

**Proprietary
Notice**

Insightful Corporation owns both this software program and its documentation. Both the program and documentation are copyrighted with all rights reserved by Insightful Corporation.

The correct bibliographical reference for this document is as follows:

Getting Started with S-PLUS 7.0 for UNIX, Insightful Corporation, Seattle, WA.

Printed in the United States.

Copyright Notice

Copyright © 1988-2005, Insightful Corporation. All rights reserved.

Insightful Corporation
1700 Westlake Avenue N, Suite 500
Seattle, Washington 98109
USA

Trademarks

Insightful, Insightful Corporation, “Insightful intelligence from data,” S-PLUS, S-PLUS Graphlets, Graphlet, and InFact are trademarks or registered trademarks of Insightful Corporation in the United States and other countries. Insightful Miner, S+FinMetrics, S+SeqTrial, S+SpatialStats, S+ArrayAnalyzer, S+NuoOPT, S+EnvironmentalStats and S+Wavelets are trademarks of Insightful Corporation. Intel is a registered trademark, and Pentium a trademark, of Intel Corporation; Microsoft, Windows, MS-DOS, and Excel are registered trademarks, and Windows NT a trademark, of Microsoft Corporation. All product names mentioned herein may be trademarks or registered trademarks of their respective companies.

ACKNOWLEDGMENTS

S-PLUS would not exist without the pioneering research of the Bell Labs S team at AT&T (now Lucent Technologies): John Chambers, Richard A. Becker (now at AT&T Laboratories), Allan R. Wilks (now at AT&T Laboratories), Duncan Temple Lang, and their colleagues in the statistics research departments at Lucent: William S. Cleveland, Trevor Hastie (now at Stanford University), Linda Clark, Anne Freeny, Eric Grosse, David James, José Pinheiro, Daryl Pregibon, and Ming Shyu.

Insightful Corporation thanks the following individuals for their contributions to this and earlier releases of S-PLUS: Douglas M. Bates, Leo Breiman, Dan Carr, Steve Dubnoff, Don Edwards, Jerome Friedman, Kevin Goodman, Perry Haaland, David Hardesty, Frank Harrell, Richard Heiberger, Mia Hubert, Richard Jones, Jennifer Lasecki, W.Q. Meeker, Adrian Raftery, Brian Ripley, Peter Rousseeuw, J.D. Spurrier, Anja Struyf, Terry Therneau, Rob Tibshirani, Katrien Van Driessen, William Venables, and Judy Zeh.

CONTENTS

Acknowledgments	iii
Chapter 1 Tutorial	1
Introduction	2
Quick Tour	4
Extended Tour: Examining Environmental Data	12
Using the Commands Window	21
Index	27

Contents

Introduction	2
Quick Tour	4
Starting S-PLUS	4
Getting Data	5
Creating a 2D Graph	6
Linear Regression	7
Creating a 3D Graph	10
Extended Tour: Examining Environmental Data	12
Importing Data	12
Creating a 2D Graph	12
Changing Graph Features	13
Using Trellis Graphics for Multipanel Conditioning	14
Applying Statistical Models	16
Creating a 3D Graph	19
Getting Help	19
Using the Commands Window	21
Overview	21
Fitting a Linear Model	22
Getting Help	23
Creating Graphics with the Java Graphics Device	24
Creating Windows Metafiles	25

INTRODUCTION

This tutorial is designed to acquaint you with S-PLUS 7.0 for UNIX and Linux. It includes the following information:

- A quick tour of the 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 can be found in the text file **RELNOTES.TXT** at the top level of your installation, and 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 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 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 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, 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 S-PLUS.

Starting S-PLUS There are a number of ways to start S-PLUS from a command line.

- Using the S-PLUS command line without Java:
`Splus`
- Using the S-PLUS command line supporting Java calls, Java graphics, and the Java help interface:
`Splus -j`
- Using the S-PLUS graphical user interface:

`Splus -g or Splus -g &`

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

- Attaching the S-PLUS Big Data Library before running S-PLUS (available in the Enterprise Developer edition only):
`Splus -bigdata`
- Starting S-PLUS with the Eclipse Workbench (available in the Enterprise Developer edition only):

`Splus -w | -workbench`

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

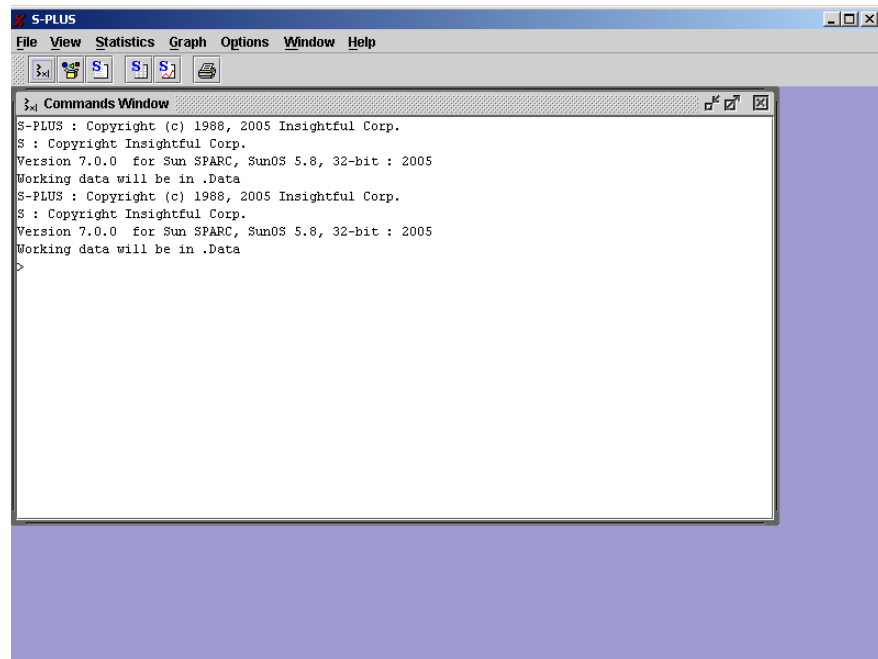


Figure 1.1: Using the `Splus -g` option, the **Commands** window is displayed when you start S-PLUS.

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`: $100/\text{Mileage}$. 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.

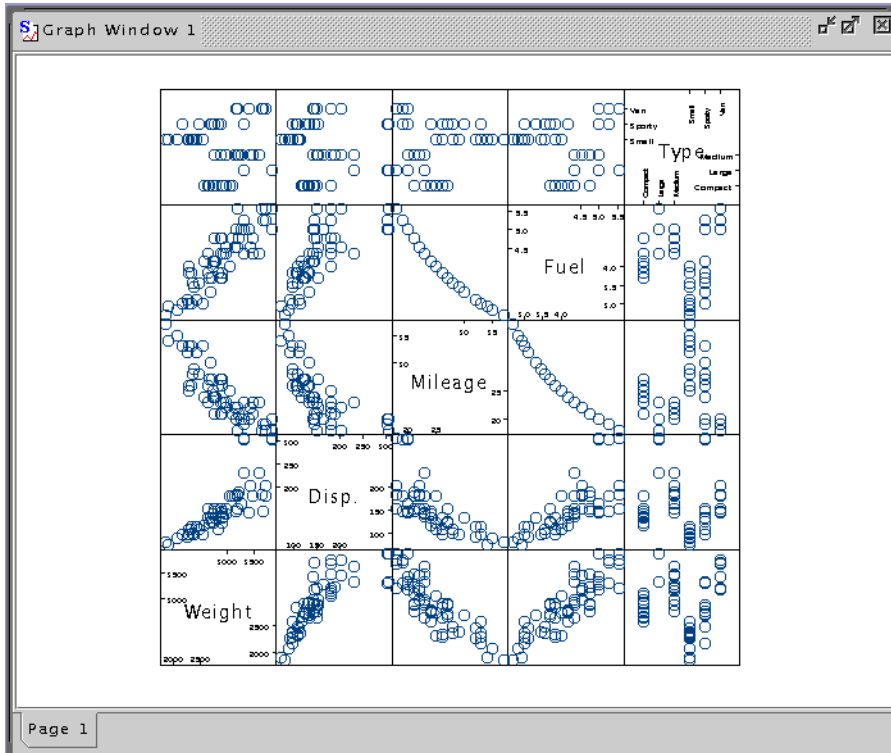


Figure 1.2: A scatterplot matrix of the `fuel.frame` data shows a number of strong relationships.

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.

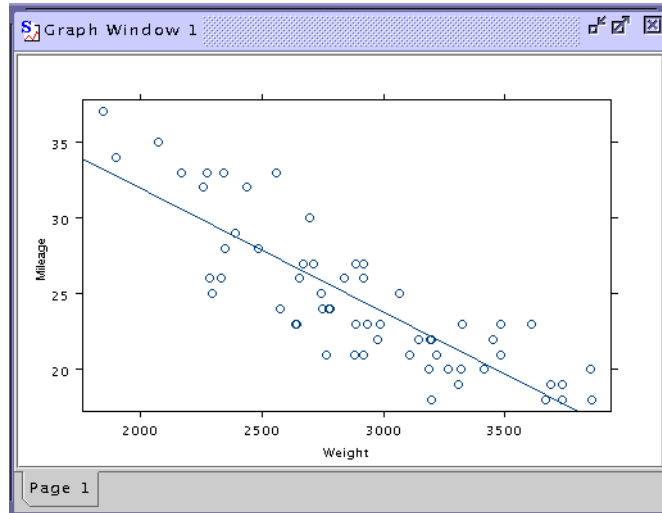


Figure 1.3: A linear fit of *Mileage vs. Weight* in the *fuel.frame* data.

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

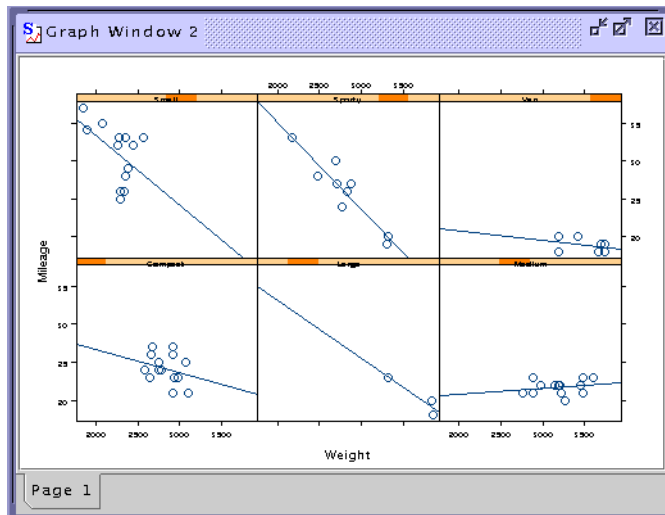


Figure 1.4: A Trellis view of the fuel.frame data.

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.

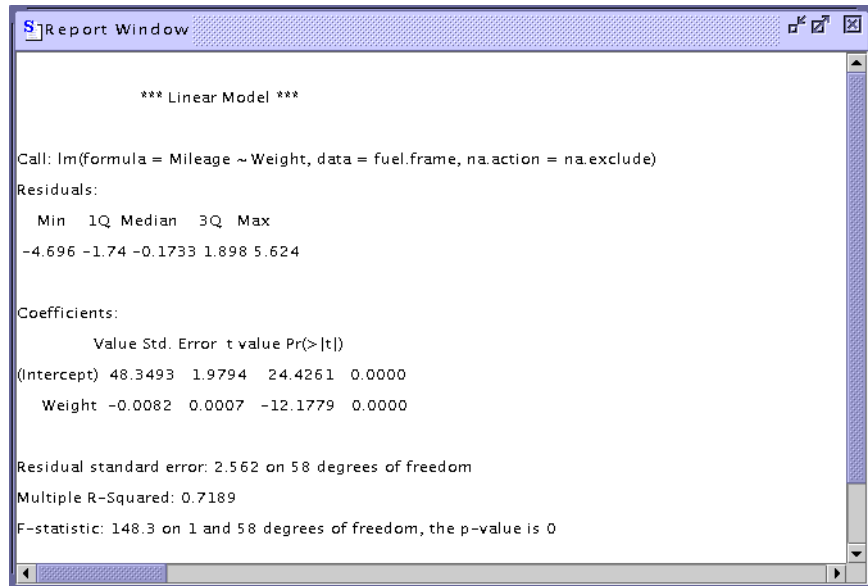


Figure 1.5: A *Report* window showing a linear fit for the `fuel1` frame data.

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.

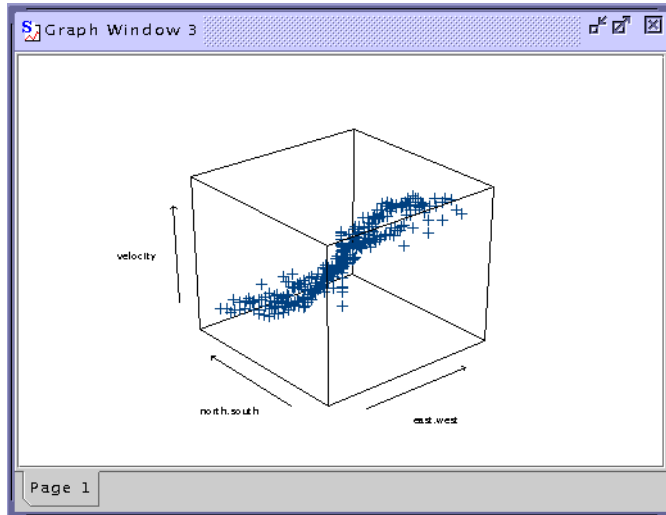


Figure 1.6: *A point cloud of the galaxy data.*

