

*Computing Primer  
for  
Applied Linear  
Regression, Third Edition  
Using JMP*

Katherine St. Clair & Sanford Weisberg  
Department of Mathematics, Colby College  
School of Statistics, University of Minnesota  
August 3, 2009

©2005, Sanford Weisberg

**Home Website: [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr)**



# Contents

<i>Introduction</i>	<i>1</i>
0.1 <i>Organization of this primer</i>	<i>4</i>
0.2 <i>Data files</i>	<i>5</i>
0.2.1 <i>Documentation</i>	<i>5</i>
0.2.2 <i>Getting the data files for JMP</i>	<i>6</i>
0.2.3 <i>Getting the data in text files</i>	<i>6</i>
0.2.4 <i>An exceptional file</i>	<i>6</i>
0.3 <i>Scripts</i>	<i>6</i>
0.4 <i>The very basics</i>	<i>7</i>
0.4.1 <i>Reading a data file</i>	<i>7</i>
0.4.2 <i>Saving text output and graphs</i>	<i>9</i>
0.4.3 <i>Normal, F, t and <math>\chi^2</math> tables</i>	<i>10</i>
0.5 <i>Abbreviations to remember</i>	<i>12</i>
0.6 <i>Copyright and Printing this Primer</i>	<i>12</i>
1 <i>Scatterplots and Regression</i>	<i>13</i>
1.1 <i>Scatterplots</i>	<i>13</i>
1.2 <i>Mean functions</i>	<i>16</i>
1.3 <i>Variance functions</i>	<i>17</i>
1.4 <i>Summary graph</i>	<i>17</i>

1.5	<i>Tools for looking at scatterplots</i>	17
1.6	<i>Scatterplot matrices</i>	17
2	<i>Simple Linear Regression</i>	19
2.1	<i>Ordinary least squares estimation</i>	19
2.2	<i>Least squares criterion</i>	19
2.3	<i>Estimating <math>\sigma^2</math></i>	20
2.4	<i>Properties of least squares estimates</i>	20
2.5	<i>Estimated variances</i>	20
2.6	<i>Comparing models: The analysis of variance</i>	20
2.7	<i>The coefficient of determination, <math>R^2</math></i>	21
2.8	<i>Confidence intervals and tests</i>	22
2.9	<i>The Residuals</i>	24
3	<i>Multiple Regression</i>	27
3.1	<i>Adding a term to a simple linear regression model</i>	27
3.2	<i>The Multiple Linear Regression Model</i>	27
3.3	<i>Terms and Predictors</i>	27
3.4	<i>Ordinary least squares</i>	28
3.5	<i>The analysis of variance</i>	28
3.6	<i>Predictions and fitted values</i>	29
4	<i>Drawing Conclusions</i>	31
4.1	<i>Understanding parameter estimates</i>	31
4.1.1	<i>Rate of change</i>	31
4.1.2	<i>Sign of estimates</i>	31
4.1.3	<i>Interpretation depends on other terms in the mean function</i>	31
4.1.4	<i>Rank deficient and over-parameterized models</i>	31
4.2	<i>Experimentation versus observation</i>	32
4.3	<i>Sampling from a normal population</i>	32
4.4	<i>More on <math>R^2</math></i>	32
4.5	<i>Missing data</i>	32
4.6	<i>Computationally intensive methods</i>	34
5	<i>Weights, Lack of Fit, and More</i>	35
5.1	<i>Weighted Least Squares</i>	35
5.1.1	<i>Applications of weighted least squares</i>	36
5.1.2	<i>Additional comments</i>	38

5.2	<i>Testing for lack of fit, variance known</i>	38
5.3	<i>Testing for lack of fit, variance unknown</i>	38
5.4	<i>General F testing</i>	38
5.5	<i>Joint confidence regions</i>	39
6	<i>Polynomials and Factors</i>	41
6.1	<i>Polynomial regression</i>	41
6.1.1	<i>Polynomials with several predictors</i>	41
6.1.2	<i>Using the delta method to estimate a minimum or a maximum</i>	44
6.1.3	<i>Fractional polynomials</i>	44
6.2	<i>Factors</i>	44
6.2.1	<i>No other predictors</i>	45
6.2.2	<i>Adding a predictor: Comparing regression lines</i>	46
6.3	<i>Many factors</i>	46
6.4	<i>Partial one-dimensional mean functions</i>	46
6.5	<i>Random coefficient models</i>	50
7	<i>Transformations</i>	51
7.1	<i>Transformations and scatterplots</i>	51
7.1.1	<i>Power transformations</i>	51
7.1.2	<i>Transforming only the predictor variable</i>	51
7.1.3	<i>Transforming the response only</i>	52
7.1.4	<i>The Box and Cox method</i>	54
7.2	<i>Transformations and scatterplot matrices</i>	54
7.2.1	<i>The 1D estimation result and linearly related predictors</i>	55
7.2.2	<i>Automatic choice of transformation of the predictors</i>	55
7.3	<i>Transforming the response</i>	55
7.4	<i>Transformations of non-positive variables</i>	55
8	<i>Regression Diagnostics: Residuals</i>	57
8.1	<i>The residuals</i>	57
8.1.1	<i>Difference between <math>\hat{e}</math> and <math>e</math></i>	57
8.1.2	<i>The hat matrix</i>	57
8.1.3	<i>Residuals and the hat matrix with weights</i>	57
8.1.4	<i>The residuals when the model is correct</i>	58
8.1.5	<i>The residuals when the model is not correct</i>	58
8.1.6	<i>Fuel consumption data</i>	58
8.2	<i>Testing for curvature</i>	59

8.3	<i>Nonconstant variance</i>	59
8.3.1	<i>Variance Stabilizing Transformations</i>	59
8.3.2	<i>A diagnostic for nonconstant variance</i>	59
8.3.3	<i>Additional comments</i>	60
8.4	<i>Graphs for model assessment</i>	60
8.4.1	<i>Checking mean functions</i>	60
8.4.2	<i>Checking variance functions</i>	60
9	<i>Outliers and Influence</i>	61
9.1	<i>Outliers</i>	61
9.1.1	<i>An outlier test</i>	61
9.1.2	<i>Weighted least squares</i>	61
9.1.3	<i>Significance levels for the outlier test</i>	61
9.1.4	<i>Additional comments</i>	62
9.2	<i>Influence of cases</i>	62
9.2.1	<i>Cook's distance</i>	62
9.2.2	<i>Magnitude of <math>D_i</math></i>	62
9.2.3	<i>Computing <math>D_i</math></i>	63
9.2.4	<i>Other measures of influence</i>	63
9.3	<i>Normality assumption</i>	63
10	<i>Variable Selection</i>	65
10.1	<i>The Active Terms</i>	65
10.1.1	<i>Collinearity</i>	66
10.1.2	<i>Collinearity and variances</i>	67
10.2	<i>Variable selection</i>	67
10.2.1	<i>Information criteria</i>	67
10.2.2	<i>Computationally intensive criteria</i>	67
10.2.3	<i>Using subject-matter knowledge</i>	67
10.3	<i>Computational methods</i>	67
10.3.1	<i>Subset selection overstates significance</i>	70
10.4	<i>Windmills</i>	70
10.4.1	<i>Six mean functions</i>	70
10.4.2	<i>A computationally intensive approach</i>	70
11	<i>Nonlinear Regression</i>	71
11.1	<i>Estimation for nonlinear mean functions</i>	71
11.2	<i>Inference assuming large samples</i>	71

11.3	<i>Bootstrap inference</i>	72
11.4	<i>References</i>	72
12	<i>Logistic Regression</i>	73
12.1	<i>Binomial Regression</i>	73
12.1.1	<i>Mean Functions for Binomial Regression</i>	73
12.2	<i>Fitting Logistic Regression</i>	73
12.2.1	<i>One-predictor example</i>	74
12.2.2	<i>Many Terms</i>	76
12.2.3	<i>Deviance</i>	78
12.2.4	<i>Goodness of Fit Tests</i>	78
12.3	<i>Binomial Random Variables</i>	79
12.3.1	<i>Maximum likelihood estimation</i>	79
12.3.2	<i>The Log-likelihood for Logistic Regression</i>	79
12.4	<i>Generalized linear models</i>	79
	<i>References</i>	81
	<i>Index</i>	83

# 0

---

## *Introduction*

This computer primer supplements the book *Applied Linear Regression* (ALR), third edition, by Sanford Weisberg, published by John Wiley & Sons in 2005. It shows you how to do the analyses discussed in ALR using one of several general-purpose programs that are widely available throughout the world. All the programs have capabilities well beyond the uses described here. Different programs are likely to suit different users. We expect to update the primer periodically, so check [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr) to see if you have the most recent version. The versions are indicated by the date shown on the cover page of the primer.

Our purpose is largely limited to using the packages with ALR, and we will not attempt to provide a complete introduction to the packages. If you are new to the package you are using you will probably need additional reference material.

There are a number of methods discussed in ALR that are not (as yet) a standard part of statistical analysis, and some methods are not possible without writing your own programs to supplement the package you choose. *The exceptions to this rule are R and S-Plus. For these two packages we have written functions you can easily download and use for nearly everything in the book.*

Here are the programs for which primers are available.

**R** is a *command line* statistical package, which means that the user types a statement requesting a computation or a graph, and it is executed immediately. You will be able to use a package of functions for R that

will let you use all the methods discussed in ALR; we used R when writing the book.

R also has a programming language that allows automating repetitive tasks. R is a favorite program among academic statisticians because it is free, works on Windows, Linux/Unix and Macintosh, and can be used in a great variety of problems. There is also a large literature developing on using R for statistical problems. The main website for R is [www.r-project.org](http://www.r-project.org). From this website you can get to the page for downloading R by clicking on the link for CRAN, or, in the US, going to [cran.us.r-project.org](http://cran.us.r-project.org).

Documentation is available for R on-line, from the website, and in several books. We can strongly recommend two books. The book by Fox (2002) provides a fairly gentle introduction to R with emphasis on regression. We will from time to time make use of some of the functions discussed in Fox's book that are not in the base R program. A more comprehensive introduction to R is Venables and Ripley (2002), and we will use the notation `VR[3.1]`, for example, to refer to Section 3.1 of that book. Venables and Ripley has more computerese than does Fox's book, but its coverage is greater and you will be able to use this book for more than linear regression. Other books on R include Verzani (2005), Maindonald and Braun (2002), Venables and Smith (2002), and Dalgaard (2002). We used R Version 2.0.0 on Windows and Linux to write the package. A new version of R is released twice a year, so the version you use will probably be newer. If you have a fast internet connection, downloading and upgrading R is easy, and you should do it regularly.

**S-Plus** is very similar to R, and most commands that work in R also work in S-Plus. Both are variants of a statistical language called "S" that was written at Bell Laboratories before the breakup of AT&T. Unlike R, S-Plus is a commercial product, which means that it is not free, although there is a free student version available at [elms03.e-academy.com/splus](http://elms03.e-academy.com/splus). The website of the publisher is [www.insightful.com/products/splus](http://www.insightful.com/products/splus). A library of functions very similar to those for R is also available that will make S-Plus useful for all the methods discussed in ALR.

S-Plus has a well-developed graphical user interface or GUI. Many new users of S-Plus are likely to learn to use this program through the GUI, not through the command-line interface. In this primer, however, we make no use of the GUI.

If you are using S-Plus on a Windows machine, you probably have the manuals that came with the program. If you are using Linux/Unix, you may not have the manuals. In either case the manuals are available online; for Windows see the **Help** → **Online Manuals**, and for Linux/Unix use

```
> cd 'Splus $HOME'/doc
```

```
> ls
```

and see the pdf documents there. Chambers and Hastie (1993) provides the basics of fitting models with S languages like S-Plus and R. For a more general reference, we again recommend Fox (2002) and Venables and Ripley (2002), as we did for R. We used S-Plus Version 6.0 Release 1 for Linux, and S-Plus 6.2 for Windows. Newer versions of both are available.

**SAS** is the largest and most widely distributed statistical package in both industry and education. SAS also has a GUI. While it is possible to do *some* data analysis using the SAS GUI, the strength of this program is in the ability to write SAS programs, in the editor window, and then submit them for execution, with output returned in an output window. We will therefore view SAS as a *batch* system, and concentrate mostly on writing SAS commands to be executed. The website for SAS is [www.sas.com](http://www.sas.com).

SAS is very widely documented, including hundreds of books available through amazon.com or from the SAS Institute, and extensive on-line documentation. Muller and Fetterman (2003) is dedicated particularly to regression. We used Version 9.1 for Windows. We find the on-line documentation that accompanies the program to be invaluable, although learning to read and understand SAS documentation isn't easy.

Although SAS is a programming language, adding new functionality can be very awkward and require long, confusing programs. These programs could, however, be turned into SAS *macros* that could be reused over and over, so in principle SAS could be made as useful as R or S-Plus. We have not done this, but would be delighted if readers would take on the challenge of writing macros for methods that are awkward with SAS. Anyone who takes this challenge can send us the results ([sandy@stat.umn.edu](mailto:sandy@stat.umn.edu)) for inclusion in later revisions of the primer.

We have, however, prepared *script files* that give the programs that will produce all the output discussed in this primer; you can get the scripts from [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr).

**JMP** is another product of SAS Institute, and was designed around a clever and useful GUI. A student version of JMP is available. The website is [www.jmp.com](http://www.jmp.com). We used JMP Version 5.1 on Windows.

Documentation for the student version of JMP, called JMP-In, comes with the book written by Sall, Creighton and Lehman (2005), and we will write JMP-START[3] for Chapter 3 of that book, or JMP-START[P360] for page 360. The full version of JMP includes very extensive manuals; the manuals are available on CD only with JMP-In. Freund, Littell and Creighton (2003) discusses JMP specifically for regression.

JMP has a scripting language that could be used to add functionality to the program. We have little experience using it, and would be happy

to hear from readers on their experience using the scripting language to extend JMP to use some of the methods discussed in ALR that are not possible in JMP without scripting.

**SPSS** evolved from a batch program to have a very extensive graphical user interface. In the primer we use only the GUI for SPSS, which limits the methods that are available. Like SAS, SPSS has many sophisticated tools for data base management. A student version is available. The website for SPSS is [www.spss.com](http://www.spss.com). SPSS offers hundreds of pages of documentation, including SPSS (2003), with Chapter 26 dedicated to regression models. In mid-2004, amazon.com listed more than two thousand books for which “SPSS” was a keyword. We used SPSS Version 12.0 for Windows. A newer version is available.

This is hardly an exhaustive list of programs that could be used for regression analysis. If your favorite package is missing, please take this as a challenge: try to figure out how to do what is suggested in the text, and write your own primer! Send us a PDF file ([sandy@stat.umn.edu](mailto:sandy@stat.umn.edu)) and we will add it to our website, or link to yours.

One program missing from the list of programs for regression analysis is Microsoft’s spreadsheet program Excel. While *a few* of the methods described in the book can be computed or graphed in Excel, most would require great endurance and patience on the part of the user. There are many add-on statistics programs for Excel, and one of these may be useful for comprehensive regression analysis; we don’t know. If something works for you, please let us know!

A final package for regression that we should mention is called Arc. Like R, Arc is free software. It is available from [www.stat.umn.edu/arc](http://www.stat.umn.edu/arc). Like JMP and SPSS it is based around a graphical user interface, so most computations are done via point-and-click. Arc also includes access to a complete computer language, although the language, lisp, is considerably harder to learn than the S or SAS languages. Arc includes all the methods described in the book. The use of Arc is described in Cook and Weisberg (1999), so we will not discuss it further here; see also Weisberg (2005).

## 0.1 ORGANIZATION OF THIS PRIMER

The primer often refers to specific problems or sections in ALR using notation like ALR[3.2] or ALR[A.5], for a reference to Section 3.2 or Appendix A.5, ALR[P3.1] for Problem 3.1, ALR[F1.1] for Figure 1.1, ALR[E2.6] for an equation and ALR[T2.1] for a table. Reference to, for example, “Figure 7.1,” would refer to a figure in this primer, not to ALR. Chapters, sections, and homework problems are numbered in this primer as they are in ALR. Consequently, the section headings in primer refers to the material in ALR, and not necessarily the material in the primer. Many of the sections in this primer don’t have any

Table 0.1 The data file `htwt.txt`.

---

Ht	Wt
169.6	71.2
166.8	58.2
157.1	56
181.1	64.5
158.4	53
165.6	52.4
166.7	56.8
156.5	49.2
168.1	55.6
165.3	77.8

---

material because that section doesn't introduce any new issues with regard to computing. The index should help you navigate through the primer.

There are four versions of this primer, one for R and S-Plus, and one for each of the other packages. All versions are available for free as PDF files at [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr).

Anything you need to type into the program will always be in **this font**. Output from a program depends on the program, but should be clear from context. We will write `File` to suggest selecting the menu called "File," and `Transform → Recode` to suggest selecting an item called "Recode" from a menu called "Transform." You will sometimes need to push a button in a dialog, and we will write "push OK" to mean "click on the button marked 'OK'." For non-English versions of some of the programs, the menus may have different names, and we apologize in advance for any confusion this causes.

## 0.2 DATA FILES

### 0.2.1 Documentation

Documentation for nearly all of the data files is contained in ALR; look in the index for the first reference to a data file. Separate documentation can be found in the file `alr3data.pdf` in PDF format at the web site [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr).

The data are available in a *package* for R, in a *library* for S-Plus and for SAS, and as a directory of files in special format for JMP and SPSS. In addition, the files are available as plain text files that can be used with these, or any other, program. Table 0.1 shows a copy of one of the smallest data files called `htwt.txt`, and described in ALR[P3.1]. This file has two variables, named *Ht* and *Wt*, and ten cases, or rows in the data file. The largest file is `wm5.txt` with 62,040 cases and 14 variables. This latter file is so large that it is handled differently from the others; see Section 0.2.4.

A few of the data files have missing values, and these are generally indicated in the file by a place-holder in the place of the missing value. For example, for R and S-Plus, the placeholder is NA, while for SAS it is a period “.” Different programs handle missing values a little differently; we will discuss this further when we get to the first data set with a missing value in Section 4.5.

### 0.2.2 Getting the data files for JMP

Go to the JMP page at [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr), and follow the directions to download the directory of data files in a special format for use with JMP. To use a file, you can either double-click on its name, or start JMP, select File → Open, and and browse to the file name. To data referred to in the text as `heights.txt` will be called `heights.jmp`.

### 0.2.3 Getting the data in text files

You can download the data as a directory of plain text files, or as individual files; see [www.stat.umn.edu/alr/data](http://www.stat.umn.edu/alr/data). *Missing values on these files are indicated with a ?.* *If your program does not use this missing value character, you may need to substitute a different character using an editor.*

### 0.2.4 An exceptional file

**The file `wm5.txt` is not included in any of the compressed files, or in the libraries.** This one file is nearly five megabytes long, requiring as much space as all the other files combined. If you need this file, for ALR[P10.12], you can download it separately from [www.stat.umn.edu/alr/data](http://www.stat.umn.edu/alr/data).

## 0.3 SCRIPTS

For R, S-Plus, and SAS, we have prepared *script files* that can be used while reading this primer. For R and S-Plus, the scripts will reproduce nearly every computation shown in ALR; indeed, these scripts were used to do the calculations in the first place. For SAS, the scripts correspond to the discussion given in this primer, but will not reproduce everything in ALR. The scripts can be downloaded from [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr) for R, S-Plus or SAS.

Although both JMP and SPSS have scripting or programming languages, we have not prepared scripts for these programs. Some of the methods discussed in ALR are not possible in these programs without the use of scripts, and so we encourage readers to write scripts in these languages that implement these ideas. Topics that require scripts include bootstrapping and computer intensive methods, ALR[4.6]; partial one-dimensional models, ALR[6.4], inverse response plots, ALR[7.1, 7.3], multivariate Box-Cox transformations, ALR[7.2],

Yeo-Johnson transformations, ALR[7.4], and heteroscedasticity tests, ALR[8.3.2]. There are several other places where usability could be improved with a script.

If you write scripts you would like to share with others, let me know (sandy@stat.umn.edu) and I'll make a link to them or add them to the website.

## 0.4 THE VERY BASICS

Before you can begin doing any useful computing, you need to be able to read data into the program, and after you are done you need to be able to save and print output and graphs. All the programs are a little different in how they handle input and output, and we give some of the details here.

### 0.4.1 Reading a data file

Reading data into a program is surprisingly difficult. We have tried to ease this burden for you, at least when using the data files supplied with ALR, by providing the data in a special format for each of the programs. There will come a time when you want to analyze real data, and then you will need to be able to get your data into the program. Here are some hints on how to do it.

**JMP** We provide the data files for use with JMP both as *plain text files* and as files in the special format for JMP; see the web site for information on downloading these files. We expect that most readers will want to use the “.jmp” files because they are much easier to use. After you download the files in the format, you can use them with JMP with one of the following simple methods.

1. Browse to the file you want, and double-click on its name.
2. Start JMP, and select Open Data Table from the JMP starter, and then browse to the file you want.
3. Start JMP, and select File → Open, and then browse to the file you want.

Reading plain data files is an important skill in using any program, and we discuss here how you could read the plain data files for the book that you can get from [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr) into JMP.

We suggest that you change your “import preferences” for importing data. On Windows, if someone else administers your computer for you, you may not have permission to change the preferences file; you should then skip the next two paragraphs.

To change the import preferences of JMP, select File → Preferences or the PREFERENCES button from the starter window, then choose the Text Import/

Export tab. Since the ALR data files use a space to separate the variable columns, under the field Import Settings check the option labelled `space` and press OK. Your changed preferences become permanent, until you change them again.

A data file from ALR can now be used by selecting `File → Open` or pushing `OPEN DATA TABLE` from the Starter window. From the dialog box choose the directory containing the data files. To see the `.txt` files, you may need to select either “Text import files” or “All files” from the `Files of type` menu near the bottom of the dialog. You can then select the `.txt` file of interest. Always double check that JMP has correctly determined the type of each of the variables entered, either continuous, for numerical data, nominal, for values that are really categories, or ordinal, or ordered categories. We won’t use ordinal variables with the data sets in ALR.

If you choose not to change the import preferences of JMP you can still read a data file by following the `File → Open` menu to the Open Data File Window. You again change the file type to `Text Import` but then check the option `Attempt to Discern Format` (instead of using the preselected option `Use Text Import Preferences`) and press Open.

Finally, if you know the data file you are reading contains missing values, or if you follow the steps above and find missing values, select the file type `Text Import Preview` from the open file dialog. Press `DELIMITED` to import the file and click OK in the preview dialog. We used this option because we found that data files imported under the saved preferences above will treat the missing value indicators as characters and convert their columns to a nominal type.

An open data file in JMP contains two components: the left component is the “Data Table panel” and the right is the “Data grid”. The data grid is a spreadsheet containing the actual data. The data table panel contains panels for the table, columns, and rows. From the column panel, shown in Figure 0.1, you can change the variable type (continuous, nominal, ordinal) by clicking on the small box to the left of the variable name. These types are referred to as *modeling types* in JMP because they determine the type of analysis that can be performed with the variable. For example, fitting two variables using the Fit Y by X analysis platform will result in a regression analysis if the two variables are continuous, but changing the independent variable to ordinal will result in a change of analysis, now a one-way analysis of variance will be fit.

Any variable can be transformed and added to the data table by right clicking on an empty column and selecting `New Column` or by selecting `Cols → New Column`. In the dialog that appears you name and describe the new variable. To determine the values to enter in the column, choose the `NEW PROPERTY` button and pick `Formula` from the list. The formula editor shown in Figure 0.2 will appear and you enter the appropriate formula. This editor can not only define a transformation of existing variables, but also calculate built-in statistical functions of column variables (e.g. mean, standard deviation) or even



Fig. 0.1 JMP column panel for the data `cakes.txt`. The N, O, C boxes to the left of the first three variables show they have modeling types nominal, ordinal, and continuous, respectively.

generate a column of random variables. `JMP-START[4]` gives a good overview of the formula editor. When a column is defined by a formula a yellow cross will appear to the right of the variable name and it will remain linked to any other columns used in the formula. Changing these columns will result in the appropriate changes in the formula column.

#### 0.4.2 Saving text output and graphs

All the programs have many ways of saving text output and graphs. We will make no attempt to be comprehensive here.

**JMP** A simple way to save both tabular output and graphs in *JMP* is to copy and paste them into a word processing document. If you want to copy a whole report, simply choose `Edit → Copy` to store the active window to the clipboard, then paste the results in your document. You can save just a portion of the results by choosing the “fat-plus” tool from the toolbar or `Tools → Selection`, then clicking on the table, column, or graph you would like save. You then copy and paste as before. You can also save selected graphs or tables by choosing `Edit → Save Selection As`. This option only allows you to save the selected item as graphics with an extension `.png`, `.jpg`, or `.wmf`.

*JMP* also provides methods which allow you to customize report output and save it as a graphic, text, Word, `.rtf`, or `.html` file. The text format is the only format which doesn’t save graphs. You cannot save output directly from the report, instead you must first duplicate it by selecting `Edit → Journal` or `Edit → Layout`. Both commands will reproduce the report in a new window and allow you to customize the look of the report. The layout option provides a few more editing options such as allowing you to ungroup parts of a report. Select `File → Save` and choose a format above to save a journal or layout window.

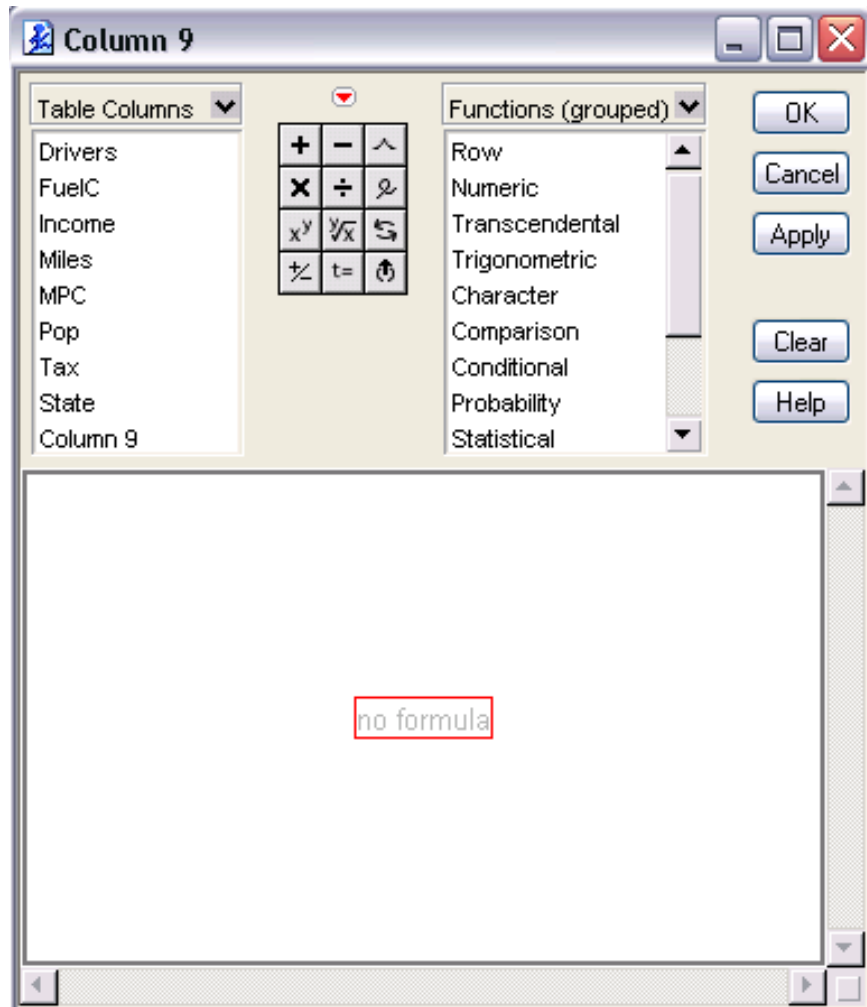


Fig. 0.2 JMP Formula Editor dialog for the data file fue12001.txt.

#### 0.4.3 Normal, $F$ , $t$ and $\chi^2$ tables

ALR does not include tables for looking up critical values and significance levels for standard distributions like the  $t$ ,  $F$  and  $\chi^2$ . Although these values can be computed with any of the programs we discuss in the primers, doing so is easy only with R and S-Plus. Also, the computation is fairly easy with Microsoft Excel. Table 0.2 shows the functions you need using Excel.

**JMP** You can get  $t$ ,  $F$  and  $\chi^2$  significance levels and critical values by writing a one line script using the JMP scripting language. First, select

*Table 0.2* Functions for computing  $p$ -values and critical values using Microsoft Excel. The definitions for these functions are not consistent, sometimes corresponding to two-tailed tests, sometimes giving upper tails, and sometimes lower tails. Read the definitions carefully. The algorithms used to compute probability functions in Excel are of dubious quality, but for the purpose of determining  $p$ -values or critical values, they should be adequate; see Knüsel (2005) for more discussion.

Function	What it does
<code>normsinv(p)</code>	Returns a value $q$ such that the area to the left of $q$ for a standard normal random variable is $p$ .
<code>normsdist(q)</code>	The area to the left of $q$ . For example, <code>normsdist(1.96)</code> equals 0.975 to three decimals.
<code>tinv(p,df)</code>	Returns a value $q$ such that the area to the left of $- q $ and the area to the right of $+ q $ for a $t(df)$ distribution equals $p$ . This gives the critical value for a two-tailed test.
<code>tdist(q,df,tails)</code>	Returns $p$ , the area to the left of $q$ for a $t(df)$ distribution if $tails = 1$ , and returns the sum of the areas to the left of $- q $ and to the right of $+ q $ if $tails = 2$ , corresponding to a two-tailed test.
<code>finv(p,df1,df2)</code>	Returns a value $q$ such that the area to the <i>right</i> of $q$ on a $F(df_1, df_2)$ distribution is $p$ . For example, <code>finv(.05,3,20)</code> returns the 95% point of the $F(3,20)$ distribution.
<code>fdist(q,df1,df2)</code>	Returns $p$ , the area to the <i>right</i> of $q$ on a $F(df_1, df_2)$ distribution.
<code>chiinv(p,df)</code>	Returns a value $q$ such that the area to the <i>right</i> of $q$ on a $\chi^2(df)$ distribution is $p$ .
<code>chidist(q,df)</code>	Returns $p$ , the area to the <i>right</i> of $q$ on a $\chi^2(df)$ distribution.

File  $\rightarrow$  New  $\rightarrow$  Script or select New Script from the JMP starter. This will open a new window. In this window, type a command similar to one of the following six **including the required trailing “;”**:

```
Show(ChiSquare Distribution(11.07, 5));
Show(ChiSquare Quantile(.95, 5, 0));
Show(F Distribution(3.32, 2, 3));
Show(F Quantile(0.95, 2, 10, 0));
Show(t Distribution(.9, 5));
Show(t Quantile(.95, 2.5));
```

The first of these will return the area to the left of 11.07 on the  $\chi^2(5)$  distribution, while the second returns the quantile of the  $\chi^2(5)$  distribution such that the area to the left of this value is 0.95. The next two commands serve a

similar function for the  $F$  distribution, the first for the  $F(2, 3)$  and the second as shown for the  $F(2, 10)$ , and the last two for the  $t$  distribution.

If you typed all six of these commands into a script window and then selected Edit → Run script, the following output would appear in the Log window (you may need to select View → Log to see this window):

```
ChiSquare Distribution(11.07, 5):0.949990381377595
ChiSquare Quantile(0.95, 5, 0):11.0704976935164
F Distribution(3.32, 2, 3):0.826393360224315
F Quantile(0.95, 2, 10, 0):4.1028210151304
t Distribution(0.9, 5):0.795314399827688
t Quantile(0.95, 2.5):2.55821861413594
```

The first of these shows that the area to the left of 11.07 for the  $\chi^2(5)$  distribution is .95. The second shows that the .95 quantile of  $\chi^2(5)$  is 11.07. The area to the left of 3.32 on the  $F(2, 3)$  distribution is .83; if this were a test statistic, where the usual  $p$ -value is given by an upper tail, this would correspond to a  $p$ -value of  $1 - 0.84 = 0.17$ . The critical value for a test at level .05 with the  $F(2, 10)$  distribution is 4.10.

For more information on these probability functions, and other functions that are available in JMP, go to the Index tab of the on-line help and type “probability functions.”

## 0.5 ABBREVIATIONS TO REMEMBER

ALR refers to the textbook, Weisberg (2005). VR refers to Venables and Ripley (2002), our primary reference for R and S-Plus. JMP-START refers to Sall, Creighton and Lehman (2005), the primary reference for JMP. Information typed by the user looks like **this**. References to menu items look like File or Transform → Recode. The name of a BUTTON to push in a dialog uses this font.

## 0.6 COPYRIGHT AND PRINTING THIS PRIMER

Copyright © 2005, by Sanford Weisberg. Permission is granted to download and print this primer. Bookstores, educational institutions, and instructors are granted permission to download and print this document for student use. Printed versions of this primer may be sold to students for cost plus a reasonable profit. The website reference for this primer is [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr). Newer versions may be available from time to time.

# 1

---

## *Scatterplots and Regression*

### 1.1 SCATTERPLOTS

A principal tool in regression analysis is the two-dimensional scatterplot. All statistical packages can draw these plots. We concentrate mostly on the basics of drawing the plot. Most programs have options for modifying the appearance of the plot. For these, you should consult documentation for the program you are using.

**JMP** JMP is organized around *analysis platforms*. Once you select a platform, JMP will produce a set of graphs and computations that are appropriate for that platform. For example, the Fit Y by X analysis platform is generally appropriate for simple linear regression, and if you select this platform you will get a few graphs and some standard calculations like estimates. Options within the platform allow you draw a few additional graphs and perform auxiliary calculations, as will be shown as we progress through this primer.

ALR[F1.1] can be drawn by reading the data file `heights.txt` into JMP as described in Section 0.4.1, then selecting the menu items `Analyze` → `Fit Y by X`. Assign *Dheight* to the response Y and *Mheight* to the factor X by either selecting the variable and clicking the correct Y or X button or by dragging the variable name to the correct box with your mouse. After pressing OK, JMP automatically chooses a scatterplot for this fit because both *Dheight* and *Mheight* are continuous variables.

As discussed in ALR, the scale of the plot for this data is particularly important and we would like to draw it so that the length of each axis is

the same. The plot initially produced in JMP will most likely not meet these requirements but it can be modified easily. The horizontal axis can be modified by either right clicking anywhere on the body of the plot and choosing the option `Size/Scale` → `X Axis` or by right clicking on horizontal axis (with the "grabber" hand selected at `Tools` → `Grabber`) and choosing `Axis Setting`. Both methods produce a dialog window where we will enter 55 as the minimum value and 75 as the maximum value. Although there is no data analytic reason to do so, we can reproduce ALR[F1.1] exactly by unchecking the `Tickmark` box besides `Minor`, then changing the increment value to 5 so that a major tick mark appears every 5 inches. The same procedure is followed to change the vertical axis. The title and any variable names can be changed by left clicking on them and entering the new name. The pixel size of the plot can be changed by right clicking on the body of the plot and choosing `Size/Scale` → `Frame Size`. Entering equal horizontal and vertical pixels sizes will produce a square plot, which is ideal for these data. Finally, the point size can be adjusted by right clicking on the body of the plot and choosing `Marker Size`, then selecting any desired size. All of these modification can be used with any plot produced in JMP.

The graphs in JMP are *linked* to each other and to the data grid for the current data set. This means that we can modify a graph by interacting with the data grid. To obtain an *approximation to* ALR[F1.2], for example, select some of the points you want to remove from the graph by dragging the mouse over them with the left button down, and then select `Rows` → `Hide/Unhide`. You can repeat this to delete other groups of points. When you want to return the points to the graph, select `Rows` → `Row selection` → `Select all rows`, and then `Rows` → `Hide/Unhide`.

ALR[F.12] can be reproduced exactly, but the procedure is too tedious to be used very often. The *row state* of any case, or row in the data grid, can be modified by selecting, or highlighting, its row. ALR[F1.2] can be drawn in JMP by hiding the points which we do *not* want to see plotted, which are the points that satisfy the conditions  $Mheight \leq 57.5$  or  $58.5 < Mheight \leq 62.5$  or  $63.5 < Mheight \leq 67.5$  or  $Mheight > 68.5$ .

The method we suggest for selecting these points uses the `Rows` menu, but you can also use the popup menu found by clicking on the red triangle to the left of the `Rows` section of the data table (located to the left of the data). First make sure no data points are currently selected by choosing `Rows` → `Clear Row States`. Next choose `Rows` → `Row Selection` → `Select Where`. The window that appears will be used to select cases to be hidden. First, from the `Currently Selected Rows` section you need to choose the option `Extend Current Selection`. This option will add any rows which satisfy the conditions you enter to those rows previously selected. The conditions on *Mheight* listed above must be entered one at a time. To select rows with  $Mheight \leq 57.5$  first choose *Mheight* from the variable list, then select `is less than or equal to` from the pop-down condition list, enter 57.5 in the box following the condition and, finally, select `ADD CONDITION`. By pressing `OK` all the rows that have

*Mheight* values less than or equal to 57.5 will be highlighted. To this selection we next want to add the rows where  $58.5 < Mheight \leq 62.5$ . This can be done by again following the menus Rows  $\rightarrow$  Row Selection  $\rightarrow$  Select Where to the selected rows popup window. Choose **Extend Current Selection** and select *Mheight*. Then add *both* conditions **is greater than 58.5 and is less than or equal to 62.5** to the conditions box. Note that the Select Rows section to the right of this box will become active when more than one condition is listed. We will use the default option **if all conditions are met** because the other option, **if any condition is met**, will cause all rows to be selected (since all *Mheights* are greater than 58.5 *or* less than 62.5). After pressing OK, the rows satisfying this second condition will be added to those already selected. This procedure must be repeated twice more, first to select the rows where  $63.5 < Mheight \leq 67.5$  and second to select rows where  $Mheight > 68.5$ . Remember to always select the option **Extend Current Selection** so that the previous selections are not erased.

When all points satisfying the four conditions are selected, select Rows  $\rightarrow$  Hide/Unhide. A little mask icon will then appear next to the selected points denoting that this points will be hidden from view in any graphic. You will notice that if you still have ALR[F1.1] active, the points which we want to hide should disappear. If you did not have any scatterplot of *Mheight* versus *Dheight* already available, simply follow the Analyze  $\rightarrow$  Fit Y by X link to create such a plot. As long as the correct points are hidden (look for the masks) then the resulting plot will be ALR[F1.2] and the only step remaining would be to modify the axes. One final hint: when using the popup Rows menu from the data table, I found it easy accidentally to delete the selected points since it was the first item in the popup Rows menu. This is easily fixed by selecting Edit  $\rightarrow$  Undo Delete Rows.

The next figure, ALR[F1.3], uses the `forbes.txt` data file. Once this file is read into JMP, select Analyze  $\rightarrow$  Fit Y by X and enter *Pressure* as the response and *Temp* as the factor. To obtain ALR[F1.3A] click on the red triangle next to the graph title Bivariate Fit of Pressure by Temp and select **Fit Line** from the popup menu. This action will automatically plot the regression line to the original scatterplot and give you results from the OLS fit. The residual plot is drawn by selecting **Plot Residuals** from the Linear Fit popup menu directly beneath the scatterplot. The residual plot will appear below the linear fit tables. You will probably need to right-click on the plot and select **Size/Scale  $\rightarrow$  Frame size** to resize the plot. A nice feature of JMP graphs is that you can simply click on a point to see its observation number, and the corresponding case is highlighted in the data grid.

To obtain ALR[F1.4] we first must take the base 10 log of the *Pressure* variable. This transformed variable is *Lpres* in the Forbes data file. If this variable were not provided, we could transform *Pressure* as follows. Right click on an empty column, then select **New Column**. Enter the name of this new variable, say *logPressure*, and the press **NEW PROPERTY** and select **Formula** from the popup menu to obtain the formula editor shown in Figure 0.2,

page 10. Choose *Pressure* from the variable list so that it appears editing area. Pick **Transcendental** from the function browser and select **Log10** from the popup menu. Click OK to complete the formula and again to complete formation of the new variable. Then simply follow the steps above for creating ALR[F1.3] but use *logPressure* as the response variable.

The data file `wblake.txt` is used to draw ALR[F1.5] by plotting *Age* versus *Length* and then adding the regression line and the line which joins the mean *Length* for each *Age*. To draw this in JMP we read in the data as usual and check that both variables are of type continuous. Then select **Analyze** → **Fit Y by X** and enter *Length* as the response and *Age* as the factor. Add the OLS line by selecting **Fit Line** from the popup menu then, from the same menu, choose **Fit Each Value**. This command will treat the factor as a discrete variable and compute a one-way analysis of variance and plot the fitted values. This corresponds to fitting the mean of *Length* for each unique value of *Age*, then connecting the points with a line on the scatterplot. If you wish to make this line dashed, select **Line Style** from the popup menu next this fit and choose the dashed line from the style options.

ALR[F1.7] is drawn by using **Fit Y by X** to plot *Gain* as the response and *A* as the factor using variables from the data file `Turkey.txt`. To change the color and type of point plotted for the three different values of *S*, right click in the body of the plot and choose **Row Legend** from the popup menu. In the dialog box select *S* as the column whose value is used to set the marker, then check the box **Set Marker by Value** to change the plotting character. After you press OK the graph will be plotted with a different character type and color for each value of *S* and a legend will appear to the right of the plot.

## 1.2 MEAN FUNCTIONS

**JMP** To draw ALR[F1.8] in JMP we follow the steps used to obtain ALR[F1.1] and add the regression line with the usual **Fit Line** command. The new line in the plot ALR[F1.8] has slope of 1 and intercept of 0. This line can be added by choosing **Fit Special** from the popup plot menu then checking both boxes for **Constrain Intercept to:** and **Constrain Slope to:**. Since the default value for the intercept is 0 and for the slope is 1, the dialog should look like Figure 1.1 after this step. After pressing OK, the line that is added to the JMP plot will correspond to the dashed line in ALR[F1.8].

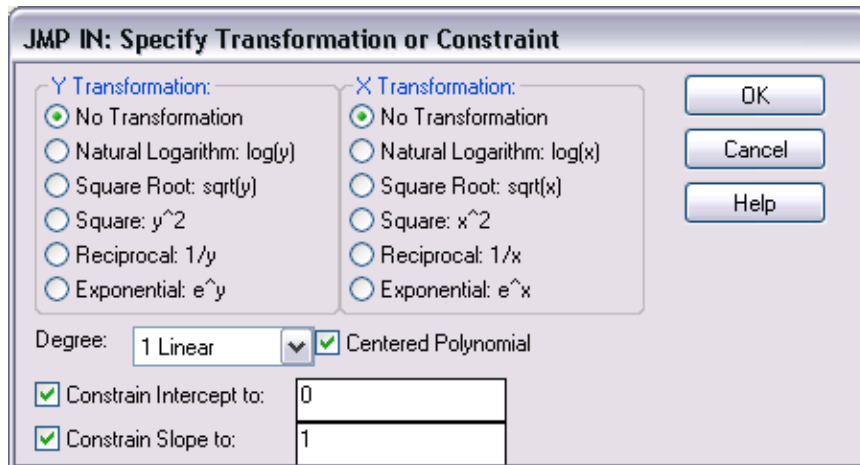


Fig. 1.1 JMP Fit Special dialog for adding a line with an intercept of zero and slope of one.

### 1.3 VARIANCE FUNCTIONS

#### 1.4 SUMMARY GRAPH

**JMP** JMP-START[10,P.275] contains a special topics section on why graphics are important and tells you step-by-step how to reproduce ALR[F1.9].

#### 1.5 TOOLS FOR LOOKING AT SCATTERPLOTS

**JMP** JMP does not contain any methods for adding a loess smoother to a scatterplot, but the particular smoother that is used is unimportant for the methods discussed in ALR.. From the Fit Y by X platform we can add a cubic spline smoother to the scatterplot by choosing Fit Spline from the plot popup menu. Choosing  $\lambda = 1000$  results in a smoother similar to the loess curve in ALR[F1.10].

#### 1.6 SCATTERPLOT MATRICES

**JMP** To obtain ALR[F1.11] in JMP we first must transform the variables in the data file `fue12001.txt`. This is accomplished by right clicking on an empty column and selecting New Column from the popup menu. Each new variable can be made by entering its new name into the Column Name box, then selecting Formula from the popup NEW PROPERTY button. For the

variables *Dlic*, *Income*, and *Fuel* you just enter the transformation formula into the editing area using the keypad to indicate the arithmetic. To obtain *logMiles* select Log from the **Transcendental** list of functions and choose *Miles* from the table columns list. The Log function will calculate the natural log but you can change it to base 2 as follows: in the editing area highlight *Miles* in red by clicking on it, then type a comma. The base of the log should now appear and can be changed to 2 by clicking on **Table Columns** and selecting **Constants**. Choose 2 from this list and the function is complete.

The scatterplot matrix is drawn by selecting **Analyze** → **Multivariate Methods** → **Multivariate** and entering *Tax*, *Dlic*, *Income*, *logMiles* and *Fuel* into the Y box. The scatterplot matrix which appears will have 95% bivariate normal density ellipses on each plot. To remove these ellipses select **Density Ellipses** from the popup plot menu. The order of the variables in this matrix can be changed by clicking a variable name and dragging it to the desired location.

## Problems

**1.1.** Boxplots would be useful in a problem like this because they display level (median) and variability (distance between the quartiles) simultaneously.

**JMP** To obtain summary statistics and boxplots of *Length* for each *Age* select **Analyze** → **Distribution** from the menu. Choose *Length* as the Y variable and *Age* as the "by" variable and press OK.

You can obtain a graph of the standard deviation for each *Age* by selecting **Graph** → **Variability/Gage Chart** and entering *Length* as the Y variable and *Age* as the X variable and pressing OK. The second plot is the standard deviation plot and you can edit the first plot to obtain boxplots for each *Age* level.

**1.2.**

**JMP** You can resize any scatterplot in JMP by moving the cursor the edge of the plot (not the edge of the graphic border) until it changes to a double arrow, then click and drag the edge to expand.

**1.3.**

**JMP** Details on transforming to  $\log_2$  are given in Section 1.6 when explaining ALR[F1.11].

# 2

---

## *Simple Linear Regression*

### 2.1 ORDINARY LEAST SQUARES ESTIMATION

All the computations for simple regression depend on only a few summary statistics; the formulas are given in the text, and in this section we show how to do the computations step-by-step. All computer packages will do these computations automatically, as we show in Section 2.6.

### 2.2 LEAST SQUARES CRITERION

**JMP** The summary statistics used to compute the least squares estimators for the Forbes data are not straightforward to obtain in JMP. To obtain the means of *Temp* and *Lpres* select **Analyze** → **Distribution** and place both variables in the Y box and press OK. The resulting output will, by default, give you histograms, quantiles, and moments for both variables and included in the moments will be the means of each and the number of cases in the data set.

The quantities  $SXX$ ,  $SYY$ , and  $SXY$  do not appear to be directly available from JMP, but the covariance matrix is available by following **Analyze** → **Multivariate Methods** → **Multivariate** and entering *Temp* and *Lpres* in the Y box. Only the correlation matrix and scatterplot matrix appear in the output, so to get the covariance matrix click the red triangle next to **Multivariate** and select **Covariance Matrix** from the popup menu. Then use your calculator and multiply the covariance matrix terms by  $(n - 1) = 17 - 1$  to

obtain the sums of squares needed. Finally, using the formulas in ALR[2.2] calculate the least squares estimates.

### 2.3 ESTIMATING $\sigma^2$

### 2.4 PROPERTIES OF LEAST SQUARES ESTIMATES

### 2.5 ESTIMATED VARIANCES

The estimated variances of coefficient estimates are computed using the summary statistics we have already obtained. These will also be computed automatically linear regression fitting methods, as shown in the next section.

### 2.6 COMPARING MODELS: THE ANALYSIS OF VARIANCE

Computing the analysis of variance and  $F$  test by hand requires only the value of  $RSS$  and of  $SS_{reg} = SYY - RSS$ . We can then follow the outline given in ALR[2.6].

**JMP** In JMP, graphics and analysis go hand-in-hand, so if you follow the steps outlined in Section 1.1 to draw scatterplots, then you are one step away from obtaining the simple linear regression fit. To review, after the data file `forbes.txt` is read into JMP select the menu items `Analyze` → `Fit Y by X`. Enter `Lpres` as the response variable and `Temp` as the factor variable and press OK. This action will only produce the scatterplot seen in ALR[F1.3] but if you click on the red triangle next to “Bivariate Fit of Lpres by Temp” and select `Fit Line` from the popup menu, the regression line will be added to the plot and the linear fit summarize. This is the fit for the standard mean function  $E(Lpres|Temp) = \beta_0 + \beta_1 Temp$ . If you would like to fit regression through the origin, select the menu option `Fit Special` (instead of `Fit Line`) and in the dialog check the option `Constrain Intercept to:` and enter 0 as the intercept. Note that this mean model is not relevant for the Forbes data.

The output for the linear regression fit is displayed in tables below the scatterplot. The estimated mean function is given under the `Linear Fit` header and for the Forbes data is

$$Lpres = -42.13778 + 0.8954937 Temp$$

The analysis of variance table is also given and is summarized as follows:

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	1	425.63910	425.639	2962.785
Error	15	2.15493	0.144	Prob > F
C. Total	16	427.79402		<.0001

The first two rows in this table match the values given in ALR[T2.4], except that JMP places the  $p$ -value (Prob > F) under the F statistic. Actually JMP treats the “F Ratio” and “Prob > F” as two separate columns and, as described shortly, they can also be modified separately. The additional line “C. Total” is the same as the “Total” line in ALR[T2.3].

The summary of the coefficient estimates is given in a Parameter Estimates table:

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	-42.13778	3.340199	-12.62	<.0001
Temp	0.8954937	0.016452	54.43	<.0001

The coefficient estimates for the linear fit are given along with their standard error,  $t$ -statistic and  $p$ -value for testing whether coefficients are equal to zero.

JMP refers to all the tables and graphs produced by a platform as a *report* and calls individual tables *report tables*. These tables are active, meaning the numbers and columns in them can be modified after they are displayed as output. For instance, doubling clicking on a column will let you change the numeric format and field width of the column. This allows us to see the actual size of the  $p$ -values above by double clicking and then selecting **Best** from the format menu. Another feature of these tables is that certain columns and hence output may be hidden from view. These columns can be added by right clicking on a table and selecting **Columns** from the menu. A list of all columns available for this table will appear and choosing an unchecked column will add it to the table while choosing a checked column will remove it from the table. The format of the table can be changed by selecting **Sort by Columns** from the right-click menu.

## 2.7 THE COEFFICIENT OF DETERMINATION, $R^2$

**JMP** The value of  $R^2$  can be calculated from the values obtained using the sums of squares from Section 2.2 or it can be read from the Summary of Fit table given by the linear regression fit. For the Forbes data this table is:

RSquare	0.994963
RSquare Adj	0.994627
Root Mean Square Error	0.379027
Mean of Response	139.6053
Observations (or Sum Wgts)	17

For this table, “RSquare” is  $R^2$  and “Root Mean Square Error” is  $\hat{\sigma}$ , the standard error of regression. The value given for “RSquare Adj” is not discussed in ALR and so can be ignored.

## 2.8 CONFIDENCE INTERVALS AND TESTS

Confidence intervals and tests can be computed using the formulas in ALR[2.8], in much the same way as the previous computations were done.

**JMP** The JMP report table Parameter Estimates provides 95% confidence intervals for each coefficient, but they are hidden from view. You can see them by right clicking on the table and selecting **Columns** → **Lower 95%** and then **Columns** → **Upper 95%**. The intervals for both parameter estimates are then added to the table. If you would like to calculate intervals at other levels one option is simply to use the parameter estimates and standard error presented in the table and calculate the values by hand using the appropriate  $t$  multiplier. Hypothesis tests for coefficients can be constructed in the same manner.

A second option is to use the *scripting* capabilities of JMP. Anything that can be done with the graphical interface has an analogous script that can accomplish the same task. We suggest using the menu commands whenever possible, and to use scripting only as a last resort. What we will present here are simple commands which can serve as an alternative to the “hand” calculations above, but we do not intend to give a full description of JMP scripting syntax or structure. You can investigate scripting further with the references available with your software or by visiting the website [www.jmpdiscovery.com](http://www.jmpdiscovery.com).

Scripting works by entering commands in a script window, running the script, then viewing the output in a log window. The log window will either appear by default when you start JMP or you may have to open it by selecting **View** → **Log**. To open a script window select **File** → **New** → **Script** or press the **NEW SCRIPT** button in the JMP starter window.

We will start with the simple task of determining the  $t$ -value used in the calculation of a 90% confidence interval. The degrees of freedom for the regression fit of the Forbes data is  $17 - 2 = 15$  so the appropriate script command is

```
t Quantile(1-.1/2,15);
```

After this is entered in the script window select **Edit** → **Run Script**. This will produce the  $t$ -value of 1.75305 in the log window. To calculate the  $p$ -value for a two-sided hypothesis enter and run the following script:

```
2*(1- t Distribution(2.137,15));
```

where 2.137 is the value of the  $t$ -statistic (see ALR[2.8.1]). The value for a one-sided test can be found by deleting “2\*” from the command.

We have also provided a script which will automatically calculate confidence intervals for parameters at the level you specify. In order to use this script you must save the Parameter Estimates table by right clicking and selecting **Make Into Matrix**. Store the value as a global variable (this should be the default option) and enter “Parameters” into the name box and click OK. Give the matrix this name and that the first and second columns of the Parameter

table are the estimates and standard errors, respectively, otherwise the script will not work. After this is done the only values you need to modify in the following script are the desired level of the interval (e.g. 0.95, 0.90) and the degrees of freedom used in the  $t$ -value.

```
level= .90;           //change the CI level as needed
df= 15;              //enter DF Error from ANOVA table

a = (1-level)/2;
low=Parameters[0,1]+(t Quantile(a,df))*Parameters[0,2];
high=Parameters[0,1]+(t Quantile(1-a,df))*Parameters[0,2];
ci=low || high;
ci2= New Table ("New CI for Parameter Estimates");
ci2<<New Column("Lower " || char(level*100) || "%",Values(low));
ci2<<New Column("Upper " || char(level*100) || "%",Values(high));
```

The confidence intervals for this script will be presented in a new table, with each row corresponding to a row in the Parameter Estimates table from the regression analysis.

### Prediction and fitted values

**JMP** You can easily obtain the fitted values (or predictions) for each observation by selecting **Save Predicteds** from the Linear Fit popup menu. This will save the values as a new *formula* variable in the data table. Thus you can enter a new predictor value, say  $Temp = 220$ , and the predicted value will appear in the column of predicted  $Lpres$ .

Using the Fit Y by X platform, confidence intervals for predicted values and the mean function can only be obtained by estimating standard errors using the values obtained in Sections 2.2 and 2.3 and the equations in ALR[2.8.3] and ALR[2.8.4]. One hint: instead of calculating  $SXX$  as in Section 2.2, we can rearrange ALR[E2.8] to read

$$SXX = (SYY - RSS) / \hat{\beta}_1^2$$

and use the values  $RSS$ ,  $SYY$ , and  $\hat{\beta}_1$  from the ANOVA and coefficients tables.

Both the individual and mean confidence intervals can be added to the scatterplot by using the Linear Fit popup menu and selecting **Confid Curves Indiv** for the prediction intervals and **Confid Curves Fit** for the mean function intervals. The default level for both curves is 95% but this can be changed by selecting **Set Alpha Level** from the same menu.

JMP does provide values of these confidence intervals when using the platform **Fit Model** to fit a linear regression model. It also provides a way to obtain the predicted values and standard errors for new predictor values. This platform will be introduced in Chapter 3 to fit multiple linear regression but its implementation is basically the same as the Fit Y by X platform.

## 2.9 THE RESIDUALS

**JMP** Details on how to draw residual plots were first given in Section 1.1 for the Forbes residual plot ALR[F1.3B]. Residual plots are drawn by selecting Plot Residuals from the Linear Fit popup menu.

The final note for this section is about deleting a case from the analysis, for example, case 12 was deleted from the Forbes' data. This can be done by highlighting case 12 in the data table and under the Rows popup menu selecting Exclude/Unexclude. Any analysis run will not include the selected case.

### Problems

#### 2.2.

**JMP** We suggest parts 2.2.5 and 2.2.6 be done with the Fit Model platform. When two data sets are open in JMP, the data editor which is the active window will be the data used in any analysis or graphing commands. Select the data set to fit and choose Analyze → Fit Model. Enter the response in the Y field then select the predictor and click ADD. From the Emphasis list choose the option Minimal Report and click RUN MODEL. This will give you the OLS fit with similar output as the Fit Y by X analysis.

For part 2.2.5, save the predicted values and their standard errors by selecting Save Columns → Predicted Values and Save Columns → Std Error of Individual from the popup menu next to the Response section of the output. You can then compute the  $z$ -score by adding a column whose value will equal the  $z$ -score formula in the text.

Part 2.2.6 can be done by fitting Hooker's data using Fit Model, then saving the predicted value *formula* and standard error *formula* for the fitted values. To do this, select Save Columns → Prediction Formula and Save Columns → StdErr Pred Formula. Next select the column  $u_1$  from the Forbes data table, then right click on the highlighted part and copy the values. At the bottom of the  $u_1$  column of the Hooker data table, right click and add seventeen rows. Select these rows by clicking on the first new row and dragging down to the last row, then right click and select Paste. This will add the Forbes predictor values as well as compute their predicted values and standard errors of fit according to the previously saved formulas from Hooker's data. Add a column to calculate the standard error for prediction using the formula  $sepred = (\hat{\sigma}^2 + (sefit)^2)^{1/2}$  and then compute the  $z$ -scores.

#### 2.7.

**JMP** To fit a regression through the origin, instead of selecting Fit Line from the scatterplot menu, select Fit Special and check the box for the option Constrain Intercept to: and enter 0 as the intercept value.

**2.10.**

**JMP** This data file contains missing values so import it using the file type **Text Import Preview**.

To only analyze the cases which satisfy  $HamiltonRank \leq 50$  you need to select all other cases then exclude them from analysis and hide them in plots. To do so select **Rows** → **Row Selection** → **Select Where**, then choose *HamiltonRank* from the list of variables, **is greater than** from the list of conditions and enter 50 as the condition value. Add the condition and press OK. To exclude these cases from the regression fit select **Rows** → **Exclude/Unexclude** and to hide these cases on a scatterplot select **Rows** → **Hide/Unhide**.



# 3

---

## *Multiple Regression*

### 3.1 ADDING A TERM TO A SIMPLE LINEAR REGRESSION MODEL

**JMP** JMP can produce added-variable plots *after* you have fit a multiple linear regression model and therefore we will defer discussing the plots until after we have discussed fitting the model.

### 3.2 THE MULTIPLE LINEAR REGRESSION MODEL

### 3.3 TERMS AND PREDICTORS

**JMP** The summary statistics in ALR[3.3] for the data file `fuel2001.txt` can be easily obtained after transforming the predictors into the terms used for the multiple regression model. These transformations were done in Section 1.6 to draw the scatterplot matrix of terms, so please refer back to that section for details on obtaining the variables *Dlic*, *Income*, *logMiles*, and *Fuel*.

The summary statistics in ALR[T3.1] can be calculated by selecting the menu item **Analyze** → **Distribution** and placing all five variables in the Y box and pressing OK. The resulting output will, by default, give you histograms, quantiles, and moments (e.g. mean and standard deviation) for all variables.

The sample correlation matrix in ALR[T3.2] can be obtained by selecting **Analyze** → **Multivariate Methods** → **Multivariate** and entering all five variables in the Y box. This will create the correlation matrix and the scatterplot matrix for the selected terms.

### 3.4 ORDINARY LEAST SQUARES

**JMP** To obtain a sample covariance matrix, follow the steps in Section 3.3 to get the correlation matrix, then click the red triangle next to Multivariate and select **Covariance Matrix** from the popup menu.

To fit a multiple regression model in JMP we must use the **Fit Model** platform as opposed to the **Fit Y by X** platform used for simple regression. After transforming the fuel variables as needed, select **Analyze** → **Fit Model** from the JMP menus. Enter *Tax* into the Y variable box and enter all predictor terms into the Model Effects box using the **ADD** button. To obtain basic OLS result tables such as summary of fit, analysis of variance and parameter estimates, select **Minimal Report** from the Emphasis list of output options. Press **RUN MODEL** to fit the model and obtain these tables.

Just as with the simple linear regression model, the first three report tables given for this fit are the Summary of Fit, Analysis of Variance, and Parameter Estimates and details about these tables are given in Section 2.6 and 2.7. In addition, an Effect Tests table is also presented which contains results for testing the model excluding each predictor versus the full model. More on this table is given in the following section.

Section 2.8 gives the details for obtaining coefficient confidence intervals and tests. You can use the same script given at the end of that section to obtain coefficient confidence intervals at any level desired.

### 3.5 THE ANALYSIS OF VARIANCE

**JMP** The analysis of variance report table from the multiple regression fit corresponds to ALR[T3.4]. To obtain a sequential ANOVA table corresponding to ALR[T3.5] select **Estimates** → **Sequential Tests** from the popup menu next to the header Response Fuel. This table is fit in the order the terms were added in the model dialog, therefore to obtain a different sequential ordering, return to the **Fit Model** window (which should still be open) and arrange the predictors in the correct order and run the model.

Tests of partial models are given in the Effects Tests report table, which is given after the model is fit with the minimal summary emphasis. These partial tests correspond to hypotheses like the one given in ALR[E3.22]. For the Fuel data this table looks like:

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
<i>Tax</i>	1	1	18263.939	4.3373	0.0429
<i>Dlic</i>	1	1	56770.383	13.4819	0.0006
<i>Income</i>	1	1	32939.700	7.8225	0.0075
<i>logMiles</i>	1	1	34573.068	8.2104	0.0063

For the test done in ALR[3.5.2], the sum of squares for *Tax* above is 18263.939, the difference between the sum of squares for the model excluding *Tax* and

the sum of squares for the full model. The  $F$ -ratio and significance level are the same as the values given in ALR[3.5.2].

### 3.6 PREDICTIONS AND FITTED VALUES

**JMP** As mentioned at the end of Section 2.8, the Fit Model platform allows for the formulas of predicted values and the standard errors of fitted values to be saved as column variables. To obtain the fitted values and standard errors for *new* predictor values, we can add these new values to the data table and the saved formulas will produce the estimated values. Note that when regression terms themselves are formulas we cannot add a new predictor value at the end of its column, but must add the untransformed variable values which will result in the desired value of the regression term.

To obtain 95% confidence intervals for the predicted values or for the mean function, select **Save Columns** from the popup response menu then **Indiv Confidence Interval** or **Mean Confidence Interval**. The intervals are added to the data table as new columns. From this same menu, to get the predicted values you can either choose **Predicted Values** or **Prediction Formula**. The latter option allows you to obtain a new predicted value from predictors values added to the data table after the model was fit.

Before we discuss how to obtain standard errors in JMP we must point out that the error which ALR refers to as “standard error of prediction” is “standard error of individual” in JMP and the ALR “standard error of fitted value” is “standard error of prediction” in JMP. We will use the ALR name when discussing these values.

Standard error for predicted values can be added to the data table by selecting **Save Columns** → **Std Error of Individual** from the popup response menu. From the same menu option, we can save the standard error for the fitted values by either its value, **Std Error of Predicted**, or its formula, **StdErr Pred Formula**. As with the predicted values, the formula option will calculate the standard error of the fitted value for any new predictor values added to the data table. This formula option is not available for the standard error of individual predicted values, but we can add a column containing the formula for this error after the *formula* for standard error of the fitted values has been saved. You can then add a new column for prediction errors by using the formula  $sepred = (\hat{\sigma}^2 + (sefit)^2)^{1/2}$ . The value of  $\hat{\sigma}$  is given in the Summary of Fit table under “Root Mean Square Error” and the value for *sefit* will be the column variable *PredSE Fuel*, the standard error for the fitted values of the fuel data.

### Problems

**JMP** Several of these problems concern added-variable plots. JMP does not produce added-variable plots as described in ALR[3.1.2] but instead gives *leverage plots*. The difference between these plots is minimal, with the main difference being the scale of the horizontal axis. The actual shape of the point cloud will be the same in both added-variable and leverage plots. You can ignore the dashed-lines on the leverage plots.

Leverage plots can be obtained before or after a multiple regression model is run. If a model has been fit using the **Minimal Report** emphasis, then leverage plots for all predictors can be obtained by selecting **Row Diagnostics** → **Plot Effect Leverage** from the Response popup menu. It may at first look as if this command did not produce any output, but by clicking the blue triangle next to the “Effect Details” header you will open this section of the report and the leverage plots will appear.

An alternate way to obtain these plots is to select the emphasis option **Effect Leverage** and then run the model. This output option will give you leverage plots for each predictor, as well as the plot of the fitted versus the response values and the residual plot. The report tables given with this option are the same four as detailed above.

# 4

---

## *Drawing Conclusions*

The first three sections of this chapter do not introduce any new computational methods; everything you need is based on what has been covered in previous chapters. The last two sections, on missing data and on computationally intensive methods introduce new computing issues.

### **4.1 UNDERSTANDING PARAMETER ESTIMATES**

#### **4.1.1 Rate of change**

#### **4.1.2 Sign of estimates**

#### **4.1.3 Interpretation depends on other terms in the mean function**

#### **4.1.4 Rank deficient and over-parameterized models**

**JMP** An over-parameterized model is handled easily in JMP. When exact collinearity between predictors is determined to exist, JMP will display the linear formula and calculate parameter estimates for each term which cannot be written as a linear combination of the terms preceding it in the Fit Model effects list.

Consider the Berkeley Guidance Study example from ALR[4.1.3]. The variables  $DW9$  and  $DW18$  are linear combinations of the terms  $WT2$ ,  $WT9$ , and  $WT18$ . Using the Fit Model platform, when the model effects are entered in the order  $W2$ ,  $WT9$ ,  $WT18$ ,  $DW9$ , and  $DW18$ , the regression of *Soma* on

these terms results in the usual regression report tables as well as a table titled “Singularity Details” with these results:

$$\begin{aligned} WT2 &= WT9 - DW9 \\ WT9 &= WT18 - DW18 \end{aligned}$$

The program has discovered that  $DW9 = WT9 - WT2$  and  $DW18 = WT18 - WT9$ , and shows you these equalities, in a slightly different form. Since JMP is a computer program, it works with *near equalities*, so the equal signs really mean “are equal within machine accuracy.”

The Parameter Estimates table will add a column denoting which terms are “Biased” and “Zeroed”. A biased term has a linear dependency with other terms but its parameter is estimated. A zeroed term also has a linear dependency but, because it can be written as a linear combination of the terms preceding it in the Model Effect list, it is deleted from the model. For the Berkeley data model above the Parameter Estimates table is:

Term		Estimate	Std Error	t Ratio	Prob> t
Intercept		1.5921008	0.674247	2.36	0.0212
WT2	Biased	-0.115643	0.061695	-1.87	0.0653
WT9	Biased	0.0562477	0.020111	2.80	0.0068
WT18	Biased	0.0483385	0.010602	4.56	<.0001
DW9	Zeroed	0	0	.	.
DW18	Zeroed	0	0	.	.

If you had entered the terms into the mean function in a different order, you could get different terms marked “Biased” and “Zeroed.”

## 4.2 EXPERIMENTATION VERSUS OBSERVATION

## 4.3 SAMPLING FROM A NORMAL POPULATION

## 4.4 MORE ON $R^2$

## 4.5 MISSING DATA

The data files that are included with ALR use “NA” as a place holder for missing values. Some packages may not recognize this as a missing value indicator, and so you may need to change this character using an editor to the appropriate character for your program.

**JMP** A text file containing missing values denoted by “?” (or “NA”) can be imported into JMP as discussed in Section 0.4.1. The default method in JMP of importing data from a text file will handle missing value indicators incorrectly, and turn the corresponding column into a nominal or text variable. This is fixed by using the “Text Import Preview” option of the Open File

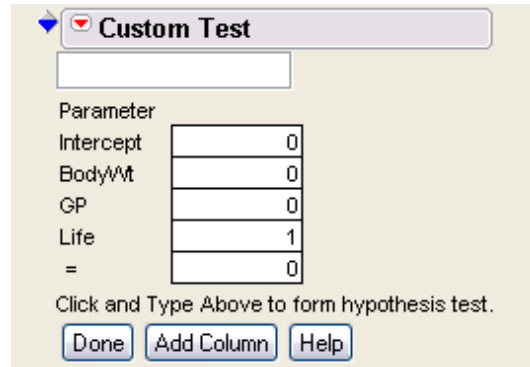


Fig. 4.1 JMP dialog used to compare two regression fits of the data `sleep1.txt`.

dialog. This will correctly convert any non-numeric value (e.g. “?” or “NA”) to the JMP missing value indicator shown in the data grid as a “•”.

You can determine the number of missing cases for any variable by using the Distribution platform first described in Section 2.2. After obtaining the standard output from this analysis, you can select the option **Display Options** → **More Moments** from the popup menu next to each variable’s name. One of the values added to the Moments table will be “N Missing”, the number of cases missing for this variable.

A regression model will not use any cases which have missing values in one or more of the variables in the model. If you would like to compare two models you may run into problems if you fit each model separately in JMP. Consider the data file `sleep1.txt`. The regression fit of *SWS* on *BodyWt*, *Life*, *GP* will be based on the 42 complete cases of this list of variables, while a separate fit of *SWS* on *BodyWt* and *GP* will be based on the 44 complete cases of this smaller list of variables. You can’t then compare these two fits with their analysis of variance values because they are based on different cases. The solution to this problem is to compare the two models with the Custom Test option available after the larger model is fit.

In the Response popup menu for the full model fit select **Estimates** → **Custom Tests**. This will add a Custom Test dialog at the bottom of the report. To compare the reduced and full models place a 1 next to any variable which you omit from the full model to obtain the reduced model. Add a column for each variable omitted and enter a 1 in the appropriate row. Figure 4.1 gives this dialog for the model comparison for the sleep data where the only variable omitted is *Life*. After pressing Done, the model comparison will be based on the 42 complete cases for the whole list of variables. The output for this test will be:

Value	0.0061500092
Std Error	0.031942626
t Ratio	0.1925329874

Prob> t	0.8483500168
SS	0.370382458
Sum of Squares	0.370382458
Numerator DF	1
F Ratio	0.0370689512
Prob > F	0.8483500168

The first table gives the parameter estimates for the omitted variable and the second table gives the sum of squares for the difference between the model full and reduced model along with the  $F$ -statistic and  $p$ -value.

#### 4.6 COMPUTATIONALLY INTENSIVE METHODS

**JMP** Computation of the bootstrap or other computationally intensive methods are possible with JMP, but require using the scripting language. We have not worked out how to do this with JMP, but would be glad to see the scripts developed by others for this purpose.

# 5

---

## *Weights, Lack of Fit, and More*

### 5.1 WEIGHTED LEAST SQUARES

**JMP** Both the Fit Y by X and Fit Model platforms can be used to obtain a WLS regression fit with JMP. If not already defined in your data set, you must first form a column of weights. In the physics data from ALR[5.1], the weights need to be defined as a transformation of the variable  $SD$ . Let  $w_i$  be the weights where  $w_i = 1/SD^2$ . Then using either regression platform, enter the Response  $y$ , Effect  $x$  and Weight  $w_i$ . The standard output given for this WLS will be the summary of fit, analysis of variance, and parameter estimate tables. The name of the variable used as weights will be given directly beneath the title of the fit. Also, the predictions intervals will use the correctly weighted standard error,  $sepred = (\hat{\sigma}^2/w_i + sefit(y | X = \mathbf{x}_i))^{1/2}$ .

Most of the output provided for WLS can be used as if a OLS model had been fit, but one expectation is the residuals. The residuals obtained with a Save Residuals command is the vector  $y - \hat{y}$  but, as we will see later, with WLS a more reasonable set of residuals is the vector  $\sqrt{w}(y - \hat{y})$ . These are called *Pearson residuals* in ALR and in some software such as R and S-Plus but they are not provided by any Save option available with Fit Y by X or Fit Model. To obtain Pearson residuals, we must save the residuals  $y - \hat{y}$  and then create a new column variable using the formula  $\sqrt{w}(y - \hat{y})$ .

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	146.06417	3.331985	43.84	<.0001
x	492.2217	20.40487	24.12	<.0001
(x-0.15376)^2	1597.5047	250.5869	6.38	0.0004

Parameter Estimates				
Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	183.83046	6.459063	28.46	<.0001
x	0.9709023	85.36876	0.01	0.9912
x^2	1597.5047	250.5869	6.38	0.0004

Fig. 5.1 Centered and uncentered parameter estimates in JMP for the WLS polynomial fit of the physics data from ALR[5.2].

### 5.1.1 Applications of weighted least squares

**JMP** Weighted least squares fitting (as in ALR[T5.3]) can be obtained with either the Fit Y by X platform or Fit Model platform. Using the Fit Y by X dialog, enter  $y$ ,  $x$ ,  $wt$  as the Response, Factor, and Weight, respectively, where  $wt$  is as defined in Section 5.1. From the popup menu next to Bivariate Fit, select Fit Polynomial  $\rightarrow$  2, quadratic to fit the *centered* polynomial mean function,

$$E(Y|X = x) = \eta_0 + \eta_1 x + \eta_2 (x - \bar{x})^2$$

To obtain the uncentered fit,

$$E(Y|X = x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

select Fit Special from the same pop-up menu, then uncheck **Centered Polynomial** and choose 2 **Quadratic** from the Degree choices.

The report tables produced by each command are similar to those obtained with simple linear regression. Both the centered and uncentered fits will produce the same Summary of Fit and Analysis of Variance tables, though the sequential ANOVA table in ALR[T5.3] is not available with this platform. The difference between fits can be seen in the Parameter Estimates table and the fitted mean functions. The estimated mean function for the centered analysis is

$$y = 146.06417 + 492.2217 x + 1597.5047 (x - 0.15376)^2$$

and for the uncentered analysis is

$$y = 183.83046 + 0.9709023 x + 1597.5047 x^2$$

Both Parameter Estimates tables are given in Figure 5.1.

The centered and uncentered WLS fits will plot the same line on the scatterplot of  $x$  and  $y$ . To obtain ALR[F5.1] use the Fit Line command from the pop-up Plot menu.

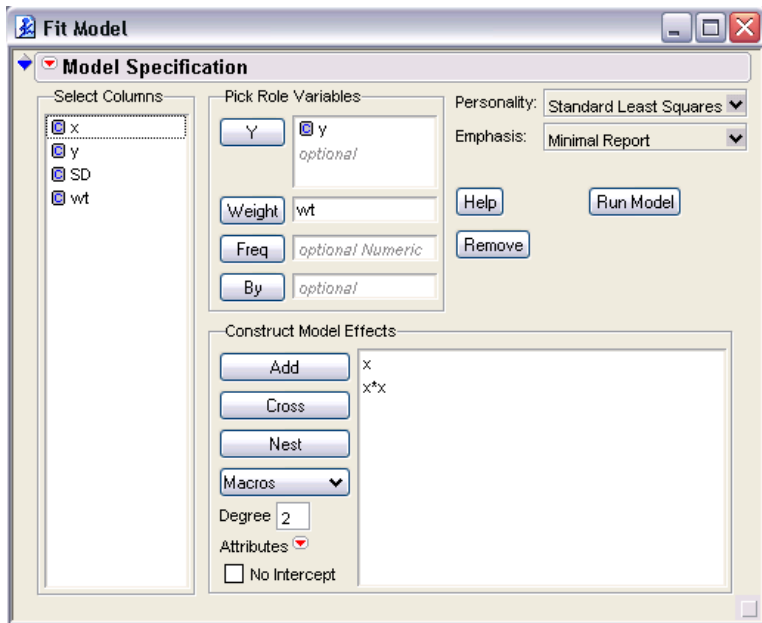


Fig. 5.2 The Fit Model dialog window for the WLS polynomial fit of the physics data from ALR[5.2].

This WLS polynomial fit can also be estimated using the Fit Model platform. In the dialog for this platform, move  $y$  and  $wt$  to the Y and Weight boxes and select  $x$  but *do not* move it to the Effects box. Instead, enter 2 in the Degree box and pick Polynomial to Degree from the drop-down list of MACROS. This will add the correct polynomial terms to the Effects box, as seen in Figure 5.2. By clicking RUN MODEL the centered polynomial WLS regression will be fit. To obtain results for the uncentered regression, uncheck Center Polynomials from the pop-up menu alongside the Model Specifications header of the Fit Model dialog. The three standard report tables will be given after this model is fit and the sequential analysis of variance table can be obtained by selecting Estimates  $\rightarrow$  Sequential Tests.

The scatterplot of  $y$  and  $x$ , called the Regression plot, is also given since only one variable was used in the mean function. The quadratic mean function is added to this plot but, unlike the Fit Y by X analysis, the regression line for the linear mean function cannot be added this plot. To draw ALR[F5.1] use the Fit Y by X platform as discussed above.

### 5.1.2 Additional comments

## 5.2 TESTING FOR LACK OF FIT, VARIANCE KNOWN

### 5.3 TESTING FOR LACK OF FIT, VARIANCE UNKNOWN

**JMP** The test for lack-of-fit is provided by default in JMP when there is exact replication of the predictors. Both Fit Y by X and Fit Model platforms will produce a lack-of-fit table which contains the lack-of-fit and pure error sum of squares and the  $F$ -test for lack-of-fit.

We can obtain ALR[T5.5] by opening a new data table and entering the data in ALR[T5.4]. A new data table can be produced by clicking on NEW DATA TABLE or selecting File  $\rightarrow$  New  $\rightarrow$  Data Table. After the regression of  $y$  on  $x$  is fit, the Lack of Fit table for this data will be

Source	DF	Sum of Squares	Mean Square	F Ratio
Lack Of Fit	2	1.8582481	0.929124	2.3638
Pure Error	6	2.3583917	0.393065	Prob > F
Total Error	8	4.2166398		0.1750
				Max RSq
				0.7316

### 5.4 GENERAL $F$ TESTING

**JMP** The general  $F$ -tests described in ALR[5.4] can be calculated with the Custom Test option of the Fit Model platform. This procedure was briefly discussed in Section 4.5.

Consider the fuel consumption data again. Using the Minimal Report emphasis, fit the regression of *Fuel* on *Dlic*, *Income*, *logMiles*, and *Tax* as done in Chapter 3. Next from the popup Response menu, select Estimates  $\rightarrow$  Custom Test. This will add a new element to the report table similar to Figure 4.1. As can be seen in this figure, all terms in the model are contained in a column named Parameters, at the bottom of which is an “=” sign. To the right of the terms is a column where we give the null hypothesis by adding values which specify a linear combination of terms equal to some constant, usually equal to zero. If we wish to test  $q$  terms equal to zero, we must add  $q$  columns and enter a “1” once in each column for each of the  $q$  terms.

For example, suppose we wish to test

$$\begin{aligned} \text{NH: } E(\text{Fuel} | X) &= \beta_0 + \beta_1 \log\text{Miles} + \beta_2 \text{Tax} \\ \text{AH: } E(\text{Fuel} | X) &= \beta_0 + \beta_1 \log\text{Miles} + \beta_2 \text{Tax} + \beta_3 \text{Dlic} + \beta_4 \text{Income} \end{aligned}$$

Since we are testing whether the parameters for *Dlic* and *Income* are equal to zero, we must add a second column and place a 1 in the first column for *Dlic* and a 1 in the second column for *Income*. Click DONE, and the test will be run and produce the following results:

Parameter		
Intercept	0	0
Dlic	1	0
Income	0	1
logMiles	0	0
Tax	0	0
=	0	0
Value	0.4718712134	-6.13533097
Std Error	0.128513421	2.1936335745
t Ratio	3.671766028	-2.796880501
Prob> t	0.0006255639	0.0075077902
SS	56770.382674	32939.700445
Sum of Squares	108492.27402	
Numerator DF	2	
F Ratio	12.882406334	
Prob > F	0.0000360964	

Since the first column contained a one for *Dlic*, the first values are the parameter estimate and *t*-test for *Dlic* which are the same as those given in the Parameter Estimates table. The row “SS” is the sequential sum of squares for entering *Dlic* after the three other terms. The second column contains these values for the term *Income*. The final table produced gives the sum of squares and degrees of freedom for the numerator of ALR[E5.16]. The *F*-statistic and *p*-value for this test are then given.

## 5.5 JOINT CONFIDENCE REGIONS

**JMP** Confidence regions for parameter estimates are not available in JMP.

### Problems

**5.3.** Like other resampling problems, this problem is not recommended with JMP, unless you are expert at the JMP scripting language.

The bootstrap used in this problem is different from the bootstrap discussed in ALR[4.6] because rather than resampling *cases* we are resampling *residuals*. Here is the general outline of the method:

1. Fit the model of interest to the data. In this case, the model is just the simple linear regression of the response  $y$  on the predictor  $x$ . Compute the test statistic of interest, given by ALR[E5.23]. Save the fitted values  $\hat{y}$  and the residuals  $\hat{e}$ .

2. A bootstrap sample will consist of the original  $x$  and a new  $y^*$ , where  $y^* = \hat{y} + e^*$ . The  $i$ th element of  $e^*$  is obtained by sampling from  $\hat{e}$  with replacement.
3. Given the bootstrap data  $(x, y^*)$ , compute and save  $\text{ALR}[\text{E5.23}]$ .
4. Repeat steps 2 and 3  $B$  times, and summarize results.

# 6

---

## *Polynomials and Factors*

### 6.1 POLYNOMIAL REGRESSION

**JMP** ALR[F6.2] is a plot of the design points in the cakes data, with the center points slightly jittered to avoid overprinting. JMP allows for jittering of points along the horizontal axis when the variable plotted on this axis is ordinal or nominal. Change *X1* to one of these types as discussed in Section 0.4.1. Plot *X2* on the vertical axis and *X1* on the horizontal using the Fit Y by X platform. From the drop-down plot menu, select Display Options → Points Jittered. This will spread out the duplicated center points for jittered values of *X1* and result in a plot similar to Figure 6.1.

Polynomial models generally require creating many terms that are functions of a few base predictors. As discussed in Section 5.2, Fit Y by X can be used when one term is used in the polynomial mean function and Fit Model can be used when one or more terms are used in the polynomial mean function. Both fits will center the data by subtracting the mean of the term before calculating the higher-order terms. This is recommended, and is done to reduce correlation between the main effects and the high-order terms and increase numerical stability. Details on how to obtain the uncentered fit are given in Section 5.2.

#### 6.1.1 Polynomials with several predictors

**JMP** Be sure to return *X1* to be a continuous variable rather than some other type if you changed it to draw the figure in the last section. The second-order mean function in ALR[E6.4] can be fit in JMP using the quadratic response

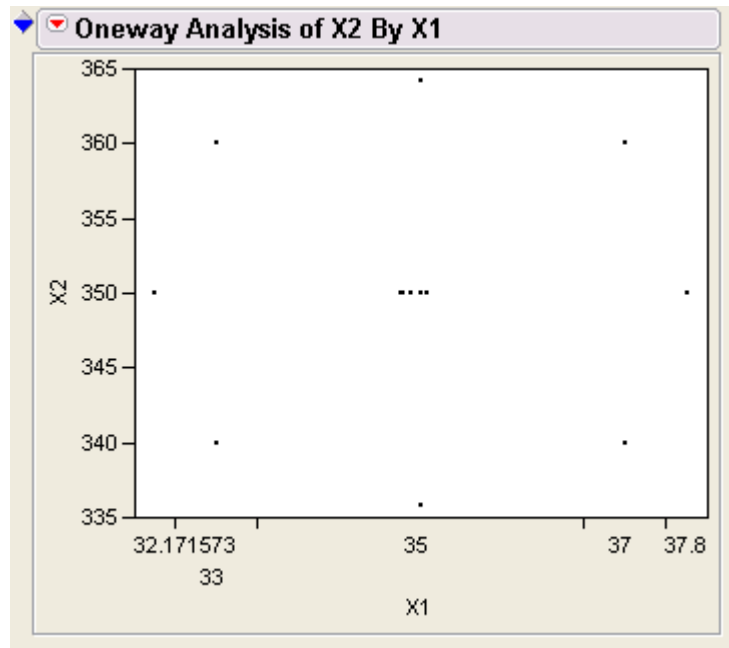


Fig. 6.1 JMP's version of the jittered scatterplot ALR[F6.2].

surface model available in the Fit Model platform. For the cakes data, enter  $y$  as the Y variable in the Fit Model dialog. Next select both  $X_1$  and  $X_2$  by shift-clicking on both variables, then select **Response Surface** from the drop down MACROS list. This will add the following terms to the model effects box:

$X_1 \& RS$	<i>Linear term for <math>X_1</math></i>
$X_2 \& RS$	<i>Linear term for <math>X_2</math></i>
$X_1 * X_1$	<i>Quadratic term for <math>X_1</math></i>
$X_1 * X_2$	<i>Interaction term for <math>X_1</math> and <math>X_2</math></i>
$X_2 * X_2$	<i>Quadratic term for <math>X_2</math></i>

JMP adds “& RS” to the names of the linear terms (often called *main effects* in response surfaces problems like this one). By clicking RUN MODEL the centered polynomial model will be fit.

JMP produces plots similar in spirit to ALR[F6.3], called the *Prediction profiler* and the *Interaction profiles*. The Prediction profiler is an interactive graphical tool that allows you to explore a fitted response surface in a very interesting and useful way. When you fit the quadratic response surface to the cakes data, you will initially get Figure 6.2. In this figure, there is one graph for each base predictor in the mean function, in this problem  $X_1$  and  $X_2$ . The figure for  $X_1$ , for example, we fix  $X_2 = 350$  and so the curve is given

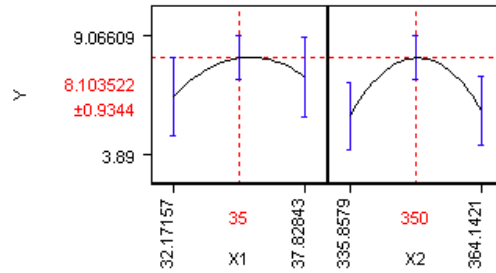


Fig. 6.2 Prediction profiler from the response surface option in the Fit models platform.

by, using the uncentered formula,

$$E(Y|X = \widehat{x}_1, X_2 = 350) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2(350) + \hat{\beta}_3 x_1^2 + \hat{\beta}_4(x_1)(350) + \hat{\beta}_5(350^2) \quad (6.1)$$

$$= (\hat{\beta}_0 + 350\hat{\beta}_2 + 350^2\hat{\beta}_5) + (\hat{\beta}_1 + 350\hat{\beta}_4)x_1 + \hat{\beta}_3 x_1^2 \quad (6.2)$$

which is a quadratic polynomial in  $x_1$ . If you change the value of  $X_2$ , by dragging the dashed line in the plot for  $X_2$  to the left or the right, then the new value of  $X_2$  will replace 350 in (6.2), and the plot will be redrawn. If the interaction term has a zero coefficient, or there is no interaction in the mean function, then the prediction profiles will not change shape, but if there is an interaction, then the profiles will change shape. These graphs can be very useful in exploring the shape of the fitted response curve<sup>1</sup> Further information about these useful plots is given in JMP-START[P413].

The Interaction profiles provide a different summary of this same information, as shown in Figure 6.3. These noninteractive graphs give plots of the response versus each of the base terms, with a separate response curves. For example, for  $X_1$  we get the plot of (6.2) with  $X_2$  fixed at 335.88 and 364.14.

<sup>1</sup>When you change the fixed value for  $X_2$ , the curve for  $X_1$  changes. Because JMP apparently rescales all the plots when a value is changed, the curve for  $X_2$  may appear to change, but this is due to the rescaling, not to the change in the value of  $X_2$ .

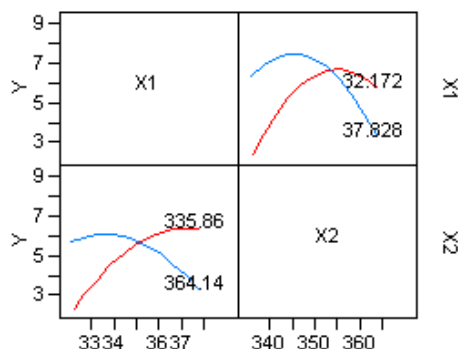


Fig. 6.3 Interaction profiles from the response surface option in the Fit models platform.

### 6.1.2 Using the delta method to estimate a minimum or a maximum

### 6.1.3 Fractional polynomials

## 6.2 FACTORS

Factors are a slippery topic because different computer programs will handle them in different ways. In particular, while SAS and SPSS use the same default for defining factors, JMP, R and S-Plus all used different defaults. A factor represents a qualitative variable with say  $a$  levels by  $a - 1$  (or, if no intercept is in the model, possibly  $a$ ) dummy variables. ALR[E6.16] describes one method for defining the dummy variables, using the following rules:

1. If a factor  $A$  has  $a$  levels, create  $a$  dummy variables  $U_1, \dots, U_a$ , such that  $U_j$  has the value one when the level of  $A$  is  $j$ , and value zero everywhere else.
2. Obtain a set of  $a - 1$  dummy variables to represent factor  $A$  by dropping one of the dummy variables. For example, using the default coding in R, the first dummy variable  $U_1$  is dropped, while in SAS and SPSS the last dummy variable is dropped.
3. JMP and S-Plus use a completely different method.

Most of the discussion in ALR assumes the R default for defining dummy variables.

**JMP** Factors can be used with the Fit model platform, but not with Fit Y by X. JMP turns a variable into a factor if the variable is of modeling type Ordinal or Nominal. The only difference between Ordinal and Nominal is the dummy variables that are created. *Using Ordinal factors will generally reproduce the results in ALR.*

If a factor is of type Ordinal, then JMP uses the parameterization discussed in ALR[6.2] that *drops the dummy variable for the first level of the factor*. If the factor is of type Nominal, then JMP uses the *effects parametrization*. If factor  $A$  has  $a$  levels, JMP will create  $a - 1$  dummy variables  $A[1], \dots, A[a - 1]$  defined to be

$$A[j] = \begin{cases} 1 & \text{if the level of } A \text{ is } j \\ -1 & \text{if the level of } A \text{ is } a \\ 0 & \text{otherwise} \end{cases}$$

The parameter estimates for these dummy variables are the difference between the mean response at the level and the mean response across all levels. To get this difference for the last level,  $a$ , take the negative sum of the parameter estimates across all other levels of  $A$ .

Always include the intercept when fitting the model because JMP will still only fit  $a - 1$  dummy variable defined above and the parameter estimates will lose the interpretation just described.

### 6.2.1 No other predictors

**JMP** We will illustrate using effects coding with the sleep data. The regression of  $TS$  on the factor  $D$  is really one-way analysis of variance, as discussed at great length in JMP-START[9]. Change the modeling type of  $D$  to nominal then fit the model using the Fit Model command. JMP will produce a few new tables when a predictor in a factor. One of interest is the Effect Details table which gives the mean of  $TS$  at each level of  $D$ . The Analysis of Variance table will be the same as the table given in ALR[T6.1B] but the Parameter Estimates table will differ because the dummy variables are not defined the same way. This table in JMP will only give the parameter estimates for the first four levels of  $D$ . To obtain the estimates for all five levels select Estimates  $\rightarrow$  Expanded Estimates from the Response popup menu. The Expanded Estimates table is given in Figure 6.4.

The estimate of the intercept for this model is the mean of  $TS$  across all levels of  $D$  and from Figure 6.4 is equal to 9.6052. Since the mean of  $TS$  for  $D = 1$  is equal to 13.0833, we can see that the parameter estimate for level one,  $D[1]$ , is indeed equal to the difference between the level mean and the overall mean,  $13.0833 - 9.8052 = 3.4781$ . We can also see that the negative sum of the estimates for  $D[1]$  through  $D[4]$  equals the estimate for  $D[5]$ .

If you set  $D$  to have modeling type Ordinal before fitting the regression, you will get results that closely parallel those given in ALR.

Term	Estimate	Std Error	t Ratio	Prob> t
Intercept	9.6051746	0.522602	18.38	<.0001
D[1]	3.4781587	0.863903	4.03	0.0002
D[2]	2.1448254	0.9389	2.28	0.0264
D[3]	0.7048254	1.060613	0.66	0.5092
D[4]	-0.794063	1.104329	-0.72	0.4753
D[5]	-5.533746	1.220636	-4.53	<.0001

Fig. 6.4 JMP's parameter estimates for the regression of  $TS$  on  $D$  from ALR[6.2.1].

### 6.2.2 Adding a predictor: Comparing regression lines

**JMP** To obtain the plots and model fits from ALR[6.2.2] use the Fit Model platform and uncheck the option Center Polynomials from the popup Model Specification menu.

**Model 1** Use model effects  $D$ ,  $\log BW$ ,  $D * \log BW$ . All three terms can be added to the effects box at once by selecting both predictors by pressing control while clicking on their names, then selecting Macros  $\rightarrow$  Full Factorial.

**Model 2** Use the model effects  $D$  and  $\log BW$ .

**Model 3** Use the model effects  $\log BW$  and  $D * \log BW$ . The interaction term is specified by control-clicking both  $D$  and  $\log BW$  then pressing the CROSS button. A caution alert will be given after running the model because the interaction term is fit without the main effect  $D$  in the model. Press continue in the dialog to obtain the common intercept fit.

**Model 4** Use the model effect  $\log BW$ .

All these models will produce regression plots corresponding to those given in ALR[F6.6]. To get different plotting symbols for each level of  $D$ , right click on the plot and select Row Legend. This allows you to select a column whose row values determine the plotting symbol and/or color. Pick  $D$  from the column list then check the plotting option you prefer and press OK.

## 6.3 MANY FACTORS

### 6.4 PARTIAL ONE-DIMENSIONAL MEAN FUNCTIONS

ALR[F6.8] is much more compelling in color, and is shown here as Figure 6.5.

**JMP** The partial one-dimensional mean function is fit in JMP using the nonlinear model platform. We show how to do this here for the Australian

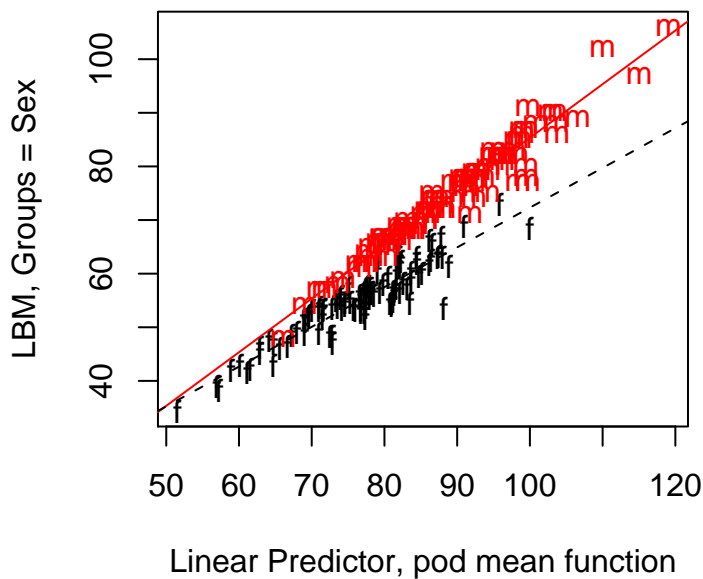


Fig. 6.5 ALR[F6.8] in color.

Institute of Sport data. The procedure is fairly complex, thereby limiting the usefulness of this methodology with JMP.

For the Australian Institute of Sport data, the mean function given in ALR[E6.26] must be specified with the Formula Editor. Define a new column with the name of this function, say *Expect*, and choose the Formula property. The parameters in the mean function are created by selecting Table Columns and changing it to Parameters list. Figure 6.6 shows this list with all parameters defined. Before these parameters are defined, only **New Parameter** will be in the list. Click on it once and in the dialog enter **b0** as the name and 1 as the starting value. Repeat this for the other five parameters in the mean function. Using both the Parameters and Table Columns lists, create the formula shown in Figure 6.6.

After the mean function is created, the nonlinear model can be fit by selecting **Analyze** → **Modeling** → **Nonlinear**. In the dialog, enter *LBM* as the Response and the mean function, *Expect*, as the Predictor Formula and press OK. Unlike linear regression, the nonlinear model is not fit at this point. Instead you get the dialog in Figure 6.7 which allows you to control the fitting process.

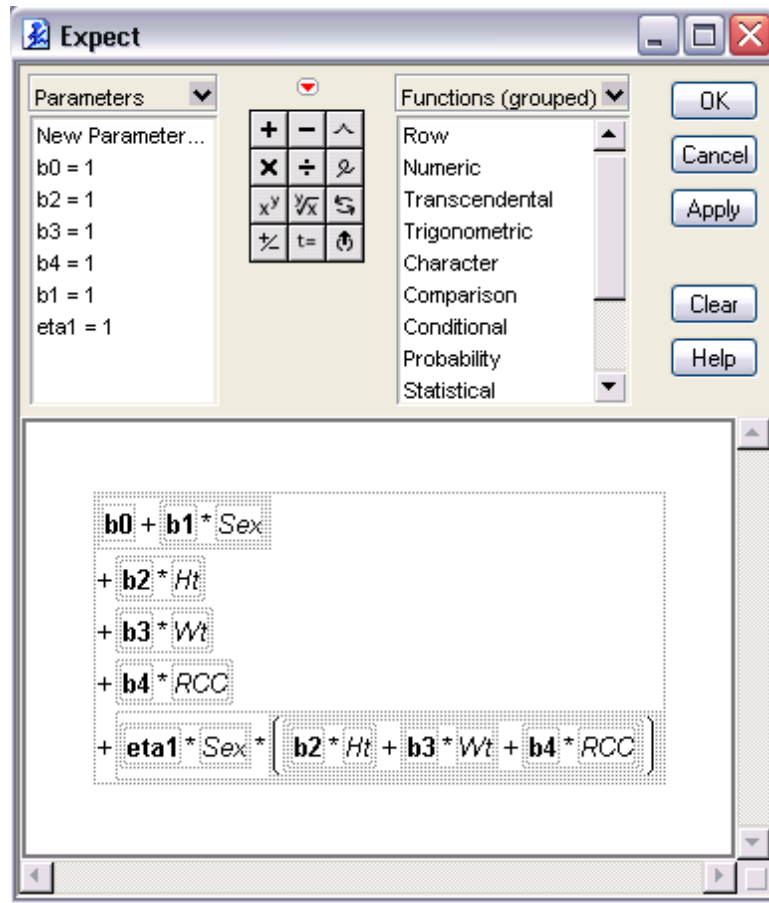


Fig. 6.6 Formula editor for defining the partial one-dimensional mean function  $ALR[E6.26]$ .

Since the initial values of the parameters have already been given, press OK and the nonlinear model will be fit. The solution given for this model is

SSE	DFE	MSE	RMSE
1185.9110807	196	6.0505667	2.45979

Parameter	Estimate	ApproxStdErr
b0	-14.6564018	6.46448633
b2	0.1462637949	0.03424361
b3	0.709342094	0.0241639
b4	0.7247602339	0.58540196
b1	12.847198671	3.76342086
eta1	-0.258749133	0.03446363

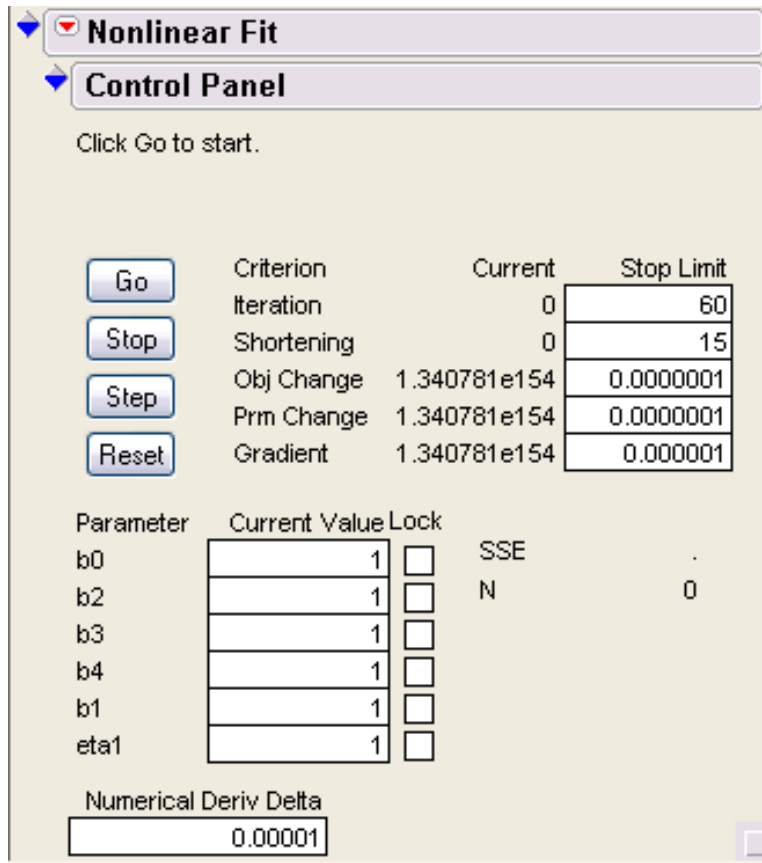


Fig. 6.7 Nonlinear fit control panel to fit the partial one-dimensional mean function ALR[E6.26].

ALR[F6.8] can be obtained by modifying the nonlinear fitted mean function. Save the fitted function by selecting **Save Formulas** → **Save Prediction Formula** from the Nonlinear Fit popup menu. Right click on the **Fitted Expect** column of the data table and select **Formula**. Edit the formula, using the “peel operator” keypad button to remove all the terms and parameters except those for  $\hat{\beta}_2Ht + \hat{\beta}_3Wt + \hat{\beta}_4RCC$ . Next, fit *LBM* versus *Fitted Expect* and use the Row Legend to change the plotting symbols by the variable *Sex*. Fit two regressions for each sex by selecting **Group By** from the popup Plot menu and choosing the variable *Sex*. Then select the **Fit Line** plotting option to fit both lines.

## 6.5 RANDOM COEFFICIENT MODELS

**JMP** The Fit Model platform in JMP can be used to fit a slightly different version of the random coefficients model given in ALR. The default method of fitting this model is REML (REstricted Maximum Likelihood), which is also the method used to obtain results in ALR[6.5].

Consider fitting ALR[E6.27] for the chloride data. *Month* and *Cl* must be of modeling type continuous and *Type* and *Marsh* must be nominal; JMP does not allow Ordinal factors with random effects. This model has fixed main effects *Month* and *Type* and a random slope and intercept for each *Marsh*. The model that JMP will fit requires that the random effects for the intercept and for the slope be uncorrelated, but the fit in ALR allows these to be correlated. However, since JMP uses some centering, the assumption of uncorrelated random effects may be acceptable.

In the effects box of the Fit Model dialog, first enter *Month* and *Type*. Next, add the random intercept component by adding *Marsh* to the effects box, then highlight the *Marsh* variable from the effects list and the *Type* variable from the columns list and press NEST. Highlight this nested term and select **Random Effect** from the popup Attributes menu to obtain the random intercept **Marsh[Type]& Random**. The random slope is created by crossing the random intercept term with *Month*, To define this variable, select **Marsh[Type]& Random** from the effects list and *Month* from the columns list and press CROSS. Select a minimal report Emphasis and run the model.

JMP includes parameter estimates for each of the crossed and nested random terms, but since they are not actual parameters in the model, they can be ignored. In JMP the random effects are summarized by the Variance Component table .

The random coefficients model ALR[E6.29] has a random intercept component, but no random slope. To fit ALR[E6.29], remove the last term, the random slope **Marsh\*Month[Type]& Random**, from the model above then run the reduced model containing only the effects **Month**, **Type**, and **Marsh[Type]& Random**.

A somewhat similar example is given in JMP-START[P360]

7

---

# Transformations

## 7.1 TRANSFORMATIONS AND SCATTERPLOTS

### 7.1.1 Power transformations

### 7.1.2 Transforming only the predictor variable

**JMP** JMP doesn't have a scaled power transformation, ALR[E7.3], but it does allow you to plot a transformed simple linear regression line from a plot of the untransformed variables. ALR[7.3] can be drawn by loading the `ufcwc` data and plotting *Height* versus *Dbh* using the Fit Y by X platform. The simple linear regression line, corresponding to  $\lambda = 1$ , is added by selecting Fit Line from the plot menu. The remaining two lines are OLS lines for the regressions of *Height* on  $\log(\text{Dbh})$ ,  $\lambda = 0$ , and *Height* on  $1/\text{Dbh}$ ,  $\lambda = -1$ . The first of the lines is added by selecting Fit Special from the plot menu, then checking the X transformation **Natural Logarithm** and pressing OK, and the second uses the **Reciprocal**. *Be sure to transform the horizontal or X axis and not the vertical or Y axis.* In addition to adding this OLS line to the plot, the regression output from this fit will be given. We can see the *RSS* for the log fit is 152,232, which is smaller than 193,739, the *RSS* from the untransformed linear regression fit. The inverse transformation fit is obtained by checking the **Reciprocal** transformation option in the Fit Special dialog. The *RSS* for this fit is 197,352. Using the logic of ALR[7.1.2], we would choose the log transformation of *Dbh* ( $\lambda = 0$ ) as the best choice of transformation because it has the smallest *RSS*. The plot with all three fits is shown in Figure 7.1.

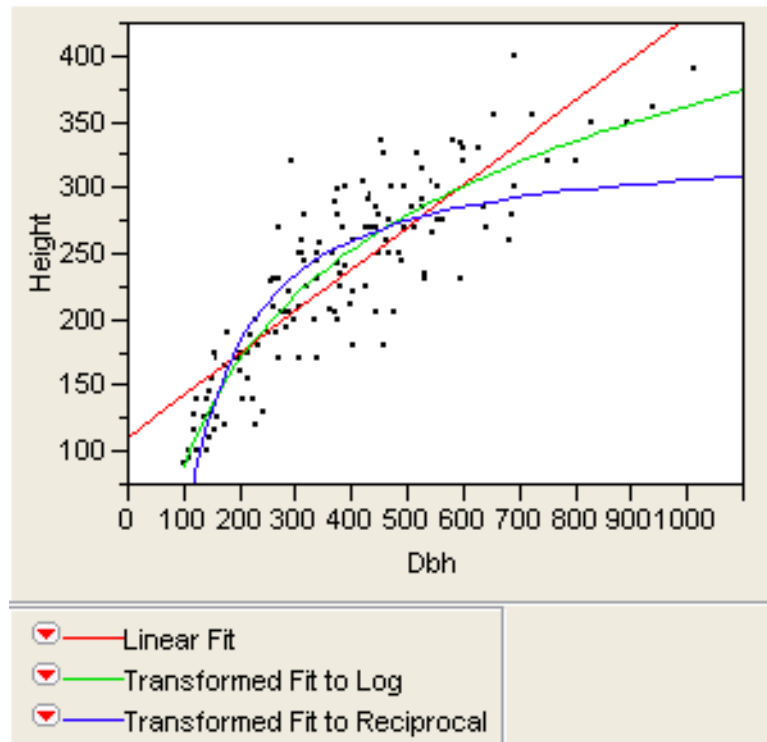


Fig. 7.1 JMP's version of ALR[F7.3].

### 7.1.3 Transforming the response only

**JMP** You can draw the inverse fitted value plots as described in ALR[7.1.3], but it takes several steps:

1. Use the Fit Model platform to fit a regression with the untransformed response and mean function you want to use that might include transformed predictors, factors, interactions, and so on.
2. From this fit, save the fitted values by selecting **Save predicted**s from the popup menu for the model you have fit.
3. Use the Fit Y by X platform, and set the predicted values to the the Response (vertical variable) and response to by the "X, Factor" variable (this is, after all, an *inverse* fitted value plot).
4. Use the method outlined in Section 7.1.2 above to select a transformation visually.

We will use the data in `highway.txt` to show an example of this method. The predictor transformations from ALR[7.2.2] will be used in the model fit, so we

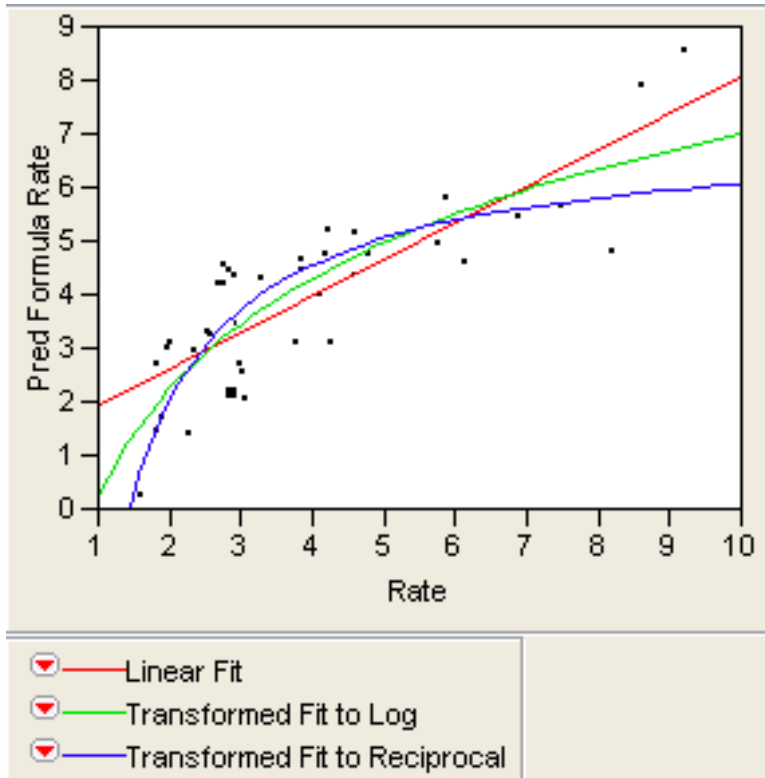


Fig. 7.2 Transforming the response in the highway data.

first must define the new terms  $\log(Len)$ ,  $\log(ADT)$ ,  $\log(Trks)$ , and  $\log Sigs1$ . The first three terms are equal to the log of the appropriate predictor, but the last term is equal to  $\log Sigs1 = \log((Len \times Sigs + 1)/Len)$ . Using the Fit Model platform, fit the regression of  $Rate$  on  $\log(Len)$ ,  $\log(ADT)$ ,  $\log(Trks)$ ,  $Slim$ ,  $Shld$  and  $\log Sigs1$ , then save the predicted values.

If  $Pred Rate$  is the column name of the saved predicted values, fit the regression of  $Pred Rate$  on  $Rate$  using the Fit Y by X platform. The scatterplot produced is the inverse response plot. By following the steps given in Section 7.1.2, we can determine the best  $\lambda$  for the transformation of  $Rate$  by adding the linear, log, and reciprocal fitted lines to the plot. This plot is given in Figure 7.2. The  $RSS$  for the reciprocal, log, and linear fits are, respectively, 34.72, 30.73, and 32.46. From the inverse response plot and the  $RSS$  values, we can conclude that the log transformation is the best choice of transformation.

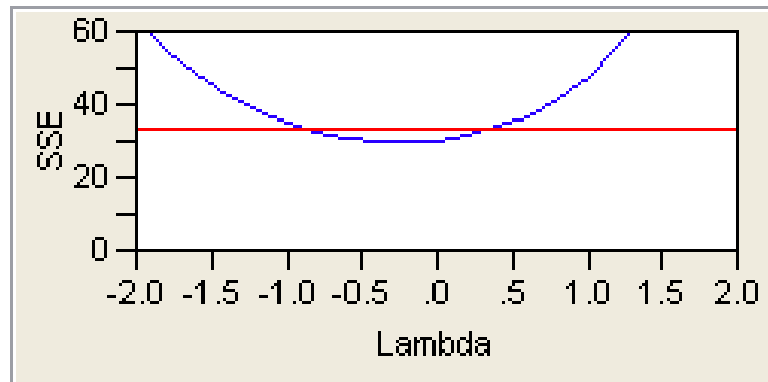


Fig. 7.3 The Box-Cox response transformation in JMP.

#### 7.1.4 The Box and Cox method

**JMP** The Box-Cox transformation method is available in the Fit Model platform. Continuing with the highway data from the last section, select Factor Profiling → Box Cox Y Transformation from the response menu of the regression fit of *Rate* on  $\log(\text{Len})$ ,  $\log(\text{ADT})$ ,  $\log(\text{Trks})$ , *Slim*, *Shld* and  $\log(\text{Sigs1})$ . The result is the graph shown in Figure 7.3

The figure ALR[F7.8] is a plot with  $\lambda$  on the horizontal axis, and  $-.5 \log(RSS/n)$  on the vertical axis, while the plot produced by JMP is of  $\lambda$  on the horizontal axis versus *RSS* on the vertical axis. Because of the minus sign, the maximum of ALR[F7.8] corresponds to the minimum of Figure 7.3. The curve in ALR allows reading a confidence interval off the graph; to do so with the JMP figure requires that you do a side calculation using the formulas in ALR.

The value of the best transformation of *Rate*, the value of  $\lambda$  that minimizes the curve, can be save to the data table by selecting Save Best Transformation from the Box-Cox popup menu. The value of  $\hat{\lambda}_y$  used in this transformation can be found by checking the formula of this saved column.

## 7.2 TRANSFORMATIONS AND SCATTERPLOT MATRICES

The scatterplot matrix is the central graphical object in learning about regression models. You should draw them all the time; all problems with many continuous predictors should start with one.

**JMP** The scatterplot matrix in ALR[F7.5] can be duplicated in JMP using the Multivariate platform described in Section 1.6. This plot doesn't allow the addition of fitted or smoothed lines. You can change the plotting symbol or color by right-clicking on the plot and selecting Row Legend. For the highway

data, it may be useful to color the symbols according to the value of the variable *Hwy*.

### 7.2.1 The 1D estimation result and linearly related predictors

### 7.2.2 Automatic choice of transformation of the predictors

**JMP** JMP does not have a multivariate extension to the Box and Cox method that can be used to automatically transformation multiple predictors. To handle data with many predictors, we suggest start by viewing the scatterplot matrix and making appropriate transformation using the log and range rules discussed in ALR[7.1.1]. If predictors in the scatterplot matrix still look nonlinear after applying these rules, try finding transformations using individual scatterplots, as done in Section 7.1.2. Once the predictors are adequately transformed, then use the Box and Cox method from Section 7.1.4 to determine if a transformation of the response is needed.

## 7.3 TRANSFORMING THE RESPONSE

**JMP** See Sections 7.1.3- 7.1.4 above for the example in this section.

## 7.4 TRANSFORMATIONS OF NON-POSITIVE VARIABLES

**JMP** JMP does not provide a method like the Yeo-Johnson family to transform non-positive variables.



# 8

---

## *Regression Diagnostics: Residuals*

### 8.1 THE RESIDUALS

**JMP** Table 8.1 lists the quantities described in ALR[8] and ALR[9] available in JMP after a regression model is fit using the Fit Model platform.

#### 8.1.1 Difference between $\hat{e}$ and $e$

#### 8.1.2 The hat matrix

**JMP** The diagonal elements of the hat matrix are saved to the data table by selecting **Save Columns** → **Hats** from the popup response menu. There is no simple way in JMP to obtain the full  $n \times n$  hat matrix.

#### 8.1.3 Residuals and the hat matrix with weights

As pointed out in ALR[8.1.3], the residuals for WLS are  $\sqrt{w_i} \times (y_i - \hat{y}_i)$ . Whatever computer program you are using, you need to check to see how residuals are defined.

**JMP** The residual for WLS are *not* the values saved by selecting **Save Columns** → **Residuals**. They can be calculated by saving these unweighted residuals and creating a new column with a formula equal to ALR[E8.13].

**Table 8.1** JMP regression options relating to residuals. These values are available from the popup response menu in the **Fit Model** platform. The saved values are given as columns in the data table. Only options discussed in ALR are listed.

JMP	ALR
<i>Save Columns options.</i>	
Prediction Formula	Predicted values, ALR[E8.2], for OLS and WLS. New fitted values are calculated using the formula when predictors values are added or changed.
Predicted Values	Predicted values, ALR[E8.2], for OLS and WLS. Cannot be used to calculate new fitted values.
Residuals	Residual values, ALR[E8.4], for OLS only.
Studentized Residuals	Get ready for name confusion. These are called <i>standardized</i> residuals in ALR, and are, defined at ALR[E9.3]. JMP does not compute what ALR calls Studentized residuals.
Std Error of Residual	The square root of the residual variance given in ALR[E8.6].
Hats	Leverages, ALR[E8.11], for OLS and WLS.
Cook's D Influence	Cook's distance, ALR[E9.6].
<i>Row Diagnostics options.</i>	
Plot Actual by Predicted	Produces plot of response versus predicted.
Plot Effect Leverage	Produces added-variable plots.
Plot Residual by Predicted	Produces a residual plot.
Plot Residual by Row	Plots residual by case number.
Press	Displays <i>PRESS</i> statistic, ALR[E10.10].

#### 8.1.4 The residuals when the model is correct

#### 8.1.5 The residuals when the model is not correct

#### 8.1.6 Fuel consumption data

**JMP** The plots in ALR[F8.5] must be made separately in JMP by saving the residuals and fitted values, then plotting them with the appropriate variable using the Fit Y by X platform. ALR[F8.5] also shows certain points identified by the variable *State* from the fuel data file. For any scatterplot, a plotting symbol is identified by case number when the selection cursor (the arrow) is placed over the symbol. Once the cursor is moved from the point, the case number is removed. To identify a point by a variable, highlight the column of the variable in the data table, then select **Cols** → **Label/Unlabel**. To make the identifier of a point stay on the plot after the cursor is moved from the point, click on the point to highlight it, then right click and select **Row Label** from the popup menu. The same steps are used to remove the label.

## 8.2 TESTING FOR CURVATURE

**JMP** JMP can check for curvature by adding squared terms to the model and using a usual  $t$ -test. Tukey's test requires three steps:

1. Fit the model of interest using Fit Model, and save the predicted values to the data grid.
2. Create a new variable with values equal to the squares of the predicted values you just saved. Call this variable `NonAdd`
3. Refit with Fit Model adding `NonAdd` to all the terms you used in step 1. The  $t$ -statistic for `NonAdd` is Tukey's test.

The significance level given in JMP for this test is incorrect because JMP will use a  $t$  distribution. The value of the statistic should be compared to the normal distribution to get the correct significance level.

For example, consider the data in `UN2.txt`. To test for curvature due to  $\log(PPgdp)$ , fit the regression of  $\log(Fertility)$  on  $\log(PPgdp)$ , `Purban` and  $\log(PPgdp)^2$ . Recall that the quadratic terms is added by selecting `log(PPgdp)` from the column list and choosing `Polynomial to Degree` from the `Macros` menu. The  $t$ -test for this quadratic term from the parameter estimates table of this fit is equal to the test statistic shown in `ALR[T8.2]`. To obtain the test statistic for `Purban`, fit the model using the terms  $\log(PPgdp)$ , `Purban` and `Purban`<sup>2</sup>. Tukey's test is computed following the outline given above.

## 8.3 NONCONSTANT VARIANCE

### 8.3.1 Variance Stabilizing Transformations

### 8.3.2 A diagnostic for nonconstant variance

**JMP** The score test of nonconstant variance can be done in JMP by following the four steps in `ALR[8.3.2]`. Consider the test for the snow geese data. Begin by saving the residuals from the fit of `photo` on `obs1`. Create a new column called `U` using the formula shown in Figure 8.1. The `Col Sum()` function will give the value  $\sum \hat{e}_j^2$  and the `Col Number()` function will equal  $n$ . Both functions are in the `Statistical` function group. Next, fit the regression of `U` on `obs1`. The `RSS` of this fit is 162.83 and is used to get the test statistic,  $S = RSS/2 = 162.82/2 = 81.41$ . The significance level of this test can be found by creating a column with the formula `1- ChiSquare Distribution(81.41,1)`. The `ChiSquare` function is in the `Probability` menu.

To get `ALR[T8.4]`, begin by using the same steps above to create `U` from the regression of `U` on `TankTemp`, `GasPres`, `GasTemp`, and `TankPres`. The `RSS` values needed to get the score statistics in `ALR[T8.4]` are found by fitting `U` with the appropriate terms.

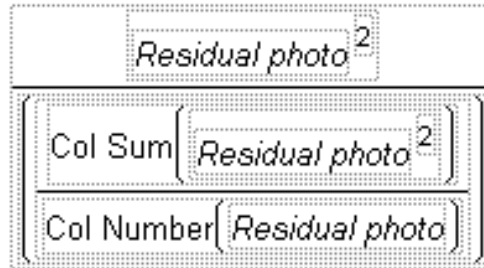


Fig. 8.1 JMP formula for creating the variable  $U$  from ALR[8.3.2].

### 8.3.3 Additional comments

## 8.4 GRAPHS FOR MODEL ASSESSMENT

### 8.4.1 Checking mean functions

**JMP** There is no procedure in JMP which will produce the marginal plots displayed in ALR[F8.13], and so this particular procedure will be difficult to use with JMP.

We suggested saving the fitted values from the regression so the two plots in ALR[F8.12] can be drawn using Fit Y by X. Compare the spline smoother fits of both plots to determine whether the mean function is adequate. To fit the smoother, select Fit Spline from the response menu. It doesn't matter which value of lambda you choose to initially fit the spline because a sidebar is provided which lets you find the best smooth visually.

To make a random linear combination of two predictors, say  $x_1$  and  $x_2$ , make a column with the formula `Random Uniform()*x1 + Random Uniform()*x2`. This new variable can be used to make plots similar to the one in ALR[F8.13D].

### 8.4.2 Checking variance functions

**JMP** Standard deviation lines cannot be added to spline smoothes.

# 9

---

## *Outliers and Influence*

### 9.1 OUTLIERS

#### 9.1.1 An outlier test

**JMP** As shown in Table 8.1, the *standardized* residuals in ALR[E9.3] can be saved to the data table in JMP by selecting **Save Columns** → **Studentized Residuals** from the popup response menu. The *Studentized* residuals in ALR[E9.4] are not available in JMP, but they can be calculated by transforming JMP's **Studentized residuals**,  $r_i$ , with the formula  $r_i \times ((n - p' - 1)/(n - p' - r_i^2))^{1/2}$ . The value  $n - p'$  is the "Error" degrees of freedom in the analysis of variance table. Since the *ordering* of Studentized and standardized residuals is always the same, you can use the above formula to compute the value of the Studentized residual only for the case with the largest  $|r_i|$  to test for a single outlier.

#### 9.1.2 Weighted least squares

#### 9.1.3 Significance levels for the outlier test

**JMP** The first step in calculating significance levels for the outlier test in JMP is to form the statistics  $t_i$  as discussed in Section 9.1.1. Suppose this column is called  $t$ . Find the absolute value of  $t$  by creating a new column which uses the **Numeric** function **Abs**. If this column is named  $abs.t$ , then the maximum value can be found by selecting **Tables** → **Summary**. In the dialog, click on  $abs.t$  and select **Max** from the **STATISTICS** popup menu and press OK.

The case number for this maximum value can be found by searching the data table, or by making a plot of the column *abs.t* with the option **Graph** → **Overlay Plot**. If the latter method is chosen, select the column *abs.t* for the Y variable and click OK. Use the pointer to determine the case number of the largest absolute *t* statistic.

Suppose we found the largest *abs.t* was 2.85, with  $n = 65$  and  $p' = 5$ . We will use a *t*-distribution with  $df = 65 - 5 - 1 = 59$  to calculate the Bonferroni bound. This lower bound is found by creating a new column with the formula  $65*2*(1 - \text{t Distribution}(2.85, 59))$  using the **Probability** function list. By subtracting the distribution from one, we get the upper tail probability. Multiplying this value by two will give a two-tailed test which is multiplied by  $n$  to get the Bonferroni bound.

#### 9.1.4 Additional comments

### 9.2 INFLUENCE OF CASES

**JMP** Table 8.1 shows the influence and distance options available in JMP. Refitting a model with a few cases excluded is very easy in JMP: Using the data grid, highlight the cases you want to remove by clicking on their row numbers; if you want to remove several cases, hold down the Control key while clicking on the row numbers. Then, select **Rows** → **Exclude/Unexclude**. Finally, refit the model you want, and fitting will be done without the excluded cases.

#### 9.2.1 Cook's distance

**JMP** Cook's distance is saved by selecting the **Cook's D Influence** option from the **Save Columns** list. See Table 8.1.

#### 9.2.2 Magnitude of $D_i$

**JMP** The plots in ALR[F9.3] can be drawn by saving JMP's **Studentized Residuals**, **Hats**, and **Cook's D Influence**. The top plot in ALR[R9.3] uses ALR's *studentized* residuals which can be calculated using Section 9.1.1. Each plot can be obtained by selecting **Graph** → **Overlay Plot** and entering one of the three variables into the Y box. After pressing OK, the variable will be plotted against the case numbers. To connect the plotted points with a line, select **Connect Thru Missing** from the popup Plot menu.

### 9.2.3 Computing $D_i$

### 9.2.4 Other measures of influence

**JMP** Added-variable plots are discussed at the end of Chapter 3, and are a regular part of the JMP Fit Model platform.

## 9.3 NORMALITY ASSUMPTION

**JMP** To draw a normal probability plot of residuals, as is seen in ALR[F9.5], first save the residuals from a regression fit. Select **Analyze** → **Distribution** and place the residuals in the **Y** box and press **OK**. The histogram and summary statistics of the residuals will appear and the probability plot can be added by selecting **Normal Quantile Plot** from the popup **Residual** menu.



# 10

---

## *Variable Selection*

### 10.1 THE ACTIVE TERMS

The first example in this chapter uses randomly generated data. This can be helpful in trying to understand issues against a background where we know the right answers. Generating random data is possible in most statistical packages, though doing so may not be easy or intuitive.

**JMP** Random data can be generated in JMP using the random variable functions provided in the formula editor. To generate the first case in ALR[10.1], press the NEW DATA TABLE button in the JMP starter window, or select File → New → Data Table. Next select Cols → Add Multiple Columns. In the columns dialog, enter the prefix `x` and choose to add 4 columns, then press OK. This will add columns for  $x_1, \dots, x_4$ . Right click on the header of  $x_1$  and select Formula from the popup list. Choose the option **Random Normal** from the **Random** function list, then press OK. Before creating the other random data, select Rows → Add rows, and in the resulting dialog, choose to add 100 rows. This will fully then create  $x_1$  with 100 observations. Set the formula for the three remaining predictors,  $x_2, x_3$ , and  $x_4$ , to be the same as the formula for the first column, to create the four independent standard normal predictors for case 1. Add an additional random normal column for the error,  $e$ , then create the response column,  $y$ , using the formula  $1 + x_1 + x_2 + e$ .

Case 2 is more complicated because the random predictors are correlated. We use the result ALR[EA.18]. If  $\mathbf{X}$  is an  $n \times p$  matrix whose rows are normally distributed with identity covariance matrix, then for any  $p \times p$  matrix  $\mathbf{S}$ , the

the matrix  $\mathbf{Z} = \mathbf{XS}$  has rows with variance  $\mathbf{S}(\mathbf{S})'$ . In our application, let  $\mathbf{S}$  be any matrix such that

$$\text{Var}(\mathbf{X}_2) = \mathbf{S}(\mathbf{S})' = \begin{pmatrix} 1 & 0 & .95 & 0 \\ 0 & 1 & 0 & -.95 \\ .95 & 0 & 1 & 0 \\ 0 & -.95 & 0 & 1 \end{pmatrix}$$

where the right side of this equation is from ALR[E10.2]. Any matrix  $\mathbf{S}$  that satisfies this equation is called a *square root* of  $\text{Var}(\mathbf{X}_2)$ . The square root is not unique. If we can find such an  $\mathbf{S}$ , then the matrix  $\mathbf{XS}$  will have rows that are normally distributed with the desired covariance matrix.

To create the correlated predictors used in case 2, use the following script

```
Var2 = [ 1 0 .95 0, 0 1 0 -.95,
        .95 0 1 0, 0 -.95 0 1];
s1 = cholesky(Var2);

dt=current data table();
x=dt<<Get As Matrix({"x1","x2","x3","x4"});
x=x*s1';
newdt=As Table(x);
```

To obtain a script window, select File  $\rightarrow$  New  $\rightarrow$  Script. Before the script in run, the last data table selected must be the data table formed for case 1 above. To run the script select Edit  $\rightarrow$  Run Script. The standard normal predictors from case 1 will be used to form the correlated predictors in case 2 by using the *Cholesky factorization* of the matrix `Var2`. These predictors are sent to a new data table with the column names *Col1*, ..., *Col4*. The response for this data table will be the column *y* which uses the formula  $1 + \text{Col1} + \text{Col2} + e$

### 10.1.1 Collinearity

**JMP** The variance inflation factors, defined following ALR[E10.5], are hidden columns of the Parameter Estimates table. To obtain this column, right click on the table and select Columns  $\rightarrow$  VIF. For case 1 in ALR[10.1], the VIF column is

```
VIF
.
1.0402133
1.0202615
1.0338504
1.0150431
```

and for case 2, the VIF column is

```
VIF
.
```

11.372096  
 10.435389  
 11.3669  
 10.443112

As expected, the VIF for case 1 are around one, while they are much larger for case 2.

### 10.1.2 Collinearity and variances

## 10.2 VARIABLE SELECTION

### 10.2.1 Information criteria

The information criteria  $ALR[E10.7]$ – $ALR[E10.9]$  depend only on the  $RSS$ ,  $p'$ , and possibly an estimate of  $\sigma^2$ , and so if these are needed for a particular model, they can be computed from the usual summaries available in a fitted regression model.

**JMP** The  $AIC$  and  $C_p$  criteria are available in JMP when using The “step-wise” personality of the Fit Model platform;  $BIC$  can only be obtained by using  $ALR[E10.8]$  and the  $RSS$  from the candidate model.

### 10.2.2 Computationally intensive criteria

Computation of  $PRESS$ ,  $ALR[E10.10]$ , is not common in regression programs, but it is easy to obtain given the residuals and leverages from a fitted model.

**JMP** The  $PRESS$  statistic for a model is obtained by fitting the regression model with the Fit Model platform, then selecting Row Diagnostics  $\rightarrow$  Press. The  $PRESS$  value for the model will be added to the Effects Tests portion of the output report.  $PRESS$  can't be used as a selection criterion.

### 10.2.3 Using subject-matter knowledge

## 10.3 COMPUTATIONAL METHODS

**JMP** JMP allows subset selection using the Stepwise personality of the Fit Model platform. This can be illustrated by example, using the highway data. Select terms and the response as usual. Before pushing the RUN MODEL button, select Personality  $\rightarrow$  Stepwise from the Personality popup menu in the upper right corner of the Fit Model dialog. After pushing RUN MODEL, you will get the dialog similar to Figure 10.1.

**All possible regressions** Look first at the top of this dialog. From the popup menu for next to Stepwise fit, you can select Stepwise fit  $\rightarrow$  All possible

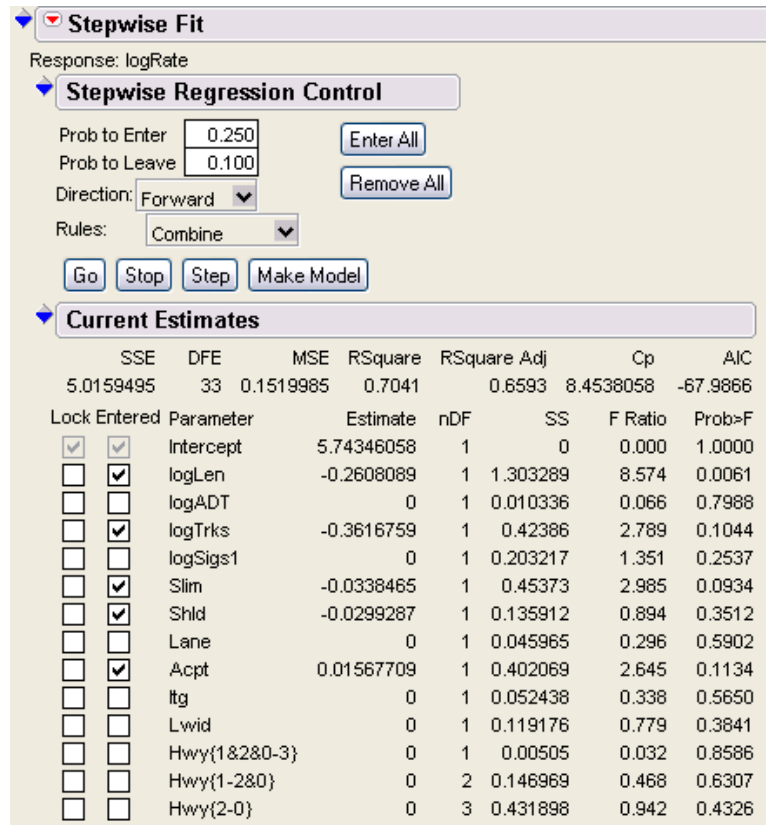


Fig. 10.1 The stepwise regression control panel and parameter estimates report table.

models, which will use a brute-force algorithm to fit all possible subset models. This can be recommended only if (1) the number of terms in the model is no more than about 7 or 8 and (2) your mean function has no factors or interactions. Selecting this option will produce a data table, and right-clicking on this table will give you a popup menu from which you can select **Columns** → **Cp** to include the  $C_p$  statistic in the table, and then **Sort by columns**, and then select **Cp**, and check the **Ascending** box to get the mean functions ordered according to  $C_p$ , from smallest to largest.

**Stepwise methods** The stepwise regression control panel seen in Figure 10.1 can be used to select regression terms based on their significance probability in the model. If your problem includes factors and interactions, always select **Rules** → **Whole Effects** to get behavior that is most similar to ALR. JMP adds/removes variables based on their  $t$ -statistics, rather than based on one of the information criteria. The “Prob to Leave” and “Prob to enter” determine a stopping rule for the stepwise method; a term will not be added, for

example, if the  $t$ -values for all currently excluded terms have  $p$ -values bigger than the value given.

You can add terms, delete terms, or go back and forth, depending on your selection in the Direction  $\rightarrow$  popup menu. Since the program only does one step at a time, you can change direction whenever you like. There is a Mixed direction which uses both forward and backward criteria. To always include a term in the mean function, check the “Entered” column then the “Lock” column next to the desired term. As seen in Figure 10.1, the intercept term is locked into all mean functions.

To use the forward criteria, start by locking in any terms wanted in all mean functions. The default direction is Forward, so next press the GO button. At each step, a term is accepted if it has a significance level below 0.25 when fitted with the terms already accepted for the mean function. If none of the remaining terms have a significance level below 0.25 then the selection method stops. A Step History table is provided that gives the significance level, sequential sum of squares,  $R^2$ , and  $C_p$  for each term which fits the criteria.

For example, consider the Highway data with all terms in ALR[T10.5]. As done in ALR[10.3], we will include  $\log(Len)$  in all mean functions so make sure its “Entered” box is the only one checked, expect for the intercept. After running the forward selection method, the Current Estimates section should have the Entered box checked for the terms  $\log(Len)$ ,  $\log(Trks)$ ,  $Slim$ , and  $Acpt$ . The Step History for this run is

Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p
logLen	Entered	0.0001	5.537263	0.3267	45.675	2
Slim	Entered	0.0000	5.30162	0.6394	10.202	3
Acpt	Entered	0.0589	0.60035	0.6748	7.9587	4
logTrks	Entered	0.1325	0.359952	0.6961	7.4145	5

To use the backwards criteria, start by locking in any terms wanted in all mean functions. Also, check the Entered box for all terms wanted in the initial mean function. From this initial model, a term will be removed if its significance level is the largest of all the terms in the mean function and if it is above the level set in the control panel. The default level in the control panel is 0.1. After the backwards method is run, a Step History summarizes the terms removed.

Again, consider the Highway data with  $\log(Len)$  is locked into the mean function. The initial mean function will contain all terms in ALR[T10.5]. After running the backward selection method with the level 0.1, the terms selected are  $\log(Len)$ ,  $\log(Sigs1)$ ,  $Slim$ , and  $Hwy$ . The Step History for this run is

Parameter	Action	"Sig Prob"	Seq SS	RSquare	Cp	p
Shld	Removed	0.9313	0.001073	0.7913	12.008	13
Itg	Removed	0.8878	0.002762	0.7911	10.027	12
Lane	Removed	0.8443	0.005159	0.7908	8.0636	11
Lwid	Removed	0.7214	0.016436	0.7898	6.1797	10

Acpt	Removed	0.3641	0.104438	0.7837	4.9179	9
logTrks	Removed	0.2882	0.142879	0.7753	3.9278	8
logADT	Removed	0.1358	0.288215	0.7582	3.965	7

### 10.3.1 Subset selection overstates significance

## 10.4 WINDMILLS

### 10.4.1 Six mean functions

### 10.4.2 A computationally intensive approach

The data for the windmill example in ALR[10.4.2] is not included with the `alr3` library, and must be downloaded separately from [www.stat.umn.edu/alr](http://www.stat.umn.edu/alr).

# 11

---

## *Nonlinear Regression*

### 11.1 ESTIMATION FOR NONLINEAR MEAN FUNCTIONS

### 11.2 INFERENCE ASSUMING LARGE SAMPLES

**JMP** The nonlinear regression platform in JMP has already been introduced in Section 6.4 to fit a partial one-dimensional mean function. To use this platform select **Analyze** → **Modeling** → **Nonlinear**. Before the model is fit, its nonlinear mean function with parameter starting values must either be in the MODEL LIBRARY provided by JMP, or you must create the function using the formula editor. The latter option was discussed in Section 6.4.

To fit the mean function ALR[E11.16] using the `turk0` data you must define the mean function because it is not available in the library. To review, select **Cols** → **New Column** and give the mean function a name, say *Mean1*. Next, select a formula property to obtain the formula editor dialog. Change the Table Columns list to the Parameters list, then click on **New Parameter** to create each of the three parameters needed in ALR[E11.16]. Using the starting values discussed in ALR[11.2], we defined the parameters `th1` with starting value 620, `th2` with starting value 180, and `th3` with starting value 8. The mean function defined with these parameters is  $\text{th1} + \text{th2} * (1 - \text{Exp}(-(\text{th3} * A)))$ .

The nonlinear model for the data in `turk0` is fit by selecting the nonlinear platform, then entering *Gain* as the response and *Mean1* as the predictor formula. After OK is pressed, the Control Panel dialog will appear giving you the option of changing the iteration criteria or the starting parameter values. Typically, no changes will be needed and the model can be fit by

pressing GO. As discussed in Section 6.4, the solution will contain the model summary statistics as well as parameter estimates and approximate standard error. When one predictor is used in the mean function, a scatterplot with the estimated mean function will also be provided. This is how to obtain ALR[F11.2]. Finally, a Correlation of Estimates table is given which is a rescaled version of the matrix ALR[E11.14].

Extra output options are available in the Nonlinear Fit popup menu. For example, the mean function, standard error for prediction or individual, and residual formulas can all be saved to the data table. The Custom Estimates option allows you to enter an expression of the parameters and JMP will calculate the expression value using current estimates and its standard error using a first-order Taylor series approximation.

The same steps above can be used with the `turkey` data to fit the mean functions ALR[E11.16]-ALR[E11.19] using nonlinear weighted least squares. Since the weights are equal to the number of pens,  $m$ , simply add  $m$  to the Weight box in the nonlinear dialog to fit any of these four models. To use a factor in a nonlinear model, you need to compute dummy variables corresponding to the levels of the factor.

### 11.3 BOOTSTRAP INFERENCE

**JMP** Bootstrap inference can only be done in JMP using scripting.

### 11.4 REFERENCES

# 12

---

## *Logistic Regression*

Both logistic regression and the normal linear models that we have discussed in earlier chapters are examples of *generalized linear models*. Many programs, including SAS, R, and S-Plus, have procedures that can be applied to any generalized linear model. Both JMP and SPSS seem to have separate procedures for logistic regression. There is a possible source of confusion in the name. Both SPSS and SAS use the name *general linear model* to indicate a relatively complex linear model, possibly with continuous terms, covariates, interactions, and possibly even random effects, but with normal errors. Thus the general linear model is a special case of the generalized linear models.

### **12.1 BINOMIAL REGRESSION**

#### **12.1.1 Mean Functions for Binomial Regression**

### **12.2 FITTING LOGISTIC REGRESSION**

**JMP** The Fit Y by X and Fit Model platforms can be used to fit logistic regression. The program decides to fit these models *if the response variable is a nominal, rather than a continuous, variable*. when the response is defined to be a nominal variable. Since JMP may have defined the response as continuous when importing the data, always be sure to check the type in the data table before fitting the model. JMP-START[12] provides details on logistic regression models in JMP.

A nominal variable treats its values as *category labels*, not as *response values*, so there is, in principle, some ambiguity as to which category corresponds to “success” and which to “failure.” When your response is coded with the values 0 and 1, JMP will equate 0 with “success” and 1 with “failure,” the opposite of what is done in ALR. If your response variable has the categories “Died” and “Survived”, it will equate “Died” with success and “Survived” with failure. In general, JMP alphabetizes the category labels, and the first category becomes “success” and the remaining category becomes “failure.” This leads to the curious choice that if you label your categories as “Failure” and “Success”, JMP will alphabetize the labels, and so since “Failure” alphabetizes before “Success,” JMP will model the probability of Failure, not the probability of success.

Although the assignment of categories to labels can be confusing, it is not really important. *Interchanging the role of the two categories will change the signs of all estimated coefficients, but will leave all other summaries, like tests and standard errors, unchanged.* In the output from a fitted model, JMP does report what it is doing.

### 12.2.1 One-predictor example

**JMP** We use the file `blowBF` as in the text to illustrate calculations. To fit the logistic regression of  $y$  on  $\log(D)$ , we can use the Fit Model platform. The response  $y$ , which has all values of 1 or 0, must be specified as a nominal variable, not continuous. The single predictor is  $\log(D)$ , using base-two logarithms. As a reminder, to transform to base-two logarithms, create a new variable, and to get the formula for this variable select Log from the **Transcendental** list of functions and choose  $D$  from the table columns list. The Log function will calculate the natural log but you can change it to base 2 by editing area highlight  $D$  in red by clicking on it, then type , 2.

The standard output consists of three parts, a graph and two blocks of text. We examine these from the bottom to the top. The bottom block is as follows:

Parameter Estimates				
Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	7.89245884	0.6325499	155.68	<.0001
logD	-2.2626148	0.1914021	139.74	<.0001

For log odds of 0.00/1.00

At the foot of the report the statement `For log odds of 0/1` tells you that JMP is treating category 0 as a “success” and category 1 as a “failure,” the opposite of what is reported in ALR[T12.1], and so the sign of the coefficient estimate for  $\log(D)$  is changed. There are a few other differences as well. The “ChiSquare” test statistics in the JMP table are the square of the  $z$ -values given in ALR[T12.1]. The JMP test statistic is compared to a  $\chi^2$  distribution to obtain the  $p$ -value in the last column; the test in ALR is compared to the

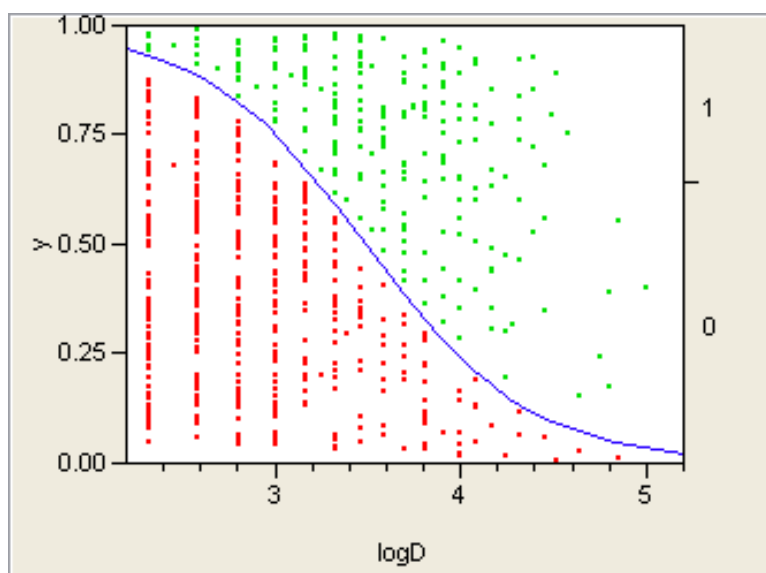


Fig. 12.1 The scatterplot produced by the Fit Y by X procedure for fitting the logistic regression of  $y$  on  $\log(D)$  for the blow down data.

normal distribution to get the  $p$ -values. Since the  $df$  are large, these two approximations give the same answers.

To obtain the residual deviance, ALR[E12.8], use the likelihood values provided in the “Whole Model Test” report table, which is displayed as a regular part of the output. For the blowdown data this table is

Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	100.48269	1	200.9654	<.0001
Full	327.62100			
Reduced	428.10369			

The residual deviance of this logistic model is two times the *Full* negative log-likelihood, or  $2 \times 327.621 = 655.242$ . Two times the *Reduced* negative log-likelihood,  $2 \times 428.104 = 856.208$ , is the null deviance for fitting the logistic regression with only the intercept term. This is the deviance for the Intercept in ALR[T12.4]. The *Difference* negative log-likelihood is the difference between the reduced and full likelihoods and the “ChiSquare” value is two times the likelihood difference. The ChiSquare value of 200.97 is the change in deviance between the first two models in ALR[T12.4]. The  $p$ -value in this table is the significance level for the test of the full logistic model against the intercept-only model. Pearson’s  $\chi^2$  is not available from the logistic fit in JMP.

The final bit of output is the plot shown in Figure 12.1, which is more or less equivalent to ALR[F12.1A]. The points in the plot are jittered vertically

much more than the points in ALR[F12.1A] to give the false impression that the response variable was any possible value between zero and one, rather than exactly zero or one. Also, the logistic regression fit produced by JMP gives the probability of “failure” ( $y = 0$ ), while the fit in ALR[12] gives the probability of “success” ( $y = 1$ ). This is why the curve in Figure 12.1 is the opposite of that in ALR[F12.1A].

The density curves in ALR[F12.1B] cannot be produced in one plot using JMP. To fit a density smooth of  $\log(D)$  for each level of  $y$ , select **Analyze** → **Distributions**. Move  $\log(D)$  to the Y box and move  $y$  to the BY box, then press OK. This will produce vertical histograms of  $\log(D)$ . To change the direction to horizontal, select **Stack** from the Distributions popup menu. To add the density smoothes to each histogram, select **Fit Distribution** → **Smooth Curve** from the popup menu for  $\log(D)$ .

The **Analyze** → **Fit model** platform can also be used to get more or less the same output, but without Figure 12.1. This platform must be used when you have more than one term in the mean function, as will be illustrated shortly.

### 12.2.2 Many Terms

**JMP** The two models in ALR[T12.2] are fit the same way described in Section 12.2.1. In particular, use the **Analyze** → **Fit models** platform. Select the response to be a categorical variable with two levels, and use predictors, for ALR[T12.2B],  $\log(D)$ ,  $S$ , and  $\log(D):S$ . Remember to uncheck the **Center Polynomials** option in the Model Specifications popup menu if you want parameter estimates to match those in ALR. The resulting output will include:

Whole Model Test				
Model	-LogLikelihood	DF	ChiSquare	Prob>ChiSq
Difference	157.23088	3	314.4618	<.0001
Full	270.87280			
Reduced	428.10369			
Parameter Estimates				
Term	Estimate	Std Error	ChiSquare	Prob>ChiSq
Intercept	3.67817162	1.4252303	6.66	0.0099
$\log D$	-0.4006952	0.4390015	0.83	0.3614
$S$	11.2054055	3.6386617	9.48	0.0021
$S*\log D$	-4.9107663	1.1403007	18.55	<.0001
For log odds of 0.00/1.00				

The parameter estimates are of opposite sign to those in ALR[T12.2B] because JMP models the probability of failure rather than the probability of success. The deviance shown in ALR[T12.2.B] is equal to twice the negative log-likelihood shown in the output above. The “ChiSquare” test shown is like an overall analysis of variance test of the hypothesis that the coefficients for everything but the constant are zero versus the alternative that they are not all zero. That hypothesis is firmly rejected here.

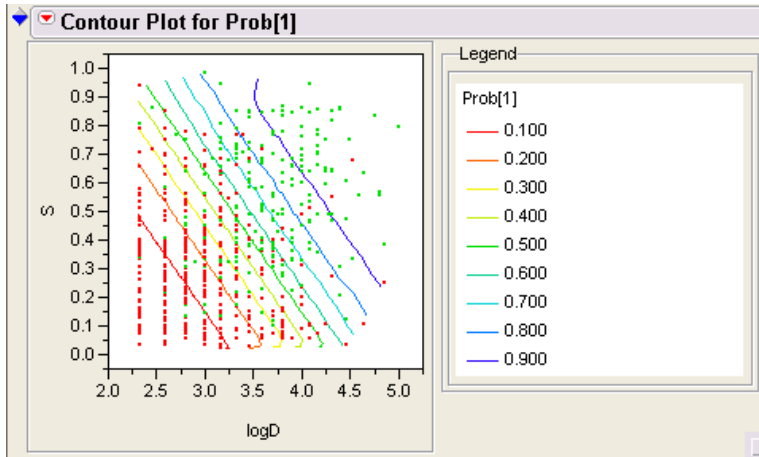


Fig. 12.2 JMP's version of ALR[F12.4A]

For any model with more than one term, you can obtain the change in deviance from adding each term to the remaining terms by selecting **Likelihood Ratio Tests** from the Logistic Fit popup menu. The change in deviance and  $p$ -values in ALR[T12.4] are found by doing this test for each of the three models considered for the blow down data.

The density smoothes in ALR[F12.3A] can be drawn as described in Section 12.2.1. ALR[F12.3B] is a scatterplot of  $S$  and  $\log(D)$  with the marker legend determined by the value of  $y$ . Recall that markers can be set by right-clicking on the plot and selecting **Row Legend**.

The plots in ALR[F12.4] can be produced in JMP by using the contour plot graphic platform. Before plotting, the probability formula for logistic fit of the no interaction and interaction mean functions must be saved by selecting **Save Probability Formula** option from the Fit menu. For each model, four values will be saved to the data table, the estimated linear combination  $\hat{\beta}'x$ , the estimated probability of  $y = 0$  and  $y = 1$ , and the most likely  $y$ . This last value is 0 when the estimated probability of 0 is greater than 0.5, otherwise it is 1.

Next select **Graph**  $\rightarrow$  **Contour Plot**. The Y role for this plot will be the probability formula. Since ALR[F12.4] plots the probability of blow down ( $y = 1$ ), place the saved formula *Prob[1]* in the Y box. The X role will be the terms for the scatterplot axes, so place  $\log(D)$  and  $S$  into this box and press OK. To add the data points to the contour plot and remove the point cloud boundary, select **Show Data Points** then **Show Boundary** from the Contour Plot menu. Figure 12.2 shows JMP's version of ALR[F12.4A]. To add or remove contours, select **Change Contours**  $\rightarrow$  **Specify Contours** from the popup Contour Plot menu.

### 12.2.3 Deviance

### 12.2.4 Goodness of Fit Tests

**JMP** The data in the `Titanic` data file look like this:

Surv	N	Class	Age	Sex
20.00	23.00	Crew	Adult	Female
192.00	862.00	Crew	Adult	Male
1.00	1.00	First	Child	Female
5.00	5.00	First	Child	Male
140.00	144.00	First	Adult	Female
57.00	175.00	First	Adult	Male
13.00	13.00	Second	Child	Female
11.00	11.00	Second	Child	Male
80.00	93.00	Second	Adult	Female
14.00	168.00	Second	Adult	Male
14.00	31.00	Third	Child	Female
13.00	48.00	Third	Child	Male
76.00	165.00	Third	Adult	Female
75.00	462.00	Third	Adult	Male

*This file format is not appropriate for use with JMP, but tools are provided in JMP to convert this file to a format that can be used. Here are the steps:*

1. JMP requires both the number of successes, `Surv`, and the number of failures, given by `N-Surv`. Start by computing a new variable called, for example, `Death=N-Surv`.
2. The response variable is a categorical variable that we will call *Outcome* with values “Surv” and “Death”. The table above has 14 rows, one for each combination of *Class*, *Age* and *Sex* that occurs in the data. The table the JMP requires will have 28 rows, 14 rows for the values of *Outcome = Surv* and 14 rows with *Outcome = Death*.

To create this new data set, select `Tables` → `Stack` and place *Surv* and *Death* in the Stack Columns box. In the text area for “Stacked data column,” type `Count`, and in the text area for ID, type `Outcome`. Press `Stack`, and a new data set will be created with values as follows:

N	Class	Age	Sex	Outcome	Count
23.00	Crew	Adult	Female	SURV	20
23.00	Crew	Adult	Female	Death	3
862.00	Crew	Adult	Male	SURV	192
862.00	Crew	Adult	Male	Death	670
1.00	First	Child	Female	SURV	1
1.00	First	Child	Female	Death	0
5.00	First	Child	Male	SURV	5

5.00	First	Child	Male	Death	0
144.00	First	Adult	Female	SURV	140
144.00	First	Adult	Female	Death	4
175.00	First	Adult	Male	SURV	57
175.00	First	Adult	Male	Death	118
13.00	Second	Child	Female	SURV	13
13.00	Second	Child	Female	Death	0
11.00	Second	Child	Male	SURV	11
11.00	Second	Child	Male	Death	0
93.00	Second	Adult	Female	SURV	80
93.00	Second	Adult	Female	Death	13
168.00	Second	Adult	Male	SURV	14
168.00	Second	Adult	Male	Death	154
31.00	Third	Child	Female	SURV	14
31.00	Third	Child	Female	Death	17
48.00	Third	Child	Male	SURV	13
48.00	Third	Child	Male	Death	35
165.00	Third	Adult	Female	SURV	76
165.00	Third	Adult	Female	Death	89
462.00	Third	Adult	Male	SURV	75
462.00	Third	Adult	Male	Death	387

To fit the model, select Analyze → Fit Model. Place *Outcome* in the Y box and *Count* in the FREQ box. For the mean function with only main effects, place the three predictors *Class*, *Age*, and *Sex* in the Model Effects box. After the model is run, the residual deviance will be the “ChiSquare” value in the Lack of Fit table. For the titanic model this table is

Lack Of Fit

Source	DF	-LogLikelihood	ChiSquare
Lack Of Fit	8	56.2833	112.5666
Saturated	13	1048.7473	Prob>ChiSq
Fitted	5	1105.0306	<.0001

The ChiSquare value is 112.57, which equals the  $G^2$  value in ALR[T12.6] for the main effects model. The other models in ALR[T12.6] are fit similarly.

## 12.3 BINOMIAL RANDOM VARIABLES

### 12.3.1 Maximum likelihood estimation

### 12.3.2 The Log-likelihood for Logistic Regression

## 12.4 GENERALIZED LINEAR MODELS

### Problems



# References

1. Chambers, J. and Hastie, T. (eds.) (1993). *Statistical Models in S*. Boca Raton, FL: CRC Press.
2. Cook, R. D. and Weisberg, S. (1982). *Residuals and Influence in Regression*. London: Chapman & Hall.
3. Cook, R. D. and Weisberg, S. (1999). *Applied Regression Including Computing and Graphics*. New York: Wiley.
4. Cook, R. D. and Weisberg, S. (2004). Partial One-Dimensional Regression Models.
5. Dalgaard, Peter (2002). *Introductory Statistics with R*. New York: Springer.
6. Davison, A. and Hinkley, D. (1997). *Bootstrap Methods and their Application*. Cambridge: Cambridge University Press.
7. Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Boca Raton: Chapman & Hall.
8. Fox, John (2002). *An R and S-Plus Companion to Applied Regression*. Thousand Oaks, CA: Sage.
9. Freund, R., Littell, R. and Creighton, L. (2003). *Regression Using JMP*. Cary, NC: SAS Institute, Inc., and New York: Wiley.
10. Furnival, G. and Wilson, R. (1974). Regression by leaps and bounds. *Technometrics*, 16, 499-511.

11. Knüsel, Leo (2005). On the accuracy of statistical distributions in Microsoft Excel 2003. *Computational Statistics and Data Analysis*, 48, 445–449.
12. Maindonald, J. and Braun, J. (2003). *Data Analysis and Graphics Using R*. Cambridge: Cambridge University Press.
13. Muller, K. and Fetterman, B. (2003). *Regression and ANOVA: An Integrated Approach using SAS Software*. Cary, NC: SAS Institute, Inc., and New York: Wiley.
14. Nelder, J. (1977). A reformulation of linear models. *Journal of the Royal Statistical Society*, A140, 48–77.
15. Pinheiro, J. and Bates, D. (2000). *Mixed-Effects Models in S and S-plus*. New York: Springer.
16. Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley & Sons, Inc.
17. Sall, J., Creighton, L. and Lehman, A. (2005). *JMP Start Statistics*, third edition. Cary, NC: SAS Institute, and Pacific Grove, CA: Duxbury. **Referred to as** JMP-START.
18. SPSS (2003). *SPSS Base 12.0 User's Guide*. Chicago, IL: SPSS, Inc.
19. Thisted, R. (1988). *Elements of Statistical Computing*. New York: Chapman & Hall.
20. Venables, W. and Ripley, B. (2000). *S Programming*. New York: Springer.
21. Venables, W. and Ripley, B. (2002). *Modern Applied Statistics with S*, 4th edition. New York: Springer. **referred to as** VR.
22. Venables, W. and Smith, D. (2002). *An Introduction to R*. Network Theory, Ltd.
23. Verzani, John (2005). *Using R for Introductory Statistics*. Boca Raton: Chapman & Hall.
24. Weisberg, S. (2005). *Applied Linear Regression*, third edition. New York: Wiley. **referred to as** ALR.
25. Weisberg, S. (2005). Lost opportunities: Why we need a variety of statistical languages. *Journal of Statistical Software*, 13(1), [www.jstatsoft.org](http://www.jstatsoft.org).

# *Index*

- Added-variable plots, 30
- Analysis platforms, 13
- Analyze
  - Distribution, 18–19, 27, 63
- Analyze
  - Distributions, 76
- Analyze
  - Fit Model, 24, 28
  - Fit model, 76
  - Fit Model, 79
  - Fit models, 76
  - Fit Y by X, 13, 15–16, 20
- Modeling
  - Nonlinear, 47, 71
- Multivariate Methods
  - Multivariate, 18
- Multivariate Methods
  - Multivariate, 19
  - Multivariate, 27
- Arc, 4
- Buttons
  - Add, 24, 28
  - Add Condition, 14
  - By, 76
  - Cross, 46, 50
  - Delimited, 8
  - Done, 38
  - Freq, 79
  - Go, 69, 72
  - Macros, 37, 42
  - Model Library, 71
  - Nest, 50
  - New Data Table, 38, 65
  - New Property, 8, 15, 17
  - New Script, 22
  - OK, 46
  - Open Data Table, 8
  - Preferences, 7
  - Run Model, 24, 28, 37, 42, 67
  - Statistics, 61
  - X, 13, 77
  - Y, 13, 62, 76–77, 79
- Change Contours
  - Specify Contours, 77
- Check marks
  - Attempt to Discern Format, 8
  - Centered Polynomial, 36
  - Constrain Intercept to:, 16, 20, 24
  - Constrain Slope to:, 16
  - Extend Current Selection, 14–15
  - if all conditions are met, 15
  - if any condition is met, 15
  - Minor, 14
  - Natural Logarithm, 51
  - Reciprocal, 51
  - Set Marker by Value, 16
  - space, 8
  - Use Text Import Preferences, 8
- Chi-squared tables, 10
- Cholesky factorization, 66

- Cols
  - Add Multiple Columns, 65
  - Label/Unlabel, 58
  - New Column, 8, 71
- Columns
  - Cp, 68
  - Lower 95%, 22
  - Upper 95%, 22
  - VIF, 66
- Commands
  - Constants, 18
  - Effect Leverage, 30
  - Formula, 8
  - is greater than, 15, 25
  - is less than or equal to, 14–15
  - Minimal Report, 24, 28, 30
  - New parameter, 47
  - Polynomial to Degree, 37
  - Probability, 62
  - Random, 65
  - Random Effect, 50
  - Response Surface, 42
  - Table Columns, 18
  - Text Import, 8
  - Text Import Preview, 8, 25
  - Transcendental, 16, 18, 74
- CRAN, 2
- Data files, 5, 7
  - Documentation, 5
  - Missing values in, 6
  - wm5.txt, 6
- Direction
  - p, 69
- Display Options
  - More Moments, 33
  - Points Jittered, 41
- Edit
  - Copy, 9
  - Journal, 9
  - Layout, 9
  - Run script, 12
  - Run Script, 22, 66
  - Save Selection As, 9
  - Undo Delete Rows, 15
- Effects parametrization, 45
- Estimates
  - Custom Test, 38
  - Custom Tests, 33
  - Expanded Estimates, 45
  - Sequential Tests, 28, 37
- Excel, 4
- Factor, 44
- Factor Profiling
  - Box Cox Y Transformation, 54
- File
  - New
    - Data Table, 38
  - New
    - Data Table, 65
  - New
    - Script, 11, 22, 66
  - Open, 6–8
  - Preferences, 7
  - Save, 9
- Fit Distribution
  - Smooth Curve, 76
- Fit Polynomial
  - 2
    - quadratic, 36
- F tables, 10
- Graph
  - Contour Plot, 77
- Graphical user interface (GUI), 3
- Graph
  - Overlay Plot, 62
- Graphs
  - saving, 9
- Graph
  - Variability/Gage Chart, 18
- Help
  - Online Manuals, 2
- Interaction profiles, 42
- Jmp files, 7
- Linked, 14
- Logarithms
  - Base two, 18, 74
- Macros, 3
- Macros
  - Full Factorial, 46
- Main effects, 42
- Missing values, 6, 32
- Normal tables, 10
- Ordinary least squares, 19
- Output
  - saving, 9
- Personality
  - Stepwise, 67
- Polynomial regression, 41
- Prediction, 23
- Prediction profiler, 42
- Quadratic, 36
- Random coefficient model, 50
- Residuals, 24
- Row Diagnostics
  - Plot Effect Leverage, 30
  - Press, 67
- Rows
  - Add rows, 65
  - Clear Row States, 14
  - Exclude/Unexclude, 25, 62
  - Hide/Unhide, 14
  - Hide/Unhide, 15

- Hide/Unhide, 25
- Row selection
  - Select all rows, 14
- Row Selection
  - Select Where, 14–15, 25
- Row state, 14
- Rules
  - Whole Effects, 68
- Save Columns
  - Hats, 57
  - Predicted Values, 24
  - Prediction Formula, 24
- Save Columns
  - Residuals, 57
- Save Columns
  - Std Error of Individual, 24, 29
  - StdErr Pred Formula, 24
  - Studentized Residuals, 61
- Save Formulas
  - Save Prediction Formula, 49
- Scripting, 22
- Scripts
  - to reproduce the book, 6
- Size/Scale
  - Frame Size, 14
  - Frame size, 15
  - X Axis, 14
- Stepwise fit
  - All possible models, 68
- Tables, 10
  - Stack, 78
  - Summary, 61
- Tools
  - Grabber, 14
  - Selection, 9
- Transformations, 15
- Transform
  - Recode, 5, 12
- T tables, 10
- T-tables, 22
- View
  - Log, 12, 22