Getting Started with TIBCO Spotfire S+® 8.1 for Solaris®/Linux®

November 2008

TIBCO Software Inc.

IMPORTANT INFORMATION

Reference

The correct bibliographic reference for this document is as follows:

Getting Started with TIBCO Spotfire S+® 8.1 for Solaris®/Linux®
TIBCO Software Inc.

Technical Support

For technical support, please visit http://spotfire.tibco.com/support and register for a support account.

TIBCO SPOTFIRE S+ BOOKS

The TIBCO Spotfire S+® documentation includes books to address your focus and knowledge level. Review the following table to help you choose the Spotfire S+ book that meets your needs. These books are available in PDF format in the following locations:

- In your Spotfire S+ installation directory (**SHOME\help** on Windows, **SHOME/doc** on UNIX/Linux).
- In the Spotfire S+ Workbench, from the **Help ► Spotfire S+ Manuals** menu item.
- In Microsoft® Windows®, in the Spotfire S+ GUI, from the **Help ► Online Manuals** menu item.

Spotfire S+ documentation.

Information you need if you...	**See the...**
Are new to the S language and the Spotfire S+ GUI, and you want an introduction to importing data, producing simple graphs, applying statistical models, and viewing data in Microsoft Excel®.	*Getting Started Guide*
Are a new Spotfire S+ user and need how to use Spotfire S+, primarily through the GUI.	*User's Guide*
Are familiar with the S language and Spotfire S+, and you want to use the Spotfire S+ plug-in, or customization, of the Eclipse Integrated Development Environment (IDE).	*Spotfire S+ Workbench User's Guide*
Have used the S language and Spotfire S+, and you want to know how to write, debug, and program functions from the **Commands** window.	*Programmer's Guide*
Are familiar with the S language and Spotfire S+, and you want to extend its functionality in your own application or within Spotfire S+.	*Application Developer's Guide*

Spotfire S+ documentation. (Continued)

Information you need if you...	**See the...**
Are familiar with the S language and Spotfire S+, and you are looking for information about creating or editing graphics, either from a **Commands** window or the Windows GUI, or using Spotfire S+ supported graphics devices.	*Guide to Graphics*
Are familiar with the S language and Spotfire S+, and you want to use the Big Data library to import and manipulate very large data sets.	*Big Data User's Guide*
Want to download or create Spotfire S+ packages for submission to the Comprehensive S-PLUS Archive Network (CSAN) site, and need to know the steps.	*Guide to Packages*
Are looking for categorized information about individual Spotfire S+ functions.	*Function Guide*
If you are familiar with the S language and Spotfire S+, and you need a reference for the range of statistical modelling and analysis techniques in Spotfire S+. Volume 1 includes information on specifying models in Spotfire S+, on probability, on estimation and inference, on regression and smoothing, and on analysis of variance.	*Guide to Statistics, Vol. 1*
If you are familiar with the S language and Spotfire S+, and you need a reference for the range of statistical modelling and analysis techniques in Spotfire S+. Volume 2 includes information on multivariate techniques, time series analysis, survival analysis, resampling techniques, and mathematical computing in Spotfire S+.	*Guide to Statistics, Vol. 2*

CONTENTS

TUTORIAL

1

INTRODUCTION

This tutorial is designed to acquaint you with TIBCO Spotfire S+ for Solaris® and Linux®. It includes the following information:

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

Release Notes for Spotfire S+ can be found in the text file **RELNOTES.TXT** at the top level of your installation. An HTML version is located at **doc/RELNOTES.HTM**. In addition, installation notes can be found in **INSTALL.TXT** at the top level of your installation, and a list of bugs fixed for the release is located at **doc/FIXEDBUG.TXT**.

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

To create a working directory named **myproj** in your home directory, type the following sequence of commands at the 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 Spotfire S+ 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

TIBCO Spotfire S+ 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 Spotfire S+ tools. In this quick tour, we do the following:

- 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 Spotfire S+.

Starting Spotfire S+

There are a several ways to start Spotfire S+ from a command line.

- Using the Spotfire S+ command line without Java:

  ```
  Splus
  ```

- Using the Spotfire S+ command line supporting Java calls, Java graphics, and the Java help interface:

  ```
  Splus -j
  ```

- Using the Spotfire S+ graphical user interface:

  `Splus -g` or `Splus -g &`

Note
As of verion 8.1, the Spotfire S+ Java-based GUI is deprecated. If you want to use a GUI with Spotfire S+, use the Spotfire S+ Workbench.

The second command using the ampersand `&` puts the GUI process into the background, freeing your xterm for other uses.

- Attaching the Spotfire S+ Big Data Library before running Spotfire S+:

```
Splus -bigdata
```

- Starting Spotfire S+ with the Spotfire S+ Workbench:

```
Splus -w | -workbench
```

Spotfire S+ includes two additional flags, `-jit` and `-helpoff`:

- The `-jit` flag works with the `-g`, `-j`, and `-userapp` flags, and allows you to turn on the Java just-in-time compiler. This makes the graphical user interface and help system run faster but introduces instabilities that often lead to crashes. In particular, the just-in-time compiler often crashes while repainting graphical user interface elements such as the JavaHelp window and the Data window.
- The `-helpoff` flag is useful only with the `-g` flag. It turns off the automatic invisible startup of the help system. The invisible startup improves initial responsiveness of the help system but adds a significant memory footprint to the current session. If you wish to optimize your available memory, this flag may prove useful.

If you use the `Splus -g` option, Spotfire S+ appears in its own window, with a main menu, toolbar and an open **Commands** window, as shown in Figure 1.1.

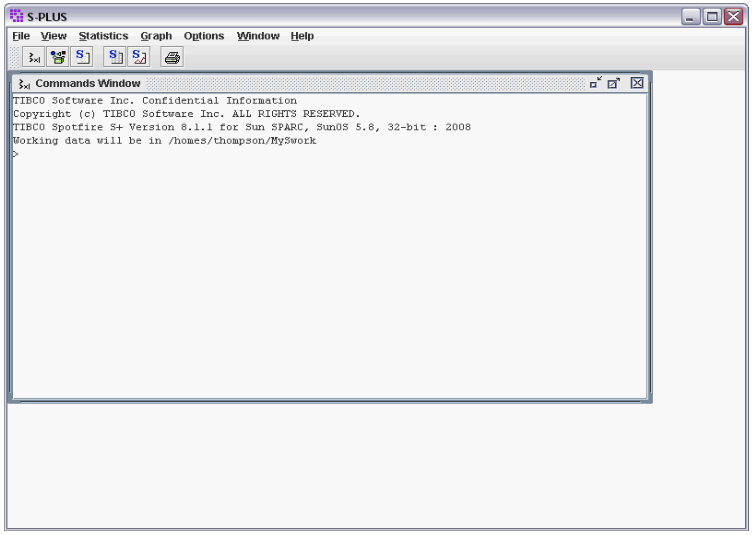

Figure 1.1: *Using the `Splus -g` option, the **Commands** window is displayed when you start Spotfire S+.*

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.

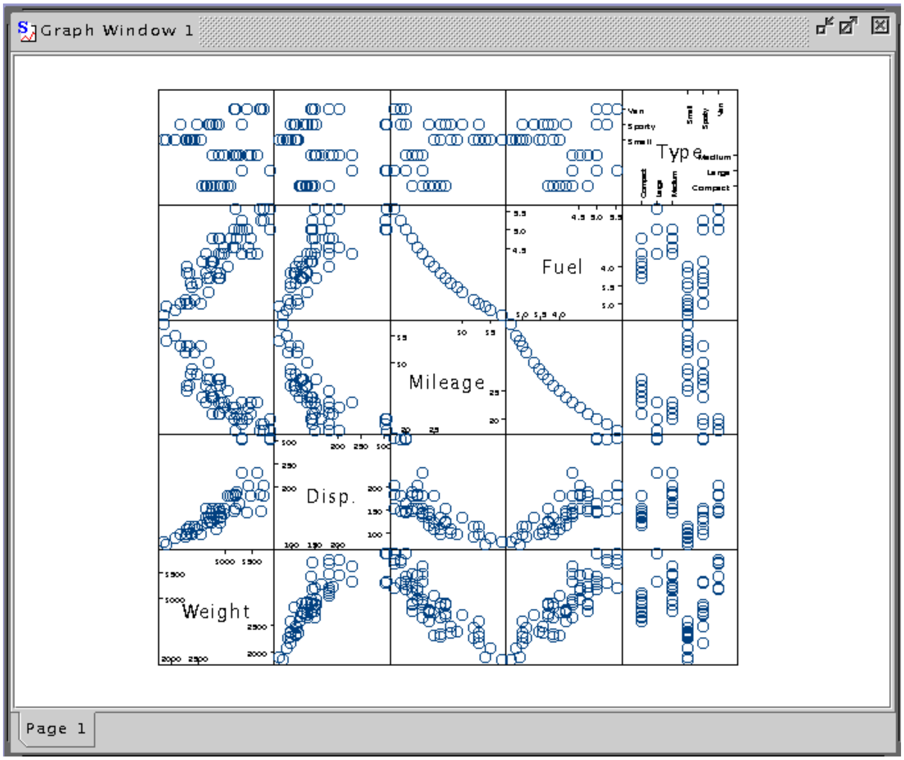

Figure 1.2: *A scatterplot matrix of the* `fuel.frame` *data shows a number of strong relationships.*

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.

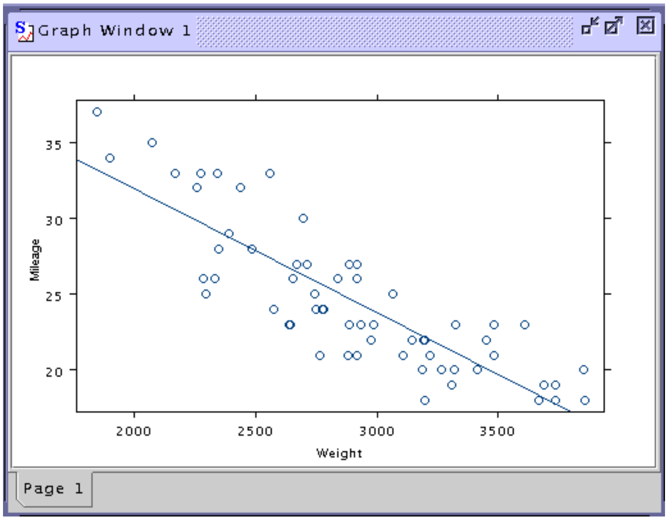

Figure 1.3: *A linear fit of* `Mileage` *vs.* `Weight` *in the* `fuel.frame` *data.*

To examine how `Vans` or `Compact` cars fit into this example, you can use TIBCO'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.

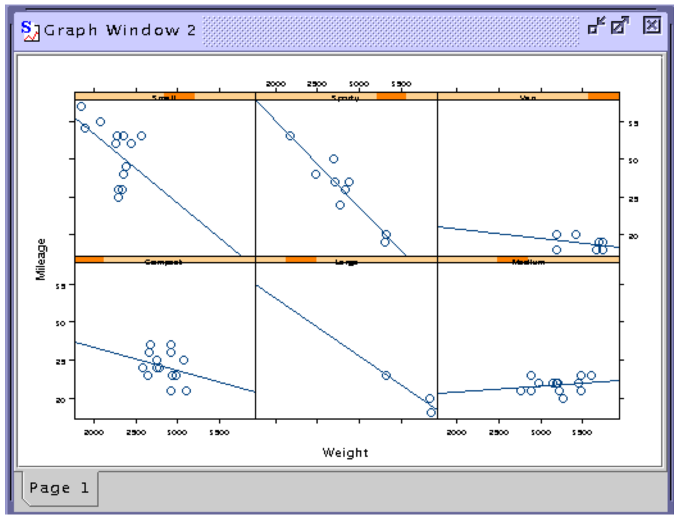

Figure 1.4: *A Trellis view of the* `fuel.frame` *data.*

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.

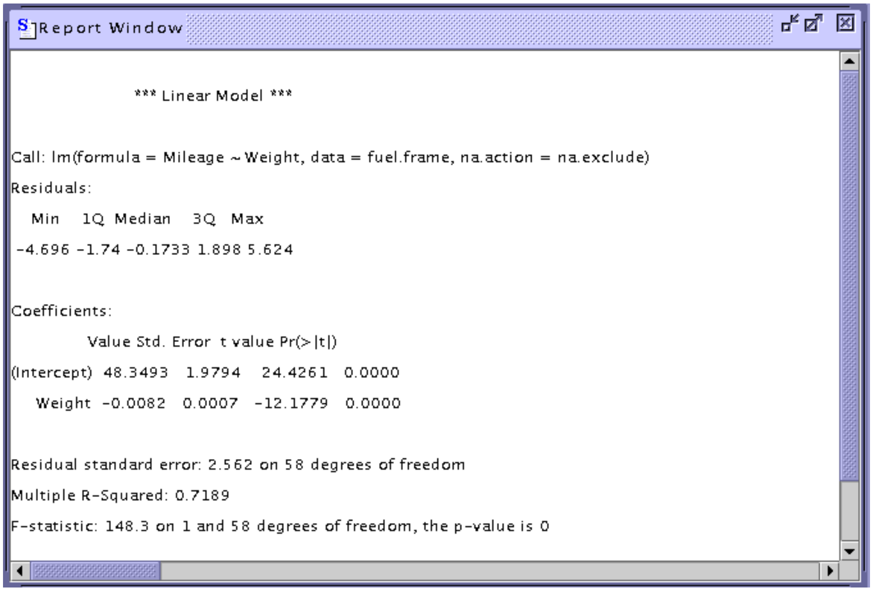

Figure 1.5: *A* ***Report*** *window showing a linear fit for the* `fuel.frame` *data.*

Creating a 3D Graph

Spotfire S+ 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.

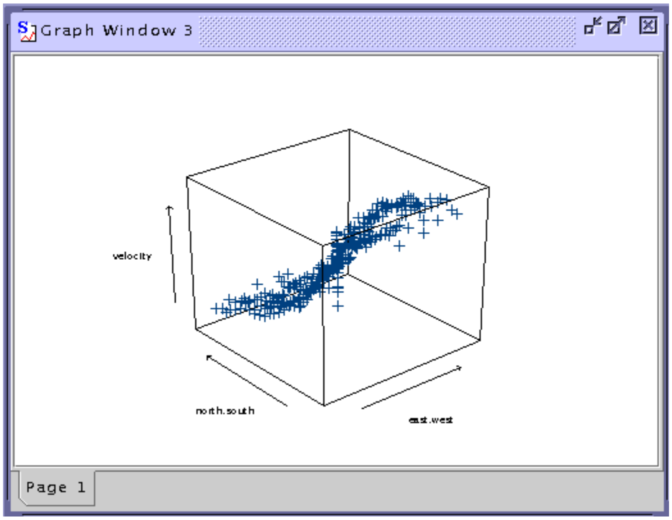

Figure 1.6: *A point cloud of the* `galaxy` *data.*

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 Spotfire S+.

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 Spotfire S+ 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.

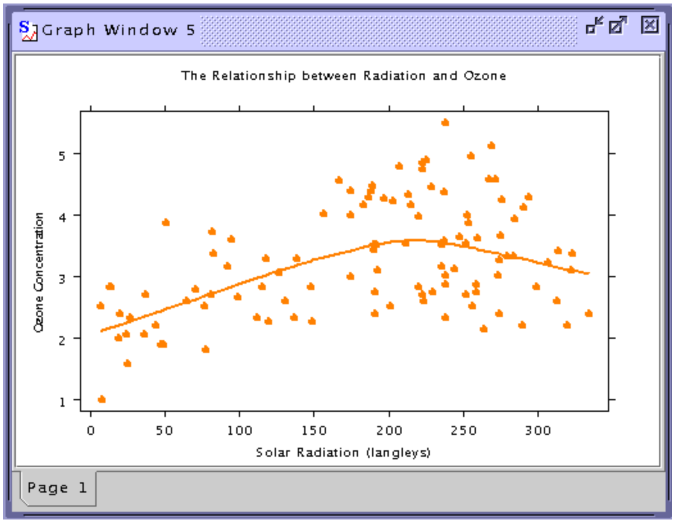

Figure 1.7: *After changing the axis labels and plot properties, our plot of the* `envirn` *data looks like this.*

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 TIBCO, 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.

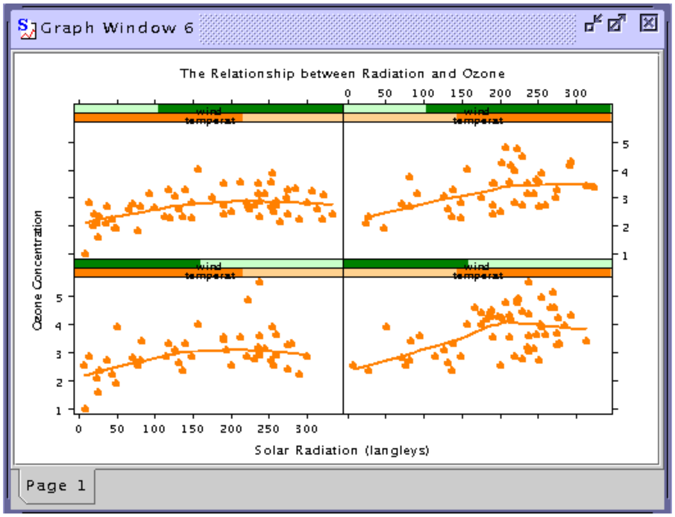

Figure 1.8: *Ozone concentration and solar radiation. This graph shows that radiation explains the variation in ozone levels beyond that explained by wind speed and temperature.*

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.

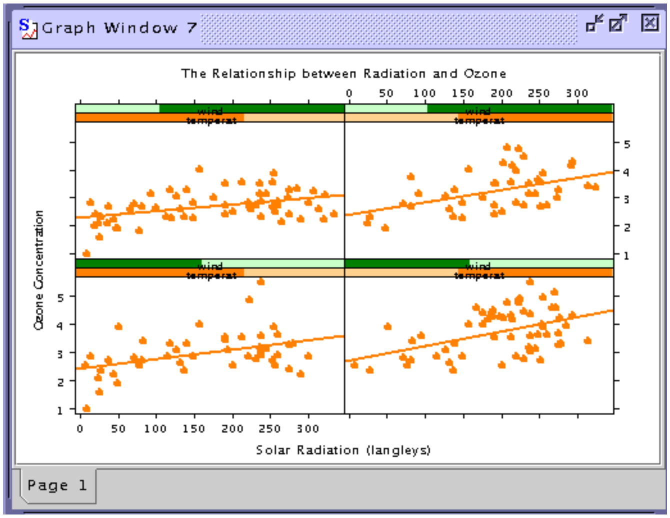

Figure 1.9: *Least squares lines have replaced the loess curves in each panel.*

Applying Statistical Models

Spotfire S+ provides an array of statistical techniques accessible through dialogs launched from the **Data** and **Statistics** menus.

All techniques built into the menus are available through the Spotfire S+ 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 Spotfire S+ 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. In the **Add** groupbox, click **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.

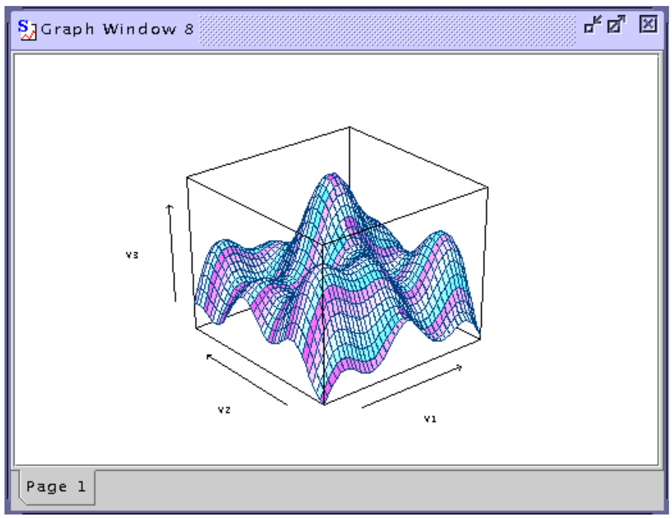

Figure 1.10: *A 3D surface plot of the* `exsurf` *data.*

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. Spotfire S+ uses the JavaHelp system from Sun Microsystems as its help browser.

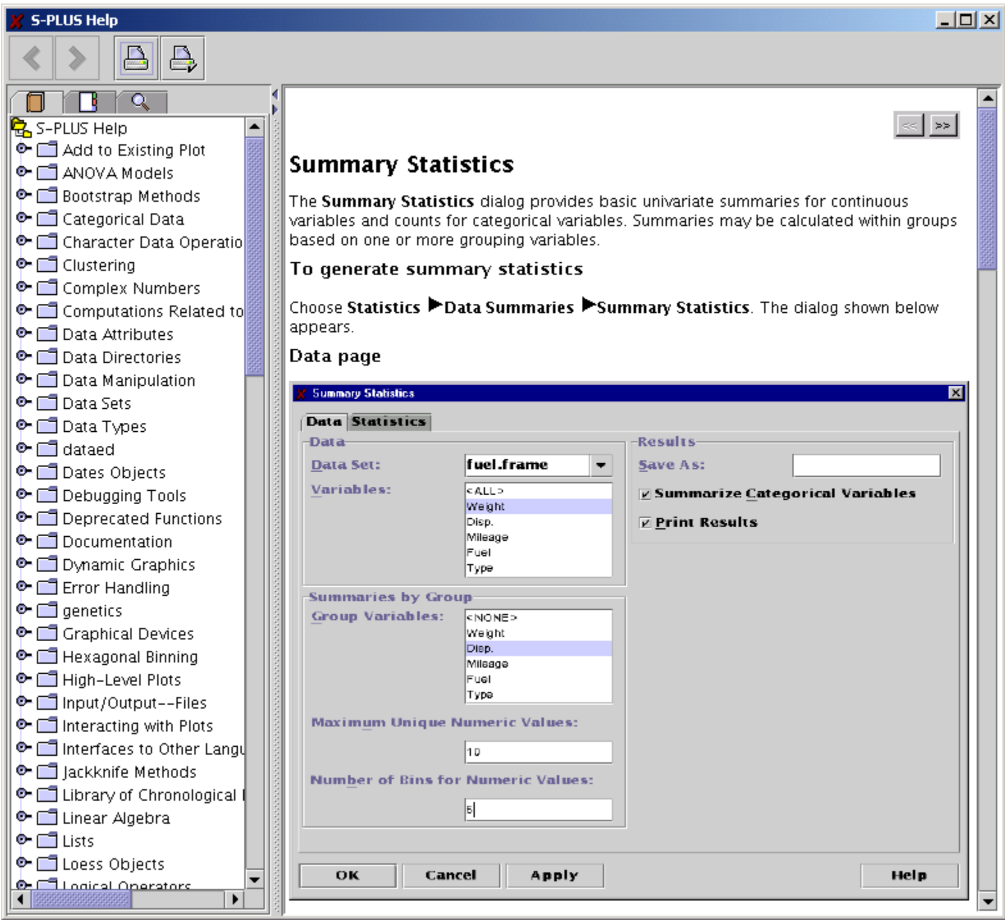

Figure 1.11: *The Spotfire S+ JavaHelp window, displaying the help file for the **Summary Statistics** dialog.*

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.

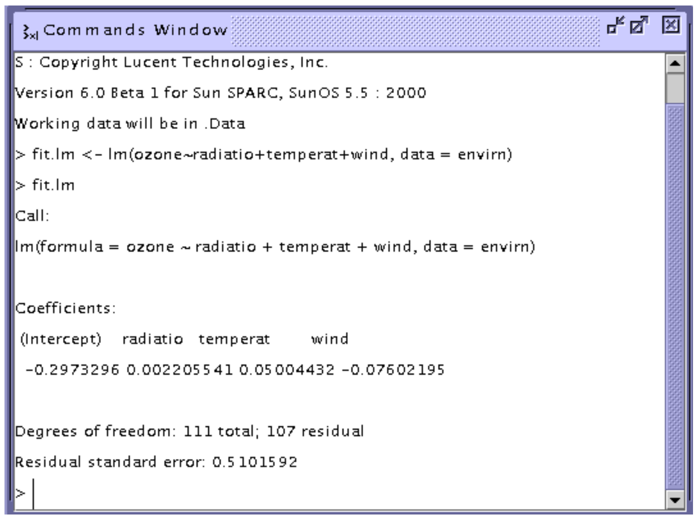

Figure 1.12: *The **Commands** window.*

Overview

The **Commands** window gives you interactive access to the **Spotfire S+** language. Everything you type in **Spotfire S+** 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, Spotfire S+ prompts you to continue the expression with the + continuation prompt.

You can type several expressions on the same line by separating them with semicolons (`;`). Spotfire S+ 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 SPOTFIRE S+ 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 Spotfire S+ is represented as some type of Spotfire S+ data object.

Most Spotfire S+ expressions are *function calls*, since Spotfire S+ 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.

Spotfire S+ commands are case-sensitive. Spotfire S+ 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 Spotfire S+ can interpret as a complete command, Spotfire S+ provides the continuation prompt + at the beginning of the next line.

Spotfire S+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 13).
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 an available Help application such as 'lynx', 'links', 'less', or 'more'.

Note

The slynx program is not distributed with Spotfire S+; however, if you want to use it as a Help browser, you can download it it separately as part of the 'pkgutils' package (using the `install.pkgutils()` function), and then `help()` will use it.

To try another text-based HTML browser, set `options(help.pager="`*`yourBrowser`*`")` where *`yourBrowser`* specifies your particular HTML browser.

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 Spotfire S+ **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 Spotfire S+ 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", format = "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 Spotfire S+ 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 Spotfire S+. 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()
```

INDEX

T