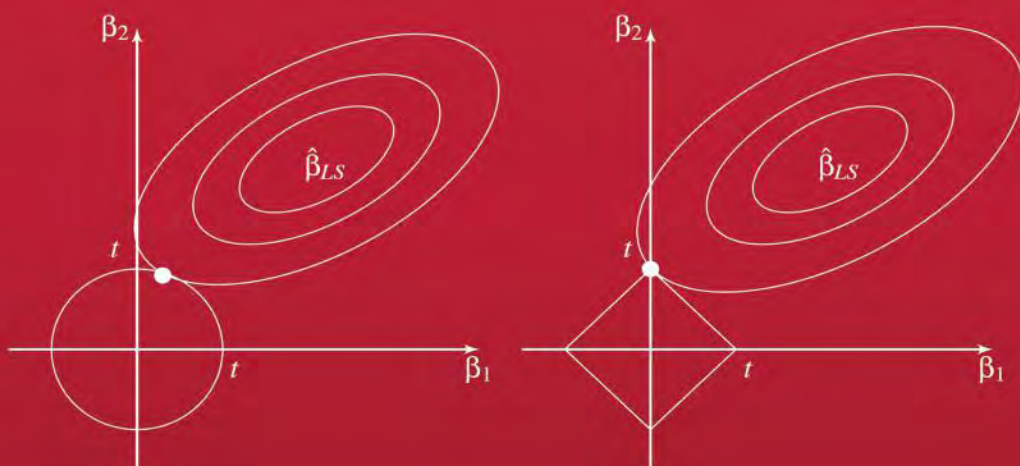


Texts in Statistical Science

Linear Models with R

Second Edition



Julian J. Faraway



CRC Press
Taylor & Francis Group

A CHAPMAN & HALL BOOK

Texts in Statistical Science

Linear Models with R

Second Edition

Julian J. Faraway

University of Bath

United Kingdom



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group an **informa** business
A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2015 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Version Date: 20140505

International Standard Book Number-13: 978-1-4398-8734-9 (eBook - PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Contents

Preface	xi
1 Introduction	1
1.1 Before You Start	1
1.2 Initial Data Analysis	2
1.3 When to Use Linear Modeling	7
1.4 History	8
2 Estimation	13
2.1 Linear Model	13
2.2 Matrix Representation	14
2.3 Estimating β	15
2.4 Least Squares Estimation	16
2.5 Examples of Calculating $\hat{\beta}$	17
2.6 Example	17
2.7 QR Decomposition	20
2.8 Gauss–Markov Theorem	22
2.9 Goodness of Fit	23
2.10 Identifiability	26
2.11 Orthogonality	28
3 Inference	33
3.1 Hypothesis Tests to Compare Models	33
3.2 Testing Examples	35
3.3 Permutation Tests	40
3.4 Sampling	42
3.5 Confidence Intervals for β	43
3.6 Bootstrap Confidence Intervals	46
4 Prediction	51
4.1 Confidence Intervals for Predictions	51
4.2 Predicting Body Fat	52
4.3 Autoregression	54
4.4 What Can Go Wrong with Predictions?	56

5	Explanation	59
5.1	Simple Meaning	59
5.2	Causality	61
5.3	Designed Experiments	62
5.4	Observational Data	63
5.5	Matching	65
5.6	Covariate Adjustment	68
5.7	Qualitative Support for Causation	69
6	Diagnostics	73
6.1	Checking Error Assumptions	73
6.1.1	Constant Variance	73
6.1.2	Normality	78
6.1.3	Correlated Errors	81
6.2	Finding Unusual Observations	83
6.2.1	Leverage	83
6.2.2	Outliers	85
6.2.3	Influential Observations	89
6.3	Checking the Structure of the Model	92
6.4	Discussion	96
7	Problems with the Predictors	99
7.1	Errors in the Predictors	99
7.2	Changes of Scale	103
7.3	Collinearity	106
8	Problems with the Error	113
8.1	Generalized Least Squares	113
8.2	Weighted Least Squares	116
8.3	Testing for Lack of Fit	119
8.4	Robust Regression	123
8.4.1	M-Estimation	123
8.4.2	Least Trimmed Squares	126
9	Transformation	133
9.1	Transforming the Response	133
9.2	Transforming the Predictors	137
9.3	Broken Stick Regression	137
9.4	Polynomials	139
9.5	Splines	141
9.6	Additive Models	144
9.7	More Complex Models	145

10 Model Selection	149
10.1 Hierarchical Models	150
10.2 Testing-Based Procedures	151
10.3 Criterion-Based Procedures	153
10.4 Summary	159
11 Shrinkage Methods	161
11.1 Principal Components	161
11.2 Partial Least Squares	172
11.3 Ridge Regression	174
11.4 Lasso	177
12 Insurance Redlining — A Complete Example	183
12.1 Ecological Correlation	183
12.2 Initial Data Analysis	185
12.3 Full Model and Diagnostics	188
12.4 Sensitivity Analysis	190
12.5 Discussion	194
13 Missing Data	197
13.1 Types of Missing Data	197
13.2 Deletion	198
13.3 Single Imputation	200
13.4 Multiple Imputation	202
14 Categorical Predictors	205
14.1 A Two-Level Factor	205
14.2 Factors and Quantitative Predictors	209
14.3 Interpretation with Interaction Terms	212
14.4 Factors With More Than Two Levels	213
14.5 Alternative Codings of Qualitative Predictors	219
15 One Factor Models	223
15.1 The Model	223
15.2 An Example	224
15.3 Diagnostics	227
15.4 Pairwise Comparisons	228
15.5 False Discovery Rate	230
16 Models with Several Factors	235
16.1 Two Factors with No Replication	235
16.2 Two Factors with Replication	239
16.3 Two Factors with an Interaction	243
16.4 Larger Factorial Experiments	246

17 Experiments with Blocks	251
17.1 Randomized Block Design	252
17.2 Latin Squares	256
17.3 Balanced Incomplete Block Design	259
A About R	265
Bibliography	267
Index	271

Preface

This is a book about linear models in Statistics. A linear model describes a quantitative response in terms of a linear combination of predictors. You can use a linear model to make predictions or explain the relationship between the response and the predictors. Linear models are very flexible and widely used in applications in physical science, engineering, social science and business. Linear models are part of the core of Statistics and understanding them well is crucial to a broader competence in the practice of statistics.

This is not an introductory textbook. You will need some basic prior knowledge of statistics as might be obtained in one or two courses at the university level. You will need to be familiar with essential ideas such as hypothesis testing, confidence intervals, likelihood and parameter estimation. You will also need to be competent in the mathematical methods of calculus and linear algebra. This is not a particularly theoretical book as I have preferred intuition over rigorous proof. Nevertheless, successful statistics requires an appreciation of the principles and it is my hope that the reader will absorb these through the many examples I present.

This book is written in three languages: English, Mathematics and R. My motivation in writing the first edition of this text was to combine these three seamlessly to allow coherent exposition of the practice of linear modeling. This was uncommon ten years ago but now this has become popular in the now large number of statistics books that integrate R. Clearly it is a method that works but it does require the reader to become somewhat fluent in R. This is not a book about learning R but like any foreign language, one becomes proficient by practicing it rather than by memorizing the dictionary. The reader is advised to look elsewhere for a basic introduction to R but should not hesitate to dive into this book and pick it up as you go. I shall try to help. See the appendix to get started.

The book website can be found at:

<http://people.bath.ac.uk/jjf23/LMR>

Ten years have passed since the first edition and the world of R has expanded enormously. Here I am taking the opportunity to fix some errors and make some improvements. This new edition makes several revisions:

1. I have reorganized the material on interpreting linear models to distinguish the main applications of prediction and explanation. Although these two share a common basis in theory, the modeling approach diverges, depending on the purpose of the analysis. In particular, I have introduced some elementary notions of causality into the chapter on explanation.

2. Several new topics have been added including the QR decomposition, splines, additive models, lasso, multiple imputation and false discovery rates. The chapter on statistical strategy from the first edition has gone but its contents have been redistributed throughout the revised edition.
3. I have made extensive use of the `ggplot2` graphics package in addition to base graphics. As with the first edition, I assume that the reader has a basic knowledge of R. Some of the R commands are quite complex. Sometimes I have explained the more difficult points but the reader will need to make some effort to fully understand these commands. This effort is a necessary part of the learning process.

My thanks to many past students and readers of the first edition whose comments and questions have helped me make many improvements to this edition. Thanks to the builders of R (R Core Team (2013)) who made all this possible.

Introduction

1.1 Before You Start

Statistics starts with a problem, proceeds with the collection of data, continues with the data analysis and finishes with conclusions. It is a common mistake of inexperienced statisticians to plunge into a complex analysis without paying attention to the objectives or even whether the data are appropriate for the proposed analysis. As Einstein said, the formulation of a problem is often more essential than its solution which may be merely a matter of mathematical or experimental skill.

To formulate the problem correctly, you must:

1. Understand the physical background. Statisticians often work in collaboration with others and need to understand something about the subject area. Regard this as an opportunity to learn something new rather than a chore.
2. Understand the objective. Again, often you will be working with a collaborator who may not be clear about what the objectives are. Beware of “fishing expeditions” — if you look hard enough, you will almost always find something, but that something may just be a coincidence.
3. Make sure you know what the client wants. You can often do quite different analyses on the same dataset. Sometimes statisticians perform an analysis far more complicated than the client really needed. You may find that simple descriptive statistics are all that are needed.
4. Put the problem into statistical terms. This is a challenging step and where irreparable errors are sometimes made. Once the problem is translated into the language of statistics, the solution is often routine. Difficulties with this step explain why artificial intelligence techniques have yet to make much impact in application to statistics. Defining the problem is hard to program. That a statistical method can read in and process the data is not enough. The results of an inapt analysis may be meaningless.

It is important to understand how the data were collected.

1. Are the data observational or experimental? Are the data a sample of convenience or were they obtained via a designed sample survey. How the data were collected has a crucial impact on what conclusions can be made.
2. Is there nonresponse? The data you do not see may be just as important as the data you do see.
3. Are there missing values? This is a common problem that is troublesome and time consuming to handle.

4. How are the data coded? In particular, how are the categorical variables represented?
5. What are the units of measurement?
6. Beware of data entry errors and other corruption of the data. This problem is all too common — almost a certainty in any real dataset of at least moderate size. Perform some data sanity checks.

1.2 Initial Data Analysis

This is a critical step that should always be performed. It is simple but it is vital. You should make numerical summaries such as means, standard deviations (SDs), maximum and minimum, correlations and whatever else is appropriate to the specific dataset. Equally important are graphical summaries. There is a wide variety of techniques to choose from. For one variable at a time, you can make boxplots, histograms, density plots and more. For two variables, scatterplots are standard while for even more variables, there are numerous good ideas for display including interactive and dynamic graphics. In the plots, look for outliers, data-entry errors, skewed or unusual distributions and structure. Check whether the data are distributed according to prior expectations.

Getting data into a form suitable for analysis by cleaning out mistakes and aberrations is often time consuming. It often takes more time than the data analysis itself. In this book, all the data will be ready to analyze, but you should realize that in practice this is rarely the case.

Let's look at an example. The National Institute of Diabetes and Digestive and Kidney Diseases conducted a study on 768 adult female Pima Indians living near Phoenix. The following variables were recorded: number of times pregnant, plasma glucose concentration at 2 hours in an oral glucose tolerance test, diastolic blood pressure (mmHg), triceps skin fold thickness (mm), 2-hour serum insulin ($\mu\text{U/ml}$), body mass index (weight in $\text{kg}/(\text{height in m}^2)$), diabetes pedigree function, age (years) and a test whether the patient showed signs of diabetes (coded zero if negative, one if positive). The data may be obtained from UCI Repository of machine learning databases at archive.ics.uci.edu/ml.

Before doing anything else, one should find out the purpose of the study and more about how the data were collected. However, let's skip ahead to a look at the data:

```
> data(pima, package="faraway")
> head(pima)
  pregnant glucose diastolic triceps insulin  bmi diabetes age test
1         6     148         72     35         0 33.6   0.627  50    1
2         1      85         66     29         0 26.6   0.351  31    0
3         8     183         64      0         0 23.3   0.672  32    1
4         1      89         66     23         94 28.1   0.167  21    0
5         0     137         40     35        168 43.1   2.288  33    1
6         5     116         74      0         0 25.6   0.201  30    0
```

The data(`pima`, `package="faraway"`) makes the data available from the `faraway` package. An alternative is to just type `require(faraway)` which makes all the data

and functions specific to this book available. You do need to install this package first as explained in Appendix A. If you get an error message about data not being found, it may be that you have forgotten to install the package.

The command `head(pima)` prints out the first six lines of the data frame. This is a good way to see what variables we have and what sort of values they take. You can type `pima` to see the whole data frame but 768 lines may be more than you want to examine.

We start with some numerical summaries:

```
> summary(pima)
  pregnant      glucose      diastolic      triceps
Min.   : 0.00    Min.   :  0    Min.   :  0.0    Min.   :  0.0
1st Qu.: 1.00    1st Qu.: 99    1st Qu.: 62.0   1st Qu.:  0.0
Median : 3.00    Median :117    Median : 72.0   Median :23.0
Mean   : 3.85    Mean   :121    Mean   : 69.1   Mean   :20.5
3rd Qu.: 6.00    3rd Qu.:140    3rd Qu.: 80.0   3rd Qu.:32.0
Max.   :17.00    Max.   :199    Max.   :122.0   Max.   :99.0

  insulin      bmi      diabetes      age
Min.   :  0.0    Min.   :  0.0    Min.   :0.078   Min.   :21.0
1st Qu.:  0.0    1st Qu.:27.3    1st Qu.:0.244   1st Qu.:24.0
Median : 30.5    Median :32.0    Median :0.372   Median :29.0
Mean   : 79.8    Mean   :32.0    Mean   :0.472   Mean   :33.2
3rd Qu.:127.2    3rd Qu.:36.6    3rd Qu.:0.626   3rd Qu.:41.0
Max.   :846.0    Max.   :67.1    Max.   :2.420   Max.   :81.0

  test
Min.   :0.000
1st Qu.:0.000
Median :0.000
Mean   :0.349
3rd Qu.:1.000
Max.   :1.000
```

The `summary()` command is a quick way to get the usual univariate summary information. At this stage, we are looking for anything unusual or unexpected, perhaps indicating a data-entry error. For this purpose, a close look at the minimum and maximum values of each variable is worthwhile. Starting with `pregnant`, we see a maximum value of 17. This is large, but not impossible. However, we then see that the next five variables have minimum values of zero. No blood pressure is not good for the health — something must be wrong. Let's look at the sorted values:

```
> sort(pima$diastolic)
 [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 [17]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 [33]  0  0  0  0  24  30  30  38  40  44  44  44  44  46  46  48  48
...etc...
```

We see that the first 35 values are zero. The description that comes with the data says nothing about it but it seems likely that the zero has been used as a missing value code. For one reason or another, the researchers did not obtain the blood pressures of 35 patients. In a real investigation, one would likely be able to question the researchers about what really happened. Nevertheless, this does illustrate the kind of misunderstanding that can easily occur. A careless statistician might overlook these presumed missing values and complete an analysis assuming that these were real ob-

served zeros. If the error was later discovered, they might then blame the researchers for using zero as a missing value code (not a good choice since it is a valid value for some of the variables) and not mentioning it in their data description. Unfortunately such oversights are not uncommon, particularly with datasets of any size or complexity. The statistician bears some share of responsibility for spotting these mistakes.

We set all zero values of the five variables to NA which is the missing value code used by R:

```
> pima$diastolic[pima$diastolic == 0] <- NA
> pima$glucose[pima$glucose == 0] <- NA
> pima$triceps[pima$triceps == 0] <- NA
> pima$insulin[pima$insulin == 0] <- NA
> pima$bmi[pima$bmi == 0] <- NA
```

The variable `test` is not quantitative but categorical. Such variables are also called *factors*. However, because of the numerical coding, this variable has been treated as if it were quantitative. It is best to designate such variables as factors so that they are treated appropriately. Sometimes people forget this and compute stupid statistics such as the “average zip code.”

```
> pima$test <- factor(pima$test)
> summary(pima$test)
 0    1
500 268
```

We now see that 500 cases were negative and 268 were positive. It is even better to use descriptive labels:

```
> levels(pima$test) <- c("negative", "positive")
> summary(pima)
```

pregnant		glucose		diastolic		triceps	
Min.	: 0.000	Min.	: 44.0	Min.	: 24.0	Min.	: 7.00
1st Qu.:	1.000	1st Qu.:	99.0	1st Qu.:	64.0	1st Qu.:	22.00
Median:	3.000	Median:	117.0	Median:	72.0	Median:	29.00
Mean:	3.845	Mean:	121.7	Mean:	72.4	Mean:	29.15
3rd Qu.:	6.000	3rd Qu.:	141.0	3rd Qu.:	80.0	3rd Qu.:	36.00
Max.	:17.000	Max.	:199.0	Max.	:122.0	Max.	: 99.00
		NA's	: 5.0	NA's	: 35.0	NA's	:227.00
insulin		bmi		diabetes		age	
Min.	: 14.00	Min.	:18.20	Min.	:0.0780	Min.	:21.00
1st Qu.:	76.25	1st Qu.:	27.50	1st Qu.:	0.2437	1st Qu.:	24.00
Median:	125.00	Median:	32.30	Median:	0.3725	Median:	29.00
Mean:	:155.55	Mean:	:32.46	Mean:	:0.4719	Mean:	:33.24
3rd Qu.:	:190.00	3rd Qu.:	:36.60	3rd Qu.:	:0.6262	3rd Qu.:	:41.00
Max.	:846.00	Max.	:67.10	Max.	:2.4200	Max.	:81.00
NA's	:374.00	NA's	:11.00				
test							
negative:500							
positive:268							

Now that we have cleared up the missing values and coded the data appropriately, we are ready to do some plots. Perhaps the most well-known univariate plot is the histogram:

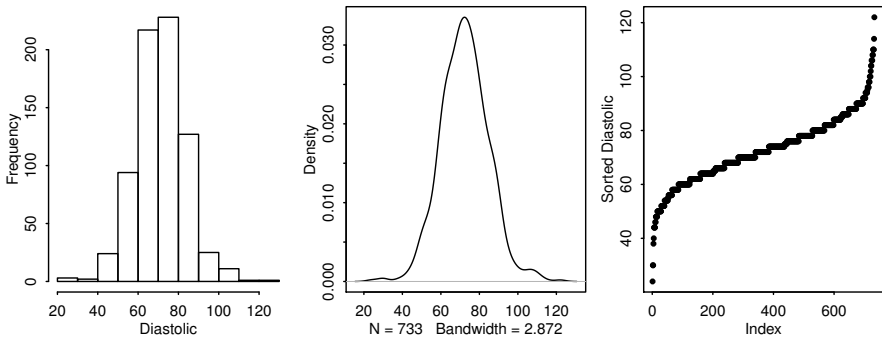


Figure 1.1 The first panel shows a histogram of the diastolic blood pressures, the second shows a kernel density estimate of the same, while the third shows an index plot of the sorted values.

```
> hist(pima$diastolic, xlab="Diastolic", main="")
```

as seen in the first panel of Figure 1.1. We see a bell-shaped distribution for the diastolic blood pressures centered around 70. The construction of a histogram requires the specification of the number of bins and their spacing on the horizontal axis. Some choices can lead to histograms that obscure some features of the data. R specifies the number and spacing of bins given the size and distribution of the data, but this choice is not foolproof and misleading histograms are possible. For this reason, some prefer to use kernel density estimates, which are essentially a smoothed version of the histogram (see Simonoff (1996) for a discussion of the relative merits of histograms and kernel estimates):

```
> plot(density(pima$diastolic, na.rm=TRUE), main="")
```

The kernel estimate may be seen in the second panel of Figure 1.1. We see that this plot avoids the distracting blockiness of the histogram. Another alternative is to simply plot the sorted data against its index:

```
> plot(sort(pima$diastolic), ylab="Sorted Diastolic")
```

The advantage of this is that we can see all the cases individually. We can see the distribution and possible outliers. We can also see the discreteness in the measurement of blood pressure — values are rounded to the nearest even number and hence we see the “steps” in the plot.

Now we show a couple of bivariate plots, as seen in Figure 1.2:

```
> plot(diabetes ~ diastolic, pima)
> plot(diabetes ~ test, pima)
```

First, we see the standard scatterplot showing two quantitative variables. Second, we see a side-by-side boxplot suitable for showing a quantitative with a qualitative variable.

The base graphics system in R is fine for producing standard plots. A useful

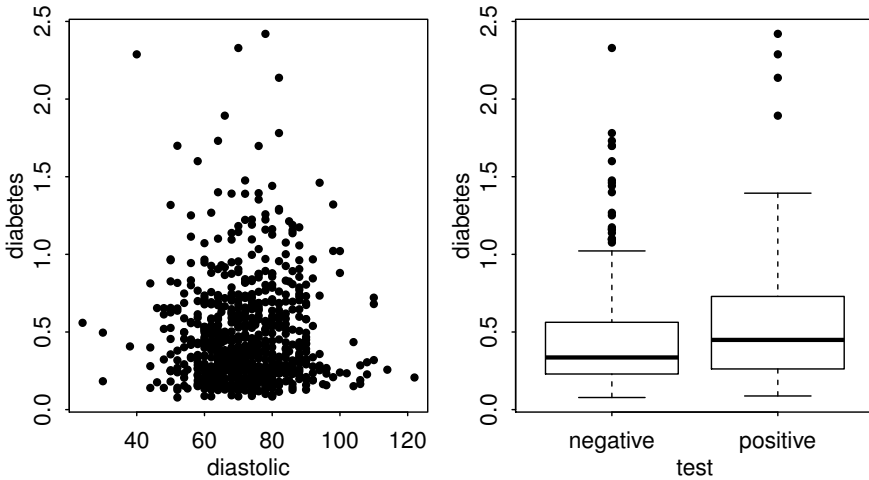


Figure 1.2 The first panel shows scatterplot of the diastolic blood pressures against diabetes function and the second shows boxplots of diastolic blood pressure broken down by test result.

alternative is provided by the `ggplot2` package which you will need to install as described in Appendix A. Here's how we can produce versions of the same plots:

```
> require(ggplot2)
> ggplot(pima, aes(x=diastolic))+geom_histogram()
> ggplot(pima, aes(x=diastolic))+geom_density()
> ggplot(pima, aes(x=diastolic, y=diabetes))+geom_point()
```

In each case, the first part of the command specifies where the data comes from (the data frame is `pima`) and an *aesthetic* using `aes`. The aesthetic specifies what you see such as position in the x or y direction or aspects such as shape or color. The second part of the command in each case is specifying the particular geometry for the plot. The results are shown in Figure 1.3. The relative advantage of `ggplot2` is more apparent in producing more complex plots involving more than just two variables. For example, we show two different ways that the varying `test` can be shown in the relationship between `diastolic` and `diabetes`:

```
> ggplot(pima, aes(x=diastolic, y=diabetes, shape=test))+geom_point()+
  theme(legend.position = "top", legend.direction = "horizontal")
> ggplot(pima, aes(x=diastolic, y=diabetes)) + geom_point(size=1) +
  facet_grid(~ test)
```

Notice how the first plot, shown in Figure 1.4, has a third element. A *theme* specifying an option regarding the appearance of the plot. In this case, we specify where the legend should appear. The second plot uses two panels. Sometimes this is the better option when crowded plots make different symbols hard to distinguish. I have also reduced the point size in the second plot to make this easier. Full details can be seen in Wickham (2009).

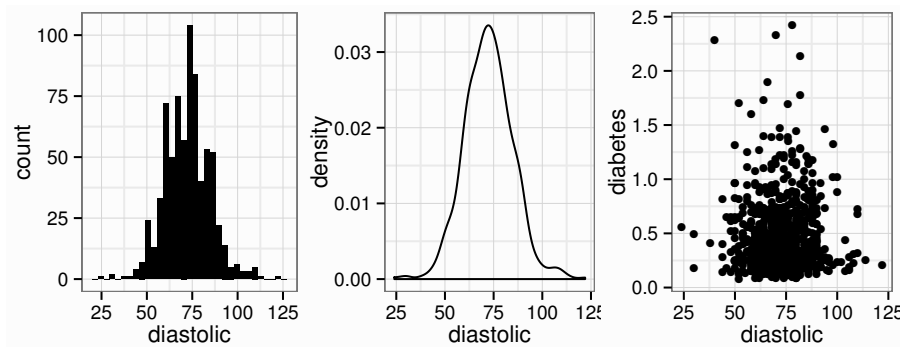


Figure 1.3 Plots of the pima data using ggplot2.

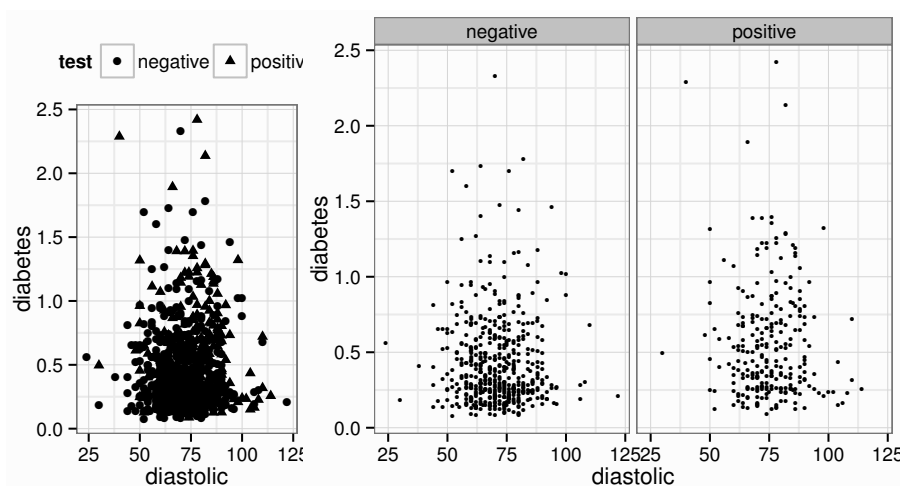


Figure 1.4 Two ways of a distinguishing a factor variable in a bivariate scatter plot.

Good graphics are vital in data analysis. They help you avoid mistakes and suggest the form of the modeling to come. They are also important in communicating your analysis to others. Many in your audience or readership will focus on the graphs. This is your best opportunity to get your message over clearly and without misunderstanding. In some cases, the graphics can be so convincing that the formal analysis becomes just a confirmation of what has already been seen.

1.3 When to Use Linear Modeling

Linear modeling is used for explaining or modeling the relationship between a single variable Y , called the *response*, *outcome*, *output* or *dependent* variable; and one or more *predictor*, *input*, *independent* or *explanatory* variables, X_1, \dots, X_p , where p is the number of predictors. We recommend you avoid using the words *independent*

and *dependent* variables for X and Y as these are easily confused with the broader meanings of terms. *Regression analysis* is another term used for linear modeling although regressions can also be nonlinear.

When $p = 1$, it is called *simple* regression but when $p > 1$ it is called *multiple* regression or sometimes *multivariate* regression. When there is more than one response, then it is called *multivariate multiple* regression or sometimes (confusingly) *multivariate* regression. We will not cover this in this book, although you can just do separate regressions on each Y .

The response must be a continuous variable, but the explanatory variables can be continuous, discrete or categorical, although we leave the handling of categorical explanatory variables to later in the book. Taking the example presented above, a regression with *diastolic* and *bmi* as X s and *diabetes* as Y would be a multiple regression involving only quantitative variables which we tackle first. A regression with *diastolic* and *test* as X s and *bmi* as Y would have one predictor that is quantitative and one that is qualitative, which we will consider later in Chapter 14 on *analysis of covariance*. A regression with *test* as X and *diastolic* as Y involves just qualitative predictors — a topic called *analysis of variance (ANOVA)*, although this would just be a simple two sample situation. A regression of *test* as Y on *diastolic* and *bmi* as predictors would involve a qualitative response. A *logistic regression* could be used, but this will not be covered in this book.

Regression analyses have two main objectives:

1. Prediction of future or unseen responses given specified values of the predictors.
2. Assessment of the effect of, or relationship between, explanatory variables and the response. We would like to infer causal relationships if possible.

You should be clear on the objective for the given data because some aspects of the resulting analysis may differ. Regression modeling can also be used in a descriptive manner to summarize the relationships between the variables. However, most end users of data have more specific questions in mind and want to direct the analysis toward a particular set of goals.

It is rare, except in a few cases in the precise physical sciences, to know (or even suspect) the true model. In most applications, the model is an empirical construct designed to answer questions about prediction or causation. It is usually not helpful to think of regression analysis as the search for some true model. The model is a means to an end, not an end in itself.

1.4 History

In the 18th century, accurate navigation was a difficult problem of commercial and military interest. Although, it is relatively easy to determine latitude from Polaris, also known as the North Star, finding longitude then was difficult. Various attempts were made to devise a method using astronomy. Contrary to popular supposition, the Moon does not always show the same face and moves such that about 60% of its surface is visible at some time.

Tobias Mayer collected data on the locations of various landmarks on Moon,

including the Manilius crater, as they moved relative to the earth. He derived an equation describing the motion of the moon (called *libration*) taking the form:

$$\text{arc} = \beta + \alpha \sin \text{ang} + \gamma \cos \text{ang}$$

He wished to obtain values for the three unknowns α , β and γ . The variables `arc`, `sinang` and `cosang` can be observed using a telescope. A full explanation of the story behind the data and the derivation of the equation can be found in Stigler (1986).

Since there are three unknowns, we need only three distinct observations of the set of three variables to find a unique solution for α , β and γ . Embarrassingly for Mayer, there were 27 sets of observations available. Astronomical measurements were naturally subject to some variation and so there was no solution that fit all 27 observations. Let's take a look at the first few lines of the data:

```
> data(manilius, package="faraway")
> head(manilius)
   arc sinang cosang group
1 13.167 0.8836 -0.4682    1
2 13.133 0.9996 -0.0282    1
3 13.200 0.9899  0.1421    1
4 14.250 0.2221  0.9750    3
5 14.700 0.0006  1.0000    3
6 13.017 0.9308 -0.3654    1
```

Mayer's solution was to divide the data into three groups so that observations within each group were similar in some respect. He then computed the sum of the variables within each group. We can also do this:

```
> (moon3 <- aggregate(manilius[,1:3], list(manilius$group), sum))
  Group.1   arc sinang cosang
1      1 118.13  8.4987 -0.7932
2      2 140.28 -6.1404  1.7443
3      3 127.53  2.9777  7.9649
```

Now there are just three equations in three unknowns to be solved. The solution is:

```
> solve(cbind(9, moon3$sinang, moon3$cosang), moon3$arc)
[1] 14.54459 -1.48982  0.13413
```

Hence the computed values of α , β and γ are -1.49, 14.5 and 0.134 respectively. One might question how Mayer selected his three groups, but this solution does not seem unreasonable.

Similar problems with more linear equations than unknowns continued to arise until 1805, when Adrien Marie Legendre published the method of least squares. Suppose we recognize that the equation is not exact and introduce an error term, ϵ :

$$\text{arc}_i = \beta + \alpha \sin \text{ang}_i + \gamma \cos \text{ang}_i + \epsilon_i$$

where $i = 1, \dots, 27$. Now we find α , β and γ that minimize the sum of the squared errors: $\sum \epsilon^2$. We will investigate this in much greater detail in the chapter to follow but for now we simply present the solution using R:

```
> lmod <- lm(arc ~ sinang + cosang, manilius)
> coef(lmod)
(Intercept)      sinang      cosang
 14.561624    -1.504581    0.091365
```

We observe that this solution is quite similar to Mayer's. The least squares solution is more satisfactory in that it requires no arbitrary division into groups. Carl Friedrich Gauss claimed to have devised the method of least squares earlier but without publishing it. At any rate, he did publish in 1809 showing that the method of least squares was, in some sense, optimal.

For many years, the method of least squares was confined to the physical sciences where it was used to resolve problems of overdetermined linear equations. The equations were derived from theory and least squares was used as a method to fit data to these equations to estimate coefficients like α , β and γ above. It was not until later in the 19th century that linear equations (or models) were suggested empirically from the data rather than from theories of physical science. This opened up the field to the social and life sciences.

Francis Galton, a nephew of Charles Darwin, was important in this extension of statistics into social science. He coined the term *regression to mediocrity* in 1875 from which the rather peculiar term *regression* derives. Let's see how this terminology arose by looking at one of the datasets he collected at the time on the heights of parents and children in Galton (1886). We load the `HistData` package of historical statistical datasets and plot some of the data as seen in Figure 1.5. You will need to install this package using `install.packages("HistData")` if you have not already done so.

```
> data(GaltonFamilies, package="HistData")
> plot(childHeight ~ midparentHeight, GaltonFamilies)
```

We see that `midparentHeight`, defined as the father's height plus 1.08 times the mother's height divided by two, is correlated with the `childHeight`, both in inches. Now we might propose a linear relationship between the two of the form:

$$\text{childHeight} = \alpha + \beta \text{midparentHeight} + \varepsilon$$

We can estimate α and β using R and plot the resulting fit as follows:

```
> lmod <- lm(childHeight ~ midparentHeight, GaltonFamilies)
> coef(lmod)
(Intercept) midparentHeight
 22.63624      0.63736
> abline(lmod)
```

For the simple case of a response y and a single predictor x , we can write the equation in the form:

$$\frac{y - \bar{y}}{SD_y} = r \frac{(x - \bar{x})}{SD_x}$$

where r is the correlation between x and y . The equation can be expressed in words as: the response in standard units is the correlation times the predictor in standard units. We can verify that this produces the same results as above by rearranging the equation in the form $y = \alpha + \beta x$ and computing the estimates:

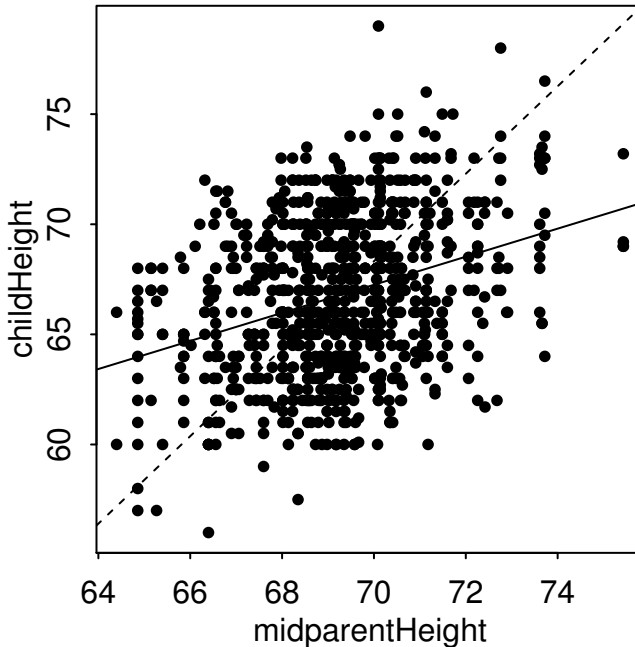


Figure 1.5 The height of child is plotted against a combined parental height defined as $(\text{father's height} + 1.08 \times \text{mother's height})/2$.

```
> (beta <- with(GaltonFamilies, cor(midparentHeight, childHeight) * sd
  (childHeight) / sd(midparentHeight)))
[1] 0.63736
> (alpha <- with(GaltonFamilies, mean(childHeight) - beta * mean(
  midparentHeight)))
[1] 22.636
```

Now one might naively expect that a child with parents who are, for example, one standard deviation above average in height, to also be one standard deviation above average in height, give or take. The supposition would set $r = 1$ in the equation and leads to a line which we compute and plot below:

```
> (beta1 <- with(GaltonFamilies, sd(childHeight) / sd(midparentHeight)
  ))
[1] 1.9859
> (alpha1 <- with(GaltonFamilies, mean(childHeight) - beta1 * mean(
  midparentHeight)))
[1] -70.689
> abline(alpha1, beta1, lty=2)
```

The resulting dashed line is added to Figure 1.5. The lines cross at the point of the averages. We can see that a child of tall parents is predicted by the least squares line to have a height which is above average but not quite as tall as the parents as the dashed line would have you believe. Similarly children of below average height

parents are predicted to have a height which is still below average but not quite as short as the parents. This is why Galton used the phrase “regression to mediocrity” and the phenomenon is sometimes called the regression effect.

This applies to any (x,y) situation like this. For example, in sports, an athlete may have a spectacular first season only to do not quite as well in the second season. Sports writers come up with all kinds of explanations for this but the regression effect is likely to be the unexciting cause. In the parents and children example, although it does predict that successive descendants in the family will come closer to the mean, it does not imply the same of the population in general since random fluctuations will maintain the variation, so no need to get too pessimistic about mediocrity! In many other applications of linear modeling, the regression effect is not of interest because different types of variables are measured. Unfortunately, we are now stuck with the rather gloomy word of regression thanks to Galton.

Regression methodology developed rapidly with the advent of high-speed computing. Just fitting a regression model used to require extensive hand calculation. As computing hardware has improved, the scope for analysis has widened. This has led to an extensive development in the methodology and the scale of problems that can be tackled.

Exercises

1. The dataset `teengamb` concerns a study of teenage gambling in Britain. Make a numerical and graphical summary of the data, commenting on any features that you find interesting. Limit the output you present to a quantity that a busy reader would find sufficient to get a basic understanding of the data.
2. The dataset `uswages` is drawn as a sample from the Current Population Survey in 1988. Make a numerical and graphical summary of the data as in the previous question.
3. The dataset `prostate` is from a study on 97 men with prostate cancer who were due to receive a radical prostatectomy. Make a numerical and graphical summary of the data as in the first question.
4. The dataset `sat` comes from a study entitled “Getting What You Pay For: The Debate Over Equity in Public School Expenditures.” Make a numerical and graphical summary of the data as in the first question.
5. The dataset `divusa` contains data on divorces in the United States from 1920 to 1996. Make a numerical and graphical summary of the data as in the first question.

Estimation

2.1 Linear Model

Let's start by defining what is meant by a linear model. Suppose we want to model the response Y in terms of three predictors, X_1 , X_2 and X_3 . One very general form for the model would be:

$$Y = f(X_1, X_2, X_3) + \varepsilon$$

where f is some unknown function and ε is the error in this representation. ε is additive in this instance, but could enter in some even more general form. Still, if we assume that f is a smooth, continuous function, that still leaves a very wide range of possibilities. Even with just three predictors, we typically will not have enough data to try to estimate f directly. So we usually have to assume that it has some more restricted form, perhaps linear as in:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$$

where β_i , $i = 0, 1, 2, 3$ are unknown *parameters*. Unfortunately this term is subject to some confusion as engineers often use the term *parameter* for what statisticians call the variables, Y , X_1 and so on. β_0 is called the *intercept* term.

Thus the problem is reduced to the estimation of four parameters rather than the infinite dimensional f . In a linear model the *parameters enter linearly* — the predictors themselves do not have to be linear. For example:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 \log X_2 + \beta_3 X_1 X_2 + \varepsilon$$

is a linear model, but:

$$Y = \beta_0 + \beta_1 X_1^{\beta_2} + \varepsilon$$

is not. Some relationships can be transformed to linearity — for example, $y = \beta_0 x_1^{\beta_1} \varepsilon$ can be linearized by taking logs. Linear models seem rather restrictive, but because the predictors can be transformed and combined in any way, they are actually very flexible. The term *linear* is often used in everyday speech as almost a synonym for simplicity. This gives the casual observer the impression that linear models can only handle small simple datasets. This is far from the truth — linear models can easily be expanded and modified to handle complex datasets. *Linear* is also used to refer to straight lines, but linear models can be curved. Truly nonlinear models are rarely absolutely necessary and most often arise from a theory about the relationships between the variables, rather than an empirical investigation.

Where do models come from? We distinguish several different sources:

1. Physical theory may suggest a model. For example, Hooke's law says that the extension of a spring is proportional to the weight attached. Models like these usually arise in the physical sciences and engineering.
2. Experience with past data. Similar data used in the past were modeled in a particular way. It is natural to see whether the same model will work with the current data. Models like these usually arise in the social sciences.
3. No prior idea exists — the model comes from an exploration of the data. We use skill and judgment to pick a model. Sometimes it does not work and we have to try again.

Models that derive directly from physical theory are relatively uncommon so that usually the linear model can only be regarded as an approximation to a complex reality. We hope it predicts well or explains relationships usefully but usually we do not believe it is exactly true. A good model is like a map that guides us to our destination. For the rest of this chapter, we will stay in the special world of Mathematics where all models are true.

2.2 Matrix Representation

We want a general solution to estimating the parameters of a linear model. We can find simple formulae for some special cases but to devise a method that will work in all cases, we need to use matrix algebra. Let's see how this can be done.

We start with some data where we have a response Y and, say, three predictors, X_1 , X_2 and X_3 . The data might be presented in tabular form like this:

$$\begin{array}{cccc} y_1 & x_{11} & x_{12} & x_{13} \\ y_2 & x_{21} & x_{22} & x_{23} \\ \dots & & \dots & \\ y_n & x_{n1} & x_{n2} & x_{n3} \end{array}$$

where n is the number of observations, or *cases*, in the dataset.

Given the actual data values, we may write the model as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i \quad i = 1, \dots, n.$$

It will be more convenient to put this in a matrix/vector representation. The regression equation is then written as:

$$y = X\beta + \epsilon$$

where $y = (y_1, \dots, y_n)^T$, $\epsilon = (\epsilon_1, \dots, \epsilon_n)^T$, $\beta = (\beta_0, \dots, \beta_3)^T$ and:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \dots & & \dots & \\ 1 & x_{n1} & x_{n2} & x_{n3} \end{pmatrix}$$

The column of ones incorporates the intercept term. One simple example is the *null model* where there is no predictor and just a mean $y = \mu + \varepsilon$:

$$\begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 \\ \dots \\ 1 \end{pmatrix} \mu + \begin{pmatrix} \varepsilon_1 \\ \dots \\ \varepsilon_n \end{pmatrix}$$

We can assume that $E\varepsilon = 0$ since if this were not so, we could simply absorb the nonzero expectation for the error into the mean μ to get a zero expectation.

2.3 Estimating β

The regression model, $y = X\beta + \varepsilon$, partitions the response into a systematic component $X\beta$ and a random component ε . We would like to choose β so that the systematic part explains as much of the response as possible. Geometrically speaking, the response lies in an n -dimensional space, that is, $y \in \mathbb{R}^n$ while $\beta \in \mathbb{R}^p$ where p is the number of parameters. If we include the intercept then p is the number of predictors plus one. It is easy to get confused as to whether p is the number of predictors or parameters, as different authors use different conventions, so be careful.

The problem is to find β so that $X\beta$ is as close to Y as possible. The best choice, the estimate $\hat{\beta}$, is apparent in the geometrical representation seen in Figure 2.1. $\hat{\beta}$ is, in this sense, the best estimate of β within the model space. The $\hat{\beta}$ values are sometimes called the regression coefficients. The response predicted by the model is $\hat{y} = X\hat{\beta}$ or Hy where H is an orthogonal projection matrix. The \hat{y} are called *predicted* or *fitted* values. The difference between the actual response and the predicted response is denoted by $\hat{\varepsilon}$ and is called the *residual*.

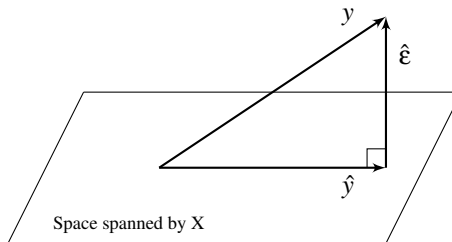


Figure 2.1 Geometrical representation of the estimation β . The data vector Y is projected orthogonally onto the model space spanned by X . The fit is represented by projection $\hat{y} = X\hat{\beta}$ with the difference between the fit and the data represented by the residual vector $\hat{\varepsilon}$.

The conceptual purpose of the model is to represent, as accurately as possible, something complex, y , which is n -dimensional, in terms of something much simpler, the model, which is p -dimensional. Thus if our model is successful, the structure in the data should be captured in those p dimensions, leaving just random variation in the residuals which lie in an $(n - p)$ -dimensional space. We have:

$$\begin{array}{rclcl} \text{Data} = & \text{Systematic Structure} & + & \text{Random Variation} \\ n \text{ dimensions} = & p \text{ dimensions} & + & (n - p) \text{ dimensions} \end{array}$$

2.4 Least Squares Estimation

The estimation of β can also be considered from a nongeometric point of view. We define the best estimate of β as the one which minimizes the sum of the squared errors:

$$\sum \varepsilon_i^2 = \varepsilon^T \varepsilon = (y - X\beta)^T (y - X\beta)$$

Differentiating with respect to β and setting to zero, we find that $\hat{\beta}$ satisfies:

$$X^T X \hat{\beta} = X^T y$$

These are called the *normal equations*. We can derive the same result using the geometric approach. Now provided $X^T X$ is invertible:

$$\begin{aligned} \hat{\beta} &= (X^T X)^{-1} X^T y \\ X \hat{\beta} &= X (X^T X)^{-1} X^T y \\ \hat{y} &= Hy \end{aligned}$$

$H = X(X^T X)^{-1} X^T$ is called the *hat matrix* and is the orthogonal projection of y onto the space spanned by X . H is useful for theoretical manipulations, but you usually do not want to compute it explicitly, as it is an $n \times n$ matrix which could be uncomfortably large for some datasets. The following useful quantities can now be represented using H .

The predicted or fitted values are $\hat{y} = Hy = X \hat{\beta}$ while the residuals are $\hat{\varepsilon} = y - X \hat{\beta} = y - \hat{y} = (I - H)y$. The residual sum of squares (RSS) is $\hat{\varepsilon}^T \hat{\varepsilon} = y^T (I - H)^T (I - H)y = y^T (I - H)y$.

Later, we will show that the least squares estimate is the best possible estimate of β when the errors ε are uncorrelated and have equal variance, i.e., $\text{var } \varepsilon = \sigma^2 I$. $\hat{\beta}$ is unbiased and has variance $(X^T X)^{-1} \sigma^2$ provided $\text{var } \varepsilon = \sigma^2 I$. Since $\hat{\beta}$ is a vector, its variance is a matrix.

We also need to estimate σ^2 . We find that $E \hat{\varepsilon}^T \hat{\varepsilon} = \sigma^2 (n - p)$, which suggests the estimator:

$$\hat{\sigma}^2 = \frac{\hat{\varepsilon}^T \hat{\varepsilon}}{n - p} = \frac{\text{RSS}}{n - p}$$

as an unbiased estimate of σ^2 . $n - p$ is called the *degrees of freedom* of the model. Sometimes you need the standard error for a particular component of $\hat{\beta}$ which can be picked out as $se(\hat{\beta}_{i-1}) = \sqrt{(X^T X)^{-1}_{ii}} \hat{\sigma}$.

2.5 Examples of Calculating $\hat{\beta}$

In a few simple models, it is possible to derive explicit formulae for $\hat{\beta}$:

1. When $y = \mu + \epsilon$, $X = \mathbf{1}$ and $\beta = \mu$ hence $X^T X = \mathbf{1}^T \mathbf{1} = n$ so:

$$\hat{\beta} = (X^T X)^{-1} X^T y = \frac{1}{n} \mathbf{1}^T y = \bar{y}$$

2. Simple linear regression (one predictor):

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

$$\begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \dots \\ \epsilon_n \end{pmatrix}$$

We can now apply the formula but a simpler approach is to rewrite the equation as:

$$y_i = \overbrace{\beta_0}^{\beta'_0} + \beta_1 \bar{x} + \beta_1 (x_i - \bar{x}) + \epsilon_i$$

so now:

$$X = \begin{pmatrix} 1 & x_1 - \bar{x} \\ \dots & \dots \\ 1 & x_n - \bar{x} \end{pmatrix} \quad X^T X = \begin{pmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}$$

Next work through the rest of the calculation to reconstruct the familiar estimate, that is:

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x}) y_i}{\sum (x_i - \bar{x})^2}$$

In higher dimensions, it is usually not possible to find such explicit formulae for the parameter estimates unless $X^T X$ happens to be a simple form. So typically we need computers to fit such models. Regression has a long history, so in the time before computers became readily available, fitting even quite simple models was a tedious time consuming task. When computing was expensive, data analysis was limited. It was designed to keep calculations to a minimum and restrict the number of plots. This mindset remained in statistical practice for some time even after computing became widely and cheaply available. Now it is a simple matter to fit a multitude of models and make more plots than one could reasonably study. The challenge now for the analyst is to choose among these intelligently to extract the crucial information in the data.

2.6 Example

Now let's look at an example concerning the number of species found on the various Galápagos Islands. There are 30 cases (Islands) and seven variables in the dataset. We start by reading the data into R and examining it:

```

> data(gala, package="faraway")
> head(gala[, -2])
  Species Area Elevation Nearest Scruz Adjacent
Baltra      58 25.09      346     0.6   0.6    1.84
Bartolome   31  1.24      109     0.6  26.3  572.33
Caldwell     3  0.21      114     2.8  58.7   0.78
Champion    25  0.10       46     1.9  47.4   0.18
Coamano      2  0.05       77     1.9   1.9  903.82
Daphne.Major 18  0.34      119     8.0   8.0   1.84

```

The variables are Species — the number of species found on the island, Area — the area of the island (km²), Elevation — the highest elevation of the island (m), Nearest — the distance from the nearest island (km), Scruz — the distance from Santa Cruz Island (km), Adjacent — the area of the adjacent island (km²). We have omitted the second column (which has the number of endemic species) because we shall not use this alternative response variable in this analysis.

The data were presented by Johnson and Raven (1973) and also appear in Weisberg (1985). I have filled in some missing values for simplicity (see Chapter 13 for how this can be done). Fitting a linear model in R is done using the `lm()` command. Notice the syntax for specifying the predictors in the model. This is part of the *Wilkinson–Rogers* notation. In this case, since all the variables are in the `gala` data frame, we must use the `data=` argument:

```

> lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
             data=gala)
> summary(lmod)

Call:
lm(formula = Species ~ Area + Elevation + Nearest + Scruz +
    Adjacent, data = gala)

Residuals:
    Min       1Q   Median       3Q      Max
-111.68  -34.90   -7.86   33.46  182.58

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.06822    19.15420   0.37   0.7154
Area        -0.02394     0.02242  -1.07   0.2963
Elevation    0.31946     0.05366   5.95 0.0000038 ***
Nearest      0.00914     1.05414   0.01   0.9932
Scruz       -0.24052     0.21540  -1.12   0.2752
Adjacent    -0.07480     0.01770  -4.23   0.0003 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61 on 24 degrees of freedom
Multiple R-squared:  0.766,    Adjusted R-squared:  0.717
F-statistic: 15.7 on 5 and 24 DF,  p-value: 6.84e-07

```

For my tastes, this output contains rather too much information. I have written an alternative called `sumary` which produces a shorter version of this. Since we will be looking at lot of regression output, the use of this version makes this book several pages shorter. Of course, if you prefer the above, feel free to add the extra “m” in the

function call. You will need to install and load my package if you want to use my version:

```
> require(faraway)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.06822    19.15420   0.37  0.7154
Area         -0.02394     0.02242  -1.07  0.2963
Elevation     0.31946     0.05366   5.95 0.0000038
Nearest       0.00914     1.05414   0.01  0.9932
Scruz        -0.24052     0.21540  -1.12  0.2752
Adjacent     -0.07480     0.01770  -4.23  0.0003

n = 30, p = 6, Residual SE = 60.975, R-Squared = 0.77
```

We can identify several useful quantities in this output. Other statistical packages tend to produce output quite similar to this. One useful feature of R is that it is possible to directly calculate quantities of interest. Of course, it is not necessary here because the `lm()` function does the job, but it is very useful when the statistic you want is not part of the prepackaged functions. First, we extract the X -matrix:

```
> x <- model.matrix(~ Area + Elevation + Nearest + Scruz + Adjacent,
                    gala)
```

and here is the response y :

```
> y <- gala$Species
```

Now let's construct $(X^T X)^{-1}$. `t()` does transpose and `%%%` does matrix multiplication. `solve(A)` computes A^{-1} while `solve(A, b)` solves $Ax = b$:

```
> xtxi <- solve(t(x) %*% x)
```

We can get $\hat{\beta}$ directly, using $(X^T X)^{-1} X^T y$:

```
> xtxi %*% t(x) %*% y
      [,1]
1      7.068221
Area   -0.023938
Elevation 0.319465
Nearest 0.009144
Scruz  -0.240524
Adjacent -0.074805
```

This is a very bad way to compute $\hat{\beta}$. It is inefficient and can be very inaccurate when the predictors are strongly correlated. Such problems are exacerbated by large datasets. A better, but not perfect, way is:

```
> solve(crossprod(x, x), crossprod(x, y))
      [,1]
1      7.068221
Area   -0.023938
Elevation 0.319465
Nearest 0.009144
Scruz  -0.240524
Adjacent -0.074805
```

where `crossprod(x, y)` computes $x^T y$. Here we get the same result as `lm()` because the data are well-behaved. In the long run, you are advised to use carefully programmed code such as found in `lm()` which uses the QR decomposition. To see more details, consult a text such as Thisted (1988) or read Section 2.7.

We can extract the regression quantities we need from the model object. Commonly used are `residuals()`, `fitted()`, `df.residual()` which gives the degrees of freedom, `deviance()` which gives the RSS and `coef()` which gives the $\hat{\beta}$. You can also extract other needed quantities by examining the model object and its summary:

```
> names(lmod)
[1] "coefficients" "residuals"      "effects"      "rank"
[5] "fitted.values" "assign"         "qr"           "df.residual"
[9] "xlevels"      "call"          "terms"        "model"
> lmodsum <- summary(lmod)
> names(lmodsum)
[1] "call"          "terms"          "residuals"     "coefficients"
[5] "aliased"       "sigma"          "df"            "r.squared"
[9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

We can estimate σ using the formula in the text above or extract it from the summary object:

```
> sqrt(deviance(lmod)/df.residual(lmod))
[1] 60.975
> lmodsum$sigma
[1] 60.975
```

We can also extract $(X^T X)^{-1}$ and use it to compute the standard errors for the coefficients. (`diag()` returns the diagonal of a matrix):

```
> xtxi <- lmodsum$cov.unscaled
> sqrt(diag(xtxi)) * 60.975
(Intercept)      Area      Elevation      Nearest      Scruz
 19.154139      0.022422      0.053663      1.054133      0.215402
  Adjacent
  0.017700
```

or get them from the summary object:

```
> lmodsum$coef[,2]
(Intercept)      Area      Elevation      Nearest      Scruz
 19.154198      0.022422      0.053663      1.054136      0.215402
  Adjacent
  0.017700
```

2.7 QR Decomposition

This section might be skipped unless you are interested in the actual calculation of $\hat{\beta}$ and related quantities. Any design matrix X can be written as:

$$X = Q \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix} = Q_f R$$

where Q is an $n \times n$ orthogonal matrix, that is $Q^T Q = Q Q^T = I$ and R is a $p \times p$ upper triangular matrix ($R_{ij} = 0$ for $i > j$). The $\mathbf{0}$ is an $(n - p) \times p$ matrix of zeroes while Q_f is the first p columns of Q .

The $RSS = (y - X\beta)^T (y - X\beta) = \|y - Z\beta\|^2$ where $\|\cdot\|$ is the Euclidean length of a vector. The matrix Q represents a rotation and does not change length. Hence:

$$RSS = \|Q^T y - Q^T X\beta\|^2 = \left\| \begin{pmatrix} f \\ r \end{pmatrix} - \begin{pmatrix} R \\ \mathbf{0} \end{pmatrix} \beta \right\|^2$$

where $\begin{pmatrix} f \\ r \end{pmatrix} = Q^T y$ for vector f of length p and vector r of length $n - p$. From this we see:

$$RSS = \|f - R\beta\|^2 + \|r\|^2$$

which can be minimized by setting β so that $R\beta = f$.

Let's see how this works for the Galápagos data. First we compute the QR decomposition:

```
> qrx <- qr(x)
```

The components of the decomposition must be extracted by other functions. For example, we can extract the Q matrix using `qr.Q`:

```
> dim(qr.Q(qrx))
[1] 30 6
```

Notice that we do not need the whole $n \times n$ matrix for the computation. The first p columns suffice so `qr.q` returns what we call Q_f . We can compute f :

```
> (f <- t(qr.Q(qrx)) %*% y)
      [,1]
[1,] -466.8422
[2,]  381.4056
[3,]  256.2505
[4,]   5.4076
[5,] -119.4983
[6,]  257.6944
```

Solving $R\beta = f$ is easy because of the triangular form of R . We use the method of backsubstitution:

```
> backsolve(qr.R(qrx), f)
      [,1]
[1,]  7.068221
[2,] -0.023938
[3,]  0.319465
[4,]  0.009144
[5,] -0.240524
[6,] -0.074805
```

where the results match those seen previously.

2.8 Gauss–Markov Theorem

$\hat{\beta}$ is a plausible estimator, but there are alternatives. Nonetheless, there are three good reasons to use least squares:

1. It results from an orthogonal projection onto the model space. It makes sense geometrically.
2. If the errors are independent and identically normally distributed, it is the maximum likelihood estimator. Loosely put, the maximum likelihood estimate is the value of β that maximizes the probability of the data that was observed.
3. The Gauss–Markov theorem states that $\hat{\beta}$ is the best linear unbiased estimate (BLUE).

To understand the Gauss–Markov theorem we first need to understand the concept of an *estimable function*. A linear combination of the parameters $\psi = c^T\beta$ is estimable if and only if there exists a linear combination $a^T y$ such that:

$$Ea^T y = c^T \beta \quad \forall \beta$$

Estimable functions include predictions of future observations, which explains why they are well worth considering. If X is of full rank, then all linear combinations are estimable.

Suppose $E\varepsilon = 0$ and $\text{var } \varepsilon = \sigma^2 I$. Suppose also that the structural part of the model, $EY = X\beta$ is correct. (Clearly these are big assumptions and so we will address the implications of this later.) Let $\psi = c^T\beta$ be an estimable function; then the Gauss–Markov theorem states that in the class of all unbiased linear estimates of ψ , $\hat{\psi} = c^T\hat{\beta}$ has the minimum variance and is unique.

We prove this theorem. Suppose $a^T y$ is some unbiased estimate of $c^T\beta$ so that:

$$\begin{aligned} Ea^T y &= c^T \beta & \forall \beta \\ a^T X\beta &= c^T \beta & \forall \beta \end{aligned}$$

which means that $a^T X = c^T$. This implies that c must be in the range space of X^T which in turn implies that c is also in the range space of $X^T X$ which means there exists a λ such that $c = X^T X\lambda$ so:

$$c^T \hat{\beta} = \lambda^T X^T X \hat{\beta} = \lambda^T X^T y$$

Now we can show that the least squares estimator has the minimum variance — pick an arbitrary estimate $a^T y$ and compute its variance:

$$\begin{aligned} \text{var}(a^T y) &= \text{var}(a^T y - c^T \hat{\beta} + c^T \hat{\beta}) \\ &= \text{var}(a^T y - \lambda^T X^T y + c^T \hat{\beta}) \\ &= \text{var}(a^T y - \lambda^T X^T y) + \text{var}(c^T \hat{\beta}) + 2\text{cov}(a^T y - \lambda^T X^T y, \lambda^T X^T y) \end{aligned}$$

but

$$\begin{aligned} \text{cov}(a^T y - \lambda^T X^T y, \lambda^T X^T y) &= (a^T - \lambda^T X^T) \sigma^2 I X \lambda \\ &= (a^T X - \lambda^T X^T X) \sigma^2 I \lambda \\ &= (c^T - c^T) \sigma^2 I \lambda = 0 \end{aligned}$$

so

$$\text{var}(a^T y) = \text{var}(a^T y - \lambda^T X^T y) + \text{var}(c^T \hat{\beta})$$

Now since variances cannot be negative, we see that:

$$\text{var}(a^T y) \geq \text{var}(c^T \hat{\beta})$$

In other words, $c^T \hat{\beta}$ has minimum variance. It now remains to show that it is unique. There will be equality in the above relationship if $\text{var}(a^T y - \lambda^T X^T y) = 0$ which would require that $a^T - \lambda^T X^T = 0$ which means that $a^T y = \lambda^T X^T y = c^T \hat{\beta}$. So equality occurs only if $a^T y = c^T \hat{\beta}$ so the estimator is unique. This completes the proof.

The Gauss–Markov theorem shows that the least squares estimate $\hat{\beta}$ is a good choice, but it does require that the errors are uncorrelated and have equal variance. Even if the errors behave, but are nonnormal, then nonlinear or biased estimates may work better. So this theorem does not tell one to use least squares all the time; it just strongly suggests it unless there is some strong reason to do otherwise. Situations where estimators other than ordinary least squares should be considered are:

1. When the errors are correlated or have unequal variance, generalized least squares should be used. See Section 8.1.
2. When the error distribution is long-tailed, then robust estimates might be used. Robust estimates are typically not linear in y . See Section 8.4.
3. When the predictors are highly correlated (collinear), then biased estimators such as ridge regression might be preferable. See Chapter 11.

2.9 Goodness of Fit

It is useful to have some measure of how well the model fits the data. One common choice is R^2 , the so-called *coefficient of determination* or *percentage of variance explained*:

$$R^2 = 1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum(y_i - \bar{y})^2} = 1 - \frac{\text{RSS}}{\text{Total SS(Corrected for Mean)}}$$

Its range is $0 \leq R^2 \leq 1$ — values closer to 1 indicating better fits. For simple linear regression $R^2 = r^2$ where r is the correlation between x and y . An equivalent definition is:

$$R^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

or

$$R^2 = \text{cor}^2(\hat{y}, y)$$

The graphical intuition behind R^2 is seen in Figure 2.2. Suppose you want to predict y . If you do not know x , then your best prediction is \bar{y} , but the variability in this prediction is high. If you do know x , then your prediction will be given by the regression fit. This prediction will be less variable provided there is some

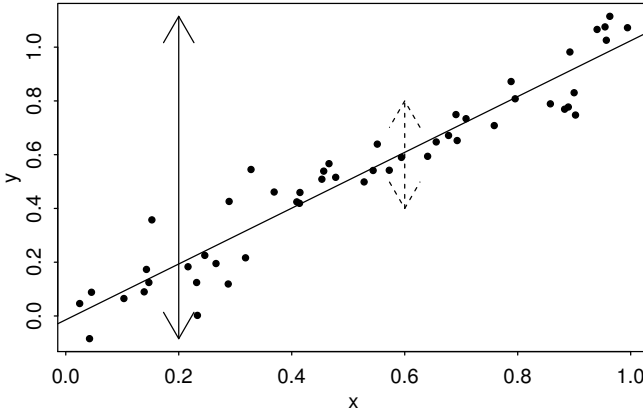


Figure 2.2 When x is not known, the best predictor of y is \bar{y} and the variation is denoted by the dotted line. When x is known, we can predict y more accurately by the solid line. R^2 is related to the ratio of these two variances.

relationship between x and y . R^2 is one minus the ratio of the sum of squares for these two predictions. Thus for perfect predictions the ratio will be zero and R^2 will be one.

Some care is necessary if there is no intercept in your model. The denominator in the first definition of R^2 has a null model with an intercept in mind when the sum of squares is calculated. Unfortunately, R uses this definition and will give a misleadingly high R^2 . If you must have an R^2 use the $\text{cor}^2(\hat{y}, y)$ definition when there is no intercept.

What is a good value of R^2 ? It depends on the area of application. In the biological and social sciences, variables tend to be more weakly correlated and there is a lot of noise. We would expect lower values for R^2 in these areas — a value of, say, 0.6 might be considered good. In physics and engineering, where most data come from closely controlled experiments, we typically expect to get much higher R^2 s and a value of 0.6 would be considered low. Some experience with the particular area is necessary for you to judge your R^2 s well.

It is a mistake to rely on R^2 as a sole measure of fit. In Figure 2.3, we see some simulated datasets where the R^2 is around 0.65 for a linear fit in all four cases. The first plot on the upper left shows an ordinary relationship while on the right the variation in x is smaller as is the residual variation. However, predictions (within the range of x) would have less variation in the second case. On the lower left, the fit looks good except for two outliers demonstrating how sensitive R^2 is to a few extreme values. On the lower right, we see that the true relationship is somewhat quadratic which shows us that R^2 doesn't tell us much about whether we have the right model.

An alternative measure of fit is $\hat{\sigma}$. This quantity is directly related to the standard errors of estimates of β and predictions. The advantage is that $\hat{\sigma}$ is measured in the

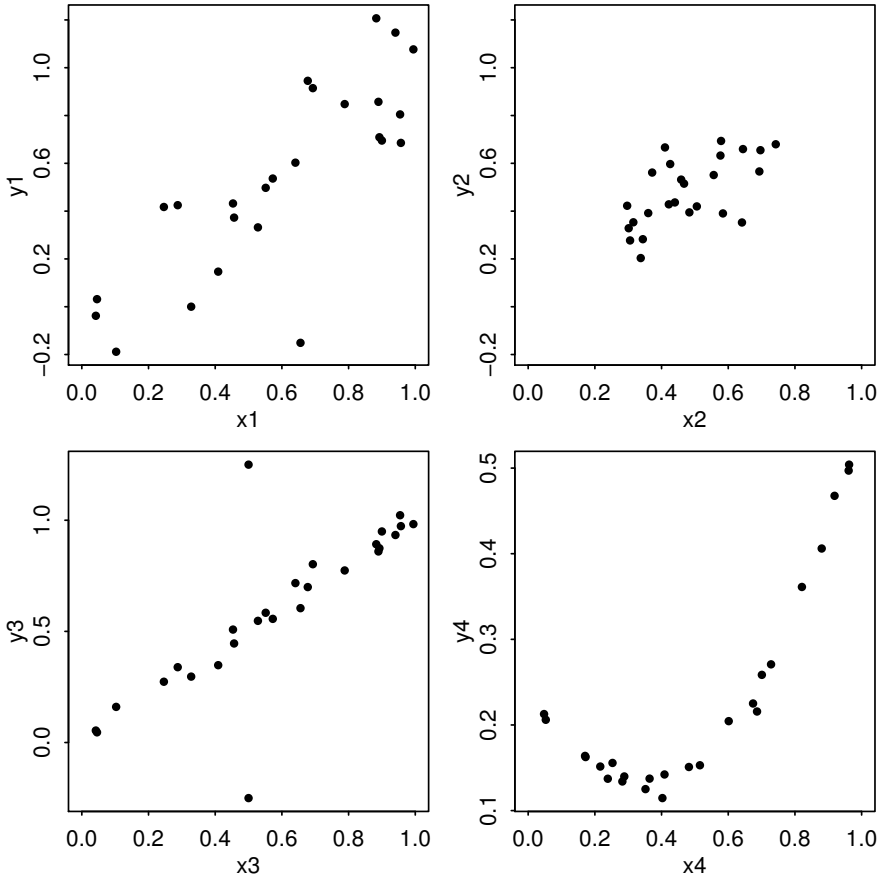


Figure 2.3 Four simulated datasets where R^2 is about 0.65. The plot on the upper left is well-behaved for R^2 . In the plot on the upper right, the residual variation is smaller than the first plot but the variation in x is also smaller so R^2 is about the same. In the plot on the lower left, the fit looks strong except for a couple of outliers while on the lower right, the relationship is quadratic.

units of the response and so may be directly interpreted in the context of the particular dataset. This may also be a disadvantage in that one must understand the practical significance of this measure whereas R^2 , being unitless, is easy to understand. The R regression summary returns both values and it is worth paying attention to both of them.

2.10 Identifiability

The least squares estimate is the solution to the normal equations:

$$X^T X \hat{\beta} = X^T y$$

where X is an $n \times p$ matrix. If $X^T X$ is singular and cannot be inverted, then there will be infinitely many solutions to the normal equations and $\hat{\beta}$ is at least partially unidentifiable. Unidentifiability will occur when X is not of full rank — that is when its columns are linearly dependent. With observational data, unidentifiability is usually caused by some oversight. Here are some examples:

1. A person's weight is measured both in pounds and kilos and both variables are entered into the model. One variable is just a multiple of the other.
2. For each individual we record the number of years of preuniversity education, the number of years of university education and also the total number of years of education and put all three variables into the model. There is an exact linear relation among the variables.
3. We have more variables than cases, that is, $p > n$. When $p = n$, we may perhaps estimate all the parameters, but with no degrees of freedom left to estimate any standard errors or do any testing. Such a model is called *saturated*. When $p > n$, then the model is sometimes called *supersaturated*. Such models are considered in large-scale screening experiments used in product design and manufacture and in bioinformatics where there are more genes than individuals tested, but there is no hope of uniquely estimating all the parameters in such a model. Different approaches are necessary.

Such problems can be avoided by paying attention. Identifiability is more of an issue in designed experiments. Consider a simple two-sample experiment, where the treatment observations are y_1, \dots, y_n and the controls are y_{n+1}, \dots, y_{m+n} . Suppose we try to model the response by an overall mean μ and group effects α_1 and α_2 :

$$y_j = \mu + \alpha_i + \varepsilon_j \quad i = 1, 2 \quad j = 1, \dots, m+n$$

$$\begin{pmatrix} y_1 \\ \dots \\ y_n \\ y_{n+1} \\ \dots \\ y_{m+n} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ & \dots & \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ \cdot & \cdot & \cdot \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \alpha_1 \\ \alpha_2 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \dots \\ \dots \\ \dots \\ \varepsilon_{m+n} \end{pmatrix}$$

Now although X has three columns, it has only rank two — $(\mu, \alpha_1, \alpha_2)$ are not identifiable and the normal equations have infinitely many solutions. We can solve this problem by imposing some constraints, $\mu = 0$ or $\alpha_1 + \alpha_2 = 0$, for example.

Statistics packages handle nonidentifiability differently. In the regression case above, some may return error messages and some may fit models because rounding error may remove the exact identifiability. In other cases, constraints may be applied but these may be different from what you expect. By default, R fits the largest identifiable model by removing variables in the reverse order of appearance in the model formula.

Here is an example. Suppose we create a new variable for the Galápagos dataset — the difference in area between the island and its nearest neighbor:

```
> gala$Adiff <- gala$Area -gala$Adjacent
```

and add that to the model:

```
> lmod <- lm(Species ~ Area+Elevation+Nearest+Scruz+Adjacent +Adiff,
gala)
> summary(lmod)

Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.06822    19.15420   0.37    0.7154
Area        -0.02394     0.02242  -1.07    0.2963
Elevation    0.31946     0.05366   5.95  0.0000038
Nearest      0.00914     1.05414   0.01    0.9932
Scruz       -0.24052     0.21540  -1.12    0.2752
Adjacent    -0.07480     0.01770  -4.23    0.0003

n = 30, p = 6, Residual SE = 60.975, R-Squared = 0.77
```

We get a message about a singularity because the rank of the design matrix X is six, which is less than its seven columns. In most cases, the cause of identifiability can be revealed with some thought about the variables, but, failing that, an eigendecomposition of $X^T X$ will reveal the linear combination(s) that gave rise to the unidentifiability — see Section 11.1.

Lack of identifiability is obviously a problem, but it is usually easy to identify and work around. More problematic are cases where we are close to unidentifiability. To demonstrate this, suppose we add a small random perturbation to the third decimal place of `Adiff` by adding a random variate from $U[-0.005, 0.005]$ where U denotes the uniform distribution. Random numbers are by nature random so results are not exactly reproducible. However, you can make the numbers come out the same every time by setting the seed on the random number generator using `set.seed()`. I have done this here so you will not wonder why your answers are not exactly the same as mine, but it is not strictly necessary.

```
> set.seed(123)
> Adiffe <- gala$Adiff+0.001*(runif(30)-0.5)
```

and now refit the model:

```
> lmod <- lm(Species ~ Area+Elevation+Nearest+Scruz +Adjacent+Adiffe,
gala)
> summary(lmod)

              Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.2964     19.4341   0.17    0.87
Area        -45122.9865  42583.3393  -1.06    0.30
Elevation     0.3130     0.0539   5.81  0.0000064
Nearest       0.3827     1.1090   0.35    0.73
Scruz        -0.2620     0.2158  -1.21    0.24
Adjacent     45122.8891  42583.3406   1.06    0.30
Adiffe       45122.9613  42583.3381   1.06    0.30

n = 30, p = 7, Residual SE = 60.820, R-Squared = 0.78
```

Notice that now all parameters are estimated, but the standard errors are very large because we cannot estimate them in a stable way. We set up this problem so we know the cause but in general we need to be able to identify such situations. We do this in Section 7.3.

2.11 Orthogonality

Orthogonality is a useful property because it allows us to more easily interpret the effect of one predictor without regard to another. Suppose we can partition X in two, $X = [X_1|X_2]$ such that $X_1^T X_2 = 0$. So now:

$$Y = X\beta + \varepsilon = X_1\beta_1 + X_2\beta_2 + \varepsilon$$

and

$$X^T X = \begin{pmatrix} X_1^T X_1 & X_1^T X_2 \\ X_2^T X_1 & X_2^T X_2 \end{pmatrix} = \begin{pmatrix} X_1^T X_1 & 0 \\ 0 & X_2^T X_2 \end{pmatrix}$$

which means:

$$\hat{\beta}_1 = (X_1^T X_1)^{-1} X_1^T y \quad \hat{\beta}_2 = (X_2^T X_2)^{-1} X_2^T y$$

Notice that $\hat{\beta}_1$ will be the same regardless of whether X_2 is in the model or not (and vice versa). So we can interpret the effect of X_1 without a concern for X_2 . Unfortunately, the decoupling is not perfect. Suppose we wish to test $H_0 : \beta_1 = 0$. We have $RSS/df = \hat{\sigma}^2$ that will be different depending on whether X_2 is included in the model or not, but the difference in F is not liable to be as large as in nonorthogonal cases.

If the covariance between vectors x_1 and x_2 is zero, then $\sum_j (x_{j1} - \bar{x}_1)(x_{j2} - \bar{x}_2) = 0$. This means that if we center the predictors, a covariance of zero implies orthogonality. As can be seen in the second example in Section 2.5, we can center the predictors without essentially changing the model provided we have an intercept term.

Orthogonality is a desirable property, but will only occur when X is chosen by the experimenter. It is a feature of a good design. In observational data, we do not have direct control over X and this is the source of many of the interpretational difficulties associated with nonexperimental data.

Here is an example of an experiment to determine the effects of column temperature, gas/liquid ratio and packing height in reducing the unpleasant odor of a chemical product that was sold for household use. Read the data in and display:

```
> data(odor, package="faraway")
> odor
  odor temp gas pack
1   66  -1  -1   0
2   39   1  -1   0
3   43  -1   1   0
4   49   1   1   0
5   58  -1   0  -1
6   17   1   0  -1
7   -5  -1   0   1
8  -40   1   0   1
```

9	65	0	-1	-1
10	7	0	1	-1
11	43	0	-1	1
12	-22	0	1	1
13	-31	0	0	0
14	-35	0	0	0
15	-26	0	0	0

The three predictors have been transformed from their original scale of measurement, for example, $\text{temp} = (\text{Fahrenheit}-80)/40$ so the original values of the predictor were 40, 80 and 120. The data is presented in John (1971) and give an example of a *central composite* design. We compute the covariance of the predictors:

```
> cov(odor[, -1])
      temp      gas      pack
temp 0.57143 0.00000 0.00000
gas  0.00000 0.57143 0.00000
pack 0.00000 0.00000 0.57143
```

The matrix is diagonal. Even if `temp` was measured in the original Fahrenheit scale, the matrix would still be diagonal, but the entry in the matrix corresponding to `temp` would change. Now fit a model, while asking for the correlation of the coefficients:

```
> lmod <- lm(odor ~ temp + gas + pack, odor)
> summary(lmod, cor=T)
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept)   15.2        9.3    1.63   0.13
temp          -12.1       12.7   -0.95   0.36
gas           -17.0       12.7   -1.34   0.21
pack          -21.4       12.7   -1.68   0.12

Residual standard error: 36 on 11 degrees of freedom
Multiple R-Squared: 0.334,    Adjusted R-squared: 0.152
F-statistic: 1.84 on 3 and 11 DF,  p-value: 0.199

Correlation of Coefficients:
      (Intercept) temp gas
temp 0.00
gas  0.00      0.00
pack 0.00      0.00 0.00
```

We see that, as expected, the pairwise correlation of all the coefficients is zero. Notice that the SEs for the coefficients are equal due to the balanced design. Now drop one of the variables:

```
> lmod <- lm(odor ~ gas + pack, odor)
> summary(lmod)
Coefficients:
      Estimate Std. Error t value Pr(>|t|)
(Intercept)   15.20        9.26    1.64   0.13
gas           -17.00       12.68   -1.34   0.20
pack          -21.37       12.68   -1.69   0.12

Residual standard error: 35.9 on 12 degrees of freedom
Multiple R-Squared: 0.279,    Adjusted R-squared: 0.159
F-statistic: 2.32 on 2 and 12 DF,  p-value: 0.141
```

The coefficients themselves do not change, but the residual SE does change slightly, which causes small changes in the SEs of the coefficients, t -statistics and p -values, but nowhere near enough to change our qualitative conclusions.

Exercises

- The dataset `teengamb` concerns a study of teenage gambling in Britain. Fit a regression model with the expenditure on gambling as the response and the sex, status, income and verbal score as predictors. Present the output.
 - What percentage of variation in the response is explained by these predictors?
 - Which observation has the largest (positive) residual? Give the case number.
 - Compute the mean and median of the residuals.
 - Compute the correlation of the residuals with the fitted values.
 - Compute the correlation of the residuals with the income.
 - For all other predictors held constant, what would be the difference in predicted expenditure on gambling for a male compared to a female?
- The dataset `uswages` is drawn as a sample from the Current Population Survey in 1988. Fit a model with weekly wages as the response and years of education and experience as predictors. Report and give a simple interpretation to the regression coefficient for years of education. Now fit the same model but with logged weekly wages. Give an interpretation to the regression coefficient for years of education. Which interpretation is more natural?
- In this question, we investigate the relative merits of methods for computing the coefficients. Generate some artificial data by:

```
> x <- 1:20
> y <- x+rnorm(20)
```

Fit a polynomial in x for predicting y . Compute $\hat{\beta}$ in two ways — by `lm()` and by using the direct calculation described in the chapter. At what degree of polynomial does the direct calculation method fail? (Note the need for the `I()` function in fitting the polynomial, that is, `lm(y ~ x + I(x^2))`).

- The dataset `prostate` comes from a study on 97 men with prostate cancer who were due to receive a radical prostatectomy. Fit a model with `lpsa` as the response and `lcavol` as the predictor. Record the residual standard error and the R^2 . Now add `lweight`, `svi`, `lbph`, `age`, `lcp`, `pgg45` and `gleason` to the model one at a time. For each model record the residual standard error and the R^2 . Plot the trends in these two statistics.
- Using the `prostate` data, plot `lpsa` against `lcavol`. Fit the regressions of `lpsa` on `lcavol` and `lcavol` on `lpsa`. Display both regression lines on the plot. At what point do the two lines intersect?
- Thirty samples of cheddar cheese were analyzed for their content of acetic acid, hydrogen sulfide and lactic acid. Each sample was tasted and scored by a panel of judges and the average taste score produced. Use the `cheddar` data to answer the following:

- (a) Fit a regression model with taste as the response and the three chemical contents as predictors. Report the values of the regression coefficients.
 - (b) Compute the correlation between the fitted values and the response. Square it. Identify where this value appears in the regression output.
 - (c) Fit the same regression model but without an intercept term. What is the value of R^2 reported in the output? Compute a more reasonable measure of the goodness of fit for this example.
 - (d) Compute the regression coefficients from the original fit using the QR decomposition showing your R code.
7. An experiment was conducted to determine the effect of four factors on the resistivity of a semiconductor wafer. The data is found in `wafer` where each of the four factors is coded as `-` or `+` depending on whether the low or the high setting for that factor was used. Fit the linear model $\text{resist} \sim x_1 + x_2 + x_3 + x_4$.
- (a) Extract the X matrix using the `model.matrix` function. Examine this to determine how the low and high levels have been coded in the model.
 - (b) Compute the correlation in the X matrix. Why are there some missing values in the matrix?
 - (c) What difference in resistance is expected when moving from the low to the high level of x_1 ?
 - (d) Refit the model without x_4 and examine the regression coefficients and standard errors? What stayed the the same as the original fit and what changed?
 - (e) Explain how the change in the regression coefficients is related to the correlation matrix of X .
8. An experiment was conducted to examine factors that might affect the height of leaf springs in the suspension of trucks. The data may be found in `truck`. The five factors in the experiment are set to `-` and `+` but it will be more convenient for us to use `-1` and `+1`. This can be achieved for the first factor by:

```
truck$B <- sapply(truck$B, function(x) ifelse(x=="-", -1, 1))
```

Repeat for the other four factors.

- (a) Fit a linear model for the height in terms of the five factors. Report on the value of the regression coefficients.
- (b) Fit a linear model using just factors B , C , D and E and report the coefficients. How do these compare to the previous question? Show how we could have anticipated this result by examining the X matrix.
- (c) Construct a new predictor called A which is set to $B+C+D+E$. Fit a linear model with the predictors A , B , C , D , E and O . Do coefficients for all six predictors appear in the regression summary? Explain.
- (d) Extract the model matrix X from the previous model. Attempt to compute $\hat{\beta}$ from $(X^T X)^{-1} X^T y$. What went wrong and why?
- (e) Use the QR decomposition method as seen in Section 2.7 to compute $\hat{\beta}$. Are the results satisfactory?
- (f) Use the function `qr.coef` to correctly compute $\hat{\beta}$.

Inference

In this chapter we show how to make hypothesis tests and construct confidence intervals. These inferential methods are the building blocks for drawing conclusions using models and data.

If you simply wish to estimate the parameters β , it is not essential to assume any distributional form for the errors ϵ . However, if we want to make any confidence intervals or perform any hypothesis tests using the most commonly used methods, we will need to do this. Now we are going to assume that the errors are normally distributed. Often this is quite reasonable but we will discuss later how we can check this assumption and how we might deal with data where the normality assumption is not justifiable.

If we have chosen to use least squares estimation, we should already have assumed that the errors are independent and identically distributed (i.i.d.) with mean 0 and variance σ^2 , so we have $\epsilon \sim N(0, \sigma^2 I)$. Now since $y = X\beta + \epsilon$, we have $y \sim N(X\beta, \sigma^2 I)$ which is a compact description of the regression model. From this we find, using the fact that linear combinations of normally distributed values are also normal, that:

$$\hat{\beta} = (X^T X)^{-1} X^T y \sim N(\beta, (X^T X)^{-1} \sigma^2)$$

3.1 Hypothesis Tests to Compare Models

Given several predictors for a response, we might wonder whether all are needed. Consider a larger model, Ω , and a smaller model, ω , which consists of a subset of the predictors that are in Ω . If there is not much difference in the fit, we would prefer the smaller model on the principle that simpler explanations are preferred. On the other hand, if the fit of the larger model is appreciably better, we will prefer it. We will take ω to represent the null hypothesis and Ω to represent the alternative. A geometrical view of the problem may be seen in Figure 3.1.

If $RSS_\omega - RSS_\Omega$ is small, then the fit of the smaller model is almost as good as the larger model and so we would prefer the smaller model on the grounds of simplicity. On the other hand, if the difference is large, then the superior fit of the larger model would be preferred. This suggests that something like:

$$\frac{RSS_\omega - RSS_\Omega}{RSS_\Omega}$$

would be a potentially good test statistic where the denominator is used for scaling purposes.

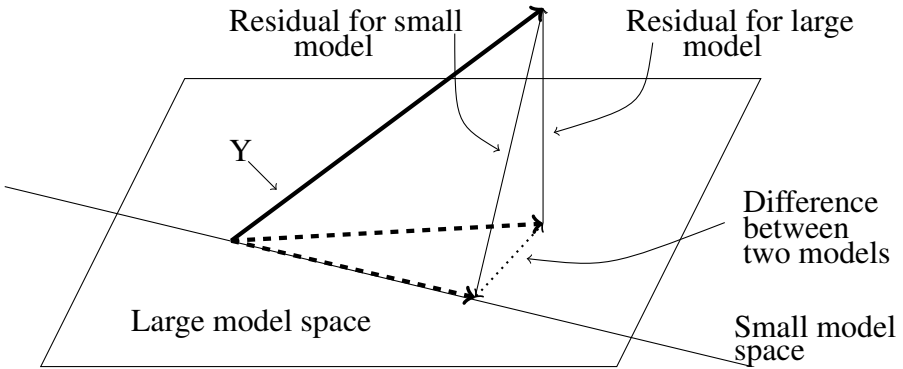


Figure 3.1 Geometric view of the comparison between big model, Ω , and small model, ω . The squared length of the residual vector for the big model is RSS_{Ω} while that for the small model is RSS_{ω} . By Pythagoras' theorem, the squared length of the vector connecting the two fits is $RSS_{\omega} - RSS_{\Omega}$. A small value for this indicates that the small model fits almost as well as the large model and thus might be preferred due to its simplicity.

As it happens, the same test statistic arises from the likelihood-ratio testing approach. We give an outline of the development: If $L(\beta, \sigma|y)$ is the likelihood function, then the likelihood-ratio statistic is:

$$\frac{\max_{\beta, \sigma \in \Omega} L(\beta, \sigma|y)}{\max_{\beta, \sigma \in \omega} L(\beta, \sigma|y)}$$

The test should reject if this ratio is too large. Working through the details, we find that for each model:

$$L(\hat{\beta}, \hat{\sigma}|y) \propto \hat{\sigma}^{-n}$$

which after some manipulation gives us a test that rejects if:

$$\frac{RSS_{\omega} - RSS_{\Omega}}{RSS_{\Omega}} > \text{a constant}$$

which is the same statistic suggested by the geometric view. Now suppose that the dimension (or number of parameters) of Ω is p and the dimension of ω is q , then we can use some more scaling to get an F -statistic which has an F -distribution under the null hypothesis:

$$F = \frac{(RSS_{\omega} - RSS_{\Omega}) / (p - q)}{RSS_{\Omega} / (n - p)} \sim F_{p-q, n-p}$$

Details of the derivation of this statistic may be found in more theoretically oriented texts such as Sen and Srivastava (1990).

Thus we would reject the null hypothesis if $F > F_{p-q, n-p}^{(\alpha)}$. The degrees of free-

dom of a model are (usually) the number of observations minus the number of parameters so this test statistic can also be written:

$$F = \frac{(RSS_{\omega} - RSS_{\Omega}) / (df_{\omega} - df_{\Omega})}{RSS_{\Omega} / df_{\Omega}}$$

where $df_{\Omega} = n - p$ and $df_{\omega} = n - q$. The same test statistic applies not just to when ω is a subset of Ω , but also to a subspace. This test is very widely used in regression and analysis of variance. When it is applied in different situations, the form of test statistic may be reexpressed in various ways. The beauty of this approach is you only need to know the general form. In any particular case, you just need to figure out which models represent the null and alternative hypotheses, fit them and compute the test statistic. The test is very versatile.

3.2 Testing Examples

Test of all the predictors

Are any of the predictors useful in predicting the response? Let the full model (Ω) be $y = X\beta + \varepsilon$ where X is a full-rank $n \times p$ matrix and the reduced model (ω) be $y = \mu + \varepsilon$. We estimate μ by \bar{y} . We write the null hypothesis as:

$$H_0 : \beta_1 = \dots \beta_{p-1} = 0$$

Now $RSS_{\Omega} = (y - X\hat{\beta})^T (y - X\hat{\beta}) = \hat{\varepsilon}^T \hat{\varepsilon}$, the residual sum of squares for the full model, while $RSS_{\omega} = (y - \bar{y})^T (y - \bar{y}) = TSS$, which is called the total sum of squares corrected for the mean. The F -statistic is:

$$F = \frac{(TSS - RSS) / (p - 1)}{RSS / (n - p)}$$

We would now refer to $F_{p-1, n-p}$ for a critical value or a p -value. Large values of F would indicate rejection of the null. Traditionally, the information in the above test is presented in an *analysis of variance* or ANOVA table. Most computer packages produce a variant on this. See Table 3.1 for a typical layout. It is not really necessary to specifically compute all the elements of the table. As the originator of the table, Fisher said in 1931, it is “nothing but a convenient way of arranging the arithmetic.” Since he had to do his calculations by hand, the table served a necessary purpose, but it is not essential now.

Source	Deg. of Freedom	Sum of Squares	Mean Square	F
Regression	$p - 1$	SS_{reg}	$SS_{reg} / (p - 1)$	F
Residual	$n - p$	RSS	$RSS / (n - p)$	
Total	$n - 1$	TSS		

Table 3.1 Analysis of variance table.

A failure to reject the null hypothesis is not the end of the game — you must still

investigate the possibility of nonlinear transformations of the variables and of outliers which may obscure the relationship. Even then, you may just have insufficient data to demonstrate a real effect, which is why we must be careful to say “fail to reject” the null rather than “accept” the null. It would be a mistake to conclude that no real relationship exists. This issue arises when a pharmaceutical company wishes to show that a proposed generic replacement for a brand-named drug is equivalent. It would not be enough in this instance just to fail to reject the null. A higher standard would be required.

When the null is rejected, this does not imply that the alternative model is the best model. We do not know whether all the predictors are required to predict the response or just some of them. Other predictors might also be added or existing predictors transformed or recombined. Either way, the overall F -test is just the beginning of an analysis and not the end.

Let’s illustrate this test using the Galápagos Islands dataset introduced in the last chapter. We fit the same model as before with the number of species as the response and the geographic variables as predictors.

```
> data(gala, package="faraway")
> lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
gala)
```

We can obtain the result of the test of all predictors by fitting the null model and the using the `anova` function:

```
> nullmod <- lm(Species ~ 1, gala)
> anova(nullmod, lmod)
Analysis of Variance Table

Model 1: Species ~ 1
Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
  Res.Df  RSS Df Sum of Sq   F Pr(>F)
1      29 381081
2      24  89231  5    291850 15.7 6.8e-07
```

We can see directly the result of the test of whether any of the predictors have significance in the model — that is, whether $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$. Since the p -value of 6.8×10^{-7} is so small, this null hypothesis is rejected.

Just for verification purposes, we can also do it directly using the F -testing formula:

```
> (rss0 <- deviance(nullmod))
[1] 381081
> (rss <- deviance(lmod))
[1] 89231
> (df0 <- df.residual(nullmod))
[1] 29
> (df <- df.residual(lmod))
[1] 24
> (fstat <- ((rss0-rss) / (df0-df)) / (rss/df))
[1] 15.699
> 1-pf(fstat, df0-df, df)
[1] 6.8379e-07
```

The information also appears in the final line of the `summary(lmod)` output as seen on page 18.

Testing one predictor

Can one particular predictor be dropped from the model? The null hypothesis is then $H_0 : \beta_i = 0$. Let Ω be the model with all the predictors of interest which has p parameters and let ω be the model with all the same predictors except predictor i . Let's test whether Area can be dropped from the full model by testing the hypothesis that the corresponding parameter is zero. Using the general method, we fit a model without Area and obtain this:

```
> lmods <- lm(Species ~ Elevation + Nearest + Scruz + Adjacent, gala)
> anova(lmods, lmod)
Analysis of Variance Table

Model 1: Species ~ Elevation + Nearest + Scruz + Adjacent
Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      25 93469
2      24 89231  1     4238  1.14  0.3
```

The p -value of 0.3 indicates that the null hypothesis cannot be rejected here.

An alternative approach is to use a t -statistic for testing the hypothesis:

$$t_i = \hat{\beta}_i / se(\hat{\beta}_i)$$

and check for significance using a t -distribution with $n - p$ degrees of freedom. It can be shown that t_i^2 is equal to the appropriate F -statistic computed using the method shown above. We can see the t -statistic and p -value in usual regression summary output:

```
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.06822    19.15420   0.37  0.7154
Area         -0.02394     0.02242  -1.07  0.2963
Elevation     0.31946     0.05366   5.95 0.0000038
Nearest       0.00914     1.05414   0.01  0.9932
Scruz        -0.24052     0.21540  -1.12  0.2752
Adjacent     -0.07480     0.01770  -4.23  0.0003

n = 30, p = 6, Residual SE = 60.975, R-Squared = 0.77
```

We can verify that this is indeed the same result. Of course, it is easier to obtain the summary output so we will use this method from now on.

It is important to be precise about what hypothesis is being tested. For example, consider another test of $H_0 : \beta_{\text{Area}} = 0$:

```
> summary(lm(Species ~ Area, gala))
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  63.7829    17.5244   3.64  0.00109
Area          0.0820     0.0197   4.16  0.00027

n = 30, p = 2, Residual SE = 91.732, R-Squared = 0.38
```

We see a very different p -value of 0.001 which changes the conclusion of the hypothesis test. The null is firmly rejected.

From this we can see that it is very important that we specify which other predictors are included in the models specifying the null and alternative. It is not sufficient simply to state the null hypothesis as $H_0 : \beta_{\text{Area}} = 0$.

Now we may ask what is the effect of area on the number of species. We postpone discussion until Chapter 5, but we can at least see that the question will have no simple answer.

Testing a pair of predictors

Suppose we were interested in whether the area of either the current island or the adjacent island had any relation to the response. This corresponds to a null hypothesis of $H_0 : \beta_{\text{Area}} = \beta_{\text{Adjacent}} = 0$ where we shall also specify that other three predictors are included in the model. We can test this by fitting a model without these two terms and constructing the F -test:

```
> lmods <- lm(Species ~ Elevation + Nearest + Scruz, gala)
> anova(lmods, lmod)
Analysis of Variance Table

Model 1: Species ~ Elevation + Nearest + Scruz
Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
  Res.Df  RSS Df Sum of Sq  F Pr(>F)
1      26 158292
2      24  89231  2     69060  9.29  0.001
```

The null hypothesis is rejected because the p -value is small. This tells us that such a simplification to remove these two predictors is not justifiable.

Now you may wonder whether we could have divined the same result from looking at the regression summary output where the corresponding p -values for the two terms are 0.3 and 0.0003. Immediately, we have the problem of making a decision based on two p -values rather than one. There is no simple way to combine these. Furthermore, each of these p -values corresponds to a test where the other predictor is included in the model. So in short, if you want to test two (or more) predictors, you need to use the single F -test. You cannot reliably use the two (in this case) t -tests.

Testing a subspace

Some tests cannot be expressed simply in terms of the inclusion or exclusion of subsets of predictors. Consider an example where we test whether the areas of the current and adjacent island can be added together and used in place of the two separate predictors. Such a test can be expressed as the hypothesis that:

$$H_0 : \beta_{\text{Area}} = \beta_{\text{Adjacent}}$$

The model corresponding to this null hypothesis represents a linear subspace of the full model. We can test this by specifying the null model and applying the F -test procedure:

```
> lmods <- lm(Species ~ I(Area+Adjacent) + Elevation + Nearest +
  Scruz, gala)
```

```
> anova(lmods, lmod)
Analysis of Variance Table

Model 1: Species ~ I(Area + Adjacent) + Elevation + Nearest + Scruz
Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      25 109591
2      24  89231  1    20360  5.48  0.028
```

The function `I()` ensures that the argument is evaluated (in this case actual addition) rather than interpreted as part of the model formula. The p -value of 0.028 indicates that the null can be rejected here and the proposed simplification to a single combined area predictor is not justifiable.

Another example of subspace testing occurs when we want to test whether a parameter can be set to a particular value. For example:

$$H_0 : \beta_{\text{Elevation}} = 0.5$$

This specifies a particular subspace of the full model. We can set a fixed term in the regression equation using an *offset*. We fit this model and compare it to the full:

```
> lmods <- lm(Species ~ Area+ offset(0.5*Elevation) + Nearest + Scruz
+ Adjacent, gala)
> anova(lmods, lmod)
Analysis of Variance Table

Model 1: Species ~ Area + offset(0.5 * Elevation) + Nearest + Scruz +
Adjacent
Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
  Res.Df  RSS Df Sum of Sq    F Pr(>F)
1      25 131312
2      24  89231  1    42081 11.3 0.0026
```

We see that the p -value is small and the null hypothesis is rejected. A simpler way to test such point hypotheses is to use a t -statistic:

$$t = (\hat{\beta} - c) / \text{se}(\hat{\beta})$$

where c is the point hypothesis. So in our example the statistic and corresponding p -value are:

```
> (tstat <- (0.31946-0.5)/0.05366)
[1] -3.3645
> 2*pt(tstat, 24)
[1] 0.0025722
```

We can see the p -value is the same as before and if we square the t -statistic

```
> tstat^2
[1] 11.32
```

we find we get the same F -value as above. This latter approach is preferred in practice since we do not need to fit two models but it is important to understand that it is equivalent to the result obtained using the general F -testing approach.

Tests we cannot do

We cannot test a non-linear hypothesis like $H_0 : \beta_j \beta_k = 1$ using the F -testing method. We would need to fit a nonlinear model and that lies beyond the scope of this book.

Also we cannot compare models that are not nested using an F -test. For example, if one model has `Area` and `Elevation` as predictors while another has `Area`, `Adjacent` and `Scruz`, we cannot use the F -test. We can ask which model is preferable. Methods for doing this will be discussed in Chapter 10.

A further difficulty can arise when the models we compare use different datasets. This can arise quite easily when values of some variables are missing as different models may use different cases depending on which are complete. F -tests are not directly possible here. Methods for dealing with this are presented in Chapter 13.

3.3 Permutation Tests

The tests we have considered thus far are based on the assumption of normal errors. Arguments based on the central limit theorem mean that even if the errors are not normal, inference based on the assumption of normality can be approximately correct provided the sample size is large enough. Unfortunately, it is not possible to say how large the sample has to be or how close to normality the error distribution has to be before the approximation is satisfactory. Permutation tests offer an alternative that needs no assumption of normality.

We can put a different interpretation on the hypothesis tests we are making. For the Galápagos dataset, we might suppose that if the number of species had no relation to the five geographic variables, then the observed response values would be randomly distributed between the islands without relation to the predictors. The F -statistic is a good measure of the association between the predictors and the response with larger values indicating stronger associations. We might then ask what the chance would be under this assumption that an F -statistic would be observed as large or larger than the one we actually observed. We could compute this exactly by computing the F -statistic for all possible ($n!$) permutations of the response variable and see what proportion exceeds the observed F -statistic. This is a permutation test. If the observed proportion is small, then we must reject the contention that the response is unrelated to the predictors. Curiously, this proportion is estimated by the p -value calculated in the usual way based on the assumption of normal errors thus saving us from the massive task of actually computing the regression on all those computations. See Freedman and Lane (1983) for details.

Let's see how we can apply the permutation test to the Galápagos data. We choose a model with just `Nearest` and `Scruz` so as to get a p -value for the F -statistic that is not too small (and therefore less interesting):

```
> lmod <- lm(Species ~ Nearest + Scruz, gala)
> lms <- summary(lmod)
```

We can extract the F -statistic in a way that will be convenient for later computation:


```

> lms$fstatistic
  value      numdf      dendif
0.60196  2.00000  27.00000
> 1-pf(lms$fstatistic[1],lms$fstatistic[2],lms$fstatistic[3])
  value
0.55493

```

The function `sample()` generates random permutations. We compute the F -statistic for 4000 randomly selected permutations and see what proportion exceeds the F -statistic for the original data:

```

> nreps <- 4000
> set.seed(123)
> fstats <- numeric(nreps)
> for(i in 1:nreps){
  lmods <- lm(sample(Species) ~ Nearest+Scruz, gala)
  fstats[i] <- summary(lmods)$fstat[1]
}
> mean(fstats > lms$fstat[1])
[1] 0.55825

```

This should take less than a minute on any relatively new computer. We could speed this process up noting that the X -matrix does not change in all 4000 regressions so we could avoid repeating a lot of the calculations by using the `update` function. However, when considering such an improvement, we should notice that the slow result will arrive well before we have even typed anything more efficient. Of course, this may not be true for examples with bigger data.

The function `set.seed` ensures that the random numbers used to generate the permutations will come out the same for you if you try this. If you don't do this, you will get a slightly different result each time because of the random selection of the permutations. That's alright — setting the seed is only necessary for exact reproducibility.

Our estimated p -value using the permutation test is 0.56, which is close to the normal theory-based value of 0.55. We could reduce variability in the estimation of the p -value simply by computing more random permutations. Since the permutation test does not depend on the assumption of normality, we might regard it as superior to the normal theory based value. In this case, the results are very similar and not close to any decision boundary. But if there was some crucial difference in the conclusion and there was some evidence of non-normal errors, then we would prefer the permutation-based test.

Tests involving just one predictor also fall within the permutation test framework. We permute that predictor rather than the response. Let's test the `Scruz` predictor in the model. We can extract the needed information from:

```

> summary(lmod)$coef[3,]
  Estimate Std. Error   t value   Pr(>|t|)
-0.44064   0.40253   -1.09467   0.28333

```

Now we perform 4000 permutations of `Scruz` and check what fraction of the t -statistics exceeds -1.09 in absolute value:

```

> tstats <- numeric(nreps)

```

```

> set.seed(123)
> for(i in 1:nreps){
  lmods <- lm(Species ~ Nearest+sample(Scruz), gala)
  tstats[i] <- summary(lmods)$coef[3,3]
}
> mean(abs(tstats) > abs(lms$coef[3,3]))
[1] 0.26775

```

The outcome is very similar to the observed normal-based p -value of 0.28. Again, in case of serious disagreement, we would prefer the permutation-based result.

The idea of permutation tests works well in conjunction with the principle of random allocation of units in designed experiments. When the values of X really have been randomly assigned to the experimental units which then produce response Y , it is easy to justify a permutation-based testing procedure to check whether there truly is any relation between X and Y . In practice, many will simply use the normal assumption-based tests but this can be done in the knowledge that the permutation tests will tend to agree, provided the assumption is justifiable.

3.4 Sampling

The method of data collection affects the conclusions we can draw. The mathematical model $Y = X\beta + \varepsilon$ describes how the response Y is generated. If we specified β , we could generate Y using a computer simulation, but how can this be related to real data?

For designed experiments, we can view nature as the computer generating the observed responses. We input X and record Y . With no cost or time constraints, we could repeat this as many times as we liked, but in practice we collect a sample of fixed size. Our inference then tells us something about the β underlying this natural process.

For observational studies, we envisage a finite *population* from which we draw the *sample* that is our data. We want to say something about the unknown population value of β , using estimated values $\hat{\beta}$ that are obtained from the sample data. We prefer that the data be a *simple random sample* of the population. We also assume that the size of the sample is a small fraction of the population size. We can also accommodate more complex random sampling designs, but this would require more complex inferential methods.

Sometimes, researchers may try to select a *representative* sample by hand. Quite apart from the obvious difficulties in doing this, the logic behind the statistical inference depends on the sample being random. This is not to say that such studies are worthless, but that it would be unreasonable to apply anything more than descriptive statistical techniques. Confidence in the conclusions from such data is necessarily suspect.

A sample of convenience is where the data are not collected according to a sampling design. In some cases, it may be reasonable to proceed as if the data were collected using a random mechanism. For example, suppose we take the first 400 people from the phone book whose names begin with the letter P. Provided there is no ethnic effect, it may be reasonable to consider this a random sample from the pop-

ulation defined by the entries in the phone book. Here we are assuming the selection mechanism is effectively random with respect to the objectives of the study. The data are as good as random. Other situations are less clear-cut and judgment will be required. Such judgments are easy targets for criticism. Suppose you are studying the behavior of alcoholics and advertise in the media for study subjects. It seems very likely that such a sample will be biased, perhaps in unpredictable ways. In cases such as this, a sample of convenience is clearly biased in which case conclusions must be limited to the sample itself. This situation reduces to the following case.

Sometimes, the sample is the complete population. In this case, one might argue that inference is not required since the population and sample values are one and the same. For the `gala` data, the sample is effectively the population or a large and biased proportion thereof. Permutation tests make it possible to give some meaning to the p -value when the sample is the population or for samples of convenience although one has to be clear that the conclusion applies only to the particular sample. Another approach that gives meaning to the p -value when the sample is the population involves the imaginative concept of “alternative worlds” where the sample or population at hand is supposed to have been randomly selected from parallel universes. This argument is more speculative.

3.5 Confidence Intervals for β

Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . They take the form:

$$\hat{\beta}_i \pm t_{n-p}^{(\alpha/2)} \text{se}(\hat{\beta})$$

Consider this model for the Galápagos data:

```
> lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
  gala)
> sumary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.06822    19.15420   0.37  0.7154
Area         -0.02394     0.02242  -1.07  0.2963
Elevation     0.31946     0.05366   5.95  0.000038
Nearest       0.00914     1.05414   0.01  0.9932
Scruz        -0.24052     0.21540  -1.12  0.2752
Adjacent     -0.07480     0.01770  -4.23  0.0003
n = 30, p = 6, Residual SE = 60.975, R-Squared = 0.77
```

We can construct individual 95% CIs for β_{Area} for which we need the 2.5% and 97.5% percentiles of the t -distribution with $30 - 6 = 24$ degrees of freedom:

```
> qt(0.975, 30-6)
[1] 2.0639
> -0.02394 + c(-1, 1) * 2.0639 * 0.02242
[1] -0.070213  0.022333
```

CIs have a duality with two-sided hypothesis tests. If the interval contains zero, this indicates that the null hypothesis $H_0 : \beta_{Area} = 0$ would not be rejected at the 5%

level. We can see from the summary that the p -value is 29.6% — greater than 5% — confirming this point. Indeed, any point null hypothesis lying within the interval would not be rejected.

The CI for $\beta_{Adjacent}$ is:

```
> -0.07480 + c(-1, 1) * 2.0639 * 0.01770
[1] -0.111331 -0.038269
```

Because zero is not in this interval, the null is rejected. Nevertheless, this CI is relatively wide in the sense that the upper limit is about three times larger than the lower limit. This means that we are not really that confident about what the exact effect of the area of the adjacent island on the number of species really is, even though the statistical significance means we are confident it is negative. A convenient way to obtain all the univariate intervals is:

```
> confint(lmod)
                2.5 %      97.5 %
(Intercept) -32.464101  46.600542
Area         -0.070216   0.022339
Elevation    0.208710    0.430219
Nearest      -2.166486    2.184774
Scruz        -0.685093    0.204044
Adjacent     -0.111336   -0.038273
```

The advantage of the confidence interval relative to the corresponding hypothesis test is that we get information about plausible ranges for the parameters. This is particularly valuable when the parameter is directly interpretable (for example, as the difference between two treatments).

The selection of a particular level of confidence level, say 95%, means we can only make tests at the 5% level. The hypothesis test approach does give a p -value which allows us to see how the acceptance or rejection of the null hypothesis depends on the choice of level. Even so it is dangerous to read too much into the relative sizes of p -values in determining the practical importance of a predictor because there is a temptation to view small p -values as indicating an important (rather than just statistically significant) effect. Confidence intervals are better in this respect because they tell us about the size of the effect.

If you are interested in more than one parameter, you can construct a $100(1 - \alpha)\%$ confidence region for β using:

$$(\hat{\beta} - \beta)^T X^T X (\hat{\beta} - \beta) \leq p \hat{\sigma}^2 F_{p, n-p}^{(\alpha)}$$

These regions are ellipsoidally shaped. Because these ellipsoids lie in higher dimensions, they cannot easily be visualized except for the two-dimensional case.

Let's see how these compare to the univariate confidence intervals. For example, we can construct the joint 95% confidence region for β_{Area} and $\beta_{Adjacent}$. First, we load in a package for drawing confidence ellipses (which is not part of base R and so may need to be installed using `install.packages("ellipse")`) and make the plot:

```
> require(ellipse)
> plot(ellipse(lmod, c(2, 6)), type="l", ylim=c(-0.13, 0))
> points(coef(lmod)[2], coef(lmod)[6], pch=19)
```

We have added the point of the least squares estimates which lies at the center of the ellipse as seen in Figure 3.2. We can also add the univariate confidence intervals for both dimensions seen as dotted lines:

```
> abline(v=confint(lmod)[2,],lty=2)
> abline(h=confint(lmod)[6,],lty=2)
```

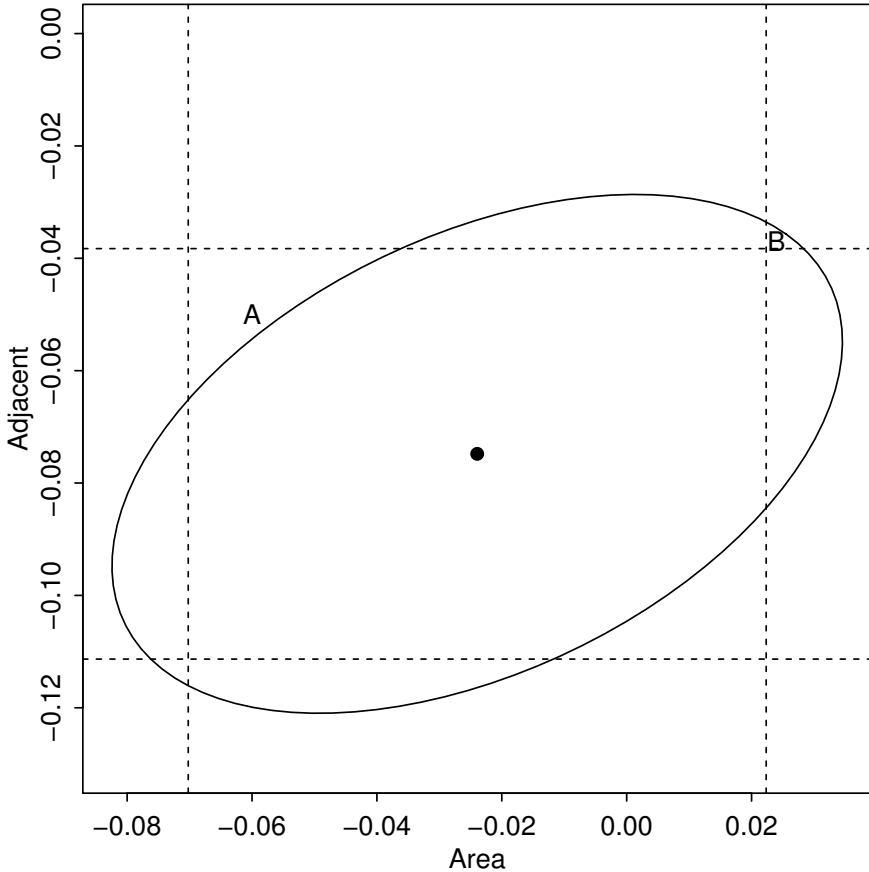


Figure 3.2 95% Confidence ellipse and intervals for β_{Area} and $\beta_{Adjacent}$. The central dot represents the point of the estimates.

We can determine the outcome of various hypotheses from the plot. The joint hypothesis $H_0 : \beta_{Area} = \beta_{Adjacent} = 0$ is rejected because the origin does not lie inside the ellipse. The hypothesis $H_0 : \beta_{Area} = 0$ is not rejected because zero does lie within the vertical dashed lines whereas the horizontal dashed lines do not encompass zero and so $H_0 : \beta_{Adjacent} = 0$ is rejected. We must also specify all the other three predictors are part of the model used to make these tests and confidence statements.

Now in this particular example, we find no disagreement between the two univariate tests and the bivariate tests. But imagine a situation where the origin lay in the region marked A on the plot. The point of the null hypothesis would lay outside the ellipse and so the bivariate test would reject while both univariate tests would not reject. The situation would reverse for a null hypothesis in the region marked B.

In both cases, the correct answer to the joint hypothesis is given by the position relative to the ellipse and not the lines. If you want to test multiple parameters, you need to use a joint testing procedure and not try to combine several univariate tests. In higher dimensions, confidence ellipses are not easily visualized so our example here is more of educational than practical value. Nevertheless, it should serve as a caution in interpreting a collection of univariate hypothesis tests or confidence intervals.

3.6 Bootstrap Confidence Intervals

The F -based and t -based confidence regions and intervals we have described depend on the assumption of normality. The bootstrap method provides a way to construct confidence statements without this assumption.

To understand how this method works, think about how we might determine the distribution of an estimator without using the mathematical methods of distribution theory. We could repeatedly generate artificial data from the true model, compute the estimate each time and gather the results to study the distribution. This technique, called simulation, is not available to us for real data, because we do not know the true model. Nevertheless, it will reveal the path to a practical solution.

Simulation

The idea is to sample from the known distribution and compute the estimate, repeating many times to find as good an estimate of the sampling distribution of the estimator as we need. For the regression problem, it is easiest to start with a sample from the error distribution since these are assumed to be independent and identically distributed:

1. Generate ε from the known error distribution.
2. Form $y = X\beta + \varepsilon$ from the known β and fixed X .
3. Compute $\hat{\beta}$.

We repeat these three steps many times. We can estimate the sampling distribution of $\hat{\beta}$ using the empirical distribution of the generated $\hat{\beta}$, which we can estimate as accurately as we please by simply running the simulation long enough. This technique is useful for a theoretical investigation of the properties of a proposed new estimator. We can see how its performance compares to other estimators. However, it is of no value for the actual data since we do not know the true error distribution and we do not know the true β .

Bootstrap

The bootstrap emulates the simulation procedure above except instead of sampling from the true model, it samples from the observed data. Remarkably, this

technique is often effective. It sidesteps the need for theoretical calculations that may be extremely difficult or even impossible. See Efron and Tibshirani (1993) for a book-length treatment of the topic. To see how the bootstrap method compares with simulation, we spell out the steps involved. In both cases, we consider X fixed.

The bootstrap method mirrors the simulation method, but uses quantities we do know. Instead of sampling from the population distribution, which we do not know in practice, we resample from the data:

1. Generate ϵ^* by sampling with replacement from $\hat{\epsilon}_1, \dots, \hat{\epsilon}_n$.
2. Form $y^* = X\hat{\beta} + \epsilon^*$.
3. Compute $\hat{\beta}^*$ from (X, y^*) .

This time, we use only quantities that we know. For very small n , it is possible to compute $\hat{\beta}^*$ for every possible sample from $\hat{\epsilon}_1, \dots, \hat{\epsilon}_n$, but usually we can only take as many samples as we have computing power available. This number of bootstrap samples can be as small as 50 if all we want is an estimate of the variance of our estimates but needs to be larger if confidence intervals are wanted.

To implement this, we need to be able to take a sample of residuals with replacement. `sample()` is good for generating random samples of indices:

```
> sample(10, rep=TRUE)
[1] 7 9 9 2 5 7 4 1 8 9
```

You will likely get a different result because the outcome is, by definition, random. To make sure we both get the same result, we have set the random number generator seed using `set.seed`. Here we have used 4000 replications which should take less than minute to compute. We set up a matrix `coefmat` to store the results. We need the residuals and fitted values from the model which we save as `resids` and `preds`. We repeatedly generate bootstrapped responses as `booty`. The update function is more efficient since we are only changing the response, not the predictors. The X -part of the computation does not need to be repeated every time.

```
> set.seed(123)
> nb <- 4000
> coefmat <- matrix(NA, nb, 6)
> resids <- residuals(lmod)
> preds <- fitted(lmod)
> for(i in 1:nb){
  booty <- preds + sample(resids, rep=TRUE)
  bmod <- update(lmod, booty ~ .)
  coefmat[i,] <- coef(bmod)
}
> colnames(coefmat) <- c("Intercept", colnames(gala[, 3:7]))
> coefmat <- data.frame(coefmat)
> apply(coefmat, 2, function(x) quantile(x, c(0.025, 0.975)))
```

	Intercept	Area	Elevation	Nearest	Scruz	Adjacent
2.5%	-24.534	-0.062786	0.22861	-1.7133	-0.60604	-0.104252
97.5%	41.439	0.018535	0.42110	2.0547	0.17248	-0.040944

The results are saved and formed into a data frame with named columns. We then compute the empirical 2.5% and 97.5% percentiles of the bootstrapped regression coefficients. These form the 95% bootstrap confidence intervals for β . Compare

these to those computed using normal theory earlier in the chapter. The position of zero, inside or outside the interval, is the same for both methods so qualitatively, the results are similar even though the numerical values differ somewhat.

We can plot the estimated densities of two of the coefficients along with the confidence intervals as seen in Figure 3.3.

```
> require(ggplot2)
> ggplot(coefmat, aes(x=Area)) + geom_density() + geom_vline(xint=c
  (-0.0628, 0.0185), lty=2)
> ggplot(coefmat, aes(x=Adjacent)) + geom_density() + geom_vline(xint=
  c(-0.104, -0.0409), lty=2)
```

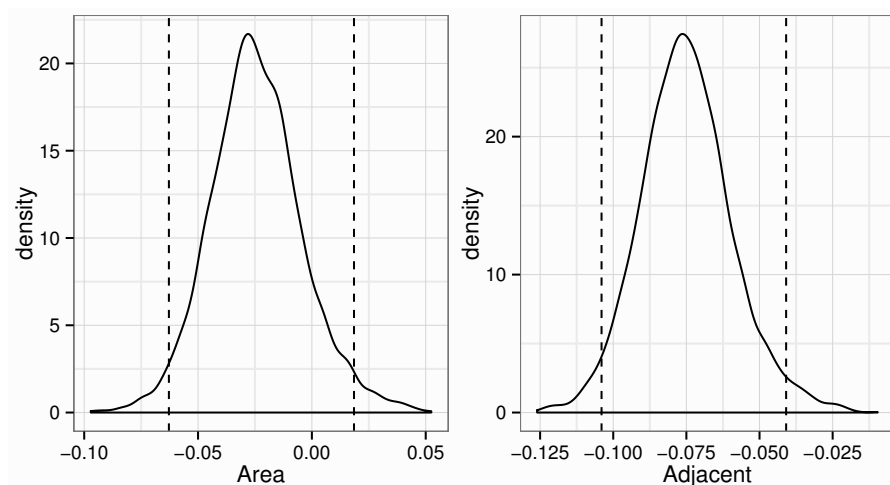


Figure 3.3 Bootstrap regression coefficient densities for β_{Area} and $\beta_{Adjacent}$; 95% confidence intervals are shown as dotted lines

We see in both cases that the bootstrap density is symmetrical and similar to a normal density. This is not always so and our method for computing the bootstrap confidence intervals does not rely on symmetry.

Bootstrap methods can also be used for hypothesis testing although permutation tests are generally better for this purpose. There are alternative resampling methods. We can resample (X, Y) pairs rather than residuals although this seems less attractive, particularly when X is regarded as fixed, such as in designed experiments. There are also more sophisticated methods for constructing the confidence intervals. For more on this, see Efron and Tibshirani (1993) or Davison (1997).

Exercises

- For the `prostate` data, fit a model with `lpsa` as the response and the other variables as predictors:
 - Compute 90 and 95% CIs for the parameter associated with `age`. Using just

these intervals, what could we have deduced about the p -value for age in the regression summary?

- (b) Compute and display a 95% joint confidence region for the parameters associated with `age` and `lbph`. Plot the origin on this display. The location of the origin on the display tells us the outcome of a certain hypothesis test. State that test and its outcome.
 - (c) In the text, we made a permutation test corresponding to the F -test for the significance of all the predictors. Execute the permutation test corresponding to the t -test for age in this model. (Hint: `{summary(g)}$coef[4,3]` gets you the t -statistic you need if the model is called `g`.)
 - (d) Remove all the predictors that are not significant at the 5% level. Test this model against the original model. Which model is preferred?
2. Thirty samples of cheddar cheese were analyzed for their content of acetic acid, hydrogen sulfide and lactic acid. Each sample was tasted and scored by a panel of judges and the average taste score produced. Use the `cheddar` data to answer the following:
 - (a) Fit a regression model with taste as the response and the three chemical contents as predictors. Identify the predictors that are statistically significant at the 5% level.
 - (b) Acetic and H₂S are measured on a log scale. Fit a linear model where all three predictors are measured on their original scale. Identify the predictors that are statistically significant at the 5% level for this model.
 - (c) Can we use an F -test to compare these two models? Explain. Which model provides a better fit to the data? Explain your reasoning.
 - (d) If H₂S is increased 0.01 for the model used in (a), what change in the taste would be expected?
 - (e) What is the percentage change in H₂S on the original scale corresponding to an additive increase of 0.01 on the (natural) log scale?
 3. Using the `teengamb` data, fit a model with `gamble` as the response and the other variables as predictors.
 - (a) Which variables are statistically significant at the 5% level?
 - (b) What interpretation should be given to the coefficient for `sex`?
 - (c) Fit a model with just `income` as a predictor and use an F -test to compare it to the full model.
 4. Using the `sat` data:
 - (a) Fit a model with total sat score as the response and `expend`, `ratio` and `salary` as predictors. Test the hypothesis that $\beta_{salary} = 0$. Test the hypothesis that $\beta_{salary} = \beta_{ratio} = \beta_{expend} = 0$. Do any of these predictors have an effect on the response?
 - (b) Now add `takers` to the model. Test the hypothesis that $\beta_{takers} = 0$. Compare this model to the previous one using an F -test. Demonstrate that the F -test and t -test here are equivalent.

5. Find a formula relating R^2 and the F -test for the regression.
6. Thirty-nine MBA students were asked about happiness and how this related to their income and social life. The data are found in `happy`. Fit a regression model with `happy` as the response and the other four variables as predictors.
 - (a) Which predictors were statistically significant at the 1% level?
 - (b) Use the `table` function to produce a numerical summary of the response. What assumption used to perform the t-tests seems questionable in light of this summary?
 - (c) Use the permutation procedure described in Section 3.3 to test the significance of the `money` predictor.
 - (d) Plot a histogram of the permutation t-statistics. Make sure you use the probability rather than frequency version of the histogram.
 - (e) Overlay an appropriate t -density over the histogram. *Hint:* Use `grid <- seq(-3, 3, length = 300)` to create a grid of values, then use the `dt` function to compute the t -density on this grid and the `lines` function to superimpose the result.
 - (f) Use the bootstrap procedure from Section 3.6 to compute 90% and 95% confidence intervals for β_{money} . Does zero fall within these confidence intervals? Are these results consistent with previous tests?
7. In the `punting` data, we find the average distance punted and hang times of 10 punts of an American football as related to various measures of leg strength for 13 volunteers.
 - (a) Fit a regression model with `Distance` as the response and the right and left leg strengths and flexibilities as predictors. Which predictors are significant at the 5% level?
 - (b) Use an F-test to determine whether collectively these four predictors have a relationship to the response.
 - (c) Relative to the model in (a), test whether the right and left leg strengths have the same effect.
 - (d) Construct a 95% confidence region for $(\beta_{RStr}, \beta_{LStr})$. Explain how the test in (c) relates to this region.
 - (e) Fit a model to test the hypothesis that it is total leg strength defined by adding the right and left leg strengths that is sufficient to predict the response in comparison to using individual left and right leg strengths.
 - (f) Relative to the model in (a), test whether the right and left leg flexibilities have the same effect.
 - (g) Test for left–right symmetry by performing the tests in (c) and (f) simultaneously.
 - (h) Fit a model with `Hang` as the response and the same four predictors. Can we make a test to compare this model to that used in (a)? Explain.

Prediction

Prediction is one of the main uses for regression models. We use subject matter knowledge and the data to build a model $y = X\beta + \varepsilon$. We may have transformed some of the variables or omitted some data points but suppose we have settled on a particular form for the model. Given a new set of predictors, x_0 , the predicted response is:

$$\hat{y}_0 = x_0^T \hat{\beta}$$

However, we need to assess the uncertainty in this prediction. Decision makers need more than just a point estimate to make rational choices. If the prediction has a wide CI, we need to allow for outcomes far from the point estimate. For example, suppose we need to predict the high water mark of a river. We may need to construct barriers high enough to withstand floods much higher than the predicted maximum. Financial projections are not so useful without a realistic estimate of uncertainty because we need to make sensible plans for when events do not turn out as well as we predict.

4.1 Confidence Intervals for Predictions

There are two kinds of predictions made from regression models. One is a predicted mean response and the other is a prediction of a future observation. To make the distinction clear, suppose we have built a regression model that predicts the rental price of houses in a given area based on predictors such as the number of bedrooms and closeness to a major highway. There are two kinds of predictions that can be made for a given x_0 :

1. Suppose a specific house comes on the market with characteristics x_0 . Its rental price will be $x_0^T \beta + \varepsilon$. Since $E\varepsilon = 0$, the predicted price is $x_0^T \hat{\beta}$, but in assessing the variance of this prediction, we must include the variance of ε .
2. Suppose we ask the question — “What would a house with characteristics x_0 rent for on average?” This selling price is $x_0^T \beta$ and is again predicted by $x_0^T \hat{\beta}$ but now only the variance in $\hat{\beta}$ needs to be taken into account.

Most times, we will want the first case, which is called “prediction of a future value,” while the second case, called “prediction of the mean response” is less commonly required. We have:

$$\text{var}(x_0^T \hat{\beta}) = x_0^T (X^T X)^{-1} x_0 \sigma^2$$

A future observation is predicted to be $x_0^T \hat{\beta} + \varepsilon$. We do not know the future ε but we expect it has mean zero so the point prediction is $x_0^T \hat{\beta}$. It is usually reasonable to

assume that the future ε is independent of $\hat{\beta}$. So a $100(1 - \alpha)$ % CI for a single future response is:

$$\hat{y}_0 \pm t_{n-p}^{(\alpha/2)} \hat{\sigma} \sqrt{1 + x_0^T (X^T X)^{-1} x_0}$$

There is a conceptual difference here because previous confidence intervals have been for parameters. Parameters are considered to be fixed but unknown — they are not random under the Frequentist approach we are using here. However, a future observation is a random variable. For this reason, it is better to call this a *prediction interval*. We are saying there is a 95% chance that the future value falls within this interval whereas it would be incorrect to say that for a parameter.

The CI for the mean response for given x_0 is:

$$\hat{y}_0 \pm t_{n-p}^{(\alpha/2)} \hat{\sigma} \sqrt{x_0^T (X^T X)^{-1} x_0}$$

This CI is typically much narrower than the prediction interval. Although we would like to have a narrower prediction interval, we should not make the mistake of using this version when forming intervals for predicted values.

4.2 Predicting Body Fat

Measuring body fat is not simple. Muscle and bone are denser than fat so an estimate of body density can be used to estimate the proportion of fat in the body. Measuring someone's weight is easy but volume is more difficult. One method requires submerging the body underwater in a tank and measuring the increase in the water level. Most people would prefer not to be submerged underwater to get a measure of body fat so we would like to have an easier method. In order to develop such a method, researchers recorded age, weight, height, and 10 body circumference measurements for 252 men. Each man's percentage of body fat was accurately estimated by an underwater weighing technique. Can we predict body fat using just the easy-to-record measurements?

We load in the data and fit a model using all thirteen predictors.

```
> data(fat, package="faraway")
> lmod <- lm(brozek ~ age + weight + height + neck + chest + abdom +
  hip + thigh + knee + ankle + biceps + forearm + wrist, data=fat)
```

We use `brozek` as the response (Brozek's equation estimates percent body fat from density). Normally, we would start with an exploratory analysis of the data and a detailed consideration of what model to use but let's be rash and just fit a model and start predicting.

Let's consider the typical man, exemplified by the median value of all the predictors.

```
> x <- model.matrix(lmod)
> (x0 <- apply(x, 2, median))
(Intercept)      age      weight      height      neck
      1.00      43.00      176.50      70.00      38.00
      chest      abdom      hip      thigh      knee
      99.65      90.95      99.30      59.00      38.50
```

```

      ankle      biceps      forearm      wrist
      22.80      32.05      28.70      18.30
> (y0 <- sum(x0*coef(lmod)))
[1] 17.493

```

Note that x_0 must include a leading one if there is a predictor in the model. The predicted body fat for this “typical” man is 17.5%. The same result may be obtained more directly using the predict function:

```

> predict(lmod, new=data.frame(t(x0)))
      1
17.493

```

The predict function is quite fussy about the new value argument being in the form of a data frame with the same format as the data used to fit the model.

Now if we want a 95% CI for the prediction, we must decide whether we are predicting the body fat for one particular man or the mean body fat for all men have these same characteristics. Here are the two intervals:

```

> predict(lmod, new=data.frame(t(x0)), interval="prediction")
      fit      lwr      upr
1 17.493  9.6178 25.369
> predict(lmod, new=data.frame(t(x0)), interval="confidence")
      fit      lwr      upr
1 17.493 16.944 18.042

```

The prediction interval ranges from 9.6% body fat up to 25.4%. This is a wide interval since there is a large practical difference between these two limits. One might question the value of such a model. The model has an R^2 of 0.75 but perhaps it is not sufficient for practical use. The confidence interval for the mean response is much narrower, indicating we can be quite sure about the average body fat of the man with the median characteristics. Such information might be useful from a public health perspective where we are concerned about populations rather than individuals.

Extrapolation occurs when we try to predict the response for values of the predictor which lie outside the range of the original data. There are two different types of extrapolation — quantitative and qualitative. Quantitative extrapolation concerns x_0 that are far from the original data. Prediction intervals become wider as we move further from the original data. For multivariate x_0 , the concept of distance from the original data is harder to define. In higher dimensions, most predictions will be substantial extrapolations. Let’s see what happens with a prediction for values at the 95th percentile of the data:

```

> (x1 <- apply(x, 2, function(x) quantile(x, 0.95)))
(intercept)      age      weight      height      neck
      1.000      67.000      225.650      74.500      41.845
      chest      abdom      hip      thigh      knee
      116.340      110.760      112.125      68.545      42.645
      ankle      biceps      forearm      wrist
      25.445      37.200      31.745      19.800
> predict(lmod, new=data.frame(t(x1)), interval="prediction")
      fit      lwr      upr
1 30.018 21.924 38.112
> predict(lmod, new=data.frame(t(x1)), interval="confidence")

```

```

      fit      lwr      upr
1 30.018 28.071 31.965

```

We can see that the confidence interval for the mean response is now almost 4% wide compared with the just over 1% width seen in the middle of the data. This is a considerable increase in uncertainty. The prediction interval is only slightly wider because this interval is dominated by the new error ε rather than the uncertainty in the estimation of β .

Unfortunately, there is a much greater source of variation here. We do not know the correct model for this data. We may do our best with the available background information and exploration of the data to find a good model. But no matter how diligent we were, there would still be substantial *model uncertainty*. We have substantial uncertainty about what form the model should take. We can account for *parametric uncertainty* using the methods we have described but *model uncertainty* is much harder to quantify.

4.3 Autoregression

Consider the data shown in the first panel of Figure 4.1 which shows the monthly number of airline passengers from the early years of air travel. This is an example of a *time series* where a response variable is followed over time. We may want to predict future values of the time series. Linear modeling can be used for this purpose.

We can see an increasing trend which looks more than linear. We can also see some seasonal variation which is increasing over time. For these reasons, we will model the logged passenger numbers. As a first attempt, we might fit a linear function of time to predict this response:

```

> data(airpass, package="faraway")
> plot(pass ~ year, airpass, type="l", ylab="Log(Passengers)")
> lmod <- lm(log(pass) ~ year, airpass)
> lines(exp(predict(lmod)) ~ year, airpass)

```

We can see that fit captures the general upward trend in numbers but does nothing to model the seasonal variation.

Suppose we wish to predict passenger numbers for the next month. We might expect this to depend on the current month. The seasonal variation suggests we also use the observed numbers from 12 months ago and since we are already considering the monthly change from the current month, it makes sense to look at this change from a year ago. Hence we use the numbers from 13 months ago as well. We combine these in a regression model:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_{12} y_{t-12} + \beta_{13} y_{t-13} + \varepsilon_t$$

This is an example of an *autoregressive* process. The response depends on past values of the response. The y_{t-i} are called *lagged* variables. We construct a matrix of lagged variables using the `embed` function. We need the current response plus up to 13 lagged variables. We name the columns of the matrix in a convenient way and form a data frame before fitting the model above:

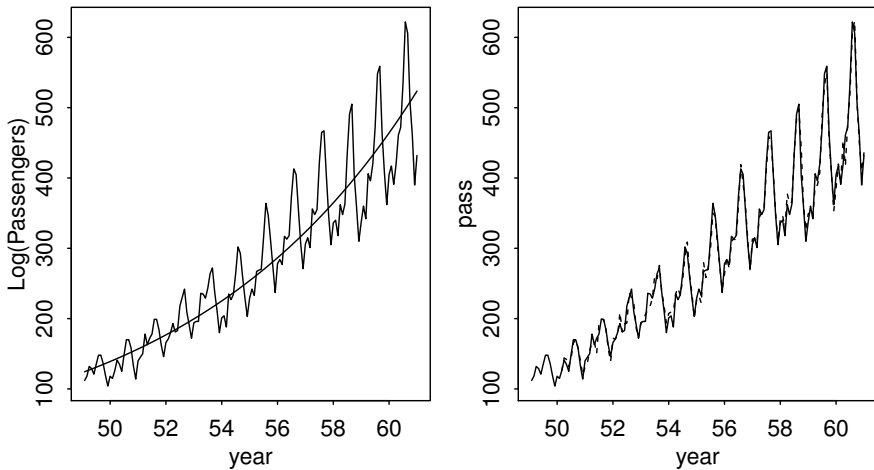


Figure 4.1 Airline passenger numbers 1949 through 1961. An exponential fit is shown on the left. On the right, the fit of an autoregression is shown using dashed lines.

```

> lagdf <- embed(log(airpass$pass), 14)
> colnames(lagdf) <- c("y", paste0("lag", 1:13))
> lagdf <- data.frame(lagdf)
> armod <- lm(y ~ lag1 + lag12 + lag13, data.frame(lagdf))
> summary(armod)

```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.1385	0.0536	2.58	0.011
lag1	0.6923	0.0619	11.19	< 2e-16
lag12	0.9215	0.0347	26.53	< 2e-16
lag13	-0.6321	0.0677	-9.34	4.2e-16

```

n = 131, p = 4, Residual SE = 0.042, R-Squared = 0.99

```

We see all three lagged variables are strongly significant and the fit judged from the R^2 is strong. It is redundant to also include a year term in the model as this linear change over time is now modeled using the lag one variable. We check the fit to the data and observe it in the second panel of Figure 4.1.

```

> plot(pass ~ year, airpass, type="l")
> lines(airpass$year[14:144], exp(predict(armod)), lty=2)

```

Now suppose we want to predict future values. The last observation in our data is:

```

> lagdf[nrow(lagdf), ]

```

y	lag1	lag2	lag3	lag4	lag5	lag6	lag7	lag8
131	6.0684	5.9661	6.1334	6.2305	6.4069	6.4329	6.2823	6.157
	lag9	lag10	lag11	lag12	lag13			
131	6.0379	5.9687	6.0331	6.0039	5.8916			

The current response becomes the lag one value and the other two values also shift. A 95% prediction interval for the (logged) number of passengers is:

```
> predict(armod, data.frame(lag1=6.0684, lag12=6.0331, lag13=6.0039),
  interval="prediction")
      fit      lwr      upr
1 6.104 6.0206 6.1873
```

Now if we want to predict the next month, we will need to plug in the predicted value of 6.104 along with two other shifted lag values. Subsequent predictions will also need to be computed in this recursive manner.

The validity of future predictions depends on the assumption that future data will follow this model. There is no statistical way to check this so we must accept the possibility that our predictions could be badly wrong and plan accordingly.

4.4 What Can Go Wrong with Predictions?

1. Bad model. The Statistician does a poor job of modelling the data.
2. Quantitative extrapolation. We try to predict outcomes for cases with predictor values much different from what we saw in the data. An example is seen above. This is a practical problem in assessing the risk from low exposure to substances which are dangerous in high quantities — consider second-hand tobacco smoke, asbestos and radon.
3. Qualitative extrapolation. We try to predict outcomes for observations that come from a different population. For example, suppose we used the models above to predict body fat for women? This is a common problem because circumstances are always changing and it's hard to judge whether the new case is comparable. We prefer experimental data to observational data but sometimes experience from the laboratory does not transfer to real life.
4. Overconfidence due to overfitting. Data analysts search around for good models for the data they have and often do too good a job in finding a fit. This can lead to unrealistically small $\hat{\sigma}$.
5. Black swans. Sometimes errors can appear to be normally distributed because you haven't seen enough data to be aware of extremes. This is of particular concern in financial applications where stock prices are characterized by mostly small changes (normally distributed) but with infrequent large changes (usually falls).

Exercises

1. For the `prostate` data, fit a model with `lpsa` as the response and the other variables as predictors.
 - (a) Suppose a new patient with the following values arrives:

<code>lcavol</code>	<code>lweight</code>	<code>age</code>	<code>lbph</code>	<code>svi</code>	<code>lcp</code>
1.44692	3.62301	65.00000	0.30010	0.00000	-0.79851
<code>gleason</code>	<code>pgg45</code>				
7.00000	15.00000				

Predict the `lpsa` for this patient along with an appropriate 95% CI.

- (b) Repeat the last question for a patient with the same values except that he is age 20. Explain why the CI is wider.
 - (c) For the model of the previous question, remove all the predictors that are not significant at the 5% level. Now recompute the predictions of the previous question. Are the CIs wider or narrower? Which predictions would you prefer? Explain.
2. Using the `teengamb` data, fit a model with `gamble` as the response and the other variables as predictors.
 - (a) Predict the amount that a male with average (given these data) status, income and verbal score would gamble along with an appropriate 95% CI.
 - (b) Repeat the prediction for a male with maximal values (for this data) of status, income and verbal score. Which CI is wider and why is this result expected?
 - (c) Fit a model with `sqrt(gamble)` as the response but with the same predictors. Now predict the response and give a 95% prediction interval for the individual in (a). Take care to give your answer in the original units of the response.
 - (d) Repeat the prediction for the model in (c) for a female with `status=20`, `income=1`, `verbal = 10`. Comment on the credibility of the result.
3. The `snail` dataset contains percentage water content of the tissues of snails grown under three different levels of relative humidity and two different temperatures.
 - (a) Use the command `xtabs(water ~ temp + humid, snail)/4` to produce a table of mean water content for each combination of temperature and humidity. Can you use this table to predict the water content for a temperature of 25°C and a humidity of 60%? Explain.
 - (b) Fit a regression model with the water content as the response and the temperature and humidity as predictors. Use this model to predict the water content for a temperature of 25°C and a humidity of 60%?
 - (c) Use this model to predict water content for a temperature of 30°C and a humidity of 75%? Compare your prediction to the prediction from (a). Discuss the relative merits of these two predictions.
 - (d) The intercept in your model is 52.6%. Give two values of the predictors for which this represents the predicted response. Is your answer unique? Do you think this represents a reasonable prediction?
 - (e) For a temperature of 25°C, what value of humidity would give a predicted response of 80% water content.
4. The dataset `mdeaths` reports the number of deaths from lung diseases for men in the UK from 1974 to 1979.
 - (a) Make an appropriate plot of the data. At what time of year are deaths most likely to occur?
 - (b) Fit an autoregressive model of the same form used for the airline data. Are all the predictors statistically significant?
 - (c) Use the model to predict the number of deaths in January 1980 along with a 95% prediction interval.

- (d) Use your answer from the previous question to compute a prediction and interval for February 1980.
 - (e) Compute the fitted values. Plot these against the observed values. Note that you will need to select the appropriate observed values. Do you think the accuracy of predictions will be the same for all months of the year?
5. For the `fat` data used in this chapter, a smaller model using only `age`, `weight`, `height` and `abdom` was proposed on the grounds that these predictors are either known by the individual or easily measured.
- (a) Compare this model to the full thirteen-predictor model used earlier in the chapter. Is it justifiable to use the smaller model?
 - (b) Compute a 95% prediction interval for median predictor values and compare to the results to the interval for the full model. Do the intervals differ by a practically important amount?
 - (c) For the smaller model, examine all the observations from case numbers 25 to 50. Which two observations seem particularly anomalous?
 - (d) Recompute the 95% prediction interval for median predictor values after these two anomalous cases have been excluded from the data. Did this make much difference to the outcome?

Explanation

Linear models can be used for prediction or explanation. Prediction is not simple but it is conceptually easier than explanation. We have been deliberately vague about the meaning of explanation. Sometimes explanation means causation but sometimes it is just a description of the relationships between the variables. Causal conclusions require stronger assumptions than those used for predictive models. This chapter looks at the conditions necessary to conclude a causal relationship and what can be said when we lack these conditions.

5.1 Simple Meaning

Let's consider the Galápagos Islands example:

```
> data(gala, package="faraway")
> lmod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
             gala)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.06822	19.15420	0.37	0.7154
Area	-0.02394	0.02242	-1.07	0.2963
Elevation	0.31946	0.05366	5.95	0.0000038
Nearest	0.00914	1.05414	0.01	0.9932
Scruz	-0.24052	0.21540	-1.12	0.2752
Adjacent	-0.07480	0.01770	-4.23	0.0003

```
n = 30, p = 6, Residual SE = 60.975, R-Squared = 0.77
```

What is the meaning of the coefficient for Elevation $\hat{\beta} = 0.31946$?

In a few examples, mostly from the physical sciences or engineering, β might represent a real physical constant. For example, we might attach weights to a spring and measure the extension. Here $\hat{\beta}_1$ will estimate a physical property of the spring. In such examples, the model is a representation of a physical law. But for the Galápagos data, our model has no such strong theoretical underpinning. It is an empirical model that we hope is a good approximation to reality. $\hat{\beta}_1$ has no direct physical meaning in this example.

Let's start with a simple interpretation: *unit increase in x_1 will produce a change of $\hat{\beta}_1$ in the response y .* This statement is rather imprecise regarding the nature of the relationship between x_1 and y so let's be more specific: Compare two islands where the second island has an elevation one meter higher than the first. We predict that the second island will have 0.32 species more than the first. Let's set aside the fact that the number species can only take integer values because the model is only intended as

an approximation. We could specify that the second island was 100m higher than the first resulting in a predicted difference of about 32 species. There are more serious problems with this interpretation.

Consider an alternative model for the number of species that uses only the elevation variable:

```
> summary(lm(Species ~ Elevation, gala))
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.3351    19.2053    0.59   0.56
Elevation     0.2008     0.0346    5.80 0.0000032

n = 30, p = 2, Residual SE = 78.662, R-Squared = 0.55
```

We can see that the predicted difference for an increase of 100m in elevation is now about 20 species more, in contrast to the 32 found for the larger model. The different values illustrate why we cannot interpret a regression coefficient for a given predictor without reference to the other predictors included in the model. We show the two fits for the elevation in Figure 5.1:

```
> plot(Species ~ Elevation, gala)
> abline(11.3, 0.201)
> colMeans(gala)
  Species  Endemics      Area Elevation  Nearest      Scruz  Adjacent
  85.233   26.100   261.709  368.033   10.060   56.977  261.098
> p <- predict(lmod, data.frame(Area=261.7, Elevation=gala$Elevation,
  Nearest=10.06, Scruz = 56.98, Adjacent=261.1))
> i <- order(gala$Elevation)
> lines(gala$Elevation[i], p[i], lty=2)
```

To show the relationship between elevation and the response for the full five-predictor model, we fix the other predictors at some typical values. We have chosen the means. We then compute the predicted response for the observed values of elevation. Notice that we needed to sort the predictor values (and the predictions in the same order) to make sure that the line could be plotted. This is called an *effect plot*. It is a good way of visualizing the meaning of the model for a given predictor. We can even compare the effects to other models as we have done in the example.

We need to be more specific in our interpretation which should now read: *a unit increase in x_1 with the other (named) predictors held constant will produce a change of β_1 in the response y .* For the simple one-predictor model, a change in elevation will also be associated with a change in the other predictors. This explains why the two predictions are different.

Unfortunately, our revised interpretation merely brings to light further difficulties. The idea of holding variables constant makes no sense for observational data such as in our Galápagos example. These observables are not under our control. We cannot change them except by some fantastic feat of engineering. There are also likely to be other variables which we have not measured and may not be aware of that also have some connection to the species diversity. We cannot possibly hold these constant.

Furthermore, our explanation contains no notion of causation. We can predict that taller islands have greater species diversity but we should not say altitude causes

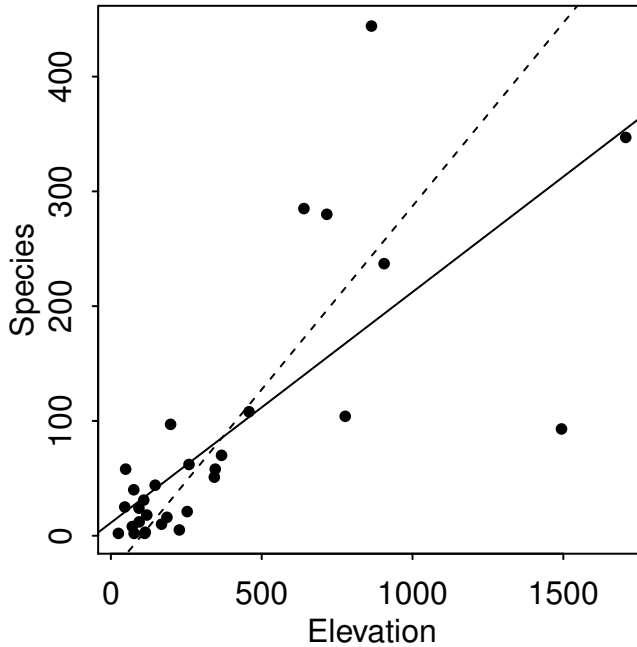


Figure 5.1 The fit for the simple model with just elevation as a predictor is shown as the solid line. The predicted response as elevation varies and the other four predictors held at their mean values is shown as a dashed line.

it. In the next section, we look at how causality might be established. For now we are able to make predictions and compare these predictions to develop some insights. The comparisons give some meaning to the regression coefficients but the information is conditional and tentative.

5.2 Causality

The meaning of causality has occupied philosophers for centuries. We take the view that the causal effect of an action is the difference between the outcomes where the action was or was not taken. For example, suppose a study compares a treatment with a control as applied to a set of patients. Let $T = 0$ for the control and $T = 1$ for the treatment. Now let y_i^T be the response for patient i when T applies. The causal effect for patient i is then defined as

$$\delta_i = y_i^1 - y_i^0$$

Now the practical problem is that we usually cannot apply both treatment and control at the same time. We only get to see one of the two outcomes: (y_i^0, y_i^1) . The outcome we do not see is called the *counterfactual*. How near can we come to approximating this quantity δ_i that we normally cannot observe?

There are some difficulties with the counterfactual definition of causality. For some variables, like a treatment in a clinical trial, it is easy to conceive of how a different version of the treatment might have been applied or how we might change future treatments. But some variables are difficult or impossible to actually change. For example, suppose we are interested in the effect of gender on some outcome. Although it seems reasonable to ask how an outcome might have differed if the subject was male rather than female, this is not a change that would be easy to make. Similarly, with the Galápagos example, although it seems natural to think about how physical geography might affect species diversity, we cannot actually change the physical geography.

Although we cannot see both the outcome and the counterfactual outcome, we may come close in some circumstances. For example, suppose the treatment is a skin cream. We could apply the treatment to one side of the face and leave the other as a control. Or we could apply the control first (which might be no treatment) and then apply the cream later. We could then observe the difference in the post- and pre-treatment responses. However, even in such cases, it is easy to see that some assumptions will be necessary about how local the effect of the treatment is in time and space. We will also need to be wary of external variables that might have differential effects. If these assumptions are reasonable, then we may proceed but in many other circumstances we will have doubts or it simply may not be possible to apply more than one treatment to a unit.

5.3 Designed Experiments

In a designed experiment, we have control over T . For example, suppose we wish to compare two physical exercise regimes. The *experimental units* are the people we use for the study. There may be some other potential predictors which we can control such as the amount of time spent exercising or the type of equipment used. Some other predictors might not be controlled, but can be measured, such as the physical characteristics of the people. Still other predictors may not be controlled or measured. We may know about these predictors or we may be unaware of them. Other possible variables, such as the temperature in the room, might be held constant. Our control over the conditions of the experiment allows us to make stronger conclusions from the analysis. Randomization is the key to success.

Consider the easiest case where we will vary only T . We have some number of subjects available for the experiment. Although we would like to know the individual causal effects δ_i , this is not possible because only one level of T can be assigned to a given subject at a given moment in time. However, we can aspire to estimate the average value of δ over the group. We should randomly assign subjects to treatment or control. Typically, we will do this so that there are an equal number of units in the two groups but this is not essential. There are two compelling reasons to use randomization.

We know that the subjects will vary in ways which may affect the response. Some of these variables may not be obvious or measurable. Randomization is the only reliable way to make sure that the two groups are not unbalanced in some way that

favors either the treatment or the control. But this first argument for randomization only ensures that the groups will be balanced on the average. It may well happen that the particular realization of the randomization is unbalanced. We have a second and more compelling argument to use. A permutation test (see Section 3.3) can be used to test the null hypothesis of no difference between the groups. The justification of the permutation test relies on the observed allocation being chosen randomly from all possible allocations to groups. By using randomization, we validate this assumption. It is true we could get an unbalanced allocation by chance but we also understand that we will reject true null hypotheses by chance. So this possibility is built into the reasoning behind the test. Remember that the permutation test tends to agree with the normal-based test provided the linear model assumptions are justified. So it is usually alright if we use the normal-based test rather than actually do the work required for the permutation test.

The results of this test apply only to the subjects used in the experiment unless we can reasonably claim that these subjects are representative of a larger population. It is best if they are a random sample but this is rarely the case in practice so further assumptions are necessary if we wish to make wider claims.

Now suppose we are aware that the experimental units differ in identifiable ways. For example, some subjects may be male and others female. We may wish to incorporate this into the design by restricting the randomization. The obvious strategy is to randomize separately the assignment of the treatments within males and females. This will ensure that the groups are balanced by sex. In this example, sex is called a *blocking* variable. Such designs are covered in detail in Chapter 17. In other cases, there may be variables which are not properties of the experimental units (like sex) but can be assigned (like time of exercise). In both situations, we will want to arrange the design so that is *orthogonal*. This concept is explained in Section 2.11. This is not essential for causal conclusions but it does greatly simplify them.

We have considered a binary treatment variable T but the same goes for continuous potential causal variables. Some additional assumptions may be necessary regarding the functional form of the relationship — for example, that T has a linear effect on the response.

5.4 Observational Data

Sometimes it is not practical or ethical to collect data from a designed experiment. We cannot control the assignment of T and so we can only obtain observational data. In some situations, we can control which cases we observe from those potentially available. A *sample survey* is used to collect the data. A good survey design can allow stronger and wider conclusions, but the data will still be observational.

On the 8th January 2008, primaries to select US presidential candidates were held in New Hampshire. In the Democratic party primary, Hillary Clinton defeated Barack Obama contrary to the expectations of pre-election opinion polls. Two different voting technologies were used in New Hampshire. Some wards (administrative districts) used paper ballots, counted by hand while others used optically scanned ballots, counted by machine. Among the paper ballots, Obama had more votes than

Clinton while Clinton defeated Obama on just the machine-counted ballots. Since the method of voting should make no causal difference to the outcome, suspicions were raised regarding the integrity of the election. The data was derived from Herron et al. (2008) where a more detailed analysis may be found.

```
> data(newhamp, package="faraway")
> colSums(newhamp[newhamp$votesys == 'D', 2:3])
Obama Clinton
86353      96890
> colSums(newhamp[newhamp$votesys == 'H', 2:3])
Obama Clinton
16926      14471
```

We focus our interest on the voting system variable `votesys` which can either be 'D' for digital or 'H' for hand.. We use the proportion voting for Obama in each ward as the response. Strictly speaking this is a binomial response and we should model it accordingly. Nevertheless, the normal is a good approximation for the binomial given a large enough sample and probabilities not close to zero or one. We have that in this sample. Even so, a binomial variance is $np(1-p)$ for proportion p and sample size n . Both these vary in this example and so the assumption of equal variance is violated. We can fix this problem by using weights as described in Section 8.2, but this would make no appreciable difference to our discussion here so we ignore it.

Let us fit a linear model with just voting system as a predictor. We create an indicator variable for this treatment where one represents hand and zero represents digital voting:

```
> newhamp$trt <- ifelse(newhamp$votesys == 'H', 1, 0)
> lmodu <- lm(pObama ~ trt, newhamp)
> summary(lmodu)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.35252	0.00517	68.15	< 2e-16
trt	0.04249	0.00851	4.99	0.0000011

n = 276, p = 2, Residual SE = 0.068, R-Squared = 0.08

The model takes the form:

$$y_i = \beta_0 + \beta_1 T_i + \varepsilon_i$$

When digital voting is used, the predictor is set to zero and the predicted proportion is therefore $\hat{\beta}_0 = 35\%$. When hand voting is used, the prediction is $\hat{\beta}_1 = 4\%$ higher. We see from the very small p -value of 0.0000011 that this difference is significantly different. So we are quite sure that Obama received a significantly higher proportion of the vote in the hand voting wards. The question is why? Did the voting method have some causal effect on the outcome?

Suppose that the correct model involved some third variable Z and took the form:

$$y_i = \beta_0^* + \beta_1^* T_i + \beta_2^* Z_i + \varepsilon_i$$

and suppose that this Z was linked to T by:

$$Z_i = \gamma_0 + \gamma_1 T_i + \varepsilon_i'$$

Z is sometimes called a *confounding* variable. Substituting the latter into the former, we find the coefficient for T is $\beta_1^* + \beta_2^* \gamma_1$. Compare this to the β_1 in the initial model where we do not include Z . There are two ways in which the conclusion could be the same for the two models. We could have $\beta_2^* = 0$ when Z has no effect on the response or $\gamma_1 = 0$ where the treatment has no effect on Z . Otherwise Z will have an effect on our conclusions and the initial model which excludes Z will provide a biased estimate of the treatment effect. In a designed experiment, we have $\gamma_1 = 0$ by the randomization in the assignment of T .

Does such a third variable Z exist for the New Hampshire voting example? Consider the proportion of votes for Howard Dean, a Democratic candidate in the previous presidential campaign in 2004. We add this term to the model:

```
> lmodz <- lm(pObama ~ trt + Dean, newhamp)
> summary(lmodz)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.22112	0.01125	19.65	<2e-16
trt	-0.00475	0.00776	-0.61	0.54
Dean	0.52290	0.04165	12.55	<2e-16

n = 276, p = 3, Residual SE = 0.054, R-Squared = 0.42

We see that the effect of the voting system is no longer statistically significant with a p -value of 0.54. The proportion voting for Dean shows a positive relationship to the proportion voting for Obama. We can also see that this third variable is related to our “treatment” variable:

```
> summary(lm(Dean ~ trt, newhamp))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.25129	0.00599	41.99	<2e-16
trt	0.09034	0.00985	9.18	<2e-16

n = 276, p = 2, Residual SE = 0.079, R-Squared = 0.24

We can see that there is an active confounder in this situation. What should we conclude? In the next section, we show how we can use counterfactual notions to clarify the effect of the voting system on preferences for Obama.

5.5 Matching

People vote for candidates based on political and character preferences. The proportion of voters choosing Howard Dean in the previous Democratic primary tells us something about the aggregate preferences of the voters in each ward in the 2008 primary. Suppose that we had been allowed to do an experiment in 2008 and we were permitted to assign the voting systems to the wards. How would we do this?

Consider a clinical trial where we compare a treatment to a control. We have a pool of available subjects who differ in identifiable ways such as sex, age, overall health condition and so on that might affect the response. We could simply randomly divide the pool in two, assigning one group to the treatment and one to the control. But the random division could be unbalanced with respect to the confounders. Although this possibility is built into the testing procedure, the approach does not make

the best use of the available information. Often a better approach is to form matched pairs where the two members of each pair are as alike as possible with respect to the confounders. For example, we would match a healthy older man with another healthy older man. Within the pairs, treatment and control are randomly assigned. We can then determine the effect of the treatment by looking at the difference in the response for each pair. We do this secure in the knowledge that we have adjusted for the effect of confounding by balancing these differences between treatment and control.

Let's return to the New Hampshire primary and apply the same procedure. We would form matched pairs based on similarity of proportion formerly supporting Dean. Within each pair, we would make a random assignment. Unfortunately, in real life, we cannot make the random assignment but we can find pairs of wards with similar values of Dean proportion where one uses hand voting and the other uses digital.

We use the Matching package of (Sekhon 2011) to find the matches:

```
> require(Matching)
> set.seed(123)
> mm <- GenMatch(newhamp$trt, newhamp$Dean, ties=FALSE, caliper=0.05,
  pop.size=1000)
> head(mm$matches[,1:2])
  [,1] [,2]
[1,]  4 218
[2,] 17 170
[3,] 18 264
[4,] 19  91
[5,] 21 233
[6,] 22 185
> newhamp[c(4, 218), c('Dean', 'pObama', 'trt')]
      Dean pObama trt
CenterHarbor 0.28495 0.34328 1
Northwood 0.28264 0.33688 0
```

Because Dean is a continuous variable, it is unlikely that we will find exact matches so we need to specify how a close a match is acceptable. A caliper of 0.05 means that we accept anything within 0.05 standard deviations of Dean. For simplicity, we also specify that no ties will be allowed so that each treatment ward will be matched to just one control. The matching method uses a genetic algorithm which has a random component. For this reason, we have set the random number generator seed so you will see the same results if you repeat this. The genetic algorithm functions more effectively with a larger 'population size' so this should be set as high as possible without causing the computation to take too long.

We examine the first few pairs of matches and see that first pair uses cases 4 and 218 which we look at specifically. For this pair of treatment and control, we see that the value of Dean is very similar.

We show all the matches in Figure 5.2. Notice that some wards go unmatched because there is not a close match of the other type available.

```
> plot(pObama ~ Dean, newhamp, pch=trt+1)
> with(newhamp, segments(Dean[mm$match[,1]], pObama[mm$match[,1]], Dean[
  mm$match[,2]], pObama[mm$match[,2]]))
```

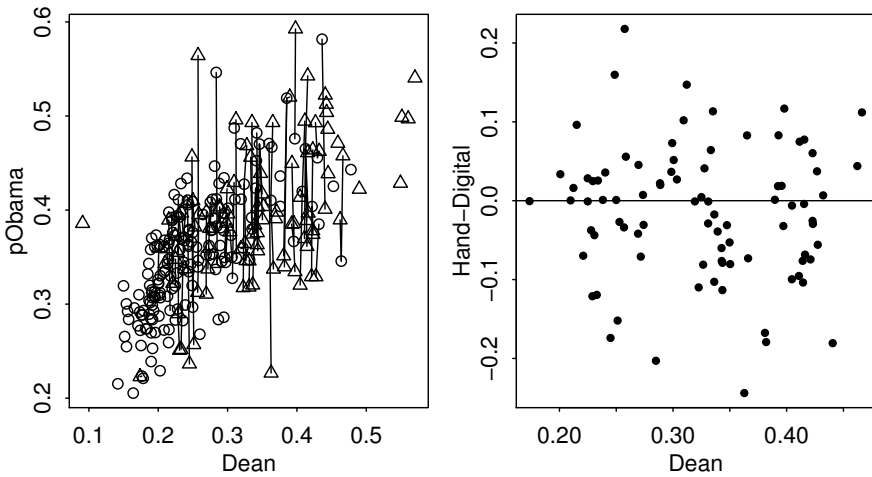


Figure 5.2 On the left, we show the matched pairs hand voting wards are shown as triangles and digital voting wards as circles. Matched pair are linked by a line. On the right, we show the matched pair differences

We compute the difference between the pairs and perform a one sample t-test. We have also plotted the differences in Figure 5.2.

```
> pdiff <- newhamp$pObama[mm$matches[,1]] - newhamp$pObama[mm$matches
[,2]]
> t.test(pdiff)

One Sample t-test

data:  pdiff
t = -1.5907, df = 86, p-value = 0.1154
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.0303409  0.0033683
sample estimates:
mean of x
-0.013486
> plot(pdiff ~ newhamp$Dean[mm$matches[,1]], xlab="Dean", ylab="Hand-
Digital")
> abline(h=0)
```

We see that the differences are not statistically significantly different from zero. This is confirmed by the plot where we can see that the matched pairs show no clear preference for hand or digital voting. Furthermore, the difference doesn't appear to depend on the Dean vote.

We were not able to do an experiment and we certainly were not able to view the counterfactual, but we achieved the next best thing. We have matched wards with similar political outlooks by using the prior Dean vote proportion. One ward has used digital voting and the other has used the manual method. Based on this comparison, we see no significant difference between the two methods. The observed unadjusted

difference is because voters inclined to pick Obama are also more likely to be present in hand voting wards

Now we haven't disproved any conspiracy theories about fixed votes but we have come up with a more plausible explanation. We do not claim our new explanation is necessarily true because there may be yet further variables that drive the relationship. Nevertheless, it is a basic principle of science that we should prefer simpler theories provided they are tenable.

There is much more to matching. We have presented a simple approach but there are better ways to do the analysis and more sophisticated ways to do the matching in the presence of covariates but now we return to our earlier linear modeling and consider how this relates to the matched analysis.

5.6 Covariate Adjustment

We can display the two earlier linear models for the proportion voting for Obama. First the unadjusted model which does not include Dean:

```
> plot(pObama ~ Dean, newhamp, pch=trt+1)
> abline(h=c(.353, 0.353+0.042), lty=1:2)
```

We can see the data and fitted models in Figure 5.3. The two horizontal lines representing the mean response in the two groups show a clear difference. We can add the two lines that represent the fitted model for the two levels of voting mechanism:

```
> abline(0.221, 0.5229)
> abline(0.221-0.005, 0.5229, lty=2)
```

Let's also add the matched pairs:

```
> with(newhamp, segments(Dean[mm$match[, 1]], pObama[mm$match[, 1]], Dean[
  mm$match[, 2]], pObama[mm$match[, 2]], col=gray(0.75)))
```

If we consider the expected response for a fixed value of `Dean`, the difference between digital and hand is given by the vertical distance between the two sloped lines. Compare this to the matching approach where we find pairs of wards, one digital, one hand, which have almost the same value of `Dean`. Each pair represents a local realization of that vertical difference. When we average these pairwise differences, we also estimate the vertical difference. Thus matching on a covariate and fitting a linear model where we adjust for the covariate by including it in the model are two ways to estimate the same thing.

Both approaches to drawing conclusions from observation studies are useful. The covariate adjustment method, sometimes called “controlling for a covariate” is easier to use and extends well to multiple confounders. However, it does require that we specify the functional form of the covariate in the model in an appropriate way. In our example, it is clear from the plots that a linear form is suitable but in higher dimensions this can be harder to check. The matching approach is more robust in that it does not require we specify this functional form. In our example, the two groups overlapped substantially so that it was possible to obtain a large number of matched pairs. When there is less overlap, the covariate adjustment approach can still make full use of the data although it is vulnerable to the dangers of extrapolation when

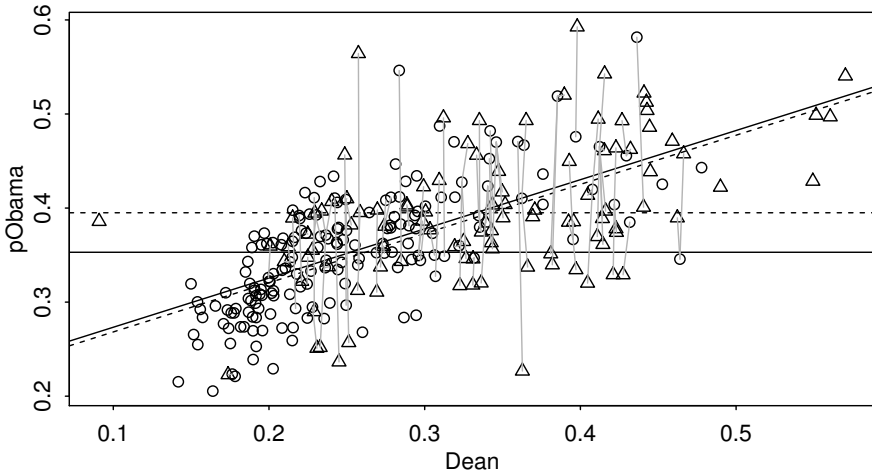


Figure 5.3 Linear models for Obama proportion. The solid lines represent the digital voting wards while the dashed lines represent the hand voting wards. The horizontal lines result from an unadjusted model while the sloped lines result from the model that includes the proportion voting for Dean. Matched pairs are linked by gray lines.

making comparisons. In the rest of this book, we shall generally use the covariate adjustment approach but the matching approach should not be discounted and can provide valuable insights.

5.7 Qualitative Support for Causation

In observational studies, there is a limit to what can be achieved with statistical manipulations. Sir Bradford Hill was a central figure in establishing the causal link between smoking and lung cancer. In Hill (1965) he laid out several general considerations that can reinforce the case for a causal link:

Strength We do not mean a high correlation or a small p -value but that $\hat{\beta}$ is large in practical terms. Known covariates can be adjusted for, while unobserved and unsuspected confounding variables could easily lead to small effects. It is less credible that some variable whose existence was previously unknown could counteract a large effect.

Consistency A similar effect has been found for different subjects under different circumstances at different times and places. If smokers, male and female, Argentinian and Swedish, both now and in the past tend to get lung cancer, then the evidence is stronger. Replication by independent research groups is particularly important in establishing causation.

Specificity The supposed causal factor is associated mostly with a particular response and not with a wide range of other possible responses. If a particular lung

disease is only prevalent in workers in a particular industry and those workers do not suffer from other problems any more than other industrial workers, then the case is stronger.

Temporality The supposed causal factor is determined or fixed before the outcome or response is generated. Sometimes it is not clear whether X causes Y or vice versa. It helps if X happens before Y if we want to establish the direction of the effect.

Gradient The response increases (or decreases) monotonely as the supposed causal variable increases. In other words, “the more you do it, the worse it gets.” The phenomenon of *hormesis* is an exception. For example, a low consumption of alcohol may have health benefits relative to abstinence but drinking too much becomes progressively more harmful.

Plausibility There is a credible theory suggesting a causal effect. The observational study might be part of an effort to support a particular theory.

Experiment A natural experiment exists where subjects have apparently been randomly assigned values of the causal variable. For example, US states have different laws that have differential effects on the causal factor. Sometimes it is reasonable to view this as a form of random assignment. Generally, we conduct observational studies because experiments are not practical but sometimes it is possible to do related experiments. For example, animal experiments might tell us something about effects for humans.

Not all of these might apply, but causation might still be present. All of these might apply, but we may not be sure of causation. Nevertheless, these considerations do add to the weight of evidence even though we cannot express the effect numerically.

Exercises

1. Use the `teengamb` data with `gamble` as the response. We focus on the effect of `sex` on the response and so we include this predictor in all models. There are eight possible models that include all, some, or none of the other three predictors. Fit all these models and report on the coefficient and significance of `sex` in each case. Comment on the stability of the effect.
2. Use the `odor` dataset with `odor` as the response and `temp` as a predictor. Consider all possible models that also include all, some or none of the other two predictors. Report the coefficient for temperature, its standard error, t-statistic and p -value in each case. Discuss what stays the same, what changes and why. Which model is best?
3. Use the `teengamb` data for this question.
 - (a) Make a plot of `gamble` on `income` using a different plotting symbol depending on the `sex`.
 - (b) Fit a regression model with `gamble` as the response and `income` and `sex` as predictors. Display the regression fit on the plot.

- (c) Use the `Matching` package to find matches on `sex`. Use the same parameters as in the chapter. How many matched pairs were found? How many cases were not matched?
 - (d) Make a plot showing which pairs were matched.
 - (e) Compute the differences in `gamble` for the matched pairs. Is there a significant non-zero difference?
 - (f) Plot the difference against `income`. In what proportion of pairs did the female gamble more than the male?
 - (g) Do the conclusions from the linear model and the matched pair approach agree?
4. Thirty-nine MBA students were asked about happiness and how this related to their income and social life. The data are found in `happy`.
- (a) Fit a regression model with `happy` as the response and the other four variables as predictors. Give an interpretation for the meaning of the `love` coefficient.
 - (b) The `love` predictor takes three possible values but mostly takes the value 2 or 3. Create a new predictor called `clove` which takes the value zero if `love` is 2 or less. Use this new predictor to replace `love` in the regression model and interpret the meaning of the corresponding coefficient. Do the results differ much from the previous model?
 - (c) Fit a model with only `clove` as a predictor and interpret the coefficient. How do the results compare to the previous outcome.
 - (d) Make a plot of `happy` on `work`, distinguishing the value `clove` by using a plotting symbol. Use `jittering` to distinguish overplotted points.
 - (e) Use the command `xtabs(~ clove + work, happy)` to produce a crosstabulation. If we wanted to match pairs on `clove` with the same value of `work`, what is the maximum number of 1 to 1 matches we could achieve?
 - (f) For each value of `work`, compute the mean difference in `happy` for the two levels of `clove`. Compute an average of these differences. Which coefficient computed earlier would be the most appropriate comparison for this average?

Diagnostics

The estimation of and inference from the regression model depend on several assumptions. These assumptions should be checked using *regression diagnostics* before using the model in earnest. We divide the potential problems into three categories:

Error We have assumed that $\epsilon \sim N(0, \sigma^2 I)$ or in words, that the errors are independent, have equal variance and are normally distributed.

Model We have assumed that the structural part of the model, $Ey = X\beta$, is correct.

Unusual observations Sometimes just a few observations do not fit the model. These few observations might change the choice and fit of the model.

Diagnostic techniques can be graphical, which are more flexible but harder to definitively interpret, or numerical, which are narrower in scope, but require less intuition. When searching for a good model, the first one we try may prove to be inadequate. Regression diagnostics often suggest specific improvements, which means model building is an iterative and interactive process. It is quite common to repeat the diagnostics on a succession of models.

6.1 Checking Error Assumptions

We wish to check the independence, constant variance and normality of the errors, ϵ . The errors are not observable, but we can examine the residuals, $\hat{\epsilon}$. These are not interchangeable with the error, as they have somewhat different properties. Recall that $\hat{y} = X(X^T X)^{-1} X^T y = Hy$ where H is the hat matrix, so that :

$$\hat{\epsilon} = y - \hat{y} = (I - H)y = (I - H)X\beta + (I - H)\epsilon = (I - H)\epsilon$$

Therefore, $\text{var } \hat{\epsilon} = \text{var } (I - H)\epsilon = (I - H)\sigma^2$ assuming that $\text{var } \epsilon = \sigma^2 I$. We see that although the errors may have equal variance and be uncorrelated, the residuals do not. Fortunately, the impact of this is usually small and diagnostics can reasonably be applied to the residuals in order to check the assumptions on the error.

6.1.1 Constant Variance

It is not possible to check the assumption of constant variance just by examining the residuals alone — some will be large and some will be small, but this proves nothing. We need to check whether the variance in the residuals is related to some other quantity.

The most useful diagnostic is a plot of $\hat{\epsilon}$ against \hat{y} . If all is well, you should see constant symmetrical variation (known as homoscedasticity) in the vertical ($\hat{\epsilon}$) direction. Nonconstant variance is also called heteroscedasticity. Nonlinearity in the structural part of the model can also be detected in this plot. In Figure 6.1, three distinct cases are illustrated.

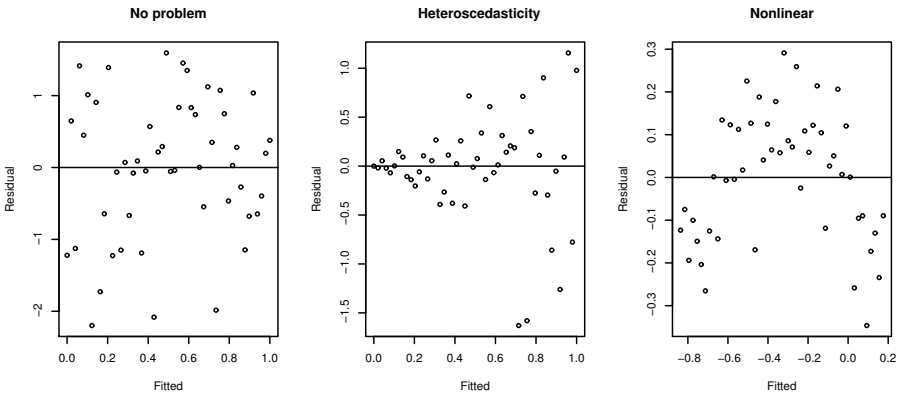


Figure 6.1 Residuals vs. fitted plots — the first suggests no change to the current model while the second shows nonconstant variance and the third indicates some nonlinearity, which should prompt some change in the structural form of the model.

It is also worthwhile to plot $\hat{\epsilon}$ against x_i for potential predictors that are in the current model as well as those not used. The same considerations in these plots should apply. For plots of residuals against predictors not in the model, any observed structure may indicate that this predictor should be included in the model.

We illustrate this using the `savings` dataset:

```
> data(savings, package="faraway")
> lmod <- lm(sr ~ pop15+pop75+dpi+ddpi, savings)
```

The basic plot, as seen in the first panel of Figure 6.2, shows residuals against fitted values. It is worth adding the $\hat{\epsilon} = 0$ line to help with the interpretation.

```
> plot(fitted(lmod), residuals(lmod), xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

We see no cause for alarm in this plot. If we would like to examine the constant variance assumption more closely, it helps to plot $\sqrt{|\hat{\epsilon}|}$ against \hat{y} . Once we have excluded the possibility of non-linearity in the first plot and the residuals look roughly symmetric, then we can effectively double the resolution by considering the absolute value of the residuals. For truly normal errors, $|\hat{\epsilon}|$ would follow what is known as a half-normal distribution (since it has a density which is simply the upper half of a normal density). Such a distribution is quite skewed, but this effect can be reduced by the square root transformation.

```
> plot(fitted(lmod), sqrt(abs(residuals(lmod))), xlab="Fitted", ylab=
  expression(sqrt(hat(epsilon))))
```

The plot, as seen in the second panel of Figure 6.2, shows approximately constant variation. A quick numerical test to check nonconstant variance can be achieved

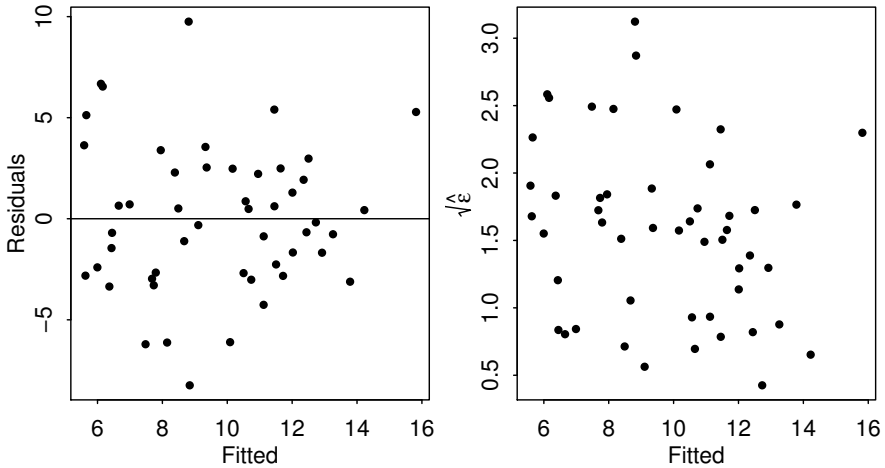


Figure 6.2 Residual vs. fitted plot for the savings data is shown on the left and the corresponding $\sqrt{|\hat{\epsilon}|}$ against \hat{y} on the right. Both plots are satisfactory.

using this regression:

```
> summary(lm(sqrt(abs(residuals(lmod))) ~ fitted(lmod)))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.1622	0.3479	6.22	1.2e-07
fitted(lmod)	-0.0614	0.0348	-1.77	0.084

n = 50, p = 2, Residual SE = 0.634, R-Squared = 0.06

This test is not quite right, as some weighting should be used and the degrees of freedom should be adjusted, but there does not seem to be a clear problem with nonconstant variance. Another difficulty is that it only tests for a linear trend in the variation. A formal test may be good at detecting a particular kind of nonconstant variance, but have no power to detect another. Residual plots are more versatile because unanticipated problems may be spotted.

A formal diagnostic test may have a reassuring aura of exactitude about it, but one needs to understand that any such test may be powerless to detect problems of an unsuspected nature. Graphical techniques are usually more effective at revealing structure that you may not have suspected. Of course, sometimes the interpretation of the plot may be ambiguous, but at least we can be sure that nothing is seriously wrong with the assumptions. For this reason, we usually prefer a graphical approach to diagnostics, with formal tests reserved for the clarification of indications discovered in the plots.

It is often hard to judge residual plots without prior experience so it is helpful to generate some artificial plots where the true relationship is known. The following four `for` loops show:

1. Constant variance
2. Strong nonconstant variance
3. Mild nonconstant variance
4. Nonlinearity

```
> par(mfrow=c(3,3))
> n <- 50
> for(i in 1:9) {x <- runif(n) ; plot(x,rnorm(n))}
> for(i in 1:9) {x <- runif(n) ; plot(x,x*rnorm(n))}
> for(i in 1:9) {x <- runif(n) ; plot(x,sqrt(x)*rnorm(n))}
> for(i in 1:9) {x <- runif(n) ; plot(x,cos(x*pi/25)+rnorm(n,sd=1))}
> par(mfrow=c(1,1))
```

Repeat, varying the sample size and `sd` to get an idea of the how plots might vary even when we know the true data generating mechanism. Type `help(Distributions)` to see how to generate data from different distributions to get an even broader picture.

The artificial generation of plots is a good way to “calibrate” diagnostic plots. It is often hard to judge whether an apparent feature is real or just random variation. Repeated generation of plots under a known model, where there is or is not a violation of the assumption that the diagnostic plot is designed to check, is helpful in making this judgment.

Now look at some residuals against predictor plots for the `savings` data:

```
> plot(savings$pop15, residuals(lmod), xlab="Population under 15", ylab="Residuals")
> abline(h=0)
> plot(savings$pop75, residuals(lmod), xlab="Population over 75", ylab="Residuals")
> abline(h=0)
```

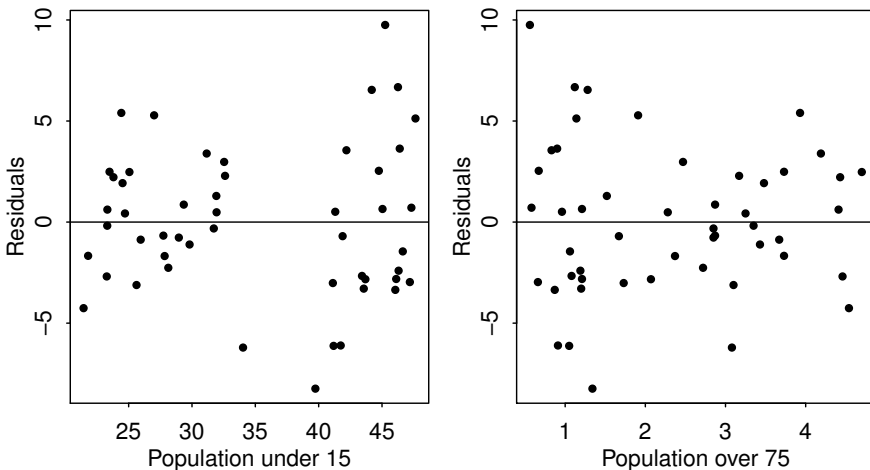


Figure 6.3 Residuals vs. predictor plots for the `savings` data.

The plots may be seen in Figure 6.3. Two groups can be seen in the first plot. Let’s

compare and test the variances in these groups. Given two independent samples from normal distributions, we can test for equal variance using the test statistic of the ratio of the two variances. The null distribution is an F with degrees of freedom given by the two samples:

```
> var.test(residuals(lmod)[savings$pop15>35], residuals(lmod)[savings$
  pop15<35])

      F test to compare two variances

data:  residuals(lmod)[savings$pop15 > 35] and residuals(lmod)[savings$
      pop15 < 35]
F = 2.7851, num df = 22, denom df = 26, p-value = 0.01358
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 1.2410 6.4302
sample estimates:
ratio of variances
      2.7851
```

A significant difference is seen.

When problems are seen in these diagnostic plots, some modification of the model is suggested. If some non-linearity is observed, perhaps in conjunction with non-constant variance, a transformation of the variables should be considered. The shape of the plot may suggest the transformation to be used. See Chapter 9 for more.

If the problem is solely one of non-constant variance with no suggestion of non-linearity then the use of weighted least squares (see Section 8.1) may be appropriate. Alternatively, when non-constant variance is seen in the plot of $\hat{\epsilon}$ against \hat{y} , a transformation of the response y to $h(y)$ where $h(\cdot)$ can be chosen so that $\text{var } h(y)$ is constant. To see how to choose h consider this expansion:

$$\begin{aligned} h(y) &= h(Ey) + (y - Ey)h'(Ey) + \dots \\ \text{var } h(y) &= 0 + h'(Ey)^2 \text{var } y + \dots \end{aligned}$$

We ignore the higher order terms. For $\text{var } h(y)$ to be constant we need:

$$h'(Ey) \propto (\text{var } y)^{-1/2}$$

which suggests:

$$h(y) = \int \frac{dy}{\sqrt{\text{var } y}} = \int \frac{dy}{SDy}$$

For example if $\text{var } y = \text{var } \epsilon \propto (Ey)^2$, then $h(y) = \log y$ is suggested while if $\text{var } \epsilon \propto (Ey)$, then $h(y) = \sqrt{y}$.

In practice, you need to look at the plot of the residuals and fitted values and take a guess at the relationship. When looking at the plot, we see the change in $SD(y)$ rather than $\text{var } y$, because the SD is in the units of the response. If your initial guess is wrong, you will find that the diagnostic plot in the transformed scale is unsatisfactory. You can simply try another transformation — some experimentation is sensible.

Sometimes it can be difficult to find a good transformation. For example, when $y_i \leq 0$ for some i , square root or log transformations will fail. You can try, say, $\log(y + \delta)$, for some small δ but this makes interpretation difficult.

Consider the residual vs. fitted plot for the Galápagos data:

```
> data(gala, package="faraway")
> lmod <- lm(Species ~ Area + Elevation + Scruz + Nearest + Adjacent,
  gala)
> plot(fitted(lmod), residuals(lmod), xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

We can see non-constant variance (and evidence of non-linearity) in the first plot of Figure 6.4. The square root transformation is often appropriate for count response

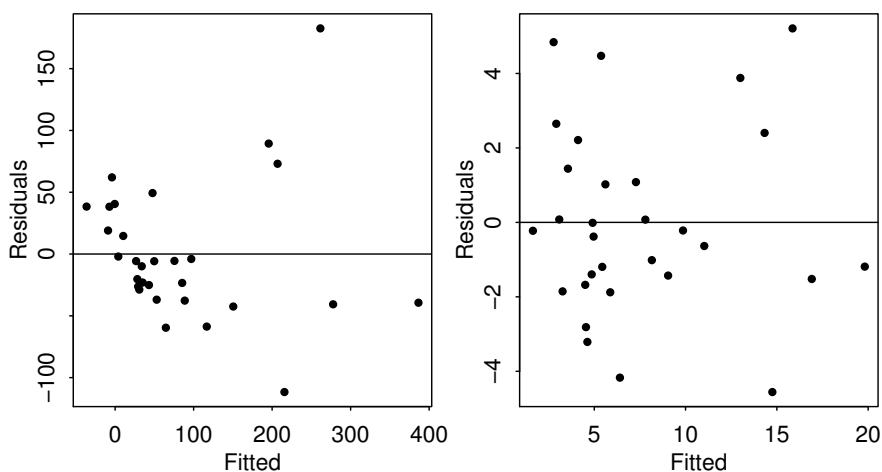


Figure 6.4 Residual vs. fitted plots for the Galápagos data before (left) and after (right) a square root transformation of the response.

data. The Poisson distribution is a good model for counts and that distribution has the property that the mean is equal to the variance, thus suggesting the square root transformation. We try that:

```
> lmod <- lm(sqrt(Species) ~ Area + Elevation + Scruz + Nearest +
  Adjacent, gala)
> plot(fitted(lmod), residuals(lmod), xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

We see in the second plot of Figure 6.4 that the variance is now constant and the signs of non-linearity have gone. Our guess at a variance stabilizing transformation worked out here, but had it not, we could always have tried something else.

6.1.2 Normality

The tests and confidence intervals we use are based on the assumption of normal errors. The residuals can be assessed for normality using a $Q-Q$ plot. This compares

the residuals to “ideal” normal observations. We plot the sorted residuals against $\Phi^{-1}\left(\frac{i}{n+1}\right)$ for $i = 1, \dots, n$. Let’s try it out on the savings data:

```
> lmod <- lm(sr ~ pop15+pop75+dpi+ddpi, savings)
> qqnorm(residuals(lmod), ylab="Residuals", main="")
> qqline(residuals(lmod))
```

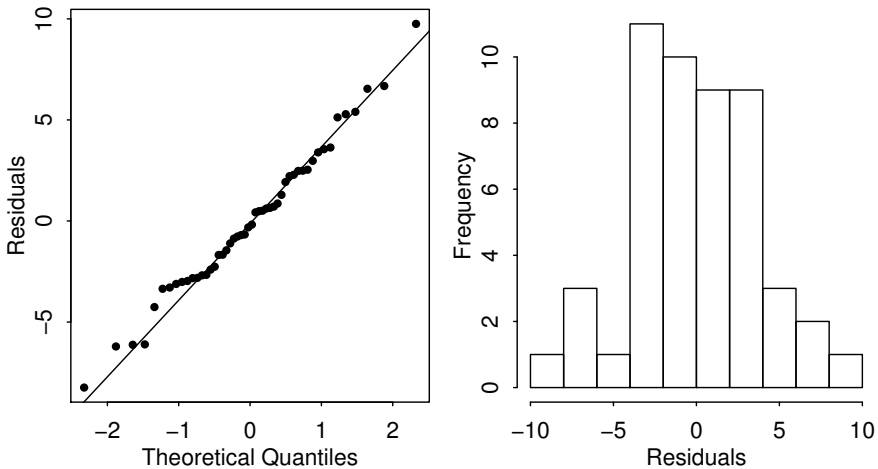


Figure 6.5 Normality checks for the savings data. A Q–Q plot is shown on the left and a histogram on the right.

See the first plot of Figure 6.5 — `qqline()` adds a line joining the first and third quartiles. It is not influenced by outliers. Normal residuals should follow the line approximately. Here, the residuals look normal.

Histograms and boxplots are not suitable for checking normality:

```
> hist(residuals(lmod), xlab="Residuals", main="")
```

The histogram seen in the second plot of Figure 6.5 does not have the expected bell shape. This is because we must group the data into bins. The choice of width and placement of these bins is problematic and the plot here is inconclusive.

We can get an idea of the variation to be expected in Q–Q plots in the following simulation. We generate data from different distributions:

1. Normal
2. Lognormal — an example of a skewed distribution
3. Cauchy — an example of a long-tailed (leptokurtic) distribution
4. Uniform — an example of a short-tailed (platykurtic) distribution

We generate nine replicates at a time from each of these test cases:

```
> par(mfrow=c(3, 3))
> n <- 50
```

```

> for(i in 1:9) {x <- rnorm(n); qqnorm(x); qqline(x)}
> for(i in 1:9) {x <- exp(rnorm(n)); qqnorm(x); qqline(x)}
> for(i in 1:9) {x <- rcauchy(n); qqnorm(x); qqline(x)}
> for(i in 1:9) {x <- runif(n); qqnorm(x); qqline(x)}
> par(mfrow=c(1,1))

```

In Figure 6.6, you can see examples of the latter three cases:

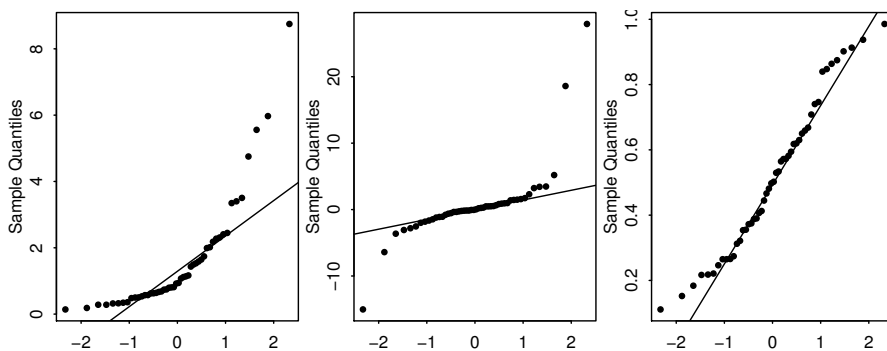


Figure 6.6 Q–Q plots of simulated data for, respectively, a skewed distribution, a long-tailed distribution and a short-tailed distribution.

It is not always easy to diagnose the problem in Q–Q plots. Sometimes extreme cases may be a sign of a long-tailed error like the Cauchy distribution or they can be just outliers. If removing such observations just results in other points becoming more prominent in the plot, the problem is likely due to a long-tailed error.

When the errors are not normal, least squares estimates may not be optimal. They will still be best linear unbiased estimates, but other *robust* estimators may be more effective. Also tests and confidence intervals are not exact. However, we can appeal to the central limit theorem which will ensure that the tests and confidence intervals constructed will be increasingly accurate approximations for larger sample sizes. Hence, we can afford to ignore the issue, provided the sample is sufficiently large or the violation not particularly severe.

When non-normality is found, the resolution depends on the type of problem found. For short-tailed distributions, the consequences of non-normality are not serious and can reasonably be ignored. For skewed errors, a transformation of the response can solve the problem. For long-tailed errors, we might just accept the non-normality and base the inference on the assumption of another distribution or use resampling methods such as the bootstrap or permutation tests. Alternatively, use robust methods, which give less weight to outlying observations but may again require resampling for the inference.

Also you may find that other diagnostics suggest changes to the model. In this changed model, the problem of non-normal errors might not occur so it is best to address any non-linearity and non-constant variance problems first.

The Shapiro–Wilk test is a formal test for normality:

```
> shapiro.test(residuals(lmod))

      Shapiro-Wilk normality test

data:  residuals(g)
W = 0.987, p-value = 0.8524
```

The null hypothesis is that the the residuals are normal. Since the p -value is large, we do not reject this hypothesis.

We can only recommend this in conjunction with a Q–Q plot at best. The p -value is not very helpful as an indicator of what action to take. With a large dataset, even mild deviations from non-normality may be detected, but there would be little reason to abandon least squares because the effects of non-normality are mitigated by large sample sizes. For smaller sample sizes, formal tests lack power.

6.1.3 Correlated Errors

It is difficult to check for correlated errors in general because there are just too many possible patterns of correlation that may occur. We do not have enough information to make any reasonable check. But some types of data have a structure which suggests where to look for problems. Data collected over time may have some correlation in successive errors. Spatial data may have correlation in the errors of nearby measurements. Data collected in blocks may show correlated errors within those blocks.

We illustrate the methods with an example of some temporal data. In general, the methods of time series analysis can be used but we will use methods more specific to the regression diagnostics problem. A combination of graphical and numerical methods is available.

The issue of global warming has attracted significant interest in recent years. Reliable records of annual temperatures taken with thermometers are only available back to the 1850s. Information about temperatures prior to this can be extracted from proxies such as tree rings. We can build a linear model to predict temperature since 1856 and then subsequently use this to predict earlier temperatures based on proxy information. The data we use here has been derived from Jones and Mann (2004). I have selected only proxies with mostly complete data. I have imputed some missing values and used some smoothing for ease of demonstration. Researchers seriously interested in historical climate should refer to the original data — see the R help page for details.

We start by fitting a model with the temperature as the response and the eight proxies as predictors. A full analysis would involve other steps but we focus on the issue of correlated errors here.

```
> data(globwarm, package="faraway")
> lmod <- lm(nhtemp ~ wusa + jasper + westgreen + chesapeake +
  tornetrask + urals + mongolia + tasman, globwarm)
```

There are missing values for `nhtemp` prior to 1856. The default behavior in R when performing a regression with missing values is to omit any case that contains a missing value. Hence this model uses only data from 1856 through 2000.

For temporal data such as these, it is sensible to plot the residuals against the time index. The plot is shown in the first panel of Figure 6.7. We have specified a missing valueless version of the data frame so that the years match up to the residuals.

```
> plot(residuals(lmod) ~ year, na.omit(globwarm), ylab="Residuals")
> abline(h=0)
```

If the errors were uncorrelated, we would expect a random scatter of points above and below the $\epsilon = 0$ line. In this plot, we see long sequences of points above or below the line. This is an indication of positive serial correlation. An alternative approach

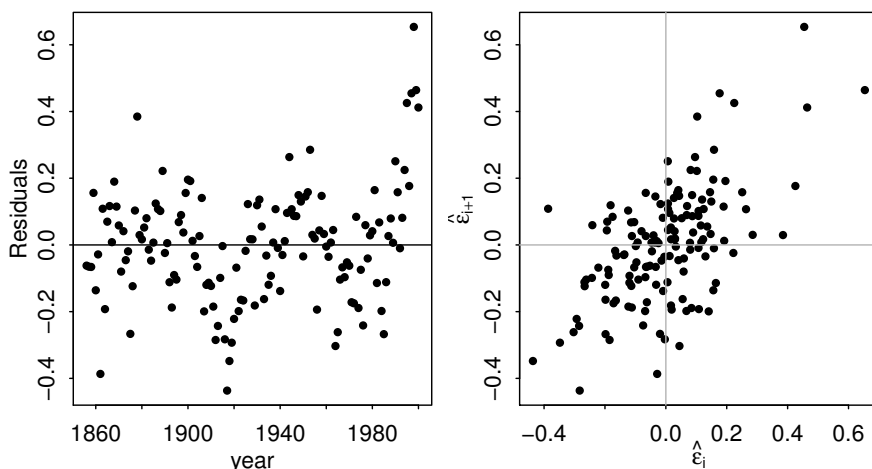


Figure 6.7 Diagnostic plots for correlated errors in the air quality data.

to checking for serial correlation is to plot successive pairs of residuals which we see in the second panel of Figure 6.7.

```
> n <- length(residuals(lmod))
> plot(tail(residuals(lmod), n-1) ~ head(residuals(lmod), n-1), xlab=
       expression(hat(epsilon)[i]), ylab=expression(hat(epsilon)[i+1]))
> abline(h=0, v=0, col=grey(0.75))
```

We can see a positive correlation again indicating positive serial correlation. If you have some doubt as to the significance of the correlation, we can model it directly:

```
> summary(lm(tail(residuals(lmod), n-1) ~ head(residuals(lmod), n-1) - 1))
              Estimate Std. Error t value Pr(>|t|)
head(residuals(lmod), n - 1)  0.5951    0.0693    8.59 1.4e-14
n = 144, p = 1, Residual SE = 0.139, R-Squared = 0.34
```

We omitted the intercept term because the residuals have mean zero. The serial correlation is confirmed. You can plot more than just successive pairs if you suspect a more complex dependence.

The Durbin–Watson test uses the statistic:

$$DW = \frac{\sum_{i=2}^n (\hat{\epsilon}_i - \hat{\epsilon}_{i-1})^2}{\sum_{i=1}^n \hat{\epsilon}_i^2}$$

The null hypothesis is that the errors are uncorrelated. The null distribution based on the assumption of uncorrelated errors follows a linear combination of χ^2 distributions. The test is implemented in the `lmtest` package. The run test is an alternative.

We can compute the Durbin–Watson statistic:

```
> require(lmtest)
> dwtest(nhtemp ~ wusa + jasper + westgreen + chesapeake + tornetrask
+ urals + mongolia + tasman, data=globwarm)

Durbin-Watson test

data:  nhtemp ~ wusa + jasper + westgreen + chesapeake + tornetrask +
      urals + mongolia + tasman
DW = 0.8166, p-value = 1.402e-15
alternative hypothesis: true autocorrelation is greater than 0
```

where the p-value indicates strong evidence of serial correlation. In this example, we are not surprised to see that the errors are correlated. We do not expect that the proxies will model temperature perfectly and that higher or lower temperatures in one year might carry over to the next.

Sometimes, serial correlation can be caused by a missing covariate. For example, suppose that there is a quadratic relation between a predictor and the response but we use only a linear term in that predictor. The diagnostics will show serial correlation in the residuals but the real source of the problem is the missing quadratic term.

Although it is possible that difficulties with correlated errors can be removed by changing the structural part of the model, sometimes we must build the correlation directly into the model. This can be achieved using the method of generalized least squares — see Chapter 8.

6.2 Finding Unusual Observations

We may find that some observations do not fit the model well. Such points are called *outliers*. Other observations change the fit of the model in a substantive manner. These are called *influential* observations. It is possible for a point to have both these characteristics. A *leverage* point is extreme in the predictor space. It has the potential to influence the fit, but does not necessarily do so. It is important to first identify such points. Deciding what to do about them can be difficult.

6.2.1 Leverage

$h_i = H_{ii}$ are called *leverages* and are useful diagnostics. Since $\text{var } \hat{\epsilon}_i = \sigma^2(1 - h_i)$, a large leverage, h_i , will make $\text{var } \hat{\epsilon}_i$ small. The fit will be attracted toward y_i . Large values of h_i are due to extreme values in the X -space. h_i corresponds to a (squared)

Mahalanobis distance defined by X which is $(x - \bar{x})^T \hat{\Sigma}^{-1} (x - \bar{x})$ where $\hat{\Sigma}$ is the estimated covariance of X . The value of h_i depends only on X and not y so leverages contain only partial information about a case.

Since $\sum_i h_i = p$, an average value for h_i is p/n . A rough rule is that leverages of more than $2p/n$ should be looked at more closely.

We will use the savings dataset as an example here:

```
> lmod <- lm(sr ~ pop15 + pop75 + dpi + ddpi, savings)
> hatv <- hatvalues(lmod)
> head(hatv)
Australia  Austria    Belgium    Bolivia    Brazil    Canada
0.067713   0.120384   0.087482   0.089471   0.069559   0.158402
> sum(hatv)
[1] 5
```

We verify that the sum of the leverages is indeed five — the number of parameters in the model.

Without making assumptions about the distributions of the predictors that would often be unreasonable, we cannot say how the leverages would be distributed. Nevertheless, we would like to identify unusually large values of the leverage. The half-normal plot is a good way to do this.

Half-normal plots are designed for the assessment of positive data. They could be used for $|\hat{\epsilon}_i|$, but are more typically useful for diagnostic quantities like the leverages. The idea is to plot the data against the positive normal quantiles.

The steps are:

1. Sort the data: $x_{[1]} \leq \dots x_{[n]}$.
2. Compute $u_i = \Phi^{-1}(\frac{n+i}{2n+1})$.
3. Plot $x_{[i]}$ against u_i .

We do not usually expect a straight line relationship since we do not necessarily expect a positive normal distribution for quantities like leverages. We are looking for outliers, which will be apparent as points that diverge substantially from the rest of the data.

We demonstrate the half-normal plot on the leverages for the savings data:

```
> countries <- row.names(savings)
> halfnorm(hatv, labs=countries, ylab="Leverages")
```

The plot is the first shown in Figure 6.8. I have plotted the country name instead of just a dot for the largest two cases, respectively, to aid identification. Leverages can also be used in scaling residuals. We have $\text{var } \hat{\epsilon}_i = \sigma^2(1 - h_i)$ which suggests the use of:

$$r_i = \frac{\hat{\epsilon}_i}{\hat{\sigma}\sqrt{1 - h_i}}$$

These are called *standardized residuals*. If the model assumptions are correct, $\text{var } r_i = 1$ and $\text{corr}(r_i, r_j)$ tend to be small. Standardized residuals are sometimes preferred in residual plots, as they have been standardized to have equal variance. Standardization can only correct for the natural non-constant variance in residuals

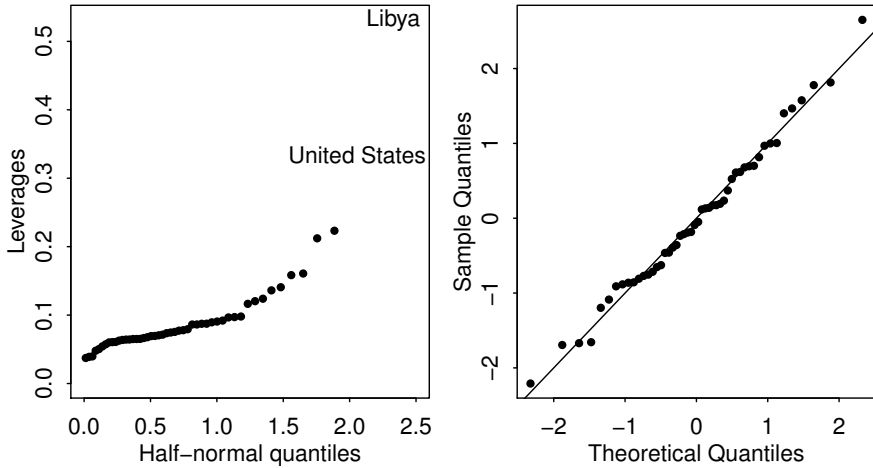


Figure 6.8 Half-normal plots for the leverages and a Q–Q plot for the standardized residuals.

when the errors have constant variance. If there is some underlying heteroscedasticity in the errors, standardization cannot correct for it. We now compute and plot the standardized residuals for the savings data:

```
> qqnorm(rstandard(lmod))
> abline(0, 1)
```

We have displayed the Q–Q plot of the standardized residuals in the second plot of Figure 6.8. Because these residuals have been standardized, we expect the points to approximately follow the $y = x$ line if normality holds. Another advantage of the standardized form is that we can judge the size easily. An absolute value of 2 would be large but not exceptional for a standardized residual whereas a value of 4 would be very unusual under the standard normal.

Some authors recommend using standardized rather than raw residuals in all diagnostic plots. However, in many cases, the standardized residuals are not very different from the raw residuals except for the change in scale. Only when there are unusually large leverages will the differences be noticeable in the shape of the plot.

6.2.2 Outliers

An outlier is a point that does not fit the current model well. We need to be aware of such exceptions. An outlier test is useful because it enables us to distinguish between truly unusual observations and residuals that are large, but not exceptional. Outliers may or may not affect the fit substantially. We simulate some data to illustrate the possibilities:

```
> set.seed(123)
> testdata <- data.frame(x=1:10, y=1:10+rnorm(10))
```

```
> lmod <- lm(y ~ x, testdata)
```

The first example adds an outlier with a central predictor value:

```
> p1 <- c(5.5, 12)
> lmod1 <- lm(y ~ x, rbind(testdata, p1))
> plot(y ~ x, rbind(testdata, p1))
> points(5.5, 12, pch=4, cex=2)
> abline(lmod)
> abline(lmod1, lty=2)
```

The plot, seen in the first panel of Figure 6.9, shows a solid regression line for the fit without the additional point marked with a cross. The dashed line shows the fit with the extra point. There is not much difference. In particular, the slopes are very similar. This is an example of a point which is an outlier, but does not have large leverage or influence.

The second example introduces an extra point well outside the range of X :

```
> p2 <- c(15, 15.1)
> lmod2 <- lm(y ~ x, rbind(testdata, p2))
> plot(y ~ x, rbind(testdata, p2))
> points(15, 15.1, pch=4, cex=2)
> abline(lmod)
> abline(lmod2, lty=2)
```

The plot, seen in the second panel of Figure 6.9, shows that the additional point makes little difference to the fitted regression line. This point has large leverage but is not an outlier and is not influential.

The third example puts the point in a different position on the response scale.

```
> p3 <- c(15, 5.1)
> lmod3 <- lm(y ~ x, rbind(testdata, p3))
> plot(y ~ x, rbind(testdata, p3))
> points(15, 5.1, pch=4, cex=2)
> abline(lmod)
> abline(lmod3, lty=2)
```

The plot, seen in the third panel of Figure 6.9, shows that this additional point changes the regression line substantially. Although this point still has a large residual, the residuals for the other points are also made much larger with the addition of this point. Hence this point is both an outlier and an influential point. We must take care to discover such points because they can have a substantial effect on the conclusions of the analysis. Just looking at $\hat{\epsilon}_i$ misses difficult observations, like that seen in the third example, which pull the regression line closer so that they conceal their true status. To detect such points, we exclude point i and recompute the estimates to get $\hat{\beta}_{(i)}$ and $\hat{\sigma}_{(i)}^2$ where (i) denotes that the i^{th} case has been excluded. Hence:

$$\hat{y}_{(i)} = x_i^T \hat{\beta}_{(i)}$$

If $\hat{y}_{(i)} - y_i$ is large, then case i is an outlier. To judge the size of a potential outlier, we need an appropriate scaling. We find:

$$\hat{\text{var}}(y_i - \hat{y}_{(i)}) = \hat{\sigma}_{(i)}^2 (1 + x_i^T (X_{(i)}^T X_{(i)})^{-1} x_i)$$

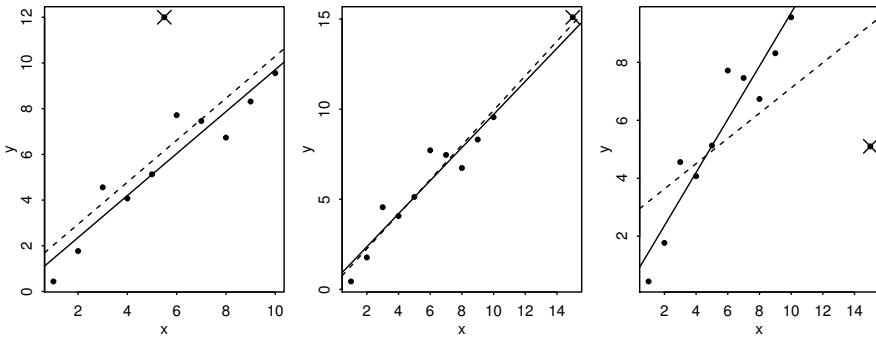


Figure 6.9 In all three plots, the additional point is marked with a cross. The solid line is the fit using the 10 original points and the dashed line is the fit with the added point.

and so we define the studentized (sometimes called jackknife or crossvalidated) residuals as:

$$t_i = \frac{y_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}(1 + x_i^T(X_{(i)}^T X_{(i)})^{-1}x_i)^{1/2}}$$

which are distributed t_{n-p-1} if the model is correct and $\epsilon \sim N(0, \sigma^2 I)$. Fortunately, there is an easier way to compute t_i :

$$t_i = \frac{\hat{\epsilon}_i}{\hat{\sigma}_{(i)}\sqrt{1 - h_i}} = r_i \left(\frac{n - p - 1}{n - p - r_i^2} \right)^{1/2}$$

which avoids doing n regressions.

Since $t_i \sim t_{n-p-1}$, we can calculate a p -value to test whether case i is an outlier. This is fine if we only test one preselected case. However, if we had $n = 100$ and tested all the cases, we would expect to find around five outliers using this procedure if we used a 5% significance level. Even though we might explicitly test only one or two large t_i s, we are implicitly testing all cases since we need to consider all the residuals to find out which ones are large. Some adjustment of the level of the test is necessary to avoid identifying an excess of outliers.

Suppose we want a level α test. Now $P(\text{all tests accept}) = 1 - P(\text{at least one rejects}) \geq 1 - \sum_i P(\text{test } i \text{ rejects}) = 1 - n\alpha$. So this suggests that if an overall level α test is required, then a level α/n should be used in each of the tests. This method is called the *Bonferroni correction* and is used in contexts other than outliers. Its biggest drawback is that it is conservative — it finds fewer outliers than the nominal level of confidence would dictate. The larger that n is, the more conservative it gets.

Let's compute studentized residuals for the savings data and pick out the largest:

```
> stud <- rstudent(lmod)
> stud[which.max(abs(stud))]
Zambia
2.8536
```

The largest residual of 2.85 is rather large for a standard normal scale, but is it an outlier? Compute the Bonferroni critical value:

```
> qt (.05 / (50*2), 44)
[1] -3.5258
```

Since 2.85 is less than 3.53, we conclude that Zambia is *not* an outlier. For simple regression, the minimum critical value occurs at $n = 23$ taking the value 3.51. This indicates that it is not worth the trouble of computing the outlier test p -value unless the studentized residual exceeds about 3.5 in absolute value.

Some points to consider about outliers:

1. Two or more outliers next to each other can hide each other.
2. An outlier in one model may not be an outlier in another when the variables have been changed or transformed. You will usually need to reinvestigate the question of outliers when you change the model.
3. The error distribution may not be normal and so larger residuals may be expected. For example, day-to-day changes in stock indices seem mostly normal, but larger changes occur from time to time.
4. Individual outliers are usually much less of a problem in larger datasets. A single point will not have the leverage to affect the fit very much. It is still worth identifying outliers if these types of observations are worth knowing about in the particular application. For large datasets, we need only to worry about clusters of outliers. Such clusters are less likely to occur by chance and more likely to represent actual structure. Finding these clusters is not always easy.

What should be done about outliers?

1. Check for a data-entry error first. These are relatively common. Unfortunately, the original source of the data may have been lost or may be inaccessible. If you can be sure that the point is truly a mistake and was not actually observed, then the solution is simple: discard it.
2. Examine the physical context — why did it happen? Sometimes, the discovery of an outlier may be of singular interest. Some scientific discoveries spring from noticing unexpected aberrations. Another example of the importance of outliers is in the statistical analysis of credit card transactions. Outliers may represent fraudulent use. In such examples, the discovery of outliers is the main purpose of the modeling.
3. Exclude the point from the analysis but try reincluding it later if the model is changed. The exclusion of one or more observations may make the difference between getting a statistically significant result or having some unpublishable research. This can lead to a difficult decision about what exclusions are reasonable. To avoid any suggestion of dishonesty, always report the existence of outliers even if you do not include them in your final model. Be aware that casual or dishonest exclusion of outliers is regarded as serious research malpractice.
4. Suppose you find outliers that cannot reasonably be identified as mistakes or aberrations, but are viewed as naturally occurring. Particularly if there are substantial

numbers of such points, it is more efficient and reliable to use robust regression, as explained in Section 8.4. Routine outlier rejection in conjunction with least squares is not a good method of estimation. Some adjustment to the inferential methods is necessary in such circumstances. In particular, the uncertainty assessment for prediction needs to reflect the fact that extreme values can occur.

5. It is dangerous to exclude outliers in an automatic manner. National Aeronautics and Space Administration (NASA) launched the *Nimbus 7* satellite to record atmospheric information. After several years of operation in 1985, the British Antarctic Survey observed a large decrease in atmospheric ozone over the Antarctic. On further examination of the NASA data, it was found that the data processing program automatically discarded observations that were extremely low and assumed to be mistakes. Thus the discovery of the Antarctic ozone hole was delayed several years. Perhaps, if this had been known earlier, the chlorofluorocarbon (CFC) phaseout would have been agreed upon earlier and the damage could have been limited. See Stolarski et al. (1986) for more.

Here is an example of a dataset with multiple outliers. Data are available on the log of the surface temperature and the log of the light intensity of 47 stars in the star cluster CYG OB1, which is in the direction of Cygnus. These data appear in Rousseeuw and Leroy (1987). Read in and plot the data:

```
> data(star, package="faraway")
> plot(star$temp, star$light, xlab="log(Temperature)", ylab="log(Light
  Intensity)")
```

There appears to be a positive correlation between temperature and light intensity, but there are four stars that do not fit the pattern. We fit a linear regression and add the fitted line to the plot:

```
> lmod <- lm(light ~ temp, star)
> abline(lmod)
```

The plot is seen in Figure 6.10 with the regression line in solid type. This line does not follow the bulk of the data because it tries to fit the four unusual points. We check whether the outlier test detects these points:

```
> range(rstudent(lmod))
[1] -2.0494 1.9058
```

No outliers are found even though we can see them clearly in the plot. The four stars on the upper left of the plot are giants. See what happens if these are excluded:

```
> lmodi <- lm(light ~ temp, data=star, subset=(temp>3.6))
> abline(lmodi, lty=2)
```

This illustrates the problem of multiple outliers. We can visualize the problems here and take corrective action, but for higher dimensional data this is much more difficult. Robust regression methods would be superior here. See Section 8.4.

6.2.3 Influential Observations

An influential point is one whose removal from the dataset would cause a large change in the fit. An influential point may or may not be an outlier and may or

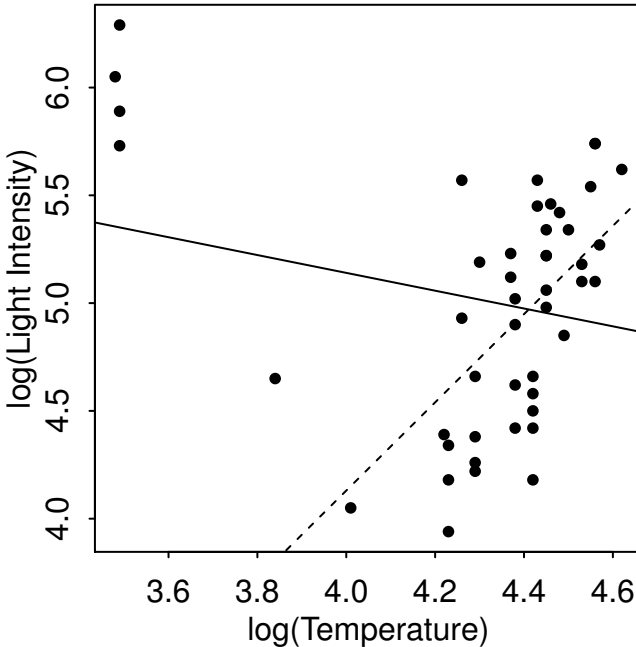


Figure 6.10 Regression line including four leftmost points is solid and excluding these points is dotted.

may not have large leverage but it will tend to have at least one of these two properties. In Figure 6.9, the third panel shows an influential point but in the second panel, the added point is not influential.

There are several measures of influence. A subscripted (i) indicates the fit without case i . We might consider the change in the fit $X^T(\hat{\beta} - \hat{\beta}_{(i)}) = \hat{y} - \hat{y}_{(i)}$, but there will be n of these length n vectors to examine. For a more compact diagnostic, we might consider the change in the coefficients $\hat{\beta} - \hat{\beta}_{(i)}$. There will be $n \times p$ of these to look at. The Cook statistics are popular influence diagnostics because they reduce the information to a single value for each case. They are defined as:

$$\begin{aligned} D_i &= \frac{(\hat{y} - \hat{y}_{(i)})^T (\hat{y} - \hat{y}_{(i)})}{p \hat{\sigma}^2} \\ &= \frac{1}{p} r_i^2 \frac{h_i}{1 - h_i} \end{aligned}$$

The first term, r_i^2 , is the residual effect and the second is the leverage. The combination of the two leads to influence. A half-normal plot of D_i can be used to identify influential observations.

Continuing with our study of the savings data:

```
> lmod <- lm(sr ~ pop15+pop75+dpi+ddpi, savings)
> cook <- cooks.distance(lmod)
> halfnorm(cook,3, labs=countries, ylab="Cook's distances")
```

The Cook statistics may be seen in the first plot of Figure 6.11. I have identified the largest three values. We now exclude the largest one (Libya) and see how the fit

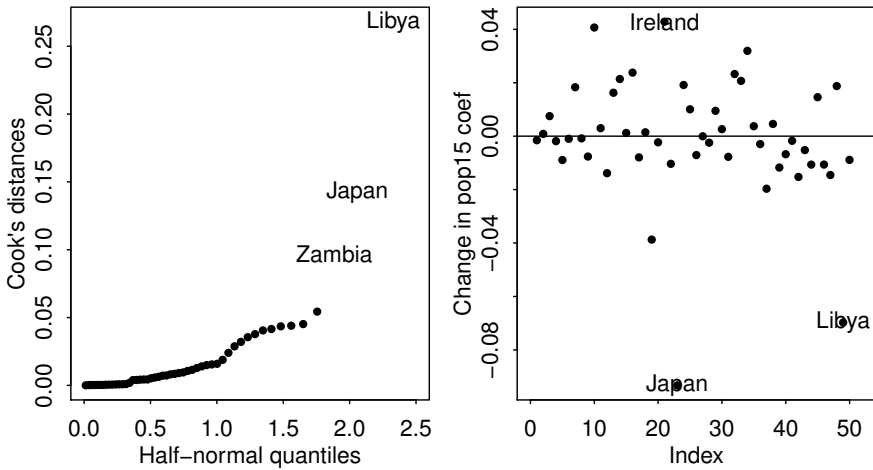


Figure 6.11 Half-normal plot of the Cook statistics and $\hat{\beta} - \hat{\beta}_{(i)}$'s for pop15 for the savings data.

changes:

```
> lmodi <- lm(sr ~ pop15+pop75+dpi+ddpi, savings, subset=(cook < max(
  cook)))
> summary(lmodi)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	24.524046	8.224026	2.98	0.0047
pop15	-0.391440	0.157909	-2.48	0.0171
pop75	-1.280867	1.145182	-1.12	0.2694
dpi	-0.000319	0.000929	-0.34	0.7331
ddpi	0.610279	0.268778	2.27	0.0281

n = 49, p = 5, Residual SE = 3.795, R-Squared = 0.36

Compared to the full data fit:

```
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	28.566087	7.354516	3.88	0.00033
pop15	-0.461193	0.144642	-3.19	0.00260
pop75	-1.691498	1.083599	-1.56	0.12553
dpi	-0.000337	0.000931	-0.36	0.71917
ddpi	0.409695	0.196197	2.09	0.04247

n = 50, p = 5, Residual SE = 3.803, R-Squared = 0.34

Among other changes, we see that the coefficient for `ddpi` changed by about 50%. We do not like our estimates to be so sensitive to the presence of just one country. It would be rather tedious to leave out each country in turn, so we examine the leave-out-one differences in the coefficients:

```
> plot(dfbeta(lmod)[,2],ylab="Change in pop15 coef")
> abline(h=0)
```

We have plotted the change in the second parameter estimate, $(\hat{\beta}_{pop15})$ when a case is left out, as seen in the second panel of Figure 6.11. The `identify` function allows interactive identification of points by clicking the left mouse button on the plot and then using the middle mouse button to finish. This plot should be repeated for the other variables. Japan sticks out on this particular plot so we examine the effect of removing it:

```
> lmodj <- lm(sr ~ pop15+pop75+dpi+ddpi,savings,subset=(countries != "
  Japan"))
> summary(lmodj)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	23.940171	7.783997	3.08	0.0036
pop15	-0.367901	0.153630	-2.39	0.0210
pop75	-0.973674	1.155450	-0.84	0.4040
dpi	-0.000471	0.000919	-0.51	0.6112
ddpi	0.334749	0.198446	1.69	0.0987

```
n = 49, p = 5, Residual SE = 3.738, R-Squared = 0.28
```

Comparing this to the full data fit, we observe several qualitative changes. Notice that the `ddpi` term is no longer significant and that the R^2 value has decreased a lot.

R provides an easy way to produce a useful selection of residual plots as seen in Figure 6.12. The first plot shows the usual residual vs. fitted plot with the addition of smoothed fitted line to aid interpretation. The second plot is the Q-Q plot of the standardized residuals. A few more extreme cases have been labeled. The third plot shows $\sqrt{|\hat{\epsilon}|}$ against \hat{y} , again with an added smoothed fit. The fourth plot shows the standardized residuals against the leverage. As the Cook statistics represent a function of these two variables, we can plot contours. Any point that lies beyond these contours might well be influential and require closer attention.

```
> plot(lmod)
```

6.3 Checking the Structure of the Model

Diagnostics can also be used to detect deficiencies in the structural part of the model, given by $EY = X\beta$. The residuals are the best clues because signs of remaining systematic structure here indicate that something is wrong. A good diagnostic can often also suggest how the model can be improved. Formal lack of fit tests can be used in limited circumstances, usually requiring replication. See Section 8.3. But such tests, even if they are available, do not indicate how to improve the model.

We have already discussed plots of $\hat{\epsilon}$ against \hat{y} and x_i . We have used these plots to check the assumptions on the error, but they can also suggest transformations of the variables which might improve the structural form of the model.

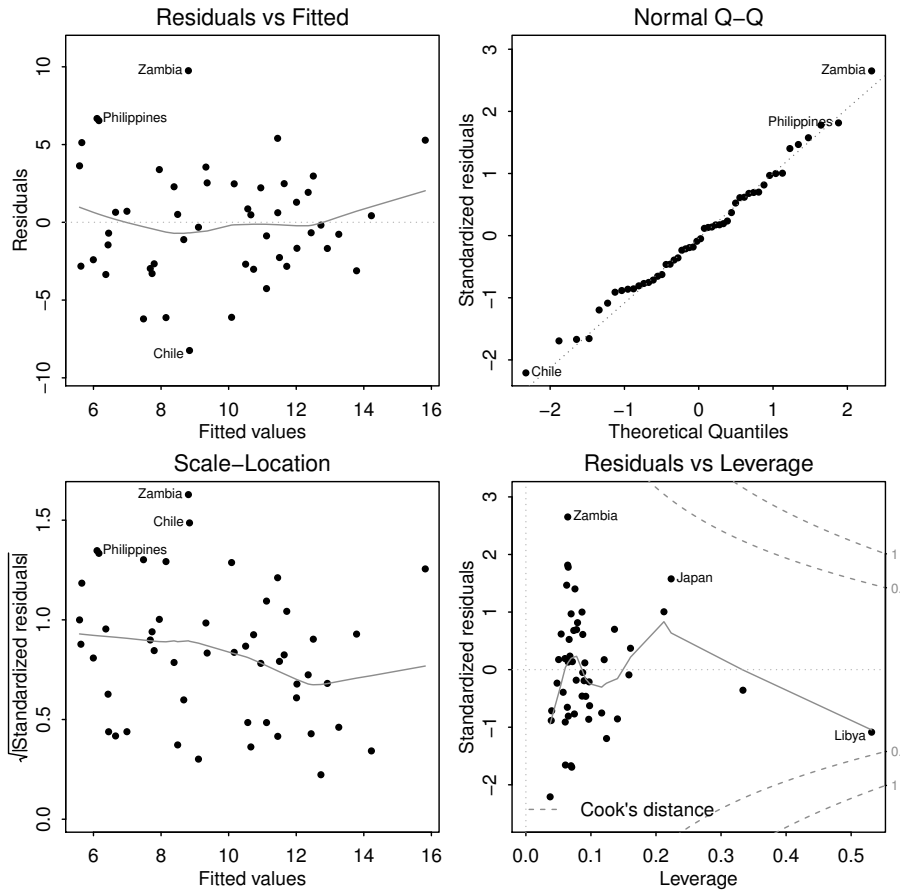


Figure 6.12 Diagnostic plots for the savings data

We can also make plots of y against each x_i . Indeed these should be part of any exploratory analysis before model fitting begins. The drawback to these plots is that the other predictors often affect the relationship between a given predictor and the response. *Partial regression* or *added variable* plots can help isolate the effect of x_i on y . We regress y on all x except x_i , and get residuals $\hat{\delta}$. These represent y with the other X -effect taken out. Similarly, if we regress x_i on all x except x_i , and get residuals $\hat{\gamma}$, we have the effect of x_i with the other X -effect taken out. The added variable plot shows $\hat{\delta}$ against $\hat{\gamma}$. Look for non-linearity and outliers and/or influential observations in the plot.

We illustrate using the savings dataset as an example again. We construct a partial regression (added variable) plot for `pop15`:

```
> d <- residuals(lm(sr ~ pop75 + dpi + ddpi, savings))
> m <- residuals(lm(pop15 ~ pop75 + dpi + ddpi, savings))
> plot(m, d, xlab="pop15 residuals", ylab="Savings residuals")
```

The plot, shown in the first panel of Figure 6.13, shows nothing remarkable. There is no sign of non-linearity or unusual points. An interesting feature of such plots is revealed by considering the regression line. We compute this for the plot and notice that it is the same as the corresponding coefficient from the full regression:

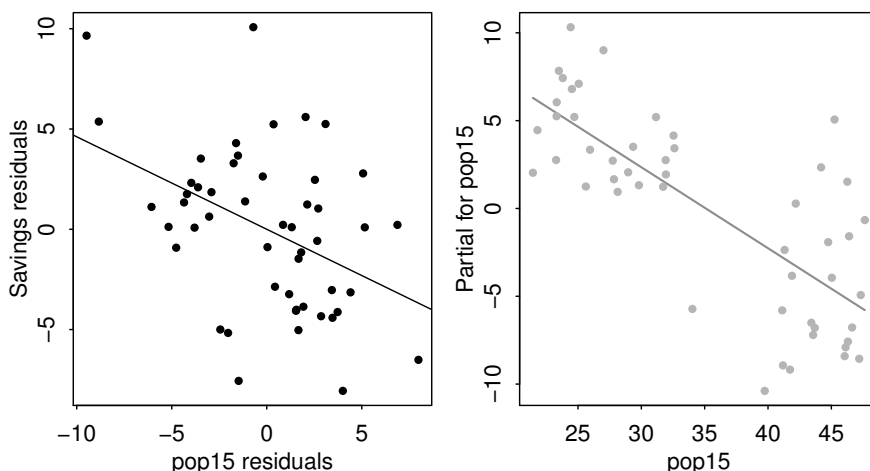


Figure 6.13 Partial regression (left) and partial residual (right) plots for the savings data.

```
> coef(lm(d ~ m))
(Intercept)          m
5.4259e-17 -4.6119e-01
> lmod <- lm(sr ~ pop15+pop75+dpi+ddpi, savings)
> coef(lmod)
(Intercept)    pop15    pop75    dpi    ddpi
28.5660865  -0.4611931  -1.6914977  -0.0003369  0.4096949
> abline(0, coef(lmod) ['pop15'])
```

The partial regression plot provides some intuition about the meanings of regression coefficients. We are looking at the marginal relationship between the response and the predictor after the effect of the other predictors has been removed. Multiple regression is difficult because we cannot visualize the full relationship because of the high dimensionality. The partial regression plot allows us to focus on the relationship between one predictor and the response, much as in simple regression.

Partial residual plots serve as an alternative to added variable plots. We construct the response with the predicted effect of the other X removed:

$$y - \sum_{j \neq i} x_j \hat{\beta}_j = \hat{y} + \hat{\epsilon} - \sum_{j \neq i} x_j \hat{\beta}_j = x_i \hat{\beta}_i + \hat{\epsilon}$$

The partial residual plot is then $\hat{\epsilon} + \hat{\beta}_i x_i$ against x_i . Again the slope on the plot will be $\hat{\beta}_i$ and the interpretation is the same. Partial residual plots are believed to be better

for non-linearity detection while added variable plots are better for outlier/influential detection.

A partial residual plot is easy to construct. The `termplot` centers the x_i so that $\hat{\epsilon} + \hat{\beta}_i(x_i - \bar{x}_i)$ can be called partial residuals and have mean zero.

```
> termplot(lmod, partial.resid=TRUE, terms=1)
```

We see two groups in the plot, seen in the second panel of 6.13. It suggests that there may be a different relationship in the two groups. We investigate this:

```
> mod1 <- lm(sr ~ pop15+pop75+dpi+ddpi, savings, subset=(pop15 > 35))
> mod2 <- lm(sr ~ pop15+pop75+dpi+ddpi, savings, subset=(pop15 < 35))
> summary(mod1)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.433969	21.155028	-0.12	0.91
pop15	0.273854	0.439191	0.62	0.54
pop75	-3.548477	3.033281	-1.17	0.26
dpi	0.000421	0.005000	0.08	0.93
ddpi	0.395474	0.290101	1.36	0.19

```
n = 23, p = 5, Residual SE = 4.454, R-Squared = 0.16
> summary(mod2)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	23.961795	8.083750	2.96	0.0072
pop15	-0.385898	0.195369	-1.98	0.0609
pop75	-1.327742	0.926063	-1.43	0.1657
dpi	-0.000459	0.000724	-0.63	0.5326
ddpi	0.884394	0.295341	2.99	0.0067

```
n = 27, p = 5, Residual SE = 2.772, R-Squared = 0.51
```

In the first regression on the subset of underdeveloped countries, we find no relation between the predictors and the response. We know from our previous examination of these data that this result is not attributable to outliers or unsuspected transformations. In contrast, there is a strong relationship in the developed countries. The strongest predictor is growth together with some relationship to proportion of population under 15. This latter effect has been reduced from prior analyses because we have reduced the range of this predictor by the subsetting operation. The graphical analysis has shown a relationship in the data that a purely numerical analysis might easily have missed.

Sometimes it can be helpful to introduce extra dimensions into diagnostic plots with the use of color, plotting symbol or size. Alternatively, plots can be faceted. The `ggplot2` package is convenient for this purpose. Here are a couple of examples:

```
> savings$status <- ifelse(savings$pop15 > 35, "young", "old")
> require(ggplot2)
> ggplot(savings, aes(x=ddpi, y=sr, shape=status)) + geom_point()
> ggplot(savings, aes(x=ddpi, y=sr)) + geom_point() + facet_grid(~ status) +
  stat_smooth(method="lm")
```

The plots in Figure 6.14 show how we can distinguish the two levels of the status variable derived from the population proportion under 15 in two different ways. In this case, the second set of plots is more effective. Note how we have added a re-

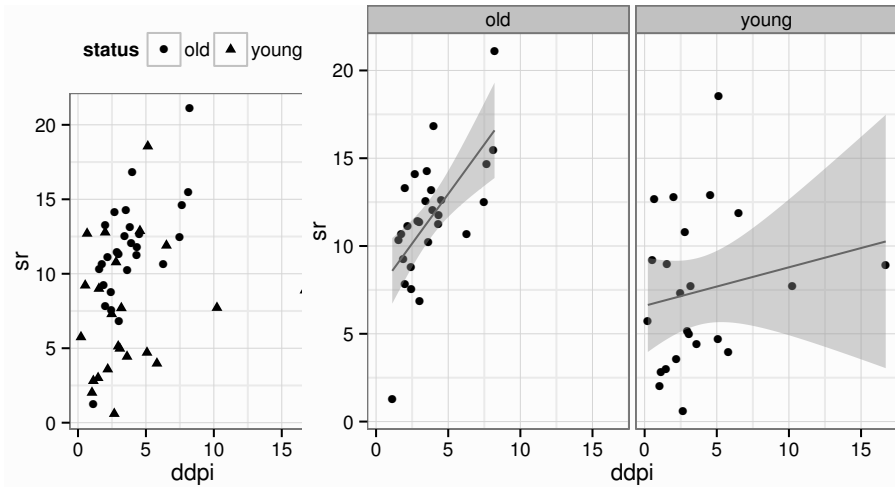


Figure 6.14 Introducing another dimension to diagnostic plots. Shape is used denote the status variable on the left while faceting is used on the right.

gression line with 95% confidence bands, making it clear how the relationship differs between the two groups.

Higher dimensional plots can also be useful for detecting structures that cannot be seen in two dimensions. These are interactive in nature so you need to try them to see how they work. We can make three-dimensional plots where color, point size and rotation are used to give the illusion of a third dimension. We can also link two or more plots so that points which are *brushed* in one plot are highlighted in another. These tools look good but it is not clear whether they actually are useful in practice. There are communication difficulties, as these plots cannot be easily printed. R does not have such tools, but GGobi is a useful free tool for exploring higher dimensional data that can be called from R. See www.ggobi.org.

Non-graphical techniques for checking the structural form of the model usually involve proposing alternative transformations or recombinations of the variables. This approach is explored in Chapter 9.

6.4 Discussion

Some assumptions are more important than others because some violations are more serious in that they can lead to very misleading conclusions. We can order these assumptions according to their importance:

1. The systematic form of the model. If you get this seriously wrong, then predictions will be inaccurate and any explanation of the relationship between the variables may be biased in misleading ways.
2. Dependence of errors. The presence of strong dependence means that there is less information in the data than the sample size may suggest. Furthermore, there is a

risk that the analyst will mistakenly introduce systematic components to the model in an attempt to deal with an unsuspected dependence in the errors. Unfortunately, it is difficult to detect dependence in the errors using regression diagnostics except in special situations such as temporal data. For other types of data, the analyst will need to rely on less testable assumptions about independence based on contextual knowledge.

3. Nonconstant variance. A failure to address this violation of the linear model assumptions may result in inaccurate inferences. In particular, prediction uncertainty may not be properly quantified. Even so, excepting serious violations, the adequacy of the inference may not be seriously compromised.
4. Normality. This is the least important assumption. For large datasets, the inference will be quite robust to a lack of normality as the central limit theorem will mean that the approximations will tend to be adequate. Unless the sample size is quite small or the errors very strongly abnormal, this assumption is not crucial to success.

Although it is not part of regression diagnostics, it is worthwhile also mentioning that an even more important assumption is that the data at hand are relevant to the question of interest. This requires some qualitative judgement and is not checkable by plots or tests.

Excercises

1. Using the `sat` dataset, fit a model with the total SAT score as the response and `expend`, `salary`, `ratio` and `takers` as predictors. Perform regression diagnostics on this model to answer the following questions. Display any plots that are relevant. Do not provide any plots about which you have nothing to say. Suggest possible improvements or corrections to the model where appropriate.
 - (a) Check the constant variance assumption for the errors.
 - (b) Check the normality assumption.
 - (c) Check for large leverage points.
 - (d) Check for outliers.
 - (e) Check for influential points.
 - (f) Check the structure of the relationship between the predictors and the response.
2. Using the `teengamb` dataset, fit a model with `gamble` as the response and the other variables as predictors. Answer the questions posed in the previous question.
3. For the `prostate` data, fit a model with `lpsa` as the response and the other variables as predictors. Answer the questions posed in the first question.
4. For the `swiss` data, fit a model with `Fertility` as the response and the other variables as predictors. Answer the questions posed in the first question.
5. Using the `cheddar` data, fit a model with `taste` as the response and the other three variables as predictors. Answer the questions posed in the first question.

6. Using the `happy` data, fit a model with `happy` as the response and the other four variables as predictors. Answer the questions posed in the first question.
7. Using the `tvdoctor` data, fit a model with `life` as the response and the other two variables as predictors. Answer the questions posed in the first question.
8. For the `divusa` data, fit a model with `divorce` as the response and the other variables, except `year` as predictors. Check for serial correlation.

Problems with the Predictors

7.1 Errors in the Predictors

The regression model $Y = X\beta + \varepsilon$ allows for Y being measured with error by having the ε term, but what if the X is measured with error? In other words, what if the X we see is not exactly the X that potentially drives Y ? It is not unreasonable that there might be errors in measuring X . For example, consider the problem of determining the effects of being exposed to a potentially hazardous substance such as second-hand tobacco smoke. Such exposure would be a predictor in such a study, but it is very hard to measure this exactly over a period of years.

One should not confuse errors in predictors with treating X as a random variable. For observational data, X could be regarded as a random variable, but the regression inference proceeds conditionally on a fixed value for X . For example, suppose we conducted a study where the blood sugar level of the participants was a predictor. Since the participants are sampled from a larger population, the blood sugar level is a random variable. However, once we select the participants and measure the blood sugar levels, these become fixed for the purposes of the regression. But our interest in this section is the situation where we are not able to measure the blood sugar levels accurately. In this case, the value we record will not be the same as the value that possibly affects the response.

Suppose that what we observe is (x_i^O, y_i^O) for $i = 1, \dots, n$ which are related to the true values (x_i^A, y_i^A) :

$$\begin{aligned} y_i^O &= y_i^A + \varepsilon_i \\ x_i^O &= x_i^A + \delta_i \end{aligned}$$

where the errors ε and δ are independent. The situation is depicted in Figure 7.1. The true underlying relationship is:

$$y_i^A = \beta_0 + \beta_1 x_i^A$$

but we only see (x_i^O, y_i^O) . Putting it together, we get:

$$y_i^O = \beta_0 + \beta_1 x_i^O + (\varepsilon_i - \beta_1 \delta_i)$$

Suppose we use least squares to estimate β_0 and β_1 . Let's assume $E\varepsilon_i = E\delta_i = 0$ and that $\text{var } \varepsilon_i = \sigma_\varepsilon^2$, $\text{var } \delta_i = \sigma_\delta^2$. Let:

$$\sigma_x^2 = \sum (x_i^A - \bar{x}^A)^2 / n \quad \sigma_{x\delta} = \text{cov}(x^A, \delta)$$

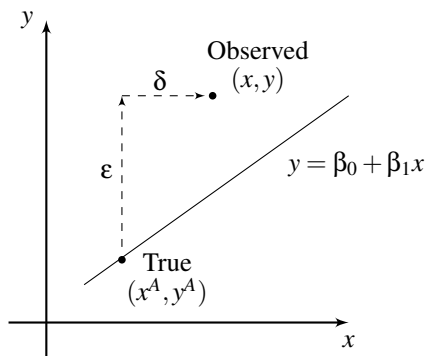


Figure 7.1 Measurement error: true vs. observed data.

For observational data, σ_x^2 is (almost) the sample variance of X^A while for a controlled experiment we can view it as just a numerical measure of the spread of the design. A similar distinction should be made for $cov(x^A, \delta)$ although in many cases, it will be reasonable to assume that this is zero.

Now $\hat{\beta}_1 = \sum(x_i - \bar{x})y_i / \sum(x_i - \bar{x})^2$ and after some calculation we find that:

$$E\hat{\beta}_1 = \beta_1 \frac{\sigma_x^2 + \sigma_{x\delta}}{\sigma_x^2 + \sigma_\delta^2 + 2\sigma_{x\delta}}$$

There are two main special cases of interest:

1. If there is no relation between X^A and δ , $\sigma_{x\delta} = 0$, this simplifies to:

$$E\hat{\beta}_1 = \beta_1 \frac{1}{1 + \sigma_\delta^2 / \sigma_x^2}$$

So $\hat{\beta}_1$ will be biased toward zero, regardless of the sample size. If σ_δ^2 is small relative to σ_x^2 , then the problem can be ignored. In other words, if the variability in the errors of observation of X is small relative to the range of X , then we need not be too concerned. For multiple predictors, the usual effect of measurement errors is also to bias the $\hat{\beta}$ in the direction of zero.

2. In controlled experiments, we need to distinguish two ways in which error in x may arise. In the first case, we measure x so although the true value is x^A , we observe x^O . If we were to repeat the measurement, we would have the same x^A , but a different x^O . In the second case, you fix x^O — for example, you make up a chemical solution with a specified concentration x^O . The true concentration would be x^A . Now if you were to repeat this, you would get the same x^O , but the x^A would be different. In this latter case we have:

$$\sigma_{x\delta} = cov(X^O - \delta, \delta) = -\sigma_\delta^2$$

and then we would have $E\hat{\beta}_1 = \beta_1$. So our estimate would be unbiased. This

seems paradoxical, until you notice that the second case effectively reverses the roles of x^A and x^O and if you get to observe the true X , then you will get an unbiased estimate of β_1 . See Berkson (1950) for a discussion of this.

If the model is used for prediction purposes, we can make the same argument as in the second case above. In repeated “experiments,” the value of x at which the prediction is to be made will be fixed, even though these may represent different underlying “true” values of x .

In cases where the error in X can simply not be ignored, we should consider alternatives to the least squares estimation of β . The simple least squares regression equation can be written as:

$$\frac{y - \bar{y}}{SD_y} = r \frac{(x - \bar{x})}{SD_x}$$

so that $\hat{\beta}_1 = rSD_y/SD_x$. Note that if we reverse the roles of x and y , we do not get the same regression equation. Since we have errors in both x and y in our problem, we might argue that neither one, in particular, deserves the role of response or predictor and so the equation should be the same either way. One way to achieve this is to set $\hat{\beta}_1 = SD_y/SD_x$. This is known as the *geometric mean functional relationship*. More on this can be found in Draper and Smith (1998). Another approach is to use the SIMEX method of Cook and Stefanski (1994), which we illustrate below.

Consider some data on the speed and stopping distances of cars in the 1920s. We plot the data, as seen in Figure 7.2, and fit a linear model:

```
> data(cars)
> plot(dist ~ speed, cars, ylab="distance")
> lmod <- lm(dist ~ speed, cars)
> sumary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.579      6.758     -2.60  0.012
speed        3.932       0.416      9.46  1.5e-12

n = 50, p = 2, Residual SE = 15.380, R-Squared = 0.65
> abline(lmod)
```

We could explore transformations and diagnostics for these data, but we will just focus on the measurement error issue. Now we investigate the effect of adding measurement error to the predictor. We plot the modified fits in Figure 7.2:

```
> lmod1 <- lm(dist ~ I(speed+rnorm(50)), cars)
> coef(lmod1)
      (Intercept) I(speed + rnorm(50))
      -15.0619      3.7582
> abline(lmod1, lty=2)
> lmod2 <- lm(dist ~ I(speed+2*rnorm(50)), cars)
> coef(lmod2)
      (Intercept) I(speed + 2 * rnorm(50))
      -5.3503      3.1676
> abline(lmod2, lty=3)
> lmod5 <- lm(dist ~ I(speed+5*rnorm(50)), cars)
> coef(lmod5)
      (Intercept) I(speed + 5 * rnorm(50))
      15.1589      1.8696
```

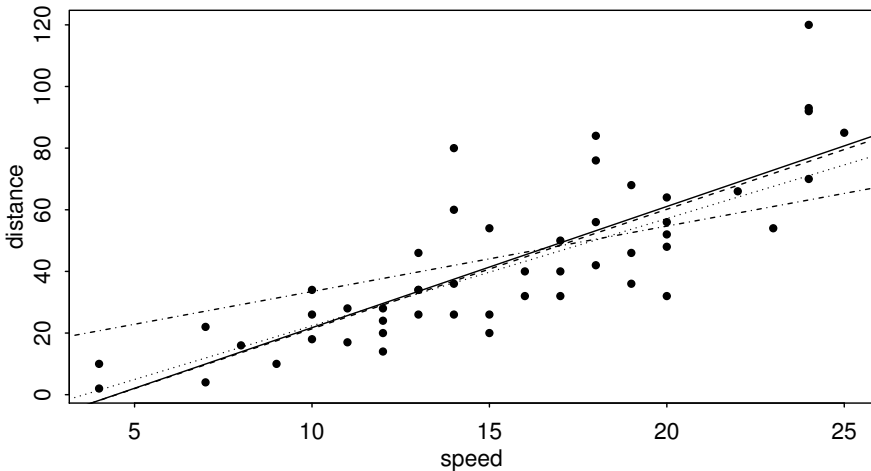


Figure 7.2 Stopping distances and speeds of cars. The least squares fit is shown as a solid line. The fits with three progressively larger amounts of measurement error on the speed are shown as dotted lines, where the slope gets shallower as the error increases.

```
> abline(lmod5, lty=4)
```

We can see that the slope becomes shallower as the amount of noise increases.

Suppose we knew that the predictor, `speed`, in the original data had been measured with a known error variance, say 0.5. Given what we have seen in the simulated measurement error models, we might extrapolate back to suggest an estimate of the slope under no measurement error. This is the idea behind SIMEX.

Here we simulate the effects of adding normal random error with variances ranging from 0.1 to 0.5, replicating the experiment 1000 times for each setting:

```
> vv <- rep(1:5/10, each=1000)
> slopes <- numeric(5000)
> for(i in 1:5000) slopes[i] <- lm(dist ~ I(speed+sqrt(vv[i])*rnorm
  (50)), cars)$coef[2]
```

Now we plot the mean slopes for each variance. We are assuming that the data have variance 0.5 so the extra variance is added to this:

```
> betas <- c(coef(lmod) [2], colMeans(matrix(slopes, nrow=1000)))
> variances <- c(0, 1:5/10)+0.5
> plot(variances, betas, xlim=c(0, 1), ylim=c(3.86, 4))
```

We fit a linear model and extrapolate to zero variance:

```
> gv <- lm(betas ~ variances)
> coef(gv)
(Intercept)  variances
  3.99975     -0.13552
> points(0, gv$coef[1], pch=3)
```

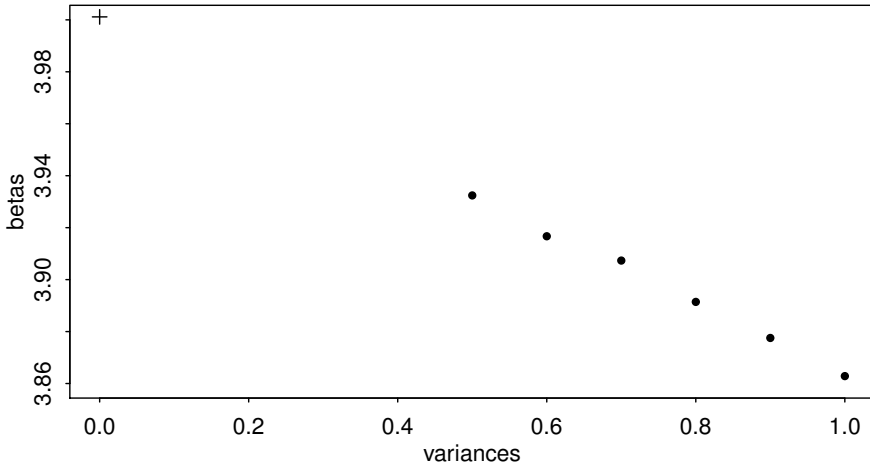


Figure 7.3 Simulation–extrapolation estimation of the unbiased slope in the presence of measurement error in the predictors. We predict $\hat{\beta} = 4.0$ at a variance of zero.

The predicted value of $\hat{\beta}$ at variance equal to zero, that is no measurement error, is 4.0. Better models for extrapolation are worth considering; see Cook and Stefanski (1994) for details. The method is implemented in the `simex` package:

```
> require(simex)
> set.seed(123)
> lmod <- lm(dist ~ speed, cars, x=TRUE)
> simout <- simex(lmod, "speed", 0.5, B=1000)
> simout
SIMEX-Variables: speed
Number of Simulations: 1000

Coefficients:
(Intercept)      speed
    -18.01         3.96
```

The results are about the same, bearing in mind that the method relies on simulation and hence random generation so the output will not be identical.

7.2 Changes of Scale

Sometimes we want to change the scale of the variables. Perhaps we need to change the units of measurement, say from inches to centimeters. A change of scale is often helpful when variables take values which are all very large or all very small. For example, we might find $\hat{\beta} = 0.000000351$. It can be misleading to deal with quantities like this because it's easy to lose track of the number of leading zeroes. Humans deal better with interpreting moderate sized numbers like $\hat{\beta} = 3.51$. So to avoid confusing yourself or your readers, a change of scale can be beneficial.

In more extreme cases, a change of scale can be needed to ensure numerical stability. Although many algorithms try to be robust against variables of widely different scales, it is possible to run into calculation errors. Most methods work more reliably when variables are on roughly similar scales.

Suppose we reexpress x_i as $(x_i + a)/b$. Rescaling x_i leaves the t - and F -tests and $\hat{\sigma}^2$ and R^2 unchanged and $\hat{\beta}_i \rightarrow b\hat{\beta}_i$. Rescaling y in the same way leaves the t - and F -tests and R^2 unchanged, but $\hat{\sigma}$ and $\hat{\beta}$ will be rescaled by b .

To demonstrate this, we use the savings data:

```
> data(savings, package="faraway")
> lmod <- lm(sr ~ pop15+pop75+dpi+ddpi, savings)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	28.566087	7.354516	3.88	0.00033
pop15	-0.461193	0.144642	-3.19	0.00260
pop75	-1.691498	1.083599	-1.56	0.12553
dpi	-0.000337	0.000931	-0.36	0.71917
ddpi	0.409695	0.196197	2.09	0.04247

n = 50, p = 5, Residual SE = 3.803, R-Squared = 0.34

The coefficient for income is rather small — let's measure income in thousands of dollars instead and refit:

```
> lmod <- lm(sr ~ pop15+pop75+I(dpi/1000)+ddpi, savings)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	28.566	7.355	3.88	0.00033
pop15	-0.461	0.145	-3.19	0.00260
pop75	-1.691	1.084	-1.56	0.12553
I(dpi/1000)	-0.337	0.931	-0.36	0.71917
ddpi	0.410	0.196	2.09	0.04247

n = 50, p = 5, Residual SE = 3.803, R-Squared = 0.34

What changed and what stayed the same?

One rather thorough approach to scaling is to convert all the variables to standard units (mean 0 and variance 1) using the `scale()` command:

```
> scsav <- data.frame(scale(savings))
> lmod <- lm(sr ~ ., scsav)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.01e-16	1.20e-01	0.00	1.0000
pop15	-9.42e-01	2.95e-01	-3.19	0.0026
pop75	-4.87e-01	3.12e-01	-1.56	0.1255
dpi	-7.45e-02	2.06e-01	-0.36	0.7192
ddpi	2.62e-01	1.26e-01	2.09	0.0425

n = 50, p = 5, Residual SE = 0.849, R-Squared = 0.34

As may be seen, the intercept is zero. This is because the regression plane always runs through the point of the averages, which because of the centering, is now at the origin. Such scaling has the advantage of putting all the predictors and the response

on a comparable scale, which makes comparisons simpler. It also allows the coefficients to be viewed as a kind of partial correlation — the values will always be between minus one and one. It also avoids some numerical problems that can arise when variables are of very different scales. The interpretation effect of this scaling is that the regression coefficients now represent the effect of a one standard unit increase in the predictor on the response in standard units — this might or might not be easy to interpret.

When the predictors are on comparable scales, it can be helpful to construct a plot of the estimates with confidence intervals as seen in Figure 7.4.

```
> edf <- data.frame(coef(lmod), confint(lmod))[-1, ]
> names(edf) <- c('Estimate', 'lb', 'ub')
> require(ggplot2)
> p <- ggplot(aes(y=Estimate, ymin=lb, ymax=ub, x=row.names(edf)), data=
+ edf) + geom_pointrange()
> p+coord_flip()+xlab("Predictor")+geom_hline(xint=0, col=gray(0.75))
```

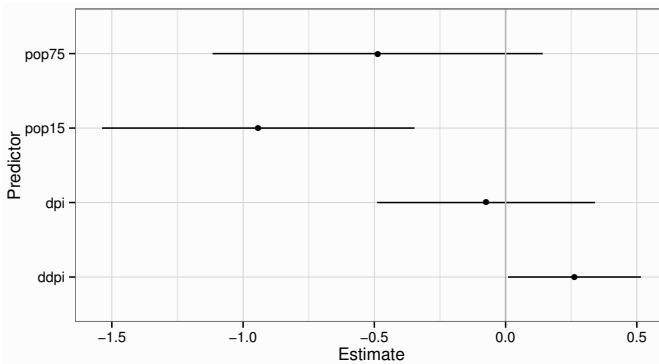


Figure 7.4 Plot of the standardized coefficients for the savings model; 95% confidence intervals are shown.

In the presence of binary predictors, scaling might be done differently. For example, we notice that the countries in the savings data divide into two clusters based on age. We can set a division at 35% for `pop15`:

```
> savings$age <- ifelse(savings$pop15 > 35, 0, 1)
```

so that younger countries are coded as zero and older countries as one. A binary predictor taking the values of 0/1 with equal probability has a standard deviation of one half. This suggests scaling the other continuous predictors by two SDs rather than one:

```
> savings$dpis <- (savings$dpi - mean(savings$dpi)) / (2 * sd(savings$dpi))
> savings$ddpis <- (savings$ddpi - mean(savings$ddpi)) / (2 * sd(savings$
+ ddpi))
> summary(lm(sr ~ age + dpis + ddpis, savings))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.82	1.01	6.75	2.2e-08
age	5.28	1.58	3.33	0.0017

dpis	-1.56	1.61	-0.97	0.3361
ddpis	2.47	1.11	2.23	0.0309

n = 50, p = 4, Residual SE = 3.800, R-Squared = 0.32

Now the interpretation of the coefficients becomes easier. The predicted difference between older and younger countries is a savings rate 5.28% higher in the former. This is a difference of two standard deviations. The same change of two standard deviations in `ddpis` means a difference of one in the new scale of `dpis`. So we could say that a typical country with a high growth rate has a savings rate 2.47% higher than one with a low growth rate. Now `ddpi` is really a continuous variable so this interpretation is just for convenient intuition. Another way to achieve a similar effect is to use a $-1/+1$ coding rather than 0/1 so that the standard scaling can be used on the continuous predictors. See Gelman (2008) for more details.

7.3 Collinearity

When some predictors are linear combinations of others, then $X^T X$ is singular, and we have a lack of identifiability as discussed in Section 2.10. Another name for this problem is exact collinearity. There is no unique least squares estimate of β . The solution may require removing some predictors.

A more challenging problem arises when $X^T X$ is close to singular but not exactly so. This is known as collinearity or sometimes, multicollinearity. Collinearity leads to imprecise estimates of β . The signs of the coefficients can be the opposite of what intuition about the effect of the predictor might suggest. The standard errors are inflated so that t -tests may fail to reveal significant factors. The fit becomes very sensitive to measurement errors where small changes in y can lead to large changes in $\hat{\beta}$.

Collinearity can be detected in several ways:

1. Examination of the correlation matrix of the predictors may reveal values close to -1 or $+1$ indicating large *pairwise* collinearities.
2. A regression of x_i on all other predictors gives R_i^2 . R_i^2 close to one indicates a problem because it means one predictor can almost be predicted exactly by a linear combination of other predictors. Repeat for all predictors. The offending linear combination may be discovered by examining the regression coefficients of these regressions.
3. Examine the eigenvalues of $X^T X$, $\lambda_1 \geq \dots \geq \lambda_p \geq 0$. Zero eigenvalues denote exact collinearity while the presence of some small eigenvalues indicates multicollinearity. The condition number κ measures the relative sizes of the eigenvalues and is defined as:

$$\kappa = \sqrt{\frac{\lambda_1}{\lambda_p}}$$

where $\kappa \geq 30$ is considered large. Other condition numbers, $\sqrt{\lambda_1/\lambda_i}$ are also worth considering because they indicate whether more than just one independent

linear combination is to blame. Alternative calculations involve standardizing the predictors and/or including the intercept term in X .

The effect of collinearity can be seen by this expression for $\text{var } \hat{\beta}_j$:

$$\text{var } \hat{\beta}_j = \sigma^2 \left(\frac{1}{1 - R_j^2} \right) \frac{1}{\sum_i (x_{ij} - \bar{x}_j)^2}$$

We can see that if the predictor x_j does not vary much, then the variance of $\hat{\beta}_j$ will be large. If R_j^2 is close to one, then the *variance inflation factor* $(1 - R_j^2)^{-1}$ will be large and so $\text{var } \hat{\beta}_j$ will also be large.

This equation also tells us which designs will minimize the variance of the regression coefficients if we have the ability to choose the X . Orthogonality means that $R_j^2 = 0$ which minimizes the variance. Also we can maximize $S_{x_j x_j}$ by spreading X as much as possible. The maximum is attained by placing half the points at the minimum practical value and half at the maximum. Unfortunately, this design assumes the linearity of the effect and would make it impossible to check for any curvature. So, in practice, we might put some design points in the middle of the range to allow checking of the fit.

Car drivers like to adjust the seat position for their own comfort. Car designers would find it helpful to know where different drivers will position the seat depending on their size and age. Researchers at the HuMoSim laboratory at the University of Michigan collected data on 38 drivers. They measured age in years, weight in pounds, height with shoes and without shoes in centimeters, seated height arm length, thigh length, lower leg length and `hipcenter`, the horizontal distance of the midpoint of the hips from a fixed location in the car in millimeters. We fit a model with all the predictors:

```
> data(seatpos, package="faraway")
> lmod <- lm(hipcenter ~ ., seatpos)
> sumary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	436.4321	166.5716	2.62	0.014
Age	0.7757	0.5703	1.36	0.184
Weight	0.0263	0.3310	0.08	0.937
HtShoes	-2.6924	9.7530	-0.28	0.784
Ht	0.6013	10.1299	0.06	0.953
Seated	0.5338	3.7619	0.14	0.888
Arm	-1.3281	3.9002	-0.34	0.736
Thigh	-1.1431	2.6600	-0.43	0.671
Leg	-6.4390	4.7139	-1.37	0.182

```
n = 38, p = 9, Residual SE = 37.720, R-Squared = 0.69
```

This model already shows the signs of collinearity. The R^2 is not small, but none of the individual predictors is significant. We take a look at the pairwise correlations:

```
> round(cor(seatpos[, -9]), 2)
```

	Age	Weight	HtShoes	Ht	Seated	Arm	Thigh	Leg
Age	1.00	0.08	-0.08	-0.09	-0.17	0.36	0.09	-0.04
Weight	0.08	1.00	0.83	0.83	0.78	0.70	0.57	0.78

HtShoes	-0.08	0.83	1.00	1.00	0.93	0.75	0.72	0.91
Ht	-0.09	0.83	1.00	1.00	0.93	0.75	0.73	0.91
Seated	-0.17	0.78	0.93	0.93	1.00	0.63	0.61	0.81
Arm	0.36	0.70	0.75	0.75	0.63	1.00	0.67	0.75
Thigh	0.09	0.57	0.72	0.73	0.61	0.67	1.00	0.65
Leg	-0.04	0.78	0.91	0.91	0.81	0.75	0.65	1.00

There are several large pairwise correlations between predictors. Now we check the eigendecomposition of $X^T X$ (not including the intercept in X):

```
> x <- model.matrix(lmod)[,-1]
> e <- eigen(t(x) %*% x)
> e$val
[1] 3653671.3630 21479.4802 9043.2253 298.9526 148.3948
[6] 81.1740 53.3619 7.2982
> sqrt(e$val[1]/e$val)
[1] 1.000 13.042 20.100 110.551 156.912 212.156 261.667 707.549
```

There is a wide range in the eigenvalues and several condition numbers are large. This means that problems are being caused by more than just one linear combination. Now check the variance inflation factors (VIFs). For the first variable this is:

```
> summary(lm(x[,1] ~ x[,-1]))$r.squared
[1] 0.49948
> 1/(1-0.49948)
[1] 1.9979
```

which is moderate in size — the VIF for orthogonal predictors is one. Now we compute all the VIFs in one go, using a function from the `faraway` package:

```
> require(faraway)
> vif(x)
      Age      Weight  HtShoes      Ht      Seated      Arm      Thigh
1.9979  3.6470  307.4294 333.1378  8.9511  4.4964  2.7629
      Leg
6.6943
```

There is much variance inflation. For example, we can interpret $\sqrt{307.4} = 17.5$ as telling us that the standard error for height with shoes is 17.5 times larger than it would have been without collinearity. We cannot apply this as a correction because we did not actually observe orthogonal data, but it does give us a sense of the size of the effect.

There is substantial instability in these estimates. Measuring the hipcenter is difficult to do accurately and we can expect some variation in these values. Suppose the measurement error had a SD of 10 mm. Let's see what happens when we add a random perturbation of this size to the response:

```
> lmod <- lm(hipcenter+10*rnorm(38) ~ ., seatpos)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 456.6518    178.1780   2.56   0.016
Age          0.8215     0.6101    1.35   0.189
Weight       0.0195     0.3540    0.06   0.956
HtShoes     -3.4522    10.4326   -0.33   0.743
Ht           1.0121    10.8357    0.09   0.926
Seated       0.5625     4.0240    0.14   0.890
```

Arm	-1.0400	4.1720	-0.25	0.805
Thigh	-1.8641	2.8454	-0.66	0.518
Leg	-4.8880	5.0423	-0.97	0.340

n = 38, p = 9, Residual SE = 40.349, R-Squared = 0.66

Although the R^2 and standard error are very similar to the previous fit, we see much larger changes in the coefficients, indicating their sensitivity to the response values caused by the collinearity.

We have too many variables that are trying to do the same job of explaining the response. We can reduce the collinearity by carefully removing some of the variables. We can then make a more stable estimation of the coefficients and come to a more secure conclusion regarding the effect of the remaining predictors on the response. But we should not conclude that the variables we drop have nothing to do with the response.

Examine the full correlation matrix above. Consider just the correlations of the length variables:

```
> round(cor(x[, 3:8]), 2)
```

	HtShoes	Ht	Seated	Arm	Thigh	Leg
HtShoes	1.00	1.00	0.93	0.75	0.72	0.91
Ht	1.00	1.00	0.93	0.75	0.73	0.91
Seated	0.93	0.93	1.00	0.63	0.61	0.81
Arm	0.75	0.75	0.63	1.00	0.67	0.75
Thigh	0.72	0.73	0.61	0.67	1.00	0.65
Leg	0.91	0.91	0.81	0.75	0.65	1.00

These six variables are strongly correlated with each other — any one of them might do a good job of representing the other. We pick height as the simplest to measure. We are not claiming that the other predictors are not associated with the response, just that we do not need them all to predict the response:

```
> lmod2 <- lm(hipcenter ~ Age + Weight + Ht, seatpos)
> summary(lmod2)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	528.29773	135.31295	3.90	0.00043
Age	0.51950	0.40804	1.27	0.21159
Weight	0.00427	0.31172	0.01	0.98915
Ht	-4.21190	0.99906	-4.22	0.00017

n = 38, p = 4, Residual SE = 36.486, R-Squared = 0.66

Comparing this with the original fit, we see that the fit is very similar in terms of R^2 , but many fewer predictors are used. Further simplification is clearly possible.

If you want to keep all your variables in the model, you should consider alternative methods of estimation such as ridge regression as described in Section 11.3.

The effect of collinearity on prediction is less serious. The accuracy of the prediction depends on where the prediction is to be made. The greater the distance is from the observed data, the more unstable the prediction. Of course, this is true for all data but collinear data covers a much smaller fraction of the predictor space than it might first appear. This means that predictions tend to be greater extrapolations than with data that are closer to the orthogonality.

Exercises

1. Using the `faithful` data, fit a regression of `duration` on `waiting`. Assuming that there was a measurement error in `waiting` of 30 seconds, use the SIMEX method to obtain a better estimate of the slope.
2. What would happen if the SIMEX method was applied to the response error variance rather than predictor measurement error variance?
3. Using the `divusa` data:
 - (a) Fit a regression model with `divorce` as the response and `unemployed`, `femlab`, `marriage`, `birth` and `military` as predictors. Compute the condition numbers and interpret their meanings.
 - (b) For the same model, compute the VIFs. Is there evidence that collinearity causes some predictors not to be significant? Explain.
 - (c) Does the removal of insignificant predictors from the model reduce the collinearity? Investigate.
4. For the `longley` data, fit a model with `Employed` as the response and the other variables as predictors.
 - (a) Compute and comment on the condition numbers.
 - (b) Compute and comment on the correlations between the predictors.
 - (c) Compute the variance inflation factors.
5. For the `prostate` data, fit a model with `lpsa` as the response and the other variables as predictors.
 - (a) Compute and comment on the condition numbers.
 - (b) Compute and comment on the correlations between the predictors.
 - (c) Compute the variance inflation factors.
6. Using the `cheddar` data, fit a linear model with `taste` as the response and the other three variables as predictors.
 - (a) Is the predictor `Lactic` statistically significant in this model?
 - (b) Give the R command to extract the p -value for the test of $\beta_{lactic} = 0$. Hint: look at `summary()$coef`.
 - (c) Add normally distributed errors to `Lactic` with mean zero and standard deviation 0.01 and refit the model. Now what is the p -value for the previous test?
 - (d) Repeat this same calculation of adding errors to `Lactic` 1000 times within `for` loop. Save the p -values into a vector. Report on the average p -value. Does this much measurement error make a qualitative difference to the conclusions?
 - (e) Repeat the previous question but with a standard deviation of 0.1. Does this much measurement error make an important difference?
7. Use the `happy` dataset with `happy` as the response and the other variables as predictors. Discuss possible rescalings of the variables with the aim of helping the interpretation of the model fit.

8. Use the `fat` data, fitting the model described in Section 4.2.
- (a) Compute the condition numbers and variance inflation factors. Comment on the degree of collinearity observed in the data.
 - (b) Cases 39 and 42 are unusual. Refit the model without these two cases and recompute the collinearity diagnostics. Comment on the differences observed from the full data fit.
 - (c) Fit a model with `brozek` as the response and just `age`, `weight` and `height` as predictors. Compute the collinearity diagnostics and compare to the full data fit.
 - (d) Compute a 95% prediction interval for `brozek` for the median values of `age`, `weight` and `height`.
 - (e) Compute a 95% prediction interval for `brozek` for `age=40`, `weight=200` and `height=73`. How does the interval compare to the previous prediction?
 - (f) Compute a 95% prediction interval for `brozek` for `age=40`, `weight=130` and `height=73`. Are the values of predictors unusual? Comment on how the interval compares to the previous two answers.

Problems with the Error

We have assumed that the error ε is independent and identically distributed (i.i.d.). Furthermore, we have also assumed that the errors are normally distributed in order to carry out the usual statistical inference. We have seen that these assumptions can often be violated and we must then consider alternatives. When the errors are dependent, we can use *generalized least squares* (GLS). When the errors are independent, but not identically distributed, we can use *weighted least squares* (WLS), which is a special case of GLS. Sometimes, we have a good idea how large the error should be, but the residuals may be much larger than we expect. This is evidence of a *lack of fit*. When the errors are not normally distributed, we can use *robust regression*.

8.1 Generalized Least Squares

Until now we have assumed that $\text{var } \varepsilon = \sigma^2 I$, but sometimes the errors have non-constant variance or are correlated. Suppose instead that $\text{var } \varepsilon = \sigma^2 \Sigma$ where σ^2 is unknown but Σ is known — in other words, we know the correlation and relative variance between the errors, but we do not know the absolute scale of the variation. For now, it might seem redundant to distinguish between σ and Σ , but we will see how this will be useful later.

We can write $\Sigma = SS^T$, where S is a triangular matrix using the Choleski decomposition, which can be viewed as a square root for a matrix. We can transform the regression model as follows:

$$\begin{aligned} y &= X\beta + \varepsilon \\ S^{-1}y &= S^{-1}X\beta + S^{-1}\varepsilon \\ y' &= X'\beta + \varepsilon' \end{aligned}$$

Now we find that:

$$\text{var } \varepsilon' = \text{var } (S^{-1}\varepsilon) = S^{-1}(\text{var } \varepsilon)S^{-T} = S^{-1}\sigma^2 SS^T S^{-T} = \sigma^2 I$$

So we can reduce GLS to ordinary least squares (OLS) by a regression of $y' = S^{-1}y$ on $X' = S^{-1}X$ which has error $\varepsilon' = S^{-1}\varepsilon$ that is i.i.d. We have transformed the problem to the standard case. In this transformed model, the sum of squares is:

$$(S^{-1}y - S^{-1}X\beta)^T (S^{-1}y - S^{-1}X\beta) = (y - X\beta)^T S^{-T} S^{-1} (y - X\beta) = (y - X\beta)^T \Sigma^{-1} (y - X\beta)$$

which is minimized by:

$$\hat{\beta} = (X^T \Sigma^{-1} X)^{-1} X^T \Sigma^{-1} y$$

We find that:

$$\text{var } \hat{\beta} = (X^T \Sigma^{-1} X)^{-1} \sigma^2$$

Since $\varepsilon' = S^{-1}\varepsilon$, diagnostics should be applied to the residuals, $S^{-1}\hat{\varepsilon}$. If we have the right Σ , then these should be approximately i.i.d.

The main problem in applying GLS in practice is that Σ may not be known and we have to estimate it. Let's take another look at the global warming data first considered in Section 6.1.3. We found evidence of serial correlation by looking at successive residuals. We demonstrate how we can model this using GLS. We start with the OLS solution by fitting a linear model:

```
> data(globwarm, package="faraway")
> lmod <- lm(nhtemp ~ wusa + jasper + westgreen + chesapeake +
  tornetrask + urals + mongolia + tasman, globwarm)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.24256	0.02701	-8.98	2e-15
wusa	0.07738	0.04293	1.80	0.07365
jasper	-0.22879	0.07811	-2.93	0.00399
westgreen	0.00958	0.04184	0.23	0.81917
chesapeake	-0.03211	0.03405	-0.94	0.34735
tornetrask	0.09267	0.04505	2.06	0.04161
urals	0.18537	0.09143	2.03	0.04457
mongolia	0.04197	0.04579	0.92	0.36100
tasman	0.11545	0.03011	3.83	0.00019

```
n = 145, p = 9, Residual SE = 0.176, R-Squared = 0.48
> cor(residuals(lmod)[-1], residuals(lmod)[-length(residuals(lmod))])
[1] 0.58334
```

In data collected over time such as this, successive errors could be correlated. We have calculated this by computing the correlation between the vector of residuals with the first and then the last term omitted. We can see a correlation of 0.58 between successive residuals. The simplest way to model this is the autoregressive form:

$$\varepsilon_{i+1} = \phi\varepsilon_i + \delta_i$$

where $\delta_i \sim N(0, \tau^2)$. The nlme package of Pinheiro and Bates (2000) contains a GLS fitting function. We can use it to fit this model:

```
> require(nlme)
> glmmod <- gls(nhtemp ~ wusa + jasper + westgreen + chesapeake +
  tornetrask + urals + mongolia + tasman, correlation=corAR1(form=
  ~year), data=na.omit(globwarm))
> summary(glmmod)
```

Generalized least squares fit by REML
Correlation Structure: AR(1)
Formula: ~year
Parameter estimate(s):
Phi
0.71099

Coefficients:

	Value	Std.Error	t-value	p-value
(Intercept)	-0.230106	0.067024	-3.4332	0.0008


```

wusa      0.066738  0.098772  0.6757  0.5004
jasper    -0.202443  0.188028 -1.0767  0.2835
westgreen -0.004403  0.089853 -0.0490  0.9610
chesapeake -0.007353  0.073498 -0.1000  0.9205
tornetrask 0.038352  0.094825  0.4044  0.6865
urals     0.241422  0.228710  1.0556  0.2930
mongolia  0.056950  0.104898  0.5429  0.5881
tasman    0.120349  0.074570  1.6139  0.1089

Residual standard error: 0.20457
Degrees of freedom: 145 total; 136 residual

```

The `lm` function drops missing value cases (which is what we want) but `gls` complains so we have removed these cases using the `na.omit` function. We notice that the residual standard error $\hat{\sigma}$ is a little larger in the GLS case as might be expected. The standard errors of $\hat{\beta}$ are much larger and none of the predictors is statistically significant in the GLS output. However, there is substantial collinearity between the predictors so this should not be interpreted as “no predictor effect”. Also you should understand that correlation between the predictors and correlation between the errors are different phenomena and there is no necessary link between the two. We see that the estimated value of ϕ is 0.71. It is not surprising to see significant autocorrelation in this example because the proxies can only partially predict the temperature and we would naturally expect some carryover effect from one year to the next. We check the confidence intervals for this term:

```

> intervals(glmod, which="var-cov")
Approximate 95% confidence intervals

Correlation structure:
      lower  est.  upper
Phi 0.50997 0.71099 0.83838
attr(,"label")
[1] "Correlation structure:"

Residual standard error:
      lower  est.  upper
0.15407 0.20457 0.27163

```

We see from the interval, (0.51, 0.84), that this term is clearly different from zero and that there is significant positive correlation. For this example, we might investigate whether a more sophisticated model should apply to errors perhaps using a so-called ARMA model. This can be implemented using the `corARMA` function.

Another situation where correlation between errors might be anticipated is where observations are grouped in some way. For example, consider an experiment to compare eight varieties of oats. The growing area was heterogeneous and so was grouped into five blocks. Each variety was sown once within each block and the yield in grams per 16-ft row was recorded. The data come from Anderson and Bancroft (1952). It is reasonable to expect that the observations within a block might have errors which are correlated, that is $cor(\epsilon_i, \epsilon_j)$ is ρ if i and j are in the same block while zero otherwise. This is called the compound symmetry assumption and is modeled as follows:

```

> glmod <- gls(yield ~ variety, oatvar, correlation = corCompSymm(form
  = ~1 | block))
> intervals(glmod)
Approximate 95% confidence intervals

Correlation structure:
      lower  est.  upper
Rho  0.065964  0.396  0.74937

```

We see that there is good evidence of a non-zero correlation of around 0.4 between the errors within the blocks. There are other ways to model the block effect and we return to this dataset in Section 17.1.

Other examples where correlated errors can arise are in spatial data where the relative locations of the observations can be used to model the error correlation. In other cases, one may suspect a correlation between errors but have no structure to suggest a parameterized form such as serial correlation or compound symmetry. The problem is that there are too many pairwise correlations to be estimated and not enough data to do it.

8.2 Weighted Least Squares

Sometimes the errors are uncorrelated, but have unequal variance where the form of the inequality is known. In such cases, Σ is diagonal but the entries are not equal. Weighted least squares (WLS) is a special case of GLS and can be used in this situation. We set $\Sigma = \text{diag}(1/w_1, \dots, 1/w_n)$, where the w_i are the *weights* so $S = \text{diag}(\sqrt{1/w_1}, \dots, \sqrt{1/w_n})$. We then regress $\sqrt{w_i}y_i$ on $\sqrt{w_i}x_i$ (although the column of ones in the X -matrix needs to be replaced with $\sqrt{w_i}$). When weights are used, the residuals must be modified to use $\sqrt{w_i}\hat{\epsilon}_i$. We see that cases with low variability get a high weight and those with high variability a low weight. Some examples:

1. Errors proportional to a predictor: $\text{var}(\epsilon_i) \propto x_i$ suggests $w_i = x_i^{-1}$. One might choose this option after observing a positive relationship in a plot of $|\hat{\epsilon}_i|$ against x_i .
2. When the Y_i are the averages of n_i observations, then $\text{var} y_i = \text{var} \epsilon_i = \sigma^2/n_i$, which suggests $w_i = n_i$. Responses that are averages arise quite commonly, but take care that the variance in the response really is proportional to the group size. For example, consider the life expectancies for different countries. At first glance, one might consider setting the weights equal to the populations of the countries, but notice that there are many other sources of variation in life expectancy that would dwarf the population size effect.
3. When the observed responses are known to be of varying quality, weights may be assigned $w_i = 1/\text{sd}(y_i)$.

Elections for the French presidency proceed in two rounds. In 1981, there were 10 candidates in the first round. The top two candidates then went on to the second round, which was won by François Mitterand over Valéry Giscard-d'Estaing. The losers in the first round can gain political favors by urging their supporters to vote for one of the two finalists. Since voting is private, we cannot know how these votes

were transferred; we might hope to infer from the published vote totals how this might have happened. Anderson and Loynes (1987) published data on these vote totals in every fourth department of France:

```
> data(fpe, package="faraway")
> fpe
      EI   A   B   C   D   E   F   G   H   J   K   A2   B2   N
Ain   260  51  64  36  23   9   5   4   4   3   3  105  114  17
Alpes  75  14  17   9   9   3   1   2   1   1   1   32   31   5
...
```

A and B stand for Mitterand's and Giscard's votes in the first round, respectively, while A2 and B2 represent their votes in the second round. C–K are the first round votes of the other candidates while EI denotes *electeur inscrits* or registered voters. All numbers are in thousands. The total number of voters in the second round was greater than the first — we can compute the difference as N. We will treat this group like another first round candidate (although we could reasonably handle this differently).

Now we can represent the transfer of votes as:

$$A2 = \beta_A A + \beta_B B + \beta_C C + \beta_D D + \beta_E E + \beta_F F + \beta_G G + \beta_H H + \beta_J J + \beta_K K + \beta_N N$$

where β_i represents the proportion of votes transferred from candidate i to Mitterand in the second round. We can equally well do this for Giscard-d'Estaing but then the β 's will simply be the remaining proportions so it's not necessary to do both. We would expect these transfer proportions to vary somewhat between departments, so if we treat the above as a regression equation, there will be some error from department to department. The error will have a variance in proportion to the number of voters because it will be like a variance of a sum rather than a mean. Since the weights should be inversely proportional to the variance, this suggests that the weights should be set to $1/EI$. Notice also that the equation has no intercept, hence the -1 in the model formula. We fit the appropriate model:

```
> lmod <- lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1, fpe, weights=1/EI)
> coef(lmod)
      A           B           C           D           E           F           G
1.06713 -0.10505  0.24596  0.92619  0.24940  0.75511  1.97221
      H           J           K           N
-0.56622  0.61164  1.21066  0.52935
```

Note that the weights do matter — see what happens when we leave them out:

```
> lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1, fpe)$coef
      A           B           C           D           E           F           G
1.07515 -0.12456  0.25745  0.90454  0.67068  0.78253  2.16566
      H           J           K           N
-0.85429  0.14442  0.51813  0.55827
```

which causes substantial changes for some of the lesser candidates. Furthermore, only the relative proportions of the weights matter — for example, suppose we multiply the weights by 53:

```
> lm(A2 ~ A+B+C+D+E+F+G+H+J+K+N-1, fpe, weights=53/EI)$coef
      A      B      C      D      E      F      G
1.06713 -0.10505 0.24596 0.92619 0.24940 0.75511 1.97221
      H      J      K      N
-0.56622 0.61164 1.21066 0.52935
```

This makes no difference.

Now there is one remaining difficulty, unrelated to the weighting, in that proportions are supposed to be between zero and one. We can impose an *ad hoc* fix by truncating the coefficients that violate this restriction either to zero or one as appropriate. This gives:

```
> lm(A2 ~ offset(A+G+K)+C+D+E+F+N-1, fpe, weights=1/EI)$coef
      C      D      E      F      N
0.22577 0.96998 0.39020 0.74424 0.60854
```

The `offset` function means no coefficient will be fit which is the same as saying the coefficient will be one. We see that voters for the Communist candidate D apparently almost all voted for the Socialist Mitterand in the second round. However, we see that around 20% of the voters for the Gaullist candidate C voted for Mitterand. This is surprising since these voters would normally favor the more right wing candidate, Giscard. This appears to be the decisive factor. We see that of the larger blocks of smaller candidates, the Ecology party voters, E, roughly split their votes as did the first round non-voters. The other candidates had very few voters and so their behavior is less interesting.

This analysis is somewhat crude and more sophisticated approaches are discussed in Anderson and Loynes (1987). The `pcls()` function in the `mgcv` package of Wood (2006) provides a solution to the constrained least squares problem which in this case requires $0 \leq \hat{\beta}_i \leq 1$. The solution obtained is:

```
> require(mgcv)
> M <- list(w=1/fpe$EI, X=model.matrix(lmod), y=fpe$A2, Ain=rbind(diag(11),-diag(11)), C=matrix(0,0,0), array(0,0), S=list(), off=NULL, p=rep(0.5,11), bin=c(rep(0,11), rep(-1,11)))
> a <- pcls(M)
> names(a) <- colnames(model.matrix(lmod))
> round(a, 3)
      A      B      C      D      E      F      G      H      J      K      N
1.000 0.000 0.208 0.969 0.359 0.743 1.000 0.367 0.000 1.000 0.575
```

The results are quite similar for the candidates C, D, E and N who have substantial numbers of votes, but the coefficients for small party candidates vary much more.

In examples where the form of the variance of ε is not completely known, we may model Σ using a small number of parameters. For example:

$$\text{sde}_i = \gamma_0 + x_1^{\gamma_1}$$

might seem reasonable in a given situation. Notice that we do not need a coefficient in front of x_1 as the specification of weights is invariant to multiplication by a constant.

Consider, for example the `cars` data used in Chapter 7:

```
> lmod <- lm(dist ~ speed, cars)
> plot(residuals(lmod) ~ speed, cars)
```

The plot (not shown) reveals that the variation in the residuals increases with speed. One solution to this problem is to set the weights according to the above form and simultaneously estimate β and γ using maximum likelihood methods. We can achieve this as follows:

```
> wlmod <- gls(dist ~ speed, data=cars, weight = varConstPower(1, form
= ~ speed))
> summary(wlmod)
Variance function:
Structure: Constant plus power of variance covariate
Formula: ~speed
Parameter estimates:
const power
3.1604 1.0224

Coefficients:
                Value Std. Error t-value p-value
(Intercept)  -11.0854   4.0524  -2.7355  0.0087
speed          3.4842   0.3202  10.8799  0.0000

Correlation:
(Intr)
speed -0.9

Standardized residuals:
      Min      Q1      Med      Q3      Max
-1.45206 -0.68982 -0.13083  0.63750  3.07570

Residual standard error: 0.76368
Degrees of freedom: 50 total; 48 residual
```

We see that $\hat{\gamma}_0 = 3.16$ and $\hat{\gamma}_1 = 1.0224$. Since the latter is so close to 1, we can see that this variance function takes quite a simple form.

8.3 Testing for Lack of Fit

How can we tell whether a model fits the data? If the model is correct, then $\hat{\sigma}^2$ should be an unbiased estimate of σ^2 . If we have a model that is not complex enough to fit the data or simply takes the wrong form, then $\hat{\sigma}^2$ will tend to overestimate σ^2 . The situation is illustrated in Figure 8.1 where the residuals from the incorrect constant fit will lead to an overestimate of σ^2 . Alternatively, if our model is too complex and overfits the data, then $\hat{\sigma}^2$ will be an underestimate.

This suggests a possible testing procedure — we should compare $\hat{\sigma}^2$ to σ^2 . But this requires us to know σ^2 . In a few cases, we might actually know σ^2 — for example, when measurement error is the only source of variation and we know its variance because we are very familiar with the measurement device. This is rather uncommon because we usually have more than just measurement error.

We might compare the $\hat{\sigma}^2$ from our chosen regression model to one obtained from another model. But this may just be the same as the F -test introduced in Chapter 3. This would indicate a preference between the two models but could not tell us that the preferred model fit the data. We need to make a comparison to some model-free estimate of σ^2 .

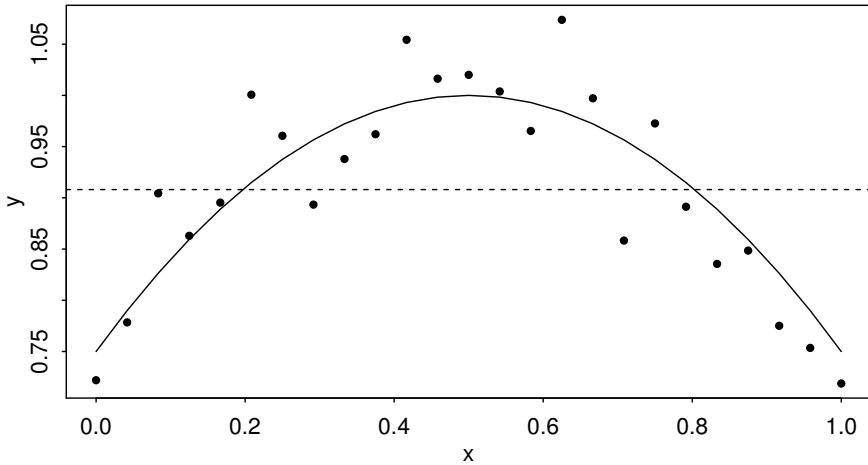


Figure 8.1 True quadratic fit shown with the solid line and incorrect linear fit shown with the dotted line. Estimate of σ^2 will be unbiased for the quadratic model, but too large for the linear model.

We can do this if we have repeated values of the response for one or more fixed values of x . These replicates do need to be truly independent. They cannot just be repeated measurements on the same subject or unit. For example, the cases in the data may be people and the response might be blood pressure. We might sensibly repeat these measurements of blood pressure. But such repeated measures would only reveal the within-subject variability or the measurement error. We need different people but with the same predictor values. This would give us the between-subject variability and allow us to construct an estimate of σ^2 that does not depend on a particular model.

Let y_{ij} be the i^{th} observation in the group of true replicates j . The “pure error” or model-free estimate of σ^2 is given by SS_{pe}/df_{pe} where:

$$SS_{pe} = \sum_j \sum_i (y_{ij} - \bar{y}_j)^2$$

where \bar{y}_j is the mean within replicate group j . The degrees of freedom are $df_{pe} = \sum_j (\# \text{replicates}_j - 1) = n - \# \text{groups}$.

There is a convenient way to compute the estimate. Fit a model that assigns one parameter to each group of observations with fixed x , then the $\hat{\sigma}^2$ from this model will be the pure error $\hat{\sigma}^2$. This model is saturated and tells us nothing interesting about the relationship between the predictors and the response. It simply fits a mean to each group of replicates. Values of x with no replication will be fit exactly and will not contribute to the estimate of σ^2 . Comparing this model to the regression model using the standard F -test gives us the lack-of-fit test.

The data for this example consist of 13 specimens of 90/10 Cu–Ni alloys with

varying percentages of iron content. The specimens were submerged in seawater for 60 days and the weight loss due to corrosion was recorded in units of milligrams per square decimeter per day. The data come from Draper and Smith (1998). We load in and plot the data, as seen in Figure 8.2:

```
> data(corrosion, package="faraway")
> plot(loss ~ Fe, corrosion, xlab="Iron content", ylab="Weight loss")
```

We fit a straight-line model:

```
> lmod <- lm(loss ~ Fe, corrosion)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   129.79      1.40     92.5 < 2e-16
Fe            -24.02      1.28    -18.8 1.1e-09

n = 13, p = 2, Residual SE = 3.058, R-Squared = 0.97
```

Now we show the regression line on the plot:

```
> abline(coef(lmod))
```

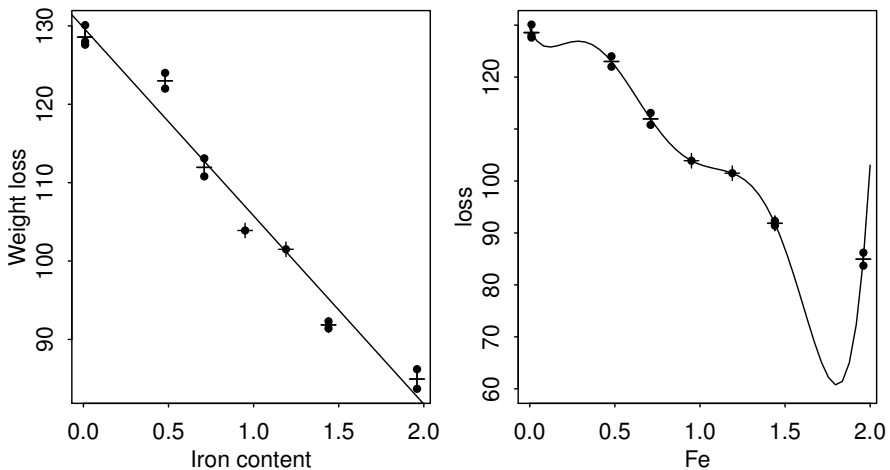


Figure 8.2 Linear fit to the Cu–Ni corrosion data is shown on the left. Group means are denoted by plus signs. A polynomial fit to the data is shown on the right.

We have an R^2 of 97% and an apparently good fit to the data. We now fit a model that reserves a parameter for each group of data with the same value of x . This is accomplished by declaring the predictor to be a factor. We will describe this in more detail in Chapter 15:

```
> lmoda <- lm(loss ~ factor(Fe), corrosion)
```

The fitted values are the means in each group and we put these on the plot:

```
> points(corrosion$Fe, fitted(lmoda), pch=3)
```

We can now compare the two models in the usual way:

```
> anova(lmod, lmoda)
Analysis of Variance Table

Model 1: loss ~ Fe
Model 2: loss ~ factor(Fe)
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1      11      102.9
2       6       11.8  5    91.1    9.28 0.0086
```

The low p-value indicates that we must conclude that there is a lack of fit. The reason is that the pure error SD $\sqrt{11.8/6} = 1.4$, is substantially less than the regression standard error of 3.06. We might investigate models other than a straight line although no obvious alternative is suggested by the plot. Before considering other models, we would first find out whether the replicates are genuine. Perhaps the low pure error SD can be explained by some correlation in the measurements. They may not be genuine replicates. Another possible explanation is that an unmeasured third variable is causing the lack of fit.

When there are replicates, it is impossible to get a perfect fit. Even when there is a parameter assigned to each group of x -values, the residual sum of squares will not be zero. For the factor-based model above, the R^2 is 99.7%. So even this saturated model does not attain a 100% value for R^2 . For these data, it is a small difference but in other cases, the difference can be substantial. In these cases, one should realize that the maximum R^2 that may be attained might be substantially less than 100% and so perceptions about a good value for R^2 should be downgraded appropriately.

These methods are good for detecting lack of fit, but if the null hypothesis is accepted, we cannot conclude that we have the true model. After all, it may be that we just did not have enough data to detect the inadequacies of the model. All we can say is that the model is not contradicted by the data.

When there are no replicates, it may be possible to group the responses for similar x , but this is not straightforward. It is also possible to detect lack of fit by less formal, graphical methods as described in Chapter 6.

A more general question is how good a fit do you really want? By increasing the complexity of the model, it is possible to fit the data more closely. By using as many parameters as data points, we can fit the data exactly. Very little is achieved by doing this since we learn nothing beyond the data and any predictions made using such a model will tend to have high variance. The question of how complex a model to fit is difficult and fundamental. For example, we can fit the mean responses for the previous example exactly using a sixth order polynomial:

```
> lmodp <- lm(loss ~ Fe+I(Fe^2)+I(Fe^3)+I(Fe^4)+I(Fe^5)+I(Fe^6),
  corrosion)
```

Now look at this fit:

```
> plot(loss ~ Fe, data=corrosion, ylim=c(60,130))
> points(corrosion$Fe, fitted(lmoda), pch=3)
> grid <- seq(0,2, len=50)
> lines(grid, predict(lmodp, data.frame(Fe=grid)))
```


as shown in the right panel of Figure 8.2. The fit of this model is excellent — for example:

```
> summary(lmodp) $r.squared
[1] 0.99653
```

but it is clearly ridiculous. There is no plausible reason corrosion loss should suddenly drop at 1.7 and thereafter increase rapidly. This is a consequence of overfitting the data. This illustrates the need not to become too focused on measures of fit like R^2 . The fit needs to reflect knowledge of the subject matter and simplicity in modeling is a virtue.

8.4 Robust Regression

When the errors are normally distributed, least squares regression is best. But when the errors follow some other distribution, other methods of model fitting may be considered. Short-tailed errors are not so much of a problem but long-tailed error distributions can cause difficulties because a few extreme cases can have a large effect on the fitted model.

It is important to think about the cause of these extreme values. In some cases, they are mistakes and we just want some way of automatically excluding them from the analysis. In other cases, the extreme values really were observed and are part of the process we are trying to model. Robust regression is designed to estimate the mean relationship between the predictors and response, $EY = X\beta$. It doesn't worry about where the outliers come from, but the analyst does need to think about this.

We developed methods of detecting outliers in Section 6.2.2. We could use these methods to remove the largest residuals as outliers and then just use least squares. This does not work very well if there are multiple outliers as such points can mutually influence the fit and effectively hide their presence. Furthermore, outlier rejection-based methods tend not to be statistically efficient for the estimation of β . Robust regression works better if you are dealing with more than one or two outliers.

In this section, we present two popular types of robust regression.

8.4.1 *M-Estimation*

M-estimates modify the least squares idea to choose β to minimize:

$$\sum_{i=1}^n \rho(y_i - x_i^T \beta)$$

Some possible choices among many for ρ are:

1. $\rho(x) = x^2$ is simply least squares.
2. $\rho(x) = |x|$ is called least absolute deviation (LAD) regression or L_1 regression.
- 3.

$$\rho(x) = \begin{cases} x^2/2 & \text{if } |x| \leq c \\ c|x| - c^2/2 & \text{otherwise} \end{cases}$$

is called Huber's method and is a compromise between least squares and LAD regression. c should be a robust estimate of σ . A value proportional to the median of $|\hat{\epsilon}|$ is suitable.

M-estimation is related to weighted least squares. The normal equations tell us that:

$$X^T(y - X\hat{\beta}) = 0$$

With weights and in non-matrix form this becomes:

$$\sum_{i=1}^n w_i x_{ij} (y_i - \sum_{j=1}^p x_{ij} \beta_j) = 0 \quad j = 1, \dots, p$$

Now differentiating the M-estimate criterion with respect to β_j and setting to zero we get:

$$\sum_{i=1}^n \rho'(y_i - \sum_{j=1}^p x_{ij} \beta_j) x_{ij} = 0 \quad j = 1, \dots, p$$

Now let $u_i = y_i - \sum_{j=1}^p x_{ij} \beta_j$ to get:

$$\sum_{i=1}^n \frac{\rho'(u_i)}{u_i} x_{ij} (y_i - \sum_{j=1}^p x_{ij} \beta_j) = 0 \quad j = 1, \dots, p$$

so we can make the identification of a weight function as $w(u) = \rho'(u)/u$. We find for our choices of ρ above that:

1. LS: $w(u)$ is constant and the estimator is simply ordinary least squares.
2. LAD: $w(u) = 1/|u|$. We see how the weight goes down as u moves away from zero so that more extreme observations get downweighted. Unfortunately, there is an asymptote at zero. This makes a weighting approach to fitting an LAD regression infeasible without some modification.
3. Huber:

$$w(u) = \begin{cases} 1 & \text{if } |u| \leq c \\ c/|u| & \text{otherwise} \end{cases}$$

We can see that this sensibly combines the downweighting of extreme cases with equal weighting for the middle cases.

There are many other choices for ρ that have been proposed. Computing an M-estimate requires some iteration because the weights depend on the residuals. The fitting methods alternate between fitting a WLS and recomputing the weights based on the residuals until convergence. We can get standard errors via WLS by $\hat{v}\hat{\beta} = \hat{\sigma}^2(X^T W X)^{-1}$ but we need to use a robust estimate of σ^2 .

We demonstrate the methods on the Galápagos Islands data. Using least squares first:

```
> data(gala, package="faraway")
> lsmold <- lm(Species ~ Area + Elevation + Nearest + Scruz +
  Adjacent, gala)
> summary(lsmold)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.06822	19.15420	0.37	0.7154
Area	-0.02394	0.02242	-1.07	0.2963
Elevation	0.31946	0.05366	5.95	0.000038
Nearest	0.00914	1.05414	0.01	0.9932
Scruz	-0.24052	0.21540	-1.12	0.2752
Adjacent	-0.07480	0.01770	-4.23	0.0003

n = 30, p = 6, Residual SE = 60.975, R-Squared = 0.77

Least squares works well when there are normal errors, but performs poorly for long-tailed errors. The Huber method is the default choice of the `rlm()` function, which is part of the MASS package of Venables and Ripley (2002).

```
> require(MASS)
> rlmmod <- rlm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent
, gala)
> summary(rlmmod)
Coefficients:
          Value Std. Error t value
(Intercept)  6.361  12.390    0.513
Area         -0.006  0.015   -0.421
Elevation     0.248  0.035    7.132
Nearest       0.359  0.682    0.527
Scruz        -0.195  0.139   -1.401
Adjacent     -0.055  0.011   -4.765

Residual standard error: 29.7 on 24 degrees of freedom
```

The R^2 statistic is not given because it does not make sense in the context of a robust regression. p -values are not given although we can use the asymptotic normality of the estimator to make approximate inferences using the t -values. We see that the same two predictors, Elevation and Adjacent, are significant. The numerical values of the coefficients have changed somewhat and the standard errors are generally smaller.

It is worth looking at the weights assigned by the final fit. We extract and name the smallest 10 weights. The remaining weights are all ones.

```
> wts <- rlmmod$w
> names(wts) <- row.names(gala)
> head(sort(wts), 10)
      SantaCruz  SantaMaria SanCristobal      Pinta  Gardner1
0.17458      0.30783      0.41423      0.53758      0.66130
      Espanola  Gardner2      Baltra  Bartolome  Caldwell
0.67950      0.85004      1.00000      1.00000      1.00000
```

We can see that a few islands are substantially discounted in the calculation of the robust fit. Provided we do not believe there are mistakes in the data for these cases, we should think carefully about what might be unusual about these islands.

The main purpose in analyzing these data is likely to explain the relationship between the predictors and the response. Although the robust fit gives numerically different output, the overall impression of what predictors are significant in explaining the response is unchanged. Thus the robust regression has provided some measure of confirmation. Furthermore, it has identified a few islands which are not fit so well by the model. If there had been more disagreement between the two sets of regression

outputs, we would know which islands are responsible and deserve a closer look. If there is a substantial difference between the two fits, we find the robust one more trustworthy.

Robust regression is not a panacea. M-estimation does not address large leverage points. It also does not help us choose which predictors to include or what transformations of the variables to make. For this data, we have seen that a transformation of the response may be helpful which would completely change the robust fit. Hence robust methods are just part of the regression modeling toolkit and not a replacement.

We can also do LAD regression using the `quantreg` package. The default option does LAD while other options allow for quantile regression:

```
> require(quantreg)
> llmod <- rq(Species ~ Area+Elevation+Nearest+Scruz+Adjacent, data=
  gala)
> summary(llmod)
Coefficients:
      coefficients lower bd upper bd
(Intercept)  1.31445   -19.87777  24.37411
Area         -0.00306   -0.03185  0.52800
Elevation    0.23211    0.12453  0.50196
Nearest      0.16366   -3.16339  2.98896
Scruz        -0.12314   -0.47987  0.13476
Adjacent     -0.05185   -0.10458  0.01739
```

Again, there is some change in the coefficients. The confidence intervals now suggest that `adjacent` is not significant.

For this example, we do not see any big qualitative difference in the coefficients and for want of evidence to the contrary, we might stick with least squares as the easiest to work with. Had we seen something different, we would need to find out the cause. Perhaps some group of observations were not being fit well and the robust regression excluded these points.

8.4.2 Least Trimmed Squares

The Huber and L_1 methods will still fail if the large errors are sufficiently numerous and extreme in value. For example, very bad data entry errors might be made or measurement equipment might malfunction in a serious way. We need methods that still fit the correct data well even in the presence of such problems. Least trimmed squares (LTS) is an example of a *resistant* regression method. Resistant methods are good for dealing with data where we expect a certain number of bad observations that we want to have no weight in the analysis.

LTS minimizes the sum of squares of the q smallest residuals, $\sum_{i=1}^q \hat{\epsilon}_{(i)}^2$ where q is some number less than n and (i) indicates sorting. This method has a high *breakdown* point because it can tolerate a large number of outliers depending on how q is chosen.

Let's see how it works on the Galapagos data:

```
> set.seed(123)
> ltsmod <- ltsreg(Species ~ Area + Elevation + Nearest + Scruz +
  Adjacent, gala)
> coef(ltsmod)
```

```
(Intercept)      Area      Elevation      Nearest      Scruz
 12.506684      1.545358      0.016725      0.523487     -0.094072
  Adjacent
-0.142592
```

The default choice of q is $\lfloor n/2 \rfloor + \lfloor (p+1)/2 \rfloor$ where $\lfloor x \rfloor$ indicates the largest integer less than or equal to x . A genetic algorithm is used by default to compute the coefficients and is non-deterministic. If you repeat the fit, you will get a slightly different result. This is why we have set the random number generator seed to ensure you get the same result as we did. It's not necessary to set the seed unless you want to have exact reproducibility.

An exhaustive search method can be used:

```
> ltsmod <- ltsreg(Species ~ Area + Elevation + Nearest + Scruz +
  Adjacent, gala, nsamp="exact")
> coef(ltsmod)
(Intercept)      Area      Elevation      Nearest      Scruz
 9.381145      1.543658      0.024125      0.811109     -0.117732
  Adjacent
-0.197923
```

This takes only a few seconds on a modern computer. For larger datasets, it will take much longer so this method might be impractical and the default method might be required.

This really does make substantial differences to the coefficients. For example, the `Area` coefficient is now substantially larger while the `Elevation` coefficient is substantially smaller than that seen in the least squares fit. However, we do not have the standard errors for the LTS regression coefficients. We now use a general method for inference that is especially useful when such theory is lacking — the bootstrap. We introduced this method in Section 3.6. The outline of the method is:

1. Generate ε^* by sampling with replacement from $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$.
2. Form $y^* = X\hat{\beta} + \varepsilon^*$.
3. Compute $\hat{\beta}^*$ from (X, y^*) .

We repeat this bootstrap process 1000 times. We construct a matrix of coefficients in which to save the results:

```
> bcoef <- matrix(0, 1000, 6)
> for(i in 1:1000){
  newy <- predict(ltsmod) + residuals(ltsmod)[sample(30, rep=T)]
  brg <- ltsreg(newy ~ Area + Elevation + Nearest + Scruz + Adjacent
    , gala, nsamp="best")
  bcoef[i,] <- brg$coef
}
```

It is not convenient to use the `nsamp="exact"` since that would require 1000 times the time it takes to make the original estimate. This may take an hour or so. Being impatient, we compromised and used the second best option of `nsamp="best"`. This likely means that our bootstrap estimates of variability will be somewhat on the high side. This illustrates a common practical difficulty with the bootstrap — it can take a long time to compute. Fortunately, this problem recedes as processor speeds increase.

We can make a 95% confidence interval for this parameter by taking the empirical quantiles:

```
> colnames(bcoef) <- names(coef(ltsmod))
> apply(bcoef, 2, function(x) quantile(x, c(0.025, 0.975)))
```

	(Intercept)	Area	Elevation	Nearest	Scruz	Adjacent
2.5%	2.0868	1.4976	-0.015118	0.18872	-0.269049	-0.23375
97.5%	24.9500	1.6230	0.065377	1.94555	0.080291	-0.15138

Zero lies outside the interval for Area, Nearest and Adjacent so we are confident that there is an effect for these predictors although the Nearest is marginal. We can get a better picture of the distribution by looking at the density and marking the confidence interval. The ggplot2 package produces clear density plots:

```
> require(ggplot2)
> bcoef <- data.frame(bcoef)
> p1 <- ggplot(bcoef, aes(x = Area)) + geom_density() + xlim(1.45, 1.65)
> p1 + geom_vline(xintercept=c(1.4976, 1.6230), linetype="dashed")
> p2 <- ggplot(bcoef, aes(x = Adjacent)) + geom_density() + xlim(-0.25, -0.13)
> p2 + geom_vline(xintercept=c(-0.23375, -0.15138), linetype="dashed")
```

See Figure 8.3. We see that the distribution is more peaked than a normal distribution with some long tails. This would be more accurate if we took more bootstrap resam-

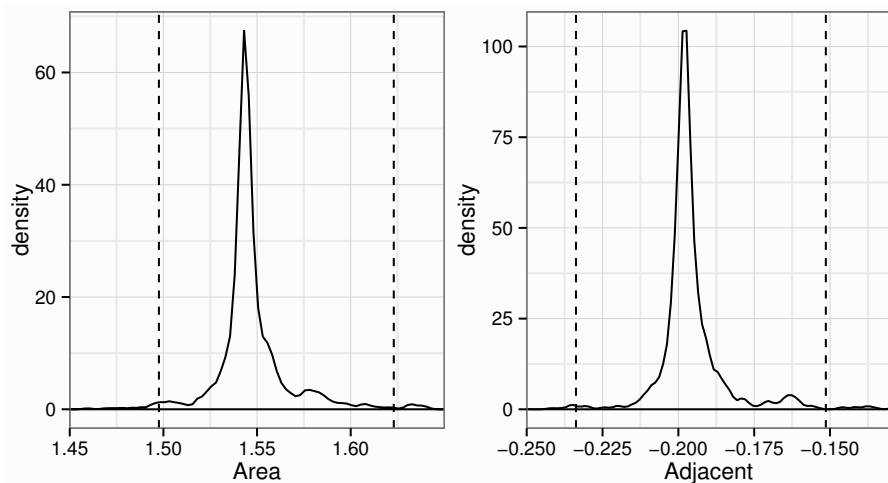


Figure 8.3 Bootstrap distribution of $\hat{\beta}_{Area}$ and $\hat{\beta}_{Adjacent}$ with 95% confidence intervals.

ples although 1000 is sufficient for our needs here. The conclusion is that the area variable is significant. That is in contrast to the conclusion from the least squares fit. Which estimates are best? An examination of the Cook distances for the least squares fit shows the island of Isabela to be very influential. If we exclude this island from the least squares fit, we find that:

```

> limod <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent
, gala, subset=(row.names(gala) != "Isabela"))
> summary(limod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  22.5861    13.4019    1.69   0.1055
Area          0.2957     0.0619    4.78  0.000080
Elevation     0.1404     0.0497    2.82  0.0096
Nearest      -0.2552     0.7217   -0.35  0.7269
Scruz        -0.0901     0.1498   -0.60  0.5534
Adjacent     -0.0650     0.0122  -5.32  0.000021

n = 29, p = 6, Residual SE = 41.646, R-Squared = 0.87

```

This fit is much closer to the LTS fit in that Area and Adjacent are very significant predictors. Thus, there are two routes to the same goal. We can use regression diagnostics in conjunction with least squares to identify bad or unusual points or we can use robust methods. The former approach is more flexible and allows for the discovery of a wider class of problems, but it is time consuming and does require human intervention. When data need to be quickly analyzed, perhaps without expert assistance or when large numbers of datasets need to be fitted, robust methods give some protection against aberrant data.

Another interesting point is that the M-estimate failed to identify the unusual island, Isabela, and gave similar results to the full data least squares fit. We can show similar behavior on another dataset — consider the star data presented in Section 6.2.2. We compute the least squares, Huber and LTS fits and display them in Figure 8.4:

```

> data(star, package="faraway")
> plot(light ~ temp, star)
> gs1 <- lm(light ~ temp, star)
> abline(coef(gs1))
> gs2 <- rlm(light ~ temp, star)
> abline(coef(gs2), lty=2)
> gs3 <- ltsreg(light ~ temp, star, nsamp="exact")
> abline(coef(gs3), lty=5)

```

Only LTS managed to capture the trend in the main group of points. The Huber estimate is almost the same as the least squares estimate.

Summary

1. Robust estimators provide protection against long-tailed errors, but they cannot overcome problems with the choice of model and its variance structure.
2. Robust estimates supply $\hat{\beta}$ and possibly standard errors without the associated inferential methods. Software and methodology for this inference require extra work. The bootstrap is a general-purpose inferential method which is useful in these situations.
3. Robust methods can be used in addition to least squares as a confirmatory method. You have cause to worry if the two estimates are far apart. The source of the difference should be investigated.

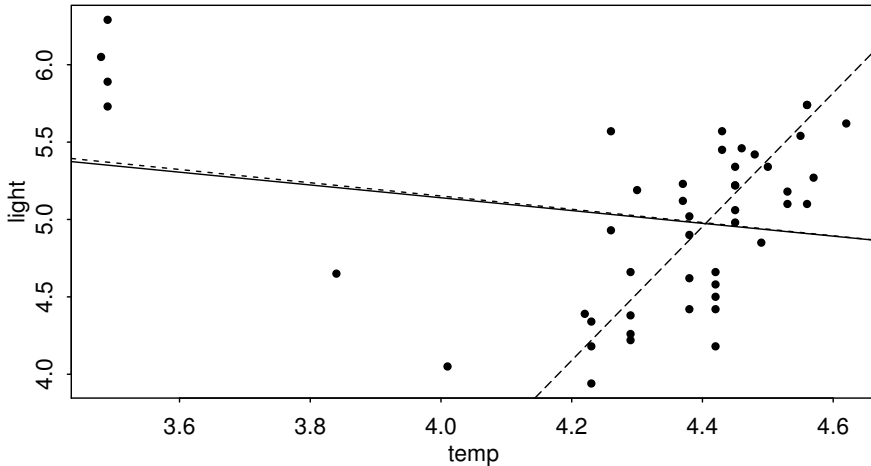


Figure 8.4 Regression fits compared. Least squares is the solid line, Huber is the dotted line and LTS is the dashed line. Only LTS finds the fit in the bulk of the data.

4. Robust estimates are useful when data need to be fit automatically without the intervention of a skilled analyst.

Exercises

1. Researchers at National Institutes of Standards and Technology (NIST) collected pipeline data on ultrasonic measurements of the depth of defects in the Alaska pipeline in the field. The depth of the defects were then remeasured in the laboratory. These measurements were performed in six different batches. It turns out that this batch effect is not significant and so can be ignored in the analysis that follows. The laboratory measurements are more accurate than the in-field measurements, but more time consuming and expensive. We want to develop a regression equation for correcting the in-field measurements.

- (a) Fit a regression model $\text{Lab} \sim \text{Field}$. Check for non-constant variance.
- (b) We wish to use weights to account for the non-constant variance. Here we split the range of `Field` into 12 groups of size nine (except for the last group which has only eight values). Within each group, we compute the variance of `Lab` as `varlab` and the mean of `Field` as `meanfield`. Supposing `pipeline` is the name of your data frame, the following R code will make the needed computations:

```
> i <- order(pipeline$Field)
> npipe <- pipeline[i,]
> ff <- gl(12,9)[-108]
> meanfield <- unlist(lapply(split(npipe$Field, ff), mean))
> varlab <- unlist(lapply(split(npipe$Lab, ff), var))
```


Suppose we guess that the the variance in the response is linked to the predictor in the following way:

$$\text{var}(Lab) = a_0 Field^{a_1}$$

Regress `log(varlab)` on `log(meanfield)` to estimate a_0 and a_1 . (You might choose to remove the last point.) Use this to determine appropriate weights in a WLS fit of `Lab` on `Field`. Show the regression summary.

- (c) An alternative to weighting is transformation. Find transformations on `Lab` and/or `Field` so that in the transformed scale the relationship is approximately linear with constant variance. You may restrict your choice of transformation to square root, log and inverse.
2. Using the `divusa` data, fit a regression model with `divorce` as the response and `unemployed`, `femlab`, `marriage`, `birth` and `military` as predictors.
 - (a) Make two graphical checks for correlated errors. What do you conclude?
 - (b) Allow for serial correlation with an AR(1) model for the errors. (Hint: Use maximum likelihood to estimate the parameters in the GLS fit by `gls(..., method="ML", ...)`). What is the estimated correlation and is it significant? Does the GLS model change which variables are found to be significant?
 - (c) Speculate why there might be correlation in the errors.
 3. For the `salmonella` dataset, fit a linear model with `colonies` as the response and `log(dose+1)` as the predictor. Check for lack of fit.
 4. For the `cars` dataset, fit a linear model with `distance` as the response and `speed` as the predictor. Check for lack of fit.
 5. Using the `stackloss` data, fit a model with `stack.loss` as the response and the other three variables as predictors using the following methods:
 - (a) Least squares
 - (b) Least absolute deviations
 - (c) Huber method
 - (d) Least trimmed squares

Compare the results. Now use diagnostic methods to detect any outliers or influential points. Remove these points and then use least squares. Compare the results.
 6. Using the `cheddar` data, fit a linear model with `taste` as the response and the other three variables as predictors.
 - (a) Suppose that the observations were taken in time order. Create a time variable. Plot the residuals of the model against time and comment on what can be seen.
 - (b) Fit a GLS model with same form as above but now allow for an AR(1) correlation among the errors. Is there evidence of such a correlation?
 - (c) Fit a LS model but with time now as an additional predictor. Investigate the significance of time in the model.
 - (d) The last two models have both allowed for an effect of time. Explain how they do this differently.

- (e) Suppose you were told, contrary to prior information, that the observations are not in time order. How would this change your interpretation of the model from (c)?
7. The `crawl` dataset contains data on a study looking at the age when babies learn to crawl as a function of ambient temperatures. There is additional information about the number of babies studied each month and the variation in the response. Make an appropriate choice of weights to investigate the relationship between crawling age and temperature.
8. The `gammaray` dataset shows the x-ray decay light curve of a gamma ray burst. Build a model to predict the flux as a function time that uses appropriate weights.
9. Use the `fat` data, fitting the model described in Section 4.2.
- (a) Fit the same model but now using Huber's robust method. Comment on any substantial differences between this model and the least squares fit.
- (b) Identify which two cases have the lowest weights in the Huber fit. What is unusual about these two points?
- (c) Plot weight (of the man) against height. Identify the two outlying cases. Are these the same as those identified in the previous question? Discuss.

Transformation

Transformations of the response and/or predictors can improve the fit and correct violations of model assumptions such as non-constant error variance. We may also consider adding additional predictors that are functions of the existing predictors like quadratic or cross-product terms.

9.1 Transforming the Response

We start with some general considerations about transforming the response. Suppose that you are contemplating a logged response in a simple regression situation:

$$\log y = \beta_0 + \beta_1 x + \varepsilon$$

In the original scale of the response, this model becomes:

$$y = \exp(\beta_0 + \beta_1 x) \cdot \exp(\varepsilon) \tag{9.1}$$

In this model, the errors enter *multiplicatively* and not *additively* as they usually do. So the use of standard regression methods for the logged response model requires that we believe the errors enter multiplicatively in the original scale. Alternatively, for small ε , $\exp(\varepsilon) \approx 1 + \varepsilon$. Substituting this into (9.1) would result in a model with additive errors but with non-constant variance.

If we propose the model:

$$y = \exp(\beta_0 + \beta_1 x) + \varepsilon$$

then we cannot linearize this model directly and nonlinear regression methods might need to be applied. However, we might also use the $\exp(\varepsilon) \approx 1 + \varepsilon$ approximation to get a weighted linear model.

In practice, we may not know how the errors enter the model additively, multiplicatively or otherwise. The best approach is to try different transforms to get the structural form of the model right and worry about the error component later. We can then check the residuals to see whether they satisfy the conditions required for linear regression. If there is a problem, we have several potential solutions as discussed in earlier chapters.

Although you may transform the response, you will probably need to express predictions in the original scale. This requires back transforming. For example, in the logged model above, your prediction would be $\exp(\hat{y}_0)$. If your prediction confidence interval in the logged scale was $[l, u]$, then you would use $[\exp l, \exp u]$. This

interval will not be symmetric, but this may be desirable. For example, the untransformed prediction intervals for the Galápagos data went below zero in Section 4.1. Transformation of the response avoids this problem.

Regression coefficients will need to be interpreted with respect to the transformed scale. There is no straightforward way of back transforming them to values that can be interpreted in the original scale. You cannot directly compare regression coefficients for models where the response transformation is different. Difficulties of this type may dissuade you from transforming the response even if this requires the use of another type of model, such as a generalized linear model.

When you use a log transformation on the response, the regression coefficients have a particular interpretation:

$$\begin{aligned}\log \hat{y} &= \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p \\ \hat{y} &= e^{\hat{\beta}_0} e^{\hat{\beta}_1 x_1} \cdots e^{\hat{\beta}_p x_p}\end{aligned}$$

An increase of one in x_1 would multiply the predicted response (in the original scale) by $e^{\hat{\beta}_1}$. Thus when a log scale is used, the regression coefficients can be interpreted in a multiplicative rather than an additive manner. A useful approximation for quick interpretations is that $\log(1+x) \approx x$ for small values of x . So for example, suppose $\hat{\beta}_1 = 0.09$; then an increase of one in x_1 would lead to about a 0.09 increase in $\log y$ which is a 9% increase in y . The approximation is quite good up to about ± 0.25 .

Box–Cox transformation

The Box–Cox method is a popular way to determine a transformation on the response. It is designed for strictly positive responses and chooses the transformation to find the best fit to the data. The method transforms the response $y \rightarrow g_\lambda(y)$ where the family of transformations indexed by λ is:

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log y & \lambda = 0 \end{cases}$$

For fixed $y > 0$, $g_\lambda(y)$ is continuous in λ . Choose $\hat{\lambda}$ using maximum likelihood. The profile log-likelihood assuming normality of the errors is:

$$L(\lambda) = -\frac{n}{2} \log(\text{RSS}_\lambda/n) + (\lambda - 1) \sum \log y_i$$

where RSS_λ is the residual sum of squares when $g_\lambda(y)$ is the response. You can compute $\hat{\lambda}$ numerically to maximize this. If the purpose of the regression model is prediction, then use $y^{\hat{\lambda}}$ as the response (no need to use $(y^{\hat{\lambda}} - 1)/\hat{\lambda}$, as the rescaling is just so $g_\lambda \rightarrow \log$ as $\lambda \rightarrow 0$ which maintains continuity in the likelihood). If explaining the model is important, you should round $\hat{\lambda}$ to the nearest interpretable value. For example, if $\hat{\lambda} = 0.46$, it would be hard to explain what this new response means, but \sqrt{y} might be easier.

Transforming the response can make the model harder to interpret so we do not

want to do it unless it is really necessary. One way to check this is to form a confidence interval for λ . A $100(1 - \alpha)\%$ confidence interval for λ is:

$$\{\lambda : L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi_1^2(1-\alpha)\}$$

This interval can be derived by inverting the likelihood ratio test of the hypothesis that $H_0 : \lambda = \lambda_0$ which uses the statistic $2(L(\hat{\lambda}) - L(\lambda_0))$ having approximate null distribution χ_1^2 . The confidence interval also tells you how much it is reasonable to round λ for the sake of interpretability.

We check whether the response in the savings data needs transformation. We will need the `boxcox` function from the MASS package:

```
> require(MASS)
```

Try it out on the savings dataset and plot the results:

```
> data(savings, package="faraway")
> lmod <- lm(sr ~ pop15+pop75+dpi+ddpi, savings)
> boxcox(lmod, plotit=T)
> boxcox(lmod, plotit=T, lambda=seq(0.5, 1.5, by=0.1))
```

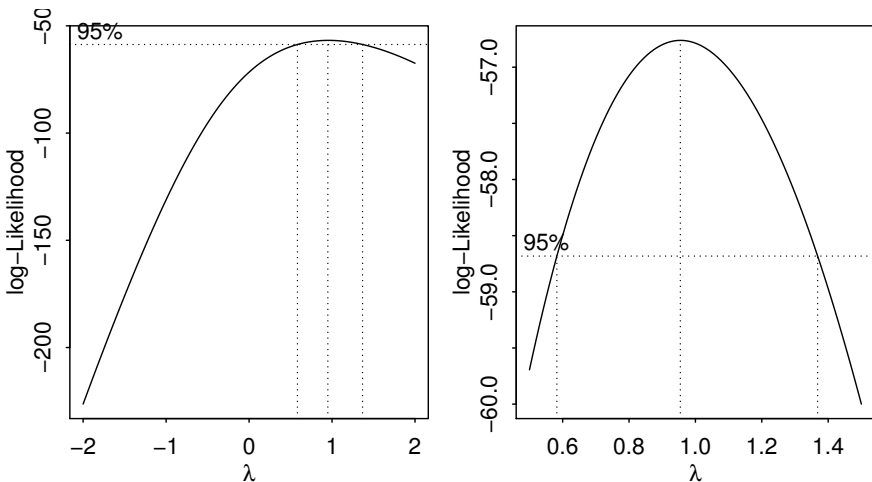


Figure 9.1 Log-likelihood plots for the Box–Cox transformation of the savings data.

The first plot shown in Figure 9.1 is too broad. We narrowed the range of λ in the second plot so that we can read off the confidence interval more easily. The confidence interval for λ runs from about 0.6 to about 1.4. We can see that there is no good reason to transform.

Now consider the Galápagos Islands dataset analyzed earlier:

```
> data(gala, package="faraway")
```

```
> lmod <- lm(Species ~ Area + Elevation + Nearest + Scrutz + Adjacent,
             gala)
> boxcox(lmod, lambda=seq(-0.25,0.75,by=0.05),plotit=T)
```

The plot is shown in the left panel of Figure 9.2. We see that perhaps a cube root

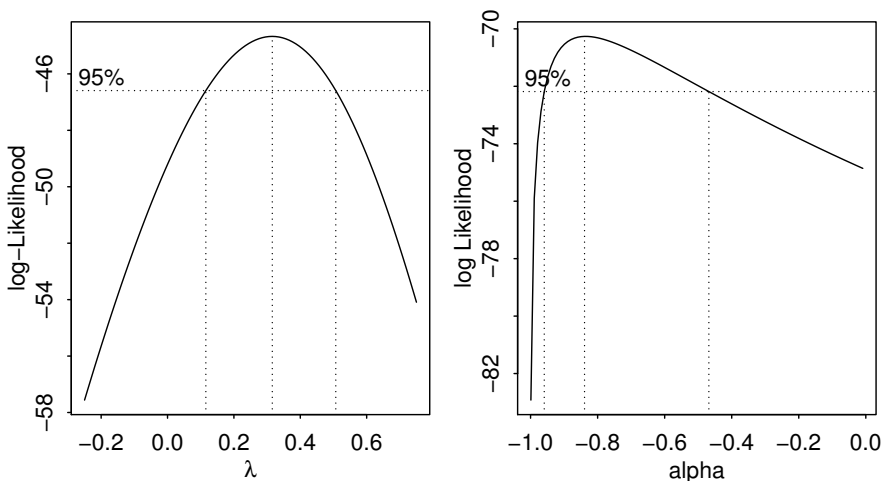


Figure 9.2 Log-likelihood plot for the Box–Cox transformation of the Galápagos data is shown on the left and the log-additive transformation on the leafburn data is shown on the right.

transformation might be best here. A square root is also a possibility, as this falls just within the confidence intervals. Certainly there is a strong need to transform.

Some general considerations concerning the Box–Cox method are:

1. The Box–Cox method gets upset by outliers — if you find $\hat{\lambda} = 5$, then this is probably the reason — there can be little justification for actually making such an extreme transformation.
2. If some $y_i < 0$, we can add a constant to all the y . This can work provided the constant is small, but this is an inelegant solution.
3. If $\max_i y_i / \min_i y_i$ is small, then the Box–Cox will not have much real effect because power transforms are well approximated by linear transformations over short intervals far from the origin.
4. There is some doubt whether the estimation of λ counts as an extra parameter to be considered in the degrees of freedom. This is a difficult question since λ is not a linear parameter and its estimation is not part of the least squares fit.

The Box–Cox method is not the only way of transforming the predictors. Another family of transformations is given by $g_\alpha(y) = \log(y + \alpha)$. We can illustrate the value of this using some data from Steel and Torrie (1980) on the burn time of tobacco leaves as a function of three chemical constituents. The `logtrans` function

in the `MASS` package performs the computation in a similar manner to the Box-Cox procedure:

```
> lmod <- lm(burntime ~ nitrogen+chlorine+potassium, leafburn)
> logtrans(lmod, plotit=TRUE, alpha=seq(-min(leafburn$burntime)
+0.001, 0, by=0.01))
```

The plot is shown in the right panel of Figure 9.2. The Box-Cox analysis of this model recommends $\hat{\lambda} = -0.34$ and the confidence interval bounds this away from zero so it seems we cannot use the more interpretable log transformation. But the plot shows that we can use a log transformation provided we subtract about 0.85 from the response. This value may be interpretable as a start-up time for the fire to get going.

For responses that are proportions (or percentages), the logit transformation, $\log(y/(1-y))$, is often used, while for responses that are correlations, Fisher's z transform, $y = 0.5 \log((1+y)/(1-y))$, is worth considering.

9.2 Transforming the Predictors

You can take a Box-Cox style approach for each of the predictors, choosing the transformation to minimize the RSS. However, this usually not the best way to find good transformations on the predictors. You can also use graphical methods such as partial residual plots to select transforming the predictors. These methods are designed to replace x in the model with $f(x)$ for some chosen f . The methods we consider below are more general in that they replace x with more than one term — $f(x) + g(x) + \dots$. This allows more flexibility since each additional term carries a parameter.

9.3 Broken Stick Regression

Sometimes we have reason to believe that different linear regression models apply in different regions of the data. For example, in the analysis of the savings data, we observed that there were two groups in the data and we might want to fit a different model to the two parts. Suppose we focus attention on just the `pop15` predictor for ease of presentation. We fit the two regression models depending on whether `pop15` is greater or less than 35%. The two fits are seen in Figure 9.3.

```
> lmod1 <- lm(sr ~ pop15, savings, subset=(pop15 < 35))
> lmod2 <- lm(sr ~ pop15, savings, subset=(pop15 > 35))
> plot(sr ~ pop15, savings, xlab="Pop'n under 15", ylab="Savings Rate")
> abline(v=35, lty=5)
> segments(20, lmod1$coef[1]+lmod1$coef[2]*20, 35, lmod1$coef[1]+lmod1$
coef[2]*35)
> segments(48, lmod2$coef[1]+lmod2$coef[2]*48, 35, lmod2$coef[1]+lmod2$
coef[2]*35)
```

A possible objection to this subsetted regression fit is that the two parts of the fit do not meet at the join. If we believe the fit should be continuous as the predictor

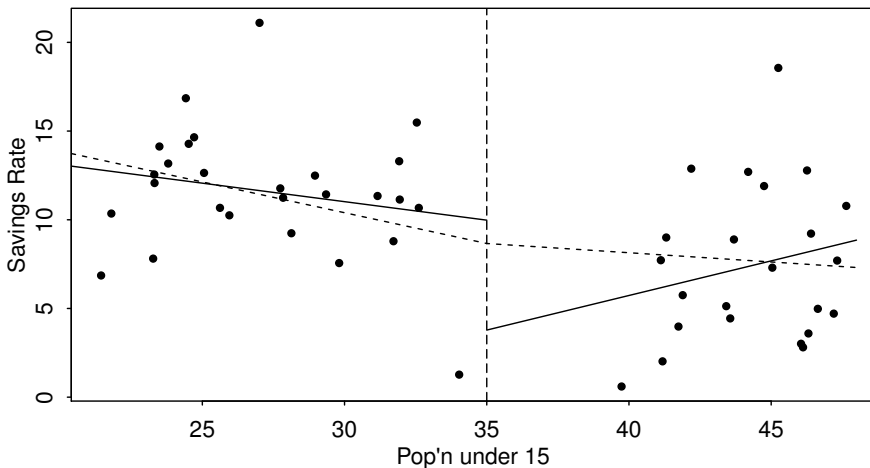


Figure 9.3 Subset regression fit is shown with the solid line, while the broken stick regression is shown with the dotted line.

varies, we should consider the broken stick regression fit. Define two *basis functions*:

$$B_l(x) = \begin{cases} c-x & \text{if } x < c \\ 0 & \text{otherwise} \end{cases}$$

and:

$$B_r(x) = \begin{cases} x-c & \text{if } x > c \\ 0 & \text{otherwise} \end{cases}$$

where c marks the division between the two groups. B_l and B_r form a first-order spline basis with a knotpoint at c . Sometimes B_l and B_r are called hockey-stick functions because of their shape. We can now fit a model of the form:

$$y = \beta_0 + \beta_1 B_l(x) + \beta_2 B_r(x) + \varepsilon$$

using standard regression methods. The two linear parts are guaranteed to meet at c . Notice that this model uses only three parameters in contrast to the four total parameters used in the subsetted regression illustrated before. A parameter has been saved by insisting on the continuity of the fit at c .

We define the two hockey-stick functions, compute and display the fit:

```
> lhs <- function(x) ifelse(x < 35, 35-x, 0)
> rhs <- function(x) ifelse(x < 35, 0, x-35)
> lmod <- lm(sr ~ lhs(pop15) + rhs(pop15), savings)
> x <- seq(20, 48, by=1)
> py <- lmod$coef[1] + lmod$coef[2]*lhs(x) + lmod$coef[3]*rhs(x)
> lines(x, py, lty=2)
```

The two (dotted) lines now meet at 35, as shown in Figure 9.3. The intercept of this model is the value of the response at the join.

We might question which fit is preferable in this particular instance. For the high pop15 countries, we see that the imposition of continuity causes a change in sign for the slope of the fit. We might argue that because the two groups of countries are so different and there are so few countries in the middle region, we might not want to impose continuity at all.

We can have more than one knotpoint simply by defining more basis functions with different knotpoints. Broken stick regression is sometimes called *segmented regression*. Allowing the knotpoints to be parameters is worth considering, but this will result in a nonlinear model. This can be implemented using the *segmented* package in R which is derived from Muggeo (2003).

9.4 Polynomials

Another way of generalizing the $X\beta$ part of the model is to add polynomial terms. In the one-predictor case, we have:

$$y = \beta_0 + \beta_1 x + \dots + \beta_d x^d + \varepsilon$$

which allows for a more flexible relationship. We usually do not believe the polynomial exactly represents any underlying reality, but it can allow us to model expected features of the relationship. A quadratic term allows for a predictor to have an optimal setting. For example, there may be a best temperature for baking bread – a hotter or colder temperature may result in a less tasty outcome. If you believe a predictor behaves in this manner, it makes sense to add a quadratic term.

There are two common ways to choose d . We can keep adding terms until the added term is not statistically significant. Alternatively, we can start with a large d and eliminate non-statistically significant terms starting with the highest order term. Let's see if we can use polynomial regression on the `ddpi` variable in the `savings` data. First, fit a linear model:

```
> summary(lm(sr ~ ddpi, savings))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.883      1.011     7.80  4.5e-10
ddpi           0.476      0.215     2.22   0.031
```

The p -value of `ddpi` is significant so move on to a quadratic term:

```
> summary(lm(sr ~ ddpi+I(ddpi^2), savings))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.1304      1.4347     3.58  0.00082
ddpi           1.7575      0.5377     3.27  0.00203
I(ddpi^2)     -0.0930      0.0361    -2.57  0.01326
```

Again the p -value of `ddpi2` is significant so move on to a cubic term:

```
> summary(lm(sr ~ ddpi+I(ddpi^2)+I(ddpi^3), savings))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.145360      2.198606     2.34   0.024
ddpi         1.746017      1.380455     1.26   0.212
```

```
I(ddpi^2)    -0.090967    0.225598    -0.40    0.689
I(ddpi^3)    -0.000085    0.009374    -0.01    0.993
```

The p -value of ddpi^3 is not significant so stick with the quadratic. You can check that starting from a large model, say starting from the fourth power, and working downward gives the same result.

Usually it is a bad idea to eliminate lower order terms from the model before the higher order terms even if they are not statistically significant. An additive change in scale would change the t -statistic of all but the highest order term. We would not want the conclusions of our study to be sensitive to such changes in the scale which ought to be inconsequential. For example, suppose we transform ddpi by subtracting 10 and refit the quadratic model:

```
> savings <- data.frame(savings,mddpi=savings$ddpi-10)
> summary(lm(sr ~ mddpi+I(mddpi^2), savings))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.4070    1.4240    9.41  2.2e-12
mddpi        -0.1022    0.3027   -0.34  0.737
I(mddpi^2)   -0.0930    0.0361   -2.57  0.013
```

We see that the quadratic term remains unchanged, but the linear term is now insignificant. Since there is often no necessary importance to zero on a scale of measurement, there is no good reason to remove the linear term in this model but not in the previous version. If you do remove lower order terms from, say, a quadratic model, be aware that this has some special meaning. Setting the intercept to zero means the regression passes through the origin while setting the linear term to zero would mean that the response is optimized at a predictor value of zero.

You have to refit the model each time a term is removed which is inconvenient and for large d there can be a problem with numerical stability. Orthogonal polynomials get around this problem by defining:

$$\begin{aligned} z_1 &= a_1 + b_1x \\ z_2 &= a_2 + b_2x + c_2x^2 \\ z_3 &= a_3 + b_3x + c_3x^2 + d_3x^3 \end{aligned}$$

etc. where the coefficients a, b, c, \dots are chosen so that $z_i^T z_j = 0$ when $i \neq j$. The expressions z are called orthogonal polynomials. The value of orthogonal polynomials has declined with advances in computing speeds although they are still worth knowing about because of their numerical stability and ease of use. The `poly()` function constructs orthogonal polynomials:

```
> lmod <- lm(sr ~ poly(ddpi, 4), savings)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.6710    0.5846   16.54 <2e-16
poly(ddpi, 4)1  9.5590    4.1338    2.31  0.025
poly(ddpi, 4)2 -10.4999    4.1338   -2.54  0.015
poly(ddpi, 4)3  -0.0374    4.1338   -0.01  0.993
poly(ddpi, 4)4   3.6120    4.1338    0.87  0.387

n = 50, p = 5, Residual SE = 4.134, R-Squared = 0.22
```

We can come to the same conclusion as above, that the quadratic model is best, with just this summary.

You can also define polynomials in more than one variable. These are sometimes called *response surface* models. A second degree model would be:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_{11}x_1^2 + \beta_{22}x_2^2 + \beta_{12}x_1x_2$$

an example of which could be fit as:

```
> lmod <- lm(sr ~ polym(pop15, ddpi, degree=2), savings)
```

We can construct a perspective plot of the fitted surface as follows. We compute the fit on a 10×10 grid of values covering the range of the predictors and see the result in Figure 9.4.

```
> pop15r <- seq(20, 50, len=10)
> ddpir <- seq(0, 20, len=10)
> pgrid <- expand.grid(pop15=pop15r, ddpi=ddpir)
> pv <- predict(lmod, pgrid)
> persp(pop15r, ddpir, matrix(pv, 10, 10), theta=45, xlab="Pop under
  15", ylab="Growth", zlab = "Savings rate", ticktype="detailed",
  shade = 0.25)
```

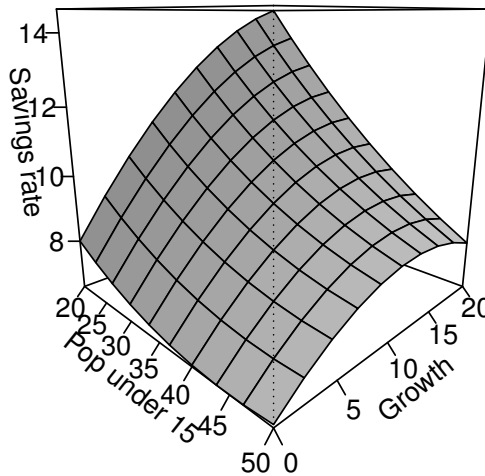


Figure 9.4 Perspective plot of the quadratic surface fit to the saving data.

9.5 Splines

Polynomials have the advantage of smoothness, but the disadvantage is that each data point affects the fit globally. This is because the power functions used for the

polynomials take nonzero values across the whole range of the predictor. In contrast, the broken stick regression method localizes the influence of each data point to its particular segment which is good, but we do not have the same smoothness as with the polynomials. There is a way we can combine the beneficial aspects of both these methods — smoothness and local influence — by using *B-spline* basis functions.

We can see why splines might be helpful by using a simulated example. Suppose we know the true model is:

$$y = \sin^3(2\pi x^3) + \varepsilon, \quad \varepsilon \sim N(0, (0.1)^2)$$

The advantage of using simulated data is that we can see how close our methods come to the truth. We generate the data and display them in the first plot of Figure 9.5.

```
> funky <- function(x) sin(2*pi*x^3)^3
> x <- seq(0,1,by=0.01)
> y <- funky(x) + 0.1*rnorm(101)
> matplot(x,cbind(y,funky(x)),type="pl",ylab="y",pch=20, lty=1, col=1)
```

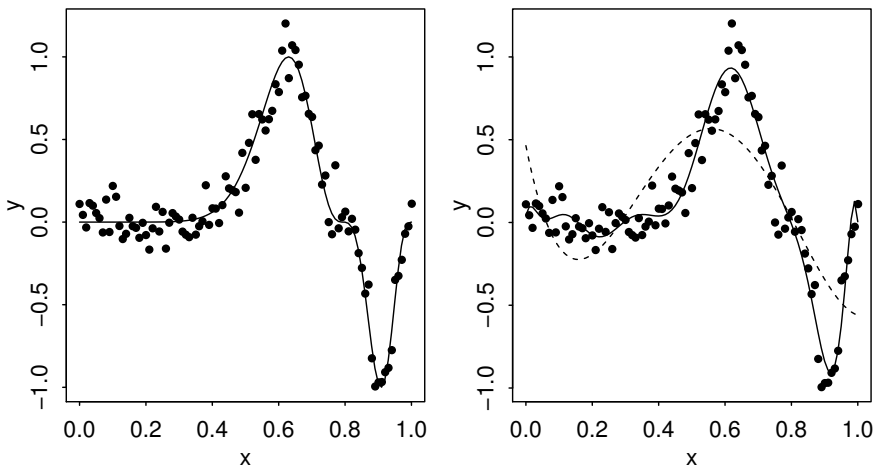


Figure 9.5 Data and true function shown on the left. Orthogonal polynomial of order 4 (dashed) and order 12 (solid) shown on the right.

We see how orthogonal polynomial bases of orders 4 and 12 do in fitting these data:

```
> g4 <- lm(y ~ poly(x,4))
> g12 <- lm(y ~ poly(x,12))
> matplot(x,cbind(y,g4$fit,g12$fit),type="pl", ylab="y",lty=c(1,2),
  pch=20, col=1)
```

The two fits are shown in the second panel of Figure 9.5. We see that order 4 is a clear underfit; order 12 is much better although the fit is too wiggly in the first section and misses the point of inflection around $x = 0.8$.

We may define a cubic B-spline basis on the interval $[a, b]$ by the following requirements on the interior basis functions with knotpoints at t_1, \dots, t_k :

1. A given basis function is nonzero on an interval defined by four successive knots and zero elsewhere. This property ensures the local influence property.
2. The basis function is a cubic polynomial for each subinterval between successive knots.
3. The basis function is continuous and is also continuous in its first and second derivatives at each knotpoint. This property ensures the smoothness of the fit.
4. The basis function integrates to one over its support.

The basis functions at the ends of the interval are defined a little differently to ensure continuity in derivatives at the edge of the interval. A full definition of B-splines and more details about their properties may be found in de Boor (2002). The broken stick regression is an example of the use of linear splines.

We now create the B-spline basis. You need to have three additional knots at the start and end to get the right basis. I have chosen to the knot locations to put more in the regions of greater curvature. I have used 12 basis functions for comparability to the orthogonal polynomial fit:

```
> require(splines)
> knots <- c(0, 0, 0, 0.2, 0.4, 0.5, 0.6, 0.7, 0.8, 0.85, 0.9, 1, 1, 1)
> bx <- splineDesign(knots, x)
> lmodb <- lm(y ~ bx -1)
> matplot(x, bx, type="l", col=1)
> matplot(x, cbind(y, lmodb$fit), type="pl", ylab="y", pch=20, lty=1,
  col=1)
```

The basis functions are seen in the first panel of Figure 9.6 while the fit appears in the second panel. We see that the fit comes very close to the truth. Regression splines are useful for fitting functions with some flexibility provided we have enough data. We can form basis functions for all the predictors in our model but we need to be careful not to use up too many degrees of freedom.

A related alternative to regression splines is smoothing splines. Suppose the model is $y_i = f(x_i) + \varepsilon_i$, so in the spirit of least squares, we might choose \hat{f} to minimize the sum of squares. Unfortunately, the unhelpful solution is $\hat{f}(x_i) = y_i$. This “join the dots” regression is almost certainly too rough. Instead, suppose we choose \hat{f} to minimize a modified least squares criterion:

$$\frac{1}{n} \sum (Y_i - f(x_i))^2 + \lambda \int [f''(x)]^2 dx$$

where $\lambda > 0$ controls the amount of smoothing and $\int [f''(x)]^2 dx$ is a *roughness penalty*. When f is rough, the penalty is large, but when f is smooth, the penalty is small. Thus the two parts of the criterion balance fit against smoothness. This is the *smoothing spline* fit. We illustrate this in Figure 9.7. We have used the default choice of smoothing which is derived from crossvalidation. The point of inflexion is captured well but the smoother part of the function on the left is fit too roughly.

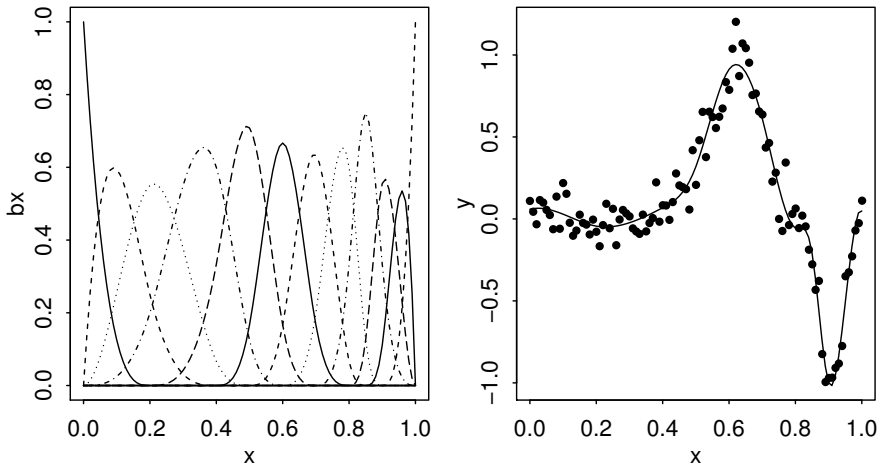


Figure 9.6 Cubic spline basis function on the left, cubic spline fit to the data on the right.

```
> ssf <- smooth.spline(x,y)
> matplot(x,cbind(y,ssf$y),type="pl",ylab="y", lty=1, pch=20, col=1)
```

9.6 Additive Models

Searching for good transformations on the predictors is difficult when there are multiple predictors. Changing the transformation on one predictor may change the best choice of transformation on another predictor. Fortunately, there is a way to simultaneously choose the transformations. An *additive* model takes the form :

$$y = \alpha + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p) + \varepsilon$$

The linear terms of the form $\beta_i X_i$ have been replaced with more flexible functional forms $f_i(X_i)$. The `mgcv` package can be used to fit such models:

```
> require(mgcv)
> gamod <- gam(sr ~ s(pop15) + s(pop75) + s(dpi) + s(ddpi), data=
  savings)
> plot(gamod)
```

The fitted functions f_i are computed using smoothing splines and are shown in Figure 9.8. Crossvalidation is used to choose the amount of smoothing. For `dpi` and `ddpi` the chosen amount of smoothing was as smooth as possible, producing a linear fit. This is also almost true for `pop75` such that a linear approximation would be very close. The interesting fit occurs for `pop15`. There is a suggestion of a minimum around 38% although given the width of the 95% confidence band and the sparsity of data as shown by the *rug* at the bottom of the display, we cannot be sure about this. It would also be possible to draw a straight line between the two bands, indicating that a linear fit may be adequate.

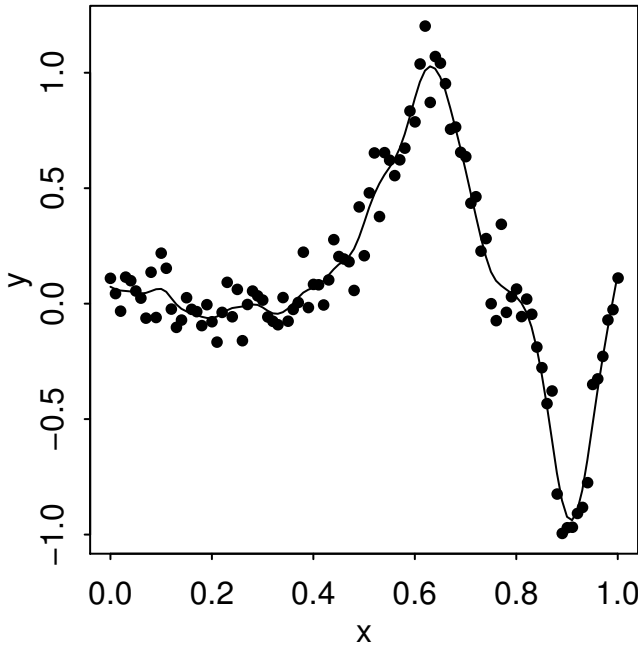


Figure 9.7 Smoothing spline fit to the simulated data.

9.7 More Complex Models

Regression analysis relies on the skill of the human analyst to make judgements about graphical displays and to incorporate subject area knowledge. When the purpose of the analysis is explanation or the sample size is relatively small, regression analysis compares well to more complex alternatives. The linear model produces interpretable parameters which are essential if we want to gain some understanding of the relationship between the variables. If we do not have much data, it is hard to justify a more complex approach.

For larger datasets where prediction is the goal, more recently developed complex models using methods from machine learning may be more effective. This is because these methods will be able to fit the data more flexibly while keeping the number of parameters under control. See Friedman, Hastie, and Tibshirani (2008).

Exercises

1. The `aatemp` data come from the U.S. Historical Climatology Network. They are the annual mean temperatures (in degrees F) in Ann Arbor, Michigan going back about 150 years.
 - (a) Is there a linear trend?

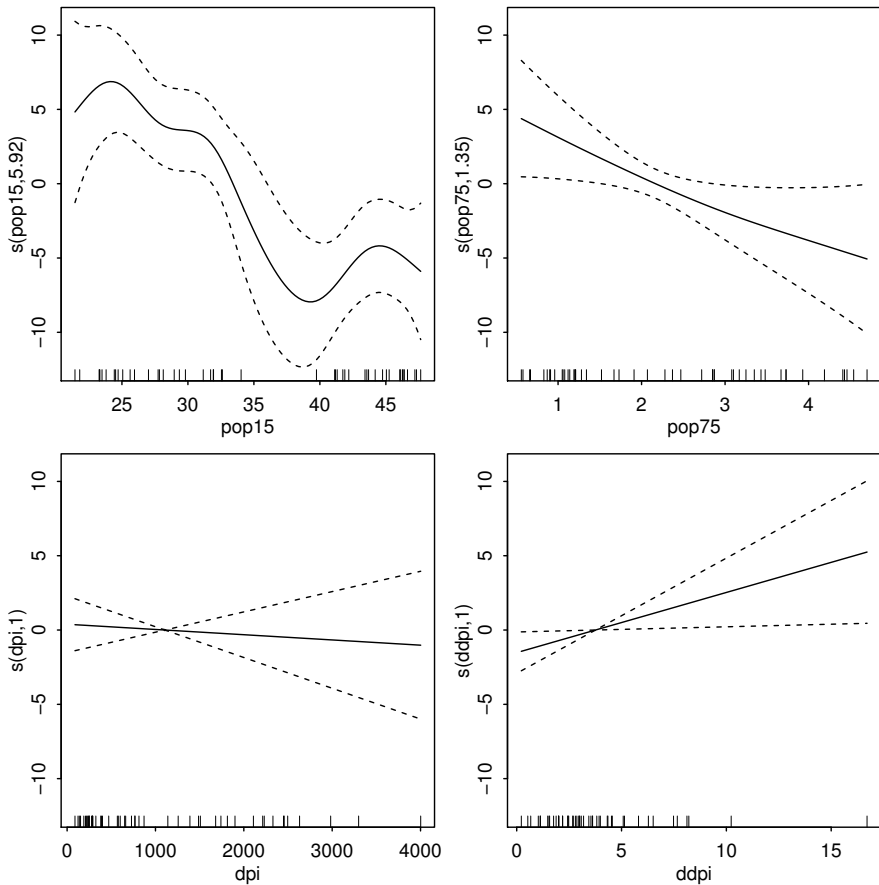


Figure 9.8 Fitted functions using the additive model for the savings data. The 95% confidence bands are shown as dotted lines.

- (b) Observations in successive years may be correlated. Fit a model that estimates this correlation. Does this change your opinion about the trend?
 - (c) Fit a polynomial model with degree 10 and use backward elimination to reduce the degree of the model. Plot your fitted model on top of the data. Use this model to predict the temperature in 2020.
 - (d) Suppose someone claims that the temperature was constant until 1930 and then began a linear trend. Fit a model corresponding to this claim. What does the fitted model say about this claim?
 - (e) Make a cubic spline fit with six basis functions evenly spaced on the range. Plot the fit in comparison to the previous fits. Does this model fit better than the straight-line model?
2. The `cornnit` data on the relationship between corn yield (bushels per acre) and

- nitrogen (pounds per acre) fertilizer application were studied in Wisconsin in 1994. Use transformations to find a good model for predicting yield from nitrogen. Use a goodness-of-fit test to check your model.
3. Using the `ozone` data, fit a model with `O3` as the response and `temp`, `humidity` and `ibh` as predictors. Use the Box–Cox method to determine the best transformation on the response.
 4. Use the `pressure` data to fit a model with `pressure` as the response and `temperature` as the predictor using transformations to obtain a good fit.
 5. Use transformations to find a good model for `volume` in terms of `girth` and `height` using the `trees` data.
 6. Use the `odor` data for this question.
 - (a) Fit a second order response surface for the `odor` response using the other three variables as predictors. How many parameters does this model use and how many degrees of freedom are left?
 - (b) Fit a model for the same response but now excluding any interaction terms but including linear and quadratic terms in all three predictors. Compare this model to the previous one. Is this simplification justified?
 - (c) Use the previous model to determine the values of the predictors which result in the minimum predicted odor.
 7. Use the `cheddar` data for this question.
 - (a) Fit an additive model for a response of `taste` with the other three variables as predictors. Is any transformation of the predictors suggested?
 - (b) Use the Box–Cox method to determine an optimal transformation of the response. Would it be reasonable to leave the response untransformed?
 - (c) Use the optimal transformation of the response and refit the additive model. Does this make any difference to the transformations suggested for the predictors?
 8. Use the `cars` data with `distance` as the response and `speed` as the predictor.
 - (a) Plot `distance` against `speed`.
 - (b) Show a linear fit to the data on the plot.
 - (c) Show a quadratic fit to the data on the plot.
 - (d) Now use `sqrt(dist)` as the response and fit a linear model. Show the fit on the same plot.
 - (e) Compute the default smoothing spline fit to the plot and display on a fresh plot of the data. How does it compare to the previous fits?

Model Selection

For all but the simplest cases we are confronted with a choice of possible regression models for our data. We may even have expanded the choice of possible models by introducing new variables derived from those available originally by making transformations, creating interactions or adding polynomial terms. In this chapter, we consider the problem of selecting the “best” subset of predictors.

More information should only be helpful so one might wonder why we do not simply include all the available variables in the model. However, we may wish to consider a smaller model for several reasons. The principle of Occam’s Razor states that among several plausible explanations for a phenomenon, the simplest is best. Applied to regression analysis, this implies that the smallest model that fits the data adequately is best.

Another consideration is that unnecessary predictors will add noise to the estimation of other quantities that interested us. Degrees of freedom will be wasted. More precise estimates and predictions might be achieved with a smaller model. In some cases, collecting data on additional variables can cost time or money so a smaller prediction model may be more economical.

Model selection is a process that should not be separated from the rest of the analysis. Other parts of the data analysis can have an impact. For example, outliers and influential points can do more than just change the current model — they can change the model we select. It is important to identify such points. Also transformations of the variables can have an impact on the model selected. Some iteration and experimentation are often necessary to find better models.

Although Occam’s Razor is a compelling heuristic, we must focus our effort on the main objective of regression modelling. We might obtain better predictions by using larger models so although smaller models might be appealing, we do not wish to compromise on predictive ability. For investigations that focus on the explanatory effect of the predictors, one should be cautious about the use of automated variable selection procedures. In such cases, attention is put on just a few predictors of interest while the remaining predictors are not of primary interest but must be controlled for. It would be unwise to expect an automated procedure to do this reliably.

When comparing potential models, we might use hypothesis testing methods to make a choice or use some criterion-based method on the relative fit to decide. We consider both these approaches in this chapter.

Even with a moderate number of potential predictors, the possible combinations of variables can become very large. Procedures that consider all possible combinations may not be practical and we must step through the space of possible models in

an incremental way. In some situations, the model space is structured hierarchically which constrains the reasonable choice of model.

10.1 Hierarchical Models

Some models have a natural hierarchy. For example, in polynomial models, x^2 is a higher order term than x . When selecting variables, it is important to respect the hierarchy. Lower order terms should not usually be removed from the model before higher order terms in the same variable. There are two common situations where this can arise:

Consider the polynomial model:

$$y = \beta_0 + \beta_1x + \beta_2x^2 + \varepsilon$$

Suppose we fit this model and find that the regression summary shows that the term in x is not significant but the term in x^2 is. If we then remove the x term, our reduced model would become:

$$y = \beta_0 + \beta_2x^2 + \varepsilon$$

However, suppose we make a scale change $x \rightarrow x + a$; then the model would become:

$$y = \beta_0 + \beta_2a^2 + 2\beta_2ax + \beta_2x^2 + \varepsilon$$

The first order x term has now reappeared. Scale changes should not make any important change to the model, but in this case an additional term has been added. This is not desirable. This illustrates why we should not remove lower order terms in the presence of higher order terms. We would not want interpretation to depend on the choice of scale. Removal of the first-order term here corresponds to the hypothesis that the predicted response is symmetric about and has an optimum at $x = 0$. Usually this hypothesis is not meaningful and should not be considered. Only when this hypothesis makes sense in the context of the particular problem could we justify the removal of the lower order term.

For models with interactions, consider the example of a second-order response surface model:

$$y = \beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_{11}x_1^2 + \beta_{22}x_2^2 + \beta_{12}x_1x_2 + \varepsilon$$

We would not normally consider removing the x_1x_2 interaction term without simultaneously considering the removal of the x_1^2 and x_2^2 terms. A joint removal would correspond to the clearly meaningful comparison of a quadratic surface and a linear one. Just removing the x_1x_2 term would correspond to a surface that is aligned with the coordinate axes. This is harder to interpret and should not be considered unless some particular meaning can be attached. Any rotation of the predictor space would reintroduce the interaction term and, as with the polynomials, we would not ordinarily want our model interpretation to depend on the particular basis for the predictors.

10.2 Testing-Based Procedures

Backward Elimination is the simplest of all variable selection procedures and can be easily implemented without special software. In situations where there is a complex hierarchy, backward elimination can be run manually while taking account of what variables are eligible for removal.

We start with all the predictors in the model and then remove the predictor with highest p -value greater than α_{crit} . Next refit the model and remove the remaining least significant predictor provided its p -value is greater than α_{crit} . Sooner or later, all “nonsignificant” predictors will be removed and the selection process will be complete.

The α_{crit} is sometimes called the “p-to-remove” and does not have to be 5%. If prediction performance is the goal, then a 15 to 20% cutoff may work best, although methods designed more directly for optimal prediction should be preferred.

Forward Selection just reverses the backward method. We start with no variables in the model and then for all predictors not in the model, we check their p -values if they are added to the model. We choose the one with lowest p -value less than α_{crit} . We continue until no new predictors can be added.

Stepwise Regression is a combination of backward elimination and forward selection. This addresses the situation where variables are added or removed early in the process and we want to change our mind about them later. At each stage a variable may be added or removed and there are several variations on exactly how this is done.

We illustrate backward elimination on some data on the 50 states from the 1970s. The data were collected from U.S. Bureau of the Census. We will take life expectancy as the response and the remaining variables as predictors:

```
> data(state)
> statedata <- data.frame(state.x77, row.names=state.abb)
> lmod <- lm(Life.Exp ~ ., statedata)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.09e+01	1.75e+00	40.59	< 2e-16
Population	5.18e-05	2.92e-05	1.77	0.083
Income	-2.18e-05	2.44e-04	-0.09	0.929
Illiteracy	3.38e-02	3.66e-01	0.09	0.927
Murder	-3.01e-01	4.66e-02	-6.46	8.7e-08
HS.Grad	4.89e-02	2.33e-02	2.10	0.042
Frost	-5.74e-03	3.14e-03	-1.82	0.075
Area	-7.38e-08	1.67e-06	-0.04	0.965

```
n = 50, p = 8, Residual SE = 0.745, R-Squared = 0.74
```

The signs of some of the coefficients match plausible expectations concerning how the predictors might affect the response. Higher murder rates decrease life expectancy as one might expect. Even so, some variables such as income, are not significant, contrary to what one might expect.

At each stage we remove the predictor with the largest p -value over 0.05. Area is the first to go:

```
> lmod <- update(lmod, . ~ . - Area)
```

```

> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.09e+01  1.75e+00  40.59 < 2e-16
Population   5.18e-05  2.92e-05   1.77  0.083
Income      -2.18e-05  2.44e-04  -0.09  0.929
Illiteracy   3.38e-02  3.66e-01   0.09  0.927
Murder      -3.01e-01  4.66e-02  -6.46  8.7e-08
HS.Grad      4.89e-02  2.33e-02   2.10  0.042
Frost       -5.74e-03  3.14e-03  -1.82  0.075
Area        -7.38e-08  1.67e-06  -0.04  0.965

n = 50, p = 8, Residual SE = 0.745, R-Squared = 0.74
> lmod <- update(lmod, . ~ . - Illiteracy)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.0657509  1.0289415  69.07 < 2e-16
Population   0.0000511  0.0000271   1.89  0.066
Income      -0.0000248  0.0002316  -0.11  0.915
Murder      -0.3000077  0.0370418  -8.10  2.9e-10
HS.Grad      0.0477580  0.0185908   2.57  0.014
Frost       -0.0059099  0.0024678  -2.39  0.021

n = 50, p = 6, Residual SE = 0.728, R-Squared = 0.74
> lmod <- update(lmod, . ~ . - Income)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.0271285  0.9528530  74.54 < 2e-16
Population   0.0000501  0.0000251   2.00  0.052
Murder      -0.3001488  0.0366095  -8.20  1.8e-10
HS.Grad      0.0465822  0.0148271   3.14  0.003
Frost       -0.0059433  0.0024209  -2.46  0.018

n = 50, p = 5, Residual SE = 0.720, R-Squared = 0.74
> lmod <- update(lmod, . ~ . - Population)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.03638  0.98326  72.25 <2e-16
Murder      -0.28307  0.03673  -7.71  8e-10
HS.Grad      0.04995  0.01520   3.29  0.002
Frost       -0.00691  0.00245  -2.82  0.007

n = 50, p = 4, Residual SE = 0.743, R-Squared = 0.71

```

The final removal of the Population variable is a close call. We may want to consider including this variable if interpretation is made easier. Notice that the R^2 for the full model of 0.736 is reduced only slightly to 0.713 in the final model. Thus the removal of four predictors causes only a minor reduction in fit.

It is important to understand that the variables omitted from the model may still be related to the response. For example:

```

> summary(lm(Life.Exp ~ Illiteracy+Murder+Frost, statedata))
      Estimate Std. Error t value Pr(>|t|)
(Intercept) 74.55672  0.58425  127.61 <2e-16
Illiteracy  -0.60176  0.29893  -2.01  0.0500
Murder      -0.28005  0.04339  -6.45  6e-08
Frost       -0.00869  0.00296  -2.94  0.0052

```

`n = 50, p = 4, Residual SE = 0.791, R-Squared = 0.67`

We see that illiteracy does have some association with life expectancy. It is true that replacing illiteracy with high school graduation rate gives us a somewhat better fitting model, but it would be insufficient to conclude that illiteracy is not a variable of interest. This demonstrates one failing of the method in that it cannot reliably distinguish between important and unimportant predictors.

Testing-based procedures are relatively cheap computationally and easy to understand, but they do have some drawbacks:

1. Because of the “one-at-a-time” nature of adding/dropping variables, it is possible to miss the “optimal” model.
2. The p -values used should not be treated too literally. There is so much multiple testing occurring that the validity is dubious. The removal of less significant predictors tends to increase the significance of the remaining predictors. This effect leads one to overstate the importance of the remaining predictors.
3. The procedures are not directly linked to final objectives of prediction or explanation and so may not really help solve the problem of interest. With any variable selection method, it is important to keep in mind that model selection cannot be divorced from the underlying purpose of the investigation. Variable selection tends to amplify the statistical significance of the variables that stay in the model. Variables that are dropped can still be correlated with the response. It would be wrong to say that these variables are unrelated to the response; it is just that they provide no additional explanatory effect beyond those variables already included in the model.
4. Stepwise variable selection tends to pick models that are smaller than desirable for prediction purposes. To give a simple example, consider the simple regression with just one predictor variable. Suppose that the slope for this predictor is not quite statistically significant. We might not have enough evidence to say that it is related to y but it still might be better to use it for predictive purposes.

Except in simple cases where only a few models are compared or in highly structured hierarchical models, testing-based variable selection should not be used. We include it here because the method is still used but should be discouraged.

10.3 Criterion-Based Procedures

If we have some idea about the purpose for which a model is intended, we might propose some measure of how well a given model meets that purpose. We could choose that model among those possible that optimize that criterion.

It would be natural to pick a model g , parameterized by θ , that is close to the true model f . We could measure the distance between g and f by

$$I(f, g) = \int f(x) \log \left(\frac{f(x)}{g(x|\theta)} \right) dx$$

Here we see that the best one predictor model uses Murder and so on. We compute and plot the AIC:

```
> AIC <- 50*log(rs$rss/50) + (2:8)*2
> plot(AIC ~ I(1:7), ylab="AIC", xlab="Number of Predictors")
```

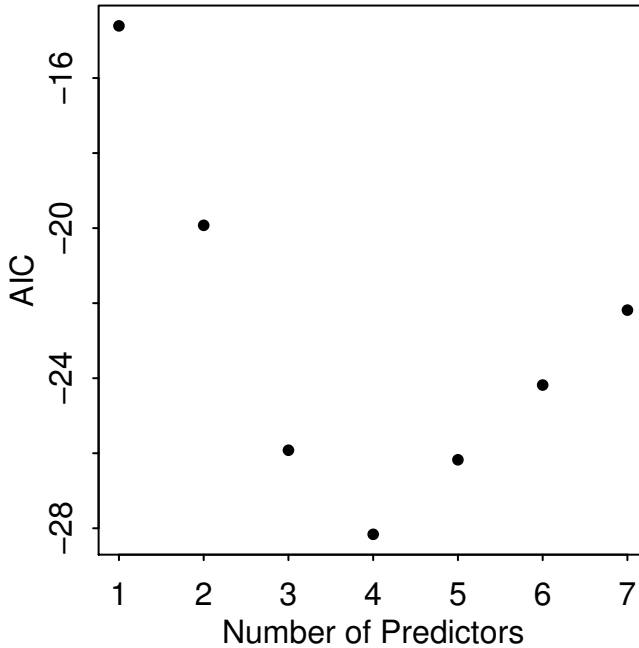


Figure 10.1 AIC for models with varying numbers of predictors using the state data.

We see in Figure 10.1 that the AIC is minimized by a choice of four predictors, namely population, murder, high school graduation and frost as determined by the logical matrix above.

Another commonly used criterion is adjusted R^2 , written R_a^2 . Recall that $R^2 = 1 - RSS/TSS$. Adding a variable to a model can only decrease the RSS and so only increase the R^2 . Hence R^2 by itself is not a good criterion, because it would always choose the largest possible model:

$$R_a^2 = 1 - \frac{RSS/(n-p)}{TSS/(n-1)} = 1 - \left(\frac{n-1}{n-p} \right) (1 - R^2) = 1 - \frac{\hat{\sigma}_{model}^2}{\hat{\sigma}_{null}^2}$$

Adding a predictor will only increase R_a^2 if it has some predictive value. We can see this because minimizing the standard error for prediction means minimizing $\hat{\sigma}^2$ which in turn means maximizing R_a^2 .

Now let's see which model the adjusted R^2 criterion selects using the plot shown in the first panel of Figure 10.2:


```
> plot(2:8,rs$adjr2,xlab="No. of Parameters",ylab="Adjusted R-square")
> which.max(rs$adjr2)
[1] 4
```

We see that the population, frost, high school graduation and murder model has the largest R_a^2 .

Our final criterion is Mallows's C_p statistic. A good model should predict well, so the average mean square error of prediction might be a good criterion:

$$\frac{1}{\sigma^2} \sum_i E(\hat{y}_i - E y_i)^2$$

which can be estimated by the C_p statistic:

$$C_p = \frac{RSS_p}{\hat{\sigma}^2} + 2p - n$$

where $\hat{\sigma}^2$ is from the model with all predictors and RSS_p indicates the RSS from a model with p parameters. For the full model $C_p = p$ exactly. If a p predictor model fits, then $E(RSS_p) = (n-p)\sigma^2$ and then $E(C_p) \approx p$. A model with a bad fit will have C_p much bigger than p . It is usual to plot C_p against p . We desire models with small p and C_p around or less than p . C_p , R_a^2 and AIC all trade-off fit in terms of RSS against complexity (p).

The C_p plot can be constructed as:

```
> plot(2:8,rs$cp,xlab="No. of Parameters",ylab="Cp Statistic")
> abline(0,1)
```

as seen in the second panel of Figure 10.2. The competition is between the four-parameter, three-predictor, model including frost, high school graduation and murder and the model also including population. Both models are on or below the $C_p = p$ line, indicating good fits. The choice is between the smaller model and the larger model, which fits a little better. Some even larger models fit in the sense that they are on or below the $C_p = p$ line, but we would not opt for these in the presence of smaller models that fit.

If there are q potential predictors, then there are 2^q possible models. For larger q , this might be too time consuming and we may need to economize by limiting the search. In such cases, the `step()` function is a cheaper alternative. The function does not evaluate the AIC for all possible models but uses a search method that compares models sequentially. Thus it bears some comparison to the stepwise method described above, but only in the method of search — there is no hypothesis testing.

```
> lmod <- lm(Life.Exp ~ ., data=statedata)
> step(lmod)
Start: AIC= -22.18
Life.Exp ~ Population + Income + Illiteracy + Murder +
HS.Grad + Frost + Area
```

	Df	Sum of Sq	RSS	AIC
- Area	1	0.0011	23.3	-24.2
- Income	1	0.0044	23.3	-24.2

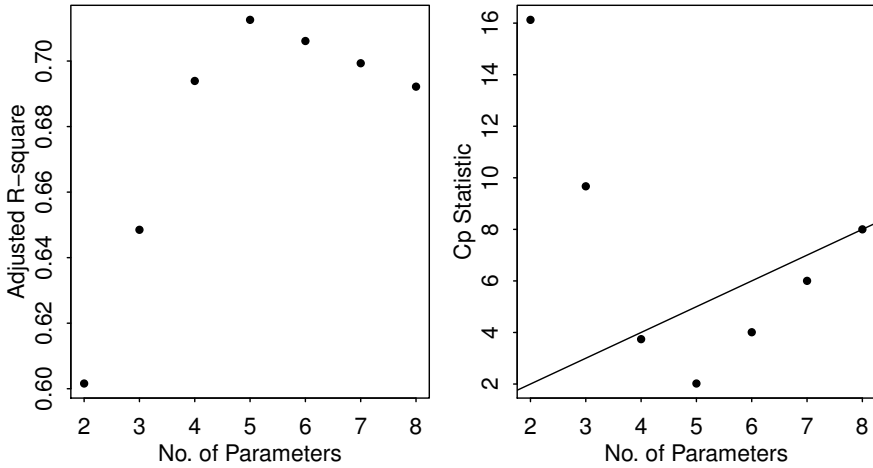


Figure 10.2 The adjusted R^2 for the state data models is on the left and the C_p plot for the same data at the right.

```

- Illiteracy 1 0.0047 23.3 -24.2
<none> 23.3 -22.2
- Population 1 1.7 25.0 -20.6
- Frost 1 1.8 25.1 -20.4
- HS.Grad 1 2.4 25.7 -19.2
- Murder 1 23.1 46.4 10.3

Step: AIC= -24.18
Life.Exp ~ Population + Income + Illiteracy + Murder +
HS.Grad + Frost

.. intermediate steps omitted ..

Step: AIC= -28.16
Life.Exp ~ Population + Murder + HS.Grad + Frost

Df Sum of Sq RSS AIC
<none> 23.3 -28.2
- Population 1 2.1 25.4 -25.9
- Frost 1 3.1 26.4 -23.9
- HS.Grad 1 5.1 28.4 -20.2
- Murder 1 34.8 58.1 15.5

Coefficients:
(Intercept) Population Murder HS.Grad Frost
71.0271285 0.0000501 -0.3001488 0.0465822 -0.0059433
    
```

The sequence of variable removal is the same as with backward elimination and the model selected is the same as for AIC.

Variable selection methods are sensitive to outliers and influential points. Let's check for high leverage points:

```
> h <- lm.influence(lmod) $hat
> names(h) <- state.abb
> rev(sort(h))
      AK      CA      HI      NV      NM      TX      NY
0.809522 0.408857 0.378762 0.365246 0.324722 0.284164 0.256950
```

We can see that Alaska has high leverage. Let's try excluding it:

```
> b<-regsubsets(Life.Exp~.,data=statedata, subset=(state.abb!="AK"))
> rs <- summary(b)
> rs$which[which.max(rs$adjr),]
(Intercept) Population      Income Illiteracy      Murder
      TRUE      TRUE      FALSE      FALSE      TRUE
      HS.Grad      Frost      Area
      TRUE      TRUE      TRUE
```

We see that Area now makes it into the model. Transforming the predictors can also have an effect. Take a look at the variables:

```
> stripchart(data.frame(scale(statedata)), method="jitter", las=2,
vertical=TRUE)
```

Jittering adds a small amount of noise (in the horizontal direction in this example). It is useful for moving apart points that would otherwise overprint each other.

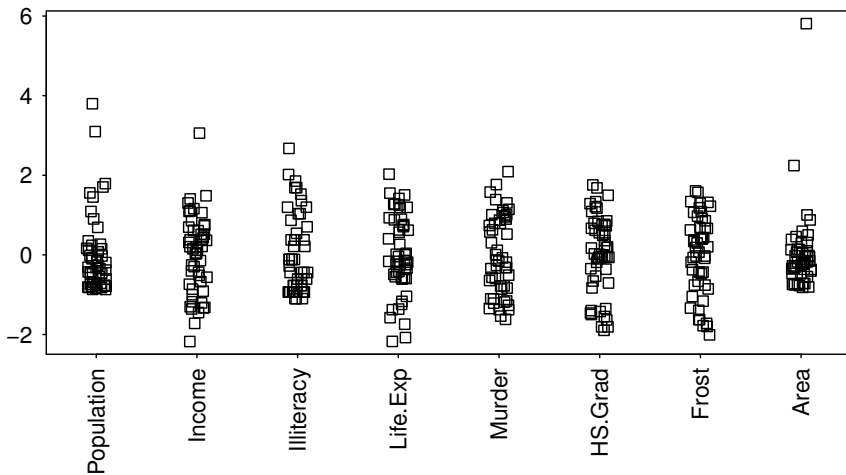


Figure 10.3 Strip charts of the state data; all variables have been standardized.

In Figure 10.3, we see that population and area are skewed — we try transforming them:

```
> b<-regsubsets(Life.Exp ~ log(Population)+Income+Illiteracy+ Murder+
HS.Grad+Frost+log(Area), statedata)
> rs <- summary(b)
> rs$which[which.max(rs$adjr),]
(Intercept) log(Population)      Income      Illiteracy
      TRUE      TRUE      FALSE      FALSE
```

Murder	HS.Grad	Frost	log(Area)
TRUE	TRUE	TRUE	FALSE

This changes the “best” model again to $\log(\text{population})$, frost, high school graduation and murder. The adjusted R^2 of 71.7% is the highest among models we have seen so far.

10.4 Summary

Variable selection is a means to an end and not an end itself. The aim is to construct a model that predicts well or explains the relationships in the data. Automatic variable selections are not guaranteed to be consistent with these goals. Use these methods as a guide only.

Hypothesis testing-based methods use a restricted search through the space of potential models and use a dubious method for choosing between models when repeated many times. Criterion-based methods typically involve a wider search and compare models in a preferable manner. For this reason, we recommend that you use a criterion-based method.

Accept the possibility that several models may be suggested which fit about as well as each other. If this happens, consider:

1. Do the models have similar qualitative consequences?
2. Do they make similar predictions?
3. What is the cost of measuring the predictors?
4. Which has the best diagnostics?

If you find models that seem roughly comparable, but lead to quite different conclusions, then it is clear that the data cannot answer the question of interest unambiguously. Be alert to the possibility that a model contradictory to the tentative conclusions might be out there.

Exercises

1. Use the `prostate` data with `lpsa` as the response and the other variables as predictors. Implement the following variable selection methods to determine the “best” model:
 - (a) Backward elimination
 - (b) AIC
 - (c) Adjusted R^2
 - (d) Mallows C_p
2. Using the `teengamb` dataset with `gamble` as the response and the other variables as predictors, repeat the work of the first question.
3. Using the `divusa` dataset with `divorce` as the response and the other variables as predictors, repeat the work of the first question.

4. Using the `trees` data, fit a model with $\log(\text{Volume})$ as the response and a second-order polynomial (including the interaction term) in `Girth` and `Height`. Determine whether the model may be reasonably simplified.
5. Fit a linear model to the `stackloss` data with `stack.loss` as the predictor and the other variables as predictors. Simplify the model if possible. Check the model for outliers and influential points. Now return to the full model, determine whether there are any outliers or influential points, eliminate them and then repeat the variable selection procedures.
6. Use the `seatos` data with `hipcenter` as the response.
 - (a) Fit a model with all eight predictors. Comment on the effect of leg length on the response.
 - (b) Compute a 95% prediction interval for the mean value of the predictors.
 - (c) Use AIC to select a model. Now interpret the effect of leg length and compute the prediction interval. Compare the conclusions from the two models.
7. Generate some simulated data as described in Section 9.5.
 - (a) Fit regression splines with 12 evenly-spaced knots using $y \sim \text{bs}(x, 12)$. You will need to load the `splines` package. Display the fit on top of the data.
 - (b) Compute the AIC for this model.
 - (c) Compute the AIC for all models with a number of knots between 3 and 20 inclusive. Plot the AIC as function of the number of parameters. Which model is best?
 - (d) Plot the fit for your selected model on top of the data.
8. Use the `odor` data with `odor` as the response.
 - (a) Fit a second-order response surface model which contains all quadratic and cross-product terms of the three predictors. Explicitly include all nine terms in your model statement. You will need to `â€œprotectâ€œ` these terms using the `I()` function. Show the regression summary.
 - (b) Use the backward elimination method with a cut-off of 5% to select a smaller model.
 - (c) Use the `step` method to select a model using AIC. Using this selected model determine the optimal values of the predictors to minimize odor.
 - (d) Use the `polym` function to fit a second-order orthogonal polynomial regression model to the response using the three predictors. Confirm that the fit is identical to (a).
 - (e) Apply the `step` procedure to this last model. What terms are considered for elimination? What model was selected? Compare the outcome with (c).

Shrinkage Methods

Sometimes it can seem like you have too many predictors. If you use them all in your regression model, problems can arise. Explanation can be difficult due to collinearity as we saw in Section 7.3. Prediction performance can also be degraded by using too many predictors. But more predictors should mean more information. That should be helpful. In this chapter, we will look at four methods that allow us to *shrink* this additional information into a more useful form. You will see later why we have used the word “shrink”.

11.1 Principal Components

Principal components analysis (PCA) is a popular method for finding low-dimensional linear structure in higher dimensional data. It has a range of purposes but let’s see how it can be helpful in regression problems.

Designed experiments typically have predictors which are mutually orthogonal. This makes the fitting and interpretation of these models simpler. For observational data, predictors are often substantially correlated. It would be nice if we could transform these predictors to orthogonality as it could make interpretation easier.

For example, consider the dimensions of the human body as measured in a study on 252 men as described in Johnson (1996) as seen in Figure 11.1. We analyzed this data in Section 4.2 where more details about motivation behind the data collection can be found.

```
> data(fat, package="faraway")
> plot(neck ~ knee, fat)
> plot(chest ~ thigh, fat)
> plot(hip ~ wrist, fat)
```

We see that the body circumferences are strongly correlated. Although we may have many predictors, there may be less information than the number of predictors might suggest. PCA aims to discover this lower dimension of variability in higher dimensional data.

Suppose we center the matrix of predictors X by subtracting the mean for each variable so that the columns of X have mean zero. We use an X that does not include a column of ones for an intercept term. We now find the orthogonal directions of greatest variation in the data:

1. Find the u_1 such that $\text{var}(u_1^T X)$ is maximized subject to $u_1^T u_1 = 1$.
2. Find u_2 such that $\text{var}(u_2^T X)$ is maximized subject to $u_1^T u_2 = 0$ and $u_2^T u_2 = 1$.

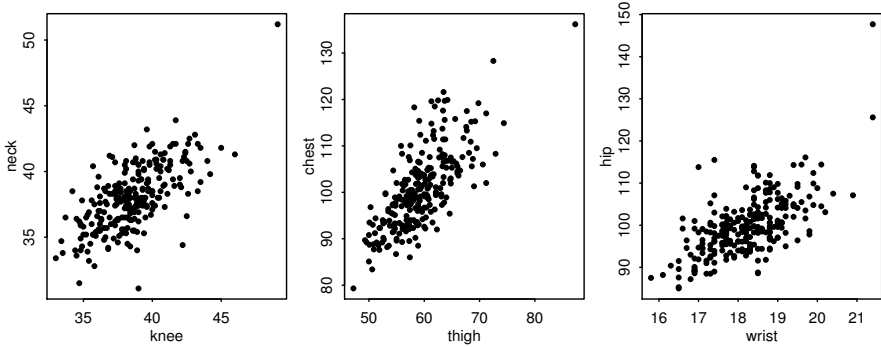


Figure 11.1 Circumferences in centimeters for body dimensions of 252 men.

- Keep finding directions of greatest variation orthogonal to those directions we have already found.

For low dimensions, the process can be iterated until all dimensions have been exhausted. For high dimensional problems, we can stop when the remaining variation is negligible.

Let's write $z_i = u_i^T X$. The z_i are known as the principal components. We can gather together the terms in the matrix form $Z = XU$ where Z and U have columns z_i and u_i respectively. U is called the rotation matrix. We can think of Z as a version of the data rotated to orthogonality.

We consider only the circumference measurements in the `fat` dataset and compute the principal components decomposition:

```
> cfat <- fat[,9:18]
> prfat <- prcomp(cfat)
> dim(prfat$rot)
[1] 10 10
> dim(prfat$x)
[1] 252 10
```

The `$rot` component is the rotation matrix U while the principal components are found in the `$x` component which we have called Z . We can learn how effective the PCA has been from the summary output:

```
> summary(prfat)
Importance of components:
      PC1      PC2      PC3      PC4      PC5      PC6
Standard deviation  15.990  4.06584  2.96596  2.00044  1.69408  1.49881
Proportion of Variance  0.867  0.05605  0.02983  0.01357  0.00973  0.00762
Cumulative Proportion  0.867  0.92304  0.95287  0.96644  0.97617  0.98378
      PC7      PC8      PC9     PC10
Standard deviation  1.30322  1.25478  1.10955  0.52737
Proportion of Variance  0.00576  0.00534  0.00417  0.00094
Cumulative Proportion  0.98954  0.99488  0.99906  1.00000
```

The summary gives the standard deviations of the principal components $u_i^T X$ for $i = 1, \dots, 10$. We also see that the first principal component explains 86.7% of the

variation in the data while the last few components account for very little of the variation. Instead of ten variables, we could use just a single variable, formed by a linear combination described by the first PC, which would represent the ten-dimensional data quite well.

The first column of the rotation matrix, u_1 , is a linear combination describing the first principal component:

```
> round(prfat$rot[, 1], 2)
      neck chest abdom hip thigh knee ankle biceps
0.12  0.50  0.66  0.42  0.28  0.12  0.06  0.15
forearm wrist
0.07  0.04
```

We see that the chest, abdomen, hip and thigh measures dominate the first principal component. However, the reason for this may simply be that these measures are larger and so more variable than the wrist or ankle circumferences. We might prefer to scale the variables for size by converting to standard units, that is, subtracting the mean and dividing by the standard deviation. We can achieve this as follows:

```
> prfatc <- prcomp(cfat, scale=TRUE)
> summary(prfatc)
Importance of components:
              PC1      PC2      PC3      PC4      PC5      PC6
Standard deviation  2.6498  0.85301  0.81909  0.70114  0.54708  0.52831
Proportion of Variance  0.7021  0.07276  0.06709  0.04916  0.02993  0.02791
Cumulative Proportion  0.7021  0.77490  0.84199  0.89115  0.92108  0.94899
              PC7      PC8      PC9      PC10
Standard deviation  0.45196  0.40539  0.27827  0.2530
Proportion of Variance  0.02043  0.01643  0.00774  0.0064
Cumulative Proportion  0.96942  0.98586  0.99360  1.0000
> round(prfatc$rot[, 1], 2)
      neck chest abdom hip thigh knee ankle biceps
0.33  0.34  0.33  0.35  0.33  0.33  0.25  0.32
forearm wrist
0.27  0.30
```

We can see that, after scaling, the proportion of variability explained by the first component drops to 70.2%. The remaining variation is more evenly spread over the other components. The first principal component has very similar coefficients for all the variables. It is useful to interpret this as “overall size” since it is roughly proportional to a mean across all these (standardized) variables. One interpretation of this dominant first principal component is that body shapes in men are mostly proportional. Bigger men tend to be just larger all-round versions of smaller men.

The other principal components describe how the data vary in ways orthogonal to this first PC. For example, we might look at the second principal component:

```
> round(prfatc$rot[, 2], 2)
      neck chest abdom hip thigh knee ankle biceps
0.00  -0.27  -0.40  -0.25  -0.19  0.02  0.62  0.02
forearm wrist
0.36  0.38
```

which is roughly a contrast between the body center measures of chest, abdomen, hip and thigh circumferences against the extremities of forearm, wrist and ankle

measures. This could be viewed as a relative measure of where the body is carrying its weight. This represents only 7.3% of the total variation so it is not substantial but is the largest component of variation after the first PC.

Like variances, principal components analysis can be very sensitive to outliers so it is essential to check for these. In addition to the usual graphical checks of the data, it is worth checking for outliers in higher dimensions. Such outliers can be hard to find. For example, consider someone with a weight of 50 kg and a height of 2 m. Neither value is individually exceptional but the combination would be very unlikely and possibly an error. Mahalanobis distance is a measure of the distance of a point from the mean that adjusts for the correlation in the data. It is defined as

$$d_i = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

where μ is a measure of center and Σ is a measure of covariance. Since we are concerned about outliers, it is sensible to use robust measures of center and covariance — these are provided by the `cov.rob()` function from the MASS package.

```
> require(MASS)
> robfat <- cov.rob(cfat)
> md <- mahalanobis(cfat, center=robfat$center, cov=robfat$cov)
> n <- nrow(cfat); p <- ncol(cfat)
> plot(qchisq(1:n/(n+1), p), sort(md), xlab=expression(paste(chi^2, "
  quantiles")), ylab="Sorted Mahalanobis distances")
> abline(0,1)
```

The `mahalanobis` function returns d_i^2 . If the data are multivariate normal with dimension m , then we expect d^2 to follow a χ_m^2 distribution. We can check this graphically with a Q–Q plot as seen in Figure 11.2. We see that there are some outliers and that we can investigate the sensitivity of the PCA to these values by re-analyzing the data after removing these points. If you do this, you will find that it makes a substantive difference, especially to the second PC. An alternative to this outlier detection approach is to use *robust PCA* methods.

So far we have not tried to link the predictors to the response in a regression model. Indeed, PCA is widely used in other applications — see Jolliffe (2002) for a book length treatment. Now let's see how we can use PCA for regression. We might have a model $\bar{y} \sim X$. We replace this by $\bar{y} \sim Z$ where typically we use only the first few columns of Z . This is known as *principal components regression* or PCR. The technique is used in two distinct ways for explanation and prediction purposes.

When the goal of the regression is to find simple, well-fitting and understandable models for the response, PCR *may* help. The PCs are linear combinations of the predictors. For example, the purpose in collecting the body fat data was to model the percentage of body fat described by the response, `brozek`. We analyzed this data before in Section 4.2 but we are using fewer predictors here. Here is the model output where we use all ten circumference predictors:

```
> lmoda <- lm(fat$brozek ~ ., data=cfat)
> summary(lmoda)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.22875     6.21431    1.16  0.24588
neck         -0.58195     0.20858   -2.79  0.00569
```

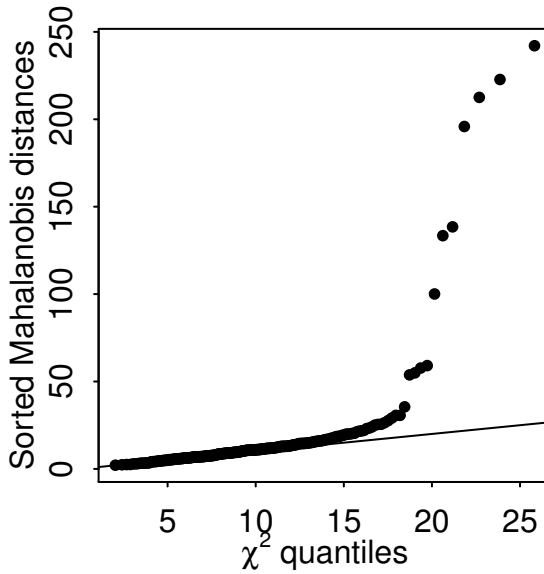


Figure 11.2 Q-Q plot of the Mahalanobis distances for the body circumference data.

```

chest      -0.09085    0.08543    -1.06    0.28866
abdom      0.96023    0.07158    13.41    < 2e-16
hip        -0.39135    0.11269    -3.47    0.00061
thigh      0.13371    0.12492     1.07    0.28554
knee       -0.09406    0.21239    -0.44    0.65828
ankle      0.00422    0.20318     0.02    0.98344
biceps     0.11120    0.15912     0.70    0.48533
forearm    0.34454    0.18551     1.86    0.06450
wrist      -1.35347    0.47141    -2.87    0.00445

n = 252, p = 11, Residual SE = 4.071, R-Squared = 0.74
    
```

It is difficult to say much about which factors might influence body fat percentage because there are clear indications of collinearity. The signs of the coefficients and their significance vary in a less than credible way. Why would abdomen circumference have a positive effect while hip circumference has negative effect? Now consider the output where we use only the first two principal components:

```

> lmodpccr <- lm(fat$brozek ~ prfatc$x[,1:2])
> sumary(lmodpccr)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    18.938     0.329   57.54 <2e-16
prfatc$x[, 1:2]PC1    1.842     0.124   14.80 <2e-16
prfatc$x[, 1:2]PC2   -3.551     0.387   -9.18 <2e-16

n = 252, p = 3, Residual SE = 5.225, R-Squared = 0.55
    
```

We have lost some explanatory power in going from ten down to two predictors.

But these two predictors are now orthogonal, meaning we can now interpret these without collinearity worries. As previously discussed, the first PC can be viewed as a measure of overall size. We can see this is associated with higher body fat. The second PC shows a negative association, meaning that men who carry relatively more of their substance in their extremities tend to be leaner. These would tend to be men who are more muscular so this result accords with what one might expect. So the PCR here has allowed us a meaningful explanation whereas the full predictor regression was opaque.

One objection to the previous analysis is that the two PCs still use all ten predictors so there has been no saving in the number of variables needed to model the response. Furthermore, we must rely on our subjective interpretation of the meaning of the PCs. To answer this, one idea is to take a few representative predictors based on the largest coefficients seen in the PCs. For example, we might pick out the abdomen circumference to represent the first PC and the difference between abdomen and ankle circumference for the second PC. The latter pair show the largest coefficients in absolute value (-0.40 and 0.62) for the second PC. Abdomen does not quite have the largest coefficient in the first PC (hip does) but it's a close choice and means we need only two predictors in the following model. We need to scale the predictors so that ankle and abdomen are on the standard units scale.

```
> lmodr <- lm(fat$brozek ~ scale(abdom) + I(scale(ankle)-scale(abdom))
, data=cfat)
> summary(lmodr)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	18.938	0.279	67.79	<2e-16
scale(abdom)	5.763	0.328	17.55	<2e-16
I(scale(ankle) - scale(abdom))	-0.995	0.314	-3.17	0.0017

```
n = 252, p = 3, Residual SE = 4.435, R-Squared = 0.68
```

We have a simple model that fits almost as well as the ten-predictor model. We can interpret it similarly to the previous model but it is easier to explain to others. Future studies might be done more cheaply because we might only need these two measures.

However, for this explanatory use of PCR to work, we typically need the predictors to measure quantities for which linear combinations are interpretable — usually the predictors would need to have the same units. So if we had used the age and weight variables found in the `fat` data example, it would have been far more difficult to interpret the linear combinations. Even in the homogenous predictor case, we need some luck and imagination to get interpretable PCs. These requirements restrict the utility of PCR for explanatory purposes. It is worth trying but you might not get anything useful from it.

We can sometimes make better predictions with a small number of principal components in Z than a much larger number of variables in X . Success requires that we make a good choice of the number of components. To illustrate the use of PCR and the other shrinkage methods in this chapter, we will use a set of data where the emphasis is on prediction but the explanatory aspects of the methods can be useful in gaining intuition about the structure of the data. A Tecator Infratec Food and Feed Analyzer working in the wavelength range of 850 to 1050 nm by the near-infrared

transmission (NIT) principle was used to collect data on samples of finely chopped pure meat and 215 samples were measured. For each sample, the fat content was measured along with a 100-channel spectrum of absorbances. Since determining the fat content via analytical chemistry is time consuming, we would like to build a model to predict the fat content of new samples using the 100 absorbances which can be measured more easily. See Thodberg (1993) for more details.

The true performance of any model is hard to determine based on just the fit to the available data. We need to see how well the model does on new data not used in the construction of the model. For this reason, we will partition the data into two parts — a *training sample* consisting of the first 172 observations that we will use to build and estimate the models and a *testing sample* of the remaining 43 observations.

Let's start with the least squares fit:

```
> data(meatspec, package="faraway")
> trainmeat <- meatspec[1:172,]
> testmeat <- meatspec[173:215,]
> modlm <- lm(fat ~ ., trainmeat)
> summary(modlm)$r.squared
[1] 0.99702
```

We see that the fit of this model is already very good in terms of R^2 . How well does this model do in predicting the observations in the test sample? We need a measure of performance — we use root mean square error (RMSE):

$$\sqrt{\left(\sum_{i=1}^n (\hat{y}_i - y_i)^2 / n\right)}$$

where $n = 43$ in this instance. We find for the training sample that:

```
> rmse <- function(x, y) sqrt(mean((x-y)^2))
> rmse(fitted(modlm), trainmeat$fat)
[1] 0.69032
```

while for the test sample:

```
> rmse(predict(modlm, testmeat), testmeat$fat)
[1] 3.814
```

We see that the performance is much worse for the test sample. This is not unusual, as the fit to the data we have almost always gives an over-optimistic sense of how well the model will do with future data. In this case, the actual error is about five times greater than the model itself suggests.

Now, it is quite likely that not all 100 predictors are necessary to make a good prediction. In fact, some of them might just be adding noise to the prediction and we could improve matters by eliminating some of them. We use the default stepwise model selection:

```
> modsteplm <- step(modlm)
> rmse(modsteplm$fit, trainmeat$fat)
[1] 0.7095
> rmse(predict(modsteplm, testmeat), testmeat$fat)
[1] 3.5902
```

The model selection step removed 28 variables. Of course, the nominal fit got a little worse as it always will when predictors are removed, but the actual performance improved somewhat from 3.81 to 3.59.

Now let's compute the PCA on the training sample predictors:

```
> meatpca <- prcomp(trainmeat[, -101])
```

We can examine the standard deviations of the principal components:

```
> round(meatpca$sdev, 3)
 [1] 5.055 0.511 0.282 0.168 0.038 0.025 0.014 0.011 0.005 0.003
....
 [91] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

We see that the first PC accounts for about ten times more variation than the second. The contribution drops off sharply. This suggests that most of the variation in the predictors can be explained with just a few dimensions.

The linear combinations u_i (or *loadings*) can be found in the rotation matrix `meatpca$rotation`. We plot these vectors against the predictor number (which represents the range of frequencies in this case) in Figure 11.3 using:

```
> matplot(1:100, meatpca$rot[, 1:3], type="l", xlab="Frequency", ylab="
=", col=1)
```

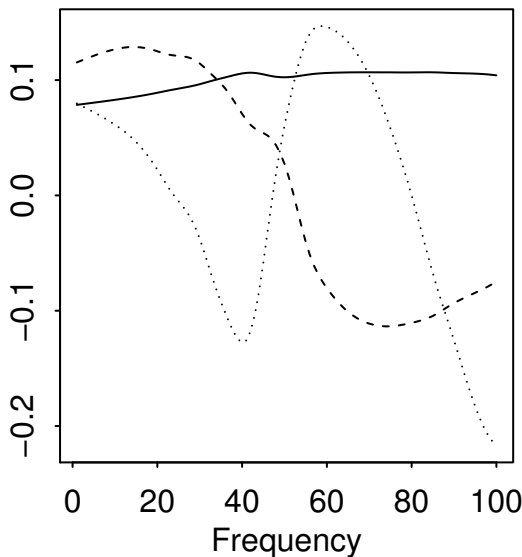


Figure 11.3 Eigenvectors for the PCA of the meat spectrometer data. The solid line corresponds to the first PC, the dotted line is for the second PC and the dashed line is for the third PC.

These vectors represent the linear combinations of the predictors that generate the PCs. We see that the first PC comes from an almost constant combination of the

frequencies. It measures whether the predictors are generally large or small. The second PC represents a contrast between the higher and lower frequencies. The third is more difficult to interpret. It is sometimes possible, as in this example, to give some meaning to the PCs. This is typically a matter of intuitive interpretation. Sometimes, no interpretation can be found and we must be satisfied with possibility of improved prediction.

We can do the PCR with output from `prcomp()` but the `pls` package contains several features which makes the computation more straightforward. We start by fitting a PCR considering up to 50 components:

```
> require(pls)
> pcrmod <- pcr(fat ~ ., data=trainmeat, ncomp=50)
```

Let's use the first four PCs to predict the response:

```
> rmse(predict(pcrmod, ncomp=4), trainmeat$fat)
[1] 4.0647
```

We do not expect as good a fit using only four variables instead of the 100. Even so, considering that, the fit is not much worse than the results from much bigger models.

PCR is an example of *shrinkage* estimation. Let's see where the name comes from. We plot the 100 slope coefficients for the full least squares fit:

```
> plot(modlm$coef[-1], xlab="Frequency", ylab="Coefficient", type="l")
```

which is shown in the left panel of Figure 11.4. We see that the coefficients' range is in the thousands and that the adjacent coefficients can be very different. This is perhaps surprising because one might expect that adjacent frequencies might have a very similar effect on the response. We plot the coefficients from the four-component fit in the panel on the right of Figure 11.4.

```
> coefplot(pcrmod, ncomp=4, xlab="Frequency", main="")
```

Here we see that the range of these coefficients is much smaller than the thousands seen for the ordinary least squares fit. Instead of wildly varying coefficients in the least squares case, we have a more stable result. This is why the effect is known as shrinkage. Furthermore, there is smoothness between adjacent frequencies as knowledge of the science behind this data might suggest.

Why use four PCs here and not some other number? The standard advice for choosing the number of PCs to represent the variation in X is to choose the number beyond which all the PC standard deviations are relatively small. A good way to determine this number is to make a *scree plot*, which simply makes an index plot of the standard deviations. We show here the square roots of the eigenvalues and only the first ten values to focus on the area of interest; see Figure 11.5:

```
> plot(meatpca$sdev[1:10], type="l", ylab="SD of PC", xlab="PC number")
```

We could make a case for using only the first PC, but there is another identifiable "elbow" at five indicating the choice of four PCs. Now let's see how well the test sample is predicted:

```
> rmse(predict(pcrmod, testmeat, ncomp=4), testmeat$fat)
[1] 4.534
```

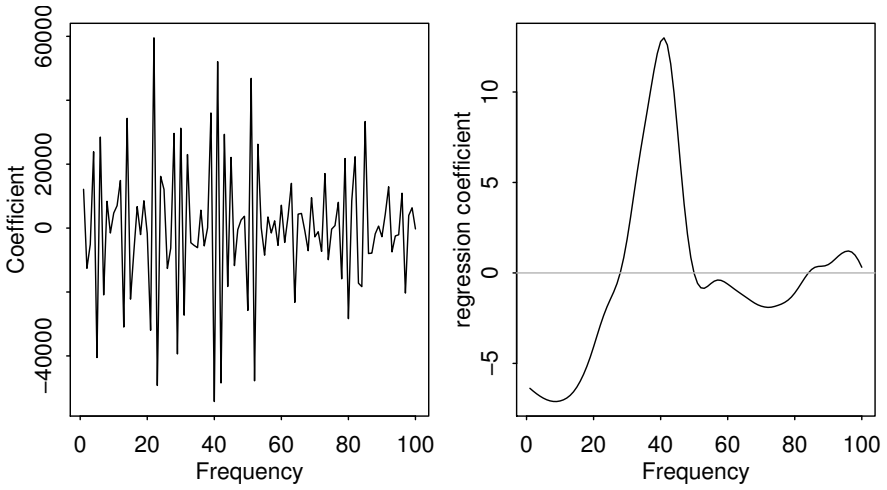


Figure 11.4 Coefficients for the least squares fit on the left and for the PCR with four components on the right.

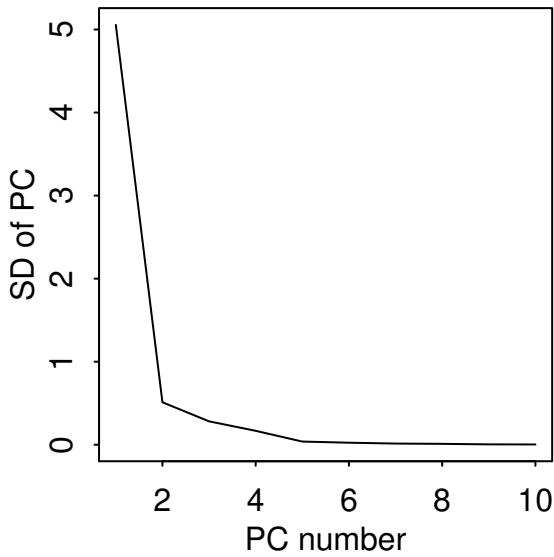


Figure 11.5 Scree plot of the standard deviations of the first ten principal components.

which is not at all impressive. It turns out that we can do better by using more PCs — we figure out how many would give the best result on the test sample:

```
> pcrmse <- RMSEP(pcrmod, newdata=testmeat)
> plot(pcrmse, main="")
> which.min(pcrmse$val)
[1] 28
> pcrmse$val[28]
[1] 1.8549
```

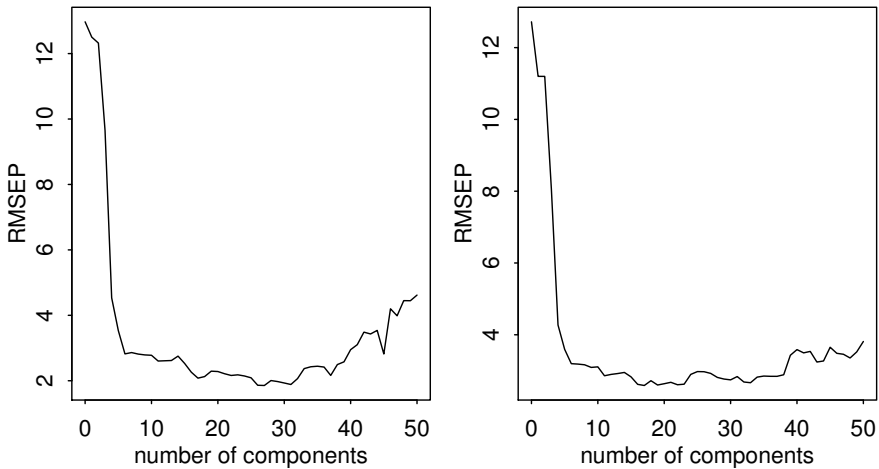


Figure 11.6 RMS for the test sample on the left and RMS estimated using CV for varying numbers of PCs on the right.

The plot of the RMSE is seen in Figure 11.6. The best result occurs for 28 PCs for which the RMSE is far better than anything achieved thus far. Of course, in practice we would not have access to the test sample in advance and so we would not know to use 28 components. We could, of course, reserve part of our original dataset for testing. This is sometimes called a *validation sample*. This is a reasonable strategy, but the downside is that we lose this sample from our estimation which degrades its quality. Furthermore, there is the question of which and how many observations should go into the validation sample. We can avoid this dilemma with the use of *crossvalidation* (CV). We divide the data into m parts, equal or close in size. For each part, we use the rest of the data as the training set and that part as the test set. We evaluate the criterion of interest, RMSE in this case. We repeat for each part and average the result.

The `pcr()` function can compute this CV. By default, the data is randomly divided into ten parts for the CV. The random division means that the outcome is not deterministic. We have set the random number generator seed using `set.seed()` (and the choice of 123 is arbitrary) so that you will get the same result as here. You do not need to set the seed unless you want the results to be completely reproducible.

```
> set.seed(123)
```



```

> pcrmod <- pcr(fat ~ ., data=trainmeat, validation="CV", ncomp=50)
> pcrCV <- RMSEP(pcrmod, estimate="CV")
> plot(pcrCV, main="")
> which.min(pcrCV$val)
[1] 18
> ypred <- predict(pcrmod, testmeat, ncomp=18)
> rmse(ypred, testmeat$fat)
[1] 2.1271

```

The crossvalidated estimates of the RMSE are shown in the right panel of Figure 11.6. The minimum occurs at 18 components. This gives an RMSE on the test sample, 2.13, which is close to the optimum.

11.2 Partial Least Squares

Partial least squares (PLS) is a method for relating a set of input variables X_1, \dots, X_m and outputs Y_1, \dots, Y_l . PLS regression is comparable to PCR in that both predict the response using some number of linear combinations of the predictors. The difference is that while PCR ignores Y in determining the linear combinations, PLS regression explicitly chooses them to predict Y as well as possible.

We will consider only univariate PLS, i.e., $l = 1$ so that Y is scalar. We will look for models of the form:

$$\hat{y} = \beta_1 T_1 + \dots + \beta_k T_k$$

where the T_i 's are mutually orthogonal linear combinations of the X s. See Figure 11.7

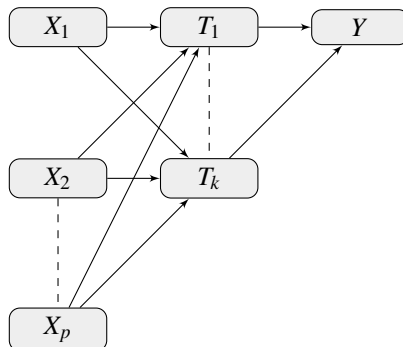


Figure 11.7 Schematic representation of partial least squares.

Various algorithms have been presented for computing PLS. Most work by iteratively determining the T_i s to predict y well, but at the same time maintaining orthogonality. One criticism of PLS is that it solves no well-defined modeling problem, which makes it difficult to distinguish between the competing algorithms on theoretical rather than empirical grounds. Garthwaite (1994) presents an intuitive algorithm, but de Jong (1993) describes the SIMPLS method, which is among the best known. Several other algorithms exist.

As with PCR, we must choose the number of components carefully. CV can be

helpful in doing this. We apply PLS to the meat spectroscopy data using CV to select the number of components. We compute the PLS on all models up to size 50. We have set the random generator seed to ensure reproducibility in light of the random division of the validation samples. We plot the linear combination used for a four-component model in the left panel of Figure 11.8. The crossvalidated estimates of the RMSE are shown in the right panel.

```
> set.seed(123)
> plsmod <- plsrfat ~ ., data=meatspec[1:172,], ncomp=50, validation="CV")
> coefplot(plsmod, ncomp=4, xlab="Frequency")
> plsCV <- RMSEP(plsmod, estimate="CV")
> plot(plsCV, main="")
```

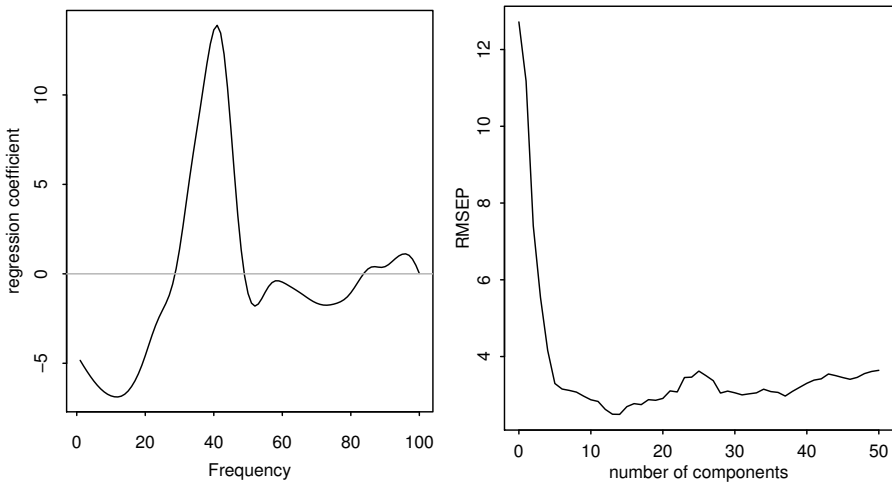


Figure 11.8 Coefficients of X for a four-component PLS model are shown on the left. Estimated CV error is shown on the right.

We see that the effective linear combination of the predictors in the four-component model is similar to the PCR indicating the shrinkage effect. As before, four components do not appear to be enough — we need around 15 components as suggested by the crossvalidated estimate of the RMSE. We need fewer components than PCR, which is expected since we are using information about the response.

Now we determine the performance on the training set for the 15-component model:

```
> ypred <- predict(plsmod, ncomp=15)
> rmse(ypred, trainmeat$fat)
[1] 1.8898
```

which is similar to PCR, but now see how we do on the test set:

```
> ytpred <- predict(plsmod, testmeat, ncomp=15)
> rmse(ytpred, testmeat$fat)
[1] 1.9718
```

which is slightly better than the 2.13 achieved by PCR.

We have not checked any diagnostics in this analysis. PLS and PCR are just as sensitive to assumptions as OLS so these are still mandatory in any full analysis.

PCR and PLS are particularly attractive methods when there are large numbers of predictors, p , relative to the sample size, n . They can still work even when $p > n$. This is not uncommon in some applications such as bioinformatics where data may give information on a large number of genes for a small sample of individuals.

PLS tends to have an advantage over PCR for prediction problems because PLS constructs its linear combination explicitly to predict the response. On the other hand, PCR is better suited for developing insights by forming linear combinations that have interesting interpretations.

Although both methods use the idea of dimension reduction, there is usually no reduction in the number of variables used since every predictor contributes to the linear combinations. If you are hoping to save the cost of measuring predictors in order to build a cheaper but still effective prediction model, the criterion-based variable selection as described in Chapter 10 or the lasso method described later in this chapter will be more useful.

11.3 Ridge Regression

Ridge regression makes the assumption that the regression coefficients (after normalization) should not be very large. This is a reasonable assumption in applications where you have a large number of predictors and you believe that many of them have some effect on the response. Hence shrinkage is embedded in the method. Ridge regression is particularly effective when the model matrix is collinear and the usual least squares estimates of β appear to be unstable.

Suppose that the predictors have been centered by their means and scaled by their standard deviations and that the response has been centered. The ridge regression estimate chooses the β that minimizes:

$$(y - X\beta)^T (y - X\beta) + \lambda \sum_j \beta_j^2$$

for some choice of $\lambda \geq 0$. The *penalty term* is $\sum_j \beta_j^2$. We want to keep this term small. Ridge regression is an example of a *penalized regression* because of the presence of this term. The ridge regression estimates of β s are given by:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

The λI introduces a “ridge” down the $X^T X$ matrix, hence the name of the method. An equivalent expression of the problem is that we choose β to minimize:

$$(y - X\beta)^T (y - X\beta) \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 \leq t^2$$

We find the least squares solution subject to an upper bound on the size of the coefficients. The use of ridge regression can also be justified from a Bayesian perspective where a prior distribution on the parameters puts more weight on smaller values.

The nature of the ridge solution is seen in the first panel of Figure 11.9. The OLS fit that minimizes the residual sum of squares (RSS) is achieved at $\hat{\beta}_{LS}$. We can draw confidence ellipses around the least squares solution $\hat{\beta}_{LS}$ of increasing size. Points on these ellipses represent solutions for β which are progressively less desirable in terms of fit. We grow the ellipses until they intersect with a circle of radius t centered at the origin. This point of intersection will satisfy $\sum_{j=1}^p \beta_j^2 = t^2$ and be the best fitting solution subject to this requirement. Of course, if $\hat{\beta}_{LS}$ is already inside the circle, the ridge and OLS solutions are identical but usually t is small enough that this rarely occurs.

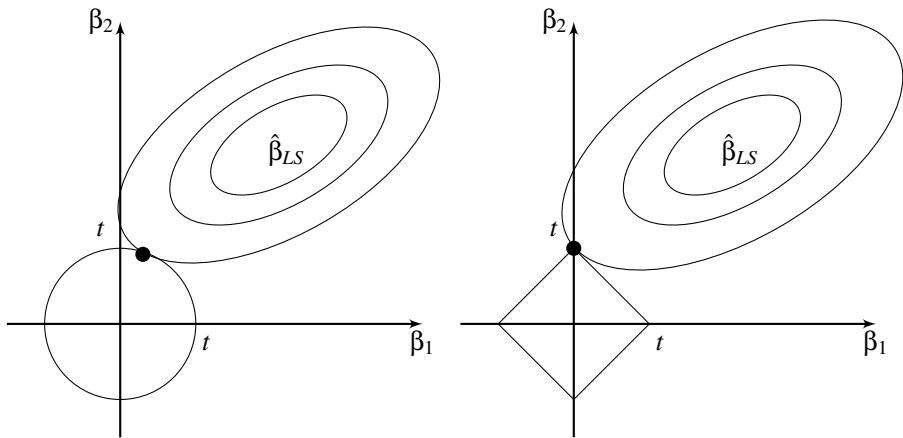


Figure 11.9 Ridge and lasso regression are illustrated. On the left, confidence ellipses of increasing level are plotted around the least squares estimate. The largest ellipse intersects the circle of radius t at the ridge estimate. On the right, the largest ellipse intersects the square at the lasso estimate.

λ (or t) may be chosen by automatic methods, but it is also sensible to plot the values of $\hat{\beta}$ as a function of λ . You should pick the smallest value of λ that produces stable estimates of β .

We demonstrate the method on the meat spectroscopy data; $\lambda = 0$ corresponds to least squares while we find that as $\lambda \rightarrow \infty$: $\hat{\beta} \rightarrow 0$. The response, training and test datasets need to be mean-centered.

```
> require(MASS)
> rgmod <- lm.ridge(fat ~ ., trainmeat, lambda = seq(0, 5e-8, len=21))
> matplot(rgmod$lambda, coef(rgmod), type="l", xlab=expression(lambda),
, ylab=expression(hat(beta)), col=1)
```

Some experimentation was necessary to determine the appropriate range of λ . The *ridge trace plot* is shown in Figure 11.10. We can select the value of λ using generalized crossvalidation (GCV) which is similar to CV but is easier to compute.

```
> which.min(rgmod$GCV)
1.75e-08
```

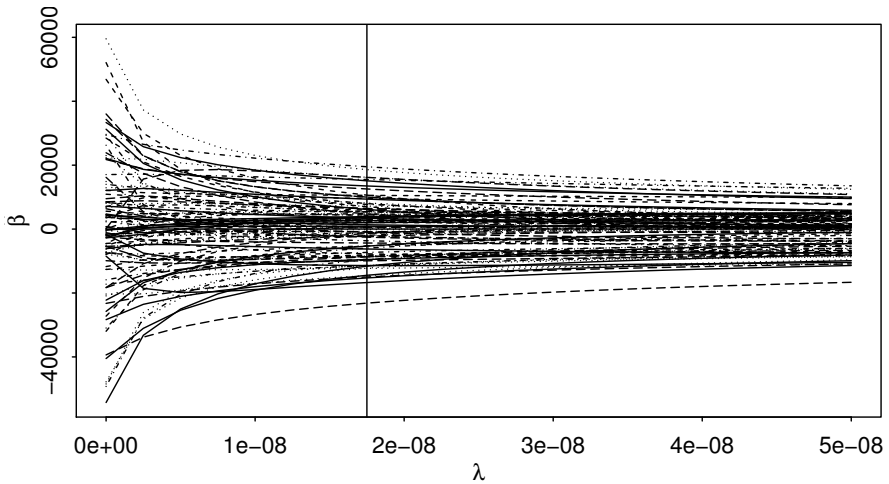


Figure 11.10 Ridge trace plot for the meat spectroscopy data. The generalized crossvalidation choice of λ is shown as a vertical line.

```
8
> abline(v=1.75e-08)
```

We will use the generalized crossvalidation (GCV) estimate of $1.75e^{-8}$. First, we compute the training sample performance. This ridge regression both centers and scales the predictors, so we need to do the same in computing the fit. Furthermore, we need to add back in the mean of the response because of the centering:

```
> ypred <- cbind(1,as.matrix(trainmeat[,-101])) %*% coef(rgmod)[8,]
> rmse(ypred, trainmeat$fat)
[1] 0.80244
```

which is comparable to the above, but for the test sample we find:

```
> ypred <- cbind(1,as.matrix(testmeat[,-101])) %*% coef(rgmod)[8,]
> rmse(ypred, testmeat$fat)
[1] 4.1011
```

which is dismayingly poor. However, a closer examination of the predictions reveals that just one of the ridge predictions is bad:

```
> c(ypred[13], ypred[13], testmeat$fat[13])
[1] 35.690 11.164 34.800
```

The PLS prediction (first) is close to the truth (third), but the ridge prediction is bad. If we remove this case:

```
> rmse(ypred[-13], testmeat$fat[-13])
[1] 1.9795
```

we get a good result. Of course, one poor prediction can ruin your reputation so PLS did better here in coping with the outlier.

Ridge regression estimates of coefficients are biased. Bias is undesirable, but it is not the only consideration. The mean-squared error (MSE) can be decomposed in the following way:

$$E(\hat{\beta} - \beta)^2 = (E(\hat{\beta} - \beta))^2 + E(\hat{\beta} - E\hat{\beta})^2$$

Thus the MSE of an estimate can be represented as the square of the bias plus the variance. Sometimes a large reduction in the variance may be obtained at the price of an increase in the bias. If the MSE is substantially reduced as a consequence, then we may be willing to accept some bias. This is the trade-off that ridge regression makes — a reduction in variance at the price of an increase in bias. This is a common dilemma.

Frank and Friedman (1993) compared PCR, PLS and ridge regression and found the best results for ridge regression. Of course, for any given dataset any of the methods may prove to be the best, so picking a winner is difficult.

11.4 Lasso

The lasso method is apparently very similar to the ridge regression method. We choose $\hat{\beta}$ to minimize:

$$(y - X\beta)^T(y - X\beta) \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| \leq t$$

where the penalty term differs in form from the ridge case. This method was introduced by Tibshirani (1996). There is no explicit solution to this problem as in the ridge regression case although it can be found quite efficiently as described within the context of a more general method called *least angle regression* as given in Efron, Hastie, Johnstone, and Tibshirani (2004). The name, lasso, is not an acronym. It refers to the rope used by cowboys to pull cattle from the herd just as the statistician selects variables from the full model.

The important difference between lasso and ridge regression is in the nature of the solutions which is illustrated in Figure 11.9. For the lasso, the L_1 constraint of $\sum_{j=1}^p |\beta_j| \leq t$ defines a square in two dimensions as seen in the figure. In higher dimensions, it defines a polytope with vertices on the coordinates' axes and edges where some number of the coordinate values will be zero. In the figure, we can see that the expanding ellipses touch the square at a vertex. We can see that, depending on the value of $\hat{\beta}_{LS}$, the correlation of variables which determines the orientation of the ellipse and the value of t , that a solution on a vertex will occur frequently. In higher dimensions, the solution will also commonly lie on an edge or vertex of the polytope. As t increases, more variables are added to the model and their coefficients become larger. For large enough t , the restriction of $\sum_{j=1}^p |\beta_j| \leq t$ is redundant and the least squares solution is returned.

For the lasso, for moderate values of t , many $\hat{\beta}_j$ tend to be zero. The use of lasso is most appropriate when we believe the effects are *sparse* — that the response can be explained by a small number of predictors with the rest having no effect. This

means that lasso can be regarded as a type of variable selection method because when $\hat{\beta}_j = 0$, the corresponding predictor x_j is effectively eliminated from the regression. In contrast, ridge regression does not eliminate any variables; it only makes the $\hat{\beta}_j$ smaller.

The `lars` package can be used to compute lasso solutions. We demonstrate its application to the state data from the variable selection chapter:

```
> require(lars)
> data(state)
> statedata <- data.frame(state.x77, row.names=state.abb)
> lmod <- lars(as.matrix(statedata[, -4]), statedata$Life)
```

The `lars` function requires the matrix of predictors as its first argument and does not accept a formula argument. As before, we use the other variables to predict the life expectancy. The results can be plotted, as seen in the first panel of Figure 11.11:

```
> plot(lmod)
```

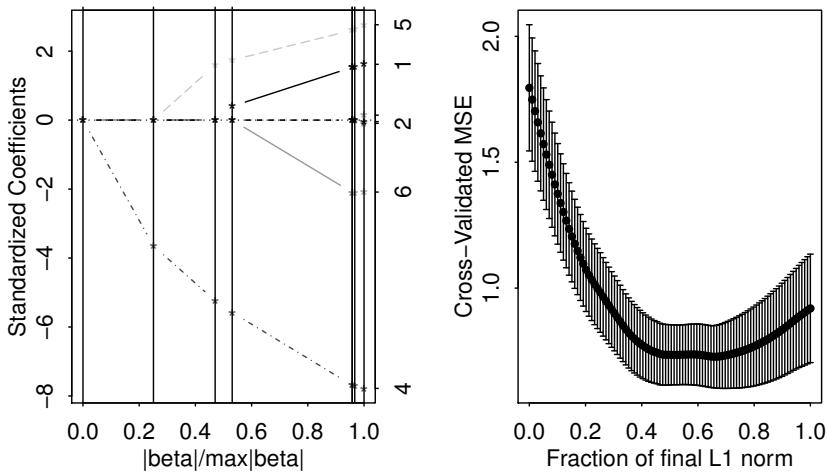


Figure 11.11 Lasso plots for the state data. The coefficients as a function of t are shown in the first panel. The crossvalidated selection of t is shown in the second panel. t has been scaled by the least squares solution.

The horizontal axis for t has been scaled by the maximum possible value for $\sum_{j=1}^p |\beta_j|$ given by the least squares solution. For the smallest value of t , only predictor #4, the murder rate, is active. As t increases, a second predictor, the high school graduation rate enters. The population and days of frost enter later. The remaining three variables do not enter the model until t is very close to the least squares solution. As t increases, we see the size of the estimated coefficients also increase.

The choice of t can be made using crossvalidation. The default method for `lars` uses a 10-fold method where the data are randomly divided into 10 groups. Nine groups are used to predict the one group for each group in turn and the overall pre-

diction performance is computed. The CV criterion along with a standard error estimated can be computed. As in previous examples where the 10-fold division is randomly selected, we set the random number generator seed to ensure reproducibility.

```
> set.seed(123)
> cvlmod <- cv.lars(as.matrix(statedata[,-4]), statedata$Life)
```

The plot is shown in the second panel of Figure 11.11. We see that a (fractional) value for t of around 0.65 is recommended. Referring to the first panel of Figure 11.11, we see this leads to a model containing four predictors. The minimizing fraction and corresponding coefficients may be obtained as:

```
> cvlmod$index[which.min(cvlmod$cv)]
[1] 0.65657
> predict(lmod, s=0.65657, type="coef", mode="fraction")$coef
  Population      Income  Illiteracy      Murder    HS.Grad
0.000023453  0.000000000  0.000000000  -0.239879419  0.035288870
      Frost      Area
-0.001694991  0.000000000
```

We see the same variables are selected as in the previous analyses. However, the coefficients are shrunk somewhat from the corresponding least squares solution:

```
> coef(lm(Life.Exp ~ Population+Murder+HS.Grad+Frost, statedata))
(Intercept) Population      Murder    HS.Grad      Frost
71.02712853  0.00005014  -0.30014880  0.04658225  -0.00594329
```

Lasso is most useful for problems with much larger numbers of variables such as the ongoing spectroscopy example of this chapter. We compute the lasso fit for the training data:

```
> trainy <- trainmeat$fat
> trainx <- as.matrix(trainmeat[,-101])
> lassomod <- lars(trainx, trainy)
```

We now compute the crossvalidation choice of t :

```
> set.seed(123)
> cvout <- cv.lars(trainx, trainy)
> cvout$index[which.min(cvout$cv)]
[1] 0.010101
```

For this choice of t , we compute the predicted values for the test data:

```
> testx <- as.matrix(testmeat[,-101])
> predlars <- predict(lassomod, testx, s=0.0101, mode="fraction")
```

The RMSE may now be computed:

```
> rmse(testmeat$fat, predlars$fit)
[1] 2.1323
```

We obtain the best result of all the methods tried so far.

It is interesting to compute the coefficients of the selected fit and examine them, as seen in Figure 11.12:


```

> predlars <- predict(lassomod, s=0.0101, type="coef", mode="fraction
")
> plot(predlars$coef, type="h", ylab="Coefficient")
> sum(predlars$coef != 0)
[1] 20

```

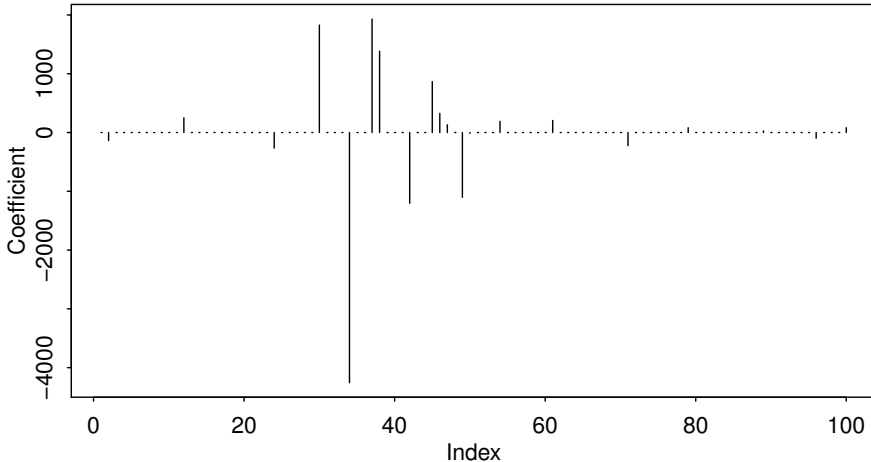


Figure 11.12 Coefficients of the CV-selected lasso fit to the spectroscopy data.

We see that only 20 frequencies are used in predicting the response. The competing methods in this chapter used every frequency except for the stepwise method which still used most of the frequencies.

For this example, there is probably no additional cost in measuring all the frequencies as opposed to just a few of them. However, in other applications, where there is some cost in recording additional predictors, the lasso method would be especially valuable. The lasso is particularly useful when we believe the effects are *sparse*, i.e., that few of the predictors have an effect. For example, this arises in the analysis of gene expression data where we have good reason to believe that only a small number of genes have an influence on the quantity of interest. Lasso is also useful in that it still works when the number of predictors p exceeds the number of observations. The use of lasso in applications where effects are not sparse, as in many socioeconomic examples, is less compelling.

Exercises

- Using the `seatpos` data, perform a PCR analysis with `hipcenter` as the response and `HtShoes`, `Ht`, `Seated`, `Arm`, `Thigh` and `Leg` as predictors. Select an appropriate number of components and give an interpretation to those you choose. Add `Age` and `Weight` as predictors and repeat the analysis. Use both models to predict the response for predictors taking these values:

Age	Weight	HtShoes	Ht	Seated
-----	--------	---------	----	--------

64.800	263.700	181.080	178.560	91.440
Arm	Thigh	Leg		
35.640	40.950	38.790		

2. Fit a PLS model to the `seatpos` data with `hipcenter` as the response and all other variables as predictors. Take care to select an appropriate number of components. Use the model to predict the response at the values of the predictors specified in the first question.
3. Fit a ridge regression model to the `seatpos` data with `hipcenter` as the response and all other variables as predictors. Take care to select an appropriate amount of shrinkage. Use the model to predict the response at the values of the predictors specified in the first question.
4. Take the `fat` data, and use the percentage of body fat, `siri`, as the response and the other variables, except `brozek` and `density` as potential predictors. Remove every tenth observation from the data for use as a test sample. Use the remaining data as a training sample building the following models:
 - (a) Linear regression with all predictors
 - (b) Linear regression with variables selected using AIC
 - (c) Principal component regression
 - (d) Partial least squares
 - (e) Ridge regression

Use the models you find to predict the response in the test sample. Make a report on the performances of the models.
5. Some near infrared spectra on 60 samples of gasoline and corresponding octane numbers can be found by `data(gasoline, package="pls")`. Compute the mean value for each frequency and predict the response for the best model using the five different methods from Question 4.
6. The dataset `kanga` contains data on the skulls of historical kangaroo specimens.
 - (a) Compute a PCA on the 18 skull measurements. You will need to exclude observations with missing values. What percentage of variation is explained by the first principal component?
 - (b) Provide the loadings for the first principal component. What variables are prominent?
 - (c) Repeat the PCA but with the variables all scaled to the same standard deviation. How do the percentage of variation explained and the first principal component differ from those found in the previous PCA?
 - (d) Give an interpretation of the second principal component.
 - (e) Compute the Mahalanobis distances and plot appropriately to check for outliers.
 - (f) Make a scatterplot of the first and second principal components using a different plotting symbol depending on the sex of the specimen. Do you think these two components would be effective in determining the sex of a skull?

Insurance Redlining — A Complete Example

In this chapter, we present a relatively complete data analysis. The example is interesting because it illustrates several of the ambiguities and difficulties encountered in statistical practice.

Insurance redlining refers to the practice of refusing to issue insurance to certain types of people or within some geographic area. The name comes from the act of drawing a red line around an area on a map. Now few would quibble with an insurance company refusing to sell auto insurance to a frequent drunk driver, but other forms of discrimination would be unacceptable.

In the late 1970s, the US Commission on Civil Rights examined charges by several Chicago community organizations that insurance companies were redlining their neighborhoods. Because comprehensive information about individuals being refused homeowners insurance was not available, the number of FAIR plan policies written and renewed in Chicago by zip code for the months of December 1977 through May 1978 was recorded. The FAIR plan was offered by the city of Chicago as a default policy to homeowners who had been rejected by the voluntary market. Information on other variables that might affect insurance writing such as fire and theft rates was also collected at the zip code level. The variables are:

race racial composition in percentage of minority
fire fires per 100 housing units
theft thefts per 1000 population
age percentage of housing units built before 1939
involact new FAIR plan policies and renewals per 100 housing units
income median family income in thousands of dollars
side north or south side of Chicago

The data come from Andrews and Herzberg (1985) where more details of the variables and the background are provided.

12.1 Ecological Correlation

Notice that we do not know the races of those denied insurance. We only know the racial composition in the corresponding zip code. This is an important difficulty that needs to be considered before starting the analysis.

When data are collected at the group level, we may observe a correlation between two variables. The ecological fallacy is concluding that the same correlation holds

at the individual level. For example, in countries with higher fat intakes in the diet, higher rates of breast cancer have been observed. Does this imply that individuals with high fat intakes are at a higher risk of breast cancer? Not necessarily. Relationships seen in observational data are subject to confounding, but even if this is allowed for, bias is caused by aggregating data. We consider an example taken from US demographic data:

```
> data(eco, package="faraway")
> plot(income ~ usborn, data=eco, xlab="Proportion US born", ylab="
  Mean Annual Income")
```

In the first panel of Figure 12.1, we see the relationship between 1998 per capita income dollars from all sources and the proportion of legal state residents born in the United States in 1990 for each of the 50 states plus the District of Columbia (D.C.). We can see a clear negative correlation. We can fit a regression line and show the

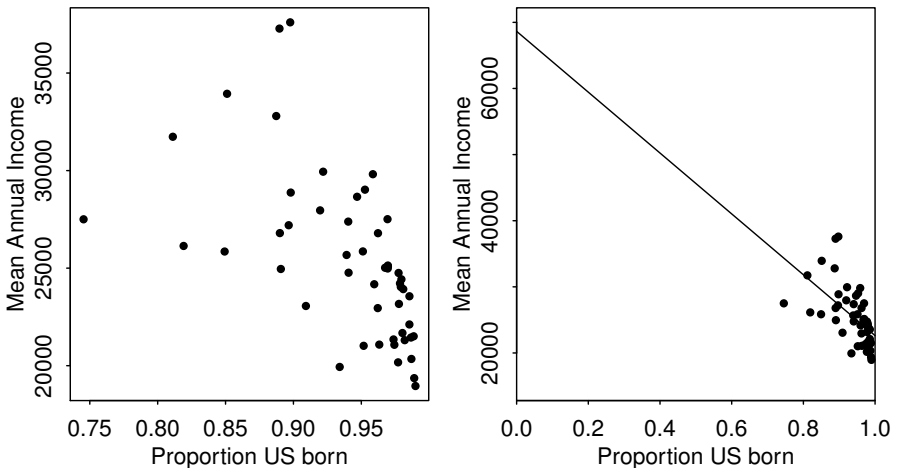


Figure 12.1 1998 annual per capita income and proportion US born for 50 states plus D.C. The plot on the right shows the same data as on the left, but with an extended scale and the least squares fit shown.

fitted line on an extended range:

```
> lmod <- lm(income ~ usborn, eco)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   68642      8739    7.85  3.2e-10
usborn       -46019      9279   -4.96  8.9e-06

n = 51, p = 2, Residual SE = 3489.541, R-Squared = 0.33
> plot(income ~ usborn, data=eco, xlab="Proportion US born", ylab="
  Mean Annual Income", xlim=c(0,1), ylim=c(15000,70000), xaxs="i")
> abline(coef(lmod))
```

We see that there is a clear statistically significant relationship between the per capita

annual income and the proportion who are US born. What does this say about the average annual income of people who are US born and those who are naturalized citizens? If we substitute `usborn=1` into the regression equation, we get $68642 - 46019 = \$22,623$, while if we put `usborn=0`, we get $\$68,642$. This suggests that on average, naturalized citizens earn three times more than US born citizens. In truth, information from the US Bureau of the Census indicates that US born citizens have an average income just slightly larger than naturalized citizens. What went wrong with our analysis?

The ecological inference from the aggregate data to the individuals requires an assumption of constancy. Explicitly, the assumption would be that the incomes of the native born do not depend on the proportion of native born within the state (and similarly for naturalized citizens). This assumption is unreasonable for these data because immigrants are naturally attracted to wealthier states.

This assumption is also relevant to the analysis of the Chicago insurance data since we have only aggregate data. We must keep in mind that the results for the aggregated data may not hold true at the individual level.

12.2 Initial Data Analysis

Start by reading the data in and examining it:

```
> data(chredlin, package="faraway")
> head(chredlin)
  race fire theft age involact income side
60626 10.0  6.2   29 60.4     0.0 11.744  n
60640 22.2  9.5   44 76.5     0.1  9.323  n
60613 19.6 10.5   36 73.5     1.2  9.948  n
60657 17.3  7.7   37 66.9     0.5 10.656  n
60614 24.5  8.6   53 81.4     0.7  9.730  n
60610 54.0 34.1   68 52.6     0.3  8.231  n
```

Summarize:

```
> summary(chredlin)
  race          fire          theft          age
Min.   : 1.00    Min.   : 2.00    Min.   : 3.0    Min.   : 2.0
1st Qu.: 3.75    1st Qu.: 5.65    1st Qu.: 22.0   1st Qu.:48.6
Median :24.50    Median :10.40    Median : 29.0   Median :65.0
Mean   :34.99    Mean   :12.28    Mean   : 32.4   Mean   :60.3
3rd Qu.:57.65    3rd Qu.:16.05    3rd Qu.: 38.0   3rd Qu.:77.3
Max.   :99.70    Max.   :39.70    Max.   :147.0   Max.   :90.1

  involact      income      side
Min.   :0.000    Min.   : 5.58    n:25
1st Qu.:0.000    1st Qu.: 8.45    s:22
Median :0.400    Median :10.69
Mean   :0.615    Mean   :10.70
3rd Qu.:0.900    3rd Qu.:11.99
Max.   :2.200    Max.   :21.48
```

We see that there is a wide range in the `race` variable, with some zip codes almost entirely minority or non-minority. This is good for our analysis since it will reduce the variation in the regression coefficient for `race`, allowing us to assess this effect

more accurately. If all the zip codes were homogeneous, we would never be able to discover an effect from these aggregated data. We also note some skewness in the theft and income variables. The response `involact` has a large number of zeros. This is not good for the assumptions of the linear model but we have little choice but to proceed. We will not use the information about north versus south side until later. Now make some graphical summaries:

```
> require(ggplot2)
> ggplot(chredlin, aes(race, involact)) + geom_point() + stat_smooth(
  method="lm")
> ggplot(chredlin, aes(fire, involact)) + geom_point() + stat_smooth(
  method="lm")
> ggplot(chredlin, aes(theft, involact)) + geom_point() + stat_smooth(
  method="lm")
> ggplot(chredlin, aes(age, involact)) + geom_point() + stat_smooth(
  method="lm")
> ggplot(chredlin, aes(income, involact)) + geom_point() + stat_smooth(
  method="lm")
> ggplot(chredlin, aes(side, involact)) + geom_point(position = position
  _jitter(width = .2, height=0))
```

The plots are seen in Figure 12.2. We have superimposed a linear fit to each pair of variables with a 95% confidence band shown in grey. Strong relationships can be seen in several of the plots. We can see some outlier and influential points. We can also see that the fitted line sometimes goes below zero which is problematic since observed values of the response cannot be negative. Jittering has been added in the final plot to avoid overplotting of symbols. Let's focus on the relationship between `involact` and `race`:

```
> summary(lm(involact ~ race, chredlin))
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.12922    0.09661    1.34   0.19
race         0.01388    0.00203    6.84  1.8e-08
n = 47, p = 2, Residual SE = 0.449, R-Squared = 0.51
```

We can clearly see that homeowners in zip codes with a high percentage of minorities are taking the default FAIR plan insurance at a higher rate than other zip codes. That is not in doubt. However, can the insurance companies claim that the discrepancy is due to greater risks in some zip codes? The insurance companies could claim that they were denying insurance in neighborhoods where they had sustained large fire-related losses and any discriminatory effect was a by-product of legitimate business practice. We plot some of the variables involved by this question in Figure 12.3.

```
> ggplot(chredlin, aes(race, fire)) + geom_point() + stat_smooth(method="
  lm")
> ggplot(chredlin, aes(race, theft)) + geom_point() + stat_smooth(method
  ="lm")
```

We can see that there is indeed a relationship between the fire rate and the percentage of minorities. We also see that there is large outlier that may have a disproportionate effect on the relationship between the theft rate and the percentage of minorities.

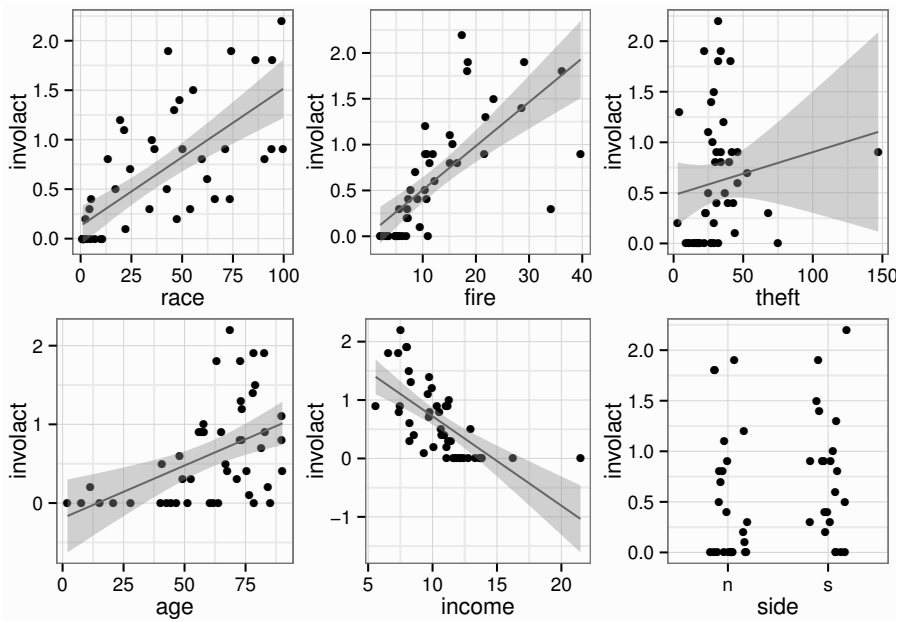


Figure 12.2 Plots of the Chicago insurance data.

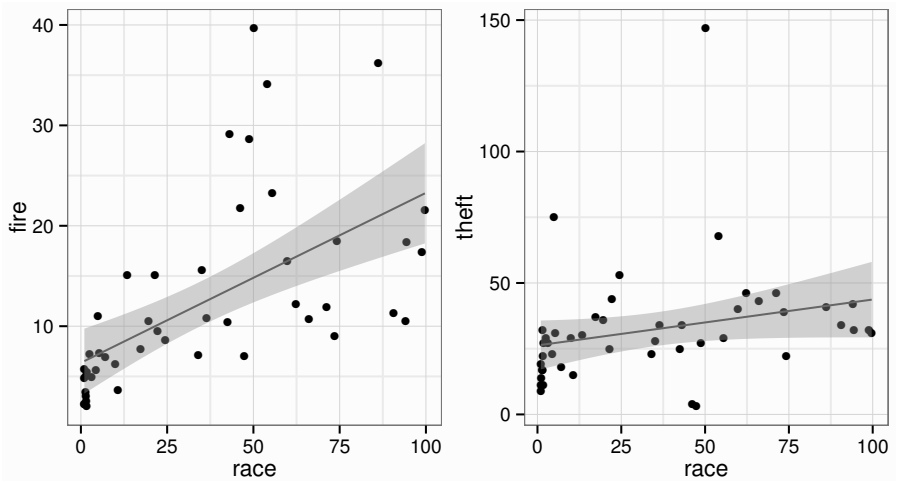


Figure 12.3 Relationship between fire, theft and race in the Chicago data.

The question of which variables should also be included in the regression so that their effect may be adjusted for is difficult. Statistically, we can do it, but the important question is whether it should be done at all. For example, it is known that the incomes of women in the United States and other countries are generally lower than those of men. However, if one adjusts for various predictors such as type of job and length of service, this gender difference is reduced or can even disappear. The controversy is not statistical but political — should these predictors be used to make the adjustment?

For the present data, suppose that the effect of adjusting for income differences was to remove the race effect. This would pose an interesting, but non-statistical question. I have chosen to include the `income` variable in the analysis just to see what happens.

I have decided to use $\log(\text{income})$ partly because of skewness in this variable, but also because income is better considered on a multiplicative rather than additive scale. In other words, \$1,000 is worth a lot more to a poor person than a millionaire because \$1,000 is a much greater fraction of the poor person's wealth.

12.3 Full Model and Diagnostics

We start with the full model:

```
> lmod <- lm(involutact ~ race + fire + theft + age + log(income),
             chredlin)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.18554	1.10025	-1.08	0.28755
race	0.00950	0.00249	3.82	0.00045
fire	0.03986	0.00877	4.55	0.000048
theft	-0.01029	0.00282	-3.65	0.00073
age	0.00834	0.00274	3.04	0.00413
log(income)	0.34576	0.40012	0.86	0.39254

```
n = 47, p = 6, Residual SE = 0.335, R-Squared = 0.75
```

Before leaping to any conclusions, we should check the model assumptions. These two diagnostic plots are seen in Figure 12.4:

```
> plot(lmod, 1:2)
```

The diagonal streak in the residual-fitted plot is caused by the large number of zero response values in the data. When $y = 0$, the residual $\hat{\epsilon} = -\hat{y} = -x^T \hat{\beta}$, hence the line. Turning a blind eye to this feature, we see no particular problem. The Q-Q plot looks fine too. This is reassuring since we know from the form of the response with so many zero values, that it cannot possibly be normally distributed. We'll rely on the central limit theorem, the size of the sample and lack of long-tailed or skewed residuals to be comfortable with the reported p -values.

We now look for transformations. We try some partial residual plots as seen in Figure 12.5:

```
> termplot(lmod, partial.resid=TRUE, terms=1:2)
```

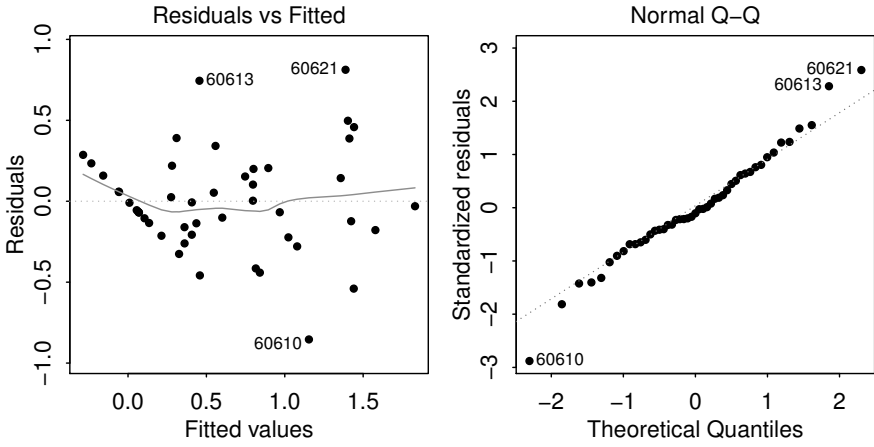



Figure 12.4 Diagnostic plots of the initial model for the Chicago insurance data.

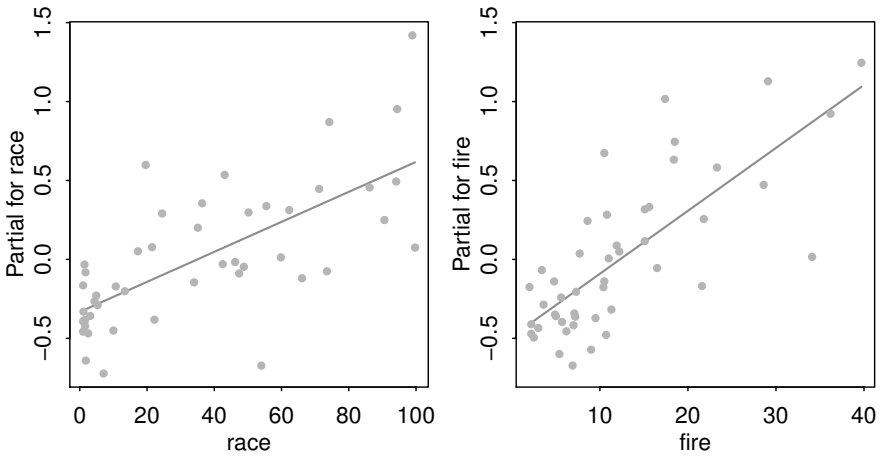


Figure 12.5 Partial residual plots for race and fire.

These plots indicate no need to transform. It would have been inconvenient to transform the `race` variable since that would have made interpretation more difficult. Fortunately, we do not need to worry about this. We examined the other partial residual plots and experimented with polynomials for the predictors. No transformation of the predictors appears to be worthwhile.

We choose to avoid a transformation of the response. The zeros in the response would have restricted the possibilities and furthermore would have made interpretation more difficult. A square root transformation is possible but whatever slim advantage this might offer, it makes explanation more problematic.

12.4 Sensitivity Analysis

How robust is our conclusion to the choice of covariates used to adjust the response? In the full model used earlier, we used all four covariates but we may wonder how sensitive our findings are to this choice. Certainly, one might question whether we should adjust the response for the average income of the zip code. Other objections or uncertainties might be raised by use of the other covariates also.

We can investigate these concerns by fitting other models that vary the choice of adjusting covariates. In this example, there are four such covariates and so there are only 16 possible combinations in which they may be added to the model. It is practical to fit and examine all these models.

The mechanism for creating all 16 models is rather complex and you may wish to skip to the output. The first line creates all subsets of (1,2,3,4). The second line creates the predictor part of the model formulae by pasting together the chosen variables. We then iterate over all 16 models, saving the terms of interest for the `race` variable:

```
> listcombo <- unlist(sapply(0:4,function(x) combn(4, x, simplify=
  FALSE)), recursive=FALSE)
> predterms <- lapply(listcombo, function(x) paste(c("race",c("fire",
  theft", "age", "log(income)")[x]), collapse="+"))
> coefm <- matrix(NA,16,2)
> for(i in 1:16){
  lmi <- lm(as.formula(paste("involact ~ ",predterms[[i]])), data=
    chredlin)
  coefm[i,] <- summary(lmi)$coef[2,c(1,4)]
}
> rownames(coefm) <- predterms
> colnames(coefm) <- c("beta", "pvalue")
> round(coefm, 4)
```

	beta	pvalue
race	0.0139	0.0000
race+fire	0.0089	0.0002
race+theft	0.0141	0.0000
race+age	0.0123	0.0000
race+log(income)	0.0082	0.0087
race+fire+theft	0.0082	0.0002
race+fire+age	0.0089	0.0001
race+fire+log(income)	0.0070	0.0160
race+theft+age	0.0128	0.0000
race+theft+log(income)	0.0084	0.0083

```

race+age+log(income)           0.0099 0.0017
race+fire+theft+age           0.0081 0.0001
race+fire+theft+log(income)   0.0073 0.0078
race+fire+age+log(income)     0.0085 0.0041
race+theft+age+log(income)    0.0106 0.0010
race+fire+theft+age+log(income) 0.0095 0.0004
    
```

The output shows the $\hat{\beta}_1$ and the associated p -values for all 16 models. We can see that the value of $\hat{\beta}_1$ varies somewhat with a high value about double the low value. But in no case does the p -value rise above 5%. So although we may have some uncertainty over the magnitude of the effect, we can be sure that the significance of the effect is not sensitive to the choice of adjusters.

Suppose the outcome had not been so clear cut and we were able to find models where the predictor of interest (in this case, `race`) was not statistically significant. The investigation would then have become more complex because we would need to consider more deeply which covariates should be adjusted for and which not. Such a discussion is beyond the scope of this book, but illustrates why causal inference is a difficult subject.

We should also be concerned whether our conclusions are sensitive to the inclusion or exclusion of a small number of cases. Influence diagnostics are useful for this purpose. We start with a plot of the differences in the coefficient caused by the removal of one point. These can be seen for the `race` variable in Figure 12.6.

```

> diags <- data.frame(lm.influence(lmod)$coef)
> ggplot(diags, aes(row.names(diags), race)) +geom_linerange(aes(ymin=race,
  ymax=race)) +theme(axis.text.x=element_text(angle=90)) + xlab("
  ZIP") +geom_hline(yint=0)
    
```

The `ggplot` function requires the data in the form of a data frame. We extract the relevant component from the the `lm.influence` call for this purpose. We can see that the largest reduction is about 0.001 which would be insufficient to change the statistical significance of this term.

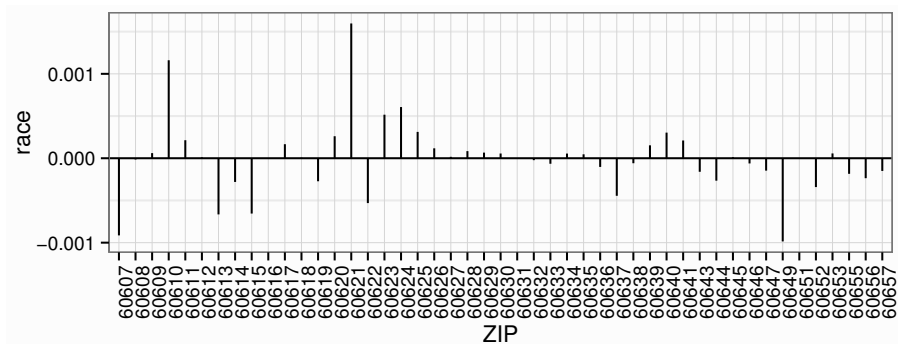


Figure 12.6 Leave-out-one change in coefficient values for $\hat{\beta}_{race}$.

It is also worth considering the influence on the adjustment covariates We plot the leave-out-one differences in $\hat{\beta}$ for `theft` and `fire`:

```
> ggplot(diags, aes(x=fire, y=theft)) + geom_text(label=row.names(diags))
```

Let's also take a look the standardized residuals and leverage which can be conveniently constructed using the default `plot` function for a linear model object:

```
> plot(lmod, 5)
```

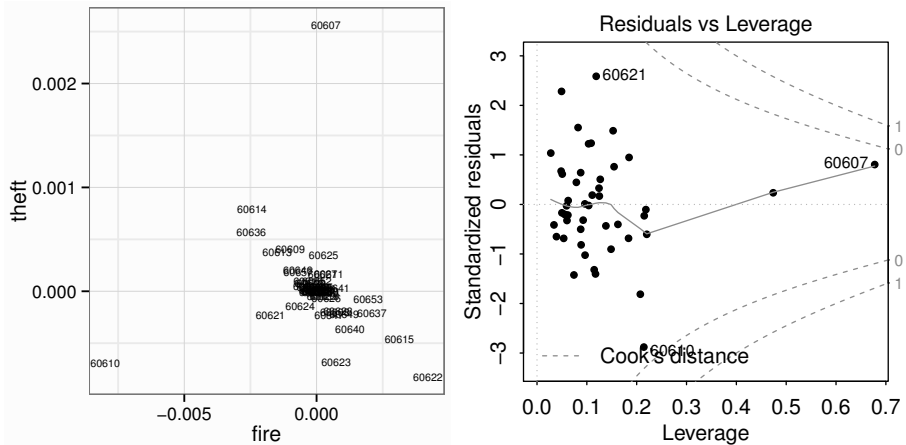


Figure 12.7 Plot of the leave-out one coefficient differences is shown on the left. Plot of the standardized residuals against the leverages is shown on the right

See Figure 12.7 where zip codes 60607 and 60610 stick out. It is worth looking at other leave-out-one coefficient plots also. We also notice that there is no standardized residual extreme enough to call an outlier. Let's take a look at the two cases:

```
> chredlin[c("60607", "60610"), ]
      race fire theft age involact income side
60607 50.2 39.7  147 83.0      0.9  7.459   n
60610 54.0 34.1   68 52.6      0.3  8.231   n
```

These are high theft and fire zip codes. See what happens when we exclude these points:

```
> match(c("60607", "60610"), row.names(chredlin))
[1] 24 6
> lmode <- lm(involact ~ race + fire + theft + age + log(income),
             chredlin, subset=-c(6, 24))
> sumary(lmode)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.57674    1.08005   -0.53   0.596
race          0.00705    0.00270    2.62   0.013
fire         0.04965    0.00857    5.79  1e-06
theft       -0.00643    0.00435   -1.48  0.147
age          0.00517    0.00289    1.79  0.082
log(income)  0.11570    0.40111    0.29  0.775
```

```
n = 45, p = 6, Residual SE = 0.303, R-Squared = 0.8
```

The predictors `theft` and `age` are no longer significant at the 5% level. The coefficient for `race` is reduced compared to the full data fit but remains statistically significant.

So we have verified that our conclusions are also robust to the exclusion of one or perhaps two cases from the data. This is reassuring since a conclusion based on the accuracy of measurement for a single case would be a cause for concern. If this problem did occur, we would need to be particularly sure of these measurements. In some situations, one might wish to drop such influential cases but this would require strong arguments that such points were in some way exceptional. In any case, it would be very important to disclose this choice in the analysis.

Now if we try very hard to poke a hole in our result, we can find this model where two cases have been dropped:

```
> modalt <- lm(involact ~ race+fire+log(income), chredlin, subset=-c
(6, 24))
> sumary(modalt)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.75326	0.83588	0.90	0.373
race	0.00421	0.00228	1.85	0.072
fire	0.05102	0.00845	6.04	3.8e-07
log(income)	-0.36238	0.31916	-1.14	0.263

```
n = 45, p = 4, Residual SE = 0.309, R-Squared = 0.79
```

In this model, `race` no longer meets the threshold for significance. However, there is no compelling reason to advocate for this model against the large weight of other alternatives we have considered.

This illustrates a wider problem with regression modeling in that the data usually do not unequivocally suggest one particular model. It is easy for independent analysts to apply similar methods but in different orders and in somewhat different ways resulting in different model choices. See Faraway (1994) for some examples. For this reason, the good analyst explores the data thoroughly and considers multiple models. One might settle on one final model but confidence in the conclusions will be enhanced if it can be shown that competing models result in similar conclusions. Our analysis in this chapter demonstrates this concern for alternatives but there is an unavoidable reliance on human judgement. An unscrupulous analyst can explore a large number of models but report only the one that favors a particular conclusion.

A related concept is *model uncertainty*. We surely do not know the true model for this data and somehow our conclusions should reflect this. The regression summary outputs provide standard errors and p -values that express our uncertainty about the parameters of the model but they do not reflect the uncertainty about the model itself. This means that we will tend to be more confident about our inferences than is justified. There are several possible ways to mitigate this problem. One simple approach is data splitting as used in the running example on the `meat` data in Chapter 11. Another idea is to bootstrap the whole data analysis as demonstrated by Faraway (1992). Alternatively, it may be possible to use *model averaging* as in Raftery, Madigan, and Hoeting (1997).

12.5 Discussion

There is some ambiguity in the conclusion here. These reservations have several sources. There is some doubt because the response is not a perfect measure of people being denied insurance. It is an aggregate measure that raises the problem of ecological correlations. We have implicitly assumed that the probability a minority homeowner would obtain a FAIR plan after adjusting for the effect of the other covariates is constant across zip codes. This is unlikely to be true. If the truth is simply a variation about some constant, then our conclusions will still be reasonable, but if this probability varies in a systematic way, then our conclusions may be off the mark. It would be a very good idea to obtain some individual level data.

We have demonstrated statistical significance for the effect of race on the response. But statistical significance is not the same as practical significance. The largest value of the response is only 2.2% and most other values are much smaller. Using our preferred models, the predicted difference between 0% minority and 100% minority is about 1%. So while we may be confident that some people are affected, there may not be so many of them. We would need to know more about predictors like insurance renewal rates to say much more but the general point is that the size of the p -value does not tell you much about the practical size of the effect.

There is also the problem of a potential latent variable that might be the true cause of the observed relationship. Someone with first-hand knowledge of the insurance business might propose one. This possibility always casts a shadow of doubt on our conclusions.

Another issue that arises in cases of this nature is how much the data should be aggregated. For example, suppose we fit separate models to the two halves of the city. Fit the model to the south of Chicago:

```
> lmod <- lm(involutact ~ race+fire+theft+age, subset=(side == "s"),
  chredlin)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.23441	0.23774	-0.99	0.338
race	0.00595	0.00328	1.81	0.087
fire	0.04839	0.01689	2.87	0.011
theft	-0.00664	0.00844	-0.79	0.442
age	0.00501	0.00505	0.99	0.335

n = 22, p = 5, Residual SE = 0.351, R-Squared = 0.74

and now to the north:

```
> lmod <- lm(involutact ~ race+fire+theft+age, subset=(side == "n"),
  chredlin)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-0.31857	0.22702	-1.40	0.176
race	0.01256	0.00448	2.81	0.011
fire	0.02313	0.01398	1.65	0.114
theft	-0.00758	0.00366	-2.07	0.052
age	0.00820	0.00346	2.37	0.028

n = 25, p = 5, Residual SE = 0.343, R-Squared = 0.76

We see that `race` is significant in the north, but not in the south. By dividing the data into smaller and smaller subsets it is possible to dilute the significance of any predictor. On the other hand, it is important not to aggregate all data without regard to whether it is reasonable. Clearly a judgment has to be made and this can be a point of contention in legal cases.

There are some special difficulties in presenting this during a court case. With scientific inquiries, there is always room for uncertainty and subtlety in presenting the results, particularly if the subject matter is not contentious. In an adversarial proceeding, it is difficult to present statistical evidence when the outcome is not clear-cut, as in this example. There are particular difficulties in explaining such evidence to non-mathematically trained people.

After all this analysis, the reader may be feeling somewhat dissatisfied. It seems we are unable to come to any truly definite conclusions and everything we say has been hedged with “ifs” and “buts.” Winston Churchill once said:

Indeed, it has been said that democracy is the worst form of Government except all those other forms that have been tried from time to time.

We might say the same about statistics with respect to how it helps us reason in the face of uncertainty. It is not entirely satisfying but the alternatives are worse.

Exercises

In all the following questions, a full answer requires you to perform a complete analysis of the data including an initial data analysis, regression diagnostics, a search for possible transformations and a consideration of model selection. A report on your analysis needs to be selective in its content. You should include enough information for the steps leading to your selection of model to be clear and reproducible by the reader. But you should not include everything you tried. Dead ends can be reported in passing but do not need to be described in full detail unless they contain some message of interest. Above all your analysis should have a clear statement of the conclusion of your analysis.

1. Reliable records of temperature taken using thermometers are only available back to the 1850s, but it would be interesting to estimate global temperatures in the pre-industrial era. It is possible to obtain various *proxy* measures of temperature. Trees grow faster in warmer years so the width of tree rings (seen in tree cross-sections) provides some evidence of past temperatures. Other natural sources of proxies include coral and ice cores. Such information can go back for a thousand years or more. The dataset `globwarm` contains information on eight proxy measures and northern hemisphere temperatures back to 1856. Build a model and predict temperatures back to 1000 AD. State the uncertainty in your predictions. Comment on your findings. (Note: that this data has been modified and simplified for the purposes of this exercise — see the R help page for the data to find out about the source).
2. The `happy` dataset contains data from 39 MBA students on predictors affecting

happiness. Analyze the data to provide an interpretation for the relationship between happiness and money, sex, love and work.

3. The `mammalsleep` dataset contains data on 62 mammals. Focus your interest on the effect of predation on sleep, making proper adjustments for the effects of other predictors.

Missing Data

13.1 Types of Missing Data

Here are some of the ways that missing data can arise in a regression setting:

Missing cases Sometimes we fail to observe a complete case (x_i, y_i) . Indeed, when we draw a sample from a population, we do not observe the unsampled cases. When missing data arise in this manner, there is no difficulty since this is the standard situation for much of statistics. But sometimes there are cases we intended to sample but failed to observe. If the reason for this failure is unrelated to what would have been observed, then we simply have a smaller sample and can proceed as normal. But when data are not observed for reasons that have some connection to what we would have seen, then we have a biased sample. Sometimes, given enough information about the mechanism for missingness, we can make corrections and achieve valid inferences.

Incomplete values Suppose we run an experiment to study the lifetimes of light bulbs. We might run out of time waiting for all the lightbulbs to die and decide to end the experiment. These incomplete cases would provide the information that a bulb lasted at least some amount of time but we would not know how long it would have lasted had we waited until it died. Similar examples arise in medical trials where patient final outcomes are not known. Such cases are said to be *censored*. Methods like survival analysis or reliability analysis can handle such data.

Missing values Sometimes we observe some components of a case but not others. We might observe the values of some predictors but not others. Perhaps the predictors are observed but not the response.

In this chapter, we address the missing value problem. Biased sampling due to missing cases can sometimes be mitigated by covariate adjustment while methods for analyzing censored data may be found in other books.

What can be done? Finding the missing values is the best option, but this may not be possible because the values were never recorded or were lost in the data collection process. Next, ask why the data are missing. We can distinguish several kinds of missingness:

Missing Completely at Random (MCAR) The probability that a value is missing is the same for all cases. If we simply delete all cases with missing values from the analysis, we will cause no bias, although we may lose some information.

Missing at Random (MAR) The probability of a value being missing depends on a known mechanism. For example, in social surveys, certain groups are less likely

to provide information than others. As long as we know the group membership of the individual being sampled, then this is an example of MAR. We can delete these missing cases provided we adjust for the group membership by including this as a factor in the regression model. We will see later how we may do better than simply deleting such cases.

Missing not at Random (MNAR) The probability that a value is missing depends on some unobserved variable or, more seriously, on what value would have been observed. For example, people who have something to hide are typically less likely to provide information that might reveal something embarrassing or illegal.

Dealing with the MNAR case is difficult or even impossible so we confine our attention to MAR problems.

Determining the type of missingness for data usually requires judgement alone. There are no obvious diagnostic methods to check since the data we need for such checks are missing.

13.2 Deletion

Suppose some of the values in the Chicago insurance dataset were missing. I randomly declared some of the observations missing in this modified dataset:

```
> data(chmiss, package="faraway")
> chmiss
      race fire theft age involact income
60626 10.0  6.2   29 60.4         NA 11.744
60640 22.2  9.5   44 76.5         0.1  9.323
60613 19.6 10.5   36  NA         1.2  9.948
60657 17.3  7.7   37  NA         0.5 10.656
--- etc ---
60645  3.1  4.9   27  NA         0.0 13.731
```

Sometimes, we are warned that there are missing values and we should always be alert to this possibility in our initial data analysis. For a small dataset, simply looking at the data reveals the problem. For larger datasets, the standard summary reports on missing values:

```
> summary(chmiss)
      race           fire           theft           age
Min.   : 1.00   Min.   : 2.0   Min.   : 3.0   Min.   : 2.0
1st Qu.: 3.75   1st Qu.: 5.6   1st Qu.: 22.0  1st Qu.:48.3
Median :24.50   Median : 9.5   Median : 29.0  Median :64.4
Mean   :35.61   Mean   :11.4   Mean   : 32.7   Mean   :60.0
3rd Qu.:57.65   3rd Qu.:15.1   3rd Qu.: 38.0  3rd Qu.:78.2
Max.   :99.70   Max.   :36.2   Max.   :147.0  Max.   :90.1
NA's   : 4     NA's   :2     NA's   :4     NA's   :5

      involact           income
Min.   :0.000   Min.   : 5.58
1st Qu.:0.000   1st Qu.: 8.56
Median :0.500   Median :10.69
Mean   :0.648   Mean   :10.74
3rd Qu.:0.925   3rd Qu.:12.10
Max.   :2.200   Max.   :21.48
NA's   : 3     NA's   :2
```

We can see missing values in all variables. It is also helpful to see how many missing values appear in each case.

```
> rowSums(is.na(chmiss))
60626 60640 60613 60657 60614 60610 60611 60625 60618 60647 60622
      1      0      1      1      0      0      0      0      1      1      0
60631 60646 60656 60630 60634 60641 60635 60639 60651 60644 60624
      0      1      0      0      1      0      0      0      1      1      0
60612 60607 60623 60608 60616 60632 60609 60653 60615 60638 60629
      0      1      0      1      1      0      1      0      0      0      1
60636 60621 60637 60652 60620 60619 60649 60617 60655 60643 60628
      0      1      0      0      1      0      1      1      0      0      1
60627 60633 60645
      0      0      1
```

We see there is at most one missing value in each row. We can also plot the missing value information as seen in Figure 13.1. In this example, the missing cases are evenly scattered throughout the data. In some cases, the missing values are concentrated in some variables or cases. This can make it easier to drop variables from the analysis or delete cases without losing much information. But in this example, deleting the missing cases will lose 20 out of 47 observations.

```
> image(is.na(chmiss), axes=FALSE, col=gray(1:0))
> axis(2, at=0:5/5, labels=colnames(chmiss))
> axis(1, at=0:46/46, labels=row.names(chmiss), las=2)
```

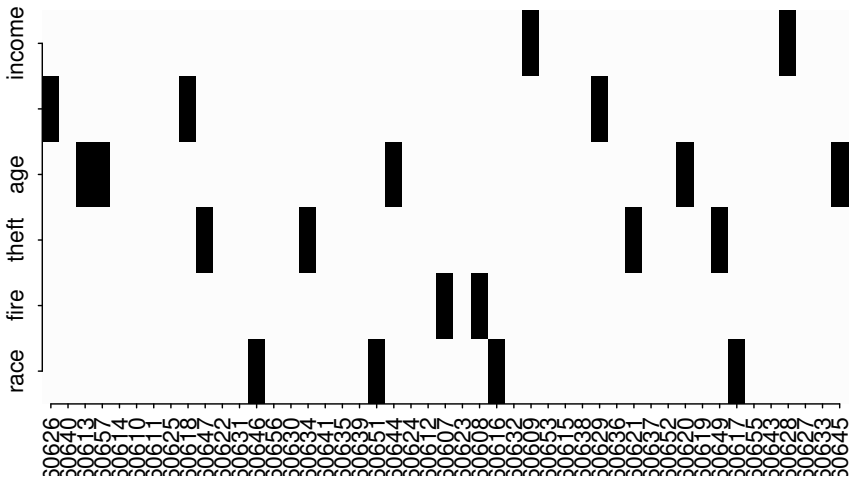


Figure 13.1 Missing values in the simulated Chicago insurance data.

Let's see what happens when we use a deletion strategy. First consider the full data fit for comparison purposes:

```
> data(chredlin, package="faraway")
> modfull <- lm(involact ~ ., chredlin)
> summary(modfull)
      Estimate Std. Error t value Pr(>|t|)
-----
```

```
(Intercept) -4.86e-01  6.02e-01  -0.81  0.42411
race         8.53e-03  2.86e-03   2.98  0.00491
fire         3.78e-02  8.98e-03   4.21  0.00014
theft        -1.02e-02  2.91e-03  -3.49  0.00118
age          7.62e-03  3.33e-03   2.29  0.02758
volact       -1.02e-02  2.77e-02  -0.37  0.71552
income       2.57e-05  3.22e-05   0.80  0.42976

n = 47, p = 7, Residual SE = 0.339, R-Squared = 0.75
```

Of course, we would not be able to see such output in practice. Now compare this to a fit to the missing version of the data:

```
> modmiss <- lm(involact ~ ., chmiss)
> summary(modmiss)
      Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.11648    0.60576   -1.84  0.07947
race          0.01049    0.00313    3.35  0.00302
fire          0.04388    0.01032    4.25  0.00036
theft        -0.01722    0.00590   -2.92  0.00822
age           0.00938    0.00349    2.68  0.01390
income        0.06870    0.04216    1.63  0.11808

n = 27, p = 6, Residual SE = 0.338, R-Squared = 0.79
```

The default behavior of the `lm` function is to omit any case with missing values. We are left with only 27 complete cases for the regression. We can see that the standard errors are larger for the missing version because we have less data to fit the model and so the estimates are less precise.

Deleting missing cases is the simplest strategy for dealing with missing data. It avoids the complexity and possible biases introduced by the more sophisticated methods that we will discuss. The drawback is that we are throwing away information that might allow more precise inference. A decision on whether deletion is an acceptable missing data strategy depends on the circumstances. If relatively few cases contain missing values, if deleting missing cases still leaves a large dataset or if you wish to communicate a simple data analysis method, the deletion strategy is satisfactory.

13.3 Single Imputation

A simple solution to the problem is to fill in or *impute* the missing values. For example, we can fill in the missing values by the variable means:

```
> (cmeans <- colMeans(chmiss, na.rm=TRUE))
      race      fire      theft      age      involact      income
35.60930 11.42444 32.65116 59.96905  0.64773 10.73587
> mchm <- chmiss
> for(i in c(1:4,6)) mchm[is.na(chmiss[,i]),i] <- cmeans[i]
```

We do not fill in missing values in the response because this is the variable we are trying to model. The cases with a missing response still have some value in imputing the other missing predictor values. Now refit:

```

> imod <- lm(involact ~ ., mchm)
> sumary(imod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.07080    0.50945   0.14   0.890
race         0.00712    0.00271   2.63   0.012
fire        0.02874    0.00939   3.06   0.004
theft       -0.00306    0.00275  -1.11   0.272
age         0.00608    0.00321   1.90   0.066
income      -0.02709    0.03168  -0.86   0.398

n = 44, p = 6, Residual SE = 0.384, R-Squared = 0.68

```

There are some important differences between these two fits. For example, `theft` and `age` are significant in the first fit, but not in the second. Also, the regression coefficients are now all closer to zero. The situation is analogous to the errors in variables case. The bias introduced by the fill-in method can be substantial and may not be compensated by the attendant reduction in variance. For this reason, mean imputation is not recommended except where the fraction of filled values is small.

Missing values may also arise for categorical variables. We can impute with the most common level or category of the variable. Alternatively, a missing value can simply be regarded as an additional level of the variable so, for example, we might have `male`, `female` and `unknown` levels.

A more sophisticated alternative to mean imputation is to use regression methods to predict the missing values of the covariates. Let's try to fill in the missing race values:

```

> lmodr <- lm(race ~ fire+theft+age+income, chmiss)
> chmiss[is.na(chmiss$race), ]
      race fire theft  age involact  income
60646  NA  5.7   11 27.9      0.0 16.250
60651  NA 15.1   30 89.8      0.8 10.510
60616  NA 12.2   46 48.0      0.6  8.212
60617  NA 10.8   34 58.0      0.9 11.156
> predict(lmodr, chmiss[is.na(chmiss$race), ])
      60646  60651  60616  60617
-17.847  26.360  70.394  32.620

```

Notice that the first prediction is negative. One trick that can be applied when the response is bounded between zero and one is the logit transformation:

$$y \rightarrow \log(y/(1-y))$$

This transformation maps the interval to the whole real line. The logit function and its inverse are defined in the `faraway` package. We now fit the model with a logit-transformed response and then back transform the predicted values, remembering to convert our percentages to proportions and vice versa at the appropriate times:

```

> lmodr <- lm(logit(race/100) ~ fire+theft+age+income, chmiss)
> ilogit(predict(lmodr, chmiss[is.na(chmiss$race), ])) * 100
      60646  60651  60616  60617
 0.41909 14.73202 84.26540 21.31213

```

We can see how our predicted values compare to the actual values:

```
> chredlin$race[is.na(chmiss$race)]
[1] 1.0 13.4 62.3 36.4
```

So our first two predictions are good, but the other two are somewhat wide of the mark. We still need to impute the missing values for the other predictors which will take us some manual effort.

Like the mean fill-in method, the regression fill-in method will also introduce a bias toward zero in the coefficients while tending to reduce the variance. The success of the regression method depends somewhat on the collinearity of the predictors — the filled-in values will be more accurate the more collinear the predictors are. However, for collinear data one might consider deleting a predictor riddled with missing values rather than imputing because other variables that are strongly correlated will bear the predictive load.

13.4 Multiple Imputation

The single imputation methods described above cause bias while deletion causes a loss of information from the data. Multiple imputation is a way to reduce the bias caused by single imputation. The problem with single imputation is that the imputed value, be it a mean or a regression-predicted value, tends to be less variable than the value we would have seen because the imputed value does not include the error variation that would normally be seen in observed data. The idea behind multiple imputation is to reinclude that error variation — we add back on the error to the imputed value. Of course, if we only did this once, the outcome would be an inferior single imputation method so we repeat the imputation multiple times.

Multiple imputation is implemented in the *Amelia* package of Honaker, King, and Blackwell (2011). There are various methods of performing the imputation, depending on the variable type. A basic assumption is that the data is multivariate normal. The methodology is quite robust to this assumption but we need to make modifications in some cases. Heavily skewed variables are best log-transformed before imputation and categorical variables need to be declared for special treatment. These issues do not apply for the insurance data. It is not thought necessary to repeat the imputation many times — we have used 25 in this instance:

```
> require(Amelia)
> set.seed(123)
> chimp <- amelia(chmiss, m=25)
```

You will notice that we have included the whole data frame in the imputation even though this includes the income variable that we shall not be using. This variable may contain information that may be helpful in predicting missing values in the variables we do include. Furthermore, inclusion makes the MAR assumption more tenable. It is also important that the imputation model used for imputing the values be at least as rich as the model we fit to the data.

We can now fit the linear model to each of the 25 imputed datasets and store the results about $\hat{\beta}$ and $se(\hat{\beta})$.

```
> betas <- NULL
```

```

> ses <- NULL
> for(i in 1:chimp$m) {
  lmod <- lm(involact ~ race+fire+theft+age, chimp$imputations[[i]])
  betas <- rbind(betas,coef(lmod))
  ses <- rbind(ses,coef(summary(lmod))[,2])
}

```

We can combine the results to get estimates of the coefficients and their standard errors. Let $\hat{\beta}_{ij}$ and s_{ij} be the estimates and standard errors for the i^{th} imputed result for $i = 1, \dots, m$ and for the j^{th} parameter. The combined estimate of $\hat{\beta}$ is obtained by averaging over the imputations:

$$\hat{\beta}_j = \frac{1}{m} \sum_i \hat{\beta}_{ij}$$

and the combined standard errors are given by:

$$s_j^2 = \frac{1}{m} \sum_i s_{ij}^2 + \text{var } \hat{\beta}_j (1 + 1/m)$$

where $\text{var } \hat{\beta}_j$ is the sample variance over the imputed $\hat{\beta}_{ij}$. The `mi.meld` function makes these calculations:

```

> (cr <- mi.meld(q=betas, se=ses))
$q.mi
  (Intercept)      race      fire      theft      age
[1,] -0.22551  0.0075952  0.035828 -0.008689  0.0070798

$se.mi
  (Intercept)      race      fire      theft      age
[1,]  0.1574  0.0019703  0.0085159  0.0045773  0.0025797

```

We can compute the t -statistics as:

```

> cr$q.mi/cr$se.mi
  (Intercept)      race      fire      theft      age
[1,] -1.4328  3.8548  4.2073 -1.8983  2.7444

```

Comparing the results to the full data fit seen in the last chapter, we see that the results are quite close excepting that the theft predictor is no longer statistically significant.

See Little and Rubin (2002) and Schafer (1997) for more about missing data methods.

Exercises

- The dataset `kanga` contains data on the skulls of historical kangaroo specimens. Ignore the sex and species variables for the purposes of this question.
 - Report on the distribution of missing values within the data according to case and variable.
 - Determine a combination of variable and case deletions that retains the most information.

- (c) Compute the PCA, omitting all missing cases, where all the variables have been scaled to have SD 1. Report on the standard deviation of the principal components.
 - (d) Perform multiple imputation 25 times, applying PCA in the same manner to each of the imputed datasets. Combine the standard deviations of the principal components in the imputed datasets. Compare your answer to the values obtained in the previous question.
2. The dataset `galamiss` contains the Galapagos data frequently featured as an example in this text as `gala` but with the original missing values left in.
 - (a) Fit a linear model using `gala` with the number of species as the response and the five geographic predictors as in earlier examples.
 - (b) Fit the same model to `galamiss` using the deletion strategy for missing values. Compare the fit to that in (a).
 - (c) Use mean value single imputation on `galamiss` and again fit the model. Compare to previous fits.
 - (d) Use a regression-based imputation using the other four geographic predictors to fill in the missing values in `galamiss`. Fit the same model and compare to previous fits.
 - (e) Use multiple imputation to handle missing values and fit the same model again. Compare to previous fits.
3. The `pima` dataset contains information on 768 adult female Pima Indians living near Phoenix.
 - (a) The analysis in Chapter 1 suggests that zero has been used as a missing value code for several of the variables. Replace these values with NA. Describe the distribution of missing values in the data.
 - (b) Fit a linear model with `diastolic` as the response and the other variables as predictors. Summarize the fit.
 - (c) Use mean value imputation to the missing cases and refit the model comparing to fit found in the previous question.
 - (d) Use a regression-based imputation using the other four geographic predictors to fill in the missing values in the data. Fit the same model and compare to previous fits.
 - (e) Use multiple imputation to handle missing values and fit the same model again. Compare to previous fits.

Categorical Predictors

Predictors that are qualitative in nature, for example, eye color, are sometimes described as *categorical* or called *factors*. The different categories of a factor variable are called levels. For example, suppose we recognize eye colors of “blue”, “green”, “brown” and “hazel”, then we would say eye color is a factor with four levels.

We wish to incorporate these predictors into the regression analysis. We start with the example of a factor with just two levels, then show how to introduce quantitative predictors into the model and end with an example using a factor with more than two levels.

14.1 A Two-Level Factor

The data for this example come from a study of the effects of childhood sexual abuse on adult females reported in Rodriguez et al. (1997): 45 women treated at a clinic, who reported childhood sexual abuse (*csa*), were measured for post-traumatic stress disorder (*ptsd*) and childhood physical abuse (*cpa*) both on standardized scales. Thirty-one women treated at the same clinic, who did not report childhood sexual abuse, were also measured. The full study was more complex than reported here and so readers interested in the subject matter should refer to the original article.

We take a look at the data and produce a summary subsetted by *csa*:

```
> data(sexab, package="faraway")
> sexab
      cpa      ptsd      csa
1  2.04786  9.71365  Abused
2  0.83895  6.16933  Abused
.....
75 2.85253  6.84304 NotAbused
76 0.81138  7.12918 NotAbused
> by(sexab, sexab$csa, summary)
sexab$csa: Abused
      cpa      ptsd      csa
Min.   :-1.11  Min.   : 5.98  Abused   :45
1st Qu.: 1.41  1st Qu.: 9.37  NotAbused: 0
Median : 2.63  Median :11.31
Mean   : 3.08  Mean   :11.94
3rd Qu.: 4.32  3rd Qu.:14.90
Max.   : 8.65  Max.   :18.99
-----
sexab$csa: NotAbused
      cpa      ptsd      csa
Min.   :-3.12  Min.   :-3.35  Abused   : 0
```

```

1st Qu.: -0.23   1st Qu.: 3.54   NotAbused: 31
Median : 1.32   Median : 5.79
Mean   : 1.31   Mean   : 4.70
3rd Qu.: 2.83   3rd Qu.: 6.84
Max.   : 5.05   Max.   : 10.91

```

Now plot the data — see Figure 14.1.

```

> plot(ptsd ~ csa, sexab)
> plot(ptsd ~ cpa, pch=as.character(csa), sexab)

```

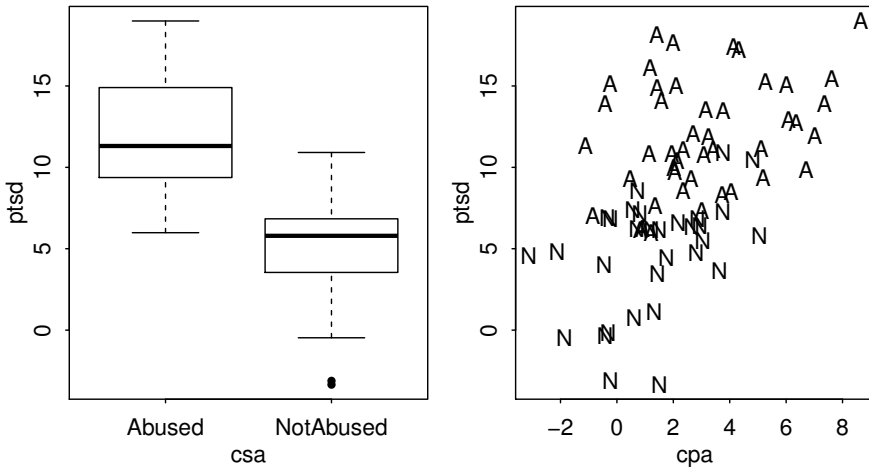


Figure 14.1 PTSD comparison of abused and non-abused subjects on the left. A = Abused and N = NotAbused on the right.

We see that those in the abused group have higher levels of PTSD than those in the non-abused in the left panel of Figure 14.1. We can test this difference:

```

> t.test(ptsd ~ csa, sexab, var.equal=TRUE)

```

```

Two Sample t-test

data: ptsd by csa
t = 8.9387, df = 74, p-value = 2.172e-13
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 5.6302 8.8603
sample estimates:
 mean in group Abused mean in group NotAbused
      11.9411              4.6959

```

and find that it is clearly significant. We have specified that the variance is equal in the two groups. The assumption is reasonable here, but it also ensures comparability with the linear modeling to follow.

Our strategy is to incorporate qualitative predictors within the $y = X\beta + \varepsilon$ framework. We can then use the estimation, inferential and diagnostic techniques that we

have already learned in a generic way. This also avoids having to learn a different set of formulae for each new type of qualitative predictor configuration.

To put qualitative predictors into the $y = X\beta + \varepsilon$ form we need to code the qualitative predictors. We can do this using *dummy variables*. For a categorical predictor (or factor) with two levels, we define dummy variables d_1 and d_2 :

$$d_i = \begin{cases} 0 & \text{is not level } i \\ 1 & \text{is level } i \end{cases}$$

Let's create these dummy variables and fit them using a linear model:

```
> d1 <- ifelse(sexab$cса == "Abused", 1, 0)
> d2 <- ifelse(sexab$cса == "NotAbused", 1, 0)
> lmod <- lm(ptsd ~ d1 + d2, sexab)
> summary(lmod)
Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.696      0.624    7.53  1.0e-10
d1             7.245      0.811    8.94  2.2e-13

n = 76, p = 2, Residual SE = 3.473, R-Squared = 0.52
```

We can see a warning about singularities and that the parameter for the second dummy variables has not been estimated. The cause of this problem can be revealed by studying the X model matrix:

```
> model.matrix(lmod)
      (Intercept) d1 d2
1                1  1  0
2                1  1  0
...
44               1  1  0
45               1  1  0
46               1  0  1
47               1  0  1
...
76               1  0  1
```

We can see that the sum of the second and third columns equals the first column. This means that X is not of full rank, having a rank of two, not three. Hence not all the parameters can be identified. This should not be surprising since we are trying to use three parameters to model only two groups.

We have more parameters than we need so the solution is to get rid of one of them. Once choice would be to eliminate d_1 :

```
> lmod <- lm(ptsd ~ d2, sexab)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.941      0.518   23.07 < 2e-16
d2           -7.245      0.811   -8.94  2.2e-13

n = 76, p = 2, Residual SE = 3.473, R-Squared = 0.52
```

Compare this to the output of the t-test. The intercept of 11.941 is the mean of the first group ("Abused") while the parameter for d_2 represents the difference between

the second and first group, i.e., $11.941 - 7.245 = 4.696$. The t -value for d_2 of -8.94 is the test statistic for the test that the difference is zero and is identical (excepting the sign) to the test statistic from the t -test computed previously. One assumption of the linear model is that the variances of the errors are equal which explains why we specified this option when computing the t -test earlier.

An alternative approach is to eliminate the intercept term:

```
> lmod <- lm(ptsd ~ d1 + d2 -1, sexab)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
d1    11.941     0.518   23.07 <2e-16
d2     4.696     0.624    7.53  1e-10

n = 76, p = 2, Residual SE = 3.473, R-Squared = 0.89
```

The advantage of this approach is that the means of the two groups are directly supplied by the parameter estimates of the two dummy variables. However, we do not get the t -test for the difference. The tests in the output correspond to hypotheses claiming the mean response in the group is zero. These are not interesting because these hypotheses are unbelievable.

Furthermore, the solution of dropping the intercept only works when there is a single factor and does not generalize to the multiple factor case. Another unpleasant side effect is that the R^2 is not correctly computed when the intercept is omitted. For these reasons, we prefer approach of dropping one of the dummy variables to dropping the intercept.

It is not necessary to explicitly form the dummy variables as R can produce these directly by just including the factor in the model formula:

```
> lmod <- lm(ptsd ~ csa, sexab)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.941     0.518   23.07 < 2e-16
csaNotAbused -7.245     0.811   -8.94 2.2e-13

n = 76, p = 2, Residual SE = 3.473, R-Squared = 0.52
```

We can check that `csa` is a factor variable:

```
> class(sexab$csa)
[1] "factor"
```

This usually happens automatically when a variable takes non-numeric values although it can be imposed directly if necessary using the `factor()` command. The dummy variables are created but one is dropped to ensure identifiability. This is known as the *reference level*. In this example, the reference level is “Abused”. At first glance, one might be perplexed as “Abused” does not appear in the model summary output. However, the mean response for the reference level is encoded in the intercept of 11.941. The parameter estimate for “NotAbused” of -7.245 is not the mean response for this level but the difference from the reference level. Hence the mean response for the “NotAbused” level is $11.941 - 7.245 = 4.696$. Earlier in our analysis, we dropped the dummy variable d_1 to get the same output.

The choice of reference level is arbitrary. In some examples, there is a natural

choice for the reference level as a control or no-treatment level. In the current example, “NotAbused” is the natural choice for the reference level. The default choice of reference level by R is the first level in alphabetical order, which would be “Abused” in this example. Because this choice is inconvenient, we change the reference level using the `relevel` command:

```
> sexab$csa <- relevel(sexab$csa, ref="NotAbused")
> lmod <- lm(ptsd ~ csa, sexab)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.696	0.624	7.53	1.0e-10
csaAbused	7.245	0.811	8.94	2.2e-13

n = 76, p = 2, Residual SE = 3.473, R-Squared = 0.52

A comparison of the outputs reveals that the fitted values and residuals are the same for either choice — the residual standard error and R^2 will be the same. But the parameterization is different.

Although we have only managed to construct a t -test using linear modeling, a good understanding of how factor variables are handled in R is essential for the more sophisticated models to follow.

14.2 Factors and Quantitative Predictors

We can see from our analysis that women who have suffered childhood sexual abuse tend to have higher levels of PTSD than those who have not. However, we also have information about varying levels of childhood physical abuse (`cpa`) which may also have an effect on PTSD. We need models that can express how a factor variable like `csa` and a quantitative variable like `cpa` might be related to a response.

Suppose we have a response y , a quantitative predictor x and a two-level factor variable represented by a dummy variable d :

$$d = \begin{cases} 0 & \text{reference level} \\ 1 & \text{treatment level} \end{cases}$$

Several possible linear models may be considered here:

1. The same regression line for both levels: $y = \beta_0 + \beta_1x + \epsilon$ or is written in R as $y \sim x$. This model allows no effect for the factor.
2. A factor predictor but no quantitative predictor: $y = \beta_0 + \beta_2d + \epsilon$. This is written as $y \sim d$ in R.
3. Separate regression lines for each group with the same slope: $y = \beta_0 + \beta_1x + \beta_2d + \epsilon$ or is written in R as $y \sim x + d$. In this case β_2 represents the vertical distance between the regression lines (i.e., the effect of the treatment).
4. Separate regression lines for each group with the different slopes: $y = \beta_0 + \beta_1x + \beta_2d + \beta_3x.d + \epsilon$ or is written in R as $y \sim x + d + d:x$ or $y \sim x*d$. To form the slope interaction term $d:x$ in the X -matrix, multiply x by d elementwise. Any interpretation of the effect of the factor will now also depend on the quantitative predictor.

Estimation and testing work just as they did before. Interpretation is easier if we can eliminate the interaction term.

We start with the separate regression lines model. $\text{ptsd} \sim \text{cpa} * \text{csa}$ is an equivalent model formula which expands to the version seen below:

```
> lmod4 <- lm(ptsd ~ cpa+csa+cpa:csa, sexab)
> summary(lmod4)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.696	0.711	5.20	1.8e-06
cpa	0.764	0.304	2.51	0.014
csaAbused	6.861	1.075	6.38	1.5e-08
cpa:csaAbused	-0.314	0.368	-0.85	0.397

n = 76, p = 4, Residual SE = 3.279, R-Squared = 0.58

We can discover the coding by examining the X -matrix:

```
> model.matrix(lmod4)
```

	(Intercept)	cpa	csaAbused	cpa:csaAbused
1	1	2.04786	1	2.04786
2	1	0.83895	1	0.83895
...				
45	1	5.11921	1	5.11921
46	1	1.49181	0	0.00000
...				
76	1	0.81138	0	0.00000

The interaction term cpa:csaAbused is represented in the fourth column of the matrix as the product of the second and third columns. We show the fitted regression lines in the first panel of Figure 14.2:

```
> plot(ptsd ~ cpa, sexab, pch=as.numeric(csa))
> abline(3.96, 0.764)
> abline(3.96+6.86, 0.764-0.314, lty=2)
```

The model can be simplified because the interaction term is not significant. We reduce to this model:

```
> lmod3 <- lm(ptsd ~ cpa+csa, sexab)
> summary(lmod3)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.975	0.629	6.32	1.9e-08
cpa	0.551	0.172	3.21	0.002
csaAbused	6.273	0.822	7.63	6.9e-11

n = 76, p = 3, Residual SE = 3.273, R-Squared = 0.58

No further simplification is possible because the remaining predictors are statistically significant. Put the two parallel regression lines on the plot, as seen in the second panel of Figure 14.2.

```
> plot(ptsd ~ cpa, sexab, pch=as.numeric(csa))
> abline(3.98, 0.551)
> abline(3.98+6.27, 0.551, lty=2)
```

The slope of both lines is 0.551, but the “Abused” line is 6.273 higher than the “NonAbused.” From the t -test earlier, the unadjusted estimated effect of childhood

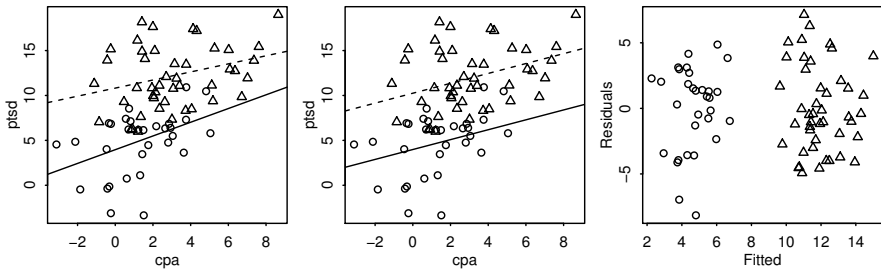


Figure 14.2 In all the three panels, the abused cases are marked with a triangle and the not abused with a circle. The separate lines model fit is shown in the first panel, the parallel lines model in the second panel and the residuals versus fitted plot in the third.

sexual abuse is 7.245. So after adjusting for the effect of childhood physical abuse, our estimate of the effect of childhood sexual abuse on PTSD is mildly reduced.

We can also compare confidence intervals for the effect of `csa`:

```
> confint(lmod3)[3,]
2.5 % 97.5 %
4.6347 7.9108
```

compared to the (5.6302, 8.8603) found for the unadjusted difference. In this particular case, the confidence intervals are about the same width. In other cases, particularly with designed experiments, adjusting for a covariate can increase the precision of the estimate of an effect.

The usual diagnostics should be checked. It is worth checking whether there is some difference related to the categorical variable as we do here:

```
> plot(fitted(lmod3), residuals(lmod3), pch=as.numeric(sexab$csa), xlab="Fitted", ylab="Residuals")
```

We see in the final panel of Figure 14.2 that there are no clear problems. Furthermore, because the two groups happen to separate, we can also see that the variation in the two groups is about the same. If this were not so, we would need to make some adjustments to the analysis, possibly using weights.

We have seen that the effect of `csa` can be adjusted for `cpa`. The reverse is also true. Consider a model with just `cpa`.

```
> lmod1 <- lm(ptsd ~ cpa, sexab)
> summary(lmod1)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.552      0.707     9.26 5.3e-14
cpa              1.033      0.212     4.87 6.3e-06

n = 76, p = 2, Residual SE = 4.359, R-Squared = 0.24
```

After adjusting for the effect of `csa`, we see size of the effect of `cpa` is reduced from 1.033 to 0.551.

Finally, we should point out that childhood physical abuse might not be the only

factor that is relevant to assessing the effects of childhood sexual abuse. It is quite possible that the two groups differ according to other variables such as socioeconomic status and age. Furthermore, we must regard all causal conclusions from observational studies with the usual sense of caution. See the original article by Rodriguez et al. (1997) for more about this.

14.3 Interpretation with Interaction Terms

A homeowner in England recorded his weekly natural gas consumption, in thousands of cubic feet, during two winter heating seasons. For the second season, cavity wall insulation had been installed. The homeowner also recorded the average weekly temperature in degrees Celsius because this would also affect gas consumption. The data may be found in the MASS package.

```
> data(whiteside, package="MASS")
```

We plot the data in Figure 14.3:

```
> require(ggplot2)
> ggplot(aes(x=Temp, y=Gas), data=whiteside)+geom_point()+facet_grid(~
  Insul)+geom_smooth(method="lm")
```

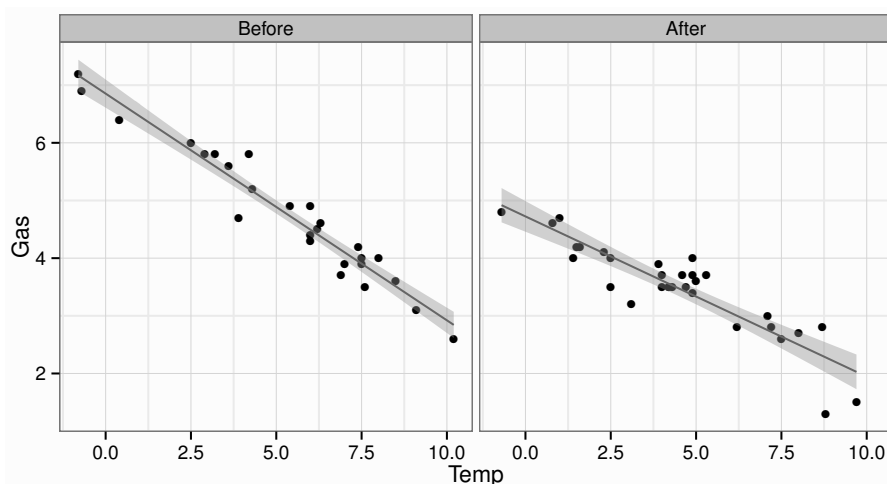


Figure 14.3 Weekly gas consumption as a function of weekly temperature before and after the installation of cavity wall insulation.

We can see that less gas is used after the insulation is installed but the difference varies by temperature. The relationships appear linear so we fit a model:

```
> lmod <- lm(Gas ~ Temp*Insul, whiteside)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.8538	0.1360	50.41	< 2e-16
Temp	-0.3932	0.0225	-17.49	< 2e-16


```

InsulAfter      -2.1300      0.1801     -11.83    2.3e-16
Temp:InsulAfter  0.1153      0.0321      3.59    0.00073
n = 56, p = 4, Residual SE = 0.323, R-Squared = 0.93

```

We would predict that the gas consumption would fall by 0.393 for each 1°C increase in temperature before insulation. After insulation, the fall in consumption per degree is only $0.393 - 0.115 = 0.278$. But the interpretation for the other two parameter estimates is more problematic since these represent predicted consumption when the temperature is zero. This is on the lower edge of the observed range of temperatures and would not represent a typical difference. For other datasets, a continuous predictor value of zero might be far outside the range and so these parameters would have little practical meaning.

The solution is to center the temperature predictor by its mean value and recompute the linear model:

```

> mean(whiteside$Temp)
[1] 4.875
> whiteside$ctemp <- whiteside$Temp - mean(whiteside$Temp)
> lmodc <- lm(Gas ~ ctemp*Insul, whiteside)
> summary(lmodc)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.9368      0.0642   76.85 < 2e-16
ctemp          -0.3932      0.0225  -17.49 < 2e-16
InsulAfter     -1.5679      0.0877  -17.87 < 2e-16
ctemp:InsulAfter 0.1153      0.0321    3.59 0.00073
n = 56, p = 4, Residual SE = 0.323, R-Squared = 0.93

```

Now we can say that the average consumption before insulation at the average temperature was 4.94 and $4.94 - 1.57 = 3.37$ afterwards. The other two coefficients are unchanged and their interpretation remains the same. Thus we can see that centering allows a more natural interpretation of the parameter estimates in the presence of interaction.

14.4 Factors With More Than Two Levels

Suppose we have a factor with f levels, then we create $f - 1$ dummy variables d_2, \dots, d_f where:

$$d_i = \begin{cases} 0 & \text{is not level } i \\ 1 & \text{is level } i \end{cases}$$

Level one is the reference level.

We demonstrate the use of multilevel factors with a study on the sexual activity and the life span of male fruitflies by Partridge and Farquhar (1981): 125 fruitflies were divided randomly into five groups of 25 each. The response was the longevity of the fruitfly in days. One group was kept solitary, while another was kept individually with a virgin female each day. Another group was given eight virgin females per day. As an additional control, the fourth and fifth groups were kept with one or eight pregnant females per day. Pregnant fruitflies will not mate. The thorax length of

each male was measured as this was known to affect longevity. The five groups are labeled isolated, low, high, one and many respectively. The purpose of the analysis is to determine the difference between the five groups if any. We start with a plot of the data, as seen in Figure 14.4.

```
> data(fruitfly, package="faraway")
> plot(longevity ~ thorax, fruitfly, pch=unclass(activity))
> legend(0.63, 100, levels(fruitfly$activity), pch=1:5)
```

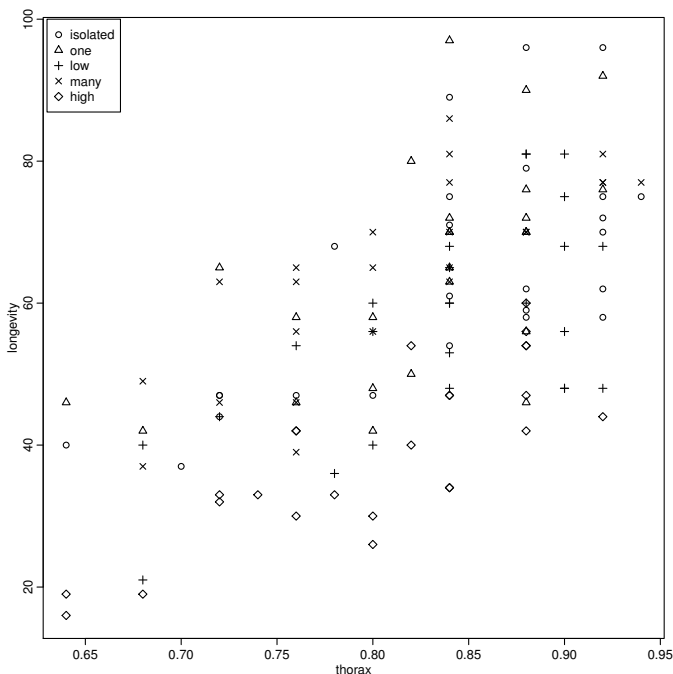


Figure 14.4 Plot of longevity in days and thorax length in millimeters of fruitflies divided into five treatment groups. Longevity for the high sexual activity group appears to be lower.

With multiple levels, it can be hard to distinguish the groups. Sometimes it is better to plot each level separately. This can be achieved nicely with the help of the `ggplot2` package:

```
> require(ggplot2)
> ggplot(aes(x=thorax, y=longevity), data=fruitfly) + geom_point() +
  facet_wrap(~ activity)
```

The plot, shown in Figure 14.5, makes it clearer that longevity for the high sexual activity group is lower.

We fit and summarize the most general linear model:

```
> lmod <- lm(longevity ~ thorax*activity, fruitfly)
```

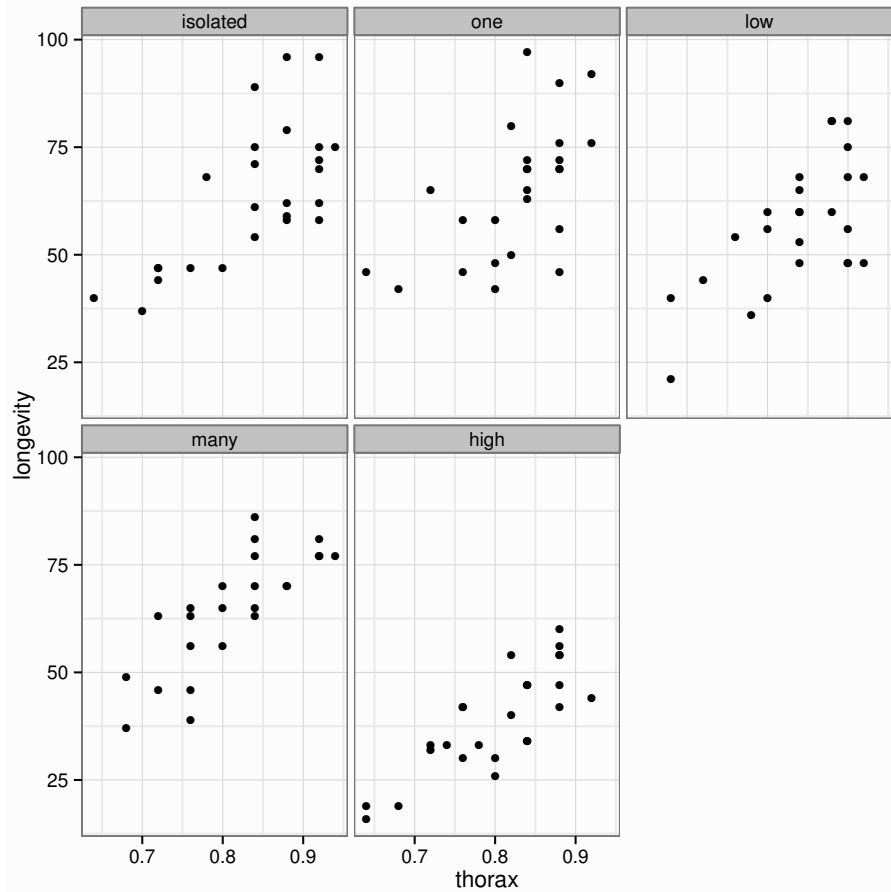


Figure 14.5 Plot of longevity in days and thorax length in millimeters of fruitflies with each treatment group shown in a separate panel.

```
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-50.242	21.801	-2.30	0.023
thorax	136.127	25.952	5.25	7.3e-07
activityone	6.517	33.871	0.19	0.848
activitylow	-7.750	33.969	-0.23	0.820
activitymany	-1.139	32.530	-0.04	0.972
activityhigh	-11.038	31.287	-0.35	0.725
thorax:activityone	-4.677	40.652	-0.12	0.909
thorax:activitylow	0.874	40.425	0.02	0.983
thorax:activitymany	6.548	39.360	0.17	0.868
thorax:activityhigh	-11.127	38.120	-0.29	0.771

```
n = 124, p = 10, Residual SE = 10.713, R-Squared = 0.65
```

Since “isolated” is the reference level, the fitted regression line within this group

is $\text{longevity} = -50.2 + 136.1 * \text{thorax}$. For “many,” it is $\text{longevity} = (-50.2 - 1.1) + (136.1 + 6.5) * \text{thorax}$. Similar calculations can be made for the other groups. Examine:

```
> model.matrix(lmod)
```

to see how the coding is done. Some diagnostics should be examined by:

```
> plot(lmod)
```

There is perhaps some heteroscedasticity, but we will let this be until later for ease of presentation. Now we see whether the model can be simplified. The model summary output is not suitable for this purpose because there are four t-tests corresponding to the interaction term while we want just a single test for this term. We can obtain this using:

```
> anova(lmod)
```

```
Analysis of Variance Table
```

```
Response: longevity
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
thorax	1	15003	15003	130.73	< 2e-16
activity	4	9635	2409	20.99	5.5e-13
thorax:activity	4	24	6	0.05	0.99
Residuals	114	13083	115		

This is a sequential analysis of variance (ANOVA) table. Starting from a null model, terms are added and sequentially tested. The models representing the null and alternatives are listed in Table 14.1. We wish to successively simplify the full model and

Null	Alternative
y~1	y~thorax
y~thorax	y~thorax+activity
y~thorax+activity	y~thorax+activity+thorax:activity

Table 14.1 Models compared in the sequential ANOVA.

then interpret the result. The interaction term `thorax:activity` is not significant, indicating that we can fit the same slope within each group. No further simplification is possible.

We notice that the F-statistic for the test of the interaction term is very small and its p -value close to one. For these data, the fitted regression lines to the five groups happen to be very close to parallel. This can, of course, just happen by chance. In some other cases, unusually large p -values have been used as evidence that data have been tampered with or “cleaned” to improve the fit. Most famously, Ronald Fisher suspected Gregor Mendel of fixing the data in some genetics experiments because the data seemed too good to be true. See Fisher (1936).

We now refit without the interaction term:

```
> lmodp <- lm(longevity ~ thorax+activity, fruitfly)
```

Do we need both `thorax` and `activity`? We could use the output above which

suggests both terms are significant. However, `thorax` is tested by itself and then `activity` is tested once `thorax` is entered into the model. We might prefer to check whether each predictor is significant once the other has been taken into account. We can do this using:

```
> drop1(lmodp, test="F")
Single term deletions

Model:
longevity ~ thorax + activity
      Df Sum of Sq  RSS AIC F value Pr(>F)
<none>                13107 590
thorax    1    12368 25476 670   111.3 <2e-16
activity  4     9635 22742 650    21.7 2e-13
```

The `drop1()` command tests each term relative to the full model. This shows that both terms are significant even after allowing for the effect of the other. Now examine the model coefficients:

```
> summary(lmodp)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -48.75      10.85   -4.49  1.6e-05
thorax        134.34      12.73   10.55 < 2e-16
activityone     2.64       2.98    0.88   0.38
activitylow    -7.01       2.98   -2.35   0.02
activitymany    4.14       3.03    1.37   0.17
activityhigh  -20.00       3.02   -6.63  1.0e-09

n = 124, p = 6, Residual SE = 10.539, R-Squared = 0.65
```

“Isolated” is the reference level. We see that the intercepts of “one” and “many” are not significantly different from this reference level. We also see that the low sexual activity group, “low,” survives about seven days less. The p-value is 0.02 and is enough for statistical significance if only one comparison is made. However, we are making more than one comparison, and so, as with outliers, a Bonferroni-type adjustment might be considered. This would erase the statistical significance of the difference. However, the high sexual activity group, “high,” has a life span 20 days less than the reference group and this is strongly significant.

Returning to the diagnostics:

```
> plot(residuals(lmodp) ~ fitted(lmodp), pch=unclass(fruitfly$activity),
       xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

is seen in the first panel of Figure 14.6. We have some non-constant variance although it does not appear to be related to the five groups. A log transformation can remove the heteroscedasticity:

```
> lmod1 <- lm(log(longevity) ~ thorax+activity, fruitfly)
> plot(residuals(lmod1) ~ fitted(lmod1), pch=unclass(fruitfly$activity),
       , xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

as seen in the second panel of Figure 14.6. One disadvantage of transformation is

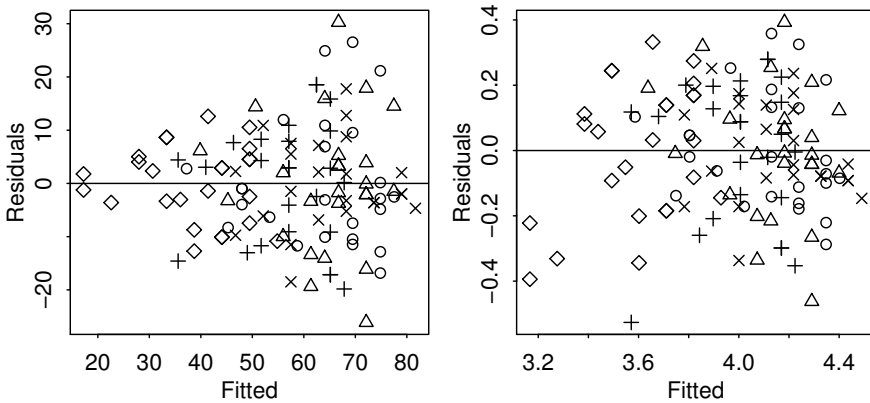


Figure 14.6 Diagnostic plots for the fruitfly data before and after log transformation of the response.

that it can make interpretation of the model more difficult. Let's examine the model fit:

```
> summary(lmod1)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.8442     0.1988    9.28  1.0e-15
thorax        2.7215     0.2333   11.67 < 2e-16
activityone    0.0517     0.0547    0.95  0.346
activitylow   -0.1239     0.0546   -2.27  0.025
activitymany  0.0879     0.0555    1.59  0.116
activityhigh  -0.4193     0.0553   -7.59  8.4e-12

n = 124, p = 6, Residual SE = 0.193, R-Squared = 0.7
```

Notice that the R^2 is higher for this model, but the p-values are similar. Because of the log transformation, we can interpret the coefficients as having a multiplicative effect:

```
> exp(coef(lmod1)[3:6])
activityone activitylow activitymany activityhigh
 1.05311    0.88350    1.09189    0.65754
```

Compared to the reference level, we see that the high sexual activity group has 0.66 times the life span (i.e., 34% less).

Why did we include `thorax` in the model? Its effect on longevity was known, but because of the random assignment of the flies to the groups, this variable will not bias the estimates of the effects of the activities. We can verify that `thorax` is unrelated to the activities:

```
> lmodh <- lm(thorax ~ activity, fruitfly)
> anova(lmodh)
Analysis of Variance Table

Response: thorax
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
activity	4	0.026	0.006	1.11	0.36
Residuals	119	0.685	0.006		

However, look what happens if we omit `thorax` from the model for longevity:

```
> lmodu <- lm(log(longevity) ~ activity, fruitfly)
> summary(lmodu)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.1193	0.0564	72.99	< 2e-16
activityone	0.0234	0.0798	0.29	0.77
activitylow	-0.1195	0.0798	-1.50	0.14
activitymany	0.0240	0.0806	0.30	0.77
activityhigh	-0.5172	0.0798	-6.48	2.2e-09

n = 124, p = 5, Residual SE = 0.282, R-Squared = 0.36

The magnitudes of the effects do not change that much but the standard errors are substantially larger. The value of including `thorax` in this model is to increase the precision of the estimates.

14.5 Alternative Codings of Qualitative Predictors

Sometimes an alternative coding of factor variables can be useful.

Let B be an $n \times k$ dummy variable matrix where $B_{ij} = 1$ if case i falls in class j and is zero otherwise. We might use B to form part of the model matrix. However, the row sums of B are all one. Since an intercept term would also be represented by a column of ones, all the parameters would not be identifiable.

Removing the intercept term is one solution, but this will not work well if there is more than one factor. A more general solution is to reduce the rank of the dummy variable matrix. Simply deleting one column would do, but any solution that avoids collinear dummy variables will work. The choice should be based on convenience of interpretation and numerical stability.

The coding is determined by a *contrast matrix* C which has dimension $k \times (k - 1)$. Some examples of C are given below. The contribution to the model matrix is then given by BC . Other columns of the model matrix might include a column of ones for the intercept and perhaps other predictors.

Treatment coding

Consider a four-level factor that will be coded using three dummy variables. This contrast matrix describes the coding, where the columns represent the dummy variables and the rows represent the levels:

```
> contr.treatment(4)
```

	2	3	4
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1

This treats level one as the standard level to which all other levels are compared so a control group, if one exists, would be appropriate for this level. The parameter for

the dummy variable then represents the difference between the given level and the first level. R assigns levels to a factor in alphabetical order by default. Treatment coding is the default choice for R.

Helmert coding

The coding here is:

```
> contr.helmert(4)
  [,1] [,2] [,3]
1  -1  -1  -1
2   1  -1  -1
3   0   2  -1
4   0   0   3
```

If there are equal numbers of observations in each level (a balanced design) then the dummy variables will be orthogonal to each other and to the intercept. This coding is not as helpful for interpretation except in some special cases.

Sum coding

This coding uses:

```
> contr.sum(4)
  [,1] [,2] [,3]
1   1   0   0
2   0   1   0
3   0   0   1
4  -1  -1  -1
```

This coding might be preferred, particularly when there is no natural reference level. We can illustrate this using a previous example:

```
> contrasts(sexab$csa) <- contr.sum(2)
> summary(lm(ptsd ~ csa, sexab))
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.318      0.405    20.53 < 2e-16
csa1           -3.623      0.405    -8.94 2.2e-13
n = 76, p = 2, Residual SE = 3.473, R-Squared = 0.52
```

The mean response is 8.3, while the “NotAbused” mean is $8.3 - 3.6$ and the “Abused” mean is $8.3 + 3.6$.

There are other choices of coding — anything that spans the $k - 1$ dimensional space will work. The choice of coding does not affect the R^2 , $\hat{\sigma}^2$ and overall F -statistic. It does affect the $\hat{\beta}$ and you do need to know what the coding is before making conclusions about $\hat{\beta}$.

Exercises

1. Using the `teengamb` data, model `gamble` as the response and the other variables as predictors, investigate the possibility of interactions between `sex` and the other predictors. Interpret your final model.

2. Using the `infmort` data, find a simple model for the infant mortality in terms of the other variables. Be alert for transformations and unusual points. Interpret your model by explaining what the regression parameter estimates mean.
3. Plot the `ToothGrowth` data with `len` as the response. Fit a linear model to the data and check for possible simplification. Display the fitted regression lines on top of the data.
4. Investigate whether the `side` variable can be used in a model for all the `chredlin` data rather than as a subsetting variable as in the analysis in the text.
5. Find a good model for the `uswages` data with `wages` as the response and all the other variables as predictors.
6. The dataset `clot` contains the clotting times of blood varying as percentage concentration of prothrombin-free plasma. There are two lots of thromboplastin.
 - (a) Plot the data using a different plotting symbol according to the lot.
 - (b) Find the transformation of the two continuous variables to form a linear relationship.
 - (c) Does the time to clot vary according to concentration differently in the two lots?
 - (d) Check the assumptions of your model using regression diagnostics.
 - (e) At what percentage concentration is the predicted time the same for the two lots?
7. The wealth in billions of dollars for 232 billionaires is given in `fortune`.
 - (a) Plot the wealth as a function of age using a different plotting symbol for the different regions of the world.
 - (b) Plot the wealth as a function of age with a separate panel for each region.
 - (c) Determine a transformation on the response to facilitate linear modeling.
 - (d) What is the relationship of age and region to wealth?
 - (e) Check the assumptions of your model using appropriate diagnostics.
8. Ankylosing spondylitis is a chronic form of arthritis. A study was conducted to determine whether daily stretching of the hip tissues would improve mobility. The data are found in `hips`. The flexion angle of the hip before the study is a predictor and the flexion angle after the study is the response.
 - (a) Plot the data using different plotting symbols for the treatment and the control status.
 - (b) Fit a model to determine whether there is a treatment effect.
 - (c) Compute the difference between the flexion before and after and test whether this difference varies between treatment and control. Contrast this approach to your previous model.
 - (d) Check for outliers. Explain why we might remove the three cases with `fbefore` less than 90. Refit an appropriate model and check for a treatment effect.
 - (e) What is the estimated size of the treatment effect? Give a 95% confidence interval.

- (f) Both legs of each subject have been included in the study as separate observations. Explain what difficulties this causes with the model assumptions.
- (g) Compute the average angles for each subject and repeat the modeling with this reduced dataset. Point out any differences in the conclusions.

One Factor Models

Linear models with only categorical predictors (or factors) have traditionally been called analysis of variance (ANOVA) problems. The idea is to partition the overall variance in the response due to each of the factors and the error. This traditional approach is exemplified by Scheffé (1959).

Although some insight can be gained by considering the problem from this perspective, the price is an increasingly complex set of specialized formulae required for each type of model depending on the configuration of factors. Instead, we take a regression-based approach by putting the model into the $y = X\beta + \varepsilon$ format and then using the inferential methods we have already developed.

The terminology used in ANOVA-type problems is sometimes different. Predictors are now all qualitative and are now typically called *factors*, which have some number of *levels*. The regression parameters are now often called *effects*. We shall consider only models where the parameters are considered fixed, but unknown — called *fixed-effects* models. *Random-effects* models are used where parameters are taken to be random variables and are not covered in this text.

15.1 The Model

Suppose we have a factor α occurring at $i = 1, \dots, I$ levels, with $j = 1, \dots, J_i$ observations per level. We use the model:

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

The parameters are not identifiable. For example, we could add some constant to μ and subtract the same constant from each α_i and the fit would be unchanged. Some restriction is necessary. Here are some possibilities:

1. Drop μ from the model and use I different dummy variables to estimate α_i for $i = 1, \dots, I$. This is feasible but does not extend well to models with more than one factor as more than one parameter needs to be dropped.
2. Set $\alpha_1 = 0$, then μ represents the expected mean response for the first level and α_i for $i \neq 1$ represents the difference between level i and level one. Level one is then called the *reference level* or *baseline level*. This can be achieved using treatment contrasts as discussed in the previous chapter.
3. Set $\sum_i \alpha_i = 0$, now μ represents the mean response over all levels and α_i , the difference from that mean. This requires the use of sum contrasts.

The choice of constraint from those listed above or otherwise will determine the coding used to generate the X -matrix. Once that is done, the parameters (effects) can be estimated in the usual way along with standard errors. No matter which valid constraint and coding choice is made, the fitted values and residuals will be the same.

Once the effects are estimated, the natural first step is to test for differences in the levels of the factor. An explicit statement of the null and alternative hypotheses would depend on the coding used. If we use the treatment coding with a reference level, then the null hypothesis would require that $\alpha_2 = \dots = \alpha_I = 0$. For other codings, the statement would differ. It is simpler to state the hypotheses in terms of models:

$$H_0: y_{ij} = \mu + \varepsilon_{ij}$$

$$H_1: y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

We compute the residual sum of squares and degrees of freedom for the two models and then use the same F-test as we have used for regression. The outcome of this test will be the same no matter what coding/restriction we use. If we do not reject the null, we are almost done — we must still check for a possible transformation of the response and outliers. If we reject the null, we must investigate which levels differ.

15.2 An Example

Twenty-four animals were randomly assigned to four different diets and blood samples were taken in a random order. The blood coagulation time was measured. These data come from Box et al. (1978):

```
> data(coagulation, package="faraway")
> coagulation
  coag diet
1    62   A
2    60   A
...etc...
23   63   D
24   59   D
```

Some preliminary graphical analysis is essential before fitting. The default plotting method in base R graphics is the side-by-side boxplot. Although this plot has strong visual impact for comparison, a stripchart can be better for smaller datasets:

```
> plot(coag ~ diet, coagulation, ylab="coagulation time")
> stripchart(coag ~ diet, coagulation, vertical=TRUE, method="stack",
  xlab="diet", ylab="coagulation time")
```

The boxplot in Figure 15.1 shows how the four levels vary but there is something odd about the display of diet C because the median and upper quartile are the same. The stripchart shows the ties in the data in diets C and D.

We are looking for several features in these plots. We must check for equality of variance in the groups, which seems satisfied in this example. We are looking for evidence of skewness showing a lack of normality. This might suggest a transformation of the response. There is no such concern in this example. Finally, we should look for outliers — there are none to be seen in this example.

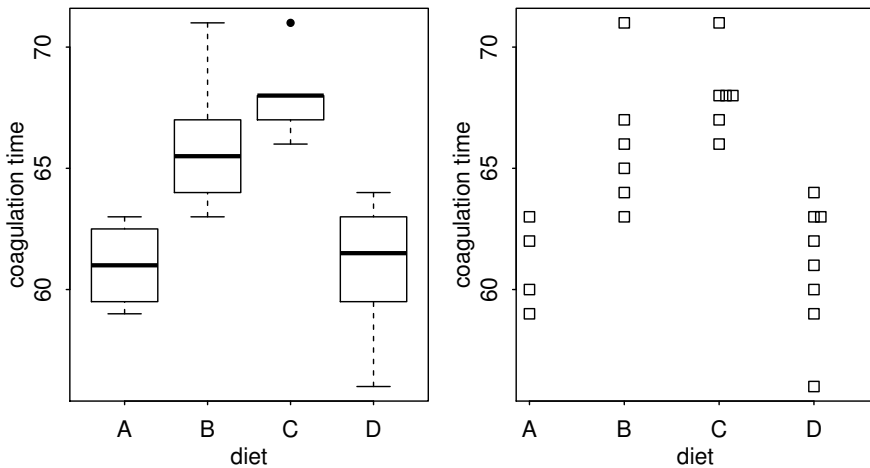


Figure 15.1 A boxplot on the left and a stripchart on the right showing the blood coagulation data.

It is not worth considering diagnostics for influence, as the leverages depend explicitly on J_i . If there is only one observation for a given level, that is $J_i = 1$, then the estimate of the effect for that level will be based on that single point. That point is clearly influential without further investigation.

Now let's fit the model using the default treatment coding:

```
> lmod <- lm(coag ~ diet, coagulation)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.10e+01  1.18e+00  51.55 < 2e-16
dietB        5.00e+00  1.53e+00   3.27 0.00380
dietC        7.00e+00  1.53e+00   4.58 0.00018
dietD        2.99e-15  1.45e+00   0.00 1.00000

n = 24, p = 4, Residual SE = 2.366, R-Squared = 0.67
```

Rounding error results in a rather unfortunate formatting of the results. The coefficients can be more cleanly seen as:

```
> round(coef(lmod), 1)
(Intercept)      dietB      dietC      dietD
           61           5           7           0
```

Group A is the reference level and has a mean of 61, groups B, C and D are 5, 7 and 0 seconds larger, respectively, on average. Examine the design matrix to understand the coding:

```
> model.matrix(lmod)
```

The three test statistics for the group levels correspond to comparisons with the reference level A. Although these are interesting, they do not specifically answer the

question of whether there is a significant difference between any of the levels. This test can be obtained as:

```
> anova(lmod)
Analysis of Variance Table

Response: coag
      Df Sum Sq Mean Sq F value    Pr(>F)
diet     3    228   76.0    13.6 0.000047
Residuals 20    112    5.6
```

We see that there is indeed a difference in the levels although this test does not tell us which levels are different from others.

We can fit the model without an intercept term as in:

```
> lmodi <- lm(coag ~ diet -1, coagulation)
> summary(lmodi)
      Estimate Std. Error t value Pr(>|t|)
dietA    61.000     1.183   51.5 <2e-16
dietB    66.000     0.966   68.3 <2e-16
dietC    68.000     0.966   70.4 <2e-16
dietD    61.000     0.837   72.9 <2e-16

n = 24, p = 4, Residual SE = 2.366, R-Squared = 1
```

We can directly read the level means. The R^2 is not correctly calculated because of the absence of an intercept. To generate the usual test that the means of the levels are equal, we would need to fit the null model and compare using an F-test:

```
> lmnnull <- lm(coag ~ 1, coagulation)
> anova(lmnnull, lmodi)
Analysis of Variance Table

Model 1: coag ~ 1
Model 2: coag ~ diet - 1
  Res.Df RSS Df Sum of Sq    F    Pr(>F)
1      23 340
2      20 112  3      228 13.6 0.000047
```

We get the same F-statistic and p-value as in the first coding.

We can also use a sum coding:

```
> options(contrasts=c("contr.sum", "contr.poly"))
> lmods <- lm(coag ~ diet , coagulation)
> summary(lmods)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)   64.000     0.498  128.54 < 2e-16
diet1         -3.000     0.974   -3.08  0.00589
diet2          2.000     0.845    2.37  0.02819
diet3          4.000     0.845    4.73  0.00013

n = 24, p = 4, Residual SE = 2.366, R-Squared = 0.67
```

So the estimated overall mean response is 64 while the estimated mean response for A is three less than the overall mean, that is, 61. Similarly, the means for B and C are 66 and 68, respectively. Since we are using the sum constraint, we compute

$\hat{\alpha}_D = -(-3 + 2 + 4) = -3$ so the mean for D is $64 - 3 = 61$. Notice that $\hat{\sigma}$ and the R^2 are the same as before.

So we can use any of these three methods and obtain essentially the same results. Dropping the intercept is least convenient since an extra step is needed to generate the F-test. Furthermore, the approach would not extend well to experiments with more than one factor, as additional constraints would be needed. The other two methods can be used according to taste. The treatment coding is most appropriate when the reference level is set to a possible control group. We will use the treatment coding by default, as does R, for the rest of this book.

15.3 Diagnostics

There are fewer diagnostics to do for ANOVA models, but it is still important to plot the residuals and fitted values and to make the Q-Q plot of the residuals. It makes no sense to transform the predictor, but it is reasonable to consider transforming the response. Diagnostics are shown in Figure 15.2.

```

> qqnorm(residuals(lmod))
> qqline(residuals(lmod))
> plot(jitter(fitted(lmod)), residuals(lmod), xlab="Fitted", ylab="
  Residuals")
> abline(h=0)

```

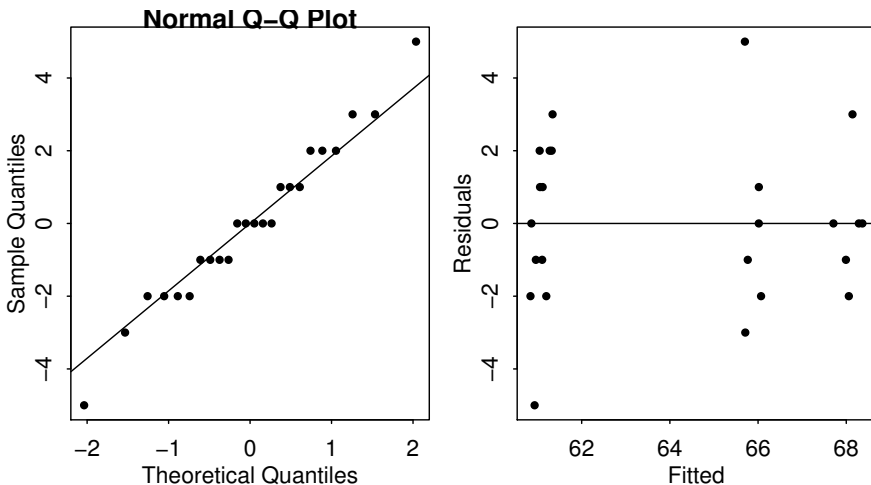


Figure 15.2 Diagnostics for the blood coagulation model.

Because the data are integers and the fitted values happen to be integers also, some discreteness is obvious in the Q-Q plot. Of course, discrete data cannot be normally distributed. However, here the residuals are approximately normal and so we can go ahead with the inference without much concern. The discreteness in the residuals and fitted values shows up in the residual-fitted plot. We have jittered the points so that they can be seen separately.

The assumption of homogeneity of the error variance can be examined using Levene's test. It computes the absolute values of the residuals and uses these as the response in a new one-way ANOVA. To reduce the possible influence of outliers, group medians rather than means should be used. A significant difference would indicate non-constant variance.

Most tests and confidence intervals (CIs) are relatively insensitive to non-constant variance so there is no need to take action unless the Levene test is significant at the 1% level.

Applying this to the diet data, we find:

```
> med <- with(coagulation, tapply(coag, diet, median))
> ar <- with(coagulation, abs(coag - med[diet]))
> anova(lm(ar ~ diet, coagulation))
Analysis of Variance Table

Response: ar
      Df Sum Sq Mean Sq F value Pr(>F)
diet    3    4.3    1.4    0.65  0.59
Residuals 20  44.5    2.2
```

Since the p-value is large, we conclude that there is no evidence of a non-constant variance.

An alternative test is due to Bartlett:

```
> bartlett.test(coag ~ diet, coagulation)

      Bartlett test of homogeneity of variances

data:  coag by diet
Bartlett's K-squared = 1.668, df = 3, p-value = 0.6441
```

Again, no difference is found. Levene's test is more robust to outliers.

15.4 Pairwise Comparisons

After detecting some difference in the levels of the factor, interest centers on which levels or combinations of levels are different. It does not make sense to ask whether a particular level is significant since this begs the question of "significantly different from what?" Any meaningful test must involve a comparison of some kind.

A pairwise comparison of level i and j can be made using a CI for $\alpha_i - \alpha_j$ using:

$$\hat{\alpha}_i - \hat{\alpha}_j \pm t_{df}^{\alpha/2} se(\hat{\alpha}_i - \hat{\alpha}_j)$$

where $se(\hat{\alpha}_i - \hat{\alpha}_j) = \hat{\sigma} \sqrt{1/J_i + 1/J_j}$ and $df = n - I$ in this case. A test for $\alpha_i = \alpha_j$ amounts to seeing whether zero lies in this interval or not. For example, let's find a 95% CI for $B - A$. From the model output, we can see that the difference is 5.0 with a standard error of 1.53. For differences not involving the reference level of A, more effort would be required to calculate these values. The interval is:

```
> 5 + c(-1, 1) * qt(0.975, 24-4) * 1.53
[1] 1.8085 8.1915
```


Since zero is not in the interval, the difference is significant. This is fine for just one test, but we are likely to be interested in more than one comparison. Suppose we do all possible pairwise tests when $\alpha = 5\%$ and the null hypothesis is in fact true. In the blood coagulation data, there are four levels and so six possible pairwise comparisons. Even if there was no difference between the four levels, there is still about a 20% chance that at least one significant difference will be found.

For experiments with more levels, the true type I error gets even higher. Using the t-based CIs for multiple comparisons is called the least significant difference (LSD) method, but it can hardly be recommended. Now one might be tempted to argue that we could choose which comparisons are interesting and so reduce the amount of testing and thus the magnitude of the problem. If we only did a few tests, then the Bonferroni adjustment (see Section 6.2.2) could be used to make a simple correction. However, the determination of which comparisons are “interesting” is usually made after seeing the fitted model. This means that all other comparisons are implicitly made even if they are not explicitly computed. On the other hand, if it can be argued that the comparisons were decided before seeing the fit, then we could make the case for the simple adjustment. However, this is rarely the case and furthermore it might be difficult to convince others that this really was your intention. We must usually find a way to adjust for *all* pairwise comparisons.

There are many ways to make the adjustment, but *Tukey’s honest significant difference (HSD)* is the easiest to understand. It depends on the studentized range distribution which arises as follows. Let X_1, \dots, X_n be i.i.d. $N(\mu, \sigma^2)$ and let $R = \max_i X_i - \min_i X_i$ be the range. Then $R/\hat{\sigma}$ has the studentized range distribution $q_{n,v}$ where v is the number of degrees of freedom used in estimating σ .

The Tukey CIs are:

$$\hat{\alpha}_i - \hat{\alpha}_j \pm \frac{q_{l,df}}{\sqrt{2}} \hat{\sigma} (1/J_i + 1/J_j)$$

When the level sample sizes J_i are very unequal, Tukey’s HSD test may become too conservative. We compute the Tukey HSD bands for the B-A difference. The critical value from the studentized range distribution is obtained from the `qtukey()` function:

```
> se <- summary(lmod)$sigma
> 5 + c(-1,1) * qtukey(0.95, 4, 24-4) / sqrt(2) * se * sqrt(1/4 + 1/6)
[1] 0.72455 9.27545
```

A convenient way to obtain all the intervals is:

```
> (tci <- TukeyHSD(aov(coag ~ diet, coagulation)))
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = coag ~ diet, data = coagulation)

$diet
      diff      lwr      upr    p adj
B-A      5  0.72455  9.2754 0.01833
C-A      7  2.72455 11.2754 0.00096
```

D-A	0	-4.05604	4.0560	1.00000
C-B	2	-1.82407	5.8241	0.47660
D-B	-5	-8.57709	-1.4229	0.00441
D-C	-7	-10.57709	-3.4229	0.00013

We find that only the $A - D$ and $B - C$ differences are not significant as the corresponding intervals contain zero. The duality between confidence intervals and hypothesis testing can also be used to produce an adjusted p -value as seen in the final column of the output. This merely confirms our impression of which differences are significant. The intervals can be plotted as seen in Figure 15.3.

95% family-wise confidence level

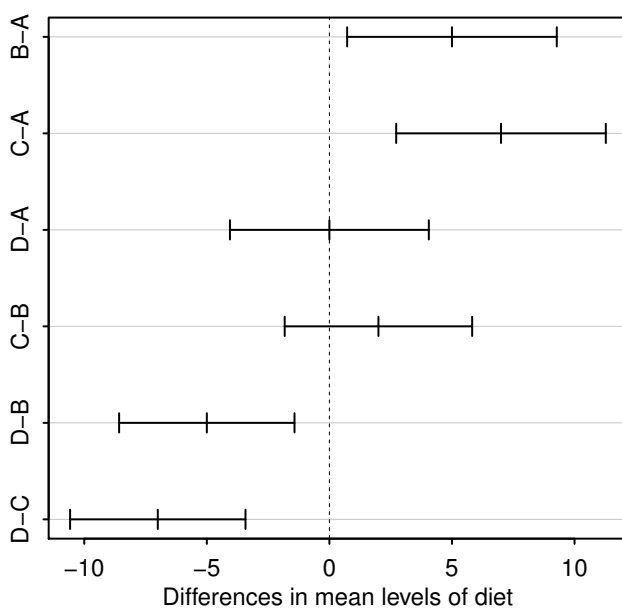


Figure 15.3 Tukey HSD 95% confidence intervals for the pairwise differences.

```
> plot(tci)
```

The Tukey method assumes the worst by focusing on the largest difference. There are other competitors like the Newman-Keuls, Duncan's multiple range and the Waller-Duncan procedure, which are less pessimistic or do not consider all possible pairwise comparisons. For a detailed description of the many available alternatives see Bretz, Hothorn, and Westfall (2010) and the R package `multcomp`.

15.5 False Discovery Rate

Consider some data taken from the Junior School Project collected from primary (US term is elementary) schools in inner London. The data is described in detail in

Mortimore et al. (1988). We focus on just two of the variables in the data — the `school`, of which there are 49, and the mathematics test scores for students from these schools. Suppose we are interested in deviations from the average and so we center the scores:

```
> data(jsp, package="faraway")
> jsp$mathcent <- jsp$math - mean(jsp$math)
```

A more pleasing plot of the data can be obtained using the `ggplot2` package as seen in Figure 15.4. We have to rotate the school labels so they can be distinguished.

```
> require(ggplot2)
> ggplot(aes(x=school,y=mathcent), data=jsp)+geom_boxplot() + theme(
  axis.text.x = element_text(angle = 90))+ylab("Centered Math
  Scores")
```

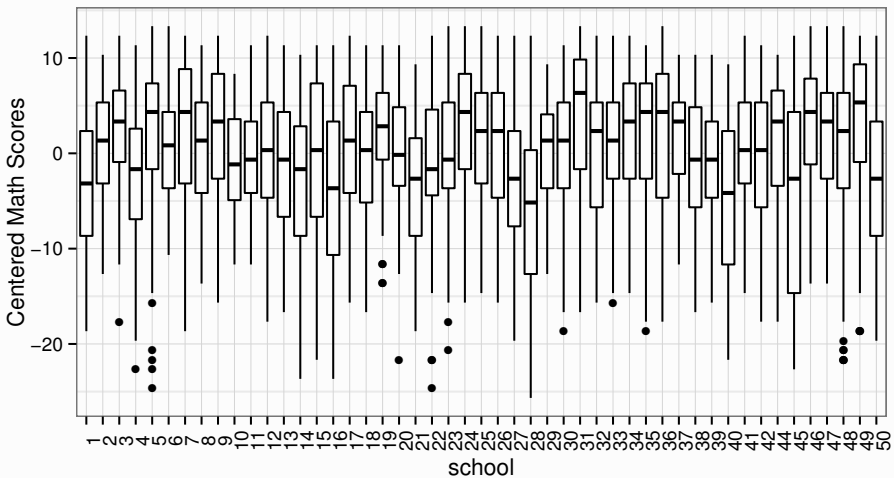


Figure 15.4 Variation-centered math scores by school.

Let's choose the parameterization that omits the intercept term:

```
> lmod <- lm(mathcent ~ school-1, jsp)
> sumary(lmod)
      Estimate Std. Error t value Pr(>|t|)
school1  -3.3685    0.7686  -4.38  1.2e-05
school2   0.6714    1.2287   0.55  0.58481
...
school50  -2.6520    0.7336  -3.62  0.00030

n = 3236, p = 49, Residual SE = 7.372, R-Squared = 0.08
```

Since we have centered the response, the t -tests, which check for differences from zero, are meaningful. We can see there is good evidence that schools 1 and 50 are significantly below average while the evidence that school 2 is above average is not statistically significant. We can test for a difference between the schools:

```
> anova(lm(mathcent ~ school, jsp))
Analysis of Variance Table

Response: mathcent
          Df Sum Sq Mean Sq F value Pr(>F)
school    48  15484      323    5.94 <2e-16
Residuals 3187 173212      54
```

We find a strongly significant difference. This comes as little surprise as the sample size is large, giving us the power to detect quite small differences. Furthermore, we may have strong prior reasons to expect differences between the schools. A more interesting question is which schools show clear evidence of under- or over-performance?

There are too many pairwise comparisons on which to focus our interest. Instead let us ask which schools have means significantly different from the average. The parameterization we have chosen makes these comparisons easy but we would expect about 5% of these differences to be significant even if the null hypothesis held.

Some adjustment is necessary. One approach is to control the *familywise error rate* (FWER) which is the overall probability of falsely declaring a difference (where none exists). The *Bonferroni correction* is a simple way to do this. We just multiply the unadjusted p -values by the number of comparisons. Any probability computed above one is truncated to one. Let's see which schools have adjusted p -values less than 5%:

```
> pvals <- summary(lmod)$coef[,4]
> padj <- p.adjust(pvals, method="bonferroni")
> coef(lmod)[padj < 0.05]
school11 school116 school21 school28 school31 school40 school45
-3.3685   -3.7374   -3.6185   -5.8286    3.8241   -4.9855   -4.6392
school50
-2.6520
```

We see that eight schools are identified with all except school 31 marked as significantly below average.

The Bonferroni correction is known to be conservative but even were we to use one of the more generous alternatives, the familywise error rate restriction imposes a high bar on the identification of significant effects. As the number of levels being compared increases, this requirement becomes ever more stringent.

An alternative approach is to control the *false discovery rate* (FDR) which is the proportion of effects identified as significant which are not real. The best known method of doing this is due to Benjamini and Hochberg (1995).

Given sorted p -values $p_{(i)}$ for $i = 1, \dots, m$ the procedure finds the largest i for which $p_{(i)} \leq \alpha_i/m$. All tests corresponding to $p_{(i)}$ up to and including this i are declared significant. We compute this for our example:

```
> names(which(sort(pvals) < (1:49)*0.05/49))
[1] "school28" "school31" "school21" "school11" "school45" "school40"
[7] "school116" "school50" "school47" "school49" "school14" "school36"
[13] "school46" "school14" "school24" "school27" "school34" "school9"
```

We see that 18 schools are identified compared to the 8 by the previous procedure. FDR is less stringent than FWER in identifying significant effects. A more convenient method of computing the adjusted p -values is:

```
> padj <- p.adjust(pvals, method="fdr")
> coef(lmod)[padj < 0.05]
```

school11	school14	school19	school114	school116	school121	school124
-3.3685	-2.6619	2.2458	-3.2898	-3.7374	-3.6185	2.6238
school127	school128	school131	school134	school136	school140	school145
-2.3058	-5.8286	3.8241	1.9359	2.3605	-4.9855	-4.6392
school146	school147	school149	school150			
3.0636	2.3969	2.7964	-2.6520			

FDR methods are more commonly used where large numbers of comparisons are necessary as often found in imaging or bioinformatics applications. In examples such as these, we expect to find some significant effects and FDR is a useful tool in reliably identifying them.

Exercises

- Using the `pulp` data, determine whether there are any differences between the operators. What is the nature of these differences?
- Determine whether there are differences in the weights of chickens according to their feed in the `chickwts` data. Perform all necessary model diagnostics.
- Using the `PlantGrowth` data, determine whether there are any differences between the groups. What is the nature of these differences? Test for a difference between the average of the two treatments and the control.
- Using the `infmort` data, perform a one-way ANOVA with `income` as the response and `region` as the predictor. Which pairs of regions are different? Now check for a good transformation on the response and repeat the comparison.
- The `anaesthetic` data provides the time to restart breathing unassisted in recovering from general anaesthetic for four treatment groups.
 - Produce a boxplot depicting the data. Comment on any features of interest.
 - Make an appropriate stripchart of the data.
 - Produce versions of the previous two plots using the `ggplot2` package. Show how you can overlay these two plots.
 - Fit a one-factor model for the recovery times and test for a difference between the two groups.
 - Try the Box-Cox transformation method. Explain what went wrong.
 - Try a square root transformation on the response. Are the diagnostics satisfactory? Is there a significant difference among the treatment groups?
- Data on the butterfat content of milk from Canadian cows of five different breeds can be found in the `butterfat` dataset. Consider only mature cows.
 - Plot the data and interpret what you see.
 - Test for a difference between the breeds.

- (c) Check the diagnostics for your model.
 - (d) Produce a plot to check for differences in butterfat between breeds. Which pairwise differences are not statistically significant?
7. Five suppliers cut denim for a jeans manufacturer. The amount of waste relative to a target was collected weekly as seen in the `denim` dataset.
- (a) Plot the data to determine which supplier wastes the least. Which supplier is best in terms in minimizing maximum weekly waste?
 - (b) Is there a significant difference in wastage between the suppliers?
 - (c) Check the regression diagnostics commenting on any violations.
 - (d) Remove two outliers and repeat the test for a significant difference. Which supplier has the lowest predicted wastage under this model?
 - (e) Check for significant pairwise differences between suppliers. Which pairs are significantly different?
 - (f) Which supplier would you pick if there was no other relevant difference between them? What if the cost of the suppliers was in numerical order with the first the most expensive and five the cheapest?

Models with Several Factors

In this chapter, we show how to model data with more than one categorical predictor. Sometimes the data can arise from observational studies but such data more commonly arises from designed experiments, often called factorial designs. If all possible combinations of the levels of the factors occur at least once, then we have a full factorial design. Repeated observations for the same combination of factor levels are called *replicates*.

We start with models involving two factors with no replication. It is possible that the factors can interact but this is difficult to investigate without replication. We consider examples with and without significant interaction and discuss how they should be interpreted. Replication can be expensive so sometimes it is better to use the experimental resources to investigate more factors. This leads us to an example with many factors but no replication.

16.1 Two Factors with No Replication

Mazumdar and Hoa (1995) report an experiment to test the strength of a thermoplastic composite depending on the power of a laser and the speed of a tape:

```
> data(composite, package="faraway")
> composite
  strength laser  tape
1   25.66   40W  slow
2   29.15   50W  slow
3   35.73   60W  slow
4   28.00   40W medium
5   35.09   50W medium
6   39.56   60W medium
7   20.65   40W  fast
8   29.79   50W  fast
9   35.66   60W  fast
```

The data has been plotted in Figure 16.1 with roles of the two factors shown in both ways to give better insight.

```
> require(ggplot2)
> ggplot(composite, aes(x=laser, y=strength, group=tape, linetype=tape)) +
  geom_line() + theme(legend.position = "top", legend.direction = "
  horizontal")
> ggplot(composite, aes(x=tape, y=strength, group=laser, linetype=laser)) +
  geom_line() + theme(legend.position = "top", legend.direction = "
  horizontal")
```

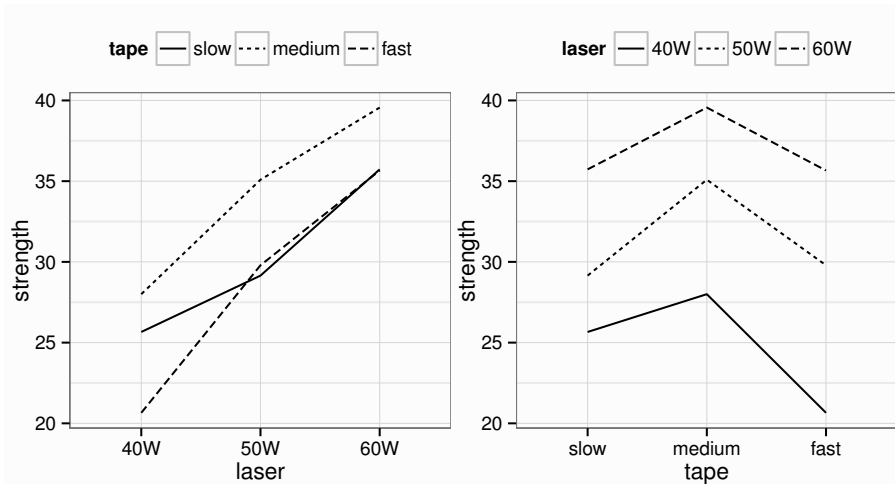


Figure 16.1 Plots for the composite data — the role of the predictors is reversed in each plot.

We can see in the first plot that medium tape speed is best for strength while in the second we see that the 60W setting for laser is best. When there is little or no interaction effect, we would expect the lines in each plot to be approximately parallel (i.e., a constant distance apart). This would be a sign that the effects are additive. Of course, we must allow for a certain amount of noise clouding the observation, but in this example, the lines do appear to be roughly parallel.

A general model for this type of data is:

$$y_{ij} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ij}$$

where $i = 1, \dots, I$ and $j = 1, \dots, J$. The $\alpha\beta$ terms represent interactions between the levels of α and β .

We fit this model to the current example:

```
> lmod <- lm(strength ~ laser + tape + laser:tape, composite)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	25.66	NA	NA	NA
laser50W	3.49	NA	NA	NA
laser60W	10.07	NA	NA	NA
tapemedium	2.34	NA	NA	NA
tapefast	-5.01	NA	NA	NA
laser50W:tapemedium	3.60	NA	NA	NA
laser60W:tapemedium	1.49	NA	NA	NA
laser50W:tapefast	5.65	NA	NA	NA
laser60W:tapefast	4.94	NA	NA	NA

n = 9, p = 9, Residual SE = NaN, R-Squared = 1

We have parameter estimates but nothing else. The model has as many parameters as cases so a perfect fit has been achieved. Unfortunately, no further inference is possible and little has been gained from this model.

We cannot check for an interaction directly by adding such a term to the model but Tukey's nonadditivity test provides another way of investigating for an interaction. The model:

$$y_{ij} = \mu + \alpha_i + \beta_j + \phi\alpha_i\beta_j + \varepsilon_{ijk}$$

is fit to the data and then we test if $\phi = 0$. This is a nonlinear model because it involves a product of parameters, $\phi\alpha_i\beta_j$ which is difficult to fit in general. We get around this problem by fitting the model in stages. First we fit the model with no interaction and extract the estimates of the main effects, noting that the first level is the reference level:

```
> lmod <- lm(strength ~ laser + tape, composite)
> coef(lmod)
(Intercept)    laser50W    laser60W    tapemedium    tapefast
 23.9178      6.5733      12.2133      4.0367      -1.4800
> lasercoefs <- rep(c(0, 6.5733, 12.2133), 3)
> tapecoefs <- rep(c(0, 4.0367, -1.4800), each=3)
```

Now we update the model with the new predictor formed from the products of the main effects:

```
> tmod <- update(lmod, . ~ . + I(lasercoefs*tapecoefs))
> anova(tmod)
Analysis of Variance Table

Response: strength
              Df Sum Sq Mean Sq F value Pr(>F)
laser         2  224.2   112.1   36.82 0.0077
tape          2   48.9    24.5    8.03 0.0624
I(lasercoefs * tapecoefs) 1    1.4    1.4    0.45 0.5503
Residuals    3    9.1    3.0
```

The p -value of 0.55 indicates a nonsignificant interaction. So for these data, we might reasonably assume $(\alpha\beta)_{ij} = 0$. The major drawback is that the test makes the assumption that the interaction effect is multiplicative in form. We have no particular reason to believe it takes this form and so this alternative hypothesis may not be looking in the right place. Although judging whether lines on a plot are parallel requires some subjective judgment, it may be a more reliable method of checking for interaction here.

Now that the issue of interactions has been addressed, we can check the significance of the main effects:

```
> anova(lmod)
Analysis of Variance Table

Response: strength
              Df Sum Sq Mean Sq F value Pr(>F)
laser         2  224.2   112.1   42.69 0.002
tape          2   48.9    24.5    9.32 0.031
Residuals    4   10.5     2.6
```

We see that both factors are significant.

The treatment coding does not take advantage of the ordered nature of both factors. Factors without an ordering to the levels are called *nominal* while those that possess a natural ordering are called *ordinal*. We can declare both to be *ordered factors* and refit:

```
> composite$laser <- as.ordered(composite$laser)
> composite$tape <- as.ordered(composite$tape)
> lmod <- lm(strength ~ laser + tape, composite)
> summary(lmod)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	31.032	0.540	57.45	5.5e-07
laser.L	8.636	0.936	9.23	0.00077
laser.Q	-0.381	0.936	-0.41	0.70466
tape.L	-1.047	0.936	-1.12	0.32594
tape.Q	-3.900	0.936	-4.17	0.01404

n = 9, p = 5, Residual SE = 1.620, R-Squared = 0.96

Instead of a coding with respect to a reference level, we have linear and quadratic terms for each factor. The coding is:

```
> round(model.matrix(lmod), 2)
```

	(Intercept)	laser.L	laser.Q	tape.L	tape.Q
1	1	-0.71	0.41	-0.71	0.41
2	1	0.00	-0.82	-0.71	0.41
3	1	0.71	0.41	-0.71	0.41
4	1	-0.71	0.41	0.00	-0.82
5	1	0.00	-0.82	0.00	-0.82
6	1	0.71	0.41	0.00	-0.82
7	1	-0.71	0.41	0.71	0.41
8	1	0.00	-0.82	0.71	0.41
9	1	0.71	0.41	0.71	0.41

We see the linear term is proportional to $(-1, 0, 1)$ representing a linear trend across the levels while the quadratic term is proportional to $(1, -2, 1)$ representing a quadratic trend.

We see that the quadratic term for laser power is not significant while there is a quadratic effect for tape speed. One of the drawbacks of a model with factors is the difficulty of extrapolating to new conditions. The information gained from the ordered factors suggests a model with numerical predictors corresponding to the level values (the values for the tape speed are obtained from the R help page for composite).

```
> composite$Ntape <- rep(c(6.42, 13, 27), each=3)
> composite$Nlaser <- rep(c(40, 50, 60), 3)
> lmodn <- lm(strength ~ Nlaser + poly(log(Ntape), 2), composite)
> summary(lmodn)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.4989	3.0592	0.16	0.87684
Nlaser	0.6107	0.0604	10.11	0.00016
poly(log(Ntape), 2)1	-1.8814	1.4791	-1.27	0.25933
poly(log(Ntape), 2)2	-6.7364	1.4791	-4.55	0.00609

n = 9, p = 4, Residual SE = 1.479, R-Squared = 0.96

We use the log of tape speed, as this results in roughly evenly spaced levels. This model fits about as well as the two-factor model but has the advantage that we make predictions for values of tape speed and laser power that were not used in the experiment. The earlier analysis with factors alone helped us discover this model, which we may not otherwise have found.

16.2 Two Factors with Replication

Consider the case when the number of observations per combination of factor levels is the same and greater than one. Such a layout results in an orthogonal design matrix. With the benefit of replication, we are now free to fit and test the full model:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

The interaction effect is tested by fitting a model without the $(\alpha\beta)_{ij}$ term and computing the usual F-test. If the interaction effect is found to be significant, we do not test the main effects even if they appear not to be significant.

In an experiment to study factors affecting the production of the polyvinyl chloride (PVC) plastic, three operators used eight different devices called resin railcars to produce PVC. For each of the 24 combinations, two samples were produced. The response is the particle size of the product. The experiment is described in Morris and Watson (1998).

We make plots with respect to each of the predictors, as seen in Figure 16.2:

```
> data(pvc, package="faraway")
> require(ggplot2)
> p <- ggplot(pvc, aes(x=operator, y=psize))+geom_point()+stat_summary(
  fun.y="mean", geom="line", aes(group=resin))
> oplmeans <- with(pvc[pvc$operator == 1,], sapply(split(psize, resin),
  mean))
> tdf <- data.frame(x=rep(0.9, 8), y=oplmeans, label=1:8)
> p + geom_text(data=tdf, aes(x=x, y=y, label=label))
> ggplot(pvc, aes(x=resin, y=psize, shape=operator))+geom_point()+stat_
  summary(fun.y="mean", geom="line", aes(group=operator, linetype=
  operator))+theme(legend.position = "top", legend.direction = "
  horizontal")
```

In the first plot, we have drawn lines connecting the mean values for each level of the resin. We can see from this plot that particle sizes tend to be largest for the first operator and smallest for the third. We see that the variance of the response is about the same for each operator. This is convenient since equality of the variance is one assumption of the linear model. We have annotated the plot to label the eight levels so that they can be readily distinguished. Note that it is difficult to distinguish eight different plotting symbols or lines so we have not attempted this. We can see that the lines are approximately parallel which leads us to expect that there is little interaction between the two plots.

In the second plot, there are eight levels of resin. For each level, we observe two responses from each of the three operators. Since three levels are easier to distinguish, we have plotted these with a separate symbol. As with the first plot, lines join

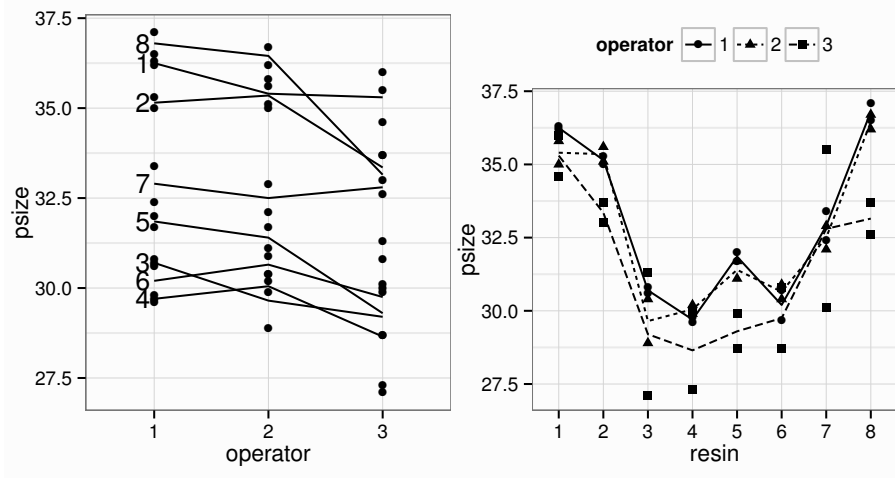


Figure 16.2 PVC data plotted by operators on the left and by resin railcars on the right. Lines connect the average responses for the levels. The railcar number appears to the left of each line on the first plot.

the averages for each level. We can see varying response by resin. The variances within the levels seem approximately equal which is again reassuring. Furthermore, the three lines are approximately parallel, suggesting little interaction.

The trouble with interaction plots is that we always expect there to be some random variation so it is sometimes difficult to distinguish true interaction from just noise. Fortunately, in this case, we have replication so we can directly test for an interaction effect. We fit the full model and check the significance of the factors:

```
> lmod <- lm(psize ~ operator*resin, pvc)
> anova(lmod)
Analysis of Variance Table

Response: psize
          Df Sum Sq Mean Sq F value Pr(>F)
operator   2   20.7    10.4    7.01  0.004
resin      7  283.9    40.6   27.44 5.7e-10
operator:resin 14   14.3     1.0    0.69  0.760
Residuals 24   35.5     1.5
```

We see that the interaction effect is not significant. This now allows a meaningful investigation of the significance of the main effects. We see that both main effects are significant.

Now one may be tempted to remove the interaction and then retest the main effects:

```
> anova(lm(psize ~ operator + resin, pvc))
Analysis of Variance Table

Response: psize
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
operator	2	20.7	10.4	7.9	0.0014
resin	7	283.9	40.6	30.9	8.1e-14
Residuals	38	49.8	1.3		

One gets a somewhat different result although the conclusions are the same. The difference lies in the $\hat{\sigma}^2$ used in the denominator of the F-tests (1.5 in the first table and 1.3 in the second). The first version has been shown to achieve greater power and is the preferred method.

We check the diagnostics, as seen in Figure 16.3:

```
> qqnorm(residuals(lmod), main="")
> qqline(residuals(lmod))
> plot(fitted(lmod), residuals(lmod), xlab="Fitted", ylab="Residuals")
> abline(h=0)
> plot(residuals(lmod) ~ operator, pvc, ylab="Residuals")
```

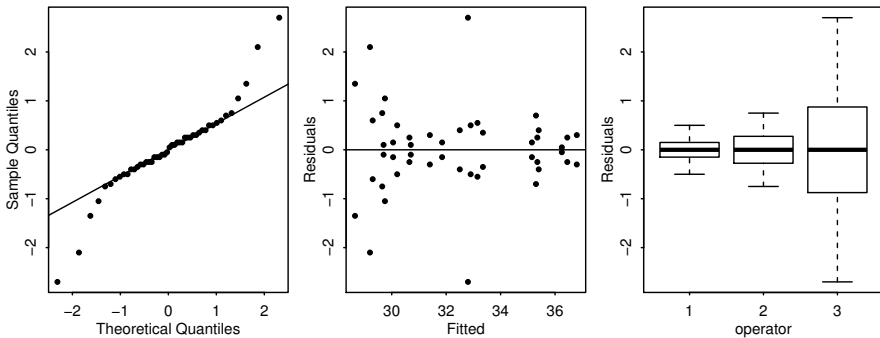


Figure 16.3 Diagnostics plots for the full model for the PVC data.

The Q-Q plot suggests that the errors are somewhat long-tailed. We can see some symmetry in the residuals vs. fitted plot. For each combination of the factors, the mean of the two replicates is the fitted value. The two residuals for that pair will be mirror images. A closer look at the more distant pairs reveals that they derive from operator 3 as seen in the third panel of Figure 16.3.

Although the plot is fairly convincing, we can test for a difference in the variance. For each pair of observations coming from a particular combination of operator and resin, we compute the two residuals. One will be positive and the other negative due to the symmetry previously noted. We take $\sqrt{|e|}$ as the new response. This measures the spread in the pair with the square root transformation reducing the skewness in the distribution. Note that we need only need one observation from each pair since these are identical.

```
> pvce <- pvc[(1:24)*2,]
> pvce$res <- sqrt(abs(residuals(lmod))[(1:24)*2])
> vmod <- lm(res ~ operator + resin, pvce)
> anova(vmod)
Analysis of Variance Table
```

```
Response: res
      Df Sum Sq Mean Sq F value Pr(>F)
operator  2  1.490   0.745   15.12 0.00032
resin    7  0.638   0.091    1.85 0.15545
Residuals 14  0.690   0.049
```

This confirms a significant difference in variation for the operators. It also tells us there is no strong evidence against constant variation among the resin cars.

This may be the most important finding from such an experiment because consistency in manufacturing is very important. In many situations, one wishes to manufacture an object to a specification. It is often possible to adjust the mean to the desired level but a high variance will be detrimental to production quality.

Even so, let us examine the main effects, bearing in mind that we do not have constant variance of the errors, making the comparisons less than optimal.

```
> lmod <- lm(psize ~ operator+resin, pvc)
> summary(lmod)
      Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.240      0.523   69.34 < 2e-16
operator2    -0.263      0.405   -0.65 0.52059
operator3    -1.506      0.405   -3.72 0.00064
resin2       -1.033      0.661   -1.56 0.12630
resin3       -5.800      0.661   -8.77 1.1e-10
resin4       -6.183      0.661   -9.35 2.1e-11
resin5       -4.800      0.661   -7.26 1.1e-08
resin6       -5.450      0.661   -8.24 5.5e-10
resin7       -2.917      0.661   -4.41 8.2e-05
resin8       -0.183      0.661   -0.28 0.78302

n = 48, p = 10, Residual SE = 1.145, R-Squared = 0.86
```

Suppose that we valued a small particle size. In this case, operator 3 produces the best results overall. If we have a choice, we would also prefer resin car 4 for the smallest particle size. However, sometimes we may not have control over a factor. For example, the resin cars might be disposable and we cannot reliably produce one to a particular specification. In such cases, there is no point in identifying which is best in the experiment. Nevertheless, we would still include the factor in the model because it would allow us to estimate the operator effects more precisely.

We can construct pairwise confidence intervals for the treatment factors using the Tukey method:

```
> TukeyHSD(aov(psize ~ operator+resin, data=pvc))
Tukey multiple comparisons of means
 95% family-wise confidence level

Fit: aov(formula = psize ~ operator + resin, data = pvc)

$operator
      diff      lwr      upr    p adj
2-1 -0.2625 -1.2497  0.72475 0.79436
3-1 -1.5063 -2.4935 -0.51900 0.00181
3-2 -1.2438 -2.2310 -0.25650 0.01068

$resin
```

	diff	lwr	upr	p adj
2-1	-1.03333	-3.15228	1.08561	0.76833
3-1	-5.80000	-7.91895	-3.68105	0.00000
4-1	-6.18333	-8.30228	-4.06439	0.00000
etc				

We see that operators 1 and 2 are not significantly different but operator 3 is different from both. There are more significant differences among the resin cars although this may not be of specific interest.

The analysis above is appropriate for the investigation of specific operators and resin cars. These factors are being treated as *fixed effects*. If the operators and resin cars were randomly selected from larger populations of those available, they should be analyzed as *random effects*. This would require a somewhat different analysis not covered here. However, we can at least see from the analysis above that the variation between resin cars is greater than that between operators.

It is important that the observations taken in each cell are genuine replications. If this is not true, then the observations will be correlated and the analysis will need to be adjusted. It is a common scientific practice to repeat measurements and take the average to reduce measurement errors. These repeat measurements are not independent observations. Data where the replicates are correlated can be handled with repeated measures models. For example, in this experiment we would need to take some care to separate the two measurements for each operator and resin car. Some knowledge of the engineering might be necessary to achieve this.

16.3 Two Factors with an Interaction

An industrial experiment examined the number of warp breaks in yarn depending on two types of wool and three levels of tension used in a weaving machine. The data may be found in Tukey (1977).

Simple plots of the data can be obtained from:

```
> plot(breaks ~ wool, warpbreaks)
> with(warpbreaks, interaction.plot(wool, tension, breaks))
```

The plots should be repeated, reversing the roles of the two predictors. More informative but harder-to-construct plots of the data are shown in Figure 16.4.

```
> require(ggplot2)
> ggplot(warpbreaks, aes(x=wool, y=breaks, shape=tension)) + geom_point (
  position = position_jitter(width = .1)) + stat_summary(fun.y="mean",
  geom="line", aes(group=tension, linetype=tension)) + theme(legend.
  position = "top", legend.direction = "horizontal")
> ggplot(warpbreaks, aes(x=tension, y=breaks, shape=wool)) + geom_point (
  position = position_jitter(width = .1)) + stat_summary(fun.y="mean",
  geom="line", aes(group=wool, linetype=wool)) + theme(legend.
  position = "top", legend.direction = "horizontal")
```

There is some evidence of non-constant variation. We can also see there may some interaction between the factors as the lines joining the mean response at each level are not close to parallel.

We can now fit a model with an interaction effect and check the diagnostics:

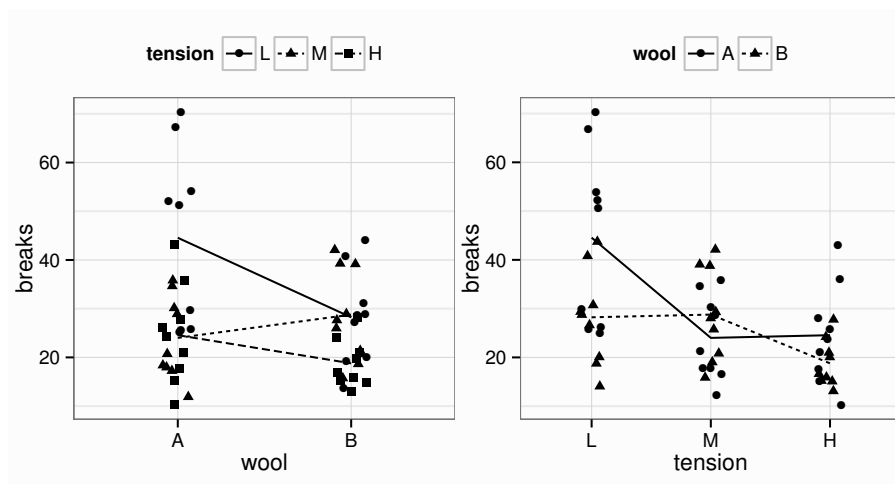


Figure 16.4 Plots of the warpbreak data. Lines join the mean responses for levels of the factor not plotted on the horizontal axis. Some horizontal jittering has been added to distinguish tied cases.

```
> lmod <- lm(breaks ~ wool*tension, warpbreaks)
> plot(residuals(lmod) ~ fitted(lmod), xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

The residuals vs. fitted plot shown in the first panel of Figure 16.5 reveals increasing variation with the response. Given that the response is a counted variable, a square root transformation is the suggested solution:

```
> lmod <- lm(sqrt(breaks) ~ wool*tension, warpbreaks)
> plot(residuals(lmod) ~ fitted(lmod), xlab="Fitted", ylab="Residuals")
> abline(h=0)
```

The diagnostic plot shown in the second panel of Figure 16.5 shows an improvement. We can now test for the significance of the effects:

```
> anova(lmod)
Analysis of Variance Table

Response: sqrt(breaks)
      Df Sum Sq Mean Sq F value Pr(>F)
wool    1    2.9   2.90    3.02 0.08854
tension  2   15.9   7.95    8.28 0.00082
wool:tension  2    7.2   3.60    3.75 0.03067
Residuals 48   46.1   0.96
```

We see that there is a significant interaction effect between the factors. This means we cannot express the effect of the choice of wool independently of the setting of the tension. We must interpret their effects together. One may also notice that the main effect for wool is not significant but this does not mean that the choice of wool has no effect. To make this point clearer, consider the regression output:

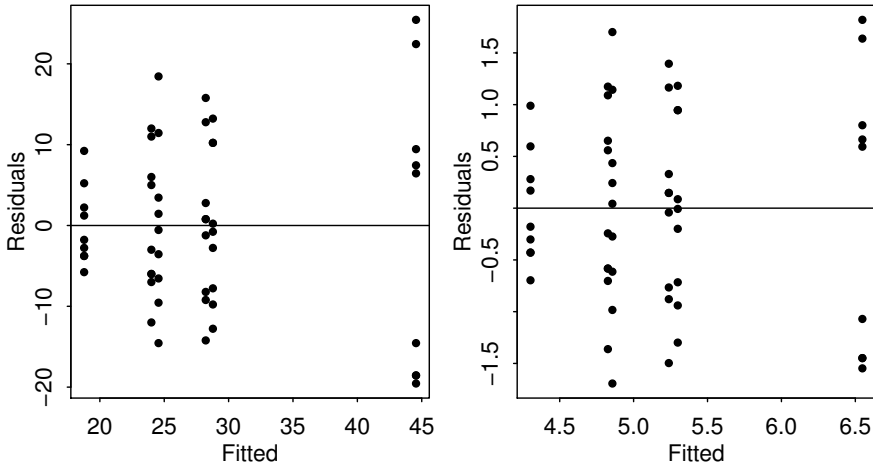


Figure 16.5 Diagnostic plots for the warpbreaks data. The plot on the left comes from the untransformed response while that on the right results from a square-rooted response.

```
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.548     0.327   20.05 < 2e-16
woolB          -1.309     0.462   -2.83 0.00669
tensionM       -1.722     0.462   -3.73 0.00051
tensionH       -1.691     0.462   -3.66 0.00062
woolB:tensionM  1.782     0.653    2.73 0.00887
woolB:tensionH  0.755     0.653    1.16 0.25335

n = 54, p = 6, Residual SE = 0.980, R-Squared = 0.36
```

In this output, we see that the main effect term for wool is significant. The reason for the discrepancy is that the two tests of wool are different. The former test compares the wool choice averaged over all levels of tension while the latter makes the comparison at the reference level of tension (low in this case). So there is no simple answer to the question of whether the choice of wool makes a difference to the response. In the presence of a significant interaction, it is better not to consider main effects alone but consider the effect of the two factors together.

The regression output above does not make it easy to interpret the nature of the interaction. Sometimes it can be easier to think of the model as having just one combined factor with a number of levels equal to the product of the two factor levels. We can achieve this as follows:

```
> lmod <- lm(sqrt(breaks) ~ wool:tension-1, warpbreaks)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
woolA:tensionL    6.548     0.327   20.1 <2e-16
woolB:tensionL    5.238     0.327   16.0 <2e-16
woolA:tensionM    4.826     0.327   14.8 <2e-16
```

woolB:tensionM	5.299	0.327	16.2	<2e-16
woolA:tensionH	4.856	0.327	14.9	<2e-16
woolB:tensionH	4.302	0.327	13.2	<2e-16

n = 54, p = 6, Residual SE = 0.980, R-Squared = 0.97

We can see that wool B at high tension produces the lowest expected number of breaks. But is this choice significantly better than the others? We can answer this question by computing the Tukey HSD intervals:

```
> TukeyHSD(aov(breaks ~ wool:tension, warpbreaks))
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = breaks ~ wool:tension, data = warpbreaks)

$`wool:tension`
      diff      lwr      upr    p adj
B:L-A:L -16.33333 -31.640 -1.02701 0.03021
A:M-A:L -20.55556 -35.862 -5.24923 0.00296
B:M-A:L -15.77778 -31.084 -0.47146 0.03982
A:H-A:L -20.00000 -35.306 -4.69368 0.00410
B:H-A:L -25.77778 -41.084 -10.47146 0.00011
A:M-B:L -4.22222 -19.529  11.08410 0.96265
B:M-B:L  0.55556 -14.751  15.86188 1.00000
A:H-B:L -3.66667 -18.973  11.63966 0.97971
B:H-B:L -9.44444 -24.751   5.86188 0.45609
B:M-A:M  4.77778 -10.529  20.08410 0.93772
A:H-A:M  0.55556 -14.751  15.86188 1.00000
B:H-A:M -5.22222 -20.529  10.08410 0.91148
A:H-B:M -4.22222 -19.529  11.08410 0.96265
B:H-B:M -10.00000 -25.306   5.30632 0.39188
B:H-A:H -5.77778 -21.084   9.52854 0.87056
```

We see that no pairwise difference involving medium or high tensions is significant. So the advice should be to avoid running the machine at low tension if possible. If we must make a choice, using wool B at high tension is most likely to be the best choice. But if there are cost differences between A and B or difficulties in running the machines at high tension, then we can see that other choices may be reasonable.

16.4 Larger Factorial Experiments

Suppose we have factors $\alpha, \beta, \gamma, \dots$ at levels $l_\alpha, l_\beta, l_\gamma, \dots$. A *full* factorial experiment has at least one run for each combination of the levels. The number of combinations is $l_\alpha l_\beta l_\gamma \dots$, which could easily be very large. The biggest model for a full factorial contains all possible interaction terms, which range from second-order, or two-way, as encountered earlier in this chapter, to high order interactions involving several factors. For this reason, full factorials are rarely executed for more than three or four factors.

There are some advantages to factorial designs. If no interactions are significant, we get several one-way experiments for the price of one. Compare this with doing a sequence of one-way experiments. It is sometimes better to use replication for

investigating another factor instead. For example, instead of doing a two-factor experiment with replication, we might use that replication to investigate a third factor but now with no replication.

The analysis of full factorial experiments is an extension of that used for the two-way ANOVA. Typically, there is no replication due to cost concerns so it is necessary to assume that some higher order interactions are zero in order to free up degrees of freedom for testing the lower order effects. Not many phenomena require a precise combination of several factors so this is not unreasonable.

Fractional factorials

Fractional factorials use only a fraction of the number of runs in a full factorial experiment. This is done to save the cost of the full experiment or to make only a few runs because the experimental material is limited. It is often possible to estimate the lower order effects with just a fraction. Consider an experiment with seven factors, each at two levels:

Effect	mean	main	2-way	3-way	4	5	6	7
Number of parameters	1	7	21	35	35	21	7	1

Table 16.1 Number of parameters in a two-level, seven-factor experiment.

If we are going to assume that higher order interactions are negligible then we do not really need $2^7 = 128$ runs to estimate the remaining parameters. We could perform only eight runs and still be able to estimate the seven main effects, though none of the interactions. In this particular example, it is hard to find a design to estimate all the two-way interactions uniquely, without a large number of runs. The construction of good designs is a complex task. For example, see Hamada and Wu (2000) for more on this. A Latin square (see Section 17.2) where all predictors are considered as factors is another example of a fractional factorial.

In fractional factorial experiments, we try to estimate many parameters with as few data points as possible. This means there are often not many degrees of freedom left. We require that σ^2 be small; otherwise there will be little chance of distinguishing significant effects. Fractional factorials are popular in engineering applications where the experiment and materials can be tightly controlled. Fractional factorials are commonly found in product design because they allow for the screening of a large number of factors. Factors identified in a screening experiment can then be more closely investigated. In the social sciences and medicine, the experimental materials, often human or animal, are much less homogeneous and less controllable, so σ^2 tends to be relatively large. In such cases, fractional factorials are of no value.

Here is an example. Speedometer cables can be noisy because of shrinkage in the plastic casing material. An experiment was conducted to find out what caused shrinkage by screening a large number of factors. The engineers started with 15 different factors: liner outside diameter, liner die, liner material, liner line speed, wire braid type, braiding tension, wire diameter, liner tension, liner temperature, coating material, coating die type, melt temperature, screen pack, cooling method and line

speed, labeled a through o. Response is percentage of shrinkage per specimen. There were two levels of each factor. The “+” indicates the high level of a factor and the “-” indicates the low level.

A full factorial would take 2^{15} runs, which is highly impractical, thus a design with only 16 runs was used where the particular runs have been chosen specially so as to estimate the mean and the 15 main effects. We assume that there is no interaction effect of any kind. The data come from Box et al. (1988).

Here is the data:

```
> data(speedo, package="faraway")
> speedo
  h d l b j f n a i e m c k g o      y
1  - - + - + + - - + + - - + 0.4850
2  + - - - - + + - - + + + - - 0.5750
3  - + - - + - + - + - + + - + - 0.0875
4  + + + - - - - - - - + + + + 0.1750
5  - - + + - - - + - + + - - + + - 0.1950
6  + - - + + - - - - + + - - + + 0.1450
7  - + - + - + - - + - + - + - + 0.2250
8  + + + + + + + - - - - - - - - 0.1750
9  - - + - + + - + - - + - + + - 0.1250
10 + - - - - + + + + - - - - + + 0.1200
11 - + - - + - + + - + - - + - + 0.4550
12 + + + - - - - + + + + - - - - 0.5350
13 - - + + - - + + - - + + - - + 0.1700
14 + - - + + - - + + - - + + - - 0.2750
15 - + - + - + - + - + - + - + - 0.3425
16 + + + + + + + + + + + + + + 0.5825
```

Perhaps you can see the pattern in the design. We can fit and examine a main-effects-only model:

```
> lmod <- lm(y ~ ., speedo)
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.582500         NA      NA      NA
h-           -0.062188         NA      NA      NA
d-           -0.060938         NA      NA      NA
l-           -0.027188         NA      NA      NA
b-            0.055937         NA      NA      NA
j-            0.000937         NA      NA      NA
f-           -0.074062         NA      NA      NA
n-           -0.006563         NA      NA      NA
a-           -0.067813         NA      NA      NA
i-           -0.042813         NA      NA      NA
e-           -0.245312         NA      NA      NA
m-           -0.027812         NA      NA      NA
c-           -0.089687         NA      NA      NA
k-           -0.068437         NA      NA      NA
g-            0.140312         NA      NA      NA
o-           -0.005938         NA      NA      NA

n = 16, p = 16, Residual SE = NaN, R-Squared = 1
F-statistic:  NaN on 15 and 0 DF,  p-value: NA
```

There are no degrees of freedom, because there are as many parameters as cases. We

cannot do any of the usual tests. It is important to understand the coding here, so look at the X-matrix:

```
> model.matrix(lmod)
      (Intercept) h d l b j f n a i e m c k g o
1                1 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0
...etc...
```

We see that “+” is coded as zero and “-” is coded as one. This unnatural ordering is because of the order of “+” and “-” in the ASCII alphabet.

We do not have any degrees of freedom so we can not make the usual F-tests. We need a different method to determine significance. Suppose there were no significant effects and the errors were normally distributed. The estimated effects would then just be linear combinations of the errors and hence normal. We now make a normal quantile plot of the main effects with the idea that outliers represent significant effects:

```
> qqnorm(coef(lmod)[-1], pch=names(coef(lmod)[-1]))
```

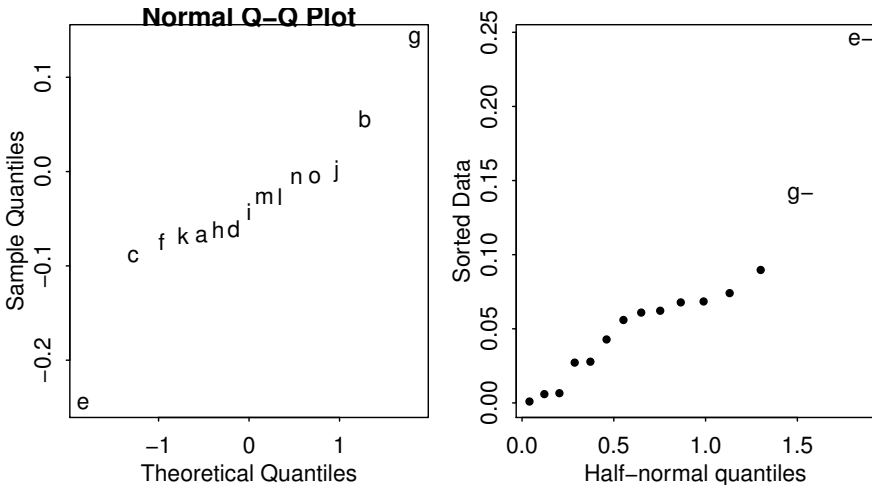


Figure 16.6 Q-Q plots of effects for speedometer cable analysis.

See Figure 16.6. Notice that “e” and possibly “g” are extreme. Since the “e” effect is negative, the “+” level of “e” increases the response. Since shrinkage is a bad thing, increasing the response is not good so we would prefer whatever “wire braid” type corresponds to the “-” level of “e”. The same reasoning for “g” leads us to expect that a larger (assuming that is “+”) would decrease shrinkage.

A half-normal plot, as described in Section 6.2.1, is better for detecting extreme points:

```
> halfnorm(coef(lmod)[-1], labs=names(coef(lmod)[-1]))
```

We might now conduct another experiment focusing on the effects of “e” and “g.”

Exercises

1. Data on the butterfat content of milk from Canadian cows of five different breeds and two different ages can be found in the `butterfat` dataset.
 - (a) Make appropriate plots of the data.
 - (b) Determine whether there is an interaction between breed and age.
 - (c) Determine whether there is a statistically significant difference between breeds and also ages.
 - (d) Present regression diagnostics for your chosen model and comment whether the assumptions have been met.
 - (e) Is the best breed in terms of butterfat content clearly superior to the second best breed?
2. The `barley` data may be found in the `lattice` package. Perform a three-way ANOVA with `yield` as the response. Check the diagnostics — you will find that two points stick out, which correspond to the same variety and site, but for different years. There is reason to suspect that the response values for these cases have been transposed. Investigate the effect of transposing these cases on the analysis.
3. Determine the important factors in the `sono` dataset where the `Intensity` is the response and the other variables are predictors.
4. Using the `rats` data, model the survival time in terms of the poison and treatment. Use the Box–Cox method to determine an appropriate transformation on the response. Which treatments and which poisons are significantly different?
5. The `peanut` data come from a fractional factorial experiment to investigate factors that affect an industrial process using carbon dioxide to extract oil from peanuts. Determine which factors are important, remembering to consider two-way interactions.
6. The “High School and Beyond” data is found in `hsb`.
 - (a) Model the math score in terms of the five factors: `gender`, `race`, `ses`, `schtyp` and `prog`. Include all second-order interactions but no higher order interactions. How many regression parameters does your model use? Explain how this can be calculated in terms of the number of levels for each factor.
 - (b) Determine whether some two-way interactions can be eliminated using the `anova` function.
 - (c) Determine whether some two-way interactions can be eliminated but now using the `drop1` function. Why do the results differ from the previous question? Which method of testing should be preferred?
 - (d) Fit a model with only main effects and compare it to the model with all two-way interactions. Which model should be preferred?
 - (e) Determine which main effects are statistically significant and report on the differences you find.
 - (f) Check the regression diagnostics for the main effects only model and report on any anomalies.

Experiments with Blocks

In a completely randomized design (CRD), the treatments are assigned to the experimental units at random. This is appropriate when the units are homogeneous, as has been assumed in the designs leading to the one- and two-way analyses of variances (ANOVAs). Sometimes, we may suspect that the units are heterogeneous, but we cannot describe the form the difference takes — for example, we may know that a group of patients are not identical, but we may have no further information about them. In this case, it is still appropriate to use a CRD. Of course, the randomization will tend to spread the heterogeneity around to avoid systematic bias, but the real justification lies in the randomization test discussed in Section 3.3. Under the null hypothesis, there is no link between a factor and the response. In other words, the responses have been assigned to the units in a way that is unlinked to the factor. This corresponds to the randomization used in assigning the levels of the factor to the units. This is why the randomization is crucial because it allows us to make this argument. Now if the difference in the response between levels of the factor seems too unlikely to have occurred by chance, we can reject the null hypothesis. The normal-based inference is approximately equivalent to the permutation-based test. Since the normal-based inference is much quicker, we usually prefer to use that.

When the experimental units are heterogeneous in a known way and can be arranged into *blocks* where the within-block variation is small, but the between-block variation is large, a *block design* can be more efficient than a CRD. We prefer to have a block size equal to the number of treatments. If this cannot be done, an *incomplete* block design must be used.

Sometimes the blocks are determined by the experimenter. For example, suppose we want to compare four treatments and have 20 patients available. We might divide the patients into five blocks of four patients each where the patients in each block have some relevant similarity. We might decide this subjectively in the absence of specific information. In other cases, the blocks are predetermined by the nature of the experiment. For example, suppose we want to test three crop varieties on five fields. Restrictions on planting, harvesting and irrigation equipment might allow us only to divide the fields into three strips. The fields are the blocks.

In a randomized block design, the treatment levels are assigned randomly within a block. This means the randomization is restricted relative to the full randomization used in the CRD. This has consequences for the inference. There are fewer possible permutations for the random assignment of the treatments; therefore, the computation of the significance of a statistic based on the permutation test would need to

be modified. Consequently, a block effect must be included in the model used for inference about the treatments, even if the block effect is not significant.

17.1 Randomized Block Design

We have one treatment factor, τ at t levels and one blocking factor, β at r levels. The model is:

$$y_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij}$$

where τ_i is the treatment effect and β_j is the blocking effect. There is one observation on each treatment in each block. This is called a randomized complete block design (RCBD). The analysis is then very similar to a two-factor experiment with no replication. We have a limited ability to detect an interaction between treatment and block. We can check for a treatment effect. We can also check the block effect, but this is only useful for future reference. Blocking is a feature of the experimental units and restricts the randomized assignment of the treatments. This means that we cannot regain the degrees of freedom devoted to blocking even if the blocking effect is not significant.

We illustrate this with an experiment to compare eight varieties of oats. The growing area was heterogeneous and so was grouped into five blocks. Each variety was sown once within each block and the yield in grams per 16-ft row was recorded. The data come from Anderson and Bancroft (1952).

We start with a look at the data:

```
> data(oatvar, package="faraway")
> xtabs(yield ~ variety + block, oatvar)
      block
variety  I  II III  IV  V
1      296 357 340 331 348
2      402 390 431 340 320
3      437 334 426 320 296
4      303 319 310 260 242
5      469 405 442 487 394
6      345 342 358 300 308
7      324 339 357 352 220
8      488 374 401 338 320
```

Simple plots of the data can be obtained in base graphics by:

```
> plot(yield ~ variety, oatvar)
> plot(yield ~ block, oatvar)
> with(oatvar, interaction.plot(variety, block, yield))
> with(oatvar, interaction.plot(block, variety, yield))
```

Alternatively, we can use `ggplot2`:

```
> require(ggplot2)
> ggplot(oatvar, aes(x=variety, y=yield, group=block, linetype=block)) +
  geom_line() + theme(legend.position = "top", legend.direction = "
  horizontal")
> ggplot(oatvar, aes(x=block, y=yield, group=block, linetype=block)) + geom_
  line()
```

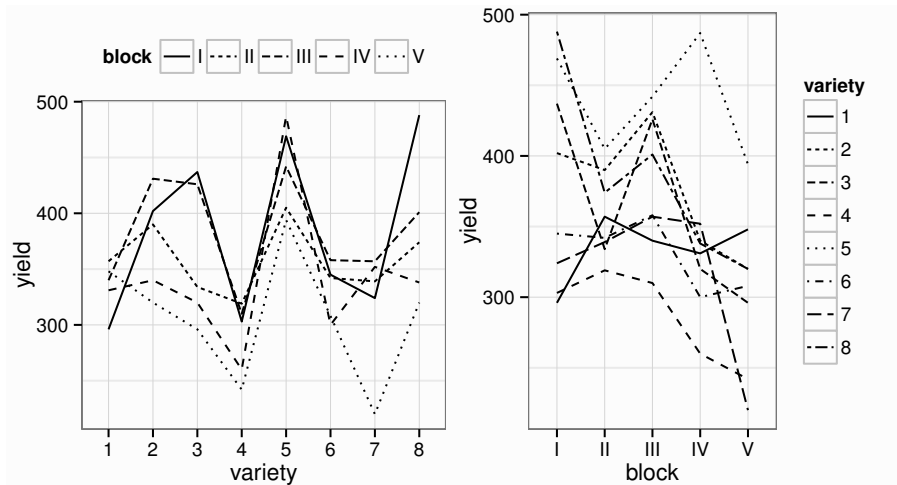



Figure 17.1 Interaction plots of oat variety data.

See Figure 17.1. There is no indication of outliers, skewness or non-constant variance. Nor is there any clear evidence of an interaction effect. We can move onto the modeling:

```
> lmod <- lm(yield ~ block+variety, oatvar)
> anova(lmod)
Analysis of Variance Table

Response: yield
      Df Sum Sq Mean Sq F value Pr(>F)
block   4  33396    8349   6.24  0.001
variety  7   77524   11075   8.28 1.8e-05
Residuals 28  37433    1337
```

Both effects are significant. The ANOVA table corresponds to a sequential testing of models, here corresponding to the sequence:

```
y ~ 1
y ~ block
y ~ block+variety
```

So here the p -value 0.001 corresponds to a comparison of the first two models in this list, while the p -value of 1.8e-05 corresponds to the test comparing the second two. The denominator in both F-tests is the mean square from the full model, here 1337. This means that a test of the block effect that leaves out the variety effect is not the same:

```
> anova(lm(yield ~ block, oatvar))
Analysis of Variance Table

Response: yield
      Df Sum Sq Mean Sq F value Pr(>F)
```

```
block      4  33396    8349    2.54  0.057
Residuals 35 114957    3284
```

There is a difference in significance in this case. This latter test is incorrect for testing the blocking effect.

Notice that if we change the order of the terms in the ANOVA, it makes no difference because of the orthogonal design:

```
> anova(lm(yield ~ variety+block, oatvar))
Analysis of Variance Table
```

```
Response: yield
      Df Sum Sq Mean Sq F value Pr(>F)
variety  7  77524   11075    8.28 1.8e-05
block    4  33396    8349    6.24  0.001
Residuals 28  37433    1337
```

By way of comparison, see what happens if we omit the first observation in the dataset — this might happen in practice if this run is lost:

```
> anova(lm(yield ~ block+variety,subset=-1, oatvar))
Analysis of Variance Table
```

```
Response: yield
      Df Sum Sq Mean Sq F value Pr(>F)
block    4  38581    9645    8.41 0.00015
variety  7  75339   10763    9.38 7.3e-06
Residuals 27  30968    1147
```

```
> anova(lm(yield ~ variety+block,subset=-1, oatvar))
Analysis of Variance Table
```

```
Response: yield
      Df Sum Sq Mean Sq F value Pr(>F)
variety  7  75902   10843    9.45 6.8e-06
block    4  38018    9504    8.29 0.00017
Residuals 27  30968    1147
```

As there is one missing observation, the design is no longer orthogonal and the order does matter, although it would not change the general conclusions. If we want to test for a treatment effect, we would prefer the first of these two tables since in that version the blocking factor is already included when we test the treatment factor. Since the blocking factor is an unalterable feature of the chosen design, this is as it should be. A convenient way to test all terms relative to the full model is:

```
> drop1(lm(yield ~ variety+block,subset=-1, oatvar),test="F")
Single term deletions
```

```
Model:
yield ~ variety + block
      Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>          30968    284
variety  7    75339 106307    319    9.38 7.3e-06
block    4    38018  68986    308    8.29 0.00017
```

Check the diagnostics, as seen in Figure 17.2:

```
> plot(lmod,which=1:2,caption=NULL)
```

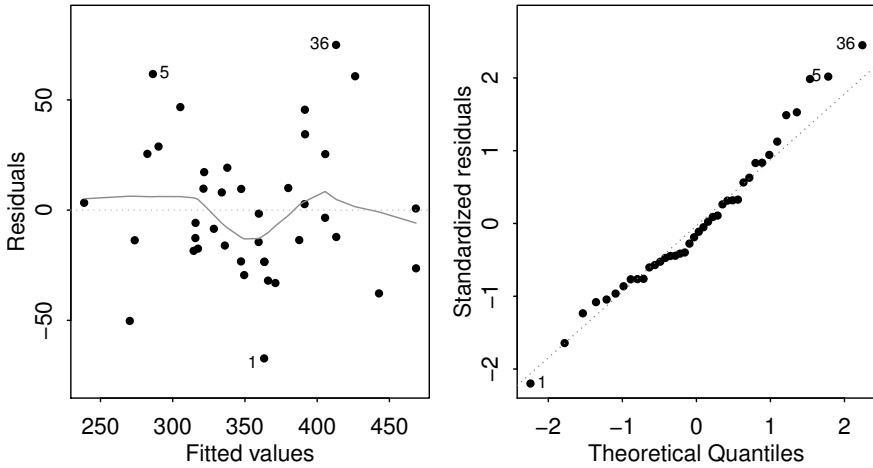


Figure 17.2 Diagnostic plots for the oat variety data.

An examination of which varieties give the highest yields and which are significantly better than others can now follow.

We did assume that the interactions were not significant. We looked at the interaction plots, but we can also execute the Tukey non-additivity test:

```
> varcoefs <- c(0,coef(lmod)[6:12])
> blockcoefs <- -c(0,coef(lmod)[2:5])
> ab <- rep(varcoefs,each=5)*rep(blockcoefs,8)
> h <- update(lmod, ~.+ab)
> anova(h)
Analysis of Variance Table

Response: yield
      Df Sum Sq Mean Sq F value Pr(>F)
block    4  33396    8349   6.06  0.0013
variety  7  77524   11075   8.03  2.8e-05
ab       1    213     213   0.15  0.6974
Residuals 27  37220    1379
```

Because the p -value of the treatment times block effect is 0.6974, we accept the null hypothesis of no interaction. Of course, the interaction may be of a non-multiplicative form, but there is little we can do about that.

Relative advantage of RCBD over CRD

We can measure precision by considering $\text{var } \hat{\tau}$ or equivalently $\hat{\sigma}^2$. We should compare the $\hat{\sigma}^2$ for designs with the same sample size. We define *relative efficiency* as $\hat{\sigma}_{CRD}^2 / \hat{\sigma}_{RCBD}^2$ where the quantities can be computed by fitting models with and without the blocking effect. For the example above:

```
> lmcrd <- lm(yield ~ variety, oatvar)
> summary(lmcrd)$sig
[1] 47.047
```

```
> summary(lmod)$sig
[1] 36.564
> (47.047/36.564)^2
[1] 1.6556
```

So a CRD would require 66% more observations to obtain the same level of precision as an RCBD.

The efficiency is not guaranteed to be greater than one. Only use blocking where there is some heterogeneity in the experimental units. The decision to block is a matter of judgment prior to the experiment. There is no guarantee that it will increase precision.

17.2 Latin Squares

Latin squares can be useful when there are two blocking variables. For example, in a field used for agricultural experiments, the level of moisture may vary across the field in one direction and the fertility in another. In an industrial experiment, suppose we wish to compare four production methods (the treatments) — A, B, C and D. We have available four machines 1, 2, 3 and 4, and four operators, I, II, III and IV. An example of a Latin square design is shown in Table 17.1.

	1	2	3	4
I	A	B	C	D
II	B	D	A	C
III	C	A	D	B
IV	D	C	B	A

Table 17.1 A 4×4 Latin square showing the treatments (A to D) used for different combinations of two factors.

Each treatment is assigned to each block once and only once. We should choose randomly from all the possible Latin square layouts. The R package `magic` can be used to generate such squares.

Let τ be the treatment factor and β and γ be the two blocking factors; then the model is:

$$y_{ijk} = \mu + \tau_i + \beta_j + \gamma_k + \varepsilon_{ijk} \quad i, j, k = 1, \dots, t$$

All combinations of i, j and k do not appear. To test for a treatment effect, fit a model without the treatment effect and compare using the F-test. The Tukey pairwise CIs are:

$$\hat{\tau}_l - \hat{\tau}_m \pm \frac{q_{t,(t-1)(t-2)}}{\sqrt{2}} \hat{\sigma} \sqrt{2/t}$$

The Latin square can be even more efficient than the RCBD provided that the blocking effects are sizable. There are some variations on the Latin square. The Latin square can be replicated if more runs are available. We need to have both block sizes to be equal to the number of treatments. This may be difficult to achieve. Latin rectangle designs are possible by adjoining Latin squares. When there are

three blocking variables, a Graeco–Latin square may be used but these rarely arise in practice.

The Latin square can also be used for comparing three treatment factors. Only t^2 runs are required compared to the t^3 required if all combinations were run. (The downside is that you cannot estimate the interactions if they exist.) This is an example of a *fractional factorial* as discussed in Section 16.4.

In an experiment reported by Davies (1954), four materials, A, B, C and D, were fed into a wear-testing machine. The response is the loss of weight in 0.1 mm over the testing period. The machine could process four samples at a time and past experience indicated that there were some differences due to the positions of these four samples. Also some differences were suspected from run to run. Four runs were made. We can display the layout of the data:

```
> data(abrasion, package="faraway")
> xtabs(wear ~ run + position, abrasion)
  position
run   1   2   3   4
  1 235 236 218 268
  2 251 241 227 229
  3 234 273 274 226
  4 195 270 230 225
```

We can view the Latin square structure corresponding to this table:

```
> matrix(abrasion$material, 4, 4, byrow=TRUE)
     [,1] [,2] [,3] [,4]
[1,] "C"  "D"  "B"  "A"
[2,] "A"  "B"  "D"  "C"
[3,] "D"  "C"  "A"  "B"
[4,] "B"  "A"  "C"  "D"
```

Plot the data:

```
> ggplot(abrasion, aes(x=run, y=wear, shape=position, group=material)) +
  geom_point() + geom_line(aes(linetype=material))
> ggplot(abrasion, aes(x=position, y=wear, shape=run, group=material)) +
  geom_point() + geom_line(aes(linetype=material))
```

Examine the plots in Figure 17.3. There appear to be differences in all variables. No outliers, skewness or unequal variance are apparent. There is no obvious sign of interaction.

Now fit the Latin square model and test each variable relative to the full model:

```
> lmod <- lm(wear ~ material+run+position, abrasion)
> drop1(lmod, test="F")
Single term deletions

Model:
wear ~ material + run + position
             Df Sum of Sq  RSS   AIC F value    Pr(>F)
<none>                 368   70.1
material    3         4622 4989 105.9   25.15 0.00085
run         3         987 1354  85.0    5.37 0.03901
position    3         1468 1836  89.9    7.99 0.01617
```

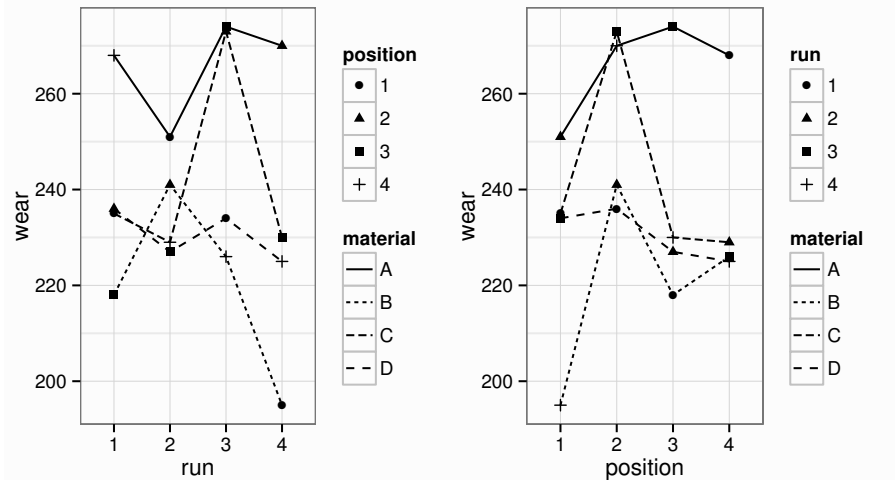


Figure 17.3 Amount of wear depending on material, run and position.

We see that all variables are statistically significant. There are clear differences between the materials. We checked the diagnostics on the model, which showed nothing remarkable. We examine the coefficients:

```
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    254.75      6.19    41.17 1.4e-08
materialB     -45.75      5.53   -8.27 0.00017
materialC     -24.00      5.53   -4.34 0.00489
materialD     -35.25      5.53   -6.37 0.00070
run2           -2.25      5.53   -0.41 0.69842
run3           12.50      5.53    2.26 0.06466
run4           -9.25      5.53   -1.67 0.14566
position2      26.25      5.53    4.74 0.00318
position3      8.50      5.53    1.54 0.17545
position4      8.25      5.53    1.49 0.18661

n = 16, p = 10, Residual SE = 7.826, R-Squared = 0.95
```

We see that material B looks best (in terms of least wear) followed by material D. Is the difference significant though? Which materials in general are significantly better than others? We need the Tukey pairwise intervals to help determine this. The width of the band is calculated in the usual manner:

```
> qtukey(0.95, 4, 6) * 5.53 / sqrt(2)
[1] 19.143
```

The width of the interval is 19.1. We can make a table of the material differences:

```
> scoefs <- c(0, coef(lmod)[2:4])
> outer(scoefs, scoefs, "-")
              materialB materialC materialD
materialB    0.00    45.75    24.00    35.25
```

materialB	-45.75	0.00	-21.75	-10.50
materialC	-24.00	21.75	0.00	11.25
materialD	-35.25	10.50	-11.25	0.00

We see that the (B, D) and (D, C) differences are not significant at the 5% level, but that all the other differences are significant.

If maximizing resistance to wear is our aim, we would pick material B but if material D offered a better price, we might have some cause to consider switching to D. The decision would need to be made with cost–quality trade-offs in mind.

Now we compute how efficient the Latin square is compared to other designs. We compare to the completely randomized design:

```
> lmodr <- lm(wear ~ material, abrasion)
> (summary(lmodr)$sig/summary(lmod)$sig)^2
[1] 3.8401
```

We see that the Latin square is 3.84 times more efficient than the CRD. This is a substantial gain in efficiency. The Latin square may also be compared to designs where we block only one of the variables. The efficiency relative to these designs is less impressive, but still worthwhile.

17.3 Balanced Incomplete Block Design

In a complete block design, the block size is equal to the number of treatments. When the block size is less than the number of treatments, an incomplete block design must be used. For example, in the oat data, suppose six oat varieties were to be compared, but each field had space for only four plots.

In an incomplete block design, the treatments and blocks are *not* orthogonal. Some treatment contrasts will not be identifiable from certain block contrasts. This is an example of *confounding*. This means that those treatment contrasts effectively cannot be examined. In a balanced incomplete block (BIB) design, all the pairwise differences are identifiable and have the same standard error. Pairwise differences are more likely to be interesting than other contrasts, so the design is constructed to facilitate this.

Suppose, we have four treatments ($t = 4$) designated A, B, C, D and the block size, $k = 3$ and there are $b = 4$ blocks. Therefore, each treatment appears $r = 3$ times in the design. One possible BIB design is:

Block 1	A	B	C
Block 2	A	B	D
Block 3	A	C	D
Block 4	B	C	D

Table 17.2 BIB design for four treatments with four blocks of size three.

Each pair of treatments appears in the same block $\lambda = 2$ times — this feature

means simpler pairwise comparison is possible. For a BIB design, we require:

$$\begin{aligned} b &\geq t > k \\ rt &= bk = n \\ \lambda(t-1) &= r(k-1) \end{aligned}$$

This last relation holds because the number of pairs in a block is $k(k-1)/2$ so the total number of pairs must be $bk(k-1)/2$. On the other hand, the number of treatment pairs is $t(t-1)/2$. The ratio of these two quantities must be λ .

Since λ has to be an integer, a BIB design is not always possible even when the first two conditions are satisfied. For example, consider $r = 4, t = 3, b = 6, k = 2$ and then $\lambda = 2$ which is OK, but if $r = 4, t = 4, b = 8, k = 2$, then $\lambda = 4/3$ so no BIB is possible. (Something called a partially balanced incomplete block design can then be used.) BIBs are also useful for competitions where not all contestants can fit in the same race.

The model we fit is the same as for the RCBD:

$$y_{ij} = \mu + \tau_i + \beta_j + \varepsilon_{ij}$$

In our example, a nutritionist studied the effects of six diets, “a” through “f,” on weight gains of domestic rabbits. From past experience with sizes of litters, it was felt that only three uniform rabbits could be selected from each available litter. There were ten litters available forming blocks of size three. The data come from Lentner and Bishop (1986). Examine the data:

```
> data(rabbit, package="faraway")
> xtabs(gain ~ treat+block, rabbit)
      block
treat b1  b10 b2  b3  b4  b5  b6  b7  b8  b9
a     0.0 37.3 40.1 0.0 44.9 0.0 0.0 45.2 44.0 0.0
b    32.6  0.0 38.1  0.0  0.0  0.0 37.3 40.6  0.0 30.6
c    35.2  0.0 40.9 34.6 43.9 40.9  0.0  0.0  0.0  0.0
d     0.0 42.3  0.0 37.5  0.0 37.3  0.0 37.9  0.0 27.5
e     0.0  0.0  0.0  0.0 40.8 32.0 40.5  0.0 38.5 20.6
f    42.2 41.7  0.0 34.3  0.0  0.0 42.8  0.0 51.9  0.0
```

The zero values correspond to no observation. The BIB structure is apparent — each pair of diets appears in the same block exactly twice. Now plot the data, as seen in Figure 17.4. In the first plot, we can distinguish the six levels of treatment with a different plotting character. In the second plot, it is difficult to distinguish the ten levels of block with different shapes so we have used a simple dot. Some jittering is necessary to distinguish the points. There is nothing remarkable about the content of the plots as we see no evidence of outliers, skewness or non-constant variance.

```
> ggplot(rabbit, aes(y=gain, x=block, shape=treat)) + geom_point() + theme(
  legend.position = "top", legend.direction = "horizontal")
> ggplot(rabbit, aes(y=gain, x=treat)) + geom_point(position = position_
  jitter(width = .1))
```

We fit the model and test the significance of the effects:

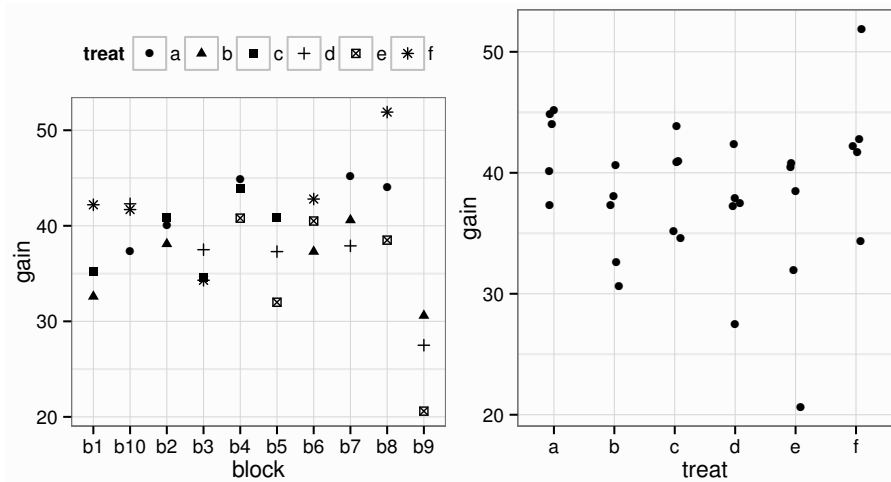


Figure 17.4 Strip plots of rabbit diet data.

```
> lmod <- lm(gain ~ block+treat,rabbit)
> drop1(lmod, test="F")
Single term deletions

Model:
gain ~ block + treat
      Df Sum of Sq  RSS   AIC F value    Pr(>F)
<none>          151  78.4
block    9       596  747 108.4    6.59 0.00076
treat    5       159  309  90.0    3.16 0.03817
```

It is best to use the drop1 function in preference to anova. This is because the order of the terms in the model makes a difference because the design is not orthogonal (because of the incompleteness). To test either term, we need the other term to be present in the model.

Now check the diagnostics:

```
> plot(lmod, which=1:2)
```

The plots are not shown, as they show nothing of interest.

Now we check which treatments differ. The Tukey pairwise CIs need to be directly constructed because this is not a complete layout. We extract the information about the treatment effects and the standard error:

```
> summary(lmod)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.0139    2.5886   13.91 5.6e-10
blockb10     3.2972    2.7960    1.18 0.2567
blockb2      4.1333    2.6943    1.53 0.1458
blockb3     -1.8028    2.6943   -0.67 0.5136
blockb4      8.7944    2.7960    3.15 0.0067
blockb5      2.3056    2.7960    0.82 0.4225
```

```

blockb6      5.4083      2.6943      2.01      0.0631
blockb7      5.7778      2.7960      2.07      0.0565
blockb8      9.4278      2.7960      3.37      0.0042
blockb9     -7.4806      2.7960     -2.68      0.0173
treatb      -1.7417      2.2418     -0.78      0.4493
treatc       0.4000      2.2418      0.18      0.8608
treatd       0.0667      2.2418      0.03      0.9767
treate      -5.2250      2.2418     -2.33      0.0341
treatf       3.3000      2.2418      1.47      0.1617

```

```
n = 30, p = 15, Residual SE = 3.170, R-Squared = 0.86
```

We see that the standard error for the pairwise comparison of treatments is 2.24. Notice that all the treatment standard errors are equal because of the BIB. Now compute the Tukey critical value:

```
> qtuity(0.95, 6, 15)
[1] 4.5947
```

So the intervals have width:

```
> 4.59*2.24/sqrt(2)
[1] 7.2702
```

We check which pairs are significantly different:

```

> tcoefs <- c(0, coef(lmod)[11:15])
> abs(outer(tcoefs, tcoefs, "-")) > 7.27
      treatb treatc treatd treate treatf
treatb FALSE  FALSE  FALSE  FALSE  FALSE
treatc FALSE  FALSE  FALSE  FALSE  FALSE
treatd FALSE  FALSE  FALSE  FALSE  FALSE
treate FALSE  FALSE  FALSE  FALSE  TRUE
treatf FALSE  FALSE  FALSE  TRUE   FALSE

```

Only the $e - f$ difference is significant.

Now let's see how much better this blocked design is than the CRD. We compute the relative efficiency:

```

> lmodt <- lm(gain ~ treat, rabbit)
> (summary(lmodt)$sig/summary(lmod)$sig)^2
[1] 3.0945

```

Blocking was well worthwhile here.

Exercises

1. The alfalfa data arise from a Latin square design where the treatment factor is inoculum and the blocking factors are shade and irrigation. Test the significance of the effects and determine which levels of the treatment factor are significantly different.
2. The eggprod comes from a randomized block experiment to determine factors affecting egg production. Is there a difference in the treatments? What is the nature of the difference? What efficiency was gained by the blocked design?

3. The `morley` data can be viewed as a randomized block experiment with `Run` as the treatment factor and `Expt` as the blocking factor. Is there a difference between runs and what efficiency is gained by blocking?
4. The `OrchardSprays` data arise from a Latin square design. Which factors show a difference? What pairs of treatments are different? Can the treatment factor be replaced with a linear term? What efficiency is gained by this design?
5. The `resceram` data arise from an experiment to test the effects of resistor shape on current noise. The resistors are mounted on plates and only three resistors will fit on a plate although there are four different shapes to be tested. Identify the design and analyze. Which shapes are different?
6. The `penicillin` data derive from the production of penicillin which uses a raw material, corn steep liquor, that is quite variable and can only be made in blends sufficient for four runs. There are four processes, A, B, C and D, for the production.
 - (a) Make suitable plots of the data.
 - (b) Check for a significant difference between the treatments.
 - (c) Is there a difference between the blends? Explain how using blends in the design and modeling improved the outcome.
 - (d) Check the model diagnostics for any problems with the assumptions.
7. Use the `penicillin` data from the previous question. Test for the significance of the treatment effect using a permutation test. Some hints:
 - (a) You will need to restrict the randomization of the permutation to within the blocks.
 - (b) Constructing such a random permutation is possible with repeated use of `sample(4)`.
 - (c) Examine the output of the `anova` function using `names` to learn how to extract the relevant F-statistic.
8. Is the standard Sudoku solution a Latin square? Does the Sudoku solution contain any additional features that might be useful for design purposes? Propose a situation where it could be helpful.

About R

Installation

R may be obtained free of charge from the R project web site at www.r-project.org.

How to learn R

The R website provides extensive resources for learning R. Several free introductory guides can be found along with a list of the many books about R which are now available. I have assumed a basic knowledge of R for readers of this text but it is not necessary to learn it in detail before starting on this book. Many readers are able to pick up the language from the examples in the text with reference to the help pages. For example, for more information on the `lm` function, type within R:

```
> help(lm)
```

More advanced users may find the R mailing lists and online forums such as Stack-Exchange helpful for more difficult questions.

Packages

This book uses some functions and data that are not part of base R. These are collected in *packages* which you may need to install. You can find out which packages are already installed in your *library* by:

```
> library()
```

Some packages such as `MASS`, `mgcv`, `nlme` and `splines` come with the base installation so you will not need to install them. If you have not already done so, you will need to select a location from which to download additional packages by setting your CRAN (Comprehensive R Archive Network) mirror by:

```
> chooseCRANmirror()
```

You can then install packages. You will surely need to install the `faraway` package which contains data and functions specific to this text:

```
> install.packages("faraway")
```

You may also wish install the other packages used in this text which are: `Amelia`, `ellipse`, `ggplot2`, `lars`, `leaps`, `lmtest`, `Matching`, `pls`, `quantreg`, `simex`. All these packages are used infrequently (except for `ggplot2`), so you might delay installation until you need them. Note that you only need to install packages once on a given computer unless you upgrade the version of R. The standard GUIs for Windows and Mac provide menu-based methods for package installation and maintenance which you may find more convenient.

Any time you want to use these particular data or functions, you will need to load the package. For example,

```
> require(faraway)
```

makes the data from this text available. If you get an error message about some data or functions not being found, it's quite possible that you have forgotten to load the necessary package.

Customization

There are several ways to customize the look of R. I set the following options to achieve the output seen in this book:

```
options(width=70,digits=5,scipen=2)
```

The `width=70` controls the width of the printed output. You may wish to increase this, depending on your screen or the width of your page. The `digits=5` reduces the number of digits shown when printing numbers from the default of seven. Note that this does not reduce the precision with which these numbers are internally stored. One might take this further — anything more than two or three significant digits in a displayed table is usually unnecessary and more importantly, distracting. `scipen=2` penalizes the use of scientific notation in reporting numbers. This notation has its uses but can make it harder to appreciate the size of a number.

I have also edited the output in the text to remove extraneous output or to improve the formatting.

Updates and Errata

The code and output shown in this book were generated under R version 3.0.2. R is regularly updated and improved so more recent versions may show some differences in the output. Sometimes these changes can break the code in the text. This is more likely to occur in the additional packages. Please refer to the book website at <http://people.bath.ac.uk/jjf23/LMR/> for information about such changes or errors found in the text.

Interfaces

The R website provides a GUI for both Mac and Windows installations. More functionality can be found elsewhere in ESS (which runs within the Emacs environment) and RStudio. Advanced users of R often find it more convenient to prepare R commands in a separate script before running these in R. Customizations to write R for many popular text editing programs are available.

Reproducible research

It is important that you be able to reproduce your analysis in the future if needed. Sometimes, other people will need to verify your findings or, more likely, you yourself will need to modify or update your analysis. At a minimum you should maintain a commented set of R commands that will reproduce your analysis.

Sometimes you will want produce a document that includes the R commands, text describing the analysis and the output including graphs. This might look something like the text of this book. The *Sweave* tool allows you to embed R within a \LaTeX document. Alternatively, the `knitr` package also provides this along with ability to use formats compatible with Microsoft Word and HTML for web display.

Bibliography

- Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on* 19(6), 716–723.
- Anderson, C. and R. Loynes (1987). *The Teaching of Practical Statistics*. New York: Wiley.
- Anderson, R. and T. Bancroft (1952). *Statistical Theory in Research*. New York: McGraw Hill.
- Andrews, D. and A. Herzberg (1985). *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. New York: Springer-Verlag.
- Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 57, 289–300.
- Berkson, J. (1950). Are there two regressions? *Journal of the American Statistical Association* 45, 165–180.
- Box, G., S. Bisgaard, and C. Fung (1988). An explanation and critique of Taguchi's contributions to quality engineering. *Quality and Reliability Engineering International* 4, 123–131.
- Box, G., W. Hunter, and J. Hunter (1978). *Statistics for Experimenters*. New York: Wiley.
- Bretz, F., T. Hothorn, and P. Westfall (2010). *Multiple comparisons using R*. Boca Raton: Chapman & Hall/CRC.
- Burnham, K. P. and D. R. Anderson (2002). *Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach*. New York: Springer.
- Cook, J. and L. Stefanski (1994). Simulation–extrapolation estimation in parametric measurement error models. *Journal of the American Statistical Association* 89, 1314–1328.
- Davies, O. (1954). *The Design and Analysis of Industrial Experiments*. New York: Wiley.
- Davison, A. C. (1997). *Bootstrap Methods and their Application*. Cambridge: Cambridge University Press.
- de Boor, C. (2002). *A Practical Guide to Splines*. New York: Springer.
- de Jong, S. (1993). SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems* 18, 251–263.

- Draper, N. and H. Smith (1998). *Applied Regression Analysis* (3rd ed.). New York: Wiley.
- Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics* 32, 407–499.
- Efron, B. and R. Tibshirani (1993). *An Introduction to the Bootstrap*. London: Chapman & Hall.
- Faraway, J. (1992). On the cost of data analysis. *Journal of Computational and Graphical Statistics* 1, 215–231.
- Faraway, J. (1994). Order of actions in regression analysis. In P. Cheeseman and W. Oldford (Eds.), *Selecting Models from Data: Artificial Intelligence and Statistics IV*, pp. 403–411. New York: Springer-Verlag.
- Fisher, R. (1936). Has Mendel's work been rediscovered? *Annals of Science* 1, 115–137.
- Frank, I. and J. Friedman (1993). A statistical view of some chemometrics tools. *Technometrics* 35, 109–135.
- Freedman, D. and D. Lane (1983). A nonstochastic interpretation of reported significance levels. *Journal of Business and Economic Statistics* 1(4), 292–298.
- Friedman, J., T. Hastie, and R. Tibshirani (2008). *Elements of Statistical Learning* (2 ed.). New York: Springer.
- Galton, F. (1886). Regression towards mediocrity in hereditary stature. *The Journal of the Anthropological Institute of Great Britain and Ireland* 15, 246–263.
- Garthwaite, P. (1994). An interpretation of partial least squares. *Journal of the American Statistical Association* 89, 122–127.
- Gelman, A. (2008). Scaling regression inputs by dividing by two standard deviations. *Statistics in Medicine* 27(15), 2865–2873.
- Hamada, M. and J. Wu (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. New York: Wiley.
- Herron, M., W. M. Jr, and J. Wand (2008). Voting Technology and the 2008 New Hampshire Primary. *Wm. & Mary Bill Rts. J.* 17, 351–374.
- Hill, A. B. (1965). The environment and disease: association or causation? *Proceedings of the Royal Society of Medicine* 58(5), 295.
- Honaker, J., G. King, and M. Blackwell (2011). Amelia II: A program for missing data. *Journal of Statistical Software* 45(7), 1–47.
- John, P. (1971). *Statistical Design and Analysis of Experiments*. New York: Macmillan.
- Johnson, M. and P. Raven (1973). Species number and endemism: the Galápagos Archipelago revisited. *Science* 179, 893–895.
- Johnson, R. (1996). Fitting percentage of body fat to simple body measurements. *Journal of Statistics Education* 4(1), 265–266.

- Jolliffe, I. (2002). *Principal Component Analysis* (2 ed.). New York: Springer Verlag.
- Jones, P. and M. Mann (2004). Climate over past millennia. *Reviews of Geophysics* 42, 1–42.
- Lentner, M. and T. Bishop (1986). *Experimental Design and Analysis*. Blacksburg, VA: Valley Book Company.
- Little, R. and D. Rubin (2002). *Statistical Analysis with Missing Data* (2nd ed.). New York: Wiley.
- Mazumdar, S. and S. Hoa (1995). Application of a Taguchi method for process enhancement of an online consolidation technique. *Composites* 26, 669–673.
- Morris, R. and E. Watson (1998). A comparison of the techniques used to evaluate the measurement process. *Quality Engineering* 11, 213–219.
- Mortimore, P., P. Sammons, L. Stoll, D. Lewis, and R. Ecob (1988). *School Matters*. Wells: Open Books.
- Muggeo, V. M. (2003). Estimating regression models with unknown break-points. *Statistics in Medicine* 22, 3055–3071.
- Partridge, L. and M. Farquhar (1981). Sexual activity and the lifespan of male fruitflies. *Nature* 294, 580–581.
- Pinheiro, J. and D. Bates (2000). *Mixed-Effects Models in S and S-PLUS*. New York: Springer-Verlag.
- R Core Team (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.
- Raftery, A. E., D. Madigan, and J. A. Hoeting (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* 92(437), 179–191.
- Rodriguez, N., S. Ryan, H. V. Kemp, and D. Foy (1997). Post-traumatic stress disorder in adult female survivors of childhood sexual abuse: A comparison study. *Journal of Consulting and Clinical Psychology* 65, 53–59.
- Rousseeuw, P. and A. Leroy (1987). *Robust Regression and Outlier Detection*. New York: Wiley.
- Schafer, J. (1997). *Analysis of Incomplete Multivariate Data*. London: Chapman & Hall.
- Scheffé, H. (1959). *The Analysis of Variance*. New York: Wiley.
- Sekhon, J. S. (2011). Multivariate and propensity score matching software with automated balance optimization: The Matching package for R. *Journal of Statistical Software* 42(7), 1–52.
- Sen, A. and M. Srivastava (1990). *Regression Analysis: Theory, Methods and Applications*. New York: Springer-Verlag.
- Simonoff, J. (1996). *Smoothing Methods in Statistics*. New York: Springer-Verlag.

- Steel, R. G. and J. Torrie (1980). *Principles and Procedures of Statistics, a Biometrical Approach*. (2nd ed.). New York: McGraw Hill.
- Stigler, S. (1986). *The History of Statistics*. Cambridge, MA: Belknap Press.
- Stolarski, R., A. Krueger, M. Schoeberl, R. McPeters, P. Newman, and J. Alpert (1986). *Nimbus 7* satellite measurements of the springtime Antarctic ozone decrease. *Nature* 322, 808–811.
- Thisted, R. (1988). *Elements of Statistical Computing*. New York: Chapman & Hall.
- Thodberg, H. H. (1993). *Ace of Bayes: Application of neural networks with pruning*. Technical Report 1132E, Maglegaardvej 2, DK-4000 Roskilde, Denmark.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* 58, 267–288.
- Tukey, J. (1977). *Exploratory Data Analysis*. Reading, MA: Addison-Wesley.
- Venables, W. and B. Ripley (2002). *Modern Applied Statistics with S-PLUS* (4th ed.). New York: Springer-Verlag.
- Weisberg, S. (1985). *Applied Linear Regression* (2nd ed.). New York: Wiley.
- Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer-Verlag.
- Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Boca Raton: Chapman & Hall/CRC.

Statistics

Part of the core of statistics, linear models are used to make predictions and explain the relationship between the response and the predictors. Understanding linear models is crucial to a broader competence in the practice of statistics. **Linear Models with R, Second Edition** explains how to use linear models in physical science, engineering, social science, and business applications. The book incorporates several improvements that reflect how the world of R has greatly expanded since the publication of the first edition.

New to the Second Edition

- Reorganized material on interpreting linear models, which distinguishes the main applications of prediction and explanation and introduces elementary notions of causality
- Additional topics, including QR decomposition, splines, additive models, Lasso, multiple imputation, and false discovery rates
- Extensive use of the ggplot2 graphics package in addition to base graphics

Like its widely praised, best-selling predecessor, this edition combines statistics and R to seamlessly give a coherent exposition of the practice of linear modeling. The text offers up-to-date insight on essential data analysis topics, from estimation, inference, and prediction to missing data, factorial models, and block designs. Numerous examples illustrate how to apply the different methods using R.

Features

- Demonstrates the flexibility of linear models in many examples
- Assumes basic knowledge of R and statistics
- Emphasizes intuition over rigorous proofs
- Presents exercises at the end of each chapter
- Includes datasets and R commands

K14039



CRC Press
Taylor & Francis Group
an **informa** business
www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
711 Third Avenue
New York, NY 10017
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

ISBN: 978-1-4398-8733-2



9 781439 887332