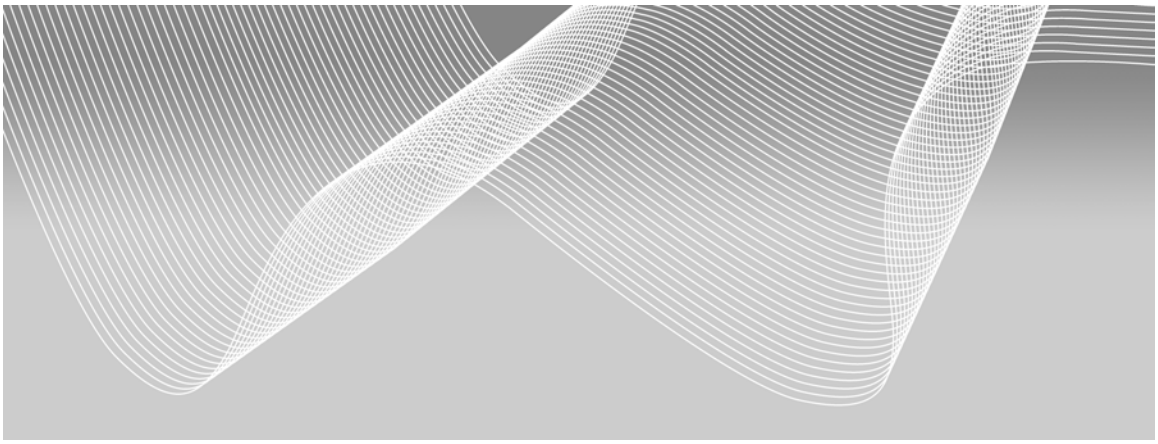


Section 1

Introduction to JSL

What is JSL? Beginning with the release of JMP 4 in 2000, a scripting language became a key feature of the software. The JMP scripting language, known as JSL, is made up of various commands that, when put together appropriately, instruct JMP to perform designated tasks automatically.

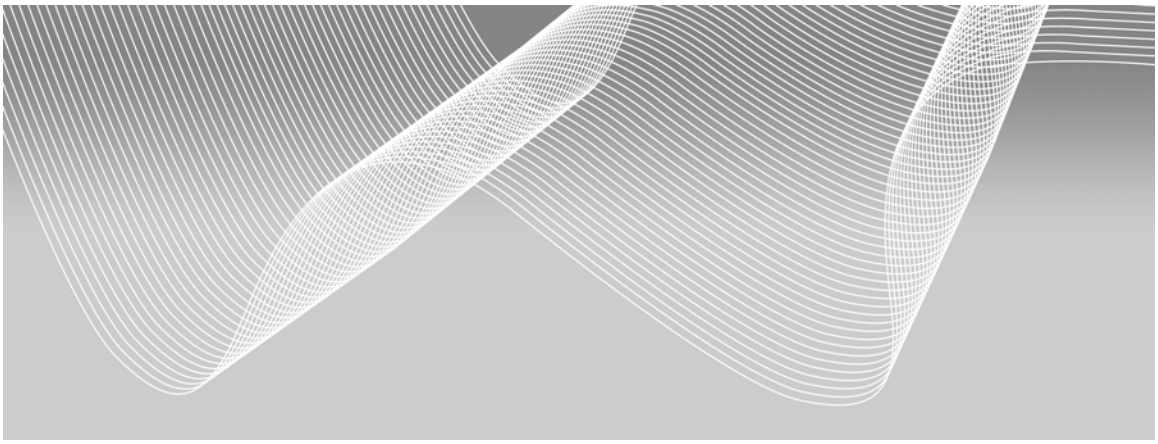
What is a JSL script? A JSL script is a program that includes a series of JSL commands that process various tasks during a single execution. JSL offers you the freedom to create scripts from the very simple and specific to the most generic and complex.

Some examples of actions that can be scripted are:

- Opening a data table
- Adding columns
- Selecting rows
- Creating subset or summary tables
- Performing various analyses
- Saving data tables, journals, and more

This list is by no means exhaustive. In fact, it is only the beginning.

It might be best to ignore all thoughts of other programming languages that you are familiar with, because JSL is quite distinctive. Please don't try to figure out what language it is similar to at this point, as it might cause unnecessary frustration and headaches!



Chapter One

Make JMP Work for You: Harnessing the Power of Scripts Generated by JMP

Overview

Have you ever worried that you will have to write pages and pages of code, and won't know where to start? There's no need to fret, because JMP, the best scripter of all, can write the scripts for you. We're going to show you how.

In this chapter, you will learn about:

- Capturing scripts from your analyses
- Using scripts from text import
- Creating a combined script composed of two separate captured scripts

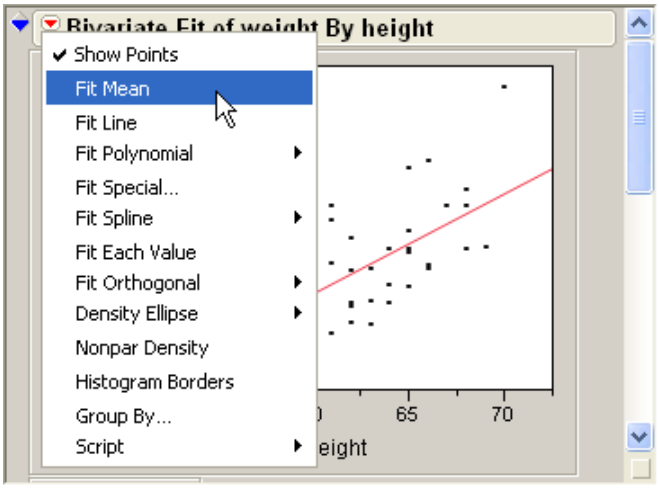
Capturing Scripts from Your Analyses

You just created a report that impresses your manager. He likes it so much that he wants the report weekly. So, what to do?

Do not panic. Instead, save the scripts and execute them next week to create your report.

In your report, you might have noticed the red triangle icons, clicked on a few of the icons, and used some of the options in the drop-down menus to add or subtract portions of your analysis. See Figure 1.1 for an example.

Figure 1.1 Bivariate Menu

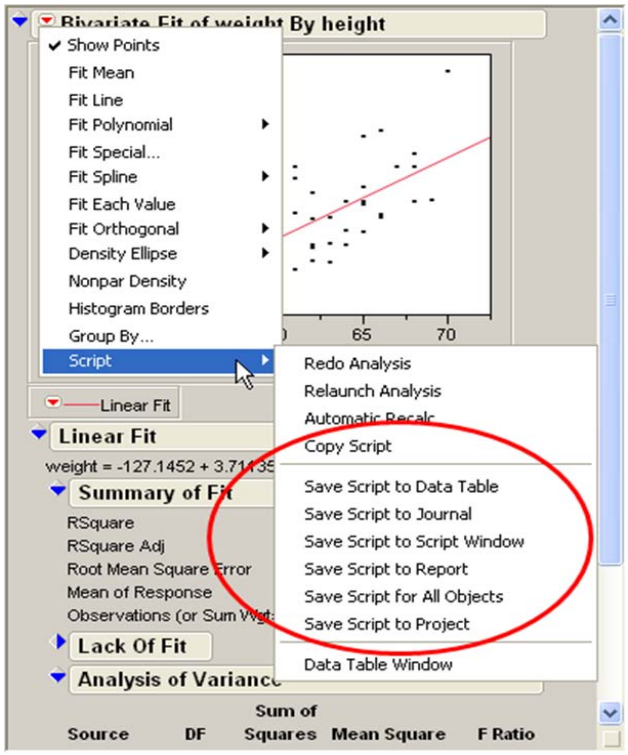


Did you know that you can capture a script of your analysis, plus many of the post-analysis changes you made? While JMP does not record every step you perform interactively, you can reproduce your results with scripting.

In Figure 1.2, notice the last item on the menu is **Script**. Selecting it opens a sub-menu that itemizes choices for saving the analysis script to regenerate the report, including most options.

Scripts that are generated by JMP can be captured in a variety of ways using selections in the **Script** menu.

Figure 1.2 Script Menu



Let's take a look at three of these options.

Copying Scripts

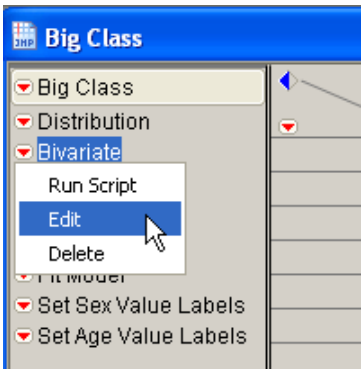
This is an easy one. The **Copy Script** option places the script on the clipboard, and then you can paste the script wherever you need it.

Saving Scripts to a Data Table

Keeping a script with its associated data is a good idea. The **Save Script to Data Table** command stores the script as a property of the data table. As a table property/script, it stays with the table until you delete it. You can add data to the table, execute the script, and see the results. And, when you save the table you also save the script. So when you give the data table to a colleague, she can open the table, execute the scripts with the data you intended, and view your reports**.**

A red triangle icon is created in the table panel and is labeled with the analysis name. Clicking on the icon opens a menu with three items: **Run Script**, **Edit**, and **Delete**.

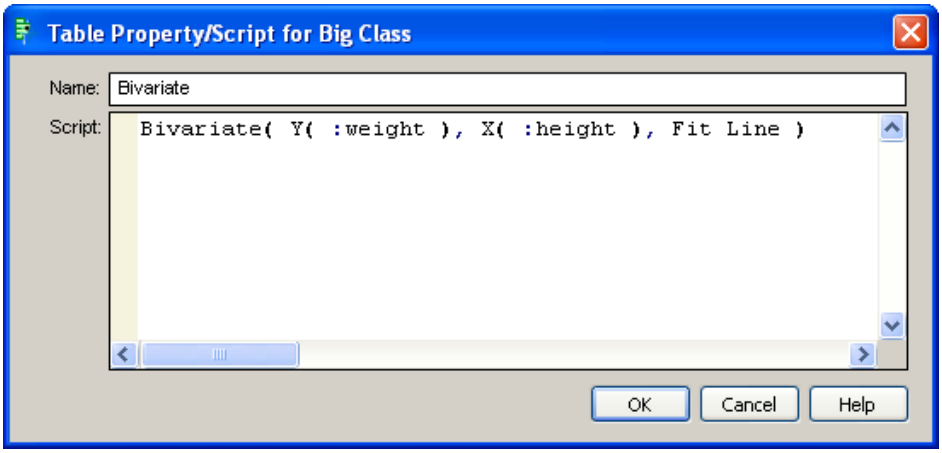
Figure 1.3 Script Property Options



Choosing **Run Script** executes the script.

Selecting **Edit** opens a window where you can view and edit the script.

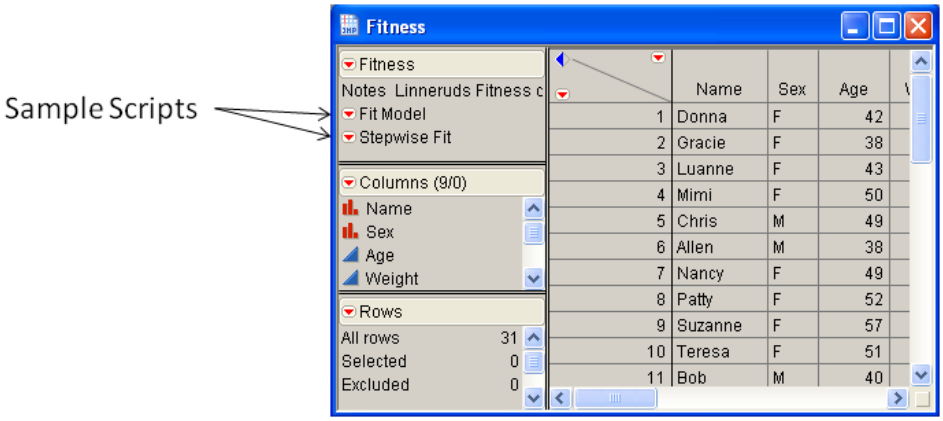
Figure 1.4 Sample Script That Was Saved as a Table Property/Script



Selecting **Delete** removes the script as a table property.

Did you know that many of the sample data tables in your JMP installation include scripts that have already been saved? Click on the red triangle icon to the left of a script that interests you, then select **Edit**, and you'll see a ready-made script.

Figure 1.5 Table Panel with Saved Scripts

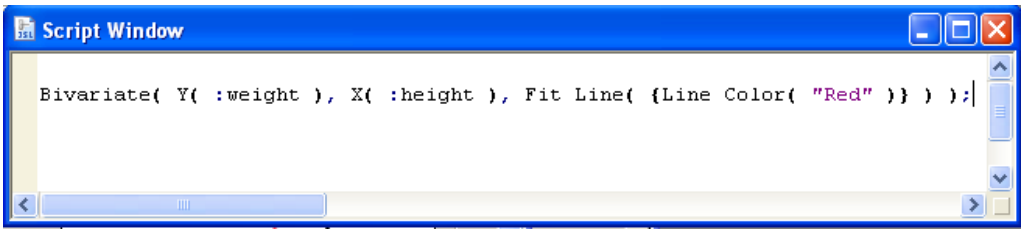


Saving Scripts to a Script Window

The **Save Script to Script Window** option places your script into a window named *Script Window*. This window is a script editor where you can edit your saved script. If this window is kept open, you can accumulate scripts from additional analyses by issuing the **Save Script to Script Window** command from those reports.

It is easy to save your script as a script file from this window. Select **File** ▶ **Save As**, and then select **Save As Type, JMP Scripts (*.jsl)**.

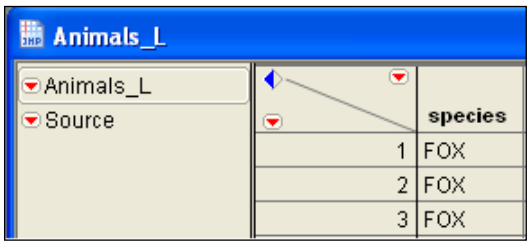
Figure 1.6 Script Window with Bivariate Script



Using Scripts from Text Import

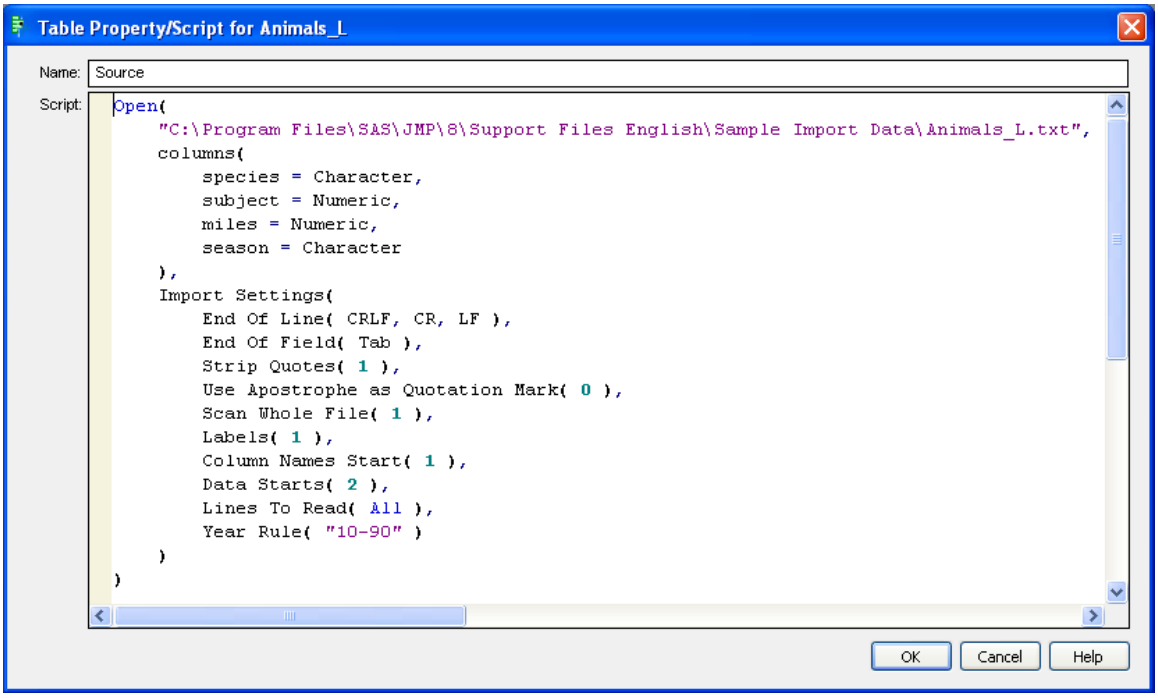
When you open a text file as a data table, an import script named **Source** is added as a table property.

Figure 1.7 Text Import Table



Click on the red triangle icon, and select **Edit** to see the code that JMP generated to reproduce the import of the text file. Note that the code consists of an **Open** statement with all possible settings needed to import the file.

Figure 1.8 Source Script for Text Import



Creating a Combined Script Composed of Two Separate Captured Scripts

Now that you've seen how easy it is to capture scripts, we are going to show you something a little more interesting. Remember that your manager wants the report generated each week? We will show you how easy it is to create a script that imports the data *and* makes the report.

In this sample, we put together a script that

- Imports the text data
- Creates a distribution analysis

Importing the Text Data

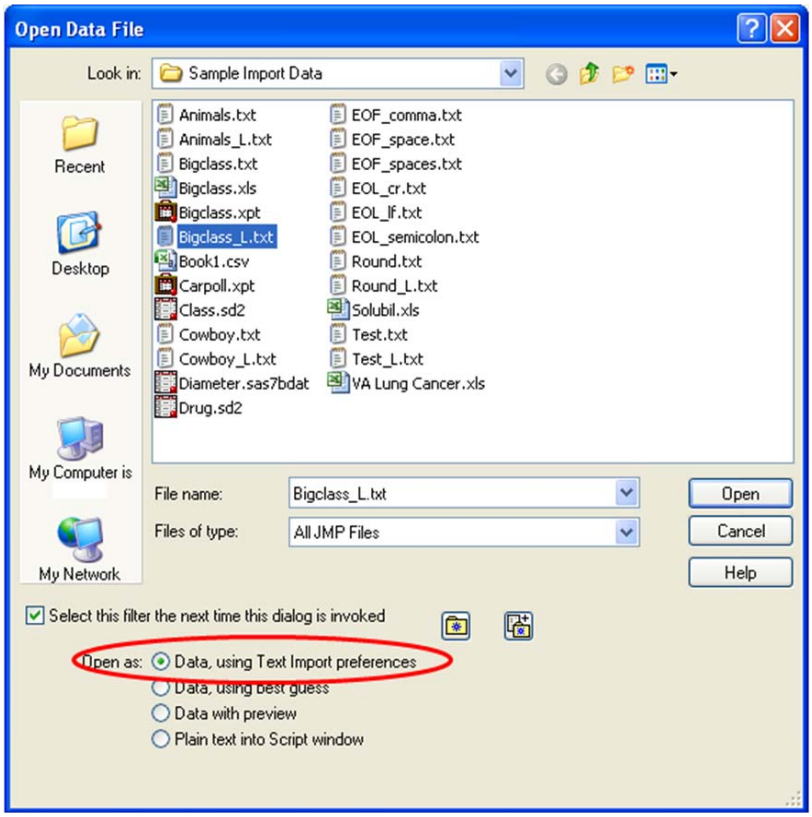
Let's begin by importing the Bigclass_L.txt file.

1. Select **File** ▶ **Open**. You can find this file in the Sample Import Data folder. For a typical JMP 8 Windows installation, you would find the file here:

   C:\Program Files\SAS\JMP\8\Support Files English\Sample Import Data

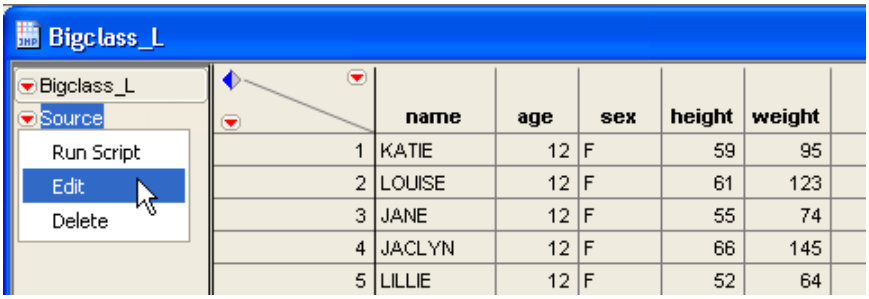
   Bigclass_L.txt is a tab-delimited file. If you use the **Data, using Text Import Preferences** option as shown in Figure 1.9, tab must be specified as an End of Field option in your text import preferences.

Figure 1.9 Open Text File for Windows



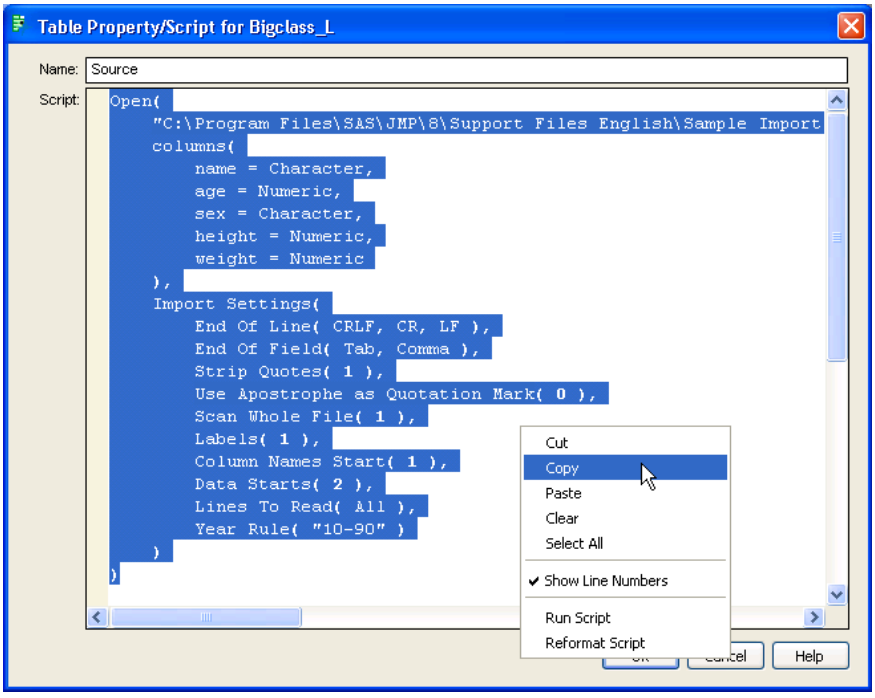
2. After the data is imported into JMP, click the red triangle icon beside **Source** and select **Edit**.

Figure 1.10 Editing the Source Script



3. In the resulting window, select the entire script and copy it by right-clicking the selected text and selecting **Copy**.

Figure 1.11 Copy of Source Script



4. Click **OK** to dismiss the Source Table Script window.
5. From the **File** menu, select **New** ▶ **Script**.
6. Paste the script by clicking the **Edit** menu and selecting **Paste**.
7. Add a semicolon after the last closing parenthesis, because you will be adding more code.

The semicolon, known as the **Glue** operator, is necessary in this case because it tells JMP to expect further JSL statements. Additional details about the **Glue** operator can be found in the *JMP Scripting Guide*.

Figure 1.12 Paste of Import Code

```
Script
Open(
    "C:\Program Files\SAS\JMP\8\Support Files English\Sample Import Data\Bigclass_L.txt",
    columns(
        name = Character,
        age = Numeric,
        sex = Character,
        height = Numeric,
        weight = Numeric
    ),
    Import Settings(
        End Of Line( CRLF, CR, LF ),
        End Of Field( Tab ),
        Strip Quotes( 1 ),
        Use Apostrophe as Quotation Mark( 0 ),
        Scan Whole File( 1 ),
        Labels( 1 ),
        Column Names Start( 1 ),
        Data Starts( 2 ),
        Lines To Read( All ),
        Year Rule( "10-90" )
    )
);
```

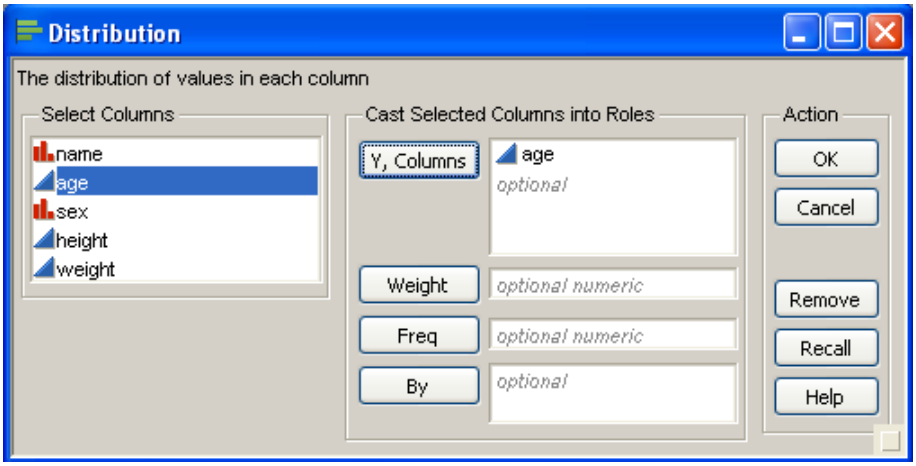
8. Press the **Enter** key a couple of times to move the cursor down a few lines.

Creating a Distribution Analysis

So far, we have pasted the script to import a text file into a Script window. Now, we will create a Distribution analysis and save its script.

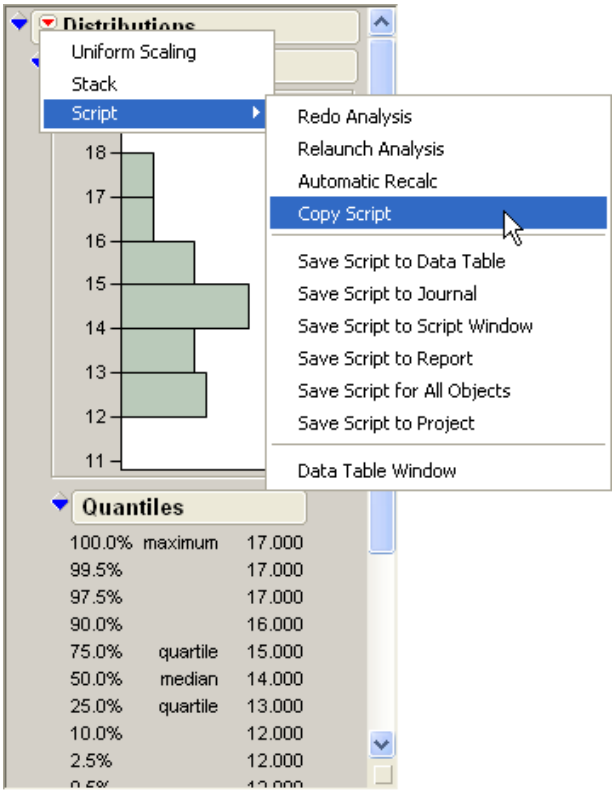
1. From the **Analyze** menu, select **Distribution**.
2. Cast **age** in the **Y, Columns** role and click **OK**.

Figure 1.13 Distribution Dialog Window



3. In the Distribution analysis window, capture the script by clicking the uppermost red triangle and selecting **Script ▶ Copy Script.**

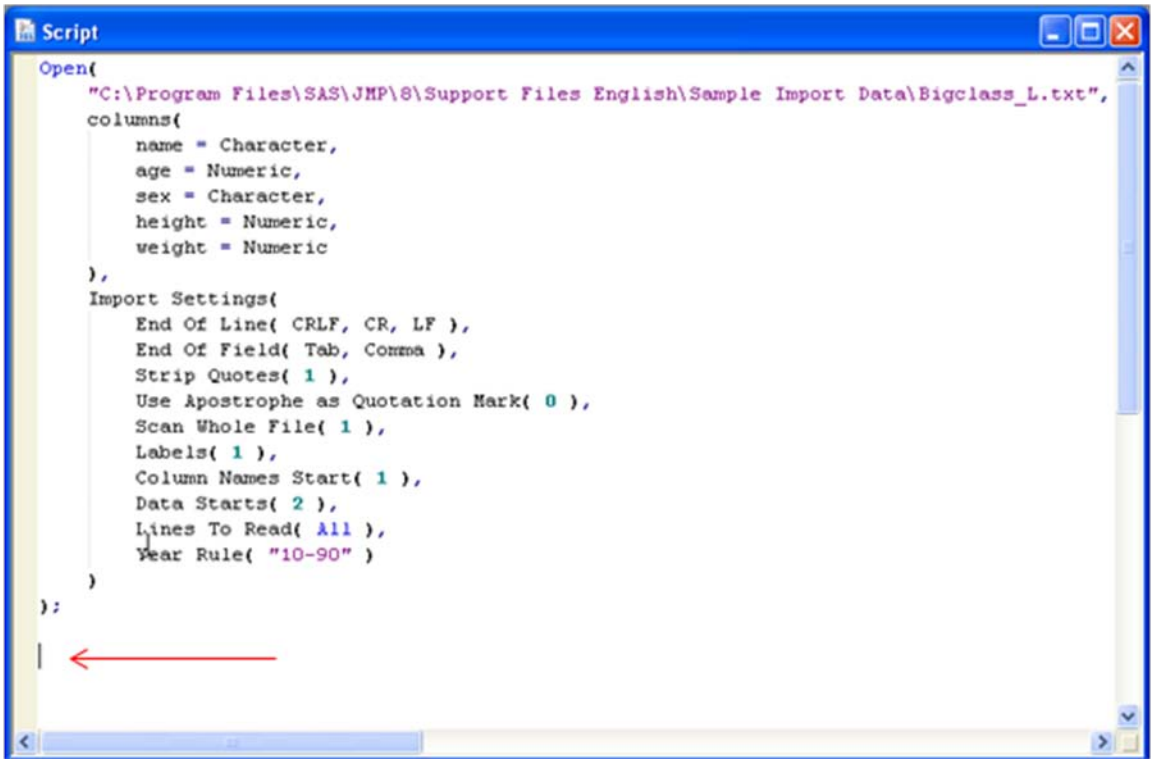
Figure 1.14 Copy Script to Clipboard



This action saves the distribution script onto the clipboard.

Now return to the script window that contains the text import script, and place the cursor in the space below the semicolon, near the bottom of the window.

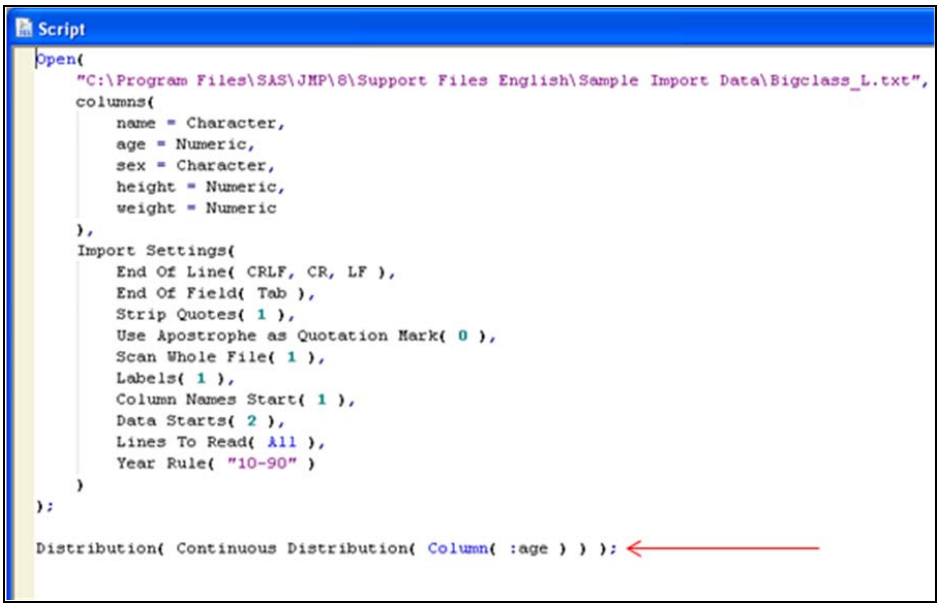
Figure 1.15 Placing the Cursor



4. To paste the distribution script from the clipboard into the Script window, select **Edit ▶ Paste**.

5. Add a semicolon after the last closing parenthesis in case you later decide to add code, and because this is a good programming practice.

Figure 1.16 Script Window with Distribution Script Pasted



6. Close the Distribution analysis window we created interactively, and then close the data table, Bigclass_L, leaving the Script window open.

Finally, let's look at the different methods we can use to initiate script execution and run the script to verify that it produces the expected results.

Executing a Script

There are several ways to execute a JSL script:

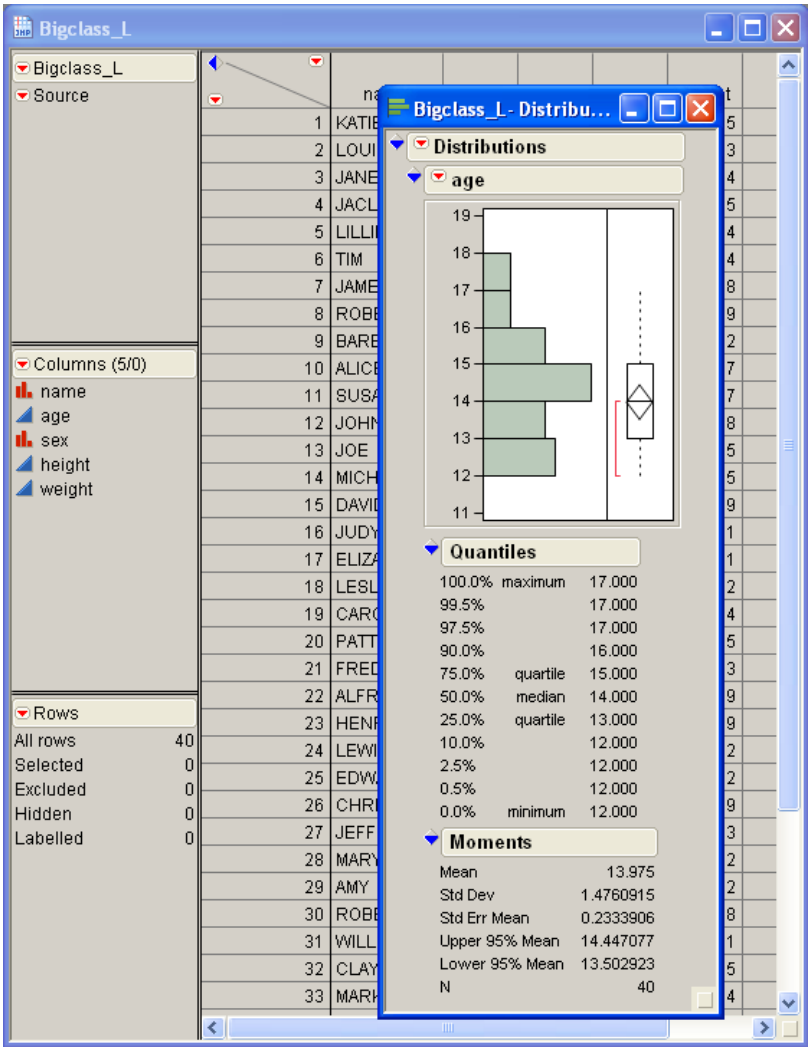
- From the **Edit** menu, select **Run Script**.
- Click the **Run Script** button on the toolbar.
- Right-click anywhere in the Script Editor window, and select **Run Script** from the pop-up menu.
- Use the keyboard shortcut for this same action: **CTRL+R**.
- Double-click a JSL file from a file browser.

To *execute* or *run* a script means the same thing, and we might use the terms interchangeably throughout this book.

For this case, we will use the first method. From the **Edit** menu, select **Run Script**.

Now you will see the text imported into a data table, and the distribution analysis executed on that data.

Figure 1.17 Results of Executing Combined Captured Scripts



Checking the Log

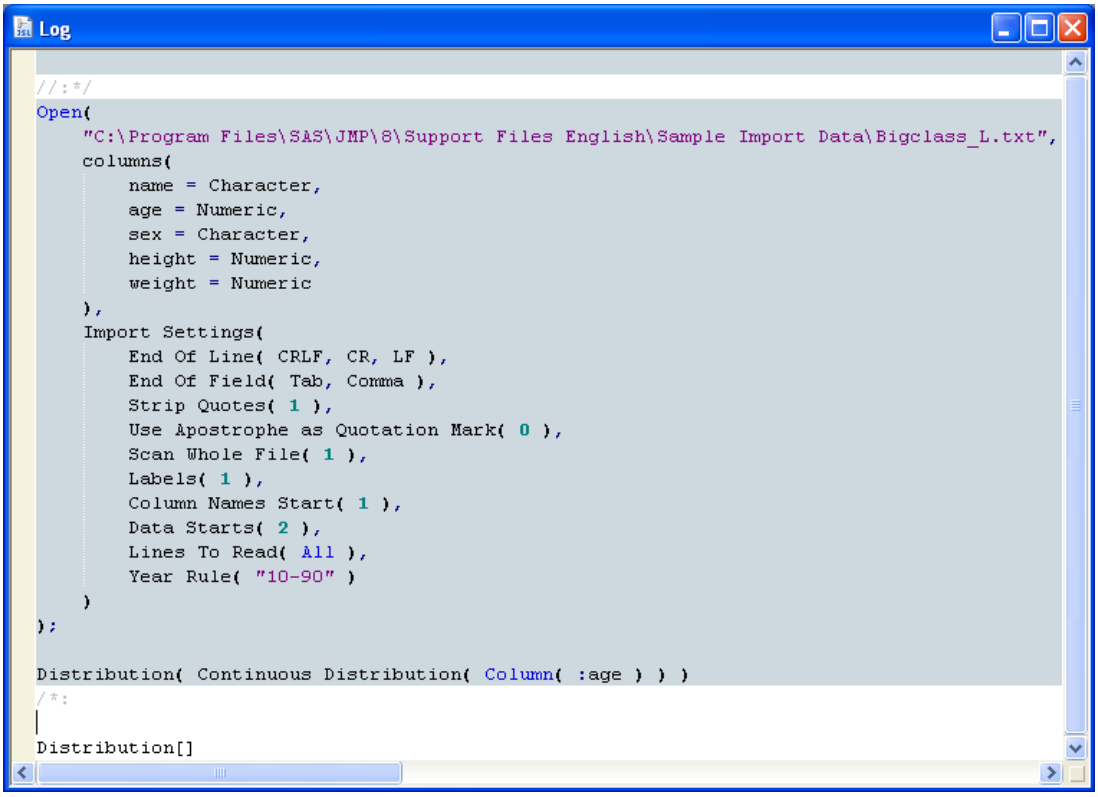
The Log window is where the code that you executed is displayed, along with any messages JMP has returned. If you have not already done so, display the Log window by clicking the **View** menu, and then select **Log**.

You can either leave the Log window docked at the bottom of your JMP window so that it is always in view, or you can allow it to float, like most of the other windows inside JMP. To float the Log window, right-click anywhere inside the log, and then select **Float Log Window**.

The Log window is also a Script Editor, which means that you can select code and execute it directly from within the log.

If your Log window was open prior to executing your script, you will see the script echoed in the log.

Figure 1.18 Log Window Showing Code

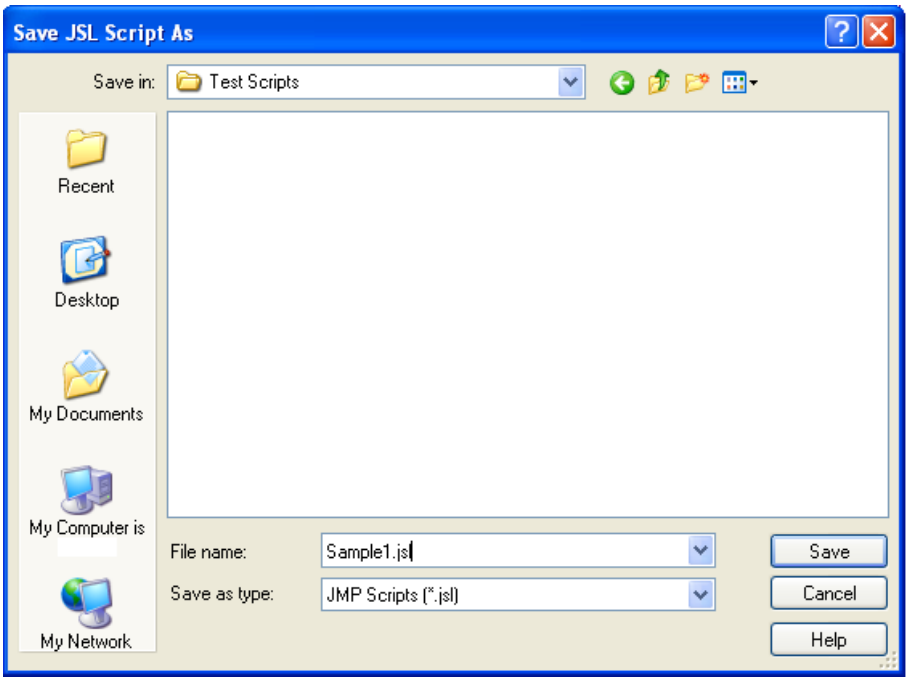


Saving a Script

Let's save the script because we will be coming back to it later.

1. Bring the script that we created to the forefront of the JMP application by selecting **Window** ▶ **Script**.
2. On the **File** menu, select **Save As**.
3. Browse to a convenient, yet memorable, location and name the script **Sample1.JSL.**
4. Click the **Save** button.

Figure 1.19 Save Sample1.JSL



You are finished! You have put together an entire script that will

- Import text data
- Create a Distribution report

Summary

Now you've learned how JMP can work for you by creating scripts of your reports and for your imported data.

What's next? It's time to roll up your sleeves, because in the next chapter we will show you how to stitch together multiple scripts using the Script Editor.