MathSoft

Getting Started with S-PLUS 6.0 for UNIX

October 2000

Data Analysis Division
MathSoft, Inc.
Seattle, Washington

Proprietary Notice

MathSoft, Inc. owns both this software program and its documentation. Both the program and documentation are copyrighted with all rights reserved by MathSoft.

The correct bibliographical reference for this document is as follows:

Getting Started with S-PLUS 6.0 for UNIX, Data Analysis Division, MathSoft, Seattle, WA.

Printed in the United States.

Copyright Notice

CONTENTS

TUTORIAL 1

INTRODUCTION

This tutorial is designed to acquaint you with S-PLUS 6.0 for UNIX and Linux. It includes the following information:

- A quick tour of the new Java-based graphical user interface for S-PLUS.
- An in-depth example using S-PLUS to analyze an environmental data set.
- A command-line tutorial, which introduces some new graphics devices.

Release Notes for S-PLUS 6.0 can be found online in the text file **$SHOME/RELNOTES.TXT** . Installation notes can be found in **$SHOME/INSTALL.TXT** , and a list of bugs fixed for the 6.0 release can be found in **$SHOME/FIXEDBUG.TXT** .

Before running S-PLUS the first time, you should create a *working directory* specifically for S-PLUS. This directory will contain any files you want to read into or export from S-PLUS, as well as a **.Data** directory to hold your S-PLUS data objects, metadata objects, and help files. These working directories are called *chapters*, and are created with the S-PLUS **CHAPTER** utility. The first time you run S-PLUS, it creates a chapter called **MySwork** which can function as a default working directory; however, it will also store more general user information. MathSoft recommends creating at least one chapter separate from **MySwork**, and using that for your day-to-day S-PLUS work.

To create a working directory named **myproj** in your home directory, type the following sequence of commands at the UNIX shell prompt and press RETURN after each command:

```
cd
mkdir myproj
cd myproj
Splus CHAPTER
```

The **CHAPTER** utility creates a **.Data** directory, which in turn contains three other directories at start-up: **__Meta**, **__Shelp**, and **__Hhelp**. The **.Data** directory contains your normal data sets and functions, the **__Meta** directory contains S-PLUS metadata such as method definitions, and the two __***help** directories contain SGML and HTML versions of help files you create for your functions. All of these databases are initially empty, except for some possible marker files.

QUICK TOUR

S-PLUS is a powerful package for analyzing data using graphics and statistics. You can import from and export to many data sources, including analytical software such as SAS, SPSS, and Matlab; spreadsheets such as Excel and Lotus; and a variety of text formats.

Once you have accessed your data, you can analyze and explore it using S-PLUS tools. In this quick tour, you perform the following tasks:

- Open a data set.
- Create several two-dimensional plots.
- Fit a linear model to your two-dimensional data.
- Create a three-dimensional plot.

Using the menus and dialogs in the graphical user interface, this quick tour briefly introduces you to a few of the most commonly used procedures in S-PLUS.

Starting S-PLUS

To start the S-PLUS graphical user interface, launch S-PLUS from a UNIX shell prompt as follows:

```
Splus -g &
```

S-PLUS appears in its own window, with a main menu, toolbar and an open **Commands** window, as shown in Figure 1.1.

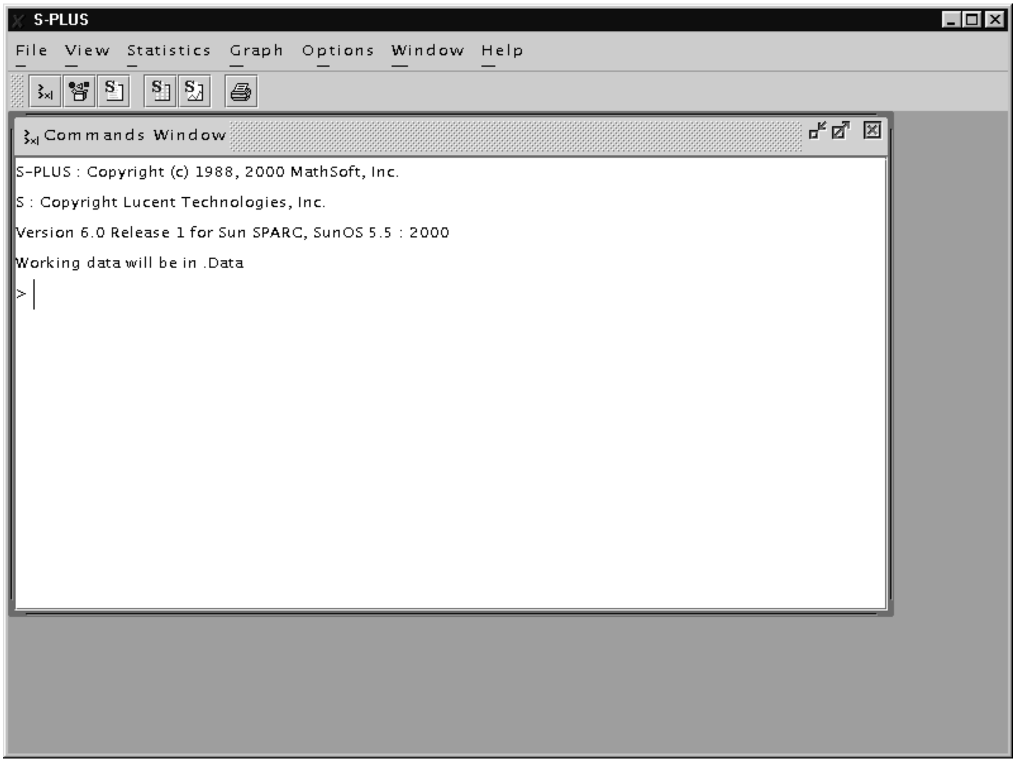

Figure 1.1: *The S-PLUS application window.*

Getting Data

Let's walk through a sample session to help you decide which new car you should buy.

1. From the main menu, select **View ▶ New Data Viewer**. The **New Data Viewer** dialog appears.
2. Type `fuel.frame` in the **Data Set** field.
3. Click **OK**. The `fuel.frame` data is loaded into a **Data Viewer** window.

The `fuel.frame` data set consists of five data columns plus a column of row names:

- `Weight`: automobile weight. This column is numeric.
- `Disp.`: engine displacement (6 liter, 8 liter, etc.). This column is numeric.
- `Mileage`: mileage in units of miles per gallon. This column is numeric.
- `Fuel`: 100/`Mileage`. This column is numeric.
- `Type`: category of vehicle (`Large`, `Medium`, `Small`, `Compact`, `Sporty`, `Van`). This column is a factor variable.

Creating a 2D Graph

A scatterplot matrix shows the relationship between each pair of variables in a data set. This is often a useful preliminary view of multivariate data.

To create a scatterplot matrix, do the following:

1. From the **Graph** menu, choose **Multiple Variables ▶ Scatterplot Matrix**. The **Scatterplot Matrix** dialog appears.
2. Type `fuel.frame` in the **Data Set** field.
3. Select **<ALL>** from the **Value** list box and **<NONE>** from the **Conditioning** list box.
4. Click **OK**. A **Graph** window appears displaying the scatterplot matrix shown in Figure 1.2.

A scatterplot matrix displays each column of data against the other selected columns. For example, to see how `Mileage` and `Fuel` are related in the `fuel.frame` data, read across the **Graph** window from `Mileage` and above `Fuel` to see the plot. The plot shows that `Mileage` and `Fuel` are directly related. You can also see a strong relationship between `Mileage` and `Weight`: heavier cars have lower mileage.

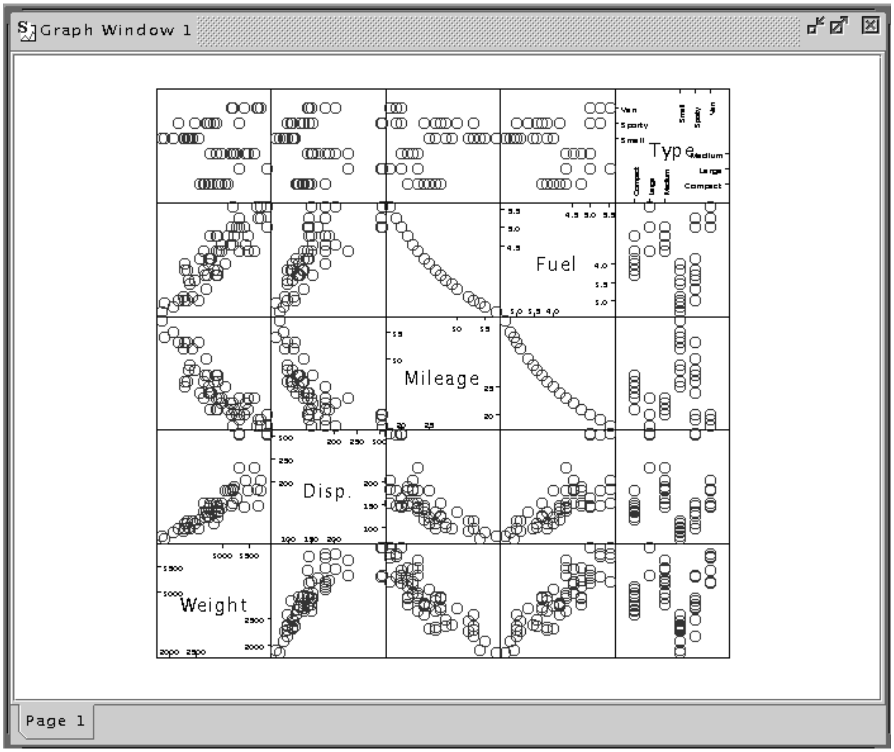

Figure 1.2: *A scatterplot matrix of the* `fuel.frame` *data shows a number of strong relationships.*

Linear Regression

Now that you're familiar with the `fuel.frame` data, let's examine the relationship between `Weight` and `Mileage` a bit more extensively.

1. Close the **Graph** window containing the scatterplot matrix.
2. From the **Graph** menu, choose **Scatter Plot**. The **Scatter Plot** dialog appears.
3. Type `fuel.frame` in the **Data Set** field.
4. Choose `Weight` as the **x Axis Value** and `Mileage` as the **y Axis Value**.

5. Click the **Fit** tab to move to the **Fit** page of the dialog. Choose **Least Squares** as the **Regression Type**.
6. Click **Apply** to create the plot. The dialog remains open.

This linear fit, displayed in Figure 1.3, shows an obvious inverse relationship: as `Weight` increases, `Mileage` decreases.

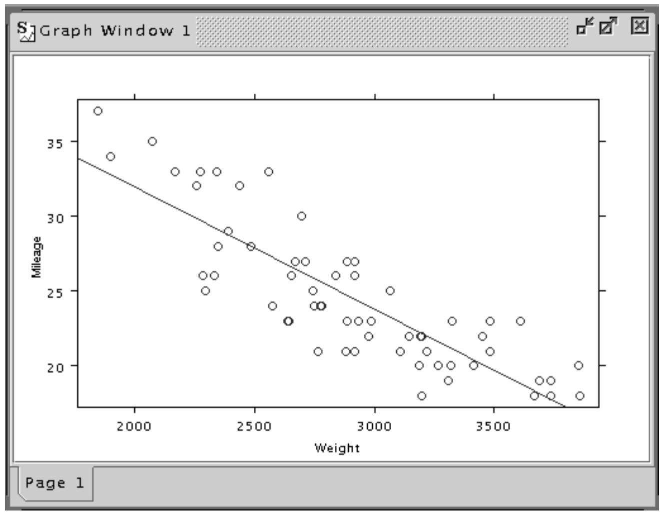

Figure 1.3: *A linear fit of* `Mileage` *vs.* `Weight` *in the* `fuel.frame` *data.*

To examine how `Vans` or `Compact` cars fit into this example, you can use MathSoft's exclusive Trellis graphics to condition `Weight` and `Mileage` on a third variable, `Type`.

1. Click on the **Data** tab in the open **Scatter Plot** dialog to return to the **Data** page.
2. Choose **Type** from the **Conditioning** list box.
3. Click **OK**.

The resulting plot is shown in Figure 1.4. The data are divided into subsamples, conditioned by `Type`. You can now see additional relationships in the data:

- `Sporty` cars, normally assumed to be gas guzzlers, actually have among the highest mileage, along with `Small` cars.
- `Compact` and `Medium` cars, often touted for higher mileage, get gas mileage similar to `Large` cars.

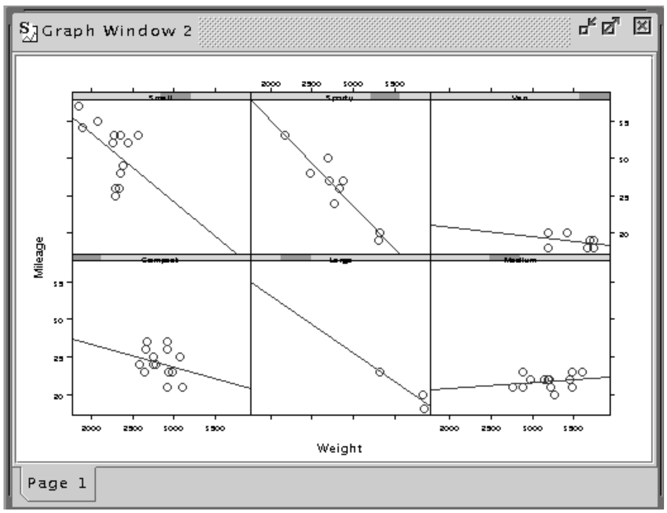

Figure 1.4: *A Trellis view of the `fuel.frame` data.*

Fitting a linear regression model

As shown in Figure 1.3, a line fits the `Mileage` data reasonably well. To create this fit analytically, proceed as follows:

1. From the **Statistics** menu, choose **Regression ▶ Linear**. The **Linear Regression** dialog appears.
2. Choose `Mileage` as the **Dependent Variable** and `Weight` as the **Independent Variable**.
3. Click **OK**.

The output is displayed in a **Report** window, as shown in Figure 1.5.

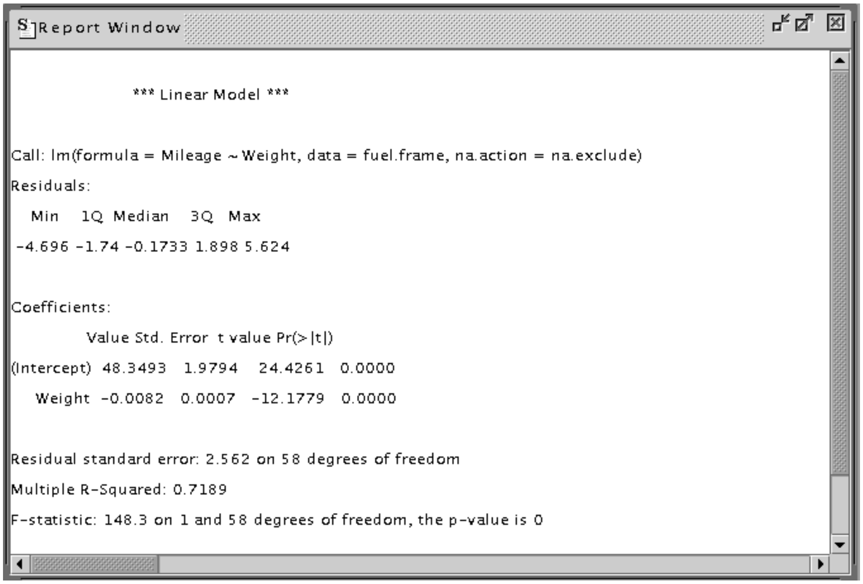

Figure 1.5: *A **Report** window showing a linear fit for the* `fuel.frame` *data.*

Creating a 3D Graph

S-PLUS offers a variety of three-dimensional plot types for powerful data visualization. To create a 3D graph, we'll use the `galaxy` data set. The `galaxy` data contains measurements of the radial velocity of a spiral galaxy measured at 323 points in the sky.

1. From the **Graph** menu, choose **Three Variables ▶ Cloud Plot**. The **Cloud Plot** dialog appears.
2. Type `galaxy` in the **Data Set** field.
3. Choose `east.west`, `north.south`, and `velocity`, respectively, as the **x Axis Value**, **y Axis Value**, and **z Axis Value**.
4. Click **OK**. The resulting plot is shown in Figure 1.6.

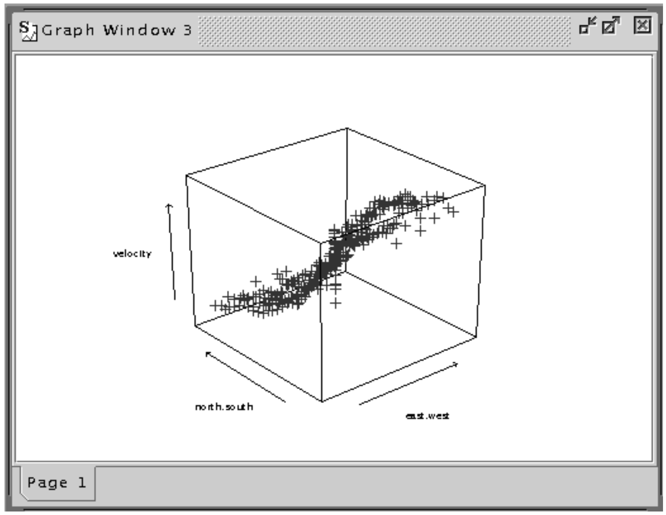

Figure 1.6: *A point cloud of the* `galaxy` *data.*

EXTENDED TOUR: EXAMINING ENVIRONMENTAL DATA

In this extended example, we import data from a SAS file. The data set contains measurements over 111 days in 1973 on ozone concentration, solar radiation, daily maximum temperature, and wind speed in the New York metropolitan area. We visually explore the data with standard and Trellis plots, and we then fit a linear model to the data. We also modify the plots for possible use in a presentation.

Importing Data

Import the environmental data from its SAS file with the following steps:

1. If you have any windows open from the Quick Tour in the previous section, close them before continuing.
2. Choose **File ▶ Import Data**.
3. Type the path to **$SHOME/library/example5/exenvirn.sd2** in the **File Name** field. Alternatively, click on the **Browse** button to navigate to the directory that contains the data file. You can find your current **SHOME** by typing `getenv("SHOME")` in the **Commands** window.
4. Under **File Format**, choose **SAS – Windows/OS2**.
5. Type `envirn` in the **Save As** field.
6. Click **OK** to import the file into S-PLUS.

Creating a 2D Graph

We are ready to visualize the data. We first create a local regression plot of the data:

1. From the **Graph** menu, choose **Scatter Plot**. The **Scatter Plot** dialog appears.
2. Choose `envirn` from the **Data Set** drop-down list.
3. Select **radiatio** as the **x Axis Value** and **ozone** as the **y Axis Value**.
4. Click the **Fit** tab to move to the **Fit** page of the dialog.

5. Choose **Loess** from the **Smoothing Type** drop-down list.
6. Type a variety of values between 0.1 and 0.9 in the **Span** field and click **Apply** to view the results. Reset the **Span** value to **0.75**, and click **Apply**.

Changing Graph Features

The S-PLUS dialogs give you extensive control over the details of your graph. You can control the thickness of individual lines and the sizes of symbols, along with colors, titles, and axis labels on your graphs.

Axes and Labels

The axis labels "ozone" and "radiatio" in the plot of the `envirn` data are only mildly informative. We can make them more informative as follows:

1. In the open **Scatter Plot** dialog, click on the **Titles** tab to move to the **Titles** page.
2. In the **X Axis Label** field, type `Solar Radiation (langleys)`.
3. In the **Y Axis Label** field, type `Ozone Concentration`.
4. Click **Apply**.

Titles

We can insert a main title at the top of our graph, as follows:

1. In the **Main Title** field, type `The Relationship Between Radiation and Ozone`.
2. Click **Apply**.

Plot Properties

Finally, we can modify the lines and symbols in the plot of the `envirn` data as follows:

1. Click the **Plot** tab to move to the **Plot** page of the open **Scatter Plot** dialog.
2. Specify the **Line/Symbol Color** as **Color 5** and the **Line Width** as **2**.

3. Specify **Circle, Solid** as the **Symbol Style**.
4. Click **Apply**.

The graph that reflects all of our changes is shown in Figure 1.7.

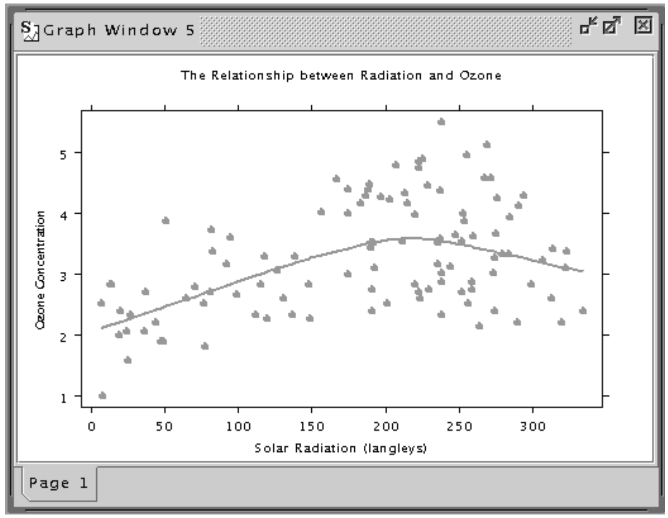

Figure 1.7: *After changing the axis labels and plot properties, our plot of the* `envirn` *data looks like this.*

Using Trellis Graphics for Multipanel Conditioning

Suppose you have a data set with multiple variables, and you want to see how plots of two variables change with variations in one or more conditioning variables. Exclusive to MathSoft, Trellis graphics are designed to display your data in a series of panels using conditioning options. Each panel contains a subset of the original data that corresponds to intervals of the conditioning variables.

Most graphs can be conditioned. To do this, the data columns used for each plot and for the conditioning variable(s) must be of equal length. By default, the axis specifications and panel display attributes such as fill color are identical for each panel.

We now apply multipanel conditioning to the loess plot we created in the previous section. The steps below provide the necessary instructions.

1. Click the **Data** tab to return to the **Data** page of the open **Scatter Plot** dialog.
2. In the **Conditioning** list box, select **temperat** and then CTRL-click **wind**.
3. Click the **Multipanel** tab to move to the **Multipanel** page of the dialog.
4. Enter 2 as the **# of Panels**. This provides two panels for each conditioning variable, so our plot will have four panels.
5. Click **Apply**. The Trellis graph in Figure 1.8 shows how the dependence of ozone on radiation varies according to levels of wind and temperature.

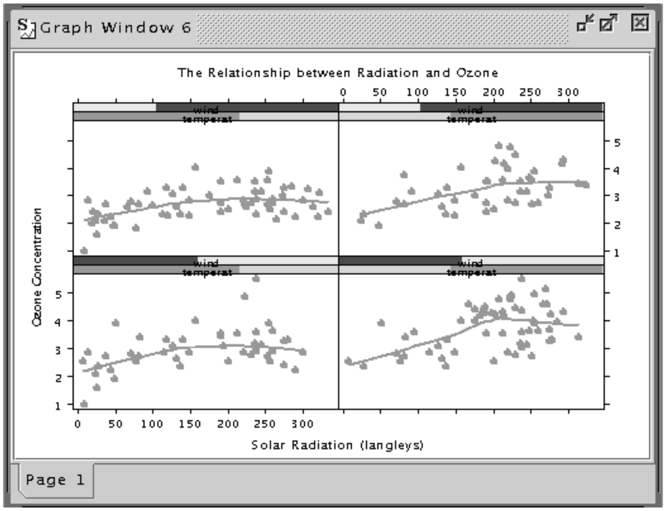

Figure 1.8: *Ozone concentration and solar radiation. This graph shows that radiation explains the variation in ozone levels beyond that explained by wind speed and temperature.*

To replace the loess curve with a least squares line in each panel, do the following:

1. Click the **Fit** tab to return to the **Fit** page of the open **Scatter Plot** dialog.
2. Choose **Least Squares** as the **Regression Type** and **None** as the **Smoothing Type**.
3. Click **Apply**.

A least squares regression line replaces the loess curve in each panel, as shown in Figure 1.9. This graph suggests that high temperatures with less wind result in the strongest dependence of ozone on radiation.

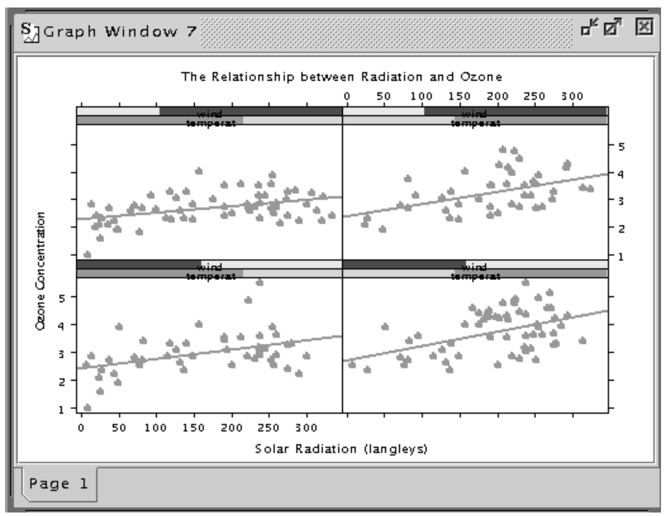

Figure 1.9: *Least squares lines have replaced the loess curves in each panel.*

Applying Statistical Models

S-PLUS provides a vast array of statistical techniques, with the most widely used techniques accessible through dialogs launched from the **Statistics** menu.

All techniques built into the menus are available through the S-PLUS language. Commands may be issued interactively in the **Commands** window. In the course of an analysis, the user may begin by fitting a model through a convenient dialog, then proceed to analyze the model and perform diagnostics through the flexible and powerful S-PLUS language.

In this section, we fit linear regression models that predict `ozone` from the temperature, radiation, and wind variables in the `envirn` data.

Data Summaries

First we look at simple summaries of the `envirn` data.

1. Choose **Statistics ► Data Summaries ► Summary Statistics**. The **Summary Statistics** dialog appears.
2. Choose `envirn` from the **Data Set** drop-down list.
3. Click **OK**. Summaries for the columns appear in a **Report** window.
4. Choose **Statistics ► Data Summaries ► Correlations**. The **Correlations and Covariances** dialog appears.
5. Choose `envirn` from the **Data Set** drop-down list.
6. Click **OK**. Correlations for the columns appear in a **Report** window.

Linear Regression

Next, we use the **Linear Regression** dialog to fit a linear model that predicts `ozone` from the other variables in the `envirn` data.

A simple model from the dialog

1. Choose **Statistics ► Regression ► Linear**. The **Linear Regression** dialog opens.
2. Choose `envirn` from the **Data Set** drop-down list.
3. Choose `ozone` as the **Dependent Variable**.
4. Choose `radiatio` as the first **Independent Variable**, then SHIFT-click on `temperat` and `wind`. The formula `ozone ~ radiatio+temperat+wind` appears in the **Formula** field.
5. Click the **Plot** tab to move to the **Plot** page of the dialog.
6. On the **Plot** page, check the box beside **Residuals vs. Fit**, then click **OK**.

The regression results appear in a **Report** window. In addition, a new **Graph** window is created that displays the chosen diagnostic plots.

Using the Formula Builder

The **Formula Builder** in the regression dialogs allows you to describe complex models by selecting variables and indicating how they are used in the model. For example, you might want to add an interaction term to the model. The **Formula Builder** lets you do this easily.

The following steps use the **Formula Builder** to add an interaction term to our simple linear model for the `envirn` data .

1. Choose **Statistics ▶ Regression ▶ Linear**.
2. Choose `envirn` from the **Data Set** drop-down list.
3. Choose `ozone` as the **Dependent Variable**, and CTRL-click to select `radiatio`, `temperat`**,** and `wind` as the **Independent Variables**. Notice that the formula reflects your selections.
4. Click the **Create Formula** button.
5. Select `radiatio` and `temperat` in the **Variables** list. Click on **Add Interaction** to include the interaction between radiation and temperature as a predictor.
6. Click **OK** to exit the **Formula Builder** dialog. The formula you generated is placed in the **Formula** field of the **Linear Regression** dialog.
7. Click **Apply** to generate the model.

More detailed results

With the following steps, we generate an ANOVA table for the linear model that includes the interaction term:

1. Click the **Results** tab to move to the **Results** page of the open **Linear Regression** dialog.
2. Check the **ANOVA Table** box and clear the **Long Output** check box. These settings provide an analysis of variance table for the linear model.
3. Click **OK**. The ANOVA table for the fit appears in the **Report** window.

Creating a 3D Graph

In this example, we use the data set `exsurf` to create a three dimensional plot.

1. Before continuing, close any open windows.
2. From the **Graph** menu, choose **Three Variables ▶ Surface Plot**.
3. Type `exsurf` in the **Data Set** field.
4. Choose `V1` as the **x Axis Variable**, `V2` as the **y Axis Variable**, and `V3` as the **z Axis Variable**.
5. Click **Apply**. The graph shown in Figure 1.10 appears.

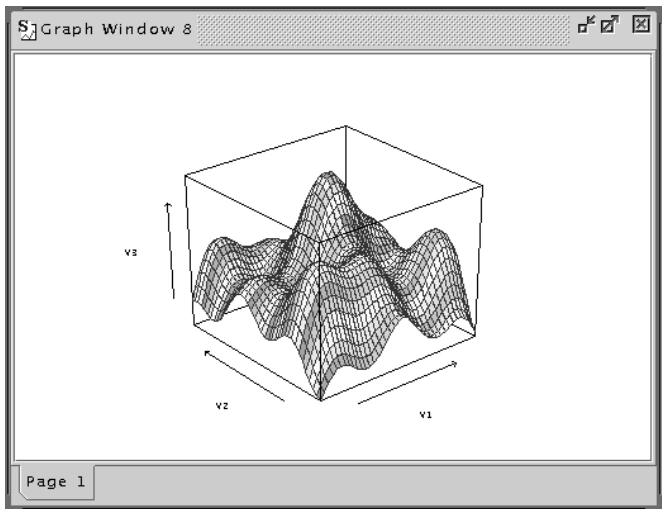

Figure 1.10: *A 3D surface plot of the* `exsurf` *data.*

Getting Help

You can obtain help from the graphical user interface at any time by selecting an option under the **Help** menu, or by clicking the **Help** button within a dialog. The help window appears as in Figure 1.11. S-PLUS 6.0 uses the JavaHelp system from Sun Microsystems as its help browser.

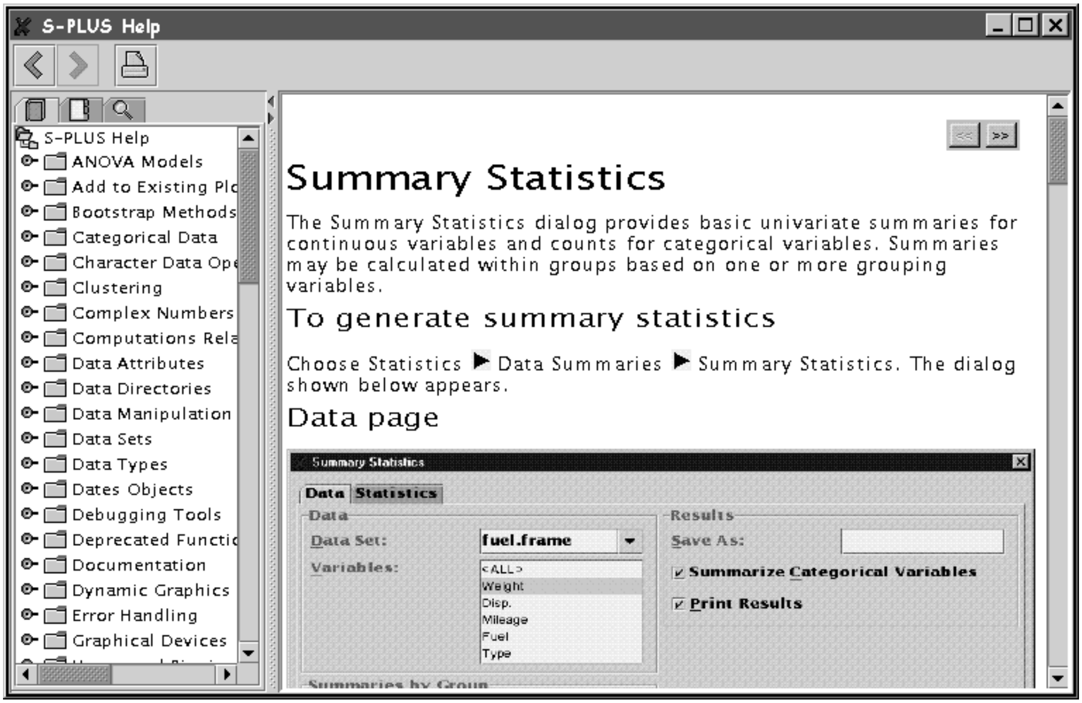

Figure 1.11: *The S-PLUS JavaHelp window, displaying the help file for the* ***Summary Statistics*** *dialog.*

USING THE COMMANDS WINDOW

For some analyses, it is more convenient to work with an interactive data analysis language than to maneuver through a series of dialogs. In this section we use the **Commands** window to fit another linear model to the `envirn` data and perform some diagnostics. If it is not already open, open the **Commands** window by choosing **View ► Commands Window**. Close all other windows before continuing with the tutorial.

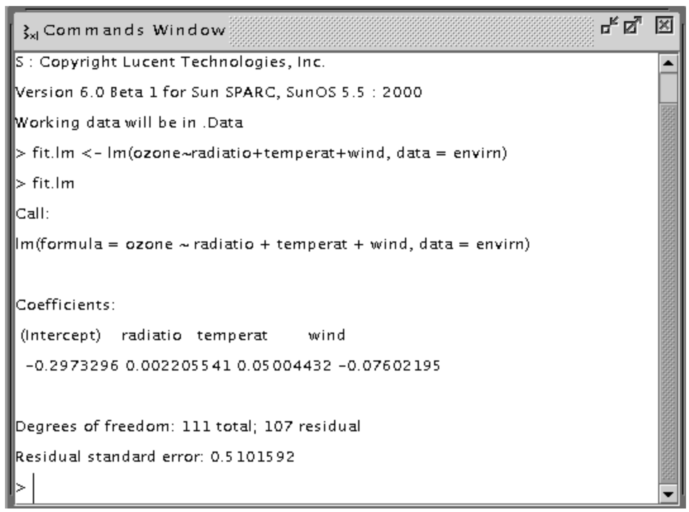

Figure 1.12: *The **Commands** window.*

Overview

The **Commands** window gives you interactive access to the **S-PLUS** language. Everything you type in **S-PLUS** is an *expression*. Expressions are evaluated when you press the ENTER key. If you press ENTER after an expression that is syntactically incomplete, it is not evaluated; however, it does not result in an error, either. Instead, S-PLUS prompts you to continue the expression with the + continuation prompt.

You can type several expressions on the same line by separating them with semicolons (`;`). S-PLUS evaluates each in succession when you press ENTER. A semicolon is not required at the end of each line, only between multiple expressions on a single line. You can include comments in S-PLUS expressions following a # symbol. Anything after the # on a line is interpreted as a comment, and is not evaluated.

The result of any expression is an *object* that may be saved by assigning it a name using the assignment operator `<-`. (The assignment operator is formed by typing a "less than" character followed immediately by a "hyphen". Do not put any spaces between the two characters.) All data used in S-PLUS is represented as some type of S-PLUS data object.

Most S-PLUS expressions are *function calls*, since S-PLUS is a functional language. To call a function, type the name of the function followed by a set of parentheses containing any arguments to the function.

S-PLUS commands are case-sensitive. S-PLUS ignores most spaces, so you can include or omit spaces in typing your expressions as you prefer. Do not place extra spaces within the name of an object, however, or between the digits of a single number, or between the `<` and `-` in the assignment operator.

The **Commands** window uses a `>` prompt. In this document, text starting with `>` is to be typed at this prompt, but the `>` should not be typed. If you must break a line before typing what S-PLUS can interpret as a complete command, S-PLUS provides the continuation prompt + at the beginning of the next line.

Fitting a Linear Model

1. If you have not already done so, use **File ▶ Import Data** to load **exenvirn.sd2** from the **$SHOME/library/example5** directory and create the `envirn` data set (see the section Importing Data on page 12).
2. To replicate the regression results from the previous section and store them in an object named `fit.lm`, type the following:

```
> fit.lm <- lm(ozone ~ radiatio + temperat + wind,
+ data = envirn)
```

3. To see a brief summary for the model, type:

```
> fit.lm
```

When we fit models for `envirn` using the **Linear Regression** dialog, we added one term to examine the interaction between temperature and radiation in determining ozone level. We now fit a model containing all two-way interactions, and explore whether the interactions are significant.

1. To fit a linear model with all two-way interactions, type:

```
> fit.int <- lm(ozone ~ (radiatio+temperat+wind)^2,
+ data = envirn)
```

2. For a brief summary of the fit, type:

```
> fit.int
```

3. For a detailed summary, type:

```
> summary(fit.int)
```

4. For an F-test comparing this model to the `fit.lm` model, type:

```
> anova(fit.lm, fit.int)
```

Getting Help

To get help for a function such as `anova` when working in the **Commands** window, type

```
> help(anova)
```

If JavaHelp is running, the help file for the function is displayed in a JavaHelp window. Otherwise, the help file is displayed in the **slynx** browser provided with S-PLUS; this is a version of the freely available **lynx** browser. To view help files in **slynx** after the help system has been started, type `help.off()` in the **Commands** window, and then request help with the `help` function.

You can copy and paste example commands from a help file directly into the **Commands** window. When doing this, you should use the CTRL-C/CTRL-V mechanism for copying and pasting; i.e., highlight the text, press CTRL-C to copy it, place the mouse cursor in the position where you want to paste it, and then press CTRL-V. The **Commands** window in the graphical user interface uses the X-selection protocol for copying and pasting. However, most other portions of the graphical user interface, including JavaHelp, use the GUI-standard CTRL-C/CTRL-V for copying and pasting.

Creating Graphics with the Java Graphics Device

In the earlier portions of this tutorial, we used the **Graph** menu to create graphics that were displayed in **Graph** windows. The standard graph window is an instance of a Java graphics device. In this section, we show how the Java graphics device can be called from the S-PLUS **Commands** window and used to create bitmap graphic files in a variety of formats.

Starting the Java graphics device

The simplest way to open a Java graphics device is as follows:

```
> java.graph()
```

This is analogous to opening most graphics devices, such as `motif`.

If you will be creating Trellis graphics, you can open the Java graphics window as a Trellis device:

```
> trellis.device(“java.graph”)
```

Creating graphics

Once you've opened the `java.graph` device, you can create graphics in it using any S-PLUS graphics command:

```
> plot(corn.rain)
> image(voice.five)
> example.dotplot()
```

Closing the device

As with all graphics devices, close the Java graphics device by issuing the following command:

```
> dev.off()
```

Creating Bitmap Graphics

To create a bitmap graphic, start `java.graph` with a `file` argument and, if necessary, a `type` argument. The supported types are JPEG, BMP, PNG, PNM, and TIFF; the default file type is JPEG. For example, to create a JPEG image of the `voice.five` data, use `java.graph` as follows:

```
> java.graph("voice.jpeg", type = "JPEG")
> image(voice.five)
> dev.off()
```

Creating Windows Metafiles

The Windows Metafile is a popular format for vector graphics. You can import Windows metafiles into Windows applications such as Microsoft Word, Adobe FrameMaker, and Microsoft PowerPoint. You can create Windows metafiles in S-PLUS using the `wmf.graph` function. This function is similar to the `pdf.graph` and `postscript` functions, which have provided vector graphics output in earlier versions of S-PLUS. In most cases, the only required argument to `wmf.graph` is a file name; the `wmf` file extension is standard, and should always be used. For example:

```
> wmf.graph("loess.wmf")
> gas.m <- loess(NOx ~ E, data=gas, span=2/3, degree=2)
> plot(gas.m)
> dev.off()
```

WHAT'S NEW IN S-PLUS 6.0 2

S-PLUS 6.0 is a major upgrade of S-PLUS, including the new or enhanced features listed below.

Statistics

- New basic statistical methods: Shapiro-Wilk test for normality, Durbin-Watson test for serial correlation, and built-in functions for skewness and kurtosis. See the help files for `shapiro.test`, `durbinWatson`, `skewness`, and `kurtosis`, respectively.
- Enhanced discriminant analysis: homoscedastic (linear), heteroscedastic (quadratic), proportional covariances, equal correlation, common principal components, and canonical discriminant function. See the chapter Discriminant Analysis in the *Guide to Statistics, Volume 2*.
- Enhanced mixed effects analysis: new, state-of-the-art code contributed by Doug Bates of the University of Wisconsin–Madison and José Pinhiero of Lucent Technologies. See the chapter Linear and Nonlinear Mixed-Effects Models in the *Guide to Statistics, Volume 1*.
- Enhanced survival analysis: a new survival library from Terry Therneau at the Mayo Clinic, along with some enhancements to the accelerated life testing functionality (`censorReg`). See the chapters on survival analysis in the *Guide to Statistics, Volume 2*.
- Enhanced quality control charts, including a number of moving average charts. See the chapter Quality Control Charts in the *Guide to Statistics, Volume 2*.

- New Missing Data library: multiple imputation, and Gaussian, logistic and conditional Gaussian models. See the file **Missing.pdf** in your **$SHOME/library/missing** directory.
- New Robust Statistics library: robust methods for computing models and detecting outliers in data. See the file **Robust.pdf** in your **$SHOME/library/robust** directory.
- Enhanced methods for handling missing values: the new `NA` method `na.exclude` behaves in the way `na.omit` used to for `survreg`. This new method works for all modeling functions except those in the NLME library. Through the functions `naresid`, `napredict`, and `nafitted` (implemented in `resid`, `predict`, and `fitted` methods, respectively), `na.exclude` allows residuals and fitted values to have the same length as the original data in the presence of `NA`s. This is achieved by adding rows of `NA`s, corresponding to the locations of missing values in the fitted data. Using `na.action = na.omit` yields complete backward compatibility for all functions except the survival functions. For consistency, `survreg`, `survfit`, `coxph`, and `censorReg` now return rows without `NA`s when `na.action=na.omit`; for the same behavior in these functions as in previous versions of S-PLUS, use `na.action=na.exclude`.

Java Interconnectivity and the GUI

- New graphical user interface: point-and-click menus and dialogs, a **Commands** window with command-line editing, and a new JavaHelp interface for the help system. The GUI includes dialogs for importing and exporting data, Trellis graphics, and most of the statistical functionality in S-PLUS. For more information, see the *User's Guide.*
- CONNECT/Java: a powerful suite of functions that allows you to call S-PLUS from Java applications, or call Java from S-PLUS with the new `.Java` interface. For more information, see the *Programmer's Guide.*
- Java graphlets: the ability to embed graphs as interactive Java applets within web pages. For more details, see the *Programmer's Guide.*

Graphics

- Improved calculations in histograms and density plots: new methods for calculating bin width in `hist` and bandwidth in `density`, contributed by Bill Venables of CSIRO Marine Laboratories and Brian Ripley of the University of Oxford. For more details, see the help files for `hist` and `density`.
- New `java.graph` device: creates bitmap graphics in JPEG, TIFF, BMP, PNG, and PNM formats. For more information, see the chapter Working With Graphics Devices in the *User's Guide*.
- New `wmf.graph` device: creates vector graphics in Windows Metafile format. See the help file for `wmf.graph`.

Import/Export

- New supported file types: SAS 7, SAS 8, and Matlab 5 files are now supported. For more details, see the help files for `importData` and `exportData`.

Documentation

- New JavaHelp interface to the help system, including index and full-text search capabilities.
- Completely revised set of written documentation: updated *Installation and Maintenance Guide*, *User's Guide*, and two volumes of the *Guide to Statistics*.
- New *Programmer's Guide*: thoroughly documents built-in functions and the S-PLUS language.

INDEX