37%) are stronger than catalog sales (11% + 9%).

❷ For year 2002, product sales for the second half (6,033 units – 1,544 + 1,160 + 1,609 + 1,720) of the year are stronger than the first half (3,069 units – 614 + 385 + 530 + 900 + 100 + 540) of the year.

❸ Within 2002, the third quarter shoe sales are 57% and fourth quarter slipper sales are 52%.

❹ For year 2001, product sales for the second half (6,436 units – 1,677 + 1,679 + 1,140 + 1,940) of the year are stronger than the first half (2,384 units – 550 + 447 + 527 + 460 + 200 + 200) of the year.

❺ Within 2001, the third-quarter shoe sales are 60% and fourth quarter slipper sales are 54%.

Click the **Sales Query** to activate it.

Click the Summary Tables Task under the Descriptive Category.

Columns tab

Drag and drop the QUANTITY, PRODUCT, CATEGORY, MARKET, REGION, DISTRICT, YEAR, and QUARTER variables from the **Variables to assign** list to the **Summary tables roles** as follows:

- Assign QUANTITY as Analysis variables.
- Assign PRODUCT, CATEGORY, MARKET, REGION, DISTRICT, YEAR, and QUARTER as Classification variables.

This step not only restricts the variables, but also classifies the type of variables used in this table.

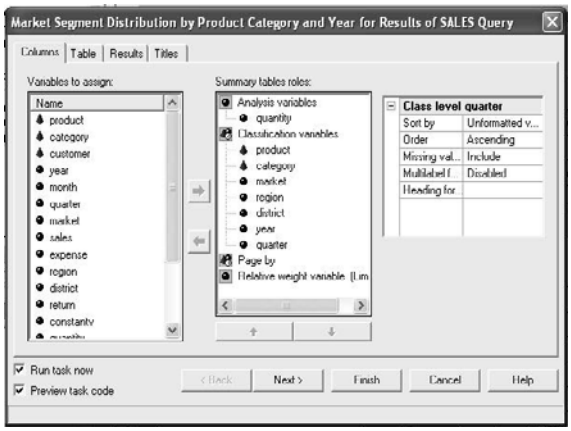


Table tab

The Table tab has three main steps:

- Drag and drop variables and statistics to table layout positions; discussion of this step starts on page 76.
- Specify Column Headers: Label, Font, Format; discussion of this step starts on page 77.
- Specify Column Value Formats: Label, Font, Format; discussion of this step starts on page 78.

Drag and drop variables and statistics to table layout properties

As discussed, this first step can be tricky because the variables and statistics must be dropped in specific locations as represented by a line or a box in the preview window. Place the cursor to the right of, or just under, the existing column or row variable.

Drag and drop the CATEGORY, MARKET, QUANTITY, YEAR, and QUARTER variables from **Available Variables** to the **Preview** window as follows:

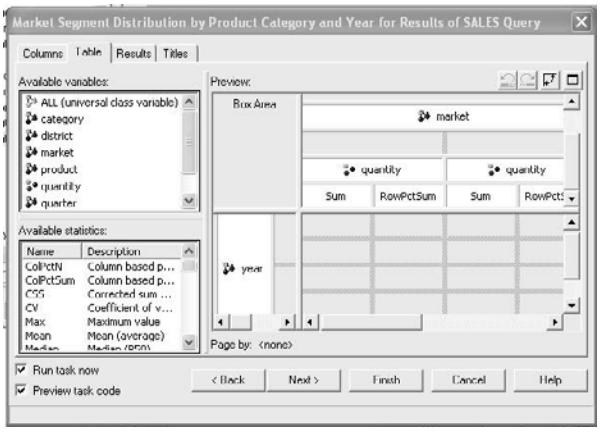
- Assign CATEGORY as the first column variable.
- Assign MARKET as the second nested column variable *just under* CATEGORY.
- Assign QUANTITY as the third nested column variable *just under* MARKET.
- Assign YEAR as the first row variable.
- Assign QUARTER as the second nested row variable *just to the right of* YEAR.

Once the column variables are positioned, drag and drop the following statistics from **Available Statistics** to the **Preview** window:

- Move SUM and ROWPCTSUM statistics to the column positions just under the first QUANTITY variables. Note that this action automatically creates the SUM and ROWPCTSUM statistics in the other corresponding column positions. The ROWPCTSUM statistics is used to calculate row percentages across columns. As discussed, see the glossary for a brief description of each statistics within the Summary Tables Task.
- Move the ALL keyword to the last row just under YEAR and QUARTER. Be sure that you place the cursor between the YEAR and QUARTER variables. This action calculates total sales.

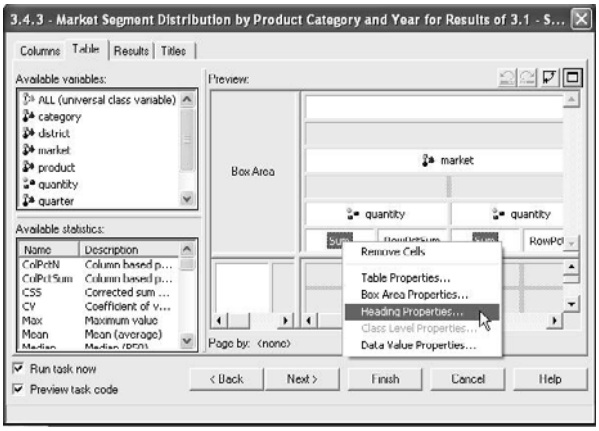
The statistics will appear as columns, and the grouping variables will appear as rows. As discussed, you can change the statistics or grouping variables. In this example, the final layout position of each statistic and variable on the table is as follows:

		CATEGORY			
		MARKET			
		QUANTITY		QUANTITY	
		SUM	ROWPCTSUM	SUM	ROWPCTSUM
YEAR	QUARTER				
	QUARTER				
ALL					



Specify Column Headers: Label, Font, Format

Specify the column header's label, font, and format for each variable and statistic by right-clicking the statistic and then selecting **Heading Properties**.



Enter **`Year`** for the YEAR column.

Enter **`Quarter`** for the QUARTER column.

Enter **`Sum`** for the SUM columns.

Enter **`%`** for the ROWPCTSUM columns.

Enter **`Market Segment`** for the MARKET column.

Enter **Product Category** for the CATEGORY column.

The window below displays the label for the first SUM column.

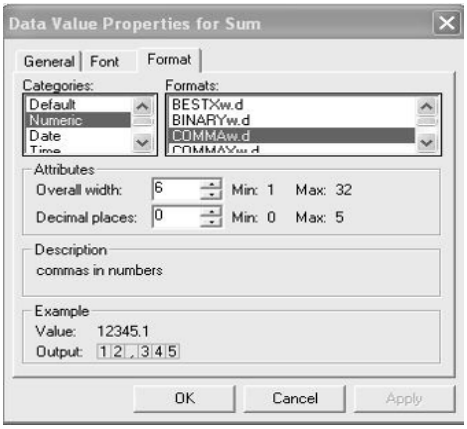
Specify Column Value Formats: Label, Font, Format

Specify the column value's label, font, and format for each statistic by right-clicking the statistic and then selecting **Data Value Properties**.

Enter the following formats for the listed statistics:

- In the numeric category, select **COMMAw.d** with an overall width of 6 and decimal place of 0 for all SUM columns. Since all SUM columns need to be consistent, SAS Learning Edition automatically selects all SUM columns when specifying column attributes. See the figure below.
- In the numeric category, select **BESTw.d** with an overall width of 3 and decimal place of 0 for both ROWPCTSUM columns. Since all ROWPCTSUM columns need to be consistent, SAS Learning Edition automatically selects all ROWPCTSUM columns when specifying column attributes.

The window below displays the format for the first SUM column.



Titles tab

Select Table Titles on the left and enter **`Table 3.8 Market Segment Distribution by Product Category and Year`**. You can also specify a footnote if needed.

Different Viewpoint: Total Sales by Market Segment

Instead of looking at the percent of market segment, you can also look at total sales by market segment. From Table 3.9, you can see that retail sales for shoes are causing the growth. Again, you can see that in both fourth quarters, shoe category sales ($14,900 and $15,900) are stronger than slipper category sales ($9,700 and $8,600). This is because the shoe category has sales of both shoes and boots. You might want to analyze the **sales per unit** to account for this condition.

Table 3.9 Market Segment Sales by Product Category and Year

		Product Category							
		Shoes				Slippers			
		Market Segment				Market Segment			
		Catalog		Retail		Catalog		Retail	
		N	Sales	N	Sales	N	Sales	N	Sales
		Sum	Sum	Sum	Sum	Sum	Sum	Sum	Sum
Year	Quarter								
2001	1	550	$4,600	.	.	460	$2,300	.	.
	2	447	$4,100	527	$4,100	200	$1,000	200	$1,000
	3	.	.	1,677	$15,100	.	.	1,140	$5,700
	4	.	.	1,679	$14,900	.	.	1,940	$9,700
2002	1	614	$800	.	.	900	$4,500	.	.
	2	385	$3,500	530	$4,800	100	$500	540	$2,700
	3	.	.	1,544	$11,800	.	.	1,160	$5,800
	4	.	.	1,609	$15,900	.	.	1,720	$8,600
Total		1,996	$13,000	7,566	$66,600	1,660	$8,300	6,700	$33,500

The only change required to create this table is to modify the Summary Tables task from the previous example. You can copy the task to save the original task or overwrite the current task:

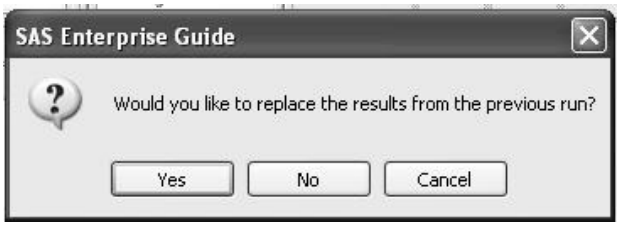
- To copy the task, right-click the task, select copy, and then paste.
- To overwrite the current task, double-click the task.

Select the **Table** tab.

- Drag and drop the variable SALES from the left window to just right of the QUANTITY variable in the table layout.
- Drag and drop the SUM statistics from the bottom-left window to replace the table layout position of ROWPCTSUM. No other change is required.
- Click the **Yes** button to replace the results from the previous run.

The statistics will appear as columns, and the grouping variables will appear as rows. As discussed, you can change the statistics or grouping variables. In this example, the final layout position of each statistic and variable on the table is as follows:

		CATEGORY			
		MARKET		MARKET	
		QUANTITY	SALES	QUANTITY	SALES
		SUM	SUM	SUM	SUM
YEAR	QUARTER				
	QUARTER				
ALL					



Grouping Data in a List Report

Answer questions such as: How do you identify monthly transactions of products that are grouped by product sales of low, average, and high?

Using the List Data task, you can list variables in any order and by any group. With this list, you can identify the monthly transactions of products that are grouped by product sales of low, average, and high.

For example, you can use the SALESGRP variable to group boot sales into low, average, and high sales categories and then display the detail monthly transactions. See the first part of this chapter to see how to create the SALESGRP variable and SALEGRP. format.

From the results in Figure 3.6, you can see that Southern and Western regions had high boot sales while all of the low sales were in the Northern region.

Figure 3.6 Listing of Boot Sales Grouped by Sales Category

Sales = Less than Zero Sales

Obs	Quarter	Month	Region	District	Market	Customer	Product Quantity	Sales
1	1	1	Northern	NE	Catalog	Acme	175	$-1400
2	1	3	Northern	NE	Catalog	Acme	112	$-900
3	3	7	Northern	NE	Retail	Acme	112	$-900
4	2	5	Northern	NE	Retail	Acme	50	$-400

Sales = Average Sales

Obs	Quarter	Month	Region	District	Market	Customer	Product Quantity	Sales
5	1	3	Northern	NE	Catalog	Acme	0	$0
6	2	5	Northern	NE	Retail	Acme	25	$200
7	1	2	Northern	NE	Catalog	Acme	87	$700
8	2	6	Northern	NE	Retail	Acme	137	$1,100
9	1	2	Northern	NE	Catalog	Acme	175	$1,400
10	2	4	Northern	NE	Catalog	Acme	175	$1,400
11	2	4	Northern	NE	Catalog	Acme	187	$1,500
12	3	7	Northern	NE	Retail	Acme	200	$1,600
13	2	6	Northern	NE	Retail	Acme	225	$1,800

(*continued on next page*)

Figure 3.6 (*continued*)

Obs	Quarter	Month	Region	District	Market	Customer	Product Quantity	Sales
14	4	12	Northern	NE	Retail	Acme	237	$1,900
15	1	1	Northern	NE	Catalog	Acme	275	$2,200
16	3	9	Northern	NE	Retail	Acme	287	$2,300
17	4	12	Northern	NE	Retail	Acme	287	$2,300
18	4	11	Western	CW	Retail	BigX	312	$2,500

Sales = High Sales

Obs	Quarter	Month	Region	District	Market	Customer	Product Quantity	Sales
19	3	8	Southern	SW	Retail	TwoFeet	350	$2,800
20	3	9	Northern	NE	Retail	Acme	375	$3,000
21	4	10	Northern	NE	Retail	Acme	375	$3,000
22	4	11	Western	CW	Retail	BigX	387	$3,100
23	4	10	Northern	NE	Retail	Acme	400	$3,200
24	3	8	Southern	SW	Retail	TwoFeet	437	$3,500

Click the **Single Product Sales Query** to activate it. If you need to rerun the parameter query, be sure to select "boots" as the product.

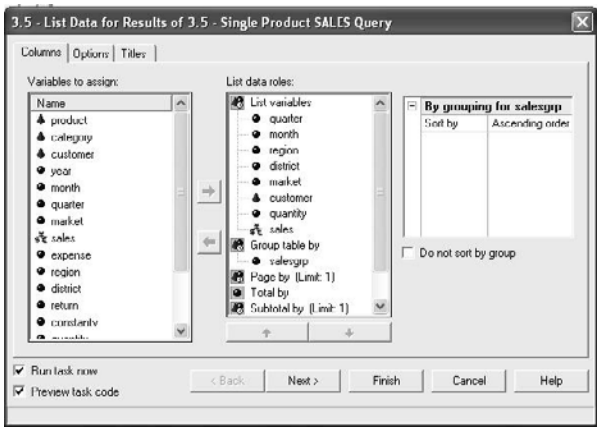
Click the **List Data Task** under the Descriptive Category.

Columns tab

Drag and drop the following variables from the **Variables to assign** list to **List data roles** as follows:

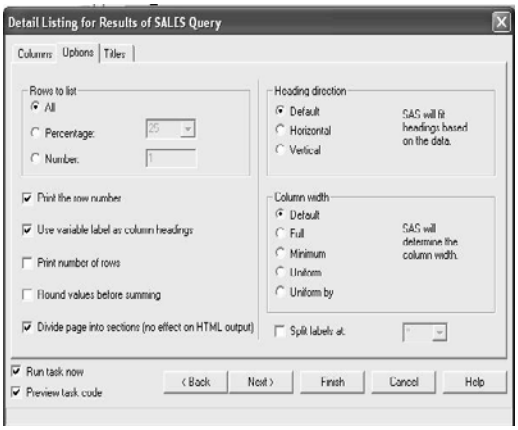
- Assign the following as List variables: QUARTER, MONTH, REGION, DISTRICT, MARKET, CUSTOMER, QUANTITY, SALES.
- Assign SALEGRP as the Group Table by variable. This is the variable created in the first part of Chapter 3.

This step not only restricts the variables in the list but also specifies the sort and group order of the list.



Options tab

Make the following selections—All Rows, Row Number, and Variable Label—to enhance the listing. You can specify other settings to change your output.



Titles tab

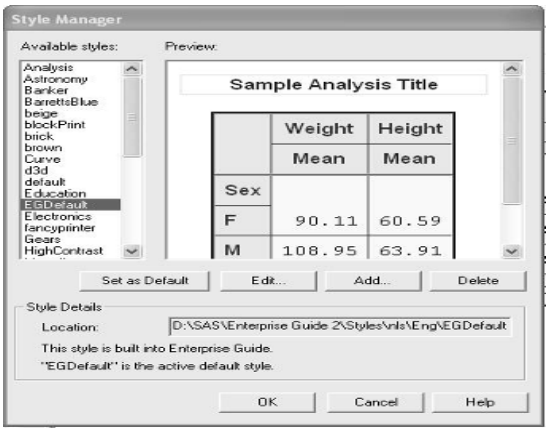
Select Report Titles on the left side and enter **`List 3.6 Listing of Boot Sales Grouped by Sales Category`** as the list title.

You can also specify a footnote if needed.

Customizing Reports and Graphs with Style Editor

When creating tables and charts, SAS Learning Edition uses a style type that specifies all of the attributes of the output such as color, font, size, and justification. As an alternative to the EGDefault style, you can select from more than 35 predefined SAS-supplied styles as shown in Figure 3.7. Because styles are based on cascading style sheets, only the HTML file type recognizes the new style type. Note that the contents of the results will not change. You can change the style type by selecting `Tools` ► `Style Manager` from the toolbar. Once you change the default style, all subsequent results use the new style type.

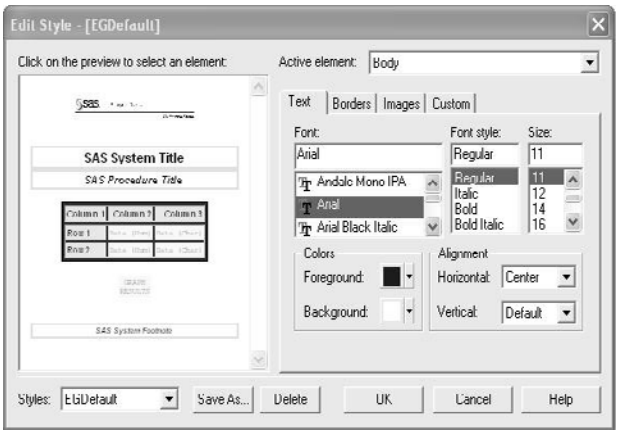
Figure 3.7 Style Manager



Not only can you enhance your reports with predefined styles, SAS Learning Edition enables you to create your own style easily using the style editor shown in Figure 3.8.

It is beyond the scope of this book to explain style elements or style templates.

Figure 3.8 Style Editor



Select the **Edit** button on the Style Manager dialog box to access this user-friendly interface. This window allows you to customize style attributes such as color, font, and size for specific sections of your output. Remember that by using the style editor, you are limited to customizing your HTML files and no other file type such as PDF.

Chapter 4

IN THIS CHAPTER

Creating Graphs for Effective Communication

Along with creating reports, you can use SAS Learning Edition to create graphs for effective communication.

Figure 4.1 Horizontal Bar Chart

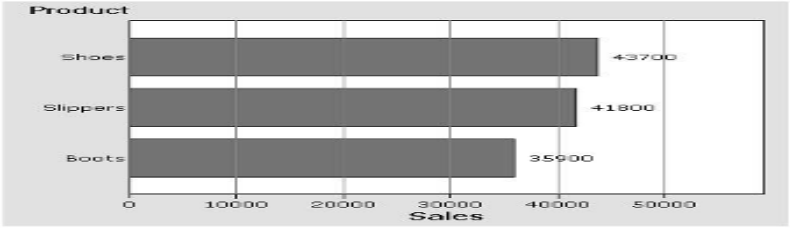
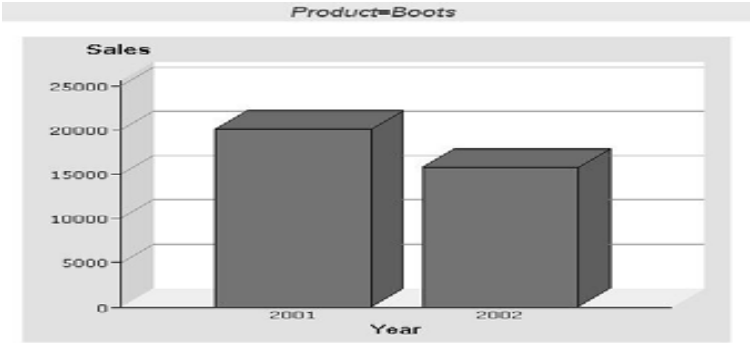


Figure 4.2 Vertical Bar Chart



Overview

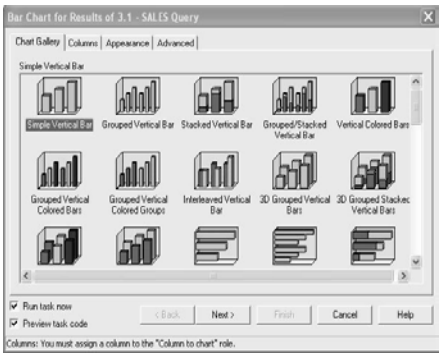
Powerful graphs are simple, consistent, and straightforward in the message conveyed. Using the Finance task, for example, you can display descriptive statistics as **charts** for any numeric variable. SAS Learning Edition is powerful enough to give you a high degree of control over your chart without needing even minor customization. Programmers and end users will appreciate the ability to create great graphs without having to remember each SAS statement's syntax.

Many examples in this chapter are created from the standard task options. It is beyond the scope of this book to show all of the numerous options available when creating graphs. The charts in this chapter answer business-related questions listed in Chapter 2.

Creating charts involves selecting the data set and then selecting the specified SAS Enterprise Guide task, variables, and options. For options not available through the task, you can modify the generated code to create the customized chart. (See the three-part series of articles on graphs by Betsy Corning in *Observations* for more information on creating and customizing a variety of charts). Unless indicated, the charts are created from the Sales Query.

As discussed in Chapter 2, it generally is easy—once you create a chart—to modify the task to graph the results by any of the viewpoints: customer, product, or organization. Because SAS Learning Edition is flexible, the vertical and horizontal variables within the Graphs task are interchangeable. You can pivot any graph created in this chapter to switch between the vertical and horizontal axes for a different focus.

When you select a graph in Tasks by Category, you can then select from a variety of chart types in the Chart Gallery.



The section below lists three general rules for creating effective graphs:

1. State the purpose of the chart.

2. Label each axis and include a legend, if needed.

3. Choose a chart type that is suitable for the data and the comparison requested as follows:

If you want to do the following	Use this type of chart
Show percent of contribution to the whole. Use a pie chart to compare percentages.	Pie
Show movement or distance. The y axis allows for more categories than using the vertical bars.	Bar
Read from left to right to review a trend over time or show magnitude of difference between groups. This is useful for comparing categories of data over time.	Vertical bar
Show relationship between two quantitative variables, and emphasize relative position of data more than the actual value.	Scatter plot

SAS Learning Edition uses just one task to create pie, horizontal bar, vertical bar, scatter plot, and box plot charts even though they appear to be separate tasks in the Tasks by Category section. When you select any graph task, you are selecting the graph type from the chart gallery. In addition, the available options depend on the graph type selected. Note also that the icons for horizontal and vertical bars are the same because they are considered to be bar charts.

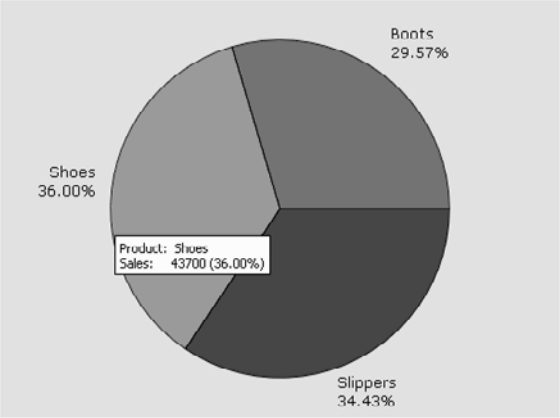
Selecting Your Output Style, Color, and Layout

A benefit of creating graphs from SAS Learning Edition is that all of the presentation details for making graphs legible, colorful, and useful are automatically controlled and specified. You do not have to know the details of the SAS/GRAPH procedures to get the results you want. Using SAS Learning Edition to create powerful, effective charts to answer business related questions is a worry-free process.

For business analysts who want to specify the details of the chart, there are options available on some tabs of the task. For example, selecting the Appearance tab in the Bar task enables you to customize the following sections of the chart: Bars, Layout, Titles, Axes, Ticks, Reference Lines, and Chart Area. The examples in this book illustrate a selection of the customizations you can implement in your projects, but you can learn more about the flexibility of SAS Learning Edition once you learn the basics. Chapter 5 discusses options to customize your charts by changing the style type.

As shown in Figure 4.3, there is an added benefit for selected charts. You can see the data values used to create the chart or details of the chart by placing your cursor over any section of the chart. For example, you can see that for Figure 4.3, total shoe sales were $43,700. This gives you the insight information needed to help explain the chart.

Figure 4.3 Display Data Value in a Chart



As shown in Figure 1.7 in Chapter 1, make sure the Graph Image format is set to ActiveX to enable this feature. This setting instructs SAS Learning Edition to create charts as ActiveX objects. This allows other users to change the graph without having SAS Learning Edition installed on their computer. For example, users can change vertical bars to horizontal bars by selecting a different chart type once by right-clicking the graph.

Finally, depending on the settings, you can copy a graph by right-clicking the graph, and then copying and pasting it directly to Microsoft Word or PowerPoint. As an alternative, you can save the graph as a JPEG file and import it into Word or PowerPoint.

Comparing Percent Distribution Using Pie Charts

Answer questions such as: Which products accounted for the largest share of total company sales?

One method to analyze percent distribution is with pie charts. Pie charts provide a nice simple visual representation by the size of each slice. You can then easily identify the largest and smallest slice because the pie chart shows all sales contributing to the whole.

As shown in Figure 4.3 (2001 and 2002 data), shoes accounted for the largest share of total company sales with 36% while boots had the lowest at 30%. This is consistent with Table 3.2, which displays the sales statistics by product.

As discussed, you can identify the regions or customer with the largest shares by changing the viewpoint. Tables 3.3 and 3.4 display the sales statistics by region and by customer, respectively.

Select the Sales Query to activate the viewpoint.

In the Task by Category tab, scroll to the Graph Category and select the Pie Task.

Chart Gallery tab

Confirm that **Simple Pie** is selected from the **Chart Gallery** tab.

Just above each chart type name, you can see a visual representation of the chart.

Notes:

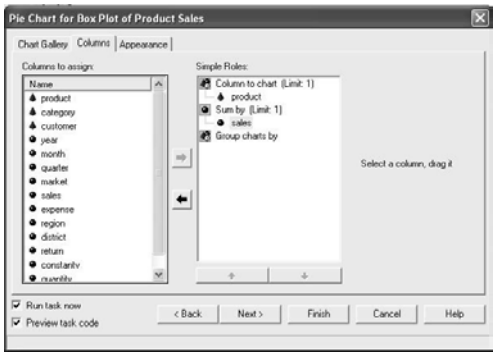
- The dialog box label is the default label unless you change the task label when finished.
- Once the task label is customized, the label displayed is still the default label if the **Chart Gallery** tab is selected.

Columns tab

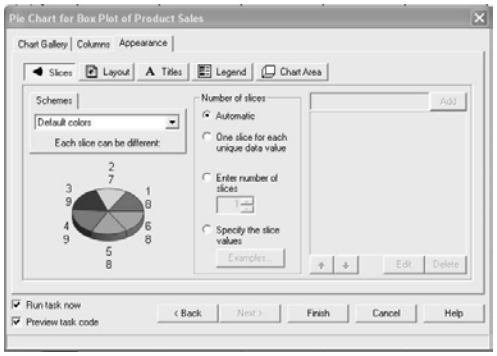
Drag and drop the PRODUCT and SALES variables from the **Columns to assign** list on the left to **Simple Roles** as follows:

- Assign PRODUCT as Column to chart.
- Assign SALES as Sum by variable.

This step plots the total sales by product as a pie chart.



Appearance tab



You do not need to change the default settings. You can specify options for each Appearance tab setting to enhance your chart.

Slices: Change the color scheme and the number of slices.

Label: Label each slice, change the outline color, make chart into 3D, and use slice color for slice labels.

Titles: Display the chart title and footnote.

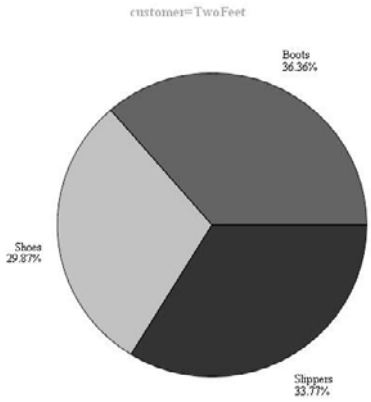
Legend: Show and position the legend along with legend label, font, and layout control.

Chart Area: Specify the background color and chart size.

Note that only the Slices setting is displayed in the **Appearance** tab.

By running the same analysis by customer, you can identify the highest product sales for each customer. As shown in Figure 4.4, the customer TwoFeet's boot sales had the largest shares with 36%.

Figure 4.4 Total Product Sales for TwoFeet

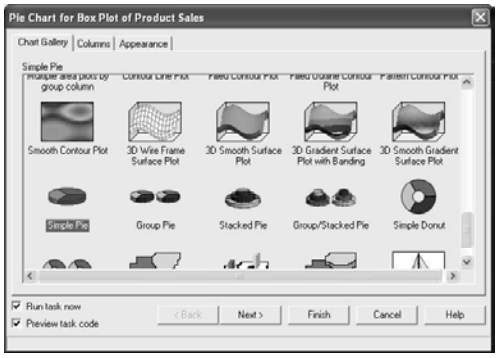


Click the Sales Query to activate it.

Click the Pie Task under the Graph Category to start a new task or copy the previous task by right-clicking, copy, and then paste.

Chart Gallery tab

Confirm that **Simple Pie** is selected. This is the same chart type as in the previous example.

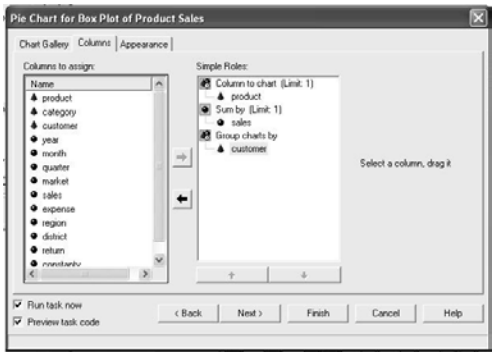


Columns tab

Drag and drop the PRODUCT, SALES, and CUSTOMER variables from the **Columns to assign** list on the left to **Simple Roles** as follows:

- Assign PRODUCT as Column to chart.
- Assign SALES as Sum by variable.
- Assign CUSTOMER as Group charts by variable.

The only change from the previous example is the CUSTOMER being assigned as the Group Chart by variable. This plots, for each customer, the total sales by product as separate charts.



Appearance tab

You do not need to change the default settings within the **Appearance** tab (Slices, Layout, Titles, Legend, Chart Area) unless you want to enhance your chart.

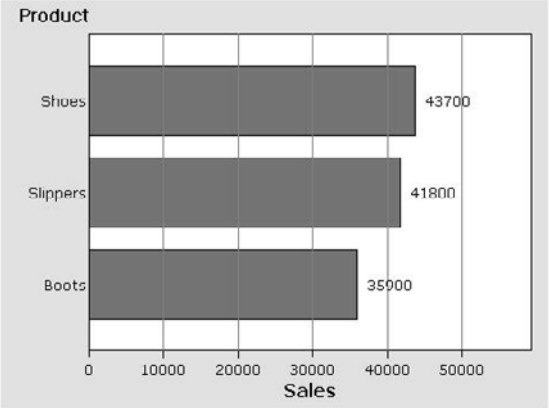
Comparing Products Using Horizontal Bars

Answer questions such as: Which products were top performers based on total sales?

In this section, you analyze the sales statistics tabulated across products. A bar chart will help you better understand and identify the products that generate the greatest sales. You will create horizontal bars that display total sales by product.

In the horizontal bar analysis shown in Figure 4.5, "Shoes" still account for the highest total sales. "Slippers" is the next, closely followed by "Boots." These numbers are consistent with Table 3.2 which displays the total product sales. Note that you can sort the bars by total sales to identify the best product. The vertical reference lines help to convey the difference between the total product sales.

Figure 4.5 Total Product Sales Using Bar Chart

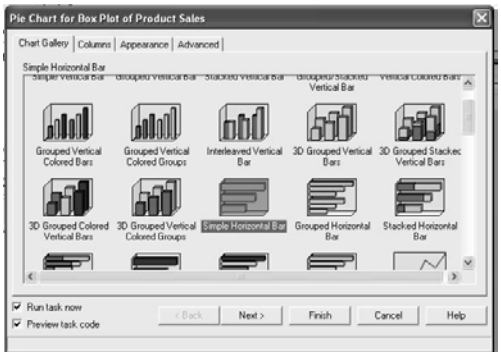


Click the **Sales Query** to activate it.

Click the **Bar Task** under the Graph Category.

Chart Gallery tab

Confirm that **Simple Horizontal Bar** is selected. It is the first of the horizontal bar charts.

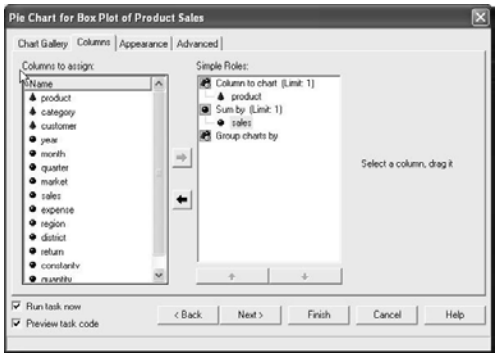


Columns tab

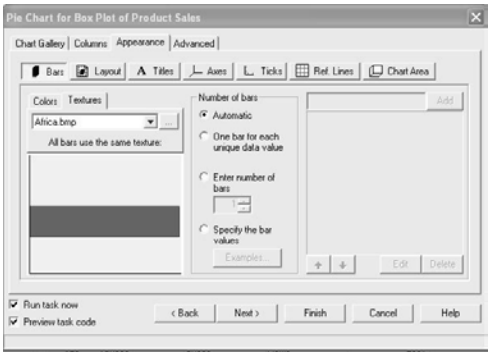
Drag and drop the PRODUCT and SALES variables from the **Columns to assign** list on the left to **Simple Roles** as follows:

- Assign PRODUCT as Column Chart.
- Assign SALES as Sum by variable.

This step will plot total sales by product as a horizontal chart.



Appearance tab



You do not need to change the default settings. You can specify options for each Appearance tab setting to enhance your chart:

Bars: Change the bar color and texture along with the number of bars.

Layout: Control the shape, order, and size of the bars.

Titles: Display the chart title and footnote.

Axes: Control the color, width, style font, and label for each axis, vertical and horizontal.

Ticks: Control the number and color of both major and minor ticks. For each axes, vertical and horizontal, there are Ticks settings.

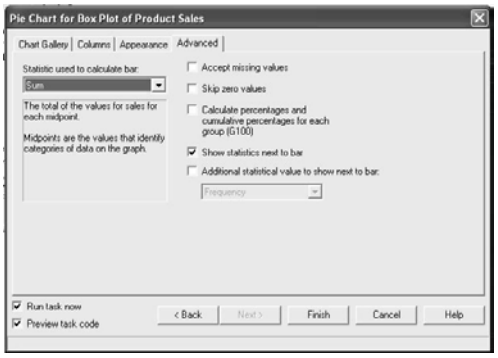
Reference Lines: Add reference lines to cross the horizontal axes. Select the automatic option for the **Properties for Horizontal**.

Chart Area: Specify the background color and chart size.

Note that only the Bars setting within the Appearance tab is displayed.

Advanced tab

Select Sum as the statistic to be used to calculate the bar. This option totals the product sales.



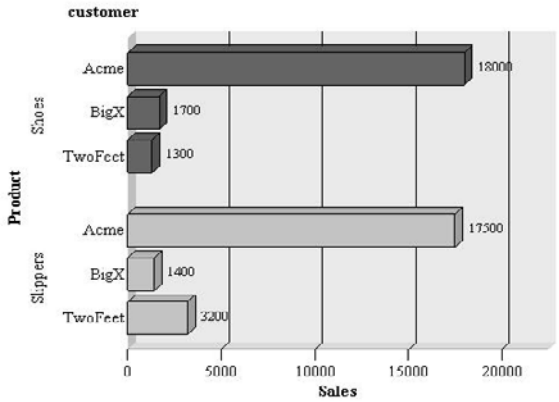
Comparing Product Groups Using Horizontal Bars

Answer questions such as: When comparing shoes and slipper sales, which customer had the highest sales?

In this example, you compare products by customers. For each product, you want to identify your best customer. In this analysis, you focus on the 2002 shoes and slipper sales data to group customer sales by product.

Figure 4.6 makes it easy to compare customers with grouped bars. The horizontal bar chart clearly shows that Acme had highest sales of $18,000 for shoes and $17,500 of slippers. The two other customers, BigX and TwoFeet, were far behind in shoes and slipper sales in 2002.

Figure 4.6 2002 Shoes and Slipper Sales by Customer Using Horizontal Bar Chart

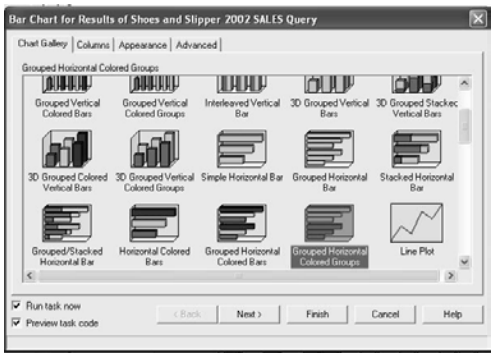


Click the Shoes and Slipper 2002 Sales Query to activate it. This query contains only the 2002 shoe and slipper sales.

Click the Bar Task under the Graph Category to start a new task or copy the previous task by right-clicking, copy, and then paste.

Chart Gallery tab

Confirm Grouped Horizontal Colored Groups is selected. Visually, you can see that the horizontal bars are grouped because of the two different colors.

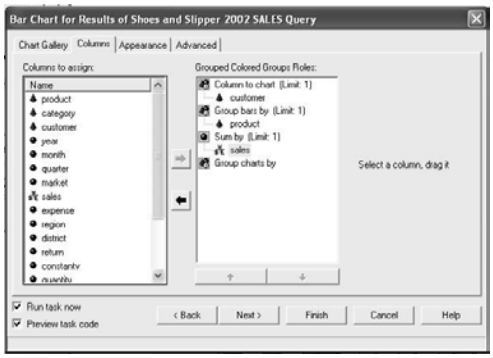


Columns tab

Drag and drop the CUSTOMER, PRODUCT, and SALES variables from the **Columns to assign** list on the left to the **Grouped Colored Groups Roles** as follows:

- Assign CUSTOMER as Column to Chart.
- Assign PRODUCT as Group Bars by variable.
- Assign SALES as Sum by variable.

The only changes from the previous example are the CUSTOMER being assigned as the Column to Chart instead of PRODUCT and the PRODUCT as Grouped by variable. This will plot 2002 shoes and slipper sales by customer as a horizontal chart.

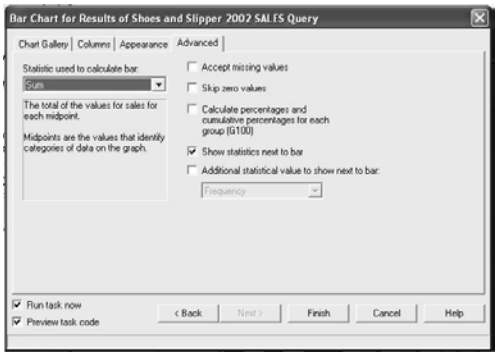


Appearance tab

You do not need to change the default settings within the Appearance tab (Bars, Layout, Titles, Axes, Ticks, Reference Lines, Chart Area) unless you want to enhance your chart.

Advanced tab

Select Sum as the statistic to be used to calculate the bar. This option totals the product sales.



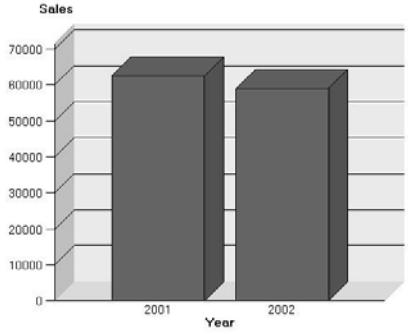
Comparing Overall Sales across Time Periods Using Vertical Bars

Answer questions such as: How much have sales risen this year?

For comparing overall sales over a period of time, it might be more useful to use **vertical bar charts**. Since graphs are generally read from left to right side, the left side could display the earliest year, and the right side could display the most recent year. This layout gives a sense of sales direction for future years by showing the trend. The horizontal reference lines help to convey the difference between the yearly sales.

Overall product sales had a slight decrease of $3,600 from $58,900 in year 2002 as compared to $62,500 in 2001. These results are consistent with Table 5.10 which displays the total product yearly sales. Since you have yearly total sales that have stayed about the same, you have achieved zero growth. If this trend continues, the flat growth might indicate a warning sign for a serious downward trend in the future.

Figure 4.7 Yearly Total Sales



Click the Sales Query to activate it.

Click the Bar Task under the Graph Category.

Chart Gallery tab

Confirm that **Simple Vertical Bar** is selected. This is the first chart type for vertical bars.

Columns tab

Drag and drop the YEAR and SALES variables from the **Columns to assign** list on the left to the **Simple Roles** as follows:

- Assign YEAR as Column to chart.
- Assign SALES as Sum by variable.

This step plots total sales by year as a vertical chart.

Appearance tab

You do not need to change the default settings except for the Ticks option. You can specify options for each Appearance tab setting to enhance your chart.

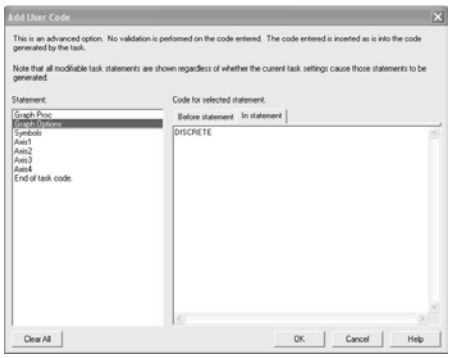
Bars Change the bar color and texture as well as the number of bars.

Layout Control the shape, order, and size of the bars.

Titles Display the chart title and footnote.

Axes Control the color, width, style font, and label for each axis (horizontal and vertical).

Ticks Set ticks for each axis (horizontal and vertical). You can establish the number and color of both major and minor ticks.



Note: Because the YEAR variable is continuous, you need to add the graph option DISCRETE in the SAS code using the **Insert Code** icon. Select the Graph Options on the left and In statement on the right. Then, enter the graph option DISCRETE in the right window.

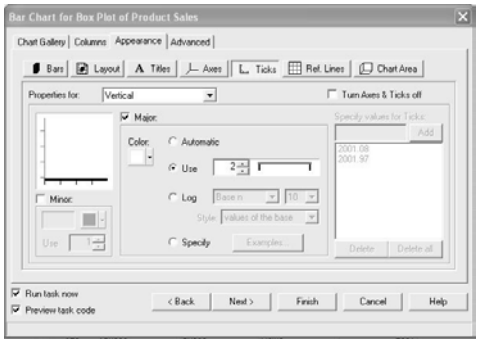
This places the DISCRETE option in the SAS code as shown below. Since the SAS code was added using the **Insert Code** button, you can continue to use the SAS Enterprise Guide task as before. See Chapter 5 for an alternative method for copying the code created from the task and then customize it with your own SAS code and options.

```
PROC gchart DATA=MYDATA.SALES
     ;
     vbar3d year /
     SUMVAR=sales
     RAXIS=AXIS1
     MAXIS=AXIS2
     AUTOREF
     SHAPE=Block
     TYPE=sum
     COUTLINE=BLACK
     FRAME

/* Start of custom user code. */
DISCRETE
/* End of custom user code. */
     ;

RUN;QUIT;
```

Reference Lines: Add reference lines to cross the horizontal axes. Select the automatic option for the **Properties for Horizontal**.

Chart Area: specifies the background color and chart size.

Note that only the Ticks setting on the Appearance tab is displayed.

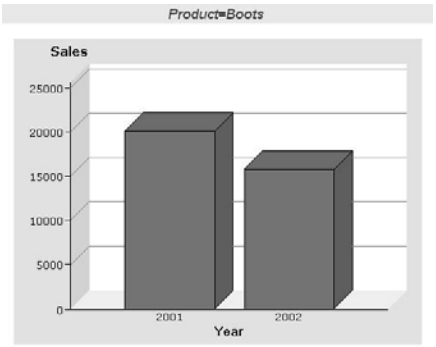


Advanced tab

Select Sum as the statistic to be used to calculate the bar. This option totals the product sales.

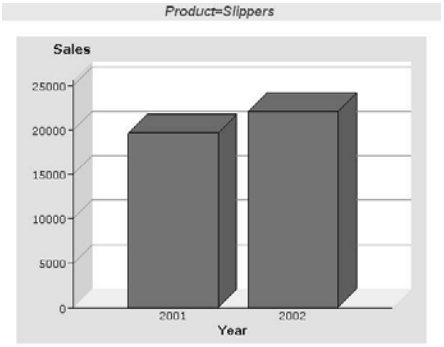
By running the same analysis by product, you can identify the growth direction for each product, as shown in Figure 4.8. For yearly total boot sales, for example, there was a decrease of $4,300 in 2002 from 2001. These results are consistent with Table 5.10 which displays the total boot yearly sales.

Figure 4.8 Yearly Total Boot Sales



For yearly total slipper sales, there is an increase of $2,400 in year 2002 from 2001, as shown in Figure 4.9. These results are consistent with Table 5.10 which displays the total slipper yearly sales.

Figure 4.9 Yearly Total Slipper Sales

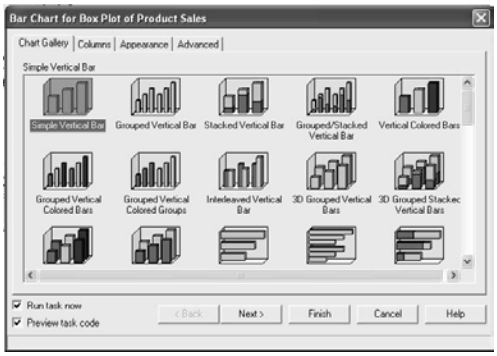


Click the Sales Query to activate it.

Click the Bar Task under the Graph Category to start a new task or copy the previous task by right-clicking, copy, and then paste.

Chart Gallery tab

Confirm that **Simple Vertical Bar** is selected. This is the same chart type as in the previous example.

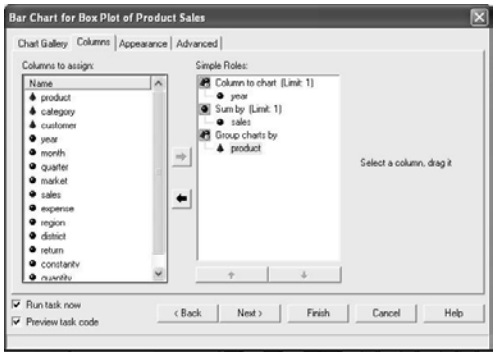


Columns tab

Drag and drop the YEAR, SALES, and PRODUCT variables from the **Columns to assign** list on the left to **Simple Roles** as follows:

- Assign YEAR as Column Chart.
- Assign SALES as Sum by variable.
- Assign PRODUCT as Group Charts by variable.

The only change from the previous example is that you assign the PRODUCT as the Group Chart by variable. This creates a bar chart in which each bar represents the total sales by year for a product plots for each product. The total sales by year as separate bars.

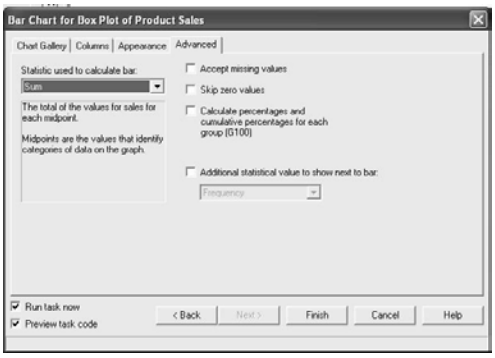


Appearance tab

You do not need to change the default settings within the Appearance tab unless you want to enhance your chart.

Advanced tab

Select Sum as the statistic to be used to calculate the bar. This option totals the product sales.



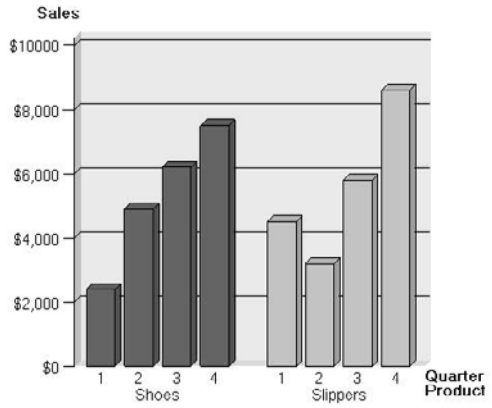
Comparing Product Sales across Time Periods Using Vertical Bars

Answer questions such as: Over a period of time, how did shoe sales compare with slipper sales?

To get a better understanding of product sales across time, you can analyze sales at the quarter level to identify the best quarters for each product. In addition, you can get a clear picture of the direction of sales across quarters by reviewing trends.

Over a period of time, the distributions of shoe and slipper sales are very similar. Both have peak sales in the third and fourth quarters of 2002. These results are consistent with Table 5.13, which displays the slipper sales by year and quarter.

Figure 4.10 2002 Shoes and Slipper Sales by Quarter

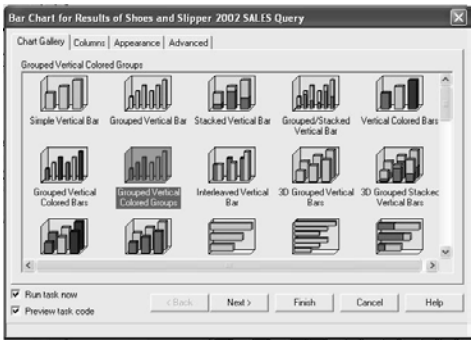


Click the Shoes and Slipper 2002 Sales Query to activate it. This query contains only the 2002 shoes and slipper sales.

Click the Bar Task under the Graph Category to start a new task or copy the previous task by right-clicking, copy, and then paste.

Chart Gallery tab

Confirm that **Grouped Vertical Colored Groups** is selected. Visually, you can see that the vertical bars are grouped because of the two different colors.

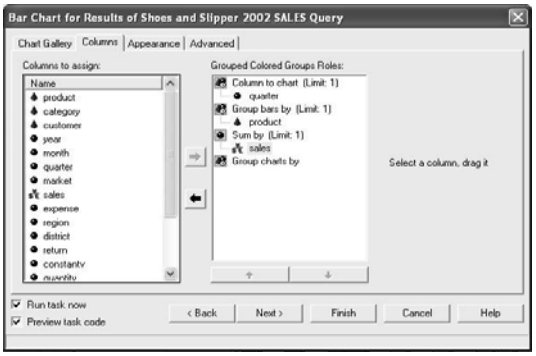


Columns tab

Drag and drop the QUARTER, PRODUCT, and SALES variables from the Columns to assign list on the left to the Grouped Colored Groups Roles on the right side as follows:

- Assign QUARTER as Column Chart.
- Assign PRODUCT as Group Bars by variable.
- Assign SALES as Sum by variable.

The only change from the previous example is the QUARTER being assigned as the Column to Chart instead of YEAR. This will plot 2002 shoes and slipper sales by quarter as a vertical chart.

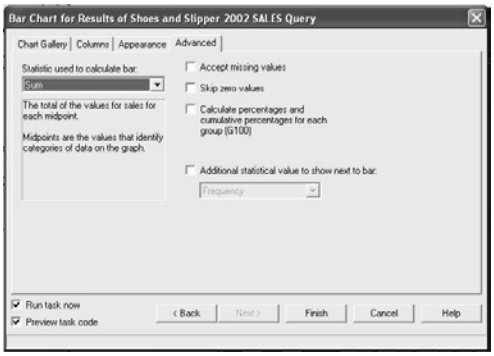


Appearance tab

You do not need to change the default settings within the Appearance tab except for the Ticks option. Because the QUARTER variable is continuous, you will need to add the graph option DISCRETE to display unique values. See Figure 4.7 for instructions on adding SAS code using the **Insert Code** icon. You can change the other options if you want to enhance your chart.

Advanced tab

Select Sum as the statistic to be used to calculate the bar. This option totals the product sales.



Chapter 5

IN THIS CHAPTER

Writing and Modifying Code to Create Reports and Graphs

This chapter shows how to access the extended capabilities of SAS beyond SAS Enterprise Guide tasks. Specifically, this chapter shows you ways to control and customize your results by either modifying the SAS code generated from SAS Enterprise Guide tasks or by writing your own SAS programs. One reason for doing this would be to specify statements and options that might not be possible using tasks. Another reason for doing this would be to specify SAS procedures that do not have a corresponding task, such as the REPORT procedure. Unless indicated, almost all SAS code in this chapter references the Sales Query.

	Sales by Year							
	2001			2002				
PRODUCT	N	Total	% Total	N	Total	% Total	QNTY Total	SALES Total
Boots	2,610	$20,100	32	2,772	$15,800	27	5,382	$35,900
Shoes	2,270	$22,700	36	1,910	$21,000	36	4,180	$43,700
Slippers	3,940	$19,700	32	4,420	$22,100	38	8,360	$41,800
Total	*8,820*	*$62,500*	*100*	*9,102*	*$58,900*	*100*	*17,922*	*$121,400*

Overview

To take full advantage of this chapter, you should be knowledgeable in SAS programming and you should plan to work directly with the SAS code. The scope and intent of this book is not to teach SAS programming but to show how easy it is to create and modify SAS code using SAS Learning Edition. To create most of the examples in this chapter, you will be writing and revising SAS code rather than using the point-and-click method described in previous chapters. The exception is Table 5.7, which can be created by combining several tasks.

Note: All of the SAS code in the examples in this chapter can be copied and executed as is without needing to be modified. Also, you can copy the SAS code from the Companion Web site for this book instead of manually entering the code.

You can create reports by submitting SAS programs using SAS Learning Edition. While most SAS procedures have an associated task, the REPORT procedure does not have a point-and-click task interface. Since the REPORT procedure is designed to create tables, most examples in this chapter show how to create tables using the REPORT procedure. Writing your own REPORT procedure code is the only way to produce tables from the REPORT procedure. Currently, there is no SAS Enterprise Guide task that enables you to produce these tables.

Creating tables using the REPORT procedure greatly increases your ability to create complex reports. The REPORT procedure offers more flexibility in report generation by combining the features of the PRINT, MEANS, and TABULATE procedures with features of the DATA STEP. Many of the tables generated using PROC REPORT cannot be recreated easily without DATA STEP programming. This is the major reason for using the REPORT procedure. Be aware that the REPORT procedure does not run in the interactive mode in SAS Learning Edition.

All tables in this chapter answer the business-related questions listed in Chapter 2. To create these customized reports and graphs, the process involves writing the code using the Code Editor and then submitting the code by selecting `Code ►Run on Local` from the toolbar. You can open the Code Editor window by starting the **Create Code** task. Note that while you can write your own SAS program containing statistical procedures such as PROC ANOVA, this book does not have an example of how to do this type of programming.

Table 5.1 shows some key differences and advantages between the SAS Enterprise Guide tasks and the Code Editor approaches.

Table 5.1 Which Path You Follow Depends on What You Want to Do

Selecting SAS Enterprise Guide Tasks (Non-programmers)	Writing and Modifying SAS Code (Programmers)
Guided analysis and reporting process by running a sequence of tasks	Total control in analysis and reporting with custom SAS programming
Query window to prepare data set	Data step manipulation to prepare data set
Automatically generate SAS code	Customize SAS code and output with ODS
Access selected SAS procedures through tasks	Access most SAS procedures including PROC REPORT*
Selected statements and options	Any SAS procedure statement or option
Automatic pointer to reference data sets	LIBNAME statement to reference data sets

* SAS Learning Edition Modules: Base SAS, SAS/GRAPH, SAS/STAT, SAS/ETS, SAS/QC

Understanding the Typical Workflow

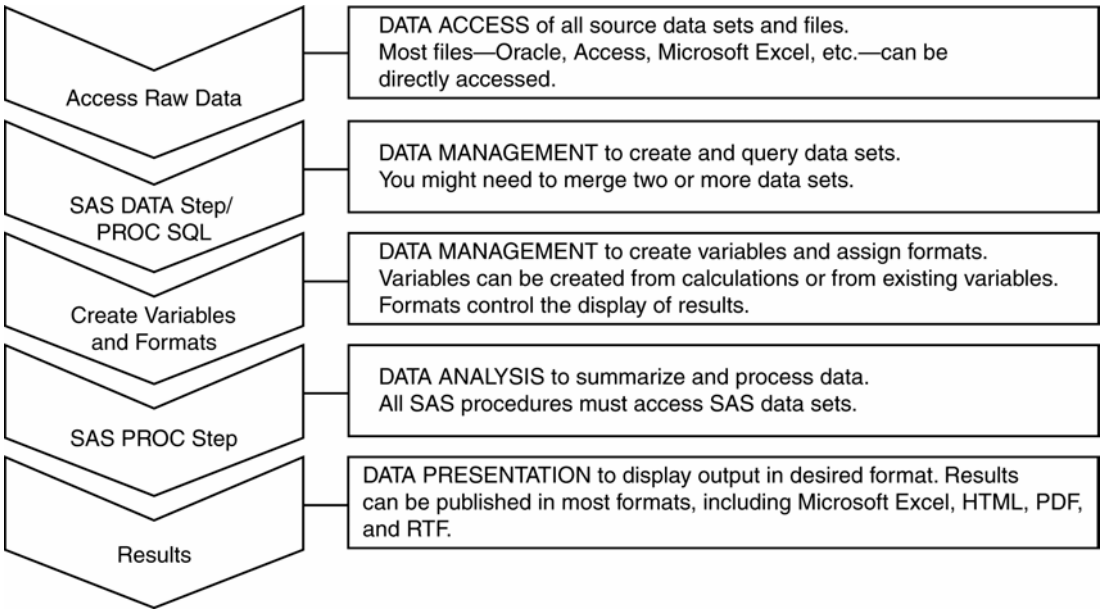
The typical workflow for using SAS Learning Edition involves

- accessing the raw data
- creating SAS data sets and variables
- querying the data if needed
- analyzing the data using tasks
- publishing the results in various formats.

Using SAS Learning Edition's point-and-click feature makes it easier to learn, modify, and create SAS code. For each task performed, SAS Learning Edition automatically saves the SAS log, list, and output files. Note that not all steps in the sequence below might be needed to produce results.

As a programmer, you have total control to bypass the SAS Enterprise Guide tasks in order to write your own SAS code. In addition, you can write one program that performs multiple SAS Enterprise Guide tasks. For example, you can create a variable and assign a format in a DATA step and then use the variable in multiple SAS procedures. It is not possible to perform these steps in any single SAS Enterprise Guide task. SAS Enterprise Guide tasks are limited to perform discrete actions and are limited to sequential execution.

Figure 5.1 Typical Workflow



Specifying SAS Procedures

Programmers can directly specify SAS procedures without selecting SAS Enterprise Guide tasks. Thus, you can write your SAS code by selecting the **Create Code** task or by selecting tasks by name. See Chapter 2 for more information on selecting tasks by name.

Table 5.2 lists selected SAS Enterprise Guide tasks along with the corresponding SAS procedure name. In addition, the table is helpful to identify the appropriate task or SAS procedure to use based on its strength and features. Programmers can save time by customizing the code generated from tasks instead of writing the code from scratch. It is important to realize that tasks have code behind them that are based on SAS procedures. By taking advantage of these SAS procedures along with customizing the code, you have access to more of the power of SAS in SAS Learning Edition.

See the article on "Writing Reports with SAS Software — What Are Your Options?" in *Observations* for an overview of reports and which SAS procedure to use for creating them.

Table 5.2 Selected SAS Enterprise Guide Tasks with Strengths and Features of Each

Selected SAS Enterprise Guide Tasks (SAS procedure)	Strengths and Features
One-Way Frequency task, **Table Analysis** task (PROC FREQ)	Produces one-way to n-way tables as frequency counts.
Summary Statistics task (PROC MEANS)	Computes descriptive statistics for numeric variables and limited inferential statistics.
List Data task (PROC PRINT)	Provides easy-to-produce data listings with titles, footnotes, and column sums.
Rank task (PROC RANK)	Computes ranks for one or more numeric variables.

(*continued on next page*)

Table 5.2 (*continued*)

Selected SAS Enterprise Guide Tasks (SAS procedure)	Strengths and Features
Create Code task (PROC REPORT)	Combines features from PROC PRINT (generate ordered lists), PROC MEANS (generate summary statistics), PROC TABULATE (nesting), and the DATA STEP (create variables and custom lines) to create attractive and complex listings and summaries.
Append Table task (PROC SQL)	Combines data from multiple data sets and advanced data manipulation to create new variables and filter conditions.
Summary Tables task (PROC TABULATE)	Easily produces hierarchical tabular reports containing descriptive statistics by columns or by rows along with percent distribution reports. It can also create multipage summary tables.
Transpose task (PROC TRANSPOSE)	Convert rows to columns or columns to rows.
Distribution Analysis task (PROC UNIVARIATE)	Computes the broadest set of descriptive statistics and inferential statistics.

SAS Enterprise Guide tasks are not intended to cover all of the functionality of the corresponding SAS procedures, but the tasks will generate the SAS code for the commonly used features. Table 5.3 lists the SAS procedures available in SAS Learning Edition 2.0.

Table 5.3 SAS Products and Procedures in SAS Learning Edition 2.0

SAS Product	SAS Procedures
Base SAS	APPEND, COMPARE, CONTENTS, COPY, CORR, DATASETS, EXPORT, FORMAT, FREQ, IMPORT, MEANS, OPTIONS, OPTLOAD, OPTSAVE, PLOT, PRINT, PRTDEF, RANK, REGISTRY, REPORT, SORT, SQL, STANDARD, SUMMARY, TABULATE, TRANSPOSE, UNIVARIATE
SAS/GRAPH	GANNO, GCHART, GCONTOUR, GMAP, GOPTIONS, GPLOT, GRADAR, G3GRID
SAS/STAT	ANOVA, BOXPLOT, GENMOD, CANCORR, CLUSTER, DISCRIM, FACTOR, FASTCLUS, FREQ, GLM, LIFETEST, LOGISTIC, NLIN, NPARA1WAY, MIXED, PHREG, PRINCOMP, REG, SURVEYSELECT, TREE, TTEST

(*continued on next page*)

Table 5.3 (*continued*)

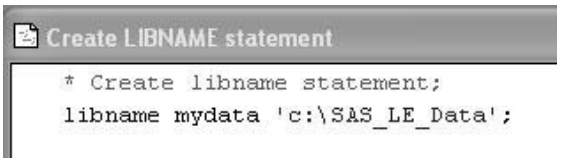
SAS Product	SAS Procedures
SAS/ETS	ARIMA, AUTOREG, EXPAND, FORECAST, PDLREG, TSCSREG
SAS/QC	CAPABILITY, PARETO, SHEWHART

As discussed, in SAS Learning Edition 2.0, there is not a task for the REPORT procedure. As a SAS programmer, you can write code that specifies any of the statements and options available in these SAS procedures. In addition, although the tasks do not support the STYLE= option in the PRINT or TABULATE procedures, you can greatly enhance your output by adding this code yourself. It is beyond the scope of this book to show the STYLE= option.

Assigning Libraries to Access SAS Data Sets

A **library** is a location where a collection of files such as data sets is stored. As a programmer, you can directly access data sets by first creating the libref. Once you run the LIBNAME statement, you can write SAS code that uses the libref to reference your data sets. If you do not take this approach, SAS Learning Edition automatically creates the libref when you add data to the project. SAS Learning Edition then uses this libref when you run a task.

To create a libref, select the Create Code task and enter the LIBNAME statement in the blank Code Editor window as follows. Next, select `Code►Run on Local` from the toolbar to submit your program.



For greater flexibility in customizing the code, it is best to use a LIBNAME statement to create librefs to the source data sets instead of using the default librefs. User-defined librefs are permanent and can be used within the project. Note that librefs can access permanent data sets and not queries created within SAS Learning Edition unless the query is exported and saved as a permanent data set.

Saving and Reusing Code

After you save and execute tasks, you can recreate the tables at any time with updated data. This is because SAS Learning Edition does not copy the source data set, but points to the original data file. Thus, if the original data file is updated, the next time you run your task, you are accessing the most current version of the data file. Also, you can share a project with colleagues so that they can run the same tasks that you saved in your project.

Modifying Generated SAS Code

As discussed, programmers might want to modify the SAS code generated from SAS Learning Edition for additional control and customization. For any task, you can select the **Preview task code** option to view the SAS code. Once you expand the window containing the SAS code, copy the code into the window for the **Create Code** task. Use the mouse to highlight all of the code, right-click to copy the code, and then paste it in the new Create

Another way to copy the code is to right-click the task item in the project window and then select **Add as Code Template**. This adds a modifiable code item to the project that is separate from the original task.

As an alternative to modifying the code using the Create Code task, you can insert SAS code in selected places using the same task. Because SAS Learning Edition limits the places where you can insert code, it might be better to use the Create Code task approach.

Note: Once you modify the code in the same task, the task is no longer associated with the code. That is, you can no longer use the task to make changes in your report because the underlying code was customized.

Let's modify the code generated from creating Table 3.2 to change the min (minimum) sales to max (maximum) sales statistics.

Table 3.2 Sales Statistics by Product

			Product Sales		
		Unit Totals	Dollar Totals	Average (Mean)	Min
Product Category	Product				
Shoes	Boots	5,382	$35,900	$1,496	$-1400
	Shoes	4,180	$43,700	$1,821	$100
Slippers	Slippers	8,360	$41,800	$1,742	$0
Total		17,922	$121,400	$1,686	$-1400

Once you copy the code, replace any reference to 'min' with a reference to 'max' to get the maximum value in the table. Once you make the change, select `Code►Run on Local` from the toolbar to submit your program. Note that all other columns remained the same. All you did was to manually change the minimum statistics to the maximum statistics in the last column.

Table 5.4 Sales Statistics (Max instead of Min) by Product

			Product Sales		
		Unit Totals	Dollar Totals	Average (Mean)	Max
Product Category	Product				
Shoes	Boots	5,382	$35,900	$1,496	$3,500
	Shoes	4,180	$43,700	$1,821	$4,000
Slippers	Slippers	8,360	$41,800	$1,742	$4,700
Total		17,922	$121,400	$1,686	$4,700

Even though this code is basically the same as that in the task used to create Table 3.2, this code is not associated with any task because it was saved using the **Create Code** task.

Overview of Basic SAS Programming Rules

If you want to modify the SAS code to customize a table, be sure that you know some basic SAS programming rules.

1. All SAS statements must end with a semi-colon.
2. Place a RUN statement at the end of each DATA step and SAS procedure. A DATA step is used to create data sets. SAS procedure steps analyze data. Place a QUIT statement at the end of each SQL procedure.
3. Follow the syntax rules for each SAS procedure.
4. Comments can be added to the SAS program to help document the process. For example:

Note: In general, SAS procedures have common statements to process data sets as shown in Table 5.5.

Table 5.5 PROC Statement Elements (PROC PRINT Example)

PROC PRINT	Identifies the SAS procedure.
DATA = MYDATA.SALES	Specifies the data set to use as the input source.
LABEL ;	Specifies SAS procedure options.
VAR PRODUCT SALES QUANTITY;	Specifies the order variables in which the variables are displayed or analyzed.
BY CUSTOMER;	Specifies by-group variables used in the display and analysis. This generally requires sorting the data set in advance.
WHERE SALES > 3000;	Applies a subset condition.
RUN;	Ends the SAS procedure. **Required**.

Be aware of how SAS handles missing data for numeric variables. In general, most procedures, by default, include missing data values in the results if your data set contains missing data values. The MEANS procedure, however, by default excludes missing values in all calculations.

You can also control the titles and footnotes of your tables with the TITLE and FOOTNOTE statement. Place the title or footnote statement before the SAS procedure to be used. Once titles and footnotes are specified, they remain in effect for all output until changed or cancelled. To cancel a title or footnote, specify the TITLE; or FOOTNOTE; statement respectively.

Note: The default footnote is

```
Generated by the SAS System (&_EGSERVERNAME, &SYSSCPL) on %SYSFUNC(DATE(),
EURDFDE9.) at %SYSFUNC(TIME(), TIMEAMPM8.)
```

The default title is task-specific. Table 5.6 shows examples of titles and footnotes.

Table 5.6 Titles and Footnotes

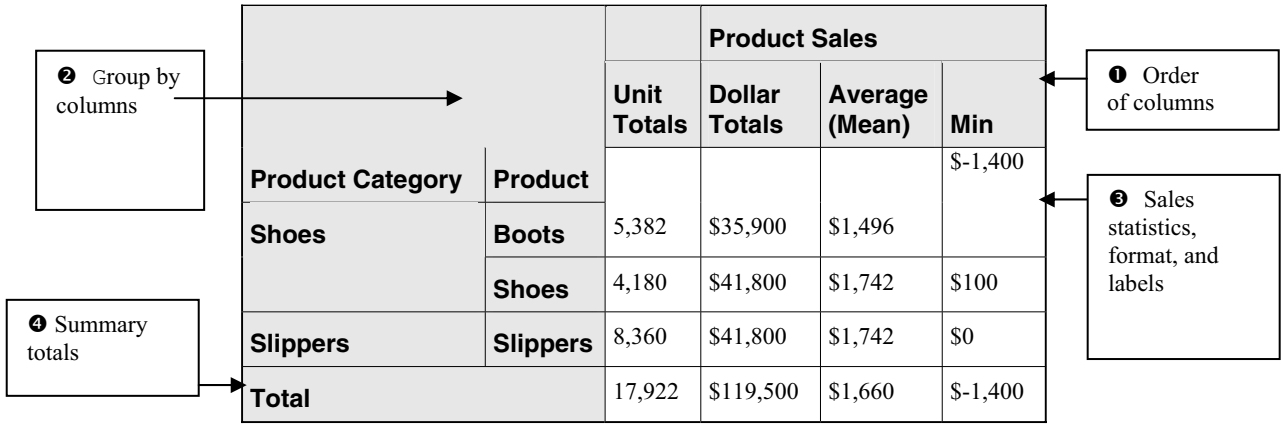
TITLE1 'This is the title of the Report';	TITLE statements print up to 10 lines of titles at the top of each page.
FOOTNOTE1 'This is a footnote.';	FOOTNOTE statements print up to 10 lines of footnotes at the bottom of each page.

Creating Your First PROC REPORT Table

As always, there is more than one way to generate a table in SAS. In this section, you recreate Table 3.2 by writing the appropriate PROC REPORT code.

Open the Code Editor window by selecting the **Create Code** task, and enter the SAS code following Table 5.7. Submit your code by selecting `Code►Run on Local` from the toolbar. (It is beyond the scope of this book to review all the statements or options in the REPORT procedure).

Table 5.7 Sales Statistics by Product Using PROC REPORT



			Product Sales		
Product Category	**Product**	**Unit Totals**	**Dollar Totals**	**Average (Mean)**	**Min**
Shoes	**Boots**	5,382	$35,900	$1,496	$-1,400
	Shoes	4,180	$41,800	$1,742	$100
Slippers	**Slippers**	8,360	$41,800	$1,742	$0
Total		17,922	$119,500	$1,660	$-1,400

Enter the SAS code for Table 5.7.

```
title 'Table 5.7 Sales Statistics by Product Using PROC REPORT ';

proc report data=mydata.sales nowd split="*" headline;

 column
  category product quantity sales sales=mean sales=min;  ❶

 define category / group  'Product Category' center;  ❷
 define product / group 'Product';
```

```
define quantity / sum 'Unit*Totals' center format=comma6.;  ❸
define sales / sum 'Dollar*Totals' format=dollar8.;
define mean / mean 'Average*(Mean)' format=dollar8.;
define min / min 'Min' format=dollar8.;

rbreak after / summarize skip ol ul;  ❹
compute after;
 category = 'Total';
endcomp;
```

❶ The COLUMN statement specifies the order of variables.

❷ The table is grouped by variables category and product.

❸ Table statistics, formats, and labels are specified.

❹ Summarize sales statistics to get total units, total sales, mean, and minimum sales.

Building Your First Report

Answer questions such as: What were the descriptive sales statistics such as total, mean, median, and standard deviation by product?

As an example for writing your first report, create a table that displays the descriptive sales statistics by product.

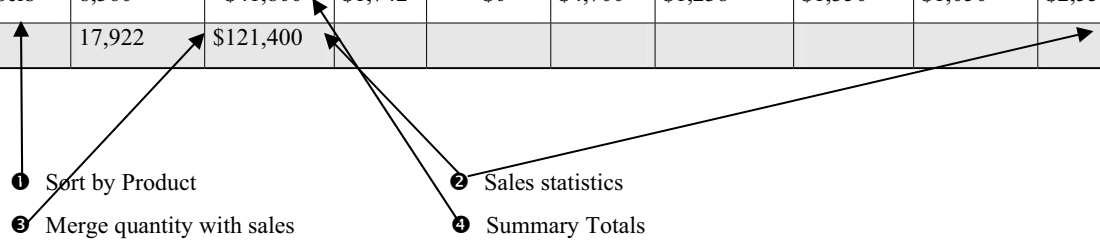
To include all descriptive statistics such as total, mean, median, and standard deviation in one table, you can write a SAS program that includes the following SAS procedures:

- PROC SORT
- PROC UNIVARIATE
- PROC TABULATE
- DATA step procedures

Open the Code Editor window by selecting the Create Code task. Enter the SAS code below to create Table 5.8. Select `Code ▸ Run on Local` from the toolbar to submit your program.

Table 5.8 Descriptive Sales Statistics by Product

Product	the sum, quantity	the sum, sales	the mean, sales	the smallest value, sales	the largest value, sales	the standard deviation, sales	the median, sales	the lower quartile, sales	the upper quartile, sales
Boots	5,382	$35,900	$1,496	$-1,400	$3,500	$1,429	$1,700	$450	$2,650
Shoes	4,180	$43,700	$1,821	$100	$4,000	$1,048	$1,750	$1,000	$2,500
Slippers	8,360	$41,800	$1,742	$0	$4,700	$1,238	$1,350	$1,050	$2,550
	17,922	$121,400							



Enter the SAS code for Table 5.8.

```
 by product;
run;

proc univariate data=sales_sort noprint;  ❷
 by product;
 var sales;
 output out=sales_uni sum=sum mean=mean std=std median=median q1=q1
        q3=q3
 qrange=qrange min=min max=max;
run;

title 'Results from PROC UNIVARIATE';
*proc contents data=sales uni;
```

```
*run;
proc univariate data=sales_sort noprint;  ❷
 by product;
 var quantity;
 output out=quantity_uni sum=ncount;
run;

title 'Table 5.8 Descriptive Sales Statistics by Product';
*proc contents data=quantity_uni;
*run;
*proc print data=quantity_uni;
*run;

proc sort data=sales_uni;
 by product;
run;
proc sort data=quantity_uni;
 by product;
run;

data sales_uni;   ❸
 merge sales_uni quantity_uni;
 by product;
run;

proc print data=sales_uni noobs label;  ❹
 var product ncount sum mean min max std median q1 q3;
 format ncount comma6. sum mean min max std median q1 q3 dollar8.;
 sum ncount sum;
run;
```

❶ Prepare the data set by sorting by the product variable using PROC SORT.

❷ Generate the descriptive statistics by product for each sales and quantity variable using PROC UNIVARIATE.

❸ Merge the descriptive statistics information from the sales and quantity variables by product using DATA STEP.

❹ Finally, generate the table using PROC PRINT. Include the NOOBS option to prevent displaying the observation numbers, and use the LABEL option to display variable labels. Include the SUM statement to display total volume and sales.

As an alternative to writing this SAS program, you can use the following SAS Enterprise Guide tasks to accomplish the same results:

- Several Query windows to sort and join data sets by product
- Several Summary Tables tasks to calculate the descriptive statistics by sales and quantity
- List Data task to display the results in table format

See Chapter 3 for examples of using these tasks.

Customized Data in a List Report

Answer questions such as: Who were the top customers for a selected product?

This example shows you how to write SAS code to calculate any sales that were greater than 10% of the total sales. Using the **Create Code** task, you can write the code to identify the top customers for a selected product and year. With this flexibility, you can compare products at the detail level.

From the results of Table 5.9, you can see that 2002 shoe sales were >= 15 % of total sales for two months (September and October) in the third and fourth quarters. These are retail sales from Acme in the Northern region. You can write the code for SAS Learning Edition to automatically identify these monthly sales transactions.

Table 5.9 Detail Listing of 2002 Shoe Sales

							SALES				
QTR	MONTH	REGION	DISTRICT	MARKET	CUST.	N	SALES	%	COST	NET	
1	1	Northern	NE	Catalog	Acme	100	$1,000	5%	$800	$200	
	2	Northern	NE	Catalog	Acme	100	$1,000	5%	$800	$200	
	3	Northern	NE	Catalog	Acme	40	$400	2%	$320	$80	
2	4	Northern	NE	Catalog	Acme	210	$2,100	10%	$1,680	$420	
	5	Northern	NE	Retail	Acme	110	$1,100	5%	$880	$220	
	6	Northern	NE	Retail	Acme	170	$1,700	8%	$1,360	$340	
3	7	Northern	NE	Retail	Acme	170	$1,700	8%	$1,360	$340	
	8	Southern	SW	Retail	TwoFeet	130	$1,300	6%	$1,040	$260	
	9	Northern	NE	Retail	Acme	320	$3,200	15%	$2,560	$640	*
4	10	Northern	NE	Retail	Acme	210	$4,000	19%	$2,400	$1,600	*
	11	Western	CW	Retail	BigX	170	$1,700	8%	$1,360	$340	
	12	Northern	NE	Retail	Acme	180	$1,800	9%	$1,440	$360	
					Total		$21,000	100%	$16,000	$5,000	*

* Greater than or equal to 15% of total sales for 2002.

Enter the SAS code for Table 5.9.

```
proc report data=mydata.sales nowd split="*" headline;

 where year = 2002 and product = 'Shoes';

 column
  (quarter month region district market customer quantity
   sales,(sum pctsum) expense net comment) ;

 define quarter / order id 'QTR' center format=quarterv.;
 define month / display id 'MONTH' format=monthv.;

 define region / display 'REGION' format=regionv.;
 define district / display 'DISTRICT' format=district.;
```

```
define quantity / display 'N' format=comma6.;
define sales / display 'SALES' sum format=dollar8.;
define expense / display 'COST' sum format=dollar8.;

define net / computed 'NET' format=dollar8.;
compute net;
 net = sales.sum - expense.sum;
endcomp;

define comment / computed width=1 ' ' flow;
compute comment/ char length=2;
 if sales.pctsum >= .15 then comment = '*';
 else comment = '';
endcomp;

define sum / 'SALES' format=dollar8.;
define pctsum /'%' format=percent6. width=3;

rbreak after / dol dul skip summarize;

compute after;
 customer='Total';

 line @4 "  ";
 line @4 "  ";
 line @4 "* Greater than or equal to 15% of total sales for 2002.";
endcomp;
```

Creating Customized Percent Distribution Reports

Answer questions such as: Which product had the highest sales by year?

Table 5.10 is a good example of an output that is not easily produced from any SAS Enterprise Guide task due to the percent of total sales calculation. Since you can write SAS programs in SAS Learning Edition, you have complete control of your table. Open the Code Editor window by selecting the Create Code task. Enter the SAS code below to create this table. Select `Code►Run on Local` from the toolbar to submit your program.

When you write a SAS program, you can write PROC REPORT code to include percentages in various formats. With this flexibility, you can get information about the largest percent sales by product each year. The percent of total sales is tabulated vertically for each year.

In this section, you want to look at the largest percent share or proportion of sales by product each year. From the results of Table 5.10, you can see that, for 2001, shoes had the highest sales ($22,700) by capturing 36% of all sales, and that slippers had the highest sales ($22,100) by capturing 38% of all sales for 2002. This is 2% higher than the closest product, shoes ($21,000), and 11% higher than boot sales ($15,800).

Table 5.10 Yearly Sales Distribution by Product

	Sales by Year							
	2001			2002				
PRODUCT	N	Total	% Total	N	Total	% Total	QNTY Total	SALES Total
Boots	2,610	$20,100	32	2,772	$15,800	27	5,382	$35,900
Shoes	2,270	$22,700	36	1,910	$21,000	36	4,180	$43,700
Slippers	3,940	$19,700	32	4,420	$22,100	38	8,360	$41,800
Total	8,820	$62,500	100	9,102	$58,900	100	17,922	$121,400

Enter the SAS code for Table 5.10.

```
proc format;
  picture pctf (round) low-<0 = " 009%" (PREFIX="-")
                      0-high = " 009%";
quit;

proc report data=mydata.sales nowd split="*" headline;

 column
  product year,(quantity sales pct_prd) quantity=grd_tot sales=grd_pct;

 define product / group 'PRODUCT';
 define year / across 'Sales by Year' center;
 define quantity / sum 'N' center format=comma6.;
 define sales / sum 'Total' format=dollar8.;

 define pct prd / computed '% Total' width=5 format=pctf.;
 define grd tot / sum 'QNTY Total' format=comma6.;
 define grd_pct / sum 'SALES Total' width=5 format=dollar8.;

 compute before;
  gt1= c3_; gt2 = _c6_; gt3 = _c8_;
 endcomp;

compute pct_prd;
   c4  = ( c3 /gt1)*100;  c7  = ( c6 /gt2)*100;
```

```
compute after;
  product = 'Total';
endcomp;
```

Different Statistics: Yearly Mean Sales

As discussed in Chapter 3, once you create a table, you can easily modify it to analyze different viewpoints of the same information. In this case, you can modify the SAS program used to create Table 5.10 to create Table 5.11, which displays yearly mean sales instead of yearly total sales distribution.

Table 5.11 Yearly Mean Sales Distribution by Product

	Sales by Year			
	2001		2002	
PRODUCT	N	Mean	N	Mean
Boots	2,610	$1,675	2,772	$1,317
Shoes	2,270	$1,892	1,910	$1,750
Slippers	3,940	$1,642	4,420	$1,842

This is easily done by:

- changing the `sum` to `mean`
- removing these variables: `pct_prd`, `grd_tot`, and `grd_pct`
- removing the `rbreak` and `compute` block code.

Also, change the label to **Mean** instead of **Total**. To make these changes, open the Code Editor window for Table 5.10 by selecting the Create Code task. Select `Code►Run on Local` from the toolbar to submit your program.

Enter the SAS code for Table 5.11.

```
proc report data=mydata.sales nowd split="*" headline;

 column
  product year,(quantity sales);

 define product / group 'PRODUCT';
 define year / across 'Sales by Year' center; define quantity / mean 'N'
center format=comma6.;
 define sales / mean 'Mean' format=dollar8.;
```

Creating Percent Change Reports

Answer questions such as:

- Based on percent sales change, where is your product in the product life cycle (introduction, growth, maturity, decline)?
- Which products had the best percent sales change by year?

Another method to analyze percent change is to calculate the difference in sales from the pervious year. Based on the percent change value, you can determine where your product is in the product life cycle (introduction, growth, maturity, decline).

Product Life Cycle

	Phase	Observation	Expectation
↗	Introduction	Slow growth	Small increase in sales over next few quarters
↑	Growth	Great growth	Maintain high profit margins
↔	Maturity	Flat sales	Maybe due to increased price competition
↘	Decline	Decreased sales	Small decrease in sales over next few quarters

In this section, you look for trends to help identify the driving force behind the sales. Based on the ranking of percent change, you can determine if your product is a leader (top three), challenger (next three), follower (middle) or niche player (smallest) (Driscoll, 2003).

From the results of Table 5.12, you can see that slipper sales grew at a 12 percent rate while boot and shoe sales actually went down by 21 percent and 7 percent respectively. From this information, you can see that slippers are in the ↑ growth phase and that you can continue to expect high profit margins from slipper sales.

The boots and shoes, however, are in the ↘ decline phase. You should not expect an increase in sales for these products unless new features of these products are introduced. Since the overall percent sales change is down by 6% ($62,500 to $58,900), new features or products need to be developed and introduced in the market to maintain market share. Open the Code Editor window by selecting the Create Code task. Enter the SAS code below to create Table 5.12. Select `Code ► Run on Local` from the toolbar to submit your program.

Table 5.12 Percent Sales Change by Product and Year

	Sales by Year					
	2001		2002			
PRODUCT	N	Total	N	Total	Net Growth	% Change
Boots	2,610	$20,100	2,772	$15,800	162	-21
Shoes	2,270	$22,700	1,910	$21,000	-360	-7
Slippers	3,940	$19,700	4,420	$22,100	480	12
Total	8,820	$62,500	9,102	$58,900	282	-6

Enter the SAS code for Table 5.12.

```
                        0-high = " 009%";
quit;

proc report data=mydata.sales nowd split="*" headline;

 column product year,(quantity sales) n_growth s_growth;

 define product / group 'PRODUCT';
 define year / across 'Sales by Year' center;
 define quantity / sum 'N' center format=comma6.;
 define sales / sum 'Total' format=dollar8.;
 define n_growth / computed 'Net Growth' width=5 format=comma6.;
 define s_growth / computed '% Change' width=5 format=pctf.;

 compute n_growth;
  n_growth = (_c4_ - _c2_);
 endcomp;

compute s_growth;
  s_growth = ((_c5_ - _c3_)/_c3_)*100;
 endcomp;

 rbreak after / summarize skip ol ul;

 compute after;
   product = 'Total';
 endcomp;
```

Answer questions such as: Which products had the best percent sales change by quarter?

Upon further investigation of the slipper sales, you can analyze the percent change by quarter. The percent change by quarter is calculated by taking the difference between the sales of the current quarter and the previous quarter. For example, there is a 13% decrease from the first quarter ($2,300) sales to the second quarter ($2,000) sales in 2001.

Looking at the results in Table 5.13, you can see that the best quarterly increases were in Third Quarter 2001 (185%) and Third Quarter 2002 (81%). The increases from the previous quarter in each year are $3,700 and $2,600 respectively. In both years, sales were seasonally strong with the fourth quarter having the highest sales

($9,700, $8,600). From the _year-to-date_ value in 2001, the cumulative total, and percent change, you can see that nearly half of all sales (51% to 100%) are in the fourth quarter. To make the changes displayed in Table 5.13, Open the Code Editor window by selecting the Create Code task. Enter the SAS code below. Select `Code►Run on Local` from the toolbar to submit your program.

Table 5.13 Percent Change of Slipper Sales by Year

			SALES		_QUARTERLY CHANGE_		_YEAR-TO-DATE_	
YEAR	**QUARTER**	**N**	**Total**	**Percent**	**Total**	**PERCENT**	**YTD**	**PERCENT**
2001	1	460	$2,300	12%	.	.	$2,300	12%
	2	400	$2,000	10%	$-300	(13%)	$4,300	22%
	3	1,140	$5,700	29%	$3,700	185%	$10,000	51%
	4	1,940	$9,700	49%	$4,000	70%	$19,700	100%
Sales Total for	2001	3,940	$19,700	100%				
2002	1	900	$4,500	20%	.	.	$4,500	20%
	2	640	$3,200	14%	$-1,300	(29%)	$7,700	35%
	3	1,160	$5,800	26%	$2,600	81%	$13,500	61%
	4	1,720	$8,600	39%	$2,800	48%	$22,100	100%
Sales Total for	2002	4,420	$22,100	100%				
Grand Sales Total		8,360	$41,800					

Enter the SAS code for Table 5.13.

```
proc report data=mydata.sales nowd split="*" headline;

 where product = 'Slippers';

 column
  ( year quarter quantity)
  (" SALES " sales pct)
  (" QUARTERLY CHANGE " diff diff_pct)
  ("_YEAR-TO-DATE_" ytd cumpct)
 ;

 define year / group 'YEAR';
 define quarter / group 'QUARTER' center format=quarterv.;
 define quantity / sum 'N' center format=comma6.;

 define sales / computed 'Total' sum format=dollar8.;
 define pct / computed 'Percent' format=percent8.;

 define diff / computed 'Total' center format=dollar8.;
 define diff_pct / computed 'PERCENT' format = percent8.;

 define ytd / computed 'YTD' format = dollar8.;
 define cumpct / computed 'PERCENT' format=percent8.;

 compute before year;
  r = 0;
  last = 0;
  total = sales.sum;
  cumtot=0;
 endcomp;

 compute diff;
  r+1;
  if r=1 then diff = .;
  if r>1 then do;
   last = sales.sum;
   if  break  eq "    " then diff = sales.sum-py;
   if  break  eq "   " then last = sales.sum;
   if _break_ ne "     " then diff = .;
  end;
  py = sales.sum;
 endcomp;
```

```
  diff_pct = (diff / (last - diff));
 endcomp;

* break after year / summarize skip ol ul;

 compute ytd;
  cumtot+sales.sum;
   ytd=cumtot;
   if _break_ = "year" then ytd = .;
   if _break_ = "year" then total = 0;
 endcomp;

 compute cumpct;
  cumpct = ytd/total;
 endcomp;

 compute pct;
  pct = sales.sum/total;
 endcomp;

* break after year / summarize ol ul;

 * add custom totals line instead of default summary line;
 compute after year;
  line @4 80*'=';
  line @4 'Sales Total for    ' +3 year 4.0 +16 quantity.sum comma6.
       +19 sales.sum dollar8. +12 pct percent8.;
  line ' ';
  line ' ';
 endcomp;

compute after;
  line @4 " |"  70*"_"  "|";
  line @4 " |   Grand Sales Total " +20 quantity.sum comma6.
       +18 sales.sum dollar8. "|";
  line @4 " |"  70*"_"  "|";
 endcomp;
```

Different Viewpoint: Percent Change of Product Sales by Year

Once the report is created, you can modify the code to analyze different viewpoints of the information. By changing the code to include all products, you can gain a better understanding of the percent change across all products.

From the results of Table 5.14, you can see that second half of 2002 is better than first half. The best quarterly increases are Third Quarter 2001 (104 %) and Second Quarter 2002 (117 %).

Table 5.14 Percent Change of Product Sales by Year

			SALES		_QUARTERLY CHANGE_		_YEAR-TO-DATE_	
YEAR	QUARTER	N	Total	Percent	Total	PERCENT	YTD	PERCENT
2001	1	1,010	$6,900	11%	.	.	$6,900	11%
	2	1,374	$10,200	16%	$3,300	48%	$17,100	27%
	3	2,817	$20,800	33%	$10,600	104%	$37,900	61%
	4	3,619	$24,600	39%	$3,800	18%	$62,500	100%
Sales Total for	2001	8,820	$62,500	100%				
2002	1	1,514	$5,300	9%	.	.	$5,300	9%
	2	1,555	$11,500	20%	$6,200	117%	$16,800	29%
	3	2,704	$17,600	30%	$6,100	53%	$34,400	58%
	4	3,329	$24,500	42%	$6,900	39%	$58,900	100%
Sales Total for	2002	9,102	$58,900	100%				
Grand Sales Total		17,922	$121,400					

The only change required to create this table is to remove the WHERE condition from example code that produced Table 5.13. This will include all products. Do this **before** running the task.

Creating Summary Reports across Time Periods (Quarter, Year)

Answer questions such as:

- What was the overall sales performance by quarter and product, customer, or region?
- How did a customer's product sales compare with other customers' product sales?
- Was there a pattern from quarter to quarter or year to year?

You can also analyze sales across time periods. With this flexibility, you can determine how much volume is sold in each quarter of the year. Is there, for example, a seasonal pattern where more sales happen toward the end of the year due to the Christmas holidays?

In this section, you look at 2002 sales by product and quarter. You can compare each product across quarters to see any patterns in the data. In Table 5.15, notice that boots and slippers have the highest sales of $8,400 and $8,600 in the fourth quarter while shoes have the highest sales of $6,200 in the third quarter.

To compare any two months, you first need to query the data set that contains the required two months. See the Query window in Chapter 3 for examples of how to do this.

This approach gives you the flexibility to compare any two time periods. Open the Code Editor window by selecting the Create Code task. Enter the SAS code below to create this table. Select `Code►Run on Local` from the toolbar to submit your program.

Table 5.15 2002 Sales by Product and Quarter

	2002 by QUARTER							
	1		2		3		4	
PRODUCT	N	Total	N	Total	N	Total	N	Total
Boots	374	$-1,600	425	$3,400	924	$5,600	1,049	$8,400
Shoes	240	$2,400	490	$4,900	620	$6,200	560	$7,500
Slippers	900	$4,500	640	$3,200	1,160	$5,800	1,720	$8,600
Total	1,514	$5,300	1,555	$11,500	2,704	$17,600	3,329	$24,500

Enter the SAS code for Table 5.15.

```
proc report data=mydata.sales nowd split="*" headline;

 where year = 2002;

 column product quarter,(quantity sales) ;

 define product / group 'PRODUCT';
 define quarter / across '2002 by QUARTER' center format=quarterv.;

 define quantity / sum 'N' center format=comma6.;
 define sales / sum 'Total' format=dollar8.;

 rbreak after / summarize skip ol ul;

 compute after;
   product = 'Total';
 endcomp;
```

Different Statistics: Mean Sales

By changing the statistics of the same data, you can display **mean sales** instead of total sales.

Table 5.16 2002 Mean Sales by Product and Quarter

	2002 by QUARTER							
	1		2		3		4	
PRODUCT	N	Mean	N	Mean	N	Mean	N	Mean
Boots	374	$-533	425	$1,133	924	$1,867	1,049	$2,800
Shoes	240	$800	490	$1,633	620	$2,067	560	$2,500
Slippers	900	$1,500	640	$1,067	1,160	$1,933	1,720	$2,867

The only modification required to create this table is to change `sum` to `mean` and remove the `rbreak` and `compute` block code in the code that produced Table 5.15.

Different Viewpoint: Sales by Customer and Quarter

By changing the viewpoint of the same data to be grouped and sorted by customer, you can see in Table 5.17 which customer bought which products in each quarter of 2002.

Table 5.17 2002 Sales by Customer and Quarter

	2002 by QUARTER							
	1		2		3		4	
CUSTOMER	N	Total	N	Total	N	Total	N	Total
TwoFeet	.	.	.	.	1,207	$8,000	.	.
Acme	1,514	$5,300	1,555	$11,500	1,497	$9,600	2,492	$18,300
BigX	.	.	.	.	.	.	837	$6,200
	1,514	$5,300	1,555	$11,500	2,704	$17,600	3,329	$24,500

The only change needed in order to create this table is in the COLUMN and DEFINE statements. Change the variable PRODUCT to CUSTOMER. No other change is required. Select `Code ► Run on Local` from the toolbar to submit your program. Click the **Yes** button to replace the results from the previous run.

Enter the SAS code for Table 5.17.

Different Viewpoint: Sales by Region and Quarter

By changing the viewpoint of the same data to be grouped and sorted by region, you can see in Table 5.18 which region had the highest quarterly sales in 2002.

Table 5.18 2002 Sales by Region and Quarter

	2002 by QUARTER							
	1		2		3		4	
REGION	N	Total	N	Total	N	Total	N	Total
Northern	1,514	$5,300	1,555	$11,500	1,497	$9,600	2,492	$18,300
Southern	.	.	.	.	1,207	$8,000	.	.
Western	.	.	.	.	.	.	837	$6,200
	1,514	$5,300	1,555	$11,500	2,704	$17,600	3,329	$24,500

The only change required to create this table is in the COLUMN and DEFINE statements. Change the variable CUSTOMER to REGION. No other change is required. In addition, add REGIONV. as the format. Select `Code►Run on Local` from the toolbar to submit your program. Click the **Yes** button to replace the results from the previous run.

Enter the SAS code for Table 5.18.

To compare customer product sales against each other, you need to combine customers and products as unique groups. This process provides the detail information required. Table 5.19 shows that Acme is one of the top customers purchasing slippers ($17,500; i.e., $4,500 + $3,200 + $2,600 + $7,200), while customers TwoFeet and BigX have not purchased any products in quarters 1, 2, and 4 and quarters 1, 2, and 3 respectively. From this information, you can clearly identify Acme as a repeat customer. Open the Code Editor window by selecting the Create Code task. Enter the SAS code below to create this table. Select `Code`►`Run on Local` from the toolbar to submit your program.

Table 5.19 2002 Sales by Customer, Product, and Quarter

		2002 by QUARTER							
		1		2		3		4	
CUSTOMER	PRODUCT	N	Total	N	Total	N	Total	N	Total
TwoFeet	Boots	.	.	.	.	437	$3,500	.	.
	Shoes	.	.	.	.	130	$1,300	.	.
	Slippers	.	.	.	.	640	$3,200	.	.
TwoFeet		.	.	.	.	1,207	$8,000	.	.
Acme	Boots	374	-$1,600	425	$3,400	487	$2,100	662	$5,300
	Shoes	240	$2,400	490	$4,900	490	$4,900	390	$5,800
	Slippers	900	$4,500	640	$3,200	520	$2,600	1,440	$7,200
Acme		1,514	$5,300	1,555	$11,500	1,497	$9,600	2,492	$18,300
BigX	Boots	.	.	.	.	.	.	387	$3,100
	Shoes	.	.	.	.	.	.	170	$1,700
	Slippers	.	.	.	.	.	.	280	$1,400
BigX		.	.	.	.	.	.	837	$6,200
Total		1,514	$5,300	1,555	$11,500	2,704	$17,600	3,329	$24,500

Enter the SAS code for Table 5.19.

```
proc report data=mydata.sales nowd split="*" headline;

 where year = 2002;
 column customer product quarter,(quantity sales) ;

 define customer / group 'CUSTOMER' format=$10.;
 define product / group 'PRODUCT';

 define quarter / across '2002 by QUARTER' center format=quarterv.;
 define quantity / sum 'N' center format=comma6.;
 define sales / sum 'Total' format=dollar8.;

 break after customer / summarize skip ol ul;
 rbreak after / dol dul skip summarize;
 compute after;
  customer='Total';
 endcomp;
```

Different Viewpoint: Sales by Region, Product, and Quarter

When combining regions with products, you know where most of your sales are coming from. Table 5.20 shows that the Northern region is better than the other regions. From reviewing Table 3.7, you can determine that this is due to the lack of catalog sales in the Southern and Western regions.

Table 5.20 2002 Sales by Region, Product, and Quarter

		2002 by QUARTER							
		1		2		3		4	
REGION	PRODUCT	N	Total	N	Total	N	Total	N	Total
Northern	Boots	374	$-1,600	425	$3,400	487	$2,100	662	$5,300
	Shoes	240	$2,400	490	$4,900	490	$4,900	390	$5,800
	Slippers	900	$4,500	640	$3,200	520	$2,600	1,440	$7,200
Northern		1,514	$5,300	1,555	$11,500	1,497	$9,600	2,492	$18,300
Southern	Boots	.	.	.	.	437	$3,500	.	.
	Shoes	.	.	.	.	130	$1,300	.	.
	Slippers	.	.	.	.	640	$3,200	.	.
Southern		.	.	.	.	1,207	$8,000	.	.
Western	Boots	.	.	.	.	.	.	387	$3,100
	Shoes	.	.	.	.	.	.	170	$1,700
	Slippers	.	.	.	.	.	.	280	$1,400
Western		.	.	.	.	.	.	837	$6,200
	Total	1,514	$5,300	1,555	$11,500	2,704	$17,600	3,329	$24,500

The only changes required to create this table are in the COLUMN and DEFINE statements and the COMPUTE block section. Change the variable CUSTOMER to REGION. In addition, add REGIONV. as the format. No other change is required. Select `Code►Run on Local` from the toolbar to submit your program. Click the **Yes** button to replace the results from the previous run.

Enter the SAS code change for Table 5.20.

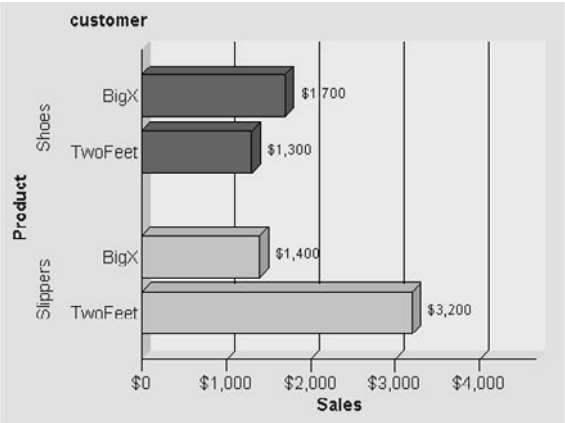
```
 define region / group 'REGION' format=regionv.;


rbreak after / dol dul skip summarize;
 compute after;
  region='Total';
 endcomp;
```

Customized Horizontal Bars

Answer questions such as: When comparing shoes and slipper sales for selected customers, which customers had higher sales?

Each of the previous examples in this chapter shows how to modify or write your own SAS code to create tables. You can also modify or write SAS code to create **graphs**. In the following example, you modify the bar chart in Figure 4.6 to exclude Acme sales. The figure now displays 2002 shoes and slipper sales by BigX and TwoFeet.

Figure 5.2 Shoes and Slipper Sales by BigX and TwoFeet



The only change required is to copy the SAS code from Figure 4.6 and add the WHERE statement. The WHERE statement excludes Acme sales. Select `Code►Run on Local` from the toolbar to submit your program.

```
GOPTIONS xpixels=&_EGCHARTWIDTH ypixels=&_EGCHARTHEIGHT;

/* Symbol Statements */
      ;

/* Axes Statements */
AXIS1
      label=('customer')
      MINOR=NONE
      ;

AXIS2
      label=('Sales')
      MINOR=NONE
      ;

TITLE "Figure 5.2 Product Sales by BigX and TwoFeet";

FOOTNOTE "Generated by the SAS System (& EGSERVERNAME, &SYSSCPL) on
%SYSFUNC(DATE(), EURDFDE9.) at %SYSFUNC(TIME(), TIMEAMPM8.)";

PROC gchart
DATA=SASUSER.QURY4769
      ;
      where customer ^= 'Acme';
      hbar3d customer /
      GROUP=product
      SUMVAR=sales
      RAXIS=AXIS2
      MAXIS=AXIS1
      AUTOREF
      SHAPE=Block
      PATTERNID=GROUP
      TYPE=sum
      COUTLINE=BLACK
      FRAME
      DESCRIPTION="Figure 5.2 Product Sales by Customer using Bar Chart"
      ;

RUN;QUIT;
```

```
/* End of task code. */
```

Migrating SAS Code to Production

All SAS code generated from SAS Enterprise Guide tasks can be saved as SAS programs. Any result created within SAS Learning Edition can be saved as separate file types such as HTML or PDF that can be shared with your colleagues. In addition, the SAS Learning Edition project that contains the results of all tasks can be shared.

While the 1000-observation limit within SAS Leaning Edition prevents the analysis of large production data sets, the SAS code developed in SAS Learning Edition can be directly used in SAS production environments. Prototype analysis within SAS Learning Edition using smaller representative samples of the production data sets facilitates the development of programs for larger data sets in production. Remember that the SAS Learning Edition, however, should be used for learning purposes only.

Appendix

The examples in this book are based on a SAS data set called Sales. The SAS statements needed to create the Sales data set are provided in this appendix. In addition, the SAS statements used to create formats for the variables in the Sales data set are provided in the PROC FORMAT statements below.

You can type the statements in this appendix into SAS Enterprise Guide to create the Sales data set and the format catalog. You can also download the statements to create the data set and format catalog from the Companion Web site for this book by going to

support.sas.com/companionsites

Select the title *Data Management and Reporting Made Easy with SAS Learning Edition 2.0* to display its Companion Web Site, and then select the link for data.

```
*************************************************************************;
* Reference location of Sales data set;
* You will need to create the folder sas_le_data on your C: drive;
*************************************************************************;

libname mydata 'c:\sas_le_data';


*************************************************************************;
* Create format catalog;
* You can copy PROC FORMAT code to the create code task to create
* formats;
*************************************************************************;

proc format;

 value regionv
       1 = 'Northern'
       2 = 'Southern'
       3 = 'Eastern'
       4 = 'Western'
       ;
```

```
 value district
       1 = 'North East'
       2 = 'South East'
       3 = 'South West'
       4 = 'North West'
       5 = 'Central West'
       ;

 value marketv
      1 = 'Catalog'
      2 = 'Retail'
      3 = 'Wholesale'
      4 = 'Other'
      ;


* Can apply to quarter variable;
 value quarterv
       1 = '1st'
       2 = '2nd'
       3 = '3rd'
       4 = '4th'
       ;

* Can apply to month variable;
 value monthv
       1 = 'JAN'
       2 = 'FEB'
       3 = 'MAR'
       4 = 'APR'
       5 = 'MAY'
       6 = 'JUN'
       7 = 'JUL'
       8 = 'AUG'
       9 = 'SEP'
       10 = 'OCT'
       11 = 'NOV'
       12 = 'DEC'
       ;

picture pctpct low-99.4 = '99%'
              99.5-high = '999%' (prefix=' (');

picture pctf (round) low-<0 = " 009%" (PREFIX="-")
                    0-high = " 009%";

quit;
```

```
*************************************************************************;
* Create sales data set;
* You can code to the create code task to create sales data set;
*************************************************************************;

data mydata.sales;

* Specify the length of character variables to be 10;
length product category customer $10.;

* Label each variable;
label
      product = 'Product'
      category = 'Product Category'
      customer = 'Customer'
      year = 'Year'
      month = 'Month'
      quarter = 'Quarter'
      market = 'Market'
      sales = 'Sales'
      expense = 'Expense'
      region = 'Region'
      district = 'District'
      return = 'Returns'
      quantity = 'Product Quantity'
;

* Assign formats to selected variables;
format market marketv.;
format region regionv.;
format district district.;

* Read data from within the data step;
input product $ category $ customer $ year month quarter market sales
      expense region district return quantity;

* Data to read into the sales data set;
cards;
Shoes      Shoes      Acme     2001 1 1  1   300  240  1  1 0 1  30
Boots      Shoes      Acme     2001 1 1  1  2200 1540  1  1 0 1 275
Slippers   Slippers   Acme     2001 1 1  1   900  540  1  1 0 1 180
Shoes      Shoes      Acme     2001 2 1  1   100   80  1  1 0 1  10
Boots      Shoes      Acme     2001 2 1  1  1400  980  1  1 0 1 175
Slippers   Slippers   Acme     2001 2 1  1     0    0  1  1 0 1   0
Shoes      Shoes      Acme     2001 3 1  1   600  480  1  1 0 1  60
```

Boots	Shoes	Acme	2001	3	1	1	0	0	1	1	0	1	0
Slippers	Slippers	Acme	2001	3	1	1	1400	840	1	1	0	1	280
Shoes	Shoes	Acme	2001	4	2	1	2600	2080	1	1	0	1	260
Boots	Shoes	Acme	2001	4	2	1	1500	1050	1	1	0	1	187
Slippers	Slippers	Acme	2001	4	2	1	1000	600	1	1	0	1	200
Shoes	Shoes	Acme	2001	5	2	2	2400	1920	1	1	0	1	240
Boots	Shoes	Acme	2001	5	2	2	-400	-280	1	1	0	1	50
Slippers	Slippers	Acme	2001	5	2	2	700	420	1	1	0	1	140
Shoes	Shoes	Acme	2001	6	2	2	1000	800	1	1	0	1	100
Boots	Shoes	Acme	2001	6	2	2	1100	770	1	1	0	1	137
Slippers	Slippers	Acme	2001	6	2	2	300	180	1	1	0	1	60
Shoes	Shoes	Acme	2001	7	3	2	3700	2960	1	1	0	1	370
Boots	Shoes	Acme	2001	7	3	2	1600	1120	1	1	0	1	200
Slippers	Slippers	Acme	2001	7	3	2	2800	1680	1	1	0	1	560
Shoes	Shoes	TwoFeet	2001	8	3	2	2800	2240	2	3	3	1	280
Boots	Shoes	TwoFeet	2001	8	3	2	2800	1960	2	3	3	1	350
Slippers	Slippers	TwoFeet	2001	8	3	2	1100	660	2	3	3	1	220
Shoes	Shoes	Acme	2001	9	3	2	1900	1520	1	1	0	1	190
Boots	Shoes	Acme	2001	9	3	2	2300	1610	1	1	0	1	287
Slippers	Slippers	Acme	2001	9	3	2	1800	1080	1	1	0	1	360
Shoes	Shoes	Acme	2001	10	4	2	2100	1680	1	1	0	1	210
Boots	Shoes	Acme	2001	10	4	2	3200	2240	1	1	0	1	400
Slippers	Slippers	Acme	2001	10	4	2	4700	2820	1	1	0	1	940
Shoes	Shoes	BigX	2001	11	4	2	2900	2320	4	5	5	1	290
Boots	Shoes	BigX	2001	11	4	2	2500	1750	4	5	5	1	312
Slippers	Slippers	BigX	2001	11	4	2	3800	2280	4	5	5	1	760
Shoes	Shoes	Acme	2001	12	4	2	2300	1840	1	1	0	1	230
Boots	Shoes	Acme	2001	12	4	2	1900	1330	1	1	0	1	237
Slippers	Slippers	Acme	2001	12	4	2	1200	720	1	1	0	1	240
Shoes	Shoes	Acme	2002	1	1	1	1000	800	1	1	0	1	100
Boots	Shoes	Acme	2002	1	1	1	-1400	-980	1	1	0	1	175
Slippers	Slippers	Acme	2002	1	1	1	1100	660	1	1	0	1	220
Shoes	Shoes	Acme	2002	2	1	1	1000	800	1	1	0	1	100
Boots	Shoes	Acme	2002	2	1	1	700	490	1	1	0	1	87
Slippers	Slippers	Acme	2002	2	1	1	2300	1380	1	1	0	1	460
Shoes	Shoes	Acme	2002	3	1	1	400	320	1	1	0	1	40
Boots	Shoes	Acme	2002	3	1	1	-900	-630	1	1	0	1	112
Slippers	Slippers	Acme	2002	3	1	1	1100	660	1	1	0	1	220
Shoes	Shoes	Acme	2002	4	2	1	2100	1680	1	1	0	1	210
Boots	Shoes	Acme	2002	4	2	1	1400	980	1	1	0	1	175
Slippers	Slippers	Acme	2002	4	2	1	500	300	1	1	0	1	100
Shoes	Shoes	Acme	2002	5	2	2	1100	880	1	1	0	1	110
Boots	Shoes	Acme	2002	5	2	2	200	140	1	1	0	1	25
Slippers	Slippers	Acme	2002	5	2	2	1300	780	1	1	0	1	260
Boots	Shoes	Acme	2002	6	2	2	1800	1260	1	1	0	1	225
Shoes	Shoes	Acme	2002	6	2	2	1700	1360	1	1	0	1	170
Slippers	Slippers	Acme	2002	6	2	2	1400	840	1	1	0	1	280

```
Boots      Shoes      Acme     2002 7   3  2  -900 -630  1  1 0 1 112
Shoes      Shoes      Acme     2002 7   3  2  1700 1360  1  1 0 1 170
Slippers   Slippers   Acme     2002 7   3  2  1400  840  1  1 0 1 280
Boots      Shoes      TwoFeet  2002 8   3  2  3500 2450  2  3 3 1 437
Shoes      Shoes      TwoFeet  2002 8   3  2  1300 1040  2  3 3 1 130
Slippers   Slippers   TwoFeet  2002 8   3  2  3200 1920  2  3 3 1 640
Boots      Shoes      Acme     2002 9   3  2  3000 2100  1  1 0 1 375
Shoes      Shoes      Acme     2002 9   3  2  3200 2560  1  1 0 1 320
Slippers   Slippers   Acme     2002 9   3  2  1200  720  1  1 0 1 240
Boots      Shoes      Acme     2002 10  4  2  3000 2100  1  1 0 1 375
Shoes      Shoes      Acme     2002 10  4  2  4000 2400  1  1 0 1 210
Slippers   Slippers   Acme     2002 10  4  2  4000 2400  1  1 0 1 800
Boots      Shoes      BigX     2002 11  4  2  3100 2170  4  5 5 1 387
Shoes      Shoes      BigX     2002 11  4  2  1700 1360  4  5 5 1 170
Slippers   Slippers   BigX     2002 11  4  2  1400  840  4  5 5 1 280
Boots      Shoes      Acme     2002 12  4  2  2300 1610  1  1 0 1 287
Shoes      Shoes      Acme     2002 12  4  2  1800 1440  1  1 0 1 180
Slippers   Slippers   Acme     2002 12  4  2  3200 1920  1  1 0 1 640
;
run;


*************************************************************************;
* Check sales data set structure;
*************************************************************************;

proc contents data = mydata.sales;
run;


*************************************************************************;
* Check sales data set contents;
*************************************************************************;

proc print data = mydata.sales;
run;
```

Glossary

ColPctSum	A column percent calculation (used in the Summary Tables task).
Format	A pattern or set of instructions SAS uses to determine how the values of a variable (or column) should be written or displayed.
Join	The combination of data in two or more SAS data sets using common variables. This is similar to the term *merge*.
Library	A location where a collection of files such as SAS data sets is stored.
Libref	A name used to reference a directory that stores a collection of files such as SAS data sets, programs, and catalog files.
Max	A maximum value (used in the Summary Tables task).
Mean	An average of all values (used in the Summary Tables task).
Min	A minimum value (used in the Summary Tables task).
RowPctSum	A row percent calculation (used in the Summary Tables task).
SAS data set	Data is stored as rows and columns and is accessed by SAS Learning Edition tasks or SAS procedures as SAS data sets.
SAS procedure	A stored process that performs a task such as sorting data or calculating mean values.
Task	A menu option to perform each of the following: Data Access, Data Management, and Data Analysis.
Variable	Represents one or more columns in a SAS data set. Values might be different across records.

References

Books

Carpenter, Art, and Charlie Shipp. 1995. *Quick Results with SAS/GRAPH*. Cary, NC: SAS Institute Inc.

Gupta, Sunil. 2003. *Quick Results with the Output Delivery System*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 1989. *SAS Language and Procedures: Usage, Version 6.* Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2002. *Getting Started with Enterprise Guide, Second Edition*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2004. *SAS Learning Edition 2.0.* Cary, NC: SAS Institute Inc.

Zelazny, Gene. 2001. *Say It with Charts: The Executive's Guide to Visual Communication, Fourth Edition*. New York: McGraw-Hill.

Articles and Papers

Blaha, James. 2003. "SAS Enterprise Guide 2.0." *Proceedings of the Fifteenth Annual Northeastern SAS Users Group Conference*, Washington, DC.

Bruns, Dan. 2000. "The Power and Simplicity of the Tabulate Procedure." *Proceedings of the Twenty-fifth Annual SAS Users Group International Conference*, Indianapolis, IN.

Busche, F. D., K. Boothe, and T. Everly. 2003. "Using Information Effectively to Make More Profitable Decisions: The 10-Letter Solution for Finance." *Proceedings of the Twenty-eighth Annual SAS Users Group International Conference*, Seattle, WA.

Carter, J., S. McDaniel, and M. Porter. 2003. "Developing Custom Analytic Tasks for SAS Enterprise Guide." *Proceedings of the Twenty-eighth Annual SAS Users Group International Conference*, Seattle, WA.

Chapman, David D. 2002. "Using PROC REPORT to Produce Tables with Cumulative Totals and Row Differences." *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Orlando, Florida.

Coleman, Ron. 1998. "The Building Blocks of PROC TABULATE." *Proceedings of the Twenty-third Annual SAS Users Group International Conference*, Nashville, TN.

Corning, Betsy. 1994a. "Designing and Producing Effective Graphs with SAS/GRAPH Software: Using Graphic Elements Effectively." *Observations: The Technical Journal for SAS Software Users*, Volume 20 Number 1, pages 73–86.

Corning, Betsy. 1994b. "Designing and Producing Effective Graphs with SAS/GRAPH Software: Making Graphs Informative." *Observations: The Technical Journal for SAS Software Users*, Volume 20 Number 2, pages 3–17.

Corning, Betsy. 1994c. "Designing and Producing Effective Graphs with SAS/GRAPH Software: Presenting Data Effectively." *Observations: The Technical Journal for SAS Software Users*, Volume 20 Number 3, pages 4–20.

Driscoll, Jeanette. 2003. "Do-It-Yourself Market Analysis." *OfficeSolutions*, May–June.

Farooqi, Ajaz. 2004. "SAS Enterprise Guide: A Quick Overview of Developing, Creating, and Successfully Delivering a Simple Project." *Proceedings of the Twenty-ninth Annual SAS Users Group International Conference*, Montreal, Canada.

Jensen, Edward. 1998. "Tapping the Power of PROC TABULATE for Analysis Reports Clients Understand." *Proceedings of the Sixth Annual Western Users of SAS Software Regional Users Group Conference*, Oakland, CA.

Jiang, Rachel, and Ed Heaton. 2003. "Enterprise Guide: A Code-Free Approach to Data Analysis and Reporting." *Proceedings of the Sixteenth Annual Northeastern SAS Users Group Conference*, Washington, DC.

LeBouton, Kim. 2004. "Getting Up to Speed with PROC REPORT." *Proceedings of the Twelfth Annual Western Users of SAS Software Regional Users Group Conference*, Pasadena, CA.

Linden, Carol, and John Green III. 1994. "Writing Reports with SAS Software: What Are Your Options?" *Observations: The Technical Journal for SAS Software Users*, Volume 20 Number 3.

Noga, Stephen M. 1998. "The Tabulate Procedure: One Step Beyond the Final Chapter." *Proceedings of the Twenty-third Annual SAS Users Group International Conference*, Nashville, TN.

Pass, Ray. 1997. "PROC REPORT: An Introduction to the Batch Language." *Proceedings of the Twenty-second Annual SAS Users Group International Conference*, San Diego, CA.

Ravenna, Andy. 2003. "SAS Enterprise Guide: Getting the Job Done." *Proceedings of the Twenty-eighth Annual SAS Users Group International Conference*, Seattle, WA.

Ritzow, Kim L. 1997. "Advanced Features of PROC Report Workshop." *Proceedings of the Twenty-second Annual SAS Users Group International Conference*, San Diego, CA.

Rud, C. Olivia. 1999. "Statistical Presentations Using the Power of PROC TABULATE." *Proceedings of the Twenty-fourth Annual SAS Users Group International Conference*, Miami Beach, FL.

Schneider, Barbara. 1998. "PROC UNIVARIATE and PROC TABULATE: A Powerful Duo to Produce Descriptive Tables Including Non-parametric Estimates." *Proceedings of the Twenty-third Annual SAS Users Group International Conference*, Nashville, TN.

Stewart, Larry, and Marje Fecht. 2002. "Tips and Tricks for Easier Reporting." *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Orlando, FL.

Suhr, Diana. 2002. "You've Got Data, Now What?" *Proceedings of the Tenth Annual Western Users of SAS Software Regional Users Group Conference*, San Diego, CA.

Walsh, Sue. 2004. "Introduction to SAS Enterprise Guide for Educators." SAS Day at University of California Irvine. April.

Winn, J. Thomas Jr. 1999. "Advanced Features of PROC TABULATE." *Proceedings of the Twenty-fourth Annual SAS Users Group International Conference*, Miami Beach, FL.

Web Sites

Business Intelligence SAS Users Group: http://www.bisug.org/index.html

SAS Enterprise Guide Documentation: http://support.sas.com/documentation/onlinedoc/guide/

SAS Learning Edition: http://support.sas.com/rnd/le/

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y

Books Available from SAS® Press

Advanced Log-Linear Models Using SAS®
by **Daniel Zelterman**

Analysis of Clinical Trials Using SAS®: A Practical Guide
by **Alex Dmitrienko, Geert Molenberghs, Walter Offen,** *and* **Christy Chuang-Stein**

Annotate: Simply the Basics
by **Art Carpenter**

Applied Multivariate Statistics with SAS® Software, Second Edition
by **Ravindra Khattree** *and* **Dayanand N. Naik**

Applied Statistics and the SAS® Programming Language, Fourth Edition
by **Ronald P. Cody** *and* **Jeffrey K. Smith**

An Array of Challenges — Test Your SAS® Skills
by **Robert Virgile**

Carpenter's Complete Guide to the SAS® Macro Language, Second Edition
by **Art Carpenter**

The Cartoon Guide to Statistics
by **Larry Gonick** *and* **Woollcott Smith**

Categorical Data Analysis Using the SAS® System, Second Edition
by **Maura E. Stokes, Charles S. Davis,** *and* **Gary G. Koch**

Cody's Data Cleaning Techniques Using SAS® Software
by **Ron Cody**

Common Statistical Methods for Clinical Research with SAS® Examples, Second Edition
by **Glenn A. Walker**

The Complete Guide to SAS® Indexes
by **Michael A. Raithel**

Debugging SAS® Programs: A Handbook of Tools and Techniques
by **Michele M. Burlew**

Efficiency: Improving the Performance of Your SAS® Applications
by **Robert Virgile**

The Essential PROC SQL Handbook for SAS® Users
by **Katherine Prairie**

Genetic Analysis of Complex Traits Using SAS®
by **Arnold M. Saxton**

A Handbook of Statistical Analyses Using SAS®, Second Edition
by **B.S. Everitt** *and* **G. Der**

Health Care Data and SAS®
by **Marge Scerbo, Craig Dickstein,** *and* **Alan Wilson**

The How-To Book for SAS/GRAPH® Software
by **Thomas Miron**

support.sas.com/pubs

Instant ODS: Style Templates for the Output Delivery System
by **Bernadette Johnson**

In the Know... SAS® Tips and Techniques From Around the Globe
by **Phil Mason**

Integrating Results through Meta-Analytic Review Using SAS® Software
by **Morgan C. Wang**
and **Brad J. Bushman**

Learning SAS® in the Computer Lab, Second Edition
by **Rebecca J. Elliott**

The Little SAS® Book: A Primer
by **Lora D. Delwiche**
and **Susan J. Slaughter**

The Little SAS® Book: A Primer, Second Edition
by **Lora D. Delwiche**
and **Susan J. Slaughter**
(updated to include Version 7 features)

The Little SAS® Book: A Primer, Third Edition
by **Lora D. Delwiche**
and **Susan J. Slaughter**
(updated to include SAS 9.1 features)

Logistic Regression Using the SAS® System: Theory and Application
by **Paul D. Allison**

Longitudinal Data and SAS®: A Programmer's Guide
by **Ron Cody**

Maps Made Easy Using SAS®
by **Mike Zdeb**

Models for Discrete Date
by **Daniel Zelterman**

Multiple Comparisons and Multiple Tests Using SAS® Text and Workbook Set
(books in this set also sold separately)
by **Peter H. Westfall, Randall D. Tobias, Dror Rom, Russell D. Wolfinger,**
and **Yosef Hochberg**

Multiple-Plot Displays: Simplified with Macros
by **Perry Watts**

Multivariate Data Reduction and Discrimination with SAS® Software
by **Ravindra Khattree**
and **Dayanand N. Naik**

Output Delivery System: The Basics
by **Lauren E. Haworth**

Painless Windows: A Handbook for SAS® Users, Third Edition
by **Jodie Gilmore**
(updated to include Version 8 and SAS 9.1 features)

The Power of PROC FORMAT
by **Jonas V. Bilenas**

PROC TABULATE by Example
by **Lauren E. Haworth**

Professional SAS® Programming Shortcuts
by **Rick Aster**

Quick Results with SAS/GRAPH® Software
by **Arthur L. Carpenter**
and **Charles E. Shipp**

Quick Results with the Output Delivery System
by **Sunil Gupta**

Quick Start to Data Analysis with SAS®
by **Frank C. Dilorio**
and **Kenneth A. Hardy**

Reading External Data Files Using SAS®: Examples Handbook
by **Michele M. Burlew**

Regression and ANOVA: An Integrated Approach Using SAS® Software
by **Keith E. Muller**
and **Bethel A. Fetterman**

SAS® Applications Programming: A Gentle Introduction
by **Frank C. Dilorio**

support.sas.com/pubs

SAS® for Forecasting Time Series, Second Edition
by **John C. Brocklebank**
and **David A. Dickey**

SAS® for Linear Models, Fourth Edition
by **Ramon C. Littell, Walter W. Stroup,**
and **Rudolf Freund**

SAS® for Monte Carlo Studies: A Guide for Quantitative Researchers
by **Xitao Fan, Ákos Felsővályi, Stephen A. Sivo,**
and **Sean C. Keenan**

SAS® Functions by Example
by **Ron Cody**

SAS® Guide to Report Writing, Second Edition
by **Michele M. Burlew**

SAS® Macro Programming Made Easy
by **Michele M. Burlew**

SAS® Programming by Example
by **Ron Cody**
and **Ray Pass**

SAS® Programming for Researchers and Social Scientists, Second Edition
by **Paul E. Spector**

SAS® Programming in the Pharmaceutical Industry
by **Jack Shostak**

SAS® Survival Analysis Techniques for Medical Research, Second Edition
by **Alan B. Cantor**

SAS® System for Elementary Statistical Analysis, Second Edition
by **Sandra D. Schlotzhauer**
and **Ramon C. Littell**

SAS® System for Mixed Models
by **Ramon C. Littell, George A. Milliken, Walter W. Stroup,** *and* **Russell D. Wolfinger**

SAS® System for Regression, Second Edition
by **Rudolf J. Freund**
and **Ramon C. Littell**

SAS® System for Statistical Graphics, First Edition
by **Michael Friendly**

The SAS® Workbook and *Solutions* Set
(books in this set also sold separately)
by **Ron Cody**

Selecting Statistical Techniques for Social Science Data: A Guide for SAS® Users
by **Frank M. Andrews, Laura Klem, Patrick M. O'Malley, Willard L. Rodgers, Kathleen B. Welch,**
and **Terrence N. Davidson**

Statistical Quality Control Using the SAS® System
by **Dennis W. King**

A Step-by-Step Approach to Using the SAS® System for Factor Analysis and Structural Equation Modeling
by **Larry Hatcher**

A Step-by-Step Approach to Using SAS® for Univariate and Multivariate Statistics, Second Edition
by **Norm O'Rourke, Larry Hatcher,**
and **Edward J. Stepanski**

Step-by-Step Basic Statistics Using SAS®: Student Guide and *Exercises*
(books in this set also sold separately)
by **Larry Hatcher**

Survival Analysis Using the SAS® System: A Practical Guide
by **Paul D. Allison**

Tuning SAS® Applications in the OS/390 and z/OS Environments, Second Edition
by **Michael A. Raithel**

Univariate and Multivariate General Linear Models: Theory and Applications Using SAS® Software
by **Neil H. Timm**
and **Tammy A. Mieczkowski**

Using SAS® in Financial Research
by **Ekkehart Boehmer, John Paul Broussard,**
and **Juha-Pekka Kallunki**

Using the SAS® Windowing Environment: A Quick Tutorial
by **Larry Hatcher**

Visualizing Categorical Data
by **Michael Friendly**

Web Development with SAS® by Example
by **Frederick Pratter**

Your Guide to Survey Research Using the SAS® System
by **Archer Gravely**

JMP® Books

JMP® for Basic Univariate and Multivariate Statistics: A Step-by-Step Guide
by **Ann Lehman, Norm O'Rourke, Larry Hatcher,** *and* **Edward J. Stepanski**

JMP® Start Statistics, Third Edition
by **John Sall, Ann Lehman,** *and* **Lee Creighton**

Regression Using JMP®
by **Rudolf J. Freund, Ramon C. Littell,** *and* **Lee Creighton**

support.sas.com/pubs