Practical Regression and Anova using R

Julian J. Faraway

July 2002

Preface

There are many books on regression and analysis of variance. These books expect different levels of preparedness and place different emphases on the material. This book is not introductory. It presumes some knowledge of basic statistical theory and practice. Students are expected to know the essentials of statistical inference like estimation, hypothesis testing and confidence intervals. A basic knowledge of data analysis is presumed. Some linear algebra and calculus is also required.

The emphasis of this text is on the practice of regression and analysis of variance. The objective is to learn what methods are available and more importantly, when they should be applied. Many examples are presented to clarify the use of the techniques and to demonstrate what conclusions can be made. There is relatively less emphasis on mathematical theory, partly because some prior knowledge is assumed and partly because the issues are better tackled elsewhere. Theory is important because it guides the approach we take. I take a wider view of statistical theory. It is not just the formal theorems. Qualitative statistical concepts are just as important in Statistics because these enable us to actually do it rather than just talk about it. These qualitative principles are harder to learn because they are difficult to state precisely but they guide the successful experienced Statistician.

Data analysis cannot be learnt without actually doing it. This means using a statistical computing package. There is a wide choice of such packages. They are designed for different audiences and have different strengths and weaknesses. I have chosen to use R (ref. Ihaka and Gentleman (1996)). Why do I use R ? The are several reasons.

1. Versatility. R is a also a programming language, so I am not limited by the procedures that are preprogrammed by a package. It is relatively easy to program new methods in R .

2. Interactivity. Data analysis is inherently interactive. Some older statistical packages were designed when computing was more expensive and batch processing of computations was the norm. Despite improvements in hardware, the old batch processing paradigm lives on in their use. R does one thing at a time, allowing us to make changes on the basis of what we see during the analysis.

3. R is based on S from which the commercial package S-plus is derived. R itself is open-source software and may be freely redistributed. Linux, Macintosh, Windows and other UNIX versions are maintained and can be obtained from the R-project at `www.r-project.org`. R is mostly compatible with S-plus meaning that S-plus could easily be used for the examples given in this book.

4. Popularity. SAS is the most common statistics package in general but R or S is most popular with researchers in Statistics. A look at common Statistical journals confirms this popularity. R is also popular for quantitative applications in Finance.

The greatest disadvantage of R is that it is not so easy to learn. Some investment of effort is required before productivity gains will be realized. This book is not an introduction to R . There is a short introduction

in the Appendix but readers are referred to the R-project web site at www.r-project.org where you can find introductory documentation and information about books on R . I have intentionally included in the text all the commands used to produce the output seen in this book. This means that you can reproduce these analyses and experiment with changes and variations before fully understanding R . The reader may choose to start working through this text before learning R and pick it up as you go.

The web site for this book is at www.stat.lsa.umich.edu/~faraway/book where data described in this book appears. Updates will appear there also.

Thanks to the builders of R without whom this book would not have been possible.

Contents

Chapter 1

Introduction

1.1 Before you start

Statistics starts with a problem, continues with the collection of data, proceeds 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 what the objectives are or even whether the data are appropriate for the proposed analysis. Look before you leap!

1.1.1 Formulation

> The formulation of a problem is often more essential than its solution which may be merely a matter of mathematical or experimental skill. *Albert Einstein*

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'll almost always find something but that something may just be a coincidence.

3. Make sure you know what the client wants. 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 may be totally meaningless.

1.1.2 Data Collection

It's important to understand how the data was collected.

- 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.
- Is there non-response? The data you don't see may be just as important as the data you do see.
- Are there missing values? This is a common problem that is troublesome and time consuming to deal with.
- How are the data coded? In particular, how are the qualitative variables represented.
- What are the units of measurement? Sometimes data is collected or represented with far more digits than are necessary. Consider rounding if this will help with the interpretation or storage costs.
- Beware of data entry errors. This problem is all too common — almost a certainty in any real dataset of at least moderate size. Perform some data sanity checks.

1.1.3 Initial Data Analysis

This is a critical step that should always be performed. It looks simple but it is vital.

- Numerical summaries - means, sds, five-number summaries, correlations.
- Graphical summaries
  - One variable - Boxplots, histograms etc.
  - Two variables - scatterplots.
  - Many variables - interactive graphics.

Look for outliers, data-entry errors and skewed or unusual distributions. Are the data distributed as you expect?

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 course, 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 a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in kg/(height in m^2)), Diabetes pedigree function, Age (years) and a test whether the patient shows signs of diabetes (coded 0 if negative, 1 if positive). The data may be obtained from UCI Repository of machine learning databases at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Of course, before doing anything else, one should find out what the purpose of the study was and more about how the data was collected. But let's skip ahead to a look at the data:

```
> library(faraway)
> data(pima)
> 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
... much deleted ...
768        1      93        70      31       0 30.4    0.315  23    0
```

The `library(faraway)` makes the data used in this book available while `data(pima)` calls up this particular dataset. Simply typing the name of the *data frame*, `pima` prints out the data. It's too long to show it all here. For a dataset of this size, one can just about visually skim over the data for anything out of place but it is certainly easier to use more direct methods.

We start with some numerical summaries:

```
> summary(pima)
    pregnant        glucose      diastolic        triceps         insulin
 Min.   : 0.00   Min.   :  0   Min.   :  0.0   Min.   : 0.0   Min.   :  0.0
 1st Qu.: 1.00   1st Qu.: 99   1st Qu.: 62.0   1st Qu.: 0.0   1st Qu.:  0.0
 Median : 3.00   Median :117   Median : 72.0   Median :23.0   Median : 30.5
 Mean   : 3.85   Mean   :121   Mean   : 69.1   Mean   :20.5   Mean   : 79.8
 3rd Qu.: 6.00   3rd Qu.:140   3rd Qu.: 80.0   3rd Qu.:32.0   3rd Qu.:127.2
 Max.   :17.00   Max.   :199   Max.   :122.0   Max.   :99.0   Max.   :846.0
      bmi          diabetes          age            test
 Min.   : 0.0   Min.   :0.078   Min.   :21.0   Min.   :0.000
 1st Qu.:27.3   1st Qu.:0.244   1st Qu.:24.0   1st Qu.:0.000
 Median :32.0   Median :0.372   Median :29.0   Median :0.000
 Mean   :32.0   Mean   :0.472   Mean   :33.2   Mean   :0.349
 3rd Qu.:36.6   3rd Qu.:0.626   3rd Qu.:41.0   3rd Qu.:1.000
 Max.   :67.1   Max.   :2.420   Max.   :81.0   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 perhaps not impossible. However, we then see that the next 5 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
 [19]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  24
 [37]  30  30  38  40  44  44  44  44  46  46  48  48  48  48  48  50  50  50
...etc...
```

We see that the first 36 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 36 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 observed zeroes. If the error was later discovered, they might then blame the researchers for using 0 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's best to designate such variables as factors so that they are treated appropriately. Sometimes people forget this and compute stupid statistics such as "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 positive. Even better is to use descriptive labels:

```
> levels(pima$test) <- c("negative","positive")
> summary(pima)
    pregnant        glucose      diastolic        triceps         insulin
 Min.   : 0.00   Min.   : 44   Min.   : 24.0   Min.   :  7.0   Min.   : 14.0
 1st Qu.: 1.00   1st Qu.: 99   1st Qu.: 64.0   1st Qu.: 22.0   1st Qu.: 76.2
 Median : 3.00   Median :117   Median : 72.0   Median : 29.0   Median :125.0
 Mean   : 3.85   Mean   :122   Mean   : 72.4   Mean   : 29.2   Mean   :155.5
 3rd Qu.: 6.00   3rd Qu.:141   3rd Qu.: 80.0   3rd Qu.: 36.0   3rd Qu.:190.0
 Max.   :17.00   Max.   :199   Max.   :122.0   Max.   : 99.0   Max.   :846.0
                 NA's   :  5   NA's   : 35.0   NA's   :227.0   NA's   :374.0
      bmi          diabetes          age            test
 Min.   :18.2   Min.   :0.078   Min.   :21.0   negative:500
 1st Qu.:27.5   1st Qu.:0.244   1st Qu.:24.0   positive:268
 Median :32.3   Median :0.372   Median :29.0
 Mean   :32.5   Mean   :0.472   Mean   :33.2
 3rd Qu.:36.6   3rd Qu.:0.626   3rd Qu.:41.0
 Max.   :67.1   Max.   :2.420   Max.   :81.0
 NA's   :11.0
```

Now that we've 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:

```
hist(pima$diastolic)
```

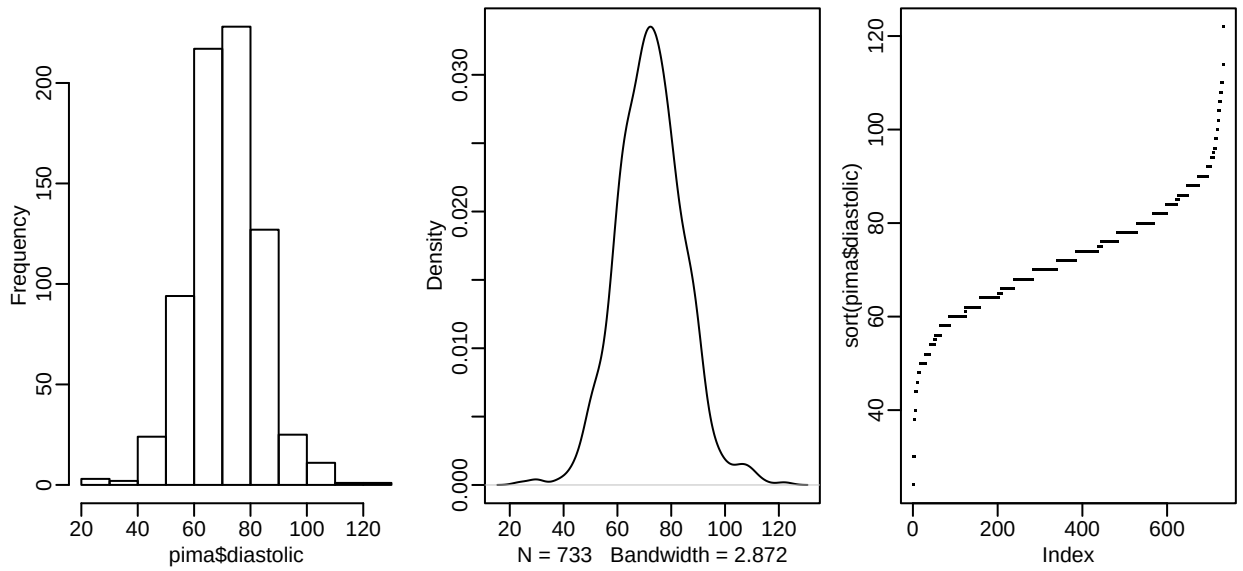

Figure 1.1: First panel shows histogram of the diastolic blood pressures, the second shows a kernel density estimate of the same while the the third shows an index plot of the sorted values

as shown 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 attempts to specify 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, I 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))
```

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

```
plot(sort(pima$diastolic),pch=".")
```

The advantage of this is we can see all the data points themselves. 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 the “steps” in the plot.

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

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

```
hist(pima$diastolic)
```

First, we see the standard scatterplot showing two quantitative variables. Second, we see a side-by-side boxplot suitable for showing a quantitative and a qualititative variable. Also useful is a scatterplot matrix, not shown here, produced by

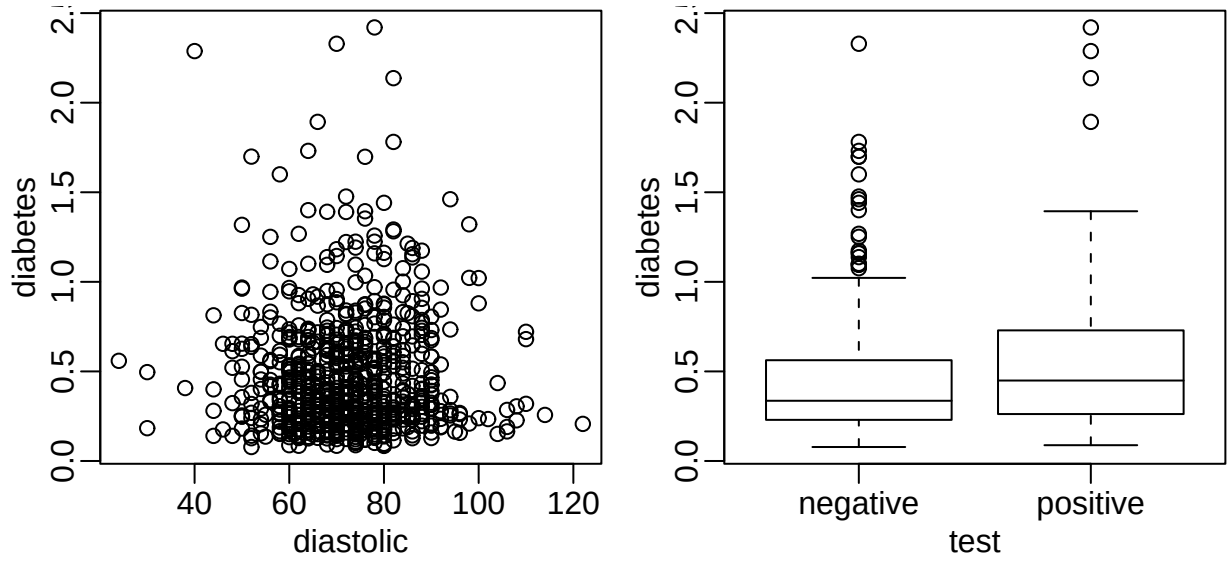

Figure 1.2: First panel shows scatterplot of the diastolic blood pressures against diabetes function and the second shows boxplots of diastolic blood pressure broken down by test result

```
> pairs(pima)
```

We will be seeing more advanced plots later but the numerical and graphical summaries presented here are sufficient for a first look at the data.

1.2 When to use Regression Analysis

Regression analysis is used for explaining or modeling the relationship between a single variable Y, called the *response*, *output* or *dependent* variable, and one or more *predictor*, *input*, *independent* or *explanatory* variables, $X_1, \ldots, X_p$. 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 Y, then it is called multivariate multiple regression which we won't be covering here.

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 course. Taking the example presented above, a regression of `diastolic` and `bmi` on `diabetes` would be a multiple regression involving only quantitative variables which we shall be tackling shortly. A regression of `diastolic` and `bmi` on `test` would involve one predictor which is quantitative which we will consider in later in the chapter on *Analysis of Covariance*. A regression of `diastolic` on just `test` would involve just qualitative predictors, a topic called *Analysis of Variance* or *ANOVA* although this would just be a simple two sample situation. A regression of `test` (the response) on `diastolic` and `bmi` (the predictors) would involve a qualitative response. A *logistic regression* could be used but this will not be covered in this book.

Regression analyses have several possible objectives including

1. Prediction of future observations.
2. Assessment of the effect of, or relationship between, explanatory variables on the response.
3. A general description of data structure.

Extensions exist to handle multivariate responses, binary responses (logistic regression analysis) and count responses (poisson regression).

1.3 History

Regression-type problems were first considered in the 18th century concerning navigation using astronomy. Legendre developed the method of least squares in 1805. Gauss claimed to have developed the method a few years earlier and showed that the least squares was the optimal solution when the errors are normally distributed in 1809. The methodology was used almost exclusively in the physical sciences until later in the 19th century. Francis Galton coined the term *regression to mediocrity* in 1875 in reference to the simple regression equation in the form

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

Galton used this equation to explain the phenomenon that sons of tall fathers tend to be tall but not as tall as their fathers while sons of short fathers tend to be short but not as short as their fathers. This effect is called the *regression effect*.

We can illustrate this effect with some data on scores from a course taught using this book. In Figure 1.3, we see a plot of midterm against final scores. We scale each variable to have mean 0 and SD 1 so that we are not distracted by the relative difficulty of each exam and the total number of points possible. Furthermore, this simplifies the regression equation to

$$y = rx$$

```
> data(stat500)
> stat500 <- data.frame(scale(stat500))
> plot(final ~ midterm,stat500)
> abline(0,1)
```

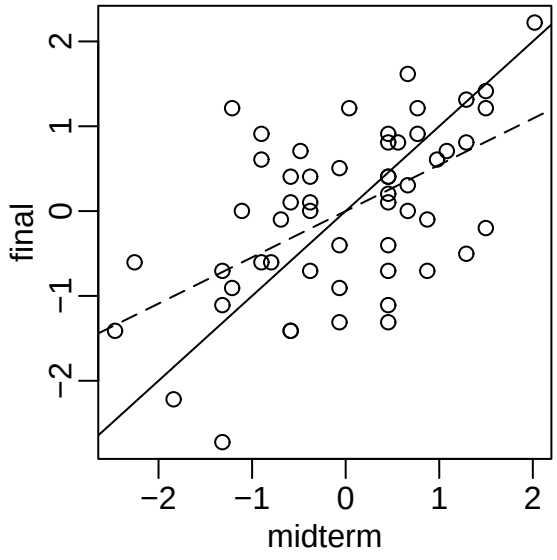

Figure 1.3: Final and midterm scores in standard units. Least squares fit is shown with a dotted line while $y = x$ is shown as a solid line

We have added the $y = x$ (solid) line to the plot. Now a student scoring, say one standard deviation above average on the midterm might reasonably expect to do equally well on the final. We compute the least squares regression fit and plot the regression line (more on the details later). We also compute the correlations.

```
> g <- lm(final ~ midterm,stat500)
> abline(g$coef,lty=5)
> cor(stat500)
         midterm    final       hw   total
midterm 1.00000 0.545228 0.272058 0.84446
final   0.54523 1.000000 0.087338 0.77886
hw      0.27206 0.087338 1.000000 0.56443
total   0.84446 0.778863 0.564429 1.00000
```

We see that the the student scoring 1 SD above average on the midterm is predicted to score somewhat less above average on the final (see the dotted regression line) - 0.54523 SD's above average to be exact. Correspondingly, a student scoring below average on the midterm might expect to do relatively better in the final although still below average.

If exams managed to measure the ability of students perfectly, then provided that ability remained unchanged from midterm to final, we would expect to see a perfect correlation. Of course, it's too much to expect such a perfect exam and some variation is inevitably present. Furthermore, individual effort is not constant. Getting a high score on the midterm can partly be attributed to skill but also a certain amount of luck. One cannot rely on this luck to be maintained in the final. Hence we see the "regression to mediocrity".

Of course this applies to any (x,y) situation like this — an example is the so-called sophomore jinx in sports when a rookie star has a so-so second season after a great first year. Although in the father-son example, it does predict that successive descendants will come closer to the mean, it does not imply the same of the population in general since random fluctuations will maintain the variation. In many other applications of regression, the regression effect is not of interest so it is unfortunate that we are now stuck with this rather misleading name.

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, then the scope for analysis has widened.

Chapter 2

Estimation

2.1 Example

Let's start with an example. Suppose that Y is the fuel consumption of a particular model of car in m.p.g. Suppose that the predictors are

1. X_1 — the weight of the car
2. X_2 — the horse power
3. X_3 — the no. of cylinders.

X_3 is discrete but that's OK. Using country of origin, say, as a predictor would not be possible within the current development (we will see how to do this later in the course). Typically the data will be available in the form of an array like this

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

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

2.2 Linear Model

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 which is additive in this instance. Since we usually don't have enough data to try to estimate f directly, 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*. β_0 is called the *intercept* term. Thus the problem is reduced to the estimation of four values rather than the complicated infinite dimensional f.

In a linear model the *parameters enter linearly* — the predictors do not have to be linear. For example

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

is linear but

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

is not. Some relationships can be transformed to linearity — for example $y = \beta_0 x_1^{\beta} \varepsilon$ can be linearized by taking logs. Linear models seem rather restrictive but because the predictors can transformed and combined in any way, they are actually very flexible. Truly non-linear models are rarely absolutely necessary and most often arise from a theory about the relationships between the variables rather than an empirical investigation.

2.3 Matrix Representation

Given the actual data, we may write

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i \quad i = 1, \ldots, n$$

but the use of subscripts becomes inconvenient and conceptually obscure. We will find it simpler both notationally and theoretically to use a matrix/vector representation. The regression equation is written as

$$y = X\beta + \varepsilon$$

where $y = (y_1 \ldots y_n)^T$, $\varepsilon = (\varepsilon_1 \ldots \varepsilon_n)^T$, $\beta = (\beta_0 \ldots \beta_3)^T$ and

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

The column of ones incorporates the intercept term. A couple of examples of using this notation are the simple no predictor, mean only model $y = \mu + \varepsilon$

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

We can assume that $E\varepsilon = 0$ since if this were not so, we could simply absorb the non-zero expectation for the error into the mean μ to get a zero expectation. For the two sample problem with a treatment group having the response $y_1, \ldots, y_m$ with mean μ_y and control group having response $z_1, \ldots, z_n$ with mean μ_z we have

$$\begin{pmatrix} y_1 \\ \ldots \\ y_m \\ z_1 \\ \ldots \\ z_n \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \ldots & \\ 1 & 0 \\ 0 & 1 \\ . & . \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mu_y \\ \mu_z \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \ldots \\ \ldots \\ \ldots \\ \varepsilon_{m+n} \end{pmatrix}$$

2.4 Estimating β

We have the regression equation $y = X\beta + \varepsilon$ - what estimate of β would best separate the systematic component $X\beta$ from the random component ε. Geometrically speaking, $y \in \mathbb{R}^n$ 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).

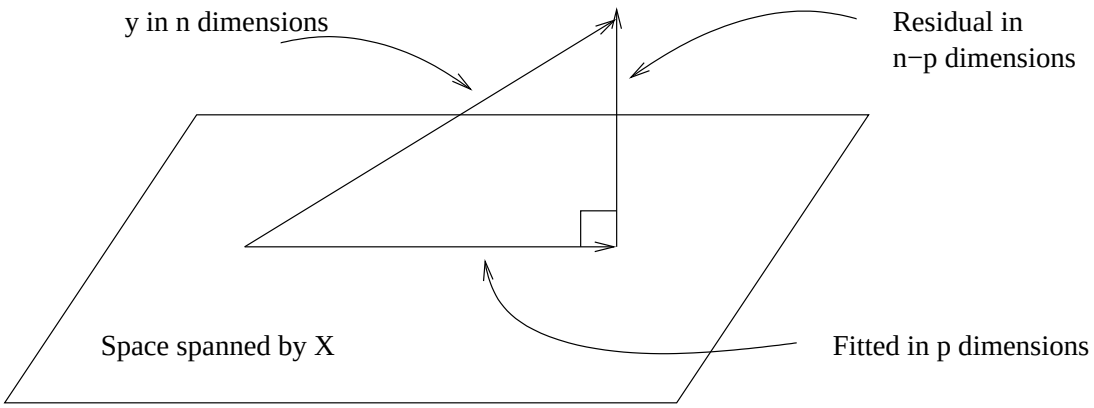

Figure 2.1: Geometric representation of the estimation β. The data vector Y is projected orthogonally onto the model space spanned by X. The fit is represented by projection $\hat{y} = X\hat{\beta}$ with the difference between the fit and the data represented by the residual vector $\hat{\varepsilon}$.

The problem is to find β such that $X\beta$ is close to Y. The best choice of $\hat{\beta}$ is apparent in the geometrical representation shown in Figure 2.4.

$\hat{\beta}$ is in some sense the best estimate of β within the model space. The response predicted by the model is $\hat{y} = X\hat{\beta}$ or Hy where H is an orthogonal projection matrix. The difference between the actual response and the predicted response is denoted by $\hat{\varepsilon}$ — the residuals.

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

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

2.5 Least squares estimation

The estimation of β can be considered from a non-geometric point of view. We might define the best estimate of β as that which minimizes the sum of the squared errors, $\varepsilon^T\varepsilon$. That is to say that the least squares estimate of β, called $\hat{\beta}$ minimizes

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

Expanding this out, we get

$$y^Ty - 2\beta X^Ty + \beta^T X^T X\beta$$

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

$$X^TX\hat{\beta} = X^Ty$$

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

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

$H = X(X^TX)^{-1}X^T$ 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 don't want to compute it explicitly as it is an $n \times n$ matrix.

- Predicted values: $\hat{y} = Hy = X\hat{\beta}$.
- Residuals: $\hat{\varepsilon} = y - X\hat{\beta} = y - \hat{y} = (I-H)y$
- Residual sum of squares: $\hat{\varepsilon}^T\hat{\varepsilon} = y^T(I-H)(I-H)y = y^T(I-H)y$

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. var $\varepsilon = \sigma^2 I$.

2.6 Examples of calculating $\hat{\beta}$

1. When $y = \mu + \varepsilon$, $X = \mathbf{1}$ and $\beta = \mu$ so $X^TX = \mathbf{1}^T\mathbf{1} = n$ so
$$\hat{\beta} = (X^TX)^{-1}X^Ty = \frac{1}{n}\mathbf{1}^Ty = \bar{y}$$
2. Simple linear regression (one predictor)
$$y_i = \alpha + \beta x_i + \varepsilon_i$$
$$\begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \dots & \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \dots \\ \varepsilon_n \end{pmatrix}$$
We can now apply the formula but a simpler approach is to rewrite the equation as
$$y_i = \overbrace{\alpha + \beta\bar{x}}^{\alpha'} + \beta(x_i - \bar{x}) + \varepsilon_i$$
so now
$$X = \begin{pmatrix} 1 & x_1 - \bar{x} \\ \dots & \\ 1 & x_n - \bar{x} \end{pmatrix} \qquad X^TX = \begin{pmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}$$

Now work through the rest of the calculation to reconstruct the familiar estimates, i.e.
$$\hat{\beta} = \frac{\sum (x_i - \bar{x})y}{\sum (x_i - \bar{x})^2}$$

In higher dimensions, it is usually not possible to find such explicit formulae for the parameter estimates unless X^TX happens to be a simple form.

2.7 Why is $\hat{\beta}$ a good estimate?

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 it is best linear unbiased estimate. (BLUE).

2.8 Gauss-Markov Theorem

First we 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^Ty such that

$$Ea^Ty = c^T\beta \qquad \forall\beta$$

Estimable functions include predictions of future observations which explains why they are worth considering. If X is of full rank (which it usually is for observational data), then all linear combinations are estimable.

Gauss-Markov theorem

Suppose $E\varepsilon = 0$ and var $\varepsilon = \sigma^2 I$. Suppose also that the structural part of the model, $EY = X\beta$ is correct. Let $\psi = c^T\beta$ be an estimable function, then in the class of all unbiased linear estimates of ψ, $\hat{\psi} = c^T\hat{\beta}$ has the minimum variance and is unique.

Proof:

We start with a preliminary calculation:

Suppose a^Ty is some unbiased estimate of $c^T\beta$ so that

$$\begin{aligned} Ea^Ty &= c^T\beta \qquad \forall\beta \\ a^TX\beta &= c^T\beta \qquad \forall\beta \end{aligned}$$

which means that $a^TX = c^T$. 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^TX which means there exists a λ such that

$$\begin{aligned} c &= X^TX\lambda \\ c^T\hat{\beta} &= \lambda^TX^TX\hat{\beta} = \lambda^TX^Ty \end{aligned}$$

Now we can show that the least squares estimator has the minimum variance — pick an arbitrary estimable function a^Ty and compute its variance:

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

but

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

so

$$\text{var}\,(a^Ty) = \text{var}\,(a^Ty - \lambda^TX^Ty) + \text{var}\,(c^T\hat{\beta})$$

Now since variances cannot be negative, we see that

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

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

Implications

The Gauss-Markov theorem shows that the least squares estimate $\hat{\beta}$ is a good choice, but if the errors are correlated or have unequal variance, there will be better estimators. Even if the errors behave but are non-normal then non-linear or biased estimates may work better in some sense. 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.
2. When the error distribution is long-tailed, then robust estimates might be used. Robust estimates are typically not linear in y.
3. When the predictors are highly correlated (collinear), then biased estimators such as ridge regression might be preferable.

2.9 Mean and Variance of $\hat{\beta}$

Now $\hat{\beta} = (X^T X)^{-1} X^T y$ so

- Mean $E\hat{\beta} = (X^T X)^{-1} X^T X\beta = \beta$ (unbiased)
- var $\hat{\beta} = (X^T X)^{-1} X^T \sigma^2 I X (X^T X)^{-1} = (X^T X)^{-1}\sigma^2$

Note that since $\hat{\beta}$ is a vector, $(X^T X)^{-1}\sigma^2$ is a variance-covariance matrix. Sometimes you want the standard error for a particular component which can be picked out as in $se(\hat{\beta}_i) = \sqrt{(X^T X)^{-1}_{ii}}\hat{\sigma}$.

2.10 Estimating σ^2

Recall that the residual sum of squares was $\hat{\varepsilon}^T\hat{\varepsilon} = y^T(I-H)y$. Now after some calculation, one can show that $E\hat{\varepsilon}^T\hat{\varepsilon} = \sigma^2(n-p)$ which shows that

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

is an unbiased estimate of σ^2. $n-p$ is the *degrees of freedom* of the model. Actually a theorem parallel to the Gauss-Markov theorem shows that it has the minimum variance among all quadratic unbiased estimators of σ^2.

2.11 Goodness of Fit

How well does the model fit the data? One measure is R^2, the so-called *coefficient of determination* or *percentage of variance explained*

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

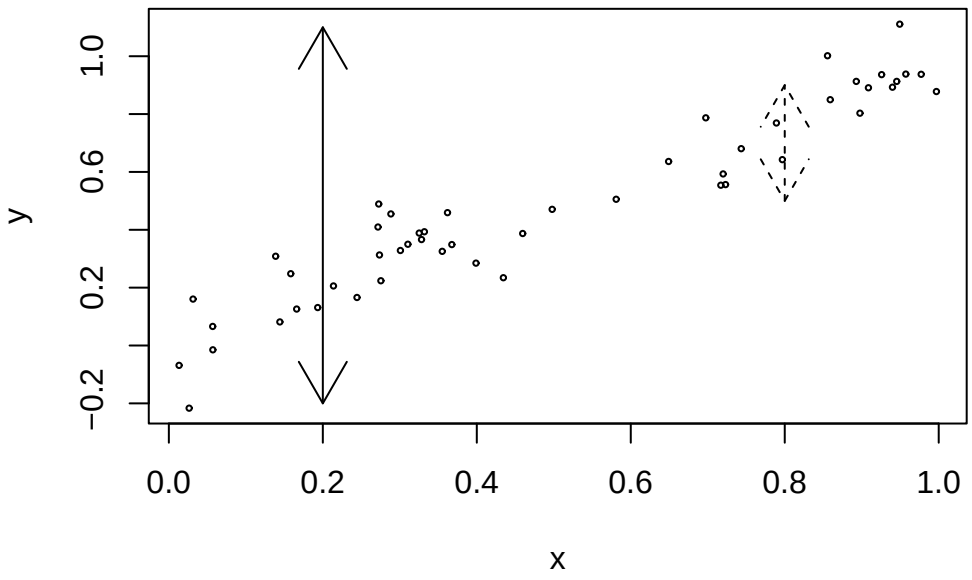

Figure 2.2: Variation in the response y when x is known is denoted by dotted arrows while variation in y when x is unknown is shown with the solid arrows

The 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

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

The graphical intuition behind R^2 is shown in Figure 2.2. Suppose you want to predict y. If you don't 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 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.

Warning: R^2 as defined here doesn't make any sense if you do not have an intercept in your model. This is because the denominator in the definition of R^2 has a null model with an intercept in mind when the sum of squares is calculated. Alternative definitions of R^2 are possible when there is no intercept but the same graphical intuition is not available and the R^2's obtained should not be compared to those for models with an intercept. Beware of high R^2's reported from models without an 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'd expect lower values for R^2 in these areas — a value of 0.6 might be considered good. In physics and engineering, where most data comes from closely controlled experiments, we expect to get much higher R^2's and a value of 0.6 would be considered low. Of course, I generalize excessively here so some experience with the particular area is necessary for you to judge your R^2's well.

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 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 whether the practical significance of this measure whereas R^2, being unitless, is easy to understand.

2.12 Example

Now let's look at an example concerning the number of species of tortoise on the various Galapagos Islands. There are 30 cases (Islands) and 7 variables in the dataset. We start by reading the data into R and examining it

```
> data(gala)
> gala
             Species Endemics    Area Elevation Nearest Scruz Adjacent
      Baltra      58       23   25.09       346     0.6   0.6     1.84
   Bartolome      31       21    1.24       109     0.6  26.3   572.33

--- cases deleted ---

     Tortuga      16        8    1.24       186     6.8  50.9    17.95
        Wolf      21       12    2.85       253    34.1 254.7     2.33
```

The variables are

`Species` The number of species of tortoise found on the island

`Endemics` The number of endemic species

`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 (km^2)

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 14 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 the so-called *Wilkinson-Rogers* notation. In this case, since all the variables are in the gala data frame, we must use the `data=` argument:

```
> gfit <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
     data=gala)
> summary(gfit)
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  3.8e-06
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

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 degrees of freedom,       p-value: 6.84e-07
```

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 pre-packaged functions.

First we make the X-matrix

```
> x <- cbind(1,gala[,-c(1,2)])
```

and here's the response y:

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

Now let's construct X^TX: `t()` does transpose and `%*%` does matrix multiplication:

```
> t(x) %*% x
Error: %*% requires numeric matrix/vector arguments
```

Gives a somewhat cryptic error. The problem is that matrix arithmetic can only be done with numeric values but `x` here derives from the data frame type. Data frames are allowed to contain character variables which would disallow matrix arithmetic. We need to force x into the matrix form:

```
> x <- as.matrix(x)
> t(x) %*% x
```

Inverses can be taken using the `solve()` command:

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

A somewhat more direct way to get $(X^TX)^{-1}$ is as follows:

```
> gfit <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
                       data=gala)
> gs <- summary(gfit)
> gs$cov.unscaled
```

The `names()` command is the way to see the components of an Splus object - you can see that there are other useful quantities that are directly available:

```
> names(gs)
> names(gfit)
```

In particular, the fitted (or predicted) values and residuals are

```
> gfit$fit
> gfit$res
```

We can get $\hat{\beta}$ directly:

```
> xtxi %*% t(x) %*% y
          [,1]
[1,]  7.068221
[2,] -0.023938
[3,]  0.319465
[4,]  0.009144
[5,] -0.240524
[6,] -0.074805
```

or in a computationally efficient and stable manner:

```
> solve(t(x) %*% x, t(x) %*% y)
          [,1]
[1,]  7.068221
[2,] -0.023938
[3,]  0.319465
[4,]  0.009144
[5,] -0.240524
[6,] -0.074805
```

We can estimate σ using the estimator in the text:

```
> sqrt(sum(gfit$resˆ2)/(30-6))
[1] 60.975
```

Compare this to the results above.

We may also obtain the standard errors for the coefficients. Also `diag()` returns the diagonal of a matrix):

```
> sqrt(diag(xtxi))*60.975
[1] 19.154139  0.022422  0.053663  1.054133  0.215402  0.017700
```

Finally we may compute R^2:

```
> 1-sum(gfit$resˆ2)/sum((y-mean(y))ˆ2)
[1] 0.76585
```

Chapter 3

Inference

Up till now, we haven't found it necessary to assume any distributional form for the errors ε. However, if we want to make any confidence intervals or perform any hypothesis tests, we will need to do this. The usual assumption is that the errors are normally distributed and in practice this is often, although not always, a reasonable assumption. We'll assume that the errors are independent and identically normally distributed with mean 0 and variance σ^2, i.e.

$$\varepsilon \sim N(0, \sigma^2 I)$$

We can handle non-identity variance matrices provided we know the form — see the section on generalized least squares later. Now since $y = X\beta + \varepsilon$,

$$y \sim N(X\beta, \sigma^2 I)$$

is a compact description of the regression model and from this we find that (using the fact that linear combinations of normally distributed values are also normal)

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

3.1 Hypothesis tests to compare models

Given several predictors for a response, we might wonder whether all are needed. Consider a large model, Ω, and a smaller model, ω, which consists of a subset of the predictors that are in Ω. By the principle of Occam's Razor (also known as the law of parsimony), we'd prefer to use ω if the data will support it. So we'll take ω to represent the null hypothesis and Ω to represent the alternative. A geometric view of the problem may be seen in Figure 3.1.

If $\mathrm{RSS}_\omega - \mathrm{RSS}_\Omega$ is small, then ω is an adequate model relative to Ω. This suggests that something like

$$\frac{\mathrm{RSS}_\omega - \mathrm{RSS}_\Omega}{\mathrm{RSS}_\Omega}$$

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

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 likelihood function, then the likelihood ratio statistic is

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

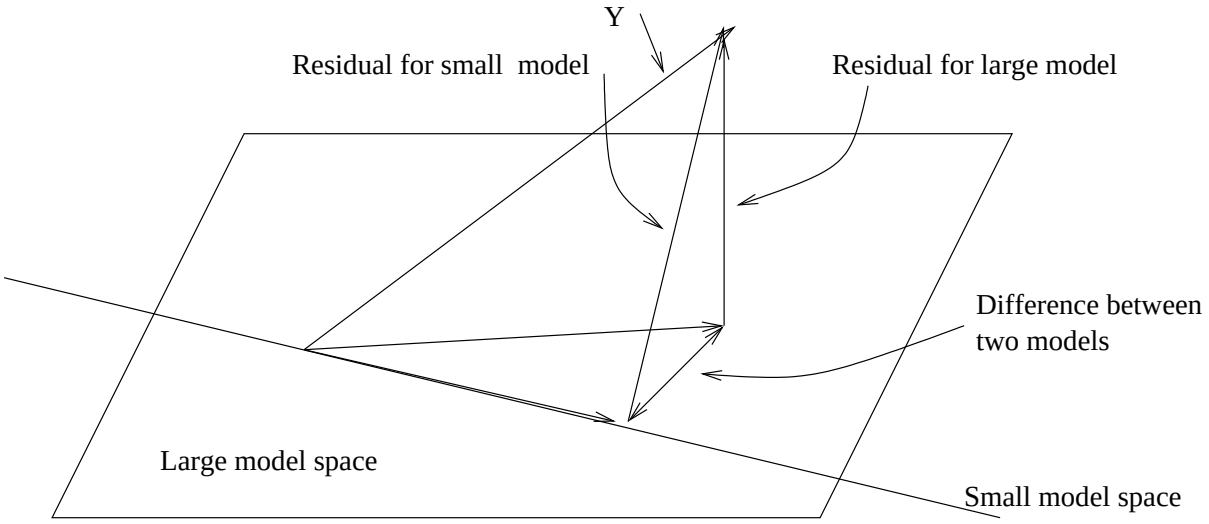

Figure 3.1: Geometric view of the comparison between big model, Ω, and small model, ω. The squared length of the residual vector for the big model is RSS_Ω while that for the small model is RSS_ω. By Pythagoras' theorem, the squared length of the vector connecting the two fits is $RSS_\omega - RSS_\Omega$. 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.

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

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

which gives us a test that rejects if

$$\frac{\hat{\sigma}^2_\omega}{\hat{\sigma}^2_\Omega} > \text{a constant}$$

which is equivalent to

$$\frac{\text{RSS}_\omega}{\text{RSS}_\Omega} > \text{a constant}$$

(constants are not the same) or

$$\frac{\text{RSS}_\omega}{\text{RSS}_\Omega} - 1 > \text{a constant} - 1$$

which is

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

which is the same statistics suggested by the geometric view. It remains for us to discover the null distribution of this statistic.

Now suppose that the dimension (no. of parameters) of Ω is q and dimension of ω is p. Now by Cochran's theorem, if the null (ω) is true then

$$\frac{\text{RSS}_\omega - \text{RSS}_\Omega}{q-p} \sim \sigma^2 \chi^2_{q-p} \qquad \frac{\text{RSS}_\Omega}{n-q} \sim \sigma^2 \chi^2_{n-q}$$

and these two quantities are independent. So we find that

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

Thus we would reject the null hypothesis if $F > F^{(\alpha)}_{q-p,n-q}$ The degrees of freedom of a model is (usually) the number of observations minus the number of parameters so this test statistic can be written

$$F = \frac{(\mathrm{RSS}_\omega - \mathrm{RSS}_\Omega)/(df_\omega - df_\Omega)}{\mathrm{RSS}_\Omega/df_\Omega}$$

where $df_\Omega = n-q$ and $df_\omega = n-p$. The same test statistic applies not just 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 re-expressed in various different 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 represents the null and alternative hypotheses, fit them and compute the test statistic. It is very versatile.

3.2 Some Examples

3.2.1 Test of all predictors

Are any of the predictors useful in predicting the response?

- Full model (Ω) : $y = X\beta + \varepsilon$ where X is a full-rank $n \times p$ matrix.
- Reduced model (ω) : $y = \mu + \varepsilon$ — predict y by the mean.

We could write the null hypothesis in this case as

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

Now

- $\mathrm{RSS}_\Omega = (y - X\hat{\beta})^T(y - X\hat{\beta}) = \hat{\varepsilon}^T\hat{\varepsilon} = \mathrm{RSS}$
- $\mathrm{RSS}_\omega = (y - \bar{y})^T(y - \bar{y})$ =SYY, which is sometimes known as the sum of squares corrected for the mean.

So in this case

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

We'd 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 table*. Most computer packages produce a variant on this. See Table 3.1. 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 some purpose but it is less useful now.

A failure to reject the null hypothesis is not the end of the game — you must still investigate the possibility of non-linear 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.

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	SYY		

Table 3.1: Analysis of Variance table

When the null is rejected, this does not imply that the alternative model is the best model. We don't know whether all the predictors are required to predict the response or just some of them. Other predictors might also be added — for example quadratic terms in the existing predictors. Either way, the overall F-test is just the beginning of an analysis and not the end.

Let's illustrate this test and others using an old economic dataset on 50 different countries. These data are averages over 1960-1970 (to remove business cycle or other short-term fluctuations). `dpi` is per-capita disposable income in U.S. dollars; `ddpi` is the percent rate of change in per capita disposable income; `sr` is aggregate personal saving divided by disposable income. The percentage population under 15 (`pop15`) and over 75 (`pop75`) are also recorded. The data come from Belsley, Kuh, and Welsch (1980). Take a look at the data:

```
> data(savings)
> savings
                sr pop15 pop75     dpi  ddpi
Australia    11.43 29.35  2.87 2329.68  2.87
Austria      12.07 23.32  4.41 1507.99  3.93
--- cases deleted ---
Malaysia      4.71 47.20  0.66  242.69  5.08
```

First consider a model with all the predictors:

```
> g <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings)
> summary(g)

Coefficients:
             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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

We can see directly the result of the test of whether any of the predictors have significance in the model. In other words, whether $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$. Since the p-value is so small, this null hypothesis is rejected.

We can also do it directly using the F-testing formula:

```
> sum((savings$sr-mean(savings$sr))^2)
[1] 983.63
> sum(g$res^2)
[1] 650.71
> ((983.63-650.71)/4)/(650.706/45)
[1] 5.7558
> 1-pf(5.7558,4,45)
[1] 0.00079026
```

Do you know where all the numbers come from? Check that they match the regression summary above.

3.2.2 Testing just one predictor

Can one particular predictor be dropped from the model? The null hypothesis would be $H_0 : \beta_i = 0$. Set it up like this

- RSS_Ω is the RSS for the model with all the predictors of interest (p parameters).
- RSS_ω is the RSS for the model with all the above predictors except predictor i.

The F-statistic may be computed using the formula from above. 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.

However, squaring the t-statistic here, i.e. t_i^2 gives you the F-statistic, so the two approaches are identical.

For example, to test the null hypothesis that $\beta_1 = 0$ i.e. that `p15` is not significant in the full model, we can simply observe that the p-value is 0.0026 from the table and conclude that the null should be rejected.

Let's do the same test using the general F-testing approach: We'll need the RSS and df for the full model — these are 650.71 and 45 respectively.

and then fit the model that represents the null:

```
> g2 <- lm(sr ~ pop75 + dpi + ddpi, data=savings)
```

and compute the RSS and the F-statistic:

```
> sum(g2$res^2)
[1] 797.72
> (797.72-650.71)/(650.71/45)
[1] 10.167
```

The p-value is then

```
> 1-pf(10.167,1,45)
[1] 0.0026026
```

We can relate this to the t-based test and p-value by

```
> sqrt(10.167)
[1] 3.1886
> 2*(1-pt(3.1886,45))
[1] 0.0026024
```

A somewhat more convenient way to compare two nested models is

```
> anova(g2,g)
Analysis of Variance Table

Model 1: sr ~ pop75 + dpi + ddpi
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     46        798
2     45        651  1    147    10.2 0.0026
```

Understand that this test of `pop15` is relative to the other predictors in the model, namely `pop75, dpi` and `ddpi`. If these other predictors were changed, the result of the test may be different. This means that it is not possible to look at the effect of `pop15` in isolation. Simply stating the null hypothesis as $H_0 : \beta_{pop15} = 0$ is insufficient — information about what other predictors are included in the null is necessary. The result of the test may be different if the predictors change.

3.2.3 Testing a pair of predictors

Suppose we wish to test the significance of variables X_j and X_k. We might construct a table as shown just above and find that both variables have p-values greater than 0.05 thus indicating that individually neither is significant. Does this mean that both X_j and X_k can be eliminated from the model? *Not necessarily*

Except in special circumstances, dropping one variable from a regression model causes the estimates of the other parameters to change so that we might find that after dropping X_j, that a test of the significance of X_k shows that it should now be included in the model.

If you really want to check the joint significance of X_j and X_k, you should fit a model with and then without them and use the general F-test discussed above. Remember that even the result of this test may depend on what other predictors are in the model.

Can you see how to test the hypothesis that both `pop75` and `ddpi` may be excluded from the model?

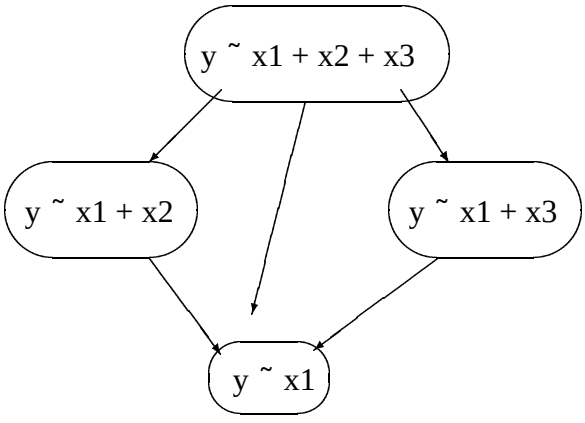

Figure 3.2: Testing two predictors

The testing choices are depicted in Figure 3.2. Here we are considering two predictors, `x2` and `x3` in the presence of `x1`. Five possible tests may be considered here and the results may not always be apparently consistent. The results of each test need to be considered individually in the context of the particular example.

3.2.4 Testing a subspace

Consider this example. Suppose that y is the miles-per-gallon for a make of car and X_j is the weight of the engine and X_k is the weight of the rest of the car. There would also be some other predictors. We might wonder whether we need two weight variables — perhaps they can be replaced by the total weight, $X_j + X_k$. So if the original model was

$$y = \beta_0 + \ldots + \beta_j X_j + \beta_k X_k + \ldots + \varepsilon$$

then the reduced model is

$$y = \beta_0 + \ldots + \beta_l (X_j + X_k) + \ldots + \varepsilon$$

which requires that $\beta_j = \beta_k$ for this reduction to be possible. So the null hypothesis is

$$H_0 : \beta_j = \beta_k$$

This defines a linear subspace to which the general F-testing procedure applies. In our example, we might hypothesize that the effect of young and old people on the savings rate was the same or in other words that

$$H_0 : \beta_{pop15} = \beta_{pop75}$$

In this case the null model would take the form

$$y = \beta_0 + \beta_{pop15}(pop15 + pop75) + \beta_{dpi} dpi + \beta_{ddpi} ddpi + \varepsilon$$

We can then compare this to the full model as follows:

```
> g <- lm(sr ~ .,savings)
> gr <- lm(sr ~ I(pop15+pop75)+dpi+ddpi,savings)
> anova(gr,g)
Analysis of Variance Table

Model 1: sr ~ I(pop15 + pop75) + dpi + ddpi
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     46        674
2     45        651  1     23    1.58   0.21
```

The period in the first model formula is short hand for all the other variables in the data frame. The function `I()` ensures that the argument is evaluated rather than interpreted as part of the model formula. The p-value of 0.21 indicates that the null cannot be rejected here meaning that there is not evidence here that young and old people need to be treated separately in the context of this particular model.

Suppose we want to test whether one of the coefficients can be set to a particular value. For example,

$$H_0 : \beta_{ddpi} = 1$$

Here the null model would take the form:

$$y = \beta_0 + \beta_{pop15} pop15 + \beta_{pop75} pop75 + \beta_{dpi} dpi + ddpi + \varepsilon$$

Notice that there is now no coefficient on the `ddpi` term. Such a fixed term in the regression equation is called an *offset*. We fit this model and compare it to the full:

```
> gr <- lm(sr ~ pop15+pop75+dpi+offset(ddpi),savings)
> anova(gr,g)
Analysis of Variance Table

Model 1: sr ~ pop15 + pop75 + dpi + offset(ddpi)
Model 2: sr ~ pop15 + pop75 + dpi + ddpi
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     46        782
2     45        651  1    131    9.05 0.0043
```

We see that the p-value is small and the null hypothesis here is soundly 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 is

```
> tstat <- (0.409695-1)/0.196197
> tstat
[1] -3.0087
> 2*pt(tstat,45)
[1] 0.0042861
```

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

```
> tstat^2
[1] 9.0525
```

we find we get the F-value. This latter approach is preferred in practice since we don't 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.

Can we test a hypothesis such as

$$H_0 : \beta_j \beta_k = 1$$

using our general theory?

No. This hypothesis is not linear in the parameters so we can't use our general method. We'd need to fit a non-linear model and that lies beyond the scope of this book.

3.3 Concerns about Hypothesis Testing

1. The general theory of hypothesis testing posits a *population* from which a *sample* is drawn — this is our data. We want to say something about the unknown *population* values β using estimated values $\hat{\beta}$ that are obtained from the *sample* data. Furthermore, we require that the data be generated using a *simple random sample* of the population. This sample is finite in size, while the population is infinite in size or at least so large that the sample size is a negligible proportion of the whole. For more complex sampling designs, other procedures should be applied, but of greater concern is the case when the data is not a random sample at all. There are two cases:

   (a) A sample of convenience is where the data is 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 phonebook 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 population defined by the entries in the phonebook. Here we are assuming the selection mechanism is effectively random with respect to the objectives of the study. An assessment of *exchangeability* is required - are the data 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 next case, where the sample is the population.

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 of conclusions from such data is necessarily suspect.

(b) The sample is the complete population in which case one might argue that inference is not required since the population and sample values are one and the same. For both regression datasets we have considered so far, the sample is effectively the population or a large and biased proportion thereof.

In these situations, we can put a different meaning to the hypothesis tests we are making. For the Galapagos 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. We might then ask what the chance would be under this assumption that an F-statistic would be observed as large or larger than one we actually observed. We could compute this exactly by computing the F-statistic for all possible (30!) permutations of the response variable and see what proportion exceed 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.

Let see how we can apply the permutation test to the savings data. I chose a model with just `pop75` and `dpi` so as to get a p-value for the F-statistic that is not too small.

```
> g <- lm(sr ~ pop75+dpi,data=savings)
> summary(g)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.056619   1.290435    5.47  1.7e-06
pop75        1.304965   0.777533    1.68     0.10
dpi         -0.000341   0.001013   -0.34     0.74

Residual standard error: 4.33 on 47 degrees of freedom
Multiple R-Squared: 0.102,      Adjusted R-squared: 0.0642
F-statistic: 2.68 on 2 and 47 degrees of freedom, p-value: 0.0791
```

We can extract the F-statistic as

```
> gs <- summary(g)
```

```
> gs$fstat
  value   numdf   dendf
 2.6796  2.0000 47.0000
```

The function `sample()` generates random permutations. We compute the F-statistic for 1000 randomly selected permutations and see what proportion exceed the the F-statistic for the original data:

```
> fstats <- numeric(1000)
> for(i in 1:1000){
+ ge <- lm(sample(sr) ~ pop75+dpi,data=savings)
+ fstats[i] <- summary(ge)$fstat[1]
+ }
> length(fstats[fstats > 2.6796])/1000
[1] 0.092
```

So our estimated p-value using the permutation test is 0.092 which is close to the normal theory based value of 0.0791. 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.

Thus it is 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 one's conclusion apply only the particular sample.

Tests involving just one predictor also fall within the permutation test framework. We permute that predictor rather than the response

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/population at hand is supposed to have been randomly selected from parallel universes. This argument is definitely more tenuous.

2. A model is usually only an approximation of underlying reality which makes the meaning of the parameters debatable at the very least. We will say more on the interpretation of parameter estimates later but the precision of the statement that $\beta_1 = 0$ exactly is at odds with the acknowledged approximate nature of the model. Furthermore, it is highly unlikely that a predictor that one has taken the trouble to measure and analyze has exactly zero effect on the response. It may be small but it won't be zero.

   This means that in many cases, we know that the point null hypothesis is false without even looking at the data. Furthermore, we know that the more data we have, the greater the power of our tests. Even small differences from zero will be detected with a large sample. Now if we fail to reject the null hypothesis, we might simply conclude that we didn't have enough data to get a significant result. According to this view, the hypothesis test just becomes a test of sample size. For this reason, I prefer confidence intervals.

3. The inference depends on the correctness of the model we use. We can partially check the assumptions about the model but there will always be some element of doubt. Sometimes the data may suggest more than one possible model which may lead to contradictory results.

4. Statistical significance is not equivalent to practical significance. The larger the sample, the smaller your p-values will be so don't confuse p-values with a big predictor effect. With large datasets it will

be very easy to get statistically significant results, but the actual effects may be unimportant. Would we really care if test scores were 0.1% higher in one state than another? Or that some medication reduced pain by 2%? Confidence intervals on the parameter estimates are a better way of assessing the size of an effect. There are useful even when the null hypothesis is not rejected because they tell us how confident we are that the true effect or value is close to the null.

Even so, hypothesis tests do have some value, not least because they impose a check on unreasonable conclusions which the data simply does not support.

3.4 Confidence Intervals for β

Confidence intervals provide an alternative way of expressing the uncertainty in our estimates. Even so, they are closely linked to the tests that we have already constructed. For the confidence intervals and regions that we will consider here, the following relationship holds. For a $100(1-\alpha)\%$ confidence region, any point that lies within the region represents a null hypothesis that would not be rejected at the $100\alpha\%$ level while every point outside represents a null hypothesis that would be rejected. So, in a sense, the confidence region provides a lot more information than a single hypothesis test in that it tells us the outcome of a whole range of hypotheses about the parameter values. Of course, by selecting the particular level of confidence for the region, we can only make tests at that level and we cannot determine the p-value for any given test simply from the region. However, since it is dangerous to read too much into the relative size of p-values (as far as how much evidence they provide against the null), this loss is not particularly important.

The confidence region tells us about plausible values for the parameters in a way that the hypothesis test cannot. This makes it more valuable.

As with testing, we must decide whether to form confidence regions for parameters individually or simultaneously. Simultaneous regions are preferable but for more than two dimensions they are difficult to display and so there is still some value in computing the one-dimensional confidence intervals.

We start with the simultaneous regions. Some results from multivariate analysis show that

$$\frac{(\hat{\beta}-\beta)^T X^T X(\hat{\beta}-\beta)}{\sigma^2} \sim \chi^2_p$$

and

$$\frac{(n-p)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p}$$

and these two quantities are independent. Hence

$$\frac{(\hat{\beta}-\beta)^T X^T X(\hat{\beta}-\beta)}{p\hat{\sigma}^2} \sim \frac{\chi^2_p/p}{\chi^2_{n-p}/(n-p)} \equiv F_{p,n-p}$$

So to form a $100(1-\alpha)$ % confidence region for β, take β such that

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

These regions are ellipsoidally shaped. Because these ellipsoids live in higher dimensions, they cannot easily be visualized.

Alternatively, one could consider each parameter individually which leads to confidence intervals which take the general form of

$$\text{estimate} \pm \text{critical value} \times \text{s.e. of estimate}$$

or specifically in this case:

$$\hat{\beta}_i \pm t_{n-p}^{(\alpha/2)} \hat{\sigma} \sqrt{(X^T X)^{-1}_{ii}}$$

It's better to consider the joint confidence intervals when possible, especially when the $\hat{\beta}$ are heavily correlated.

Consider the full model for the savings data. The `.` in the model formula stands for "every other variable in the data frame" which is a useful abbreviation.

```
> g <- lm(sr ~ ., savings)
> summary(g)
Coefficients:
             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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

We can construct individual 95% confidence intervals for the regression parameters of `pop75`:

```
> qt(0.975,45)
[1] 2.0141
> c(-1.69-2.01*1.08,-1.69+2.01*1.08)
[1] -3.8608  0.4808
```

and similarly for growth

```
> c(0.41-2.01*0.196,0.41+2.01*0.196)
[1] 0.01604 0.80396
```

Notice that this confidence interval is pretty wide in the sense that the upper limit is about 50 times larger than the lower limit. This means that we are not really that confident about what the exact effect of growth on savings really is.

Confidence intervals often have a duality with two-sided hypothesis tests. A 95% confidence interval contains all the null hypotheses that would not be rejected at the 5% level. Thus the interval for `pop75` contains zero which indicates that the null hypothesis $H_0 : \beta_{pop75} = 0$ would not be rejected at the 5% level. We can see from the output above that the p-value is 12.5% — greater than 5% — confirming this point. In contrast, we see that the interval for `ddpi` does not contain zero and so the null hypothesis is rejected for its regression parameter.

Now we construct the joint 95% confidence region for these parameters. First we load in a "library" for drawing confidence ellipses which is not part of base R:

```
> library(ellipse)
```

and now the plot:

```
> plot(ellipse(g,c(2,3)),type="l",xlim=c(-1,0))
```

add the origin and the point of the estimates:

```
> points(0,0)
> points(g$coef[2],g$coef[3],pch=18)
```

How does the position of the origin relate to a test for removing `pop75` and `pop15`?

Now we mark the one way confidence intervals on the plot for reference:

```
> abline(v=c(-0.461-2.01*0.145,-0.461+2.01*0.145),lty=2)
> abline(h=c(-1.69-2.01*1.08,-1.69+2.01*1.08),lty=2)
```

See the plot in Figure 3.3.

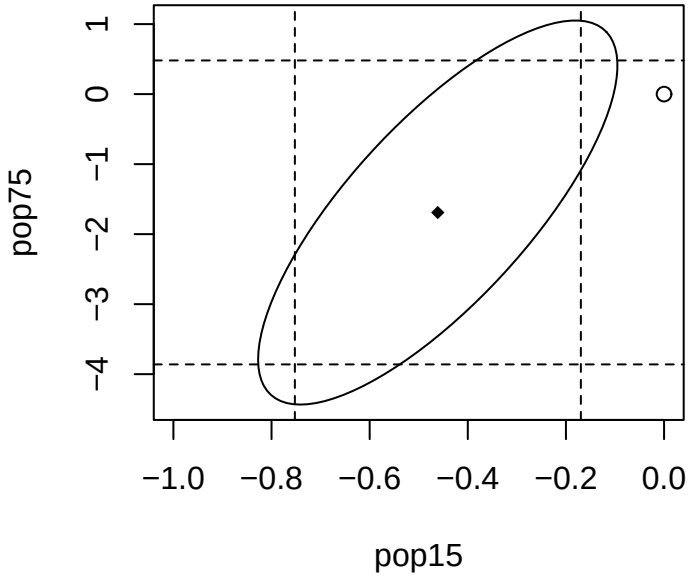

Figure 3.3: Confidence ellipse and regions for β_{pop75} and β_{pop15}

Why are these lines not tangential to the ellipse? The reason for this is that the confidence intervals are calculated individually. If we wanted a 95% chance that both intervals contain their true values, then the lines would be tangential.

In some circumstances, the origin could lie within both one-way confidence intervals, but lie outside the ellipse. In this case, both one-at-a-time tests would not reject the null whereas the joint test would. The latter test would be preferred. It's also possible for the origin to lie outside the rectangle but inside the ellipse. In this case, the joint test would not reject the null whereas both one-at-a-time tests would reject. Again we prefer the joint test result.

Examine the correlation of the two predictors:

```
> cor(savings$pop15,savings$pop75)
[1] -0.90848
```

But from the plot, we see that coefficients have a positive correlation. The correlation between predictors and the correlation between the coefficients of those predictors are often different in sign. Intuitively, this

can be explained by realizing that two negatively correlated predictors are attempting to the perform the same job. The more work one does, the less the other can do and hence the positive correlation in the coefficients.

3.5 Confidence intervals for predictions

Given a new set of predictors, x_0 what is the predicted response? Easy — just $\hat{y}_0 = x_0^T\hat{\beta}$. However, we need to distinguish between predictions of the future mean response and predictions of future observations. To make the distinction, suppose we have built a regression model that predicts the selling price of homes in a given area that is based on predictors like the number of bedrooms, closeness to a major highway etc. There are two kinds of predictions that can be made for a given x_0.

1. Suppose a new house comes on the market with characteristics x_0. Its selling 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 the house with characteristics x_0" sell 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 common.

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

A future observation is predicted to be $x_0^T\hat{\beta} + \varepsilon$ (where we don't what the future ε will turn out to be). So a $100(1-\alpha)$ % confidence interval for a single future response is

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

If on the other hand, you want a confidence interval for the average of the responses for given x_0 then use

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

We return to the Galapagos data for this example.

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

Suppose we want to predict the number of species (of tortoise) on an island with predictors 0.08,93,6.0,12.0,0.34 (same order as in the dataset). Of course it is difficult to see why in practice we would want to do this because a new island is unlikely to present itself. For a dataset like this interest would center on the structure of the model and relative importance of the predictors, so we should regard this more as a "what if?" exercise.

Do it first directly from the formula:

```
> x0 <- c(1,0.08,93,6.0,12.0,0.34)
> y0 <- sum(x0*g$coef)
> y0
[1] 33.92
```

This is the predicted no. of species which is not a whole number as the response is. We could round up to 34.

Now if we want a 95% confidence interval for the prediction, we must decide whether we are predicting the number of species on one new island or the mean response for all islands with same predictors x0. Possibly, an island might not have been surveyed for the original dataset in which case the former interval would be the one we want. For this dataset, the latter interval would be more valuable for "what if?" type calculations.

First we need the t-critical value:

```
> qt(0.975,24)
[1] 2.0639
```

You may need to recalculate the $(X^TX)^{-1}$ matrix:

```
> x <- cbind(1,gala[,3:7])
> x <- as.matrix(x)
> xtxi <- solve(t(x) %*% x)
```

The width of the bands for mean response CI is

```
> bm <- sqrt(x0 %*% xtxi %*% x0) *2.064 * 60.98
> bm
      [,1]
[1,] 32.89
```

and the interval is

```
> c(y0-bm,y0+bm)
[1]  1.0296 66.8097
```

Now we compute the prediction interval for the single future response.

```
> bm <- sqrt(1+x0 %*% xtxi %*% x0) *2.064 * 60.98
> c(y0-bm,y0+bm)
[1]  -96.17  164.01
```

What physically unreasonable feature do you notice about it? In such instances, impossible values in the confidence interval can be avoided by transforming the response, say taking logs, (explained in a later chapter) or by using a probability model more appropriate to the response. The normal distribution is supported on the whole real line and so negative values are always possible. A better choice for this example might be the Poisson distribution which is supported on the non-negative integers.

There is a more direct method for computing the CI. The function `predict()` requires that its second argument be a data frame with variables named in the same way as the original dataset:

```
> predict(g,data.frame(Area=0.08,Elevation=93,Nearest=6.0,Scruz=12,
  Adjacent=0.34),se=T)
$fit:
 33.92
```

```
$se.fit:
 15.934

$df:
[1] 24

$residual.scale:
[1] 60.975
```

The width of the mean response interval can then be calculated by multiplying the se for the fit by the appropriate t-critical value:

```
> 15.934*2.064
[1] 32.888
```

which matches what we did before. CI's for the single future response could also be derived.

3.6 Orthogonality

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

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

Notice that $\hat{\beta}_1$ will be the same regardless of whether X_2 is in the model or not (and vice versa). Now if we wish to test $H_0 : \beta_1 = 0$, it should be noted that $\mathrm{RSS}_\Omega/df = \hat{\sigma}^2_\Omega$ 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 so large as in non-orthogonal cases.

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 which is the source of much of the interpretational difficulties associated with non-experimental data.

Here's an example of an experiment to determine the effects of column temperature, gas/liquid ratio and packing height in reducing unpleasant odor of chemical product that was being sold for household use.

Read the data in and display.

```
> data(odor)
> 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 temp = (Fahrenheit-80)/40 so the original values of the predictor were 40,80 and 120. I don't know the scale of measurement for odor.

Here's the X-matrix:

```
> x <- as.matrix(cbind(1,odor[,-1]))
```

and $X^T X$:

```
> t(x) %*% x
      1 temp gas pack
   1 15    0   0    0
temp  0    8   0    0
 gas  0    0   8    0
pack  0    0   0    8
```

The matrix is diagonal. What would happen if `temp` was measured in the original Fahrenheit scale? The matrix would still be diagonal but the entry corresponding to `temp` would change.

Now fit a model:

```
> g <- lm(odor ~ temp + gas + pack, data=odor)
> summary(g,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 degrees of freedom,       p-value: 0.199

Correlation of Coefficients:
     (Intercept)     temp gas
temp   -1.52e-17
gas    -1.52e-17 4.38e-17
pack    0.00e+00 0.00e+00   0
```

Check out the correlation of the coefficients - why did that happen?. Notice that the standard errors for the coefficients are equal due to the balanced design. Now drop one of the variables:

```
> g <- lm(odor ~ gas + pack, data=odor)
> summary(g)

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 degrees of freedom,       p-value: 0.141
```

Which things changed - which stayed the same? The coefficients themselves do not change but the residual standard error does change slightly which causes small changes in the standard errors of the coefficients, t-statistics and p-values, but nowhere near enough to change our qualitative conclusions.

That was data from an experiment so it was possible to control the values of the predictors to ensure orthogonality. Now consider the savings data which is observational:

```
> g <- lm(sr ~ pop15 + pop75 + dpi + ddpi, savings)
> summary(g)
Coefficients:
             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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

Drop `pop15` from the model:

```
> g <- update(g, . ~ . - pop15)
> summary(g)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.487494   1.427662    3.84  0.00037
pop75       0.952857   0.763746    1.25  0.21849
dpi         0.000197   0.001003    0.20  0.84499
ddpi        0.473795   0.213727    2.22  0.03162
```

```
Residual standard error: 4.16 on 46 degrees of freedom
Multiple R-Squared: 0.189,      Adjusted R-squared: 0.136
F-statistic: 3.57 on 3 and 46 degrees of freedom,       p-value: 0.0209
```

What changed? By how much? Pay particular attention to `pop75`. The effect has now become positive whereas it was negative. Granted, in neither case is it significant, but it is not uncommon in other datasets for such sign changes to occur and for them to be significant.

3.7 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 — 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.

2. For each individual we record no. of years of education K-12 and no. of years of post-HS education and also the total no. of years of education and put all three variables into the model.

3. $p > n$ — more variables than cases. 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 called *supersaturated*. Oddly enough, such models are considered in large scale screening experiments used in product design and manufacture, but there is no hope of uniquely estimating all the parameters in such a model.

Such problems can be avoided by paying attention. Identifiability is more of an issue in designed experiments. Consider a simple two sample experiment:

	Response
Treatment	$y_1, \ldots, y_n$
Control	$y_{n+1}, \ldots, y_{m+n}$

Suppose we try to model the response by an overall mean μ and group effects α_1 and α_2:

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

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

Now although X has 3 columns, it has only rank 2 — $(\mu, \alpha_1, \alpha_2)$ 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 non-identifiability 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.

Identifiability means that

1. You have insufficient data to estimate the parameters of interest *or*

2. You have more parameters than are necessary to model the data.

Here's an example. Suppose we create a new variable for the savings dataset - the percentage of people between 15 and 75:

```
> pa <- 100-savings$pop15-savings$pop75
```

and add that to the model:

```
> g <- lm(sr ~ pa + pop15 + pop75 + dpi + ddpi, data=savings)
> summary(g)
Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.41e+02   1.03e+02   -1.37    0.177
pa           1.69e+00   1.08e+00    1.56    0.126
pop15        1.23e+00   9.77e-01    1.26    0.215
dpi         -3.37e-04   9.31e-04   -0.36    0.719
ddpi         4.10e-01   1.96e-01    2.09    0.042
```

We get a message about one undefined coefficient because the rank of the design matrix X is 5 but should be 6.

Let's take a look at the X-matrix:

```
> x <- as.matrix(cbind(1,pa,savings[,-1]))
> dimnames(x) <- list(row.names(savings),c("int","pa","p15","p75",
  "dpi","ddpi"))
```

If we didn't know which linear combination was causing the trouble, how would we find out? An eigen decomposition of X^TX can help:

```
> e <- eigen(t(x) %*% x)
> signif(e$values,3)
[1] 1.10e+08 1.10e+05 3.19e+03 3.74e+02 1.37e+01 1.09e-14
> signif(e$vectors,3)
int  0.000506  0.0141 -0.00125  0.000603  0.00989  1.00e+00
pa   0.034300  0.7940  0.59700  0.098100 -0.05630 -1.00e-02
p15  0.014700  0.6040 -0.79500 -0.031000  0.04800 -1.00e-02
p75  0.001610  0.0164  0.07310 -0.006840  0.99700 -1.00e-02
dpi  0.999000 -0.0363 -0.00906 -0.001170 -0.00036 -1.07e-17
ddpi 0.001740  0.0594  0.08310 -0.995000 -0.01390 -4.97e-16
```

Only the last eigenvalue is zero, indicating one linear combination is the problem. We can determine which linear combination from the last eigenvalue (last column of the matrix. From this we see that `100-pa-p15-p75=0` is the offending combination.

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 `pa` by adding a random variate from $U[-0.005, 0.005]$ where U denotes the uniform distribution:

```
> pae <- pa +0.001*(runif(50)-0.5)
```

and now refit the model:

```
> ge <- lm(sr ~ pae+pop15+pop75+dpi+ddpi,savings)
> summary(ge)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.57e+05   1.81e+05    0.87    0.391
pae         -1.57e+03   1.81e+03   -0.87    0.391
pop15       -1.57e+03   1.81e+03   -0.87    0.391
pop75       -1.57e+03   1.81e+03   -0.87    0.390
dpi         -3.34e-04   9.34e-04   -0.36    0.722
ddpi         4.11e-01   1.97e-01    2.09    0.042
```

Notice the now all parameters can be estimated but the standard errors are very large because we cannot estimate them in a stable way. We deliberately caused this problem so we know the cause but in general we need to be able to identify such situations. We do this in Chapter 9.

3.8 Summary

We have described a linear model $y = X\beta + \varepsilon$. The parameters β may be estimated using least squares $\hat{\beta} = (X^T X)^{-1} X^T y$. If we further assume that $\varepsilon \sim N(0, \sigma^2 I)$ then we can test any linear hypothesis about β, construct confidence regions for β, make predictions with confidence intervals.

3.9 What can go wrong?

Many things, unfortunately — we try to categorize them below:

3.9.1 Source and quality of the data

How the data was collected directly effects what conclusions we can draw.

1. We may have a biased sample, such as a sample of convenience, from the population of interest. This makes it very difficult to extrapolate from what we see in the sample to general statements about the population. As we have seen, in some cases the sample is the population, in which case any generalization of the conclusions is problematic.

2. Important predictors may not have been observed. This means that our predictions may be poor or we may misinterpret the relationship between the predictors and the response.

3. Observational data make causal conclusions problematic — lack of orthogonality makes disentangling effects difficult. Missing predictors add to this problem.

4. The range and qualitative nature of the data may limit effective predictions. It is unsafe to extrapolate too much. Carcinogen trials may apply large doses to mice. What do the results say about small doses applied to humans? Much of the evidence for harm from substances such as asbestos and radon comes from people exposed to much larger amounts than that encountered in a normal life. It's clear that workers in older asbestos manufacturing plants and uranium miners suffered from their respective exposures to these substances, but what does that say about the danger to you or I?

3.9.2 Error component

We hope that $\varepsilon \sim N(0, \sigma^2 I)$ but

1. Errors may be heterogeneous (unequal variance).

2. Errors may be correlated.

3. Errors may not be normally distributed.

The last defect is less serious than the first two because even if the errors are not normal, the $\hat{\beta}$'s will tend to normality due to the power of the central limit theorem. With larger datasets, normality of the data is not much of a problem.

3.9.3 Structural Component

The structural part of linear model, $Ey = X\beta$ may be incorrect. The model we use may come from 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 was modeled in a particular way. It's natural to see if the same model will work the current data. Models like these usually arise in the social sciences.

3. No prior idea - the model comes from an exploration of the data itself.

Confidence in the conclusions from a model declines as we progress through these. 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 reality which is very complex.

Most statistical theory rests on the assumption that the model is correct. In practice, the best one can hope for is that the model is a fair representation of reality. A model can be no more than a good portrait.

> All models are wrong but some are useful. *George Box*

is only a slight exaggeration. Einstein said

> So far as theories of mathematics are about reality; they are not certain; so far as they are certain, they are not about reality.

3.10 Interpreting Parameter Estimates

Suppose we fit a model to obtain the regression equation:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p$$

What does $\hat{\beta}_1$ mean? In some case, a β might represent a real physical constant, but often the statistical model is just a convenience for representing a complex reality and so the real meaning of a particular β is not obvious.

Let's start with a naive interpretation: "A unit change in x_1 will produce a change of $\hat{\beta}_1$ in the response".

For a properly designed experiment, this interpretation is reasonable provided one pays attention to concerns such as extrapolation and appropriateness of the model selected. The effects of other variables that are included in the experiment can separated out if an orthogonal design is used. For variables not included in the experiment by choice, we may eliminate their effect by holding them constant. If variables that impact the response are not included because they are not known, we use randomization to control their effect. The treatments (predictor values) are assigned to the experimental units or subjects at random. This ensures that these unknown variables will not be correlated in expectation with the predictors we do examine and allows us to come to causal conclusions. These unknown predictors do not, on the average, affect the parameter estimates of interest, but they do contribute to the residual standard error so it's sometimes better to incorporate them in the experimental design if the become known, as this allows for more precise inference.

In a few tightly controlled experiments, it is possible to claim that measurement error is the only kind of error but usually some of the "error" actually comes from the effects of unmeasured variables. We can decompose the usual model as follows:

$$\begin{aligned} y &= X\beta + \varepsilon \\ &= X\beta + Z\gamma + \delta \end{aligned}$$

where Z are unincluded predictors and δ is measurement error in the response. We can assume that $E\varepsilon = 0$ without any loss of generality, because if $E\varepsilon = c$, we could simply redefine β_0 as $\beta_0 + c$ and the error would again have expectation zero. This is another reason why it is generally unwise to remove the intercept term from the model since it acts as a sink for the mean effect of unincluded variables. So we see that ε incorporates both measurement error and the effect of other variables. In a designed experiment, provided the assignment of the experimental units is random, we have $cor(X,Z) = 0$ so that the estimate of β is unaffected in expectation by the presence of Z.

For observational data, no randomization can be used in assigning treatments to the units and orthogonality won't just happen. There are serious objections to any causal conclusions. An inference of causality is often desired but this is usually too much to expect from observational data. An unmeasured and possible unsuspected "lurking" variable Z may be the real cause of an observed relationship between y and X. See Figure 3.4. For example, we will observe a positive correlation among the shoe sizes and reading abilities of elementary school students but this relationship is driven by a lurking variable — the age of the child.

So in observational studies, because we have no control over the assignment of units, we have $cor(X,Z) \neq 0$ and the observed or worse, unobserved, presence of Z causes us great difficulty. In Figure 3.5, we see the effect of possible confounding variables demonstrated.

In observational studies, it is important to adjust for the effects of possible confounding variables such as the Z shown in Figure 3.5. If such variables can be identified, then at least their effect can be interpreted. Unfortunately, one can never be sure that the all relevant Z have been identified.

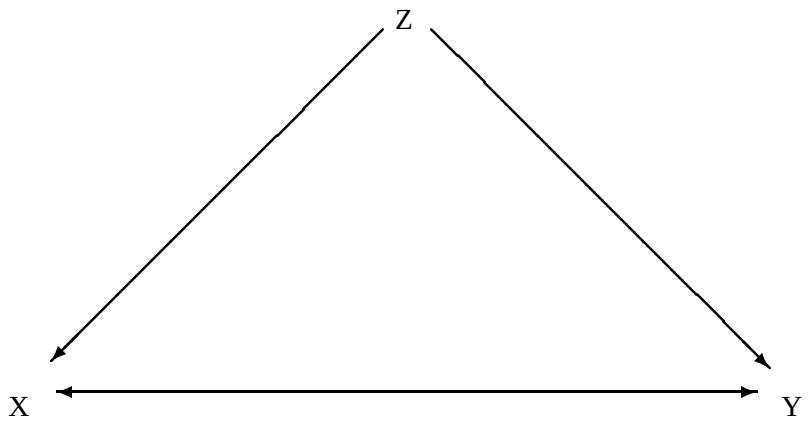

Figure 3.4: Is the relationship between x and y, really caused by z?

What if all relevant variables have been measured? In other words, suppose there are no unidentified *lurking* variables. Even then the naive interpretation does not work. Consider

$$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

but suppose we change $x_2 \to x_1 + x_2$ then

$$y = \hat{\beta}_0 + (\hat{\beta}_1 - \hat{\beta}_2) x_1 + \hat{\beta}_2 (x_1 + x_2)$$

The coefficient for x_1 has changed. Interpretation cannot be done separately for each variable. This is a practical problem because it is not unusual for the predictor of interest, x_1 in this example, to be mixed up in some way with other variables like x_2.

Let's try a new interpretation:

"$\hat{\beta}_1$ is the effect of x_1 when all the other (specified) predictors are held constant".

This too has problems. Often in practice, individual variables cannot be changed without changing others too. For example, in economics we can't expect to change tax rates without other things changing too. Furthermore, this interpretation requires the specification of the other variables - changing which other variables are included will change the interpretation. Unfortunately, there is no simple solution.

Just to amplify this consider the effect of `pop75` on the savings rate in the savings dataset. I'll fit four different models, all including `pop75` but varying the inclusion of other variables.

```
> g <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data=savings)
> summary(g)
Coefficients:
             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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

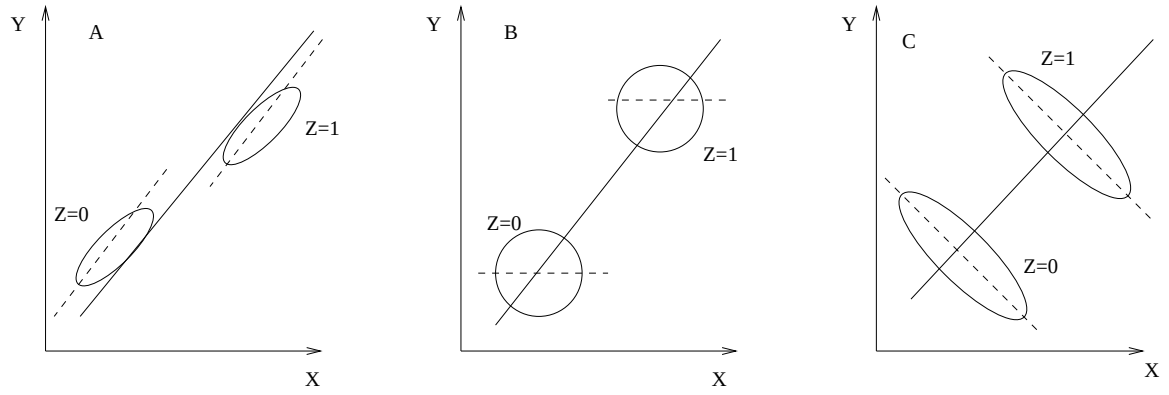

Figure 3.5: Possible confounding effects illustrated. Imagine the data is observed within the ellipses. If the effect of Z is ignored, a strong positive correlation between X and Y is observed in all three cases. In panel A, we see that when we allow for the effect of Z by observing the relationship between X and Y separately within each level of Z, that the relationship remains a positive correlation. In panel B, after allowing for Z, there is no correlation between X and Y, while in panel C, after allowing for Z, the relationship becomes a negative correlation.

It is perhaps surprising that `pop75` is not significant in this model. However, `pop75` is negatively correlated with `pop15` since countries with proportionately more younger people are likely to relatively fewer older ones and vice versa. These two variables are both measuring the nature of the age distribution in a country. When two variables that represent roughly the same thing are included in a regression equation, it is not unusual for one (or even both) of them to appear insignificant even though prior knowledge about the effects of these variables might lead one to expect them to be important.

```
> g2 <- lm(sr ~ pop75 + dpi + ddpi, data=savings)
> summary(g2)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.487494   1.427662    3.84  0.00037
pop75       0.952857   0.763746    1.25  0.21849
dpi         0.000197   0.001003    0.20  0.84499
ddpi        0.473795   0.213727    2.22  0.03162

Residual standard error: 4.16 on 46 degrees of freedom
Multiple R-Squared: 0.189,      Adjusted R-squared: 0.136
F-statistic: 3.57 on 3 and 46 degrees of freedom,       p-value: 0.0209
```

We note that the income variable `dpi` and `pop75` are both not significant in this model and yet one might expect both of them to have something to do with savings rates. Higher values of these variables are both associated with wealthier countries. Let's see what happens when we drop `dpi` from the model:

```
> g3 <- lm(sr ~ pop75 + ddpi, data=savings)
> summary(g3)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.470      1.410    3.88  0.00033
```

```
pop75          1.073      0.456    2.35  0.02299
ddpi           0.464      0.205    2.26  0.02856

Residual standard error: 4.12 on 47 degrees of freedom
Multiple R-Squared: 0.188,      Adjusted R-squared: 0.154
F-statistic: 5.45 on 2 and 47 degrees of freedom,       p-value: 0.00742
```

Now `pop75` is statistically significant with a positive coefficient. We try dropping `ddpi`:

```
> g4 <- lm(sr ~ pop75, data=savings)
> summary(g4)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.152      1.248    5.73  6.4e-07
pop75          1.099      0.475    2.31    0.025

Residual standard error: 4.29 on 48 degrees of freedom
Multiple R-Squared:  0.1,       Adjusted R-squared: 0.0814
F-statistic: 5.34 on 1 and 48 degrees of freedom,       p-value: 0.0251
```

The coefficient and p-value do not change much here due to the low correlation between `pop75` and `ddpi`.

Compare the coefficients and p-values for `pop75` throughout. Notice how the sign and significance change in Table3.2.

No. of Preds	Sign	Significant?
4	-	no
3	+	no
2	+	yes
1	+	yes

Table 3.2: Sign and Significance of $\hat{\beta}_{pop75}$

We see that the significance and the direction of the effect of `pop75` change according to what other variables are also included in the model. We see that no simple conclusion about the effect of `pop75` is possible. We must find interpretations for a variety of models. We certainly won't be able to make any causal conclusions.

In observational studies, there are steps one can take to make a stronger case for causality:

1. Try to include all relevant variables
2. Use non-statistical knowledge of the physical nature of the relationship.
3. Try a variety of models - see if a similar effect is observed. Is $\hat{\beta}_1$ similar, no matter what the model?
4. Multiple studies under different conditions can help confirm a relationship. The connection between smoking and lung cancer was suspected since the early 50's but other explanations for the effect were proposed. It was many years before other plausible explanations were eliminated.

   The news media often jump on the results of a single study but one should be suspicious of these one off results. Publication bias is a problem. Many scientific journal will not publish the results of a

study whose conclusions do not reject the null hypothesis. If different researchers keep studying the same relationship, sooner or later one of them will come up with a significant effect even if one really doesn't exist. It's not easy to find out about all the studies with negative results so it is easy to make the wrong conclusions.

Another source of bias is that researchers have a vested interest in obtaining a positive result. There is often more than one way to analyze the data and the researchers may be tempted to pick the one that gives them the results they want. This is not overtly dishonest but it does lead to a bias towards positive results.

It's difficult to assess the evidence in these situations and one can never be certain. The history of the study of the link between smoking and lung cancer shows that it takes a great deal of effort to progress beyond the observation of an association to strong evidence of causation. One can never be 100% sure.

An alternative approach is recognize that the parameters and their estimates are fictional quantities in most regression situations. The "true" values may never be known (if they even exist in the first place). Instead concentrate on predicting future values - these may actually be observed and success can then be measured in terms of how good the predictions were.

Consider a prediction made using each of the four models above:

```
> x0 <- data.frame(pop15=32,pop75=3,dpi=700,ddpi=3)
> predict(g,x0)
[1] 9.7267
> predict(g2,x0)
[1] 9.9055
> predict(g3,x0)
[1] 10.078
> predict(g4,x0)
[1] 10.448
```

Prediction is more stable than parameter estimation. This enables a rather cautious interpretation of $\hat{\beta}_1$. Suppose the predicted value of y is $\hat{y}$ for given x_1 and for other given predictor values. Now suppose we observe $x_1 + 1$ and the same other given predictor values then the predicted response is increased by $\hat{\beta}_1$. Notice that I have been careful to not to say that we have taken a specific individual and increased their x_1 by 1, rather we have observed a new individual with predictor $x_1 + 1$. To put it another way, people with yellow fingers tend to be smokers but making someone's fingers yellow won't make them more likely to smoke.

Prediction is conceptually simpler since interpretation is not an issue but you do need to worry about extrapolation.

1. Quantitative extrapolation: Is the new x_0 within the range of validity of the model. Is it close to the range of the original data? If not, the prediction may be unrealistic. Confidence intervals for predictions get wider as we move away from the data. We can compute these bands for our last model:

   ```
   > grid <- seq(0,10,0.1)
   > p <- predict(g4,data.frame(pop75=grid),se=T)
   > cv <- qt(0.975,48)
   > matplot(grid,cbind(p$fit,p$fit-cv*p$se,p$fit+cv*p$se),lty=c(1,2,2),
     type="l",xlab="pop75",ylab="Saving")
   > rug(savings$pop75)
   ```

We see that the confidence bands in Figure 3.6 become wider as we move away from the range of the data. However, this widening does not reflect the possibility that the structure of the model itself may change as we move into new territory. The uncertainty in the parametric estimates is allowed for but not uncertainty about the model itself. In Figure 3.7, we see that a model may fit well in the range of the data, but outside of that range, the predictions may be very bad.

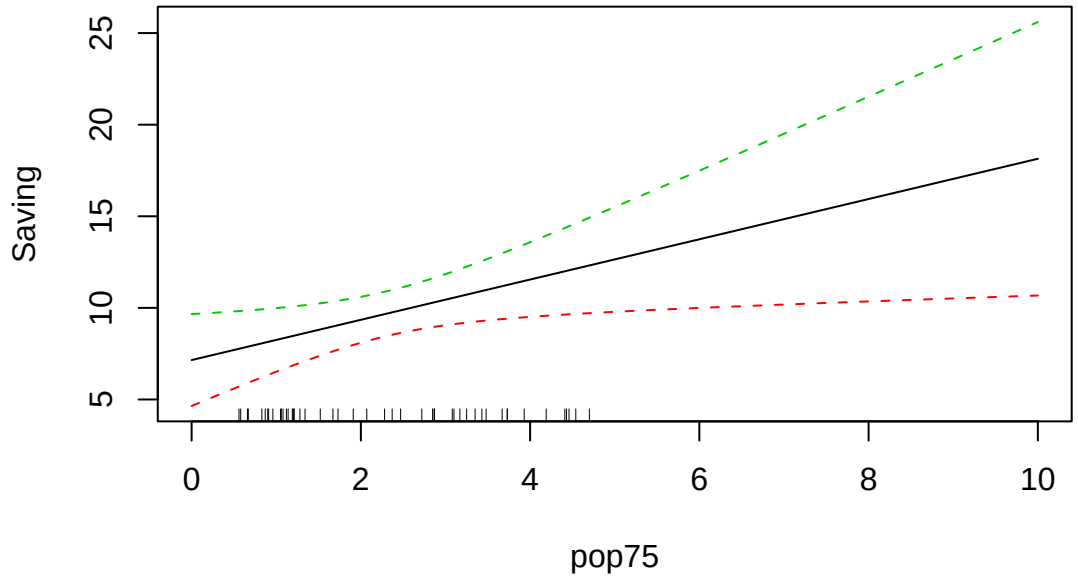

Figure 3.6: Predicted `pop75` over a range of values with 95% pointwise confidence bands for the mean response shown as dotted lines. A "rug" shows the location of the observed values of `pop75`

2. Qualitative extrapolation: Is the new x_0 drawn from the same population from which the original sample was drawn. If the model was built in the past and is to be used for future predictions, we must make a difficult judgment as to whether conditions have remained constant enough for this to work.

Let's end with a quote from the 4th century. Prediction is a tricky business — perhaps the only thing worse than a prediction is no prediction at all.

> The good Christian should beware of mathematicians and all those who make empty prophecies. The danger already exists that mathematicians have made a covenant with the devil to darken the spirit and confine man in the bonds of Hell. - St. Augustine

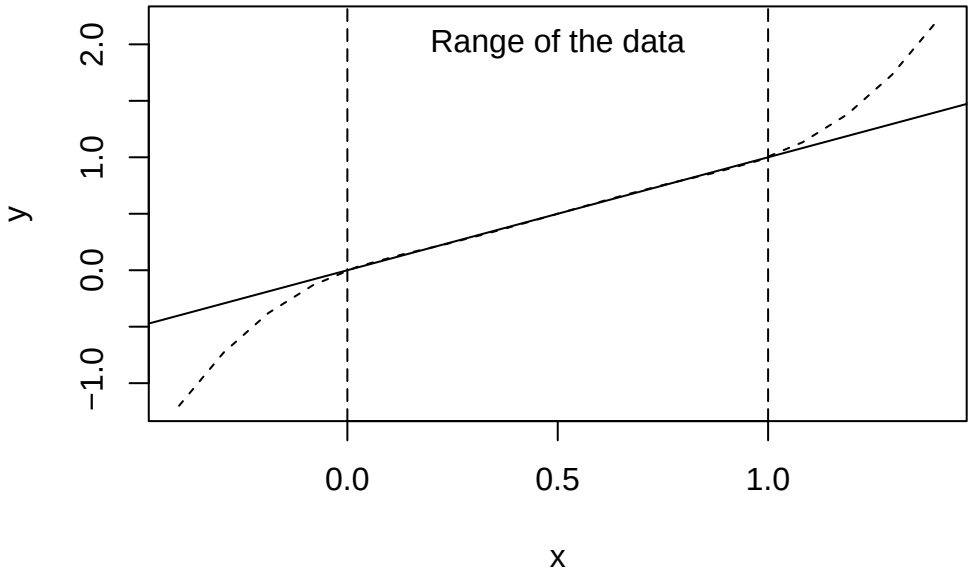

Figure 3.7: Dangers of extrapolation: The model is shown in solid, the real relationship by the dotted line. The data all lie in the predictor range [0,1]

Chapter 4

Errors in 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 the X used to generate Y?

Consider the simple regression (x_i, y_i) for $i = 1, \ldots n$.

$$\begin{aligned} y_i &= \eta_i + \varepsilon_i \\ x_i &= \xi_i + \delta_i \end{aligned}$$

where the errors ε and δ are independent. Suppose the true underlying relationship is

$$\eta_i = \beta_0 + \beta_1 \xi_i$$

but we only see (x_i, y_i). Putting it together, we get

$$y_i = \beta_0 + \beta_1 x_i + (\varepsilon_i - \beta_1 \delta_i)$$

Suppose we use least squares to estimate β_0 and β_1. Let's assume $E\varepsilon_i = E\delta_i = 0$ and that var $\varepsilon_i = \sigma^2$, var $\delta_i = \sigma_\delta^2$. Let

$$\sigma_\xi^2 = \sum (\xi_i - \bar{\xi})^2 / n \qquad \sigma_{\xi\delta} = cov(\xi, \delta)$$

where ξ are the true values of X and not random variables but we could (theoretically since they are not observed) compute statistics using their values. Now $\hat{\beta}_1 = \sum (x_i - \bar{x}) y / \sum (x_i - \bar{x})^2$ and after some calculation we find that

$$E\hat{\beta}_1 \approx \beta_1 \frac{(\sigma_\xi^2 + \sigma_{\xi\delta})}{(\sigma_\xi^2 + \sigma_\delta^2 + 2\sigma_{\xi\delta})}$$

If there is no relation between ξ and δ, this simplifies to

$$E\hat{\beta}_1 \approx \beta_1 \frac{\sigma_\xi^2}{(\sigma_\xi^2 + \sigma_\delta^2)} = \beta_1 \frac{1}{1 + \sigma_\delta^2 / \sigma_\xi^2}$$

So in general $\hat{\beta}_1$ will be biased (regardless of the sample size and typically towards zero). If σ_δ^2 is small relative to σ_ξ^2 then the problem can be ignored. In other words, if the variability in the errors of observation of X are small relative to the range of X then we need not be concerned. If not, it's a serious problem and other methods such as fitting using orthogonal rather than vertical distance in the least squares fit should be considered.

For prediction, measurement error in the x's is not such a problem since the same error will apply to the new x_0 and the model used will be the right one.

For multiple predictors, the usual effect of measurement errors is to bias the $\hat{\beta}$ in the direction of zero.

One should not confuse the 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 conditional on a fixed value for X. We make the assumption that the Y is generated conditional on the fixed value of X. Contrast this with the errors in predictors case where the X we see is not the X that was used to generate the Y.

For real data, the true values of the parameters are usually never known, so it's hard to know how well the estimation is working. Here we generate some artificial data from a known model so we know the true values of the parameters and we can tell how well we do: `runif()` generates uniform random numbers and `rnorm()` generates standard normal random numbers.

Because you will get different random numbers, your results will not exactly match mine if you try to duplicate this.

```
> x <- 10*runif(50)
> y <- x+rnorm(50)
> gx <- lm(y ~ x)
> summary(gx)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.1236     0.2765   -0.45     0.66
x             0.9748     0.0485   20.09   <2e-16

Residual standard error: 1.02 on 48 degrees of freedom
Multiple R-Squared: 0.894,      Adjusted R-squared: 0.891
F-statistic:  403 on 1 and 48 degrees of freedom,       p-value:    0
```

True values of the regression coeffs are 0 and 1 respectively. What happens when we add some noise to the predictor?

```
> z <- x + rnorm(50)
> gz <- lm(y ~ z)
> summary(gz)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.3884     0.3248     1.2     0.24
z             0.8777     0.0562    15.6   <2e-16

Residual standard error: 1.27 on 48 degrees of freedom
Multiple R-Squared: 0.835,      Adjusted R-squared: 0.832
F-statistic:  244 on 1 and 48 degrees of freedom,       p-value:    0
```

Compare the results - notice how the slope has decreased. Now add even more noise:

```
> z2 <- x+5*rnorm(50)
> gz2 <- lm(y ~ z2)
> summary(gz2)
Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.736      0.574    4.77  1.8e-05
z2             0.435      0.101    4.32  7.7e-05

Residual standard error: 2.66 on 48 degrees of freedom
Multiple R-Squared: 0.28,       Adjusted R-squared: 0.265
F-statistic: 18.7 on 1 and 48 degrees of freedom,       p-value: 7.72e-05
```

Compare again — the slope is now very much smaller. We can plot all this information in Figure 4.1.

```
> matplot(cbind(x,z,z2),y,xlab="x",ylab="y")
> abline(gx,lty=1)
> abline(gz,lty=2)
> abline(gz2,lty=5)
```

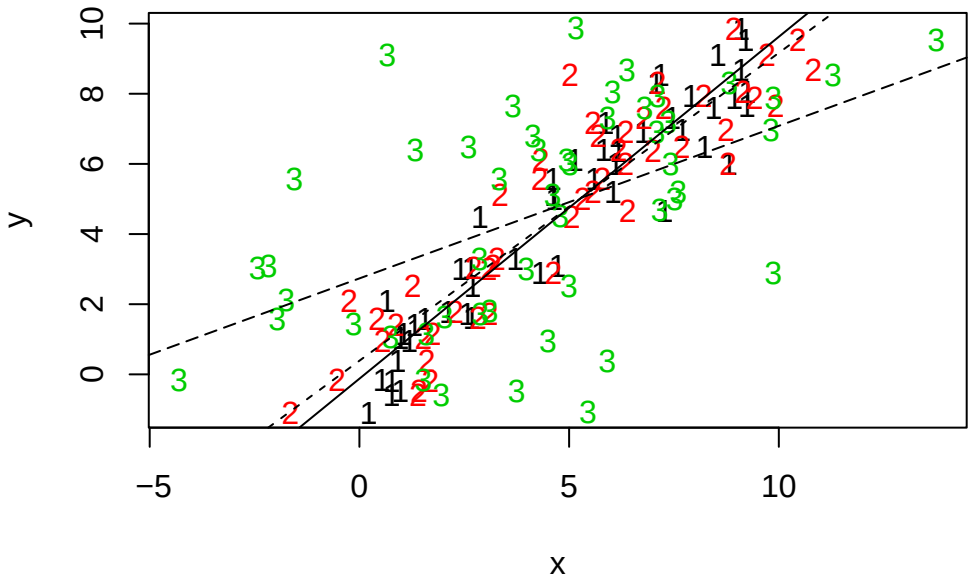

Figure 4.1: Original x shown with “1”, with small error as “2” and with large error as “3”. The regression lines for the no measurement error, small error and large error are shown as solid, dotted and dashed lines respectively.

This was just one realization - to get an idea of average behavior we need to repeat the experiment (I’ll do it 1000 times here). The slopes from the 1000 experiments are saved in the vector bc:

```
> bc <- numeric(1000)
> for(i in 1:1000){
+ y <- x + rnorm(50)
+ z <- x + 5*rnorm(50)
+ g <- lm(y ~ z)
+ bc[i] <- g$coef[2]
+ }
```

Now look at the distribution of `bc`.

```
> summary(bc)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.0106  0.2220  0.2580  0.2580  0.2950  0.4900
```

Given that the variance of a standard uniform random variable is 1/12, $\sigma_\delta^2 = 25$ and $\sigma_\xi^2 = 100/12$, we'd expect the mean to be 0.25. Remember that there is some simulation variation and the expression for the bias is only approximation, so we don't expect them to match exactly.

Chapter 5

Generalized Least Squares

5.1 The general case

Until now we have assumed that var $\varepsilon = \sigma^2 I$ but it can happen that the errors have non-constant variance or are correlated. Suppose instead that 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 don't know the absolute scale.

Generalized least squares minimizes

$$(y - X\beta)^T\Sigma^{-1}(y - X\beta)$$

which is solved by

$$\hat{\beta} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y$$

Since we can write $\Sigma = SS^T$, where S is a triangular matrix using the Choleski Decomposition, we have

$$(y - X\beta)^T S^{-T}S^{-1}(y - X\beta) = (S^{-1}y - S^{-1}X\beta)^T(S^{-1}y - S^{-1}X\beta)$$

So GLS is like regressing $S^{-1}X$ on $S^{-1}y$. Furthermore

$$\begin{aligned} y &= X\beta + \varepsilon \\ S^{-1}y &= S^{-1}X\beta + S^{-1}\varepsilon \\ y' &= X'\beta + \varepsilon' \end{aligned}$$

So we have a new regression equation $y' = X'\beta + \varepsilon'$ where if we examine the variance of the new errors, ε' we find

$$\text{var } \varepsilon' = \text{var } (S^{-1}\varepsilon) = S^{-1}(\text{var } \varepsilon)S^{-T} = S^{-1}\sigma^2 SS^T S^{-T} = \sigma^2 I$$

So the new variables y' and X' are related by a regression equation which has uncorrelated errors with equal variance. Of course, the practical problem is that Σ may not be known.

We find that

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

To illustrate this we'll use a built-in R dataset called Longley's regression data where the response is number of people employed, yearly from 1947 to 1962 and the predictors are GNP implicit price deflator (1954=100), GNP, unemployed, armed forces, noninstitutionalized population 14 years of age and over, and year. The data originally appeared in Longley (1967)

Fit a linear model.

```
> data(longley)
> g <- lm(Employed ~ GNP + Population, data=longley)
> summary(g,cor=T)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  88.9388    13.7850    6.45  2.2e-05
GNP           0.0632     0.0106    5.93  5.0e-05
Population   -0.4097     0.1521   -2.69    0.018

Residual standard error: 0.546 on 13 degrees of freedom
Multiple R-Squared: 0.979,      Adjusted R-squared: 0.976
F-statistic:  304 on 2 and 13 degrees of freedom,       p-value: 1.22e-11

Correlation of Coefficients:
           (Intercept)    GNP
GNP              0.985
Population      -0.999 -0.991
```

Compare the correlation between the variables gnp, pop and their corresponding coefficients. What do you notice?

In data collected over time such as this, successive errors could be correlated. Assuming that the errors take a simple autoregressive form:

$$\varepsilon_{i+1} = \rho\varepsilon_i + \delta_i$$

where $\delta_i \sim N(0,\tau^2)$. We can estimate this correlation ρ by

```
> cor(g$res[-1],g$res[-16])
[1] 0.31041
```

Under this assumption $\Sigma_{ij} = \rho^{|i-j|}$. For simplicity, lets assume we know that $\rho = 0.31041$. We now construct the Σ matrix and compute the GLS estimate of β along with its standard errors.

```
> x <- model.matrix(g)
> Sigma <- diag(16)
> Sigma <- 0.31041^abs(row(Sigma)-col(Sigma))
> Sigi <- solve(Sigma)
> xtxi <- solve(t(x) %*% Sigi %*% x)
> beta <- xtxi %*% t(x) %*% Sigi %*% longley$Empl
> beta
         [,1]
[1,] 94.89889
[2,]  0.06739
[3,] -0.47427
> res <- longley$Empl - x %*% beta
> sig <- sqrt(sum(res^2)/g$df)
> sqrt(diag(xtxi))*sig
[1] 14.157603  0.010867  0.155726
```

Compare with the model output above where the errors are assumed to be uncorrelated.

Another way to get the same result is to regress $S^{-1}y$ on $S^{-1}x$ as we demonstrate here:

```
> sm <- chol(Sigma)
> smi <- solve(t(sm))
> sx <- smi %*% x
> sy <- smi %*% longley$Empl
> lm(sy ~ sx-1)$coef
sx(Intercept)         sxGNP  sxPopulation
     94.89889       0.06739      -0.47427
```

In practice, we would not know that the $\rho = 0.31$ and we'd need to estimate it from the data. Our initial estimate is 0.31 but once we fit our GLS model we'd need to re-estimate it as

```
> cor(res[-1],res[-16])
[1] 0.35642
```

and then recompute the model again with $\rho = 0.35642$. This process would be iterated until convergence.

The `nlme` library contains a GLS fitting function. We can use it to fit this model:

```
> library(nlme)
> g <- gls(Employed ~ GNP + Population,
  correlation=corAR1(form= ~Year), data=longley)
> summary(g)
Correlation Structure: AR(1)
 Formula: ~Year
 Parameter estimate(s):
    Phi
0.64417

Coefficients:
             Value Std.Error t-value p-value
(Intercept) 101.858   14.1989  7.1736  <.0001
GNP           0.072    0.0106  6.7955  <.0001
Population   -0.549    0.1541 -3.5588  0.0035

Residual standard error: 0.68921
Degrees of freedom: 16 total; 13 residual
```

We see that the estimated value of ρ is 0.64. However, if we check the confidence intervals for this:

```
> intervals(g)
Approximate 95% confidence intervals

 Coefficients:
                lower       est.      upper
(Intercept) 71.183204 101.858133 132.533061
GNP          0.049159   0.072071   0.094983
Population  -0.881491  -0.548513  -0.215536
```

```
 Correlation structure:
       lower    est.   upper
Phi -0.44335 0.64417 0.96451

 Residual standard error:
  lower    est.   upper
0.24772 0.68921 1.91748
```

we see that it is not significantly different from zero.

5.2 Weighted Least Squares

Sometimes the errors are uncorrelated, but have unequal variance where the form of the inequality is known. Weighted least squares (WLS) can be used in this situation. When Σ is diagonal, the errors are uncorrelated but do not necessarily have equal variance. We can write $\Sigma = \text{diag}(1/w_1, \ldots, 1/w_n)$, where the w_i are the weights so $S = \text{diag}(\sqrt{1/w_1}, \ldots, \sqrt{1/w_n})$. So we can regress $\sqrt{w_i}x_i$ on $\sqrt{w_i}y_i$ (although the column of ones in the X-matrix needs to be replaced with $\sqrt{w_i}$. Cases with low variability should get a high weight, high variability a low weight. Some examples

1. Errors proportional to a predictor: var $(\varepsilon_i) \propto x_i$ suggests $w_i = x_i^{-1}$
2. Y_i are the averages of n_i observations then var $y_i =$ var $\varepsilon_i = \sigma^2/n_i$ suggests $w_i = n_i$.

Here's an example from an experiment to study the interaction of certain kinds of elementary particles on collision with proton targets. The experiment was designed to test certain theories about the nature of the strong interaction. The cross-section(crossx) variable is believed to be linearly related to the inverse of the energy(`energy` - has already been inverted). At each level of the momentum, a very large number of observations were taken so that it was possible to accurately estimate the standard deviation of the response(sd).

Read in and check the data:

```
> data(strongx)
> strongx
   momentum energy  crossx sd
 1        4   0.345    367 17
 2        6   0.287    311  9
 3        8   0.251    295  9
 4       10   0.225    268  7
 5       12   0.207    253  7
 6       15   0.186    239  6
 7       20   0.161    220  6
 8       30   0.132    213  6
 9       75   0.084    193  5
10      150   0.060    192  5
```

Define the weights and fit the model:

```
> g <- lm(crossx ~ energy, strongx, weights=sd^-2)
> summary(g)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   148.47       8.08    18.4  7.9e-08
energy        530.84      47.55    11.2  3.7e-06

Residual standard error: 1.66 on 8 degrees of freedom
Multiple R-Squared: 0.94,       Adjusted R-squared: 0.932
F-statistic:  125 on 1 and 8 degrees of freedom,        p-value: 3.71e-06
```

Try fitting the regression without weights and see what the difference is.

```
> gu <- lm(crossx ~ energy, strongx)
> summary(gu)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    135.0       10.1    13.4  9.2e-07
energy         619.7       47.7    13.0  1.2e-06

Residual standard error: 12.7 on 8 degrees of freedom
Multiple R-Squared: 0.955,      Adjusted R-squared: 0.949
F-statistic:  169 on 1 and 8 degrees of freedom,        p-value: 1.16e-06
```

The two fits can be compared

```
> plot(crossx ~ energy, data=strongx)
> abline(g)
> abline(gu,lty=2)
```

and are shown in Figure 5.1.

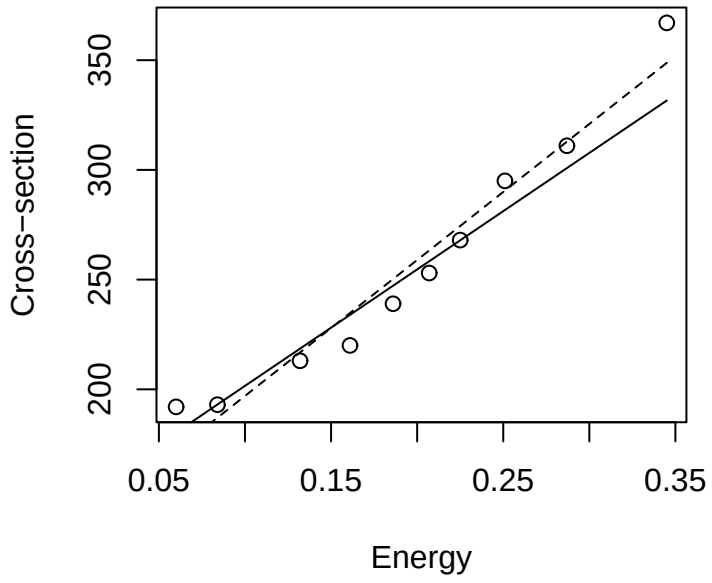

Figure 5.1: Weighted least square fit shown in solid. Unweighted is dashed.

The unweighted fit appears to fit the data better overall but remember that for lower values of energy, the variance in the response is less and so the weighted fit tries to catch these points better than the others.

5.3 Iteratively Reweighted Least Squares

In cases, where the form of the variance of ε is not completely known, we may model Σ using a small number of parameters. For example,

$$\text{var } \varepsilon_i = \gamma_0 + \gamma_1 x_1$$

might seem reasonable in a given situation. The IRWLS fitting Algorithm is

1. Start with $w_i = 1$
2. Use least squares to estimate β.
3. Use the residuals to estimate γ, perhaps by regressing $\hat{\varepsilon}^2$ on x.
4. Recompute the weights and goto 2.

Continue until convergence. There are some concerns about this — how is subsequent inference about β affected? Also how many degrees of freedom do we have? More details may be found in Carroll and Ruppert (1988).

An alternative approach is to model the variance and jointly estimate the regression and weighting parameters using likelihood based method. This can be implemented in `R` using the `gls()` function in the `nlme` library.

Chapter 6

Testing for Lack of Fit

How can we tell if 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 which is not complex enough to fit the data or simply takes the wrong form, then $\hat{\sigma}^2$ will overestimate σ^2. An example can be seen in Figure 6.1. Alternatively, if our model is too complex and overfits the data, then $\hat{\sigma}^2$ will be an underestimate.

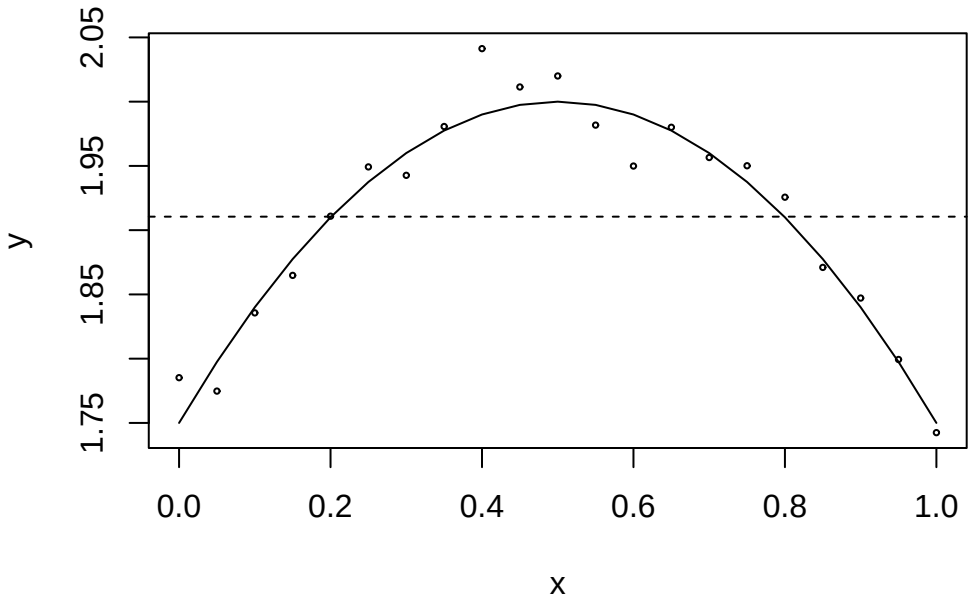

Figure 6.1: True quadratic fit shown with the solid line and incorrect linear fit shown with the dotted line. Estimate of σ^2 will be unbiased for the quadratic model but far too large for the linear model

This suggests a possible testing procedure — we should compare $\hat{\sigma}^2$ to σ^2. There are two cases — one where σ^2 is known and one where it is not.

6.1 σ^2 known

σ^2 known may be known from past experience, knowledge of the measurement error inherent in an instrument or by definition. Recall (from Section 3.4) that

$$\frac{\hat{\sigma}^2}{\sigma^2} \sim \frac{\chi^2_{n-p}}{(n-p)}$$

which leads to the test: Conclude there is a lack of fit if

$$\frac{(n-p)\hat{\sigma}^2}{\sigma^2} > \chi^{2\ (1-\alpha)}_{n-p}$$

If a lack of fit is found, then a new model is needed.

Continuing with the same data as in the weighted least squares example we test to see if a linear model is adequate. In this example, we know the variance almost exactly because each response value is the average of a large number of observations. Because of the way the weights are defined, $w_i = 1/\text{var}\, y_i$, the known variance is implicitly equal to one. There is nothing special about one - we could define $w_i = 99/\text{var}\, y_i$ and the variance would be implicitly 99. However, we would get essentially the same result as the following analysis.

```
> data(strongx)
> g <- lm(crossx ~ energy, weights=sdˆ-2, strongx)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   148.47       8.08    18.4  7.9e-08
energy        530.84      47.55    11.2  3.7e-06

Residual standard error: 1.66 on 8 degrees of freedom
Multiple R-Squared: 0.94,       Adjusted R-squared: 0.932
F-statistic:  125 on 1 and 8 degrees of freedom,        p-value: 3.71e-06
```

Examine the R^2 - do you think the model is a good fit?

Now plot the data and the fitted regression line (shown as a solid line on Figure 6.2.

```
> plot(strongx$energy,strongx$crossx,xlab="Energy",ylab="Crossection")
> abline(g$coef)
```

Compute the test statistic and the p-value:

```
> 1.66ˆ2*8
[1] 22.045
> 1-pchisq(22.045,8)
[1] 0.0048332
```

We conclude that there is a lack of fit. Just because R^2 is large does not mean that you can not do better. Add a quadratic term to the model and test again:

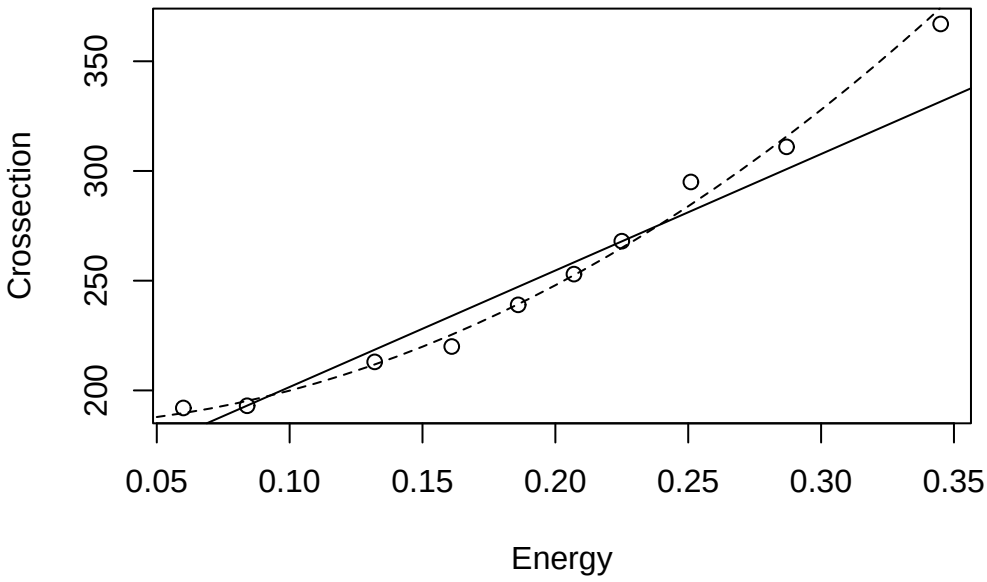

Figure 6.2: Linear and quadratic fits to the physics data

```
> g2 <- lm(crossx ~ energy + I(energy^2), weights=sd^-2, strongx)
> summary(g2)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  183.830      6.459   28.46  1.7e-08
energy         0.971     85.369    0.01  0.99124
I(energy^2) 1597.505    250.587    6.38  0.00038

Residual standard error: 0.679 on 7 degrees of freedom
Multiple R-Squared: 0.991,      Adjusted R-squared: 0.989
F-statistic:  391 on 2 and 7 degrees of freedom,        p-value: 6.55e-08
> 0.679^2*7
[1] 3.2273
> 1-pchisq(3.32273,7)
[1] 0.85363
```

This time we cannot detect a lack of fit. Plot the fit:

```
> x <- seq(0.05,0.35,by=0.01)
> lines(x,g2$coef[1]+g2$coef[2]*x+g2$coef[3]*x^2,lty=2)
```

The curve is shown as a dotted line on the plot (thanks to `lty=2`). This seems clearly more appropriate than the linear model.

6.2 σ^2 unknown

The $\hat{\sigma}^2$that is based in the chosen regression model needs to be compared to some model-free estimate of σ^2. We can do this if we have repeated y for one or more fixed x. These replicates do need to be

truly independent. They cannot just be repeated measurements on the same subject or unit. Such repeated measures would only reveal the within subject variability or the measurement error. We need to know the between subject variability — this reflects the σ^2 described in the model.

The "pure error" estimate of σ^2 is given by SS_{pe}/df_{pe} where

$$SS_{pe} = \sum_{\text{distinct x}} \sum_{given x} (y_i - \bar{y})^2$$

Degrees of freedom $df_{pe} = \sum_{distinct x}(\#replicates - 1)$

If you 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 just the one-way anova model if you are familiar with that. Comparing this model to the regression model amounts to the lack of fit test. This is usually the most convenient way to compute the test but if you like we can then partition the RSS into that due to lack of fit and that due to the pure error as in Table 6.1.

	df	SS	MS	F
Residual	n-p	RSS		
Lack of Fit	$n-p-df_{pe}$	$RSS - SS_{pe}$	$\frac{RSS-SS_{pe}}{n-p-df_{pe}}$	Ratio of MS
Pure Error	df_{pe}	SS_{pe}	SS_{pe}/df_{pe}	

Table 6.1: ANOVA for lack of fit

Compute the F-statistic and compare to $F_{n-p-df_{pe},df_{pe}}$ and reject if the statistic is too large.

Another way of looking at this is a comparison between the model of interest and a saturated model that assigns a parameter to each unique combination of the predictors. Because the model of interest represents a special case of the saturated model where the saturated parameters satisfy the constraints of the model of interest, we can use the standard F-testing methodology.

The data for this example consist of thirteen specimens of 90/10 Cu-Ni alloys with varying iron content in percent. The specimens were submerged in sea water 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 print the data

```
> data(corrosion)
> corrosion
     Fe  loss
1  0.01 127.6
2  0.48 124.0
3  0.71 110.8
4  0.95 103.9
5  1.19 101.5
6  0.01 130.1
7  0.48 122.0
8  1.44  92.3
9  0.71 113.1
10 1.96  83.7
11 0.01 128.0
12 1.44  91.4
13 1.96  86.2
```

We fit a straight line model:

```
> g <- lm(loss ~ Fe, data=corrosion)
> summary(g)
Coefficients:
            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

Residual standard error: 3.06 on 11 degrees of freedom
Multiple R-Squared: 0.97,       Adjusted R-squared: 0.967
F-statistic:  352 on 1 and 11 degrees of freedom,       p-value: 1.06e-09
```

Check the fit graphically — see Figure 6.3.

```
> plot(corrosion$Fe,corrosion$loss,xlab="Iron content",ylab="Weight loss")
> abline(g$coef)
```

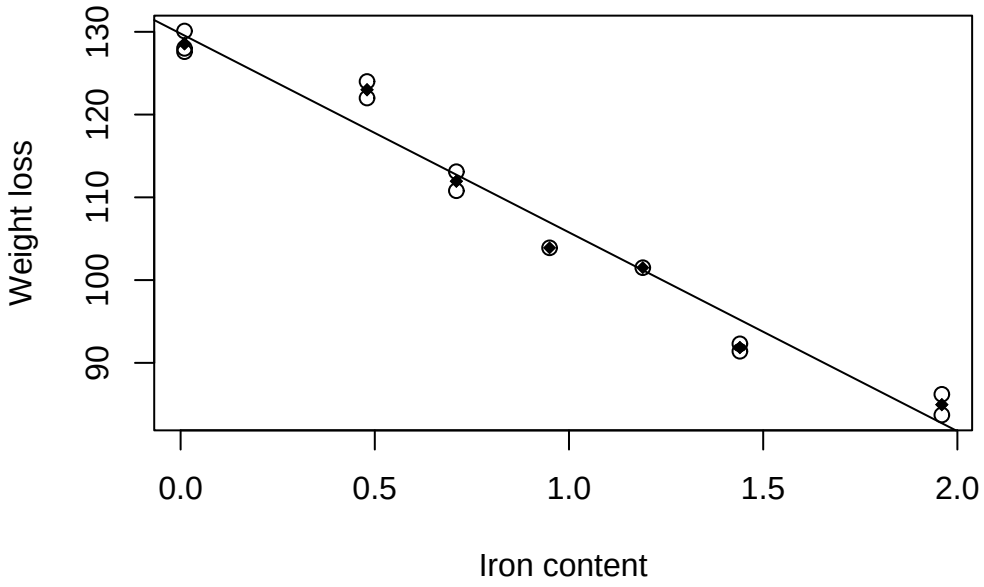

Figure 6.3: Linear fit to the Cu-Ni corrosion data. Group means denoted by black diamonds

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 a later chapter

```
> ga <- lm(loss ~ factor(Fe), data=corrosion)
```

The fitted values are the means in each group - put these on the plot:

```
> points(corrosion$Fe,ga$fit,pch=18)
```

We can now compare the two models in the usual way:

```
> anova(g,ga)
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, I would first find out whether the replicates are genuine — perhaps the low pure error SD can be explained by some correlation in the measurements. Another possible explanation is 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 parameter assigned to each group of x-values, the residual sum of squares will not be zero. For the factor 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's 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 what 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.

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 itself and any predictions made using such a model will tend to have very 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 example above exactly using a sixth order polynomial:

```
> gp <- 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$Fe,ga$fit,pch=18)
> grid <- seq(0,2,len=50)
> lines(grid,predict(gp,data.frame(Fe=grid)))
```

as shown in Figure 6.4. The fit of this model is excellent — for example:

```
> summary(gp)$r.squared
[1] 0.99653
```

but it is clearly riduculous. 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.

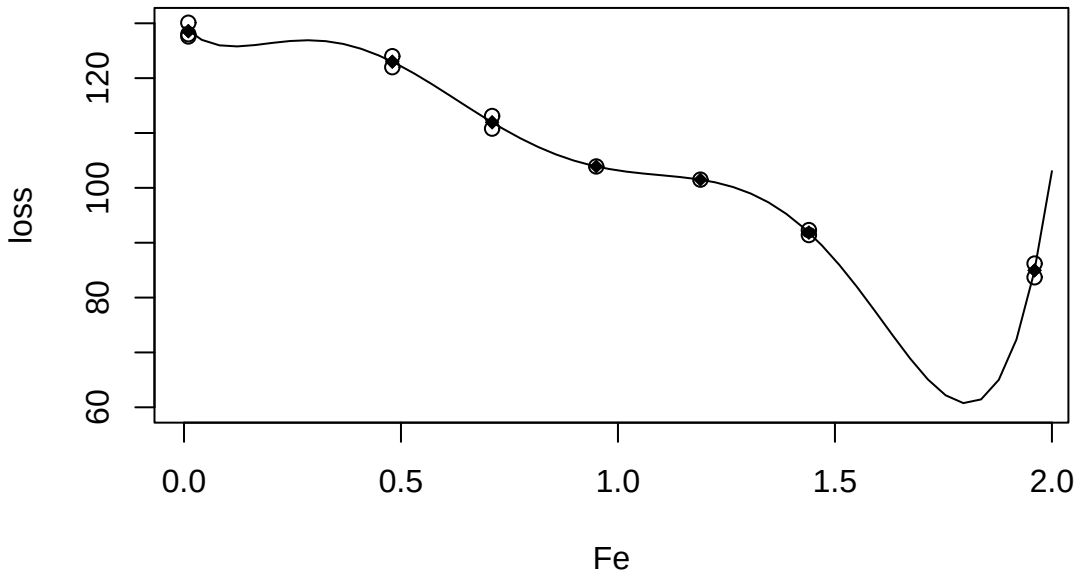

Figure 6.4: Polynomial fit to the corrosion data

Chapter 7

Diagnostics

Regression model building is often an iterative and interactive process. The first model we try may prove to be inadequate. Regression diagnostics are used to detect problems with the model and suggest improvements. This is a hands-on process.

7.1 Residuals and Leverage

We start with some basic diagnostic quantities - the residuals and the leverages. Recall that $\hat{y} = X(X^TX)^{-1}X^Ty = Hy$ where H is the hat-matrix. Now

$$\begin{aligned}\hat{\varepsilon} = y - \hat{y} &= (I-H)y \\ &= (I-H)X\beta + (I-H)\varepsilon \\ &= (I-H)\varepsilon\end{aligned}$$

So var $\hat{\varepsilon}$ = var $(I-H)\varepsilon = (I-H)\sigma^2$ assuming that var $\varepsilon = \sigma^2 I$. We see that although the errors may have equal variance and be uncorrelated the residuals do not.

$h_i = H_{ii}$ are called *leverages* and are useful diagnostics. We see that var $\hat{\varepsilon}_i = \sigma^2(1-h_i)$ so that a large leverage for h_i will make var $\hat{\varepsilon}_i$ small — in other words the fit will be "forced" to be close to y_i. The h_i depends only on X — knowledge of y is required for a full interpretation. Some facts:

$$\sum_i h_i = p \qquad h_i \geq 1/n \quad \forall i$$

An average value for h_i is p/n and a "rule of thumb" is that leverages of more than $2p/n$ should be looked at more closely. Large values of h_i are due to extreme values in X. h_i corresponds to a 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.

Also notice that var $\hat{y}$ = var $(Hy) = H\sigma^2$ so vâr $\hat{y}_i = h_i\hat{\sigma}^2$

We'll use the savings dataset as an example here. First fit the model and make an index plot of the residuals:

```
> data(savings)
> g <- lm(sr ~ pop15 + pop75 + dpi + ddpi, savings)
> plot(g$res,ylab="Residuals",main="Index plot of residuals")
```

The plot is shown in the first panel of Figure 7.1

We can find which countries correspond to the largest and smallest residuals:

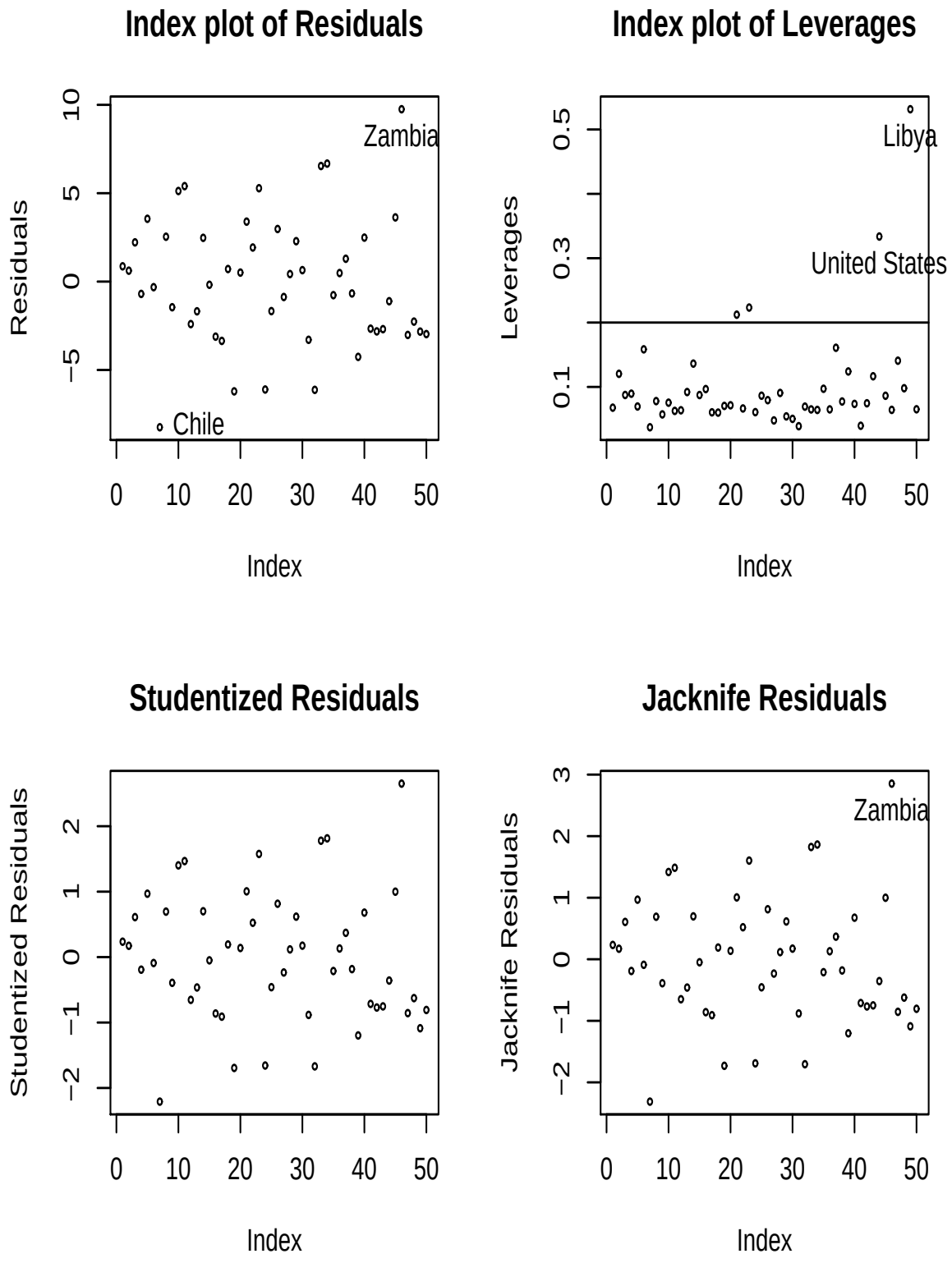

Figure 7.1: Residuals and leverages for the savings data

```
> sort(g$res)[c(1,50)]
  Chile  Zambia
-8.2422  9.7509
```

or by using the `identify()` function. We first make up a character vector of the country names using `row.names()` which gets the row names from the data frame.

```
> countries <- row.names(savings)
> identify(1:50,g$res,countries)
```

Click on the left mouse button next to the points you are interested in to identify them. When you are done, click on the middle (if not available, the right) mouse button. I have identified Chile and Zambia on the plot.

Now look at the leverage: We first extract the X-matrix here using `model.matrix()` and then compute and plot the leverages (also called "hat" values)

```
> x <- model.matrix(g)
> lev <- hat(x)
> plot(lev,ylab="Leverages",main="Index plot of Leverages")
> abline(h=2*5/50)
> sum(lev)
[1] 5
```

Notice that the sum of the leverages is equal to p which is 5 for this data. Which countries have large leverage? We have marked a horizontal line at $2p/n$ to indicate our "rule of thumb". We can see which countries exceed this rather arbitrary cut-off:

```
> names(lev) <- countries
> lev[lev > 0.2]
     Ireland         Japan United States         Libya
     0.21224       0.22331       0.33369       0.53146
```

The command `names()` assigns the country names to the elements of the vector `lev` making it easier to identify them. Alternatively, we can do it interactively like this

```
identify(1:50,lev,countries)
```

I have identified Libya and the United States as the points with the highest leverage.

7.2 Studentized Residuals

As we have seen var $\hat{\varepsilon}_i = \sigma^2(1-h_i)$ this suggests the use of

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

which are called (internally) studentized residuals. If the model assumptions are correct var $r_i = 1$ and $corr(r_i, r_j)$ tends to be small. Studentized residuals are sometimes preferred in residual plots as they have been standardized to have equal variance.

Note that studentization can only correct for the natural non-constant variance in residuals when the errors have constant variance. If there is some underlying heteroscedascity in the errors, studentization cannot correct for it.

We now get the studentized residuals for the savings data:

```
> gs <- summary(g)
> gs$sig
[1] 3.8027
> stud <- g$res/(gs$sig*sqrt(1-lev))
> plot(stud,ylab="Studentized Residuals",main="Studentized Residuals")
```

Notice the range on the axis. Which residuals are large? In this case, there is not much difference between the studentized and raw residuals apart from the scale. Only when there is unusually large leverage will the differences be noticeable.

7.3 An outlier test

An outlier is a point that does not fit the current model. We need to be aware of such exceptions. An outlier test is useful because it enables us to distinguish between truly unusual points and residuals which are large but not exceptional.

Outliers may effect the fit — see Figure 7.2. The two additional points marked points both have high leverage because they are far from the rest of the data. ▲ is not an outlier. • does not have a large residual if it is included in the fit. Only when we compute the fit without that point do we get a large residual.

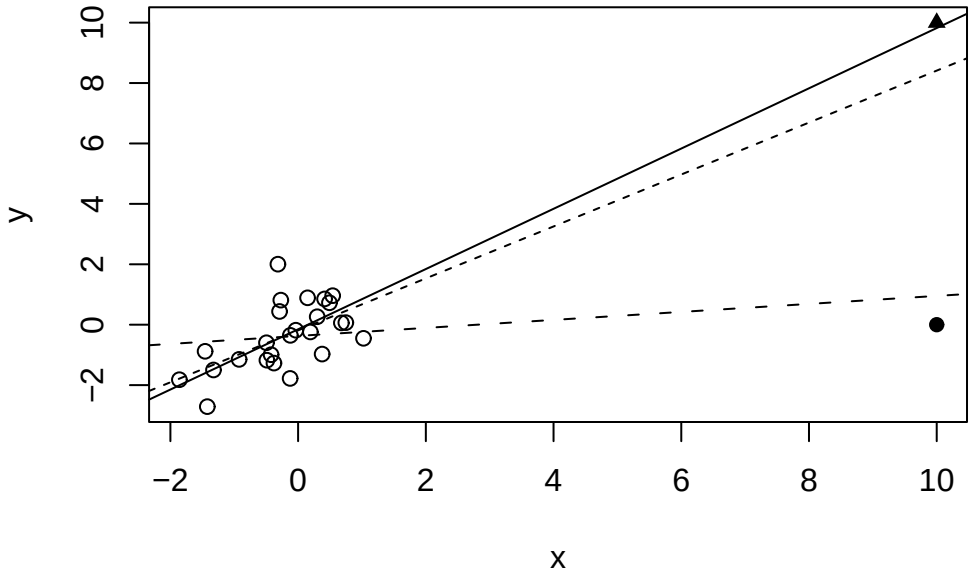

Figure 7.2: Outliers can conceal themselves. The solid line is the fit including the ▲ point but not the • point. The dotted line is the fit without either additional point and the dashed line is the fit with the • point but not the ▲ point.

We exclude point i and recompute the estimates to get $\hat{\beta}_{(i)}$ and $\hat{\sigma}^2_{(i)}$ 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. Just looking at $\hat{\varepsilon}_i$ misses those nasty points which pull the regression line so close to them that they conceal their true status. How large is large?

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

and so

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

Define the jackknife (or externally studentized or crossvalidated) residuals as

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

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

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

which avoids doing n regressions.

Since $t_i \sim t_{n-p-1}$ and we can calculate a p-value to test whether case i is an outlier. However, we are likely to want to test all cases so we must adjust the level of the test accordingly. Even though it might seems that we only test one or two large t_i's, by identifying them as large we are implicitly testing all cases. Suppose we want a level α test. Now P(all tests accept) = 1 - P(At least one rejects) $\geq 1 - \sum_i P$(Test i 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 as well. It's 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.

Now get the jacknife residuals for the savings data:

```
> jack <- rstudent(g)
> plot(jack,ylab="Jacknife Residuals",main="Jacknife Residuals")
> jack[abs(jack)==max(abs(jack))]
Zambia
2.8536
```

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

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

What do you conclude?

Notes

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 large changes occur not infrequently.

4. Individual outliers are usually much less of a problem in larger datasets. A single point won't have the leverage to affect the fit very much. It's still worth identifying outliers if these type of points are worth knowing about in the particular application. For large datasets, we need only worry about clusters of outliers. Such clusters are less likely to occur by chance and more likely to represent actual structure. Finding these cluster 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.

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 in this case may represent fraudulent use.

3. Exclude the point from the analysis but try reincluding it later if the model is changed. The exclusion of one or more points may make the difference between getting a statistical significant result or having some unpublishable research. This can lead to 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.

   It's dangerous to exclude outliers in an automatic manner. NASA launched the Nimbus 7 satellite to record atmospheric information. After several years of operation in 1985, the British Antartic Survey observed a large decrease in atmospheric ozone over the Antartic. 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 Antartic ozone hole was delayed several years. Perhaps, if this had been known earlier, the CFC phaseout would have been agreed earlier and the damage could have been limited.

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.

Read in and plot the data:

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

What do you think relationship is between temperature and light intensity? Now fit a linear regression and add the fitted line to the plot

```
> ga <- lm(light ~ temp, data=star)
> abline(ga)
```

The plot is shown in Figure 7.3 with the regression line in solid type.

Is this what you expected? Are there any outliers in the data? The outlier test does not reveal any.

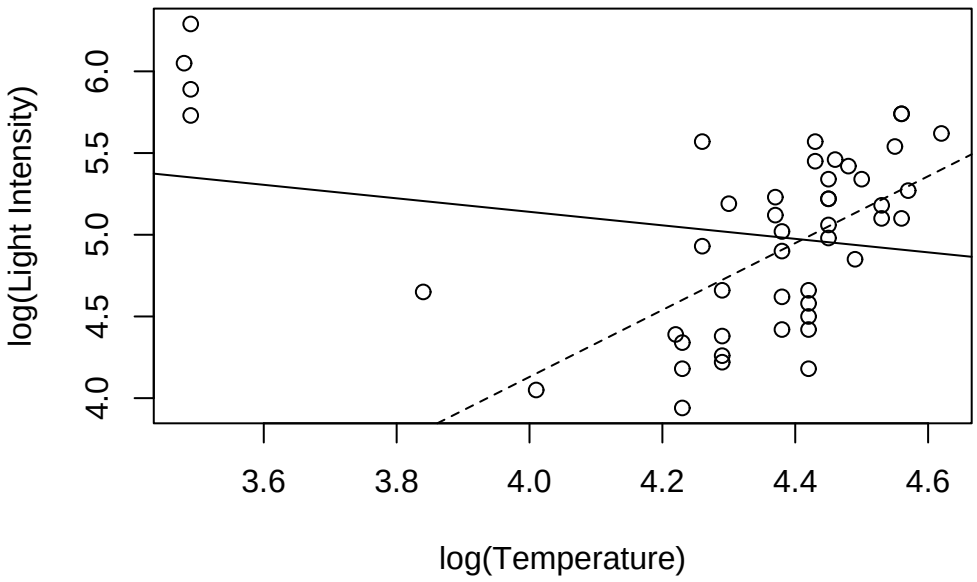

Figure 7.3: Regression line including four leftmost points is solid and excluding these points is dotted

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

We need not bother to actually compute the critical value since these values are clearly not large enough. The four stars on the upper left of the plot are giants. See what happens if these are excluded

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

This illustrates the problem of multiple outliers. We can visualize the problems here, but for higher dimensional data this is much more difficult.

7.4 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 may not have large leverage but it will tend to have at least one of those two properties. In Figure 7.2, the ▲ point is not an influential point but the • point is.

Here are some measures of influence, where the subscripted (i) indicates the fit without case i.

1. Change in the coefficients $\hat{\beta} - \hat{\beta}_{(i)}$
2. Change in the fit $X^T(\hat{\beta} - \hat{\beta}_{(i)}) = \hat{y} - \hat{y}_{(i)}$

These are hard to judge in the sense that the scale varies between datasets. A popular alternative are the Cook Statistics:

$$D_i = \frac{(\hat{\beta} - \hat{\beta}_{(i)})^T (X^T X)(\hat{\beta} - \hat{\beta}_{(i)})}{p\hat{\sigma}^2}$$

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

The first term, r_i^2, is the residual effect and the second is the leverage. The combination of the two leads to influence. An index plot of D_i can be used to identify influential points.

Continuing with our study of the savings data:

```
> cook <- cooks.distance(g)
> plot(cook,ylab="Cooks distances")
> identify(1:50,cook,countries)
```

The Cook statistics may be seen in Figure 7.4. I have identified the largest three values.

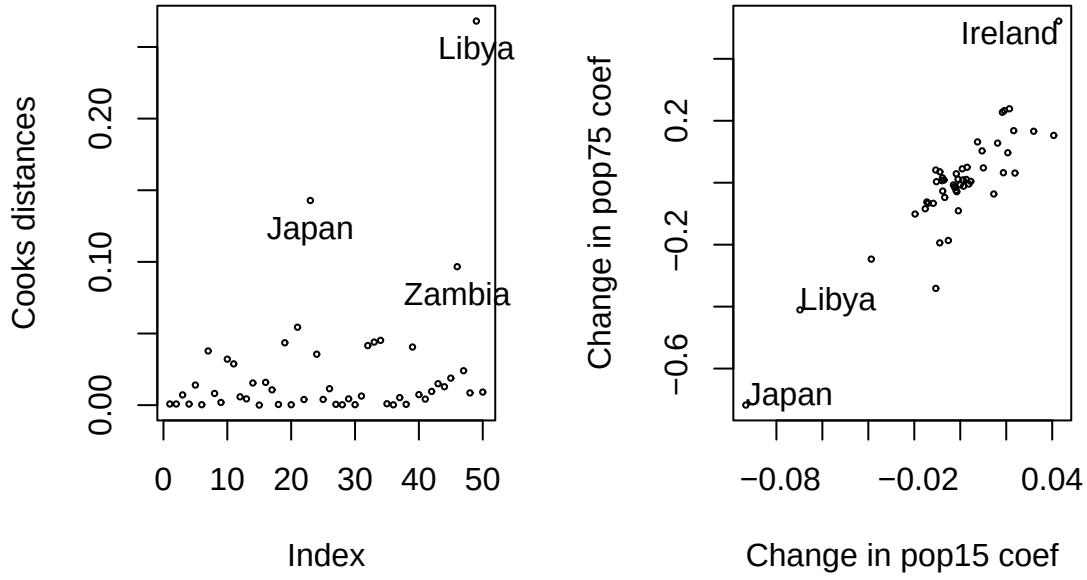

Figure 7.4: Cook Statistics and $\hat{\beta}-\hat{\beta}_{(i)}$'s for the savings data

Which ones are large? We now exclude the largest one and see how the fit changes:

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

Residual standard error: 3.79 on 44 degrees of freedom
Multiple R-Squared: 0.355,      Adjusted R-squared: 0.297
F-statistic: 6.07 on 4 and 44 degrees of freedom,       p-value: 0.000562
```

Compared to the full data fit:

```
> summary(g)
Coefficients:
             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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

What changed? The coefficient for `ddpi` changed by about 50%. We don't like our estimates to be so sensitive to the presence of just one country. It would be rather tedious to do this for each country but there's a quicker way:

```
> ginf <- lm.influence(g)
> plot(ginf$coef[,2],ginf$coef[,3],xlab="Change in pop15 coef",
              ylab="Change in pop75 coef")
> identify(ginf$coef[,2],ginf$coef[,3],countries)
```

We just plotted the change in the second and third parameter estimates when a case is left out as seen in the second panel of Figure 7.4. Try this for the other estimates - which countries stick out? Consider Japan:

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

Residual standard error: 3.74 on 44 degrees of freedom
Multiple R-Squared: 0.277,      Adjusted R-squared: 0.211
F-statistic: 4.21 on 4 and 44 degrees of freedom,       p-value: 0.00565
```

Compare to the full data fit - what qualitative changes do you observe? Notice that the `ddpi` term is no longer significant and that the the R^2 value has decreased a lot.

7.5 Residual Plots

Outliers and influential points indicate cases that are in some way individually unusual but we also need to check the assumptions of the model. Plot $\hat{\varepsilon}$ against $\hat{y}$. This is the most important diagnostic plot that

you can make. If all is well, you should see constant variance in the vertical ($\hat{\varepsilon}$) direction and the scatter should be symmetric vertically about 0. Things to look for are heteroscedascity (non-constant variance) and nonlinearity (which indicates some change in the model is necessary). In Figure 7.5, these three cases are illustrated.

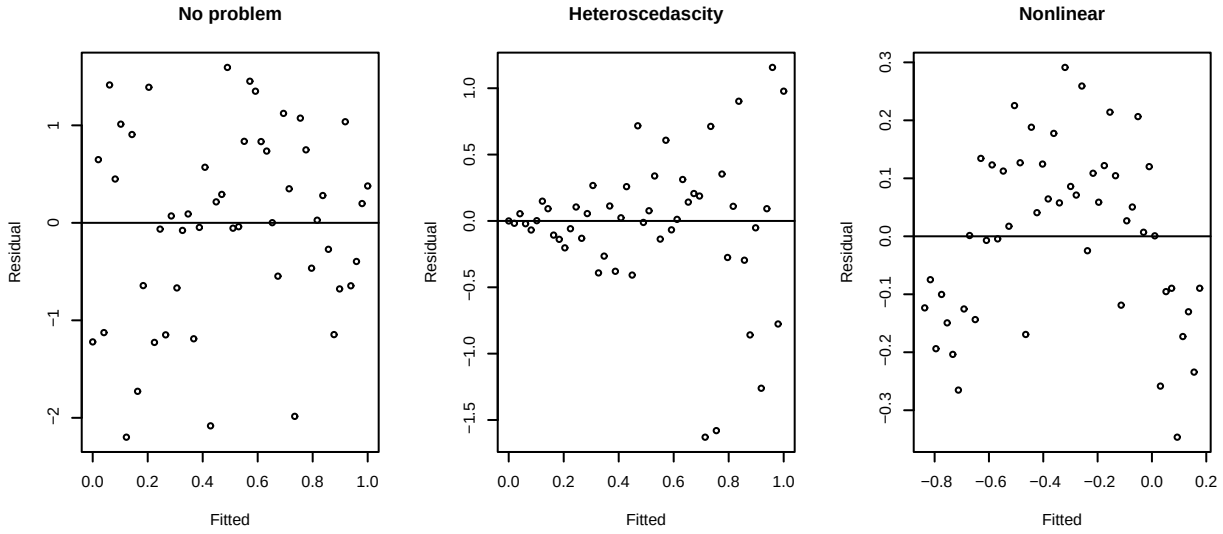

Figure 7.5: Residuals vs Fitted plots - the first suggests no change to the current model while the second shows non-constant variance and the third indicates some nonlinearity which should prompt some change in the structural form of the model

You should also plot $\hat{\varepsilon}$ against x_i (for predictors that are both in and out of the model). Look for the same things except in the case of plots against predictors not in the model, look for any relationship which might indicate that this predictor should be included.

We illustrate this using the savings dataset as an example again:

```
> g <- lm(sr ~ pop15+pop75+dpi+ddpi,savings)
```

First the residuals vs. fitted plot and the abs(residuals) vs. fitted plot.

```
> plot(g$fit,g$res,xlab="Fitted",ylab="Residuals")
> abline(h=0)
> plot(g$fit,abs(g$res),xlab="Fitted",ylab="|Residuals|")
```

The plots may be seen in the first two panels of Figure 7.5. What do you see? The latter plot is designed to check for non-constant variance only. It folds over the bottom half of the first plot to increase the resolution for detecting non-constant variance. The first plot is still needed because non-linearity must be checked.

A quick way to check non-constant variance is this regression:

```
> summary(lm(abs(g$res) ~ g$fit))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.840      1.186    4.08  0.00017
g$fit         -0.203      0.119   -1.72  0.09250
```

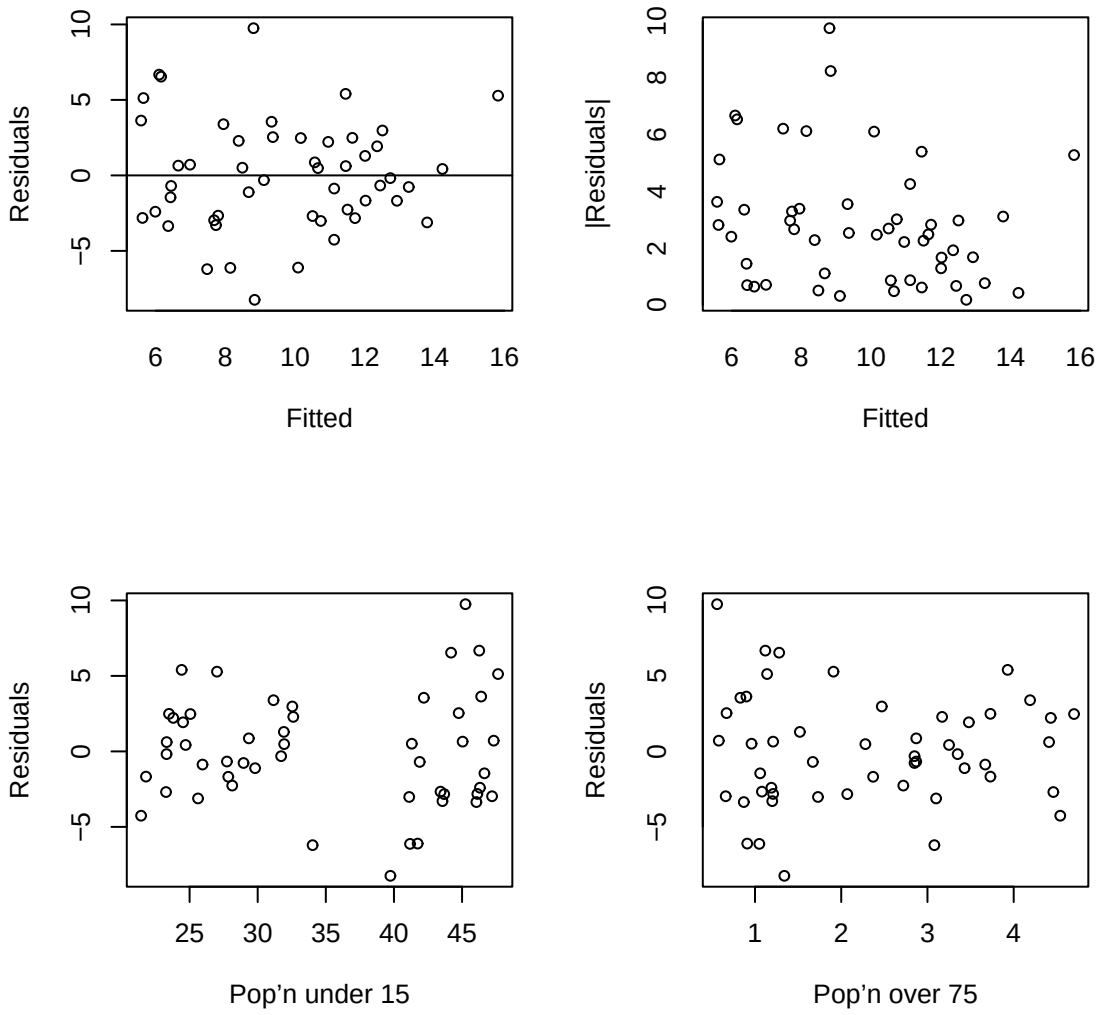

Figure 7.6: Residual plots for the savings data

```
Residual standard error: 2.16 on 48 degrees of freedom
Multiple R-Squared: 0.0578,     Adjusted R-squared: 0.0382
F-statistic: 2.95 on 1 and 48 degrees of freedom,       p-value: 0.0925
```

This test is not quite right as some weighting should be used and the degrees of freedom should be adjusted but there doesn't seem to be a clear problem with non-constant variance.

It's often had to judge residual plots without prior experience so let's show how to generate some of the artificial variety. The following four `for()` loops show

1. Constant Variance
2. Strong non-constant variance
3. Mild non-constant variance
4. Non-linearity

```
> par(mfrow=c(3,3))
> for(i in 1:9) plot(1:50,rnorm(50))
> for(i in 1:9) plot(1:50,(1:50)*rnorm(50))
> for(i in 1:9) plot(1:50,sqrt((1:50))*rnorm(50))
> for(i in 1:9) plot(1:50,cos((1:50)*pi/25)+rnorm(50))
> par(mfrow=c(1,1))
```

In this case we know the truth - do you think you would be able to come to the right conclusions for real data? Repeat to get an idea of the usual amount of variation. I recommend the artificial generation of plots as a way to "calibrate" diagnostic plots. It's often hard to judge whether an apparent feature is real or just random variation. Repeated generation of plots under a model where there is no violation of the assumption that the diagnostic plot is designed to check is helpful in making this judgement.

Now look at some residuals against predictor plots:

```
> plot(savings$pop15,g$res,xlab="Population under 15",ylab="Residuals")
> plot(savings$pop75,g$res,xlab="Population over 75",ylab="Residuals")
```

The plots may be seen in the second two panels of Figure 7.5. Can you see the two groups? Let's compare and test the variances. Given two independent samples from normal distributions, we can test for equal variance using the test statistic of the ratio of the two variance. The null distribution is a F with degrees of freedom given by the two samples.

```
> var(g$res[savings$pop15 > 35])/var(g$res[savings$pop15 <35])
[1] 2.7851
> table(savings$pop15 > 35)
FALSE  TRUE
   27    23
> 1-pf(2.7851,22,26)
[1] 0.0067875
```

A significant difference is seen

7.6 Non-Constant Variance

There are two approaches to dealing with non-constant variance. Weighted least squares is appropriate when the form of the non-constant variance is either known exactly or there is some known parametric form. Alternatively, one can transform y to $h(y)$ where $h()$ is chosen so that var $h(y)$ is constant. To see how to choose $h()$ consider this

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

We ignore the higher order terms. For var $h(y)$ to be constant we need

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

which suggests

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

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

Graphically one tends to see *SDy* rather then var *y*. Sometimes $y_i \leq 0$ for some i in which case the transformations may choke. You can try $\log(y+\delta)$ for some small δ but this makes interpretation difficult.

Consider the residual-fitted plot for the Galapagos data:

```
> gg <- lm(Species ~ Area + Elevation + Scruz + Nearest + Adjacent, gala)
> plot(gg$fit,gg$res,xlab="Fitted",ylab="Residuals",
                  main="Untransformed Response")
```

We can see non-constant variance in the first plot of Figure 7.7.

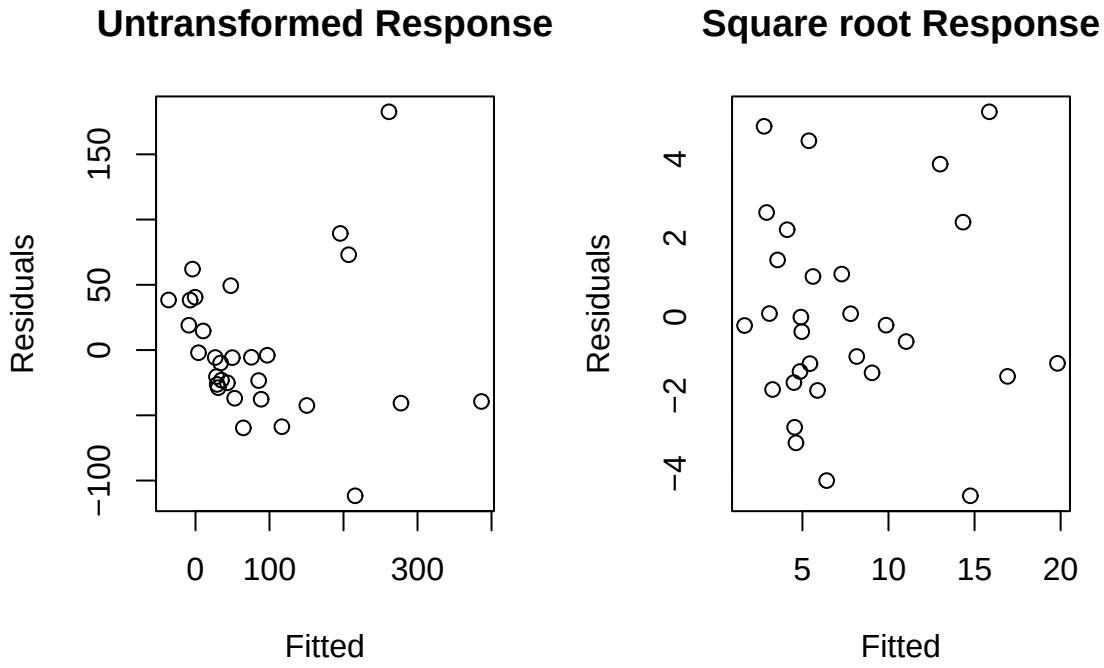

Figure 7.7: Residual-Fitted plots for the Galapagos data before and after transformation

We guess that a square root transformation will give us constant variance:

```
> gs <- lm(sqrt(Species) ~ Area+ Elevation+ Scruz+ Nearest+ Adjacent, gala)
> plot(gs$fit,gs$res,xlab="Fitted",ylab="Residuals",
                  main="Square root Response")
```

We see in the second plot of Figure 7.7 that the variance is now constant. Our guess at a variance stabilizing transformation worked out here, but had it not, we could always have tried something else. The square root transformation is often appropriate for count 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. It might be even better to go with a poisson regression rather than the normal-based regression we are using here.

There are more formal tests for non-constant variance — for example one could start by regressing $|\hat{\varepsilon}|$ on y or x_i but there is a problem in specifying the alternative hypothesis for such a test. A formal test may be good at detecting a particular kind of non-constant 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 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 one can be sure that nothing is seriously wrong with the assumptions. For this reason, I usually prefer a graphical approach to diagnostics.

7.7 Non-Linearity

Lack of fit tests can be used when there is replication which doesn't happen too often, but even if you do have it, the tests don't tell you how to improve the model. How do we check if the systematic part ($Ey = X\beta$) of the model is correct? We can look at

1. Plots of $\hat{\varepsilon}$ against $\hat{y}$ and x_i
2. Plots of y against each x_i.

but what about the effect of other x on the y vs. x_i plot?
Partial Regression or *Added Variable* plots can help isolate the effect of x_i on y.

1. Regress y on all x except x_i, get residuals $\hat{\delta}$. This represents y with the other X-effect taken out.
2. Regress x_i on all x except x_i, get residuals $\hat{\gamma}$ This represents x_i with the other X-effect taken out.
3. Plot $\hat{\delta}$ against $\hat{\gamma}$

The slope of a line fitted to the plot is $\hat{\beta}_i$ which adds some insight into the meaning of regression coefficients. Look for non-linearity and outliers and/or influential points.

Partial Residual plots are a competitor to added variable plots. These plot $\hat{\varepsilon} + \hat{\beta}_i x_i$ against x_i. To see where this comes from, look at the response with the predicted effect of the other X removed:

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

Again the slope on the plot will be $\hat{\beta}_i$ and the interpretation is the same. Partial residual plots are reckoned to be better for non-linearity detection while added variable plots are better for outlier/influential detection.

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

```
> d <- lm(sr ~ pop75 + dpi + ddpi,savings)$res
> m <- lm(pop15 ~ pop75 + dpi + ddpi,savings)$res
> plot(m,d,xlab="pop15 residuals",ylab="Saving residuals",
                    main="Partial Regression")
```

Compare the slope on the plot to the original regression and show the line on the plot (see Figure 7.7).

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

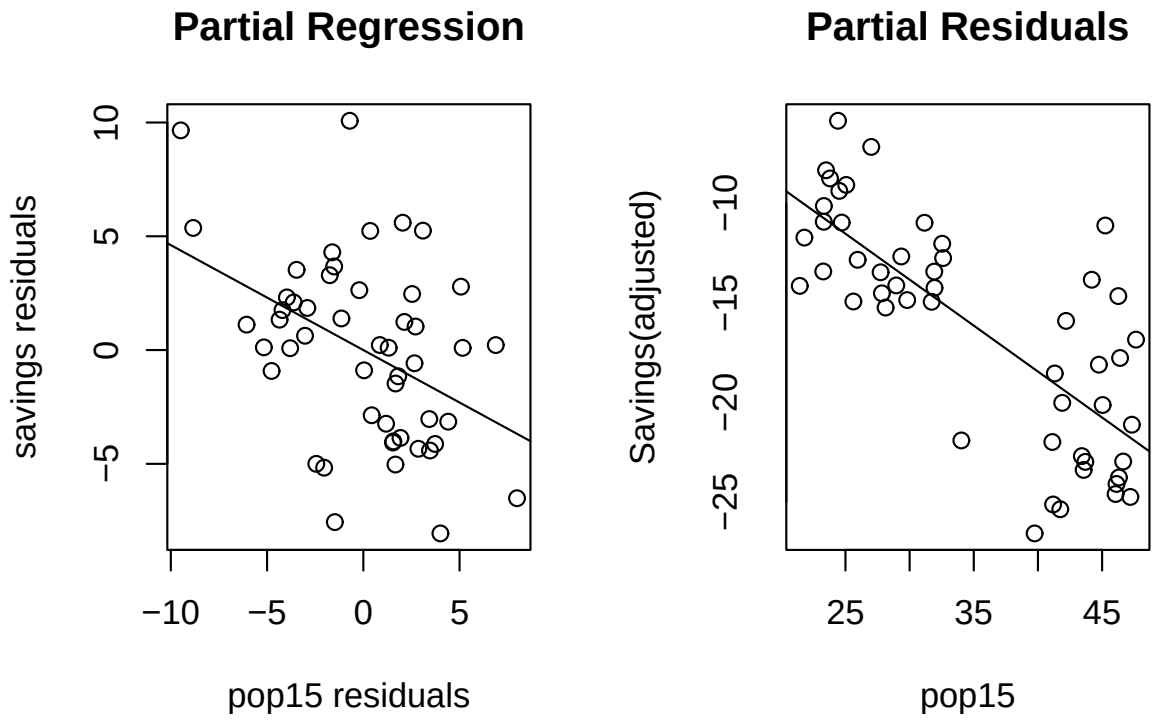

Figure 7.8: Partial regression and residual plots for the savings data

Notice how the slope in the plot and the slope for `pop15` in the regression fit are the same.

The partial regression plot also provides some intuition about the meaning 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.

A partial residual plot is easier to do:

```
> plot(savings$pop15,g$res+g$coef['pop15']*savings$pop15,xlab="pop'n under 15",
  ylab="Saving(adjusted)",main="Partial Residual")
> abline(0,g$coef['pop15'])
```

Or more directly:

```
> prplot(g,1)
```

Might there be a different relationship in the two groups?

```
> g1 <- lm(sr ~ pop15+pop75+dpi+ddpi,savings,subset=(pop15 > 35))
> g2 <- lm(sr ~ pop15+pop75+dpi+ddpi,savings,subset=(pop15 < 35))
> summary(g1)
Coefficients:
             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

Residual standard error: 4.45 on 18 degrees of freedom
Multiple R-Squared: 0.156,      Adjusted R-squared: -0.0319
F-statistic: 0.83 on 4 and 18 degrees of freedom,       p-value: 0.523

> summary(g2)
Coefficients:
             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

Residual standard error: 2.77 on 22 degrees of freedom
Multiple R-Squared: 0.507,      Adjusted R-squared: 0.418
F-statistic: 5.66 on 4 and 22 degrees of freedom,       p-value: 0.00273
```

Can you see the difference? The graphical analysis has shown a relationship in the data that a purely numerical analysis might easily have missed.

Higher dimensional plots can also be useful for detecting structure that cannot be seen in two dimensions. These are interactive in nature so you need to try them to see how they work. Two ideas are

1. Spinning - 3D plots where color, point size and rotation are used to give illusion of a third dimension.
2. Brushing - Two or more plots are linked so that point which are *brushed* in one plot are highlighted in another.

These tools look good but it's not clear whether they actually are useful in practice. Certainly there are communication difficulties as these plots cannot be easily printed. Many statistical packages allow for this kind of investigation. XGobi is a useful free UNIX-based tool for exploring higher dimensional data that has now been made extended to Windows also as Ggobi. See `www.ggobi.org`

```
> library(xgobi)
> xgobi(savings)
```

or

```
> library(Rggobi)
> ggobi(savings)
```

Most of the functionality can be discovered by experimentation and the online help.

7.8 Assessing Normality

The test 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. The steps are:

1. Sort the residuals: $\hat{\varepsilon}_{[1]} \leq \ldots \hat{\varepsilon}_{[n]}$
2. Compute $u_i = \Phi^{-1}(\frac{i}{n+1})$
3. Plot $\hat{\varepsilon}_{[i]}$ against u_i. If the residuals are normally distributed an approximately straight-line relationship will be observed.

Let's try it out on the same old data:

```
> qqnorm(g$res,ylab="Raw Residuals")
> qqline(g$res)
```

See the first plot of Figure 7.8 - `qqline()` adds a line joining the first and third quartiles - it's useful as a guide. We can plot the (externally) studentized residuals:

```
> qqnorm(rstudent(g),ylab="Studentized residuals")
> abline(0,1)
```

See the second plot of the figure. Because these residuals have been normalized, they should lie along a 45 degree line.

Histograms and boxplots are not as sensitive for checking normality:

```
> hist(g$res,10)
> boxplot(g$res,main="Boxplot of savings residuals")
```

We can get an idea of the variation to be expected in QQ-plots in the following experiment. I generate data from different distributions:

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

Here's how to generate 9 replicates at a time from each of these test cases:

```
> oldpar <- par()
> par(mfrow=c(3,3))
> for(i in 1:9) qqnorm(rnorm(50))
> for(i in 1:9) qqnorm(exp(rnorm(50)))
> for(i in 1:9) qqnorm(rcauchy(50))
> for(i in 1:9) qqnorm(runif(50))
> par(oldpar)
```

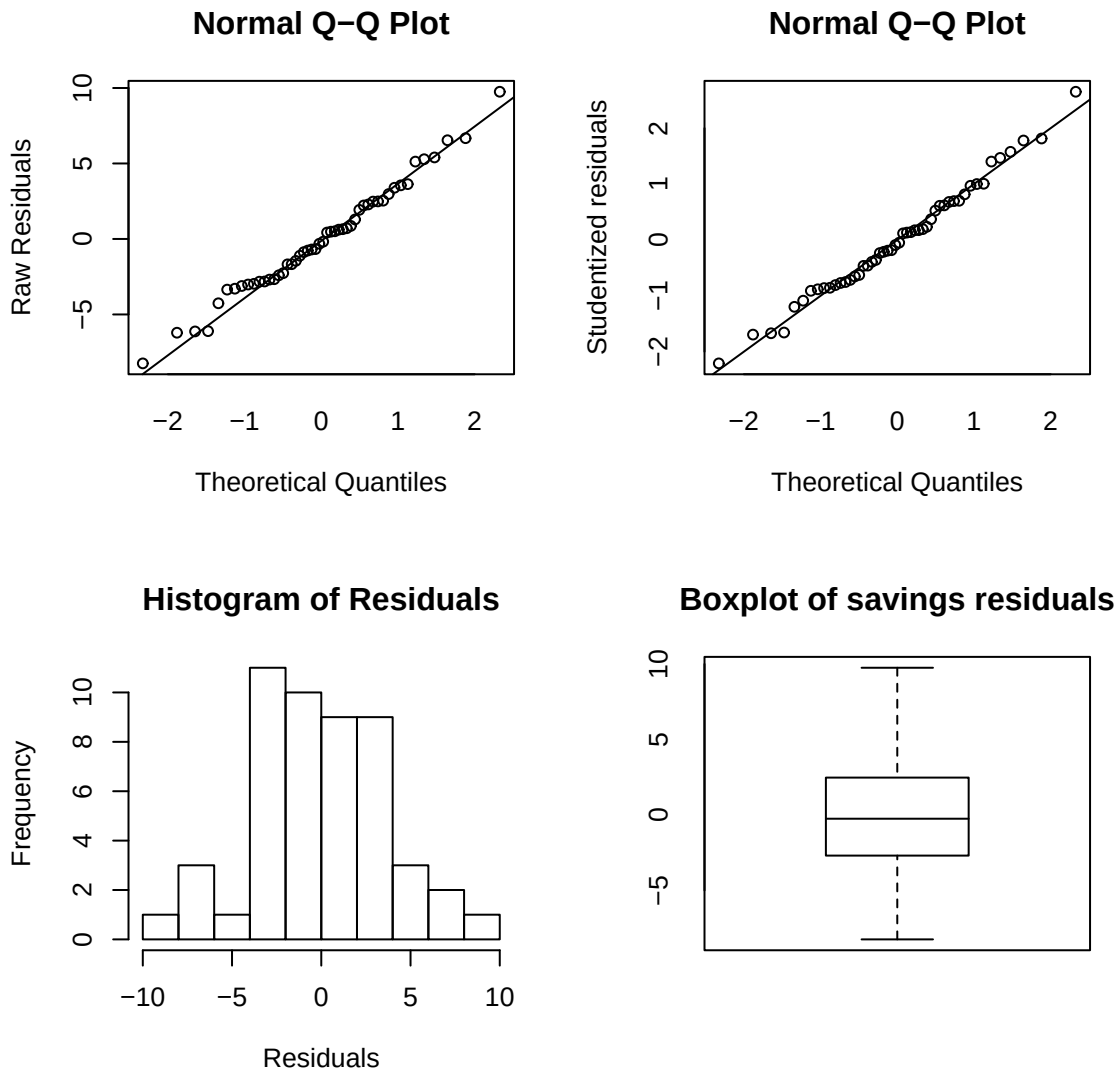

Figure 7.9: Normality checks for the savings data

We save the original settings for the graphics layout in `oldpar` and restore it after we are done. This is a useful trick when you want to experiment with changing these settings.

In Figure 7.8, you can see examples of all four cases:

It's not always easy to diagnose the problem in QQ plots.

The consequences of non-normality are

1. that the least squares estimates may not be optimal - they will still be BLUE but other *robust* estimators may be more effective.

2. that the tests and confidence intervals are invalid. However, it has been shown that only really long-tailed distributions cause a problem. Mild non-normality can safely be ignored and the larger the sample size the less troublesome the non-normality.

What to do?

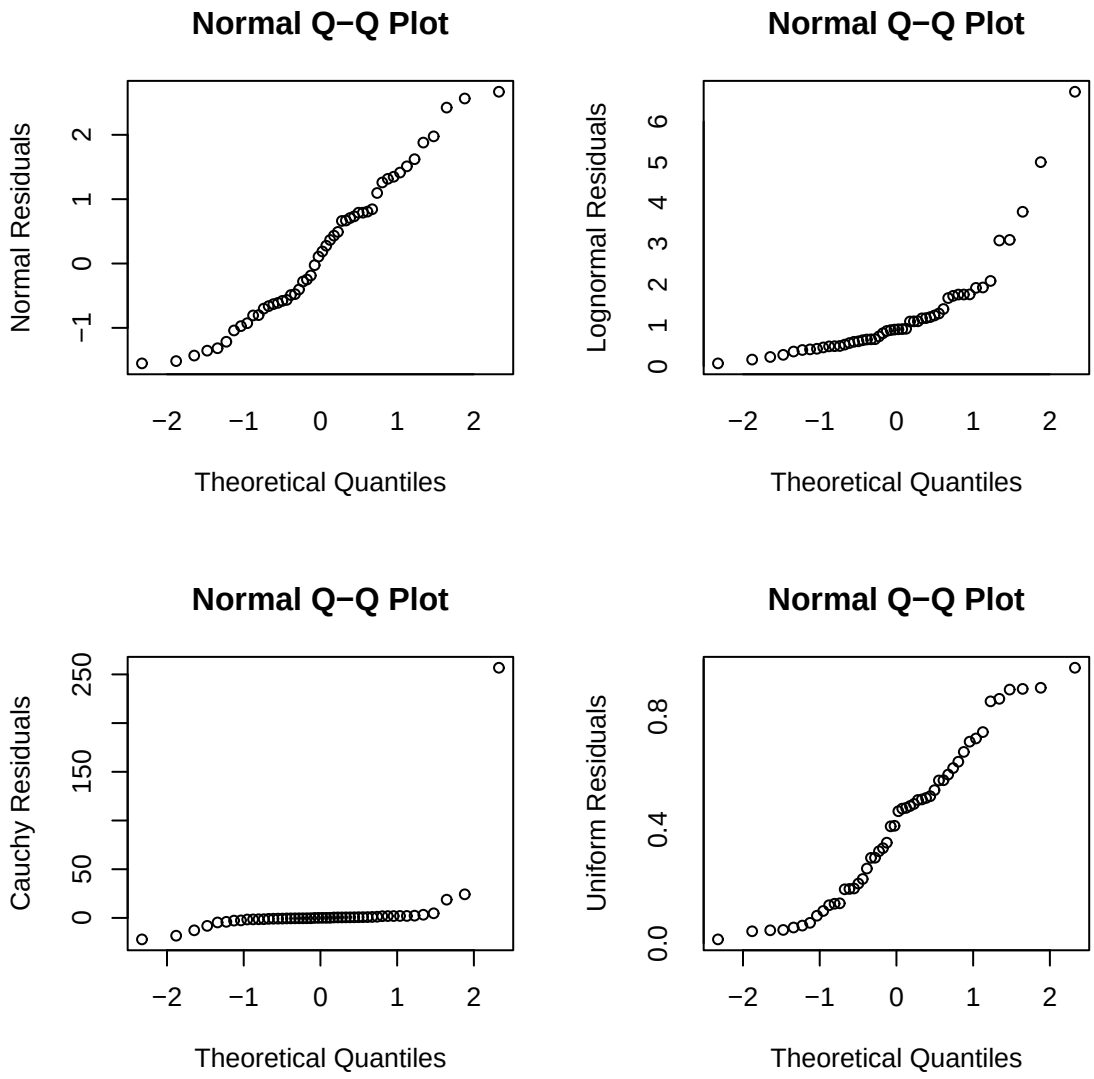

Figure 7.10: QQ plots of simulated data

1. A transformation of the response may solve the problem - this is often true for skewed errors.
2. Other changes in the model may help.
3. Accept non-normality and base the inference on the assumption of another distribution or use resampling methods such as the bootstrap or permutation tests. You don't want to do this unless absolutely necessary. Alternatively use robust methods which give less weight to outlying points. This is appropriate for long tailed distributions.
4. For short-tailed distributions, the consequences of non-normality are not serious and can reasonably be ignored.

There are formal tests for normality such as the Kolmogorov-Smirnov test but these are not as flexible as the Q-Q plot. The p-value is not very helpful as an indicator of what action to take. After all, 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.

7.9 Half-normal plots

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

The steps are:

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

We are usually not looking for a straight line relationship since we do not necessarily expect a positive normal distribution for quantities like the leverages. (If the X is multivariate normal, the leverages will have a χ^2_p distribution but there is usually no good reason to assume multivariate normality for the X.) 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 and Cook statistics for the savings data:

```
> halfnorm(lm.influence(g)$hat,labs=countries,ylab="Leverages")
> halfnorm(cooks.distance(g),labs=countries,ylab="Cook Statistics")
```

The plots are chosen in Figure 7.11 — I have plotted the country name instead of just a dot for the largest two cases respectively to aid identification. The `halfnorm()` function comes from the book library.

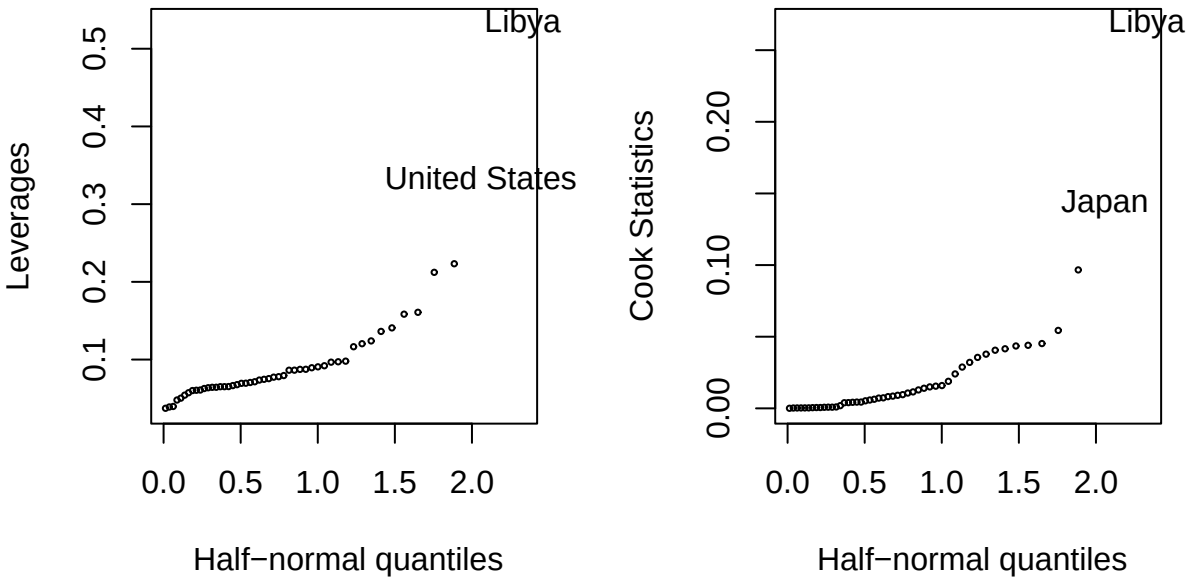

Figure 7.11: Half-normal plots for the leverages and Cook statistics

Libya shows up clearly as unusual in both plots

7.10 Correlated Errors

We assume that the errors are uncorrelated but for temporally or spatially related data this may well be untrue. For this type of data, it is wise to check the uncorrelated assumption.

1. Plot $\hat{\varepsilon}$ against time.

2. Use formal tests like the Durbin-Watson or the run test.

If you do have correlated errors, you can use GLS. This does require that you know Σ or more usually that you can estimate it. In the latter case, an iterative fitting procedure will be necessary as in IRWLS. Such problems are common in Econometrics.

For the example, we use some taken from an environmental study that measured the four variables ozone, solar radiation, temperature, and wind speed for 153 consecutive days in New York.

```
> data(airquality)
> airquality
    Ozone Solar.R Wind Temp Month Day
1      41     190  7.4   67     5   1
2      36     118  8.0   72     5   2
3      12     149 12.6   74     5   3
4      18     313 11.5   62     5   4
5      NA      NA 14.3   56     5   5
etc..
```

We notice that there are some missing values. Take a look at the data: (plot not shown)

```
> pairs(airquality,panel=panel.smooth)
```

We fit a standard linear model and check the residual-fitted plot in Figure 7.10.

```
> g <- lm(Ozone ~ Solar.R + Wind + Temp,airquality)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -64.3421    23.0547   -2.79   0.0062
Solar.R       0.0598     0.0232    2.58   0.0112
Wind         -3.3336     0.6544   -5.09  1.5e-06
Temp          1.6521     0.2535    6.52  2.4e-09

Residual standard error: 21.2 on 107 degrees of freedom
Multiple R-Squared: 0.606,      Adjusted R-squared: 0.595
F-statistic: 54.8 on 3 and 107 degrees of freedom,      p-value:    0
> plot(g$fit,g$res,xlab="Fitted",ylab="Residuals",
                  main="Untransformed Response")
```

Notice how there are only 107 degrees corresponding to the 111 complete observations. The default behavior in R when performing a regression with missing values is to exclude any case that contains a missing value. We see some non-constant variance and nonlinearity and so we try transforming the response:

```
> gl <- lm(log(Ozone) ~ Solar.R + Wind + Temp,airquality)
> plot(gl$fit,gl$res,xlab="Fitted",ylab="Residuals",main="Logged Response")
```

Suppose we are now otherwise satisfied with this model and want to check for serial correlation. The missing values in the data were not used in the construction of the model but this also breaks up the sequential pattern in the data. I get round this by reintroducing missing values into the residuals corresponding to the omitted cases.

```
> res <- rep(NA,153)
> res[as.numeric(row.names(na.omit(airquality)))] <- gl$res
```

First make an index plot of the residuals — see Figure 7.10.

```
> plot(res,ylab="Residuals",main="Index plot of residuals")
```

Is there any evidence of serial correlation? Now plot successive residuals:

```
> plot(res[-153],res[-1],xlab=expression(hat(epsilon)[i]),
              ylab=expression(hat(epsilon)[i+1]))
```

Do you see any problem? Let's check

```
> summary(lm(res[-1] ~ -1+res[-153]))
Coefficients:
          Estimate Std. Error t value Pr(>|t|)
res[-153]    0.110      0.105    1.05      0.3

Residual standard error: 0.508 on 91 degrees of freedom
Multiple R-Squared: 0.0119,     Adjusted R-squared: 0.00107
F-statistic:  1.1 on 1 and 91 degrees of freedom,       p-value: 0.297
```

We omitted the intercept term because the residuals have mean zero. We see that there is no significant correlation.

You can plot more than just successive pairs if you suspect a more complex dependence. For spatial data, more complex checks are required.

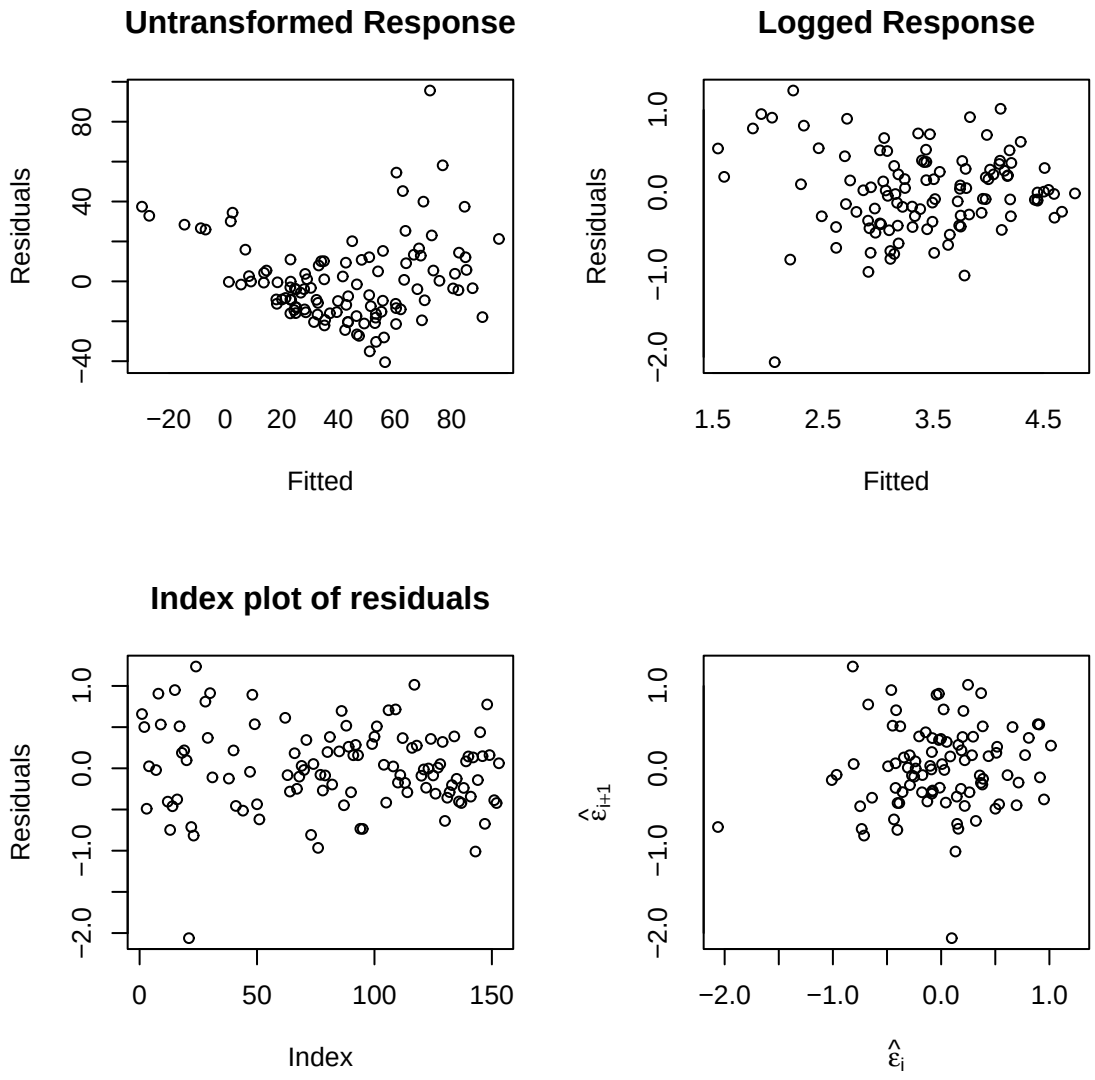

Figure 7.12: Checking for correlated errors - Index plot and scatterplot of successive residuals

Chapter 8

Transformation

Transformations of the response and predictors can improve the fit and correct violations of model assumptions such as constant error variance. We may also consider adding additional predictors that are functions of the existing predictors like quadratic or crossproduct terms.

8.1 Transforming the response

Let's 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

  $$y = \exp(\beta_0 + \beta_1 x) \cdot \exp(\varepsilon)$$

  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 that the errors enter multiplicatively in the original scale. Notice that if we believe the proper model for y to be

  $$y = \exp(\beta_0 + \beta_1 x) + \varepsilon$$

  then we cannot linearize this model and non-linear regression methods would need to be applied.

  As a practical matter, we often do not know how the errors enter the model, additively, multiplicatively or otherwise. The usual approach is to try different transforms and then check the residuals to see whether they satisfy the conditions required for linear regression. Unless you have good information that the error enters in some particular way, this is the simplest and most appropriate way to go.

- Although you may transform the response, you will probably need to express predictions in the original scale. This is simply a matter of 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 — remember what happened with the prediction confidence intervals for Galapagos data.

- Regression coefficients will need to be interpreted with respect to the transformed scale. There is no straightforward way of backtransforming them to values that can 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 one 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:

$$\log\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p$$
$$\hat{y} = e^{\hat{\beta}_0} e^{\hat{\beta}_1 x_1} \ldots e^{\hat{\beta}_p x_p}$$

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 the usual additive manner.

The Box-Cox method is a popular way to determine a tranformation 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 $y \to t_\lambda(y)$ where the family of transformations indexed by λ is

$$t_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log y & \lambda = 0 \end{cases}$$

For fixed $y > 0$, $t_\lambda(y)$ is continuous in λ. Choose λ using maximum likelihood. The profile log-likelihood assuming normality of the errors is

$$L(\lambda) = -\frac{n}{2}\log(\mathrm{RSS}_\lambda/n) + (\lambda - 1)\sum \log y_i$$

where RSS_λ is the RSS when $t_\lambda(y)$ is the response. You can compute $\hat{\lambda}$ exactly to maximize this but usually $L(\lambda)$ is just maximized over a grid of values such as $\{-2, -1, -1/2, 0, 1/2, 1, 2\}$. This ensures that the chosen $\hat{\lambda}$ is more easily interpreted. For example, if $\hat{\lambda} = 0.46$, it would be hard to explain what this new response means, but $\sqrt{y}$ would be easier.

Transforming the response can make the model harder to interpret so we don't want to do it unless it's really necessary. One way to check this is to form a confidence interval for λ. A $100(1-\alpha)\%$ confidence interval for λ is

$$\{\lambda : \quad L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi^{2(1-\alpha)}_1\}$$

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^2_1.

Does the response in the savings data need transformation? You'll need a function from the MASS library:

```
> library(MASS)
```

Try it out on the savings dataset and plot the results.

```
> data(savings)
> g <- lm(sr ~ pop15+pop75+dpi+ddpi,savings)
> boxcox(g,plotit=T)
> boxcox(g,plotit=T,lambda=seq(0.5,1.5,by=0.1))
```

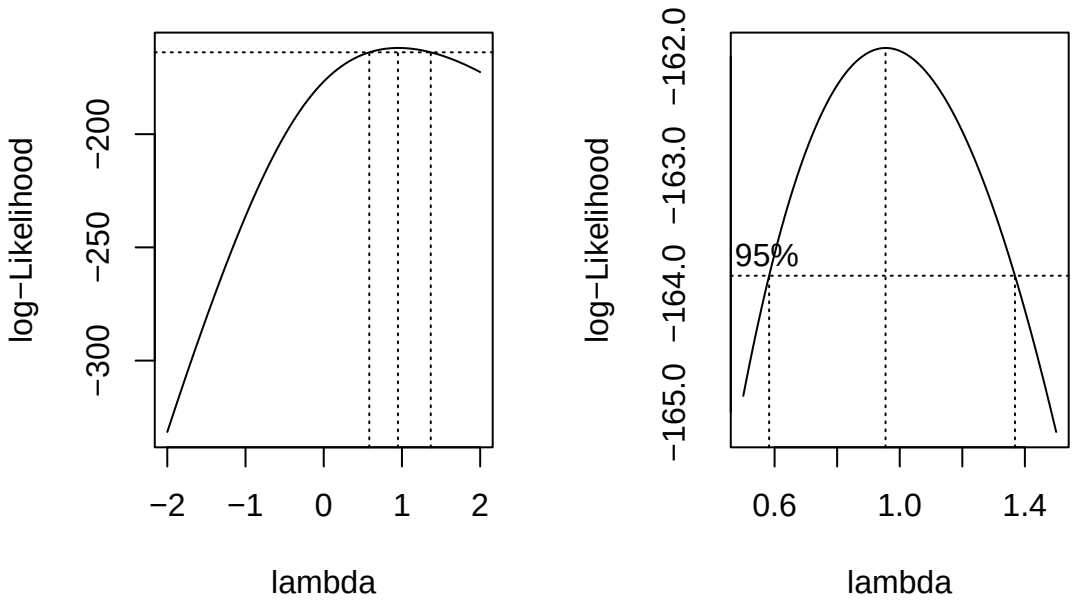

Figure 8.1: Log-likelihood plots for the Box-Cox transformation of the savings data

The first plot shown in Figure 8.1 is too broad. I 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)
> g <- lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,gala)
> boxcox(g,plotit=T)
> boxcox(g,lambda=seq(0.0,1.0,by=0.05),plotit=T)
```

The plots are shown in Figure 8.2. We see that perhaps a cube-root 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.

Notes

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. What if some $y_i < 0$? Sometimes adding a constant to all y can work provided that constant is small.

3. If $\max_i y_i / \min_i y_i$ is small then the Box-Cox won't do anything because power transforms are well approximated by linear transformations over short intervals.

4. Should the estimation of λ count as an extra parameter to be taken account of 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.

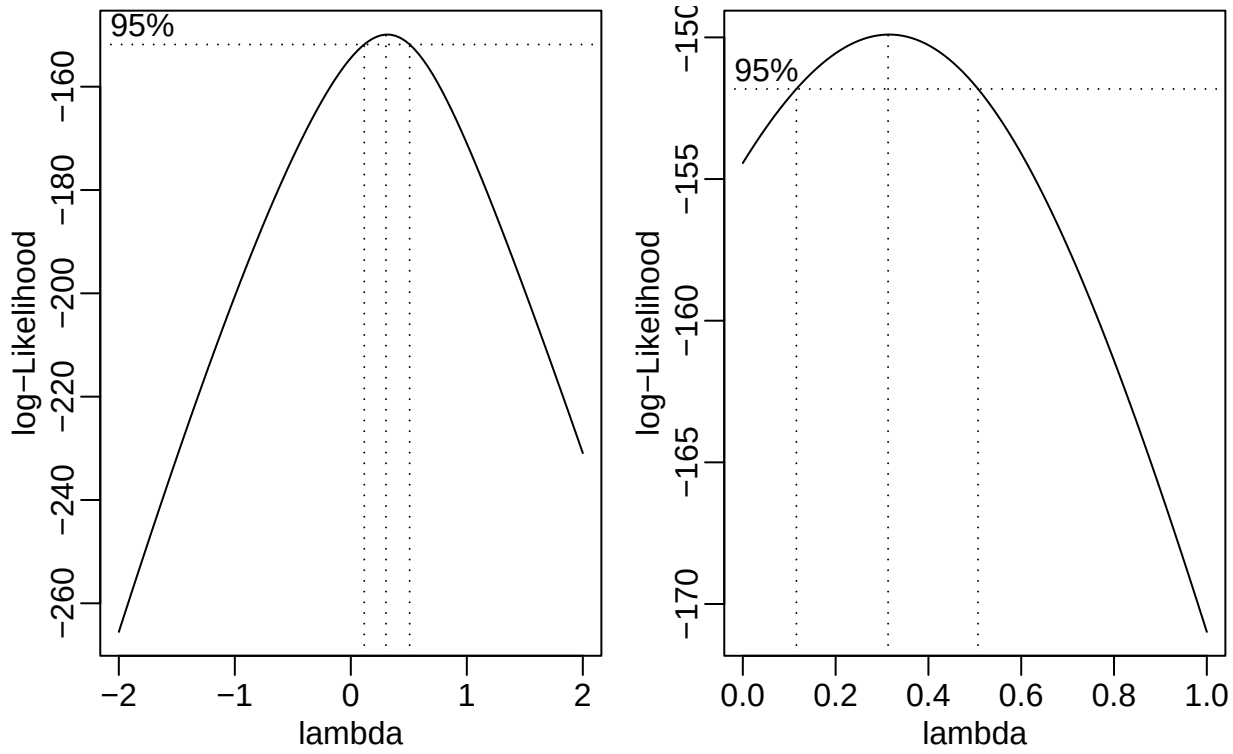

Figure 8.2: Log-likelihood plots for the Box-Cox transformation of the Galápagos data

The Box-Cox method is not the only way of transforming the predictors. 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.

8.2 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 takes time and furthermore the correct transformation for each predictor may depend on getting the others right too. Partial residuals are a good way of finding suggestions for transforming the predictors

8.2.1 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 shown in Figure 8.3.

```
> g1 <- lm(sr ~ pop15, savings, subset=(pop15 < 35))
> g2 <- lm(sr ~ pop15, savings, subset=(pop15 > 35))
> plot(savings$pop15,savings$sr,xlab="Pop'n under 15",ylab="Savings Rate")
```

```
> abline(v=35,lty=5)
> segments(20,g1$coef[1]+g1$coef[2]*20,35,g1$coef[1]+g1$coef[2]*35)
> segments(48,g2$coef[1]+g2$coef[2]*48,35,g2$coef[1]+g2$coef[2]*35)
```

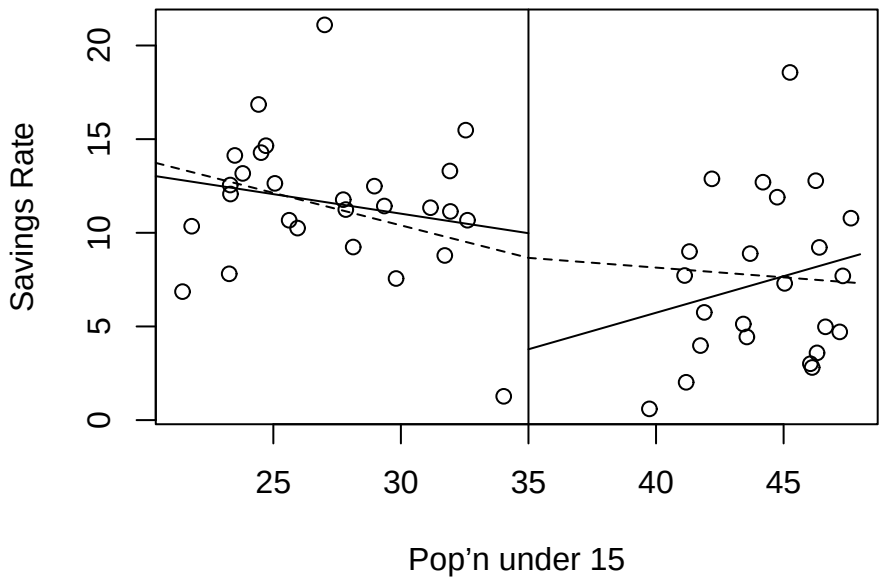

Figure 8.3: Subset regression fit is shown with the solid line while the broken stick regression is shown with the dotted line

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 varies, then this is unsatisfactory. One solution to this problem is the broken stick regression fit. Define two *basis functions*:

$$B_l(x) = \begin{cases} c-x & \texttt{if } x < c \\ 0 & \texttt{otherwise} \end{cases}$$

and

$$B_r(x) = \begin{cases} x-c & \texttt{if } x > c \\ 0 & \texttt{otherwise} \end{cases}$$

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 above. 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)
> gb <- lm(sr ~ lhs(pop15) + rhs(pop15), savings)
```

```
> x <- seq(20,48,by=1)
> py <- gb$coef[1]+gb$coef[2]*lhs(x)+gb$coef[3]*rhs(x)
> lines(x,py,lty=2)
```

The two (dotted) lines now meet at 35 as shown in Figure 8.3. 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 the two groups of countries are so different and that there are so few countries in the middle region, that we might not want to impose continuity at all.

We can have more than one knotpoint simply by defining more pairs of 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.

8.2.2 Polynomials

Another way of generalizing the $X\beta$ part of the model is to add polynomial terms. In the one-predictor case, we have

$$y = \beta_0 + \beta_1 x + \ldots + \beta_d x^d + \varepsilon$$

which allows for a more flexible relationship although we usually don't believe it exactly represents any underlying reality.

There are two ways to choose d:

1. Keep adding terms until the added term is not statistically significant.

2. Start with a large d — eliminate not statistically significant terms starting with the highest order term.

Warning: Do not eliminate lower order terms from the model 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 so brittle to such changes in the scale which ought to be inconsequential.

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

Residual standard error: 4.31 on 48 degrees of freedom
Multiple R-Squared: 0.0929,     Adjusted R-squared: 0.074
F-statistic: 4.92 on 1 and 48 degrees of freedom,       p-value: 0.0314
```

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

Residual standard error: 4.08 on 47 degrees of freedom
Multiple R-Squared: 0.205,      Adjusted R-squared: 0.171
F-statistic: 6.06 on 2 and 47 degrees of freedom,       p-value: 0.00456
```

Again the p-value of `ddpi`^{2} 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

Residual standard error: 4.12 on 46 degrees of freedom
Multiple R-Squared: 0.205,      Adjusted R-squared: 0.153
F-statistic: 3.95 on 3 and 46 degrees of freedom,       p-value: 0.0137
```

p-value of `ddpi`^{3} is not significant so stick with the quadratic. What do you notice about the other p-values? Why do we find a quadratic model when the previous analysis on transforming predictors found that the `ddpi` variable did not need transformation? Check that starting from a large model (including the fourth power) and working downwards gives the same result.

To illustrate the point about the significance of lower order terms, 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

Residual standard error: 4.08 on 47 degrees of freedom
Multiple R-Squared: 0.205,      Adjusted R-squared: 0.171
F-statistic: 6.06 on 2 and 47 degrees of freedom,       p-value: 0.00456
```

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. No advantage would be gained.

You have to refit the model each time a term is removed and for large d there can be problem with numerical stability. Orthogonal polynomials get round this problem by defining

$$\begin{aligned} z_1 &= a_1 + b_1 x \\ z_2 &= a_2 + b_2 x + c_2 x^2 \\ z_3 &= a_3 + b_3 x + c_3 x^2 + d_3 x^3 \end{aligned}$$

etc. where the coefficients $a, b, c, \ldots$ are chosen so that $z_i^T z_j = 0$ when $i \neq j$. The 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.

```
> g <- lm(sr ~ poly(ddpi,4),savings)
> summary(g)
Coefficients:
               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

Residual standard error: 4.13 on 45 degrees of freedom
Multiple R-Squared: 0.218,      Adjusted R-squared: 0.149
F-statistic: 3.14 on 4 and 45 degrees of freedom,       p-value: 0.0232
```

Can you see how we come to the same conclusion as above with just this summary? We can verify the orthogonality of the design matrix when using orthogonal polynomials:

```
> x <- model.matrix(g)
> dimnames(x) <- list(NULL,c("Int","power1","power2","power3","power4"))
> round(t(x) %*% x,3)
       Int power1 power2 power3 power4
Int     50      0      0      0      0
power1   0      1      0      0      0
power2   0      0      1      0      0
power3   0      0      0      1      0
power4   0      0      0      0      1
```

You can have more than two predictors as can be seen in this *response surface* model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2$$

8.3 Regression Splines

Polynomials have the advantage of smoothness but the disadvantage that each data point affects the fit globally. This is because the power functions used for the polynomials take non-zero 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 may define a cubic B-spline basis on the interval $[a, b]$ by the following requirements on the interior basis functions with knot-points at $t_1, \ldots, t_k$.

1. A given basis function is non-zero on 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 sub-interval between successive knots

3. The basis function is continuous and continuous in its first and second derivatives at each knot point. 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 "A practical guide to splines" by Carl De Boor.

Let's see how the competing methods do on a constructed example. Suppose we know the true model is

$$y = \sin^3(2\pi x^3) + \varepsilon, \qquad \varepsilon \sim N(0, (0.1)^2)$$

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 it in Figure 8.3.

```
> 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=18,lty=1,
            main="True Model")
```

We see how an orthogonal polynomial bases of orders 4 and 12 do in fitting this data:

```
> g4 <- lm(y ~ poly(x,4))
> g12 <- lm(y ~ poly(x,12))
> matplot(x,cbind(y,g4$fit,g12$fit),type="pll",ylab="y",pch=18,
                  lty=c(1,2),main="Orthogonal Polynomials")
```

The two fits are shown in the second panel of Figure 8.3. 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.

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 regions of greater curvature. I have used 12 basis functions for comparability to the orthogonal polynomial fit.

```
> library(splines)
> knots <- c(0,0,0,0,0.2,0.4,0.5,0.6,0.7,0.8,0.85,0.9,1,1,1,1)
> bx <- splineDesign(knots,x)
> gs <- lm(y ~ bx)
> matplot(x,bx,type="l",main="B-spline basis functions")
> matplot(x,cbind(y,gs$fit),type="pl",ylab="y",pch=18,lty=1,
  main="Spline fit")
```

The basis functions themselves are shown in the third panel of Figure 8.3 while the fit itself appears in the fourth 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.

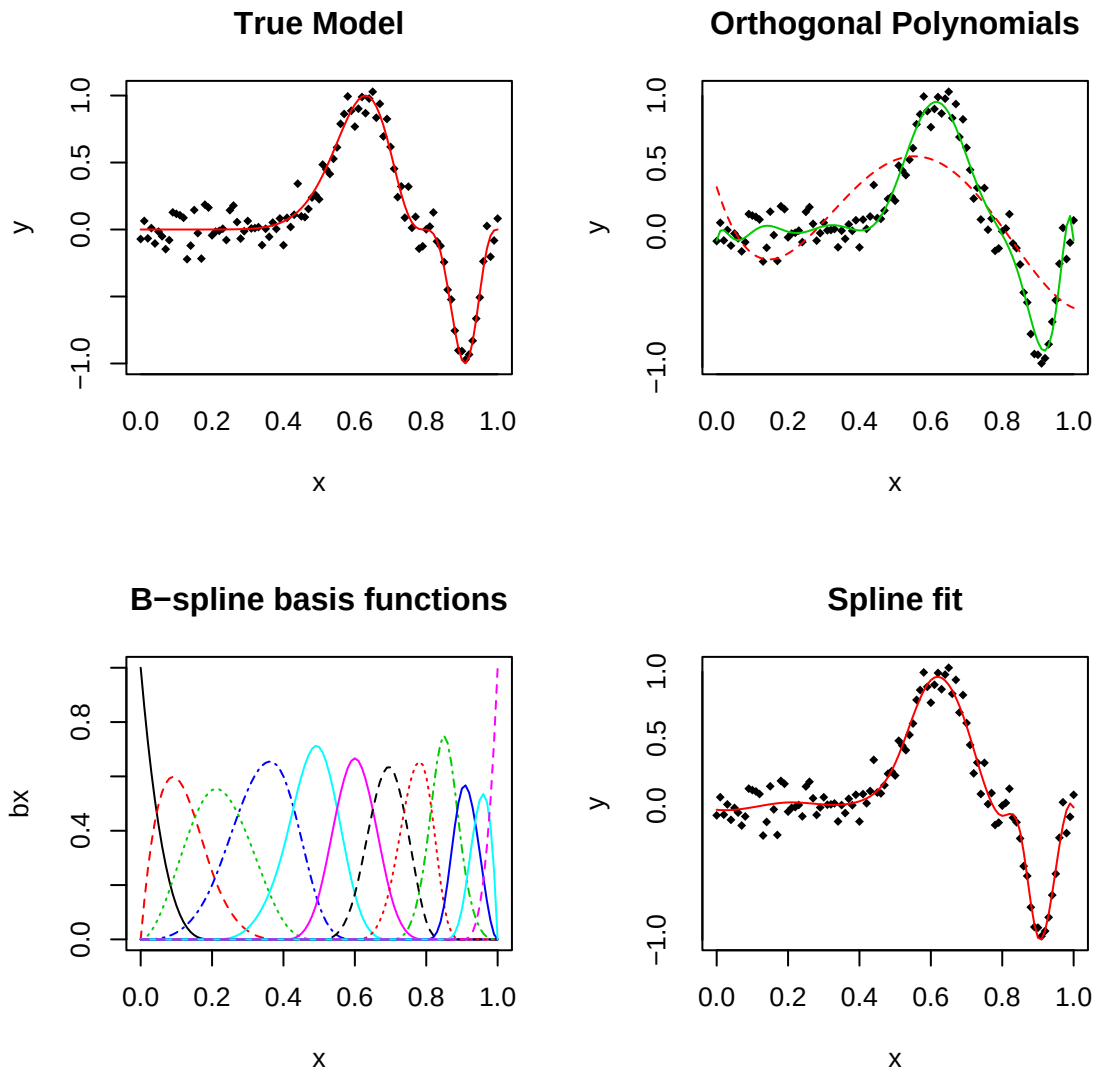

Figure 8.4: Orthogonal Splines compared to B-splines

8.4 Modern Methods

The methods described above are somewhat awkward to apply exhaustively and even then they may miss important structure because of the problem of trying to find good transformations on several variables simultaneously. One recent approach is the additive model:

$$y = \beta_0 + f_1(x_1) + f_2(x_2) + \ldots + f_p(x_p) + \varepsilon$$

where nonparametric regression techniques are used to estimated the f_i's. Alternatively, you could implement this using the regression spline bases for each predictor variable. Other techniques are ACE, AVAS, Regression Trees, MARS and neural networks.

It is important to realize the strengths and weaknesses of regression analysis. For larger data sets with relatively little noise, more recently developed complex models will be able to fit the data better while keeping the number of parameters under control. For smaller data sets or where the noise level is high (as

is typically found in the social sciences), more complex models are not justified and standard regression is most effective. One relative advantage of regression is that the models are easier to interpret in contrast to techniques like neural networks which are usually only good for predictive purposes.

Chapter 9

Scale Changes, Principal Components and Collinearity

9.1 Changes of Scale

Suppose we re-express x_i as $\frac{x_i+a}{b}$. We might want to do this because

1. Predictors of similar magnitude are easier to compare. $\hat{\beta} = 3.51$ is easier to parse than $\hat{\beta} = 0.000000351$.
2. A change of units might aid interpretability.
3. Numerical stability is enhanced when all the predictors are on a similar scale.

Rescaling x_i leaves the t and F tests and $\hat{\sigma}^2$ and R^2 unchanged and $\hat{\beta}_i \to b\hat{\beta}_i$.
Rescaling y in the same way leaves the t and F tests and R^2 unchanged but $\hat{\sigma}$ and $\hat{\beta}$ will rescaled by b.
To demonstrate this, we use same old model:

```
> g <- lm(sr ~ pop15+pop75+dpi+ddpi,savings)
> summary(g)
Coefficients:
             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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

The coefficient for income is rather small - let's measure income in thousands of dollars instead and refit:

```
> g <- lm(sr ~ pop15+pop75+I(dpi/1000)+ddpi,savings)
> summary(g)
Coefficients:
```

```
            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

Residual standard error: 3.8 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

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))
> g <- lm(sr ~ ., scsav)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.0e-16     0.1200 3.3e-15   1.0000
pop15        -0.9420     0.2954   -3.19   0.0026
pop75        -0.4873     0.3122   -1.56   0.1255
dpi          -0.0745     0.2059   -0.36   0.7192
ddpi          0.2624     0.1257    2.09   0.0425

Residual standard error: 0.849 on 45 degrees of freedom
Multiple R-Squared: 0.338,      Adjusted R-squared: 0.28
F-statistic: 5.76 on 4 and 45 degrees of freedom,       p-value: 0.00079
```

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 kind of partial correlation — the values will always be between -1 and 1. It also avoids some numerical problems that can arise when variables are of very different scales. The downside 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 not always be easy to interpret.

9.2 Principal Components

Recall that if the X matrix is orthogonal then testing and interpretation are greatly simplified. One purpose for principal components is to transform the X to orthogonality. For example, consider the case with two predictors depicted in Figure 9.1.

The original predictors, x_1 and x_2, are clearly correlated and so the X-matrix will not be orthogonal. This will complicate the interpretation of the effects of x_1 and x_2 on the response. Suppose we rotate the coordinate axes so that in the new system, the predictors are orthogonal. Furthermore, suppose we make the rotation so that the first axis lies in the direction of the greatest variation in the data, the second in the

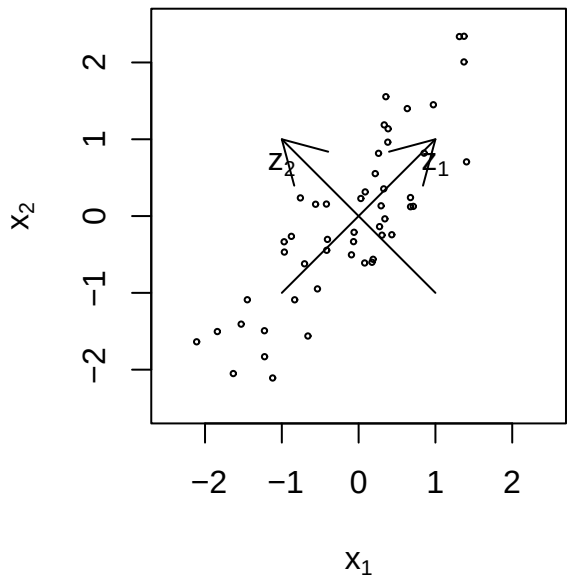

Figure 9.1: Original predictors are x_1 and x_2, principal components are z_1 and z_2

second greatest direction of variation in those dimensions remaining and so on. These rotated directions, z_1 and z_2 in our two predictor example, are simply linear combinations of the original predictors. This is the geometrical description of principal components. We now indicate how these directions may be calculated.

We wish to find a rotation $p \times p$ matrix U such that

$$Z = XU$$

and $Z^T Z = \text{diag}(\lambda_1, \ldots, \lambda_p)$ and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$. Zero eigenvalues indicate non-identifiability. Since

$$Z^T Z = U^T X^T X U$$

the eigenvalues of $X^T X$ are $\lambda_1, \ldots, \lambda_p$ and the eigenvectors of $X^T X$ are the columns of U. The columns of Z are called the principal components and these are orthogonal to each other. λ_i is the variance of Z_i.

Another way of looking at it is to try to find the linear combinations of X which have the maximum variation. We find the u_1 such that var $(u_1^T X)$ is maximized subject to $u_1^T u_1 = 1$. Now find u_2 such that var $(u_2^T X)$ is maximized subject to $u_1^T u_2 = 0$ and $u_2^T u_2 = 1$. We keep finding directions of greatest variation orthogonal to those directions we have already found.

Ideally, only a few eigenvalues will be large so that almost all the variation in X will be representable by those first few principal components.

Principal components can be effective in some situations but

1. The principal components are linear combinations of the predictors — little is gained if these are not interpretable. Generally the predictors have to be measurements of comparable quantities for interpretation to be possible

2. Principal components does not use the response. It's possible that a lesser principal component is actually very important in predicting the response.

3. There are variations which involve not using the intercept in $X^T X$ or using the correlation matrix of the predictors instead of $X^T X$.

We use the Longley data for this example: First we compute the eigendecomposition for $X^T X$:

```
> data(longley)
> x <- as.matrix(longley[,-7])
> e <- eigen(t(x) %*% x)
```

Look at the eigenvalues:

```
> e$values
[1] 6.6653e+07 2.0907e+05 1.0536e+05 1.8040e+04 2.4557e+01 2.0151e+00
```

Look at the relative size - the first is big. Consider the first eigenvector (column) below:

```
> dimnames(e$vectors)[[2]] <- paste("EV",1:6)
> round(e$vec,3)
         EV 1   EV 2   EV 3   EV 4   EV 5   EV 6
GNP def 0.050 -0.070 -0.034  0.043 -0.957 -0.273
GNP     0.191 -0.725 -0.343  0.554  0.075  0.087
Unem    0.157 -0.622  0.564 -0.521  0.007  0.011
Armed   0.128 -0.104 -0.746 -0.645  0.012  0.000
Popn    0.058 -0.038 -0.011  0.036  0.281 -0.956
Year    0.957  0.266  0.078  0.057  0.015  0.053
```

The first eigenvector is dominated by year. Now examining the X-matrix. What are the scales of the variables?

```
> x
     GNP deflator     GNP Unemployed Armed Forces Population Year
1947         83.0 234.289      235.6        159.0    107.608 1947
1948         88.5 259.426      232.5        145.6    108.632 1948
1949         88.2 258.054      368.2        161.6    109.773 1949
1950         89.5 284.599      335.1        165.0    110.929 1950
1951         96.2 328.975      209.9        309.9    112.075 1951
1952         98.1 346.999      193.2        359.4    113.270 1952
1953         99.0 365.385      187.0        354.7    115.094 1953
1954        100.0 363.112      357.8        335.0    116.219 1954
1955        101.2 397.469      290.4        304.8    117.388 1955
1956        104.6 419.180      282.2        285.7    118.734 1956
1957        108.4 442.769      293.6        279.8    120.445 1957
1958        110.8 444.546      468.1        263.7    121.950 1958
1959        112.6 482.704      381.3        255.2    123.366 1959
1960        114.2 502.601      393.1        251.4    125.368 1960
1961        115.7 518.173      480.6        257.2    127.852 1961
1962        116.9 554.894      400.7        282.7    130.081 1962
```

We see that the variables have different scales. It might make more sense to standardize the predictors before trying principal components. This is equivalent to doing principal components on the correlation matrix:

```
> e <- eigen(cor(x))
> e$values
[1] 4.60337710 1.17534050 0.20342537 0.01492826 0.00255207 0.00037671
> dimnames(e$vectors) <- list(c("GNP def","GNP","Unem","Armed","Popn",
  "Year"),paste("EV",1:6))
> round(e$vec,3)
         EV 1   EV 2   EV 3   EV 4   EV 5   EV 6
GNP def 0.462  0.058 -0.149  0.793  0.338  0.135
GNP     0.462  0.053 -0.278 -0.122 -0.150 -0.818
Unem    0.321 -0.596  0.728  0.008  0.009 -0.107
Armed   0.202  0.798  0.562 -0.077  0.024 -0.018
Popn    0.462 -0.046 -0.196 -0.590  0.549  0.312
Year    0.465  0.001 -0.128 -0.052 -0.750  0.450
```

One commonly used method of judging how many principal components are worth considering is the *scree plot* — see Figure 9.2, which is produced by

```
> plot(e$values,type="l",xlab="EV no.")
```

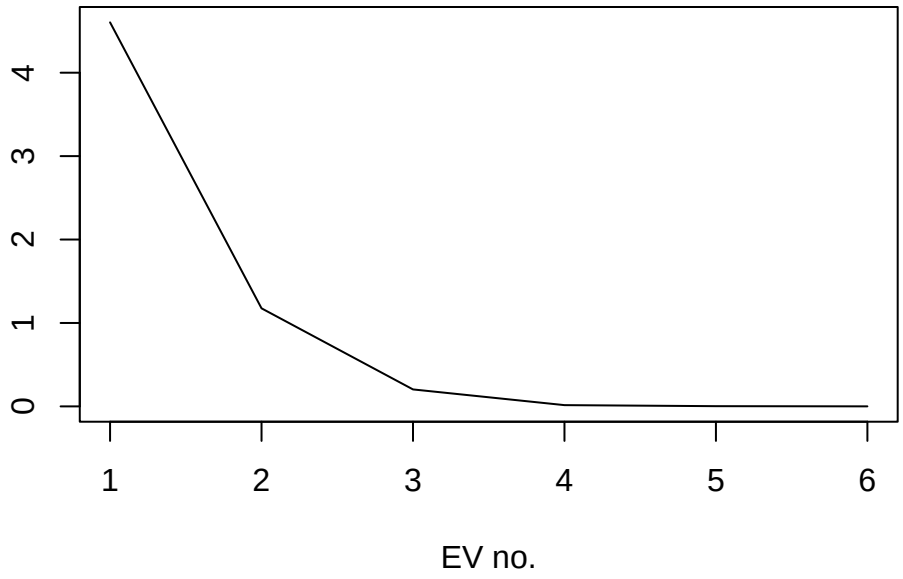

Figure 9.2: Scree plot for the principal components on the correlation of longley predictors

Often, these plots have a noticeable "elbow" — the point at which further eigenvalues are negligible in size compared to the earlier ones. Here the elbow is at 3 telling us that we need only consider 2 principal components.

One advantage of principal components is that it transforms the predictors to an orthogonal basis. To figure out the orthogonalized predictors for this data based on the eigendecomposition for the correlation matrix we must first standardize the data: The functions `scale()` does this:

```
> nx <- scale(x)
```

We can now create the orthogonalized predictors — the $Z = XU$ operation in our description above.

```
> enx <- nx %*% e$vec
```

and fit:

```
> g <- lm(longley$Emp ~ enx)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  65.3170     0.0762  857.03  < 2e-16
enxEV 1       1.5651     0.0367   42.66  1.1e-11
enxEV 2       0.3918     0.0726    5.40  0.00043
enxEV 3      -1.8604     0.1745  -10.66  2.1e-06
enxEV 4       0.3573     0.6442    0.55  0.59267
enxEV 5      -6.1698     1.5581   -3.96  0.00331
enxEV 6       6.9634     4.0555    1.72  0.12011

Residual standard error: 0.305 on 9 degrees of freedom
Multiple R-Squared: 0.995,      Adjusted R-squared: 0.992
F-statistic:  330 on 6 and 9 degrees of freedom,        p-value: 4.98e-10
```

Notice that the p-values of the 4th and 6th eigenvectors are not significant while the the 5th is. Because the directions of the eigenvectors are set successively in the greatest remaining direction of variation in the predictors, it is natural that they be ordered in significance in predicting the response. However, there is no guarantee of this — we see here that the 5th eigenvectors is significant while the fourth is not even though there is about six times greater variation in the fourth direction than the fifth. In this example, it hardly matters since most of the variation is explained by the earlier values, but look out for this effect in other dataset in the first few eigenvalues.

We can take a look at the $(X^TX)^{-1}$ matrix:

```
> round(summary(g)$cov.unscaled,2)
            (Intercept) enxEV 1 enxEV 2 enxEV 3 enxEV 4 enxEV 5 enxEV 6
(Intercept)        0.06    0.00    0.00    0.00    0.00    0.00    0.00
enxEV 1            0.00    0.01    0.00    0.00    0.00    0.00    0.00
enxEV 2            0.00    0.00    0.06    0.00    0.00    0.00    0.00
enxEV 3            0.00    0.00    0.00    0.33    0.00    0.00    0.00
enxEV 4            0.00    0.00    0.00    0.00    4.47    0.00    0.00
enxEV 5            0.00    0.00    0.00    0.00    0.00   26.12    0.00
enxEV 6            0.00    0.00    0.00    0.00    0.00    0.00  176.97
```

Principal components are really only useful if we can interpret the meaning of the new linear combinations. Look back at the first eigenvector - this is roughly a linear combination of all the (standardized variables). Now plot each of the variables as they change over time — see Figure 9.2.

```
> par(mfrow=c(3,2))
> for(i in 1:6) plot(longley[,6],longley[,i],xlab="Year",
  ylab=names(longley)[i])
```

What do you notice? This suggests we identify the first principal component with a time trend effect. The second principal component is roughly a contrast between numbers unemployed and in the armed forces. Let's try fitting a regression with those two components:

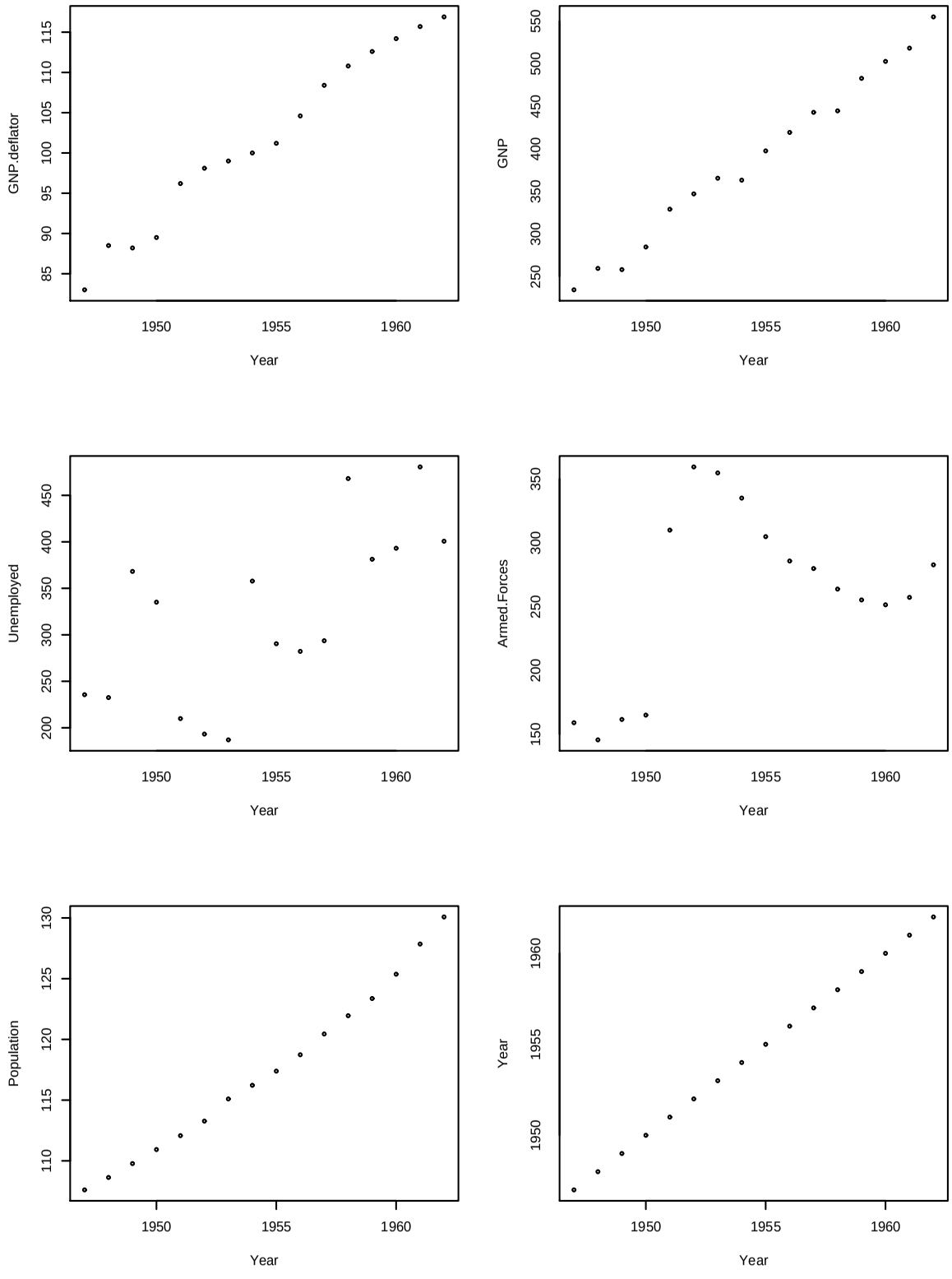

Figure 9.3: The Longley data

```
> summary(lm(Employed ~ Year + I(Unemployed-Armed.Forces),longley))
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  -1.39e+03   7.89e+01   -17.6  1.8e-10
Year                          7.45e-01   4.04e-02    18.5  1.0e-10
I(Unemployed - Armed.Forces) -4.12e-03   1.53e-03    -2.7    0.018

Residual standard error: 0.718 on 13 degrees of freedom
Multiple R-Squared: 0.964,      Adjusted R-squared: 0.958
F-statistic:  173 on 2 and 13 degrees of freedom,       p-value: 4.29e-10
```

This approaches the fit of the full model and is easily interpretable. We could do more work on the other principal components.

This is illustrates a typical use of principal components for regression. Some intuition is required to form new variables as combinations of older ones. If it works, a simplified and interpretable model is obtained, but it doesn't always work out that way.

9.3 Partial Least Squares

Partial Least Squares is a method for relating a set of input variables $X_1, \ldots, X_m$ and outputs $Y_1, \ldots, Y_l$. PLS has some relationship to principal component regression (PCR). PCR regresses the response on the principal components of X while PLS finds the best orthogonal linear combinations of X for predicting Y.

We will consider only univariate PLS — that is to say $l = 1$ so that Y is scalar. This is the typical multiple regression setting. We will attempt to find models of the form

$$\hat{y} = \beta_1 T_1 + \ldots + \beta_p T_p$$

where T_k is a linear combination of the X's. See Figure 9.4

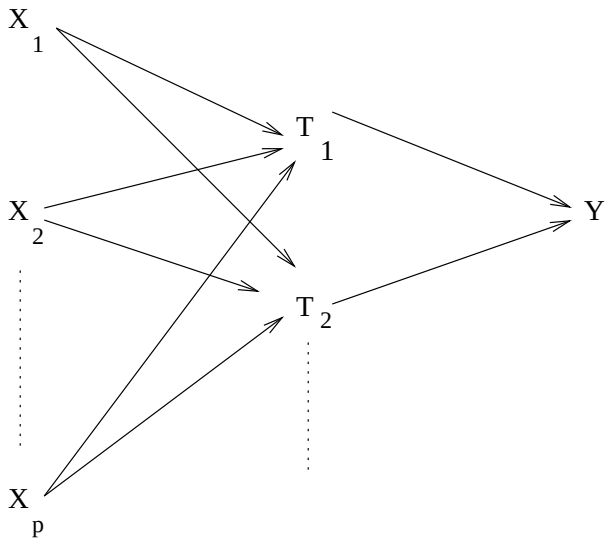

Figure 9.4: Schematic representation of Partial Least Squares

We'll assume that all variables have been centered to have mean 0 — this means that our intercept terms will always be zero. Here is the algorithm for determining the T's.

1. Regress Y on each X_i in turn to get b_{1i}.
2. Form
$$T_1 = \sum_{i=1}^{m} w_{1i} b_{1i} X_{1i}$$
where the weights w_{1i} sum to one.
3. Regress Y on T_1 and each X_i on T_1. The residuals from these regressions have the effect of T_1 removed. Replace Y and each X_i by the residuals of each corresponding regression.
4. Go back to step one updating the index.

There are various choices for the weighting scheme:

1. Set $w_{ij} = 1/m$ giving each predictor equal weight.
2. Set $w_{ij} \propto \text{var} X_j$. This is the most common choice. The variances of the b_{ij} are then inversely proportional to varX_j which does make some sense.

The algorithm ensures that the components T_i are uncorrelated just as the principal components are uncorrelated. This means that $\hat{\beta}_i$ will not change as more components are added to or subtracted from the model.

For this example, we again use the Longley data. We will not need intercept terms in any of the regressions because of the centering.

```
> x <- as.matrix(longley[,-7])
> y <- longley$Emp
> cx <- sweep(x,2,apply(x,2,mean))
> cy <- y -mean(y)
```

Now do the PCR using a more direct method than we used above:

```
> library(mva)
> ex <- princomp(cx)
> g <- lm(cy ~ ex$scores -1)
> summary(g)
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
ex$scoresComp.1  0.025095   0.000603   41.65  1.5e-12
ex$scoresComp.2  0.014038   0.000888   15.82  2.1e-08
ex$scoresComp.3  0.029991   0.002152   13.94  7.1e-08
ex$scoresComp.4 -0.061765   0.058332   -1.06   0.3146
ex$scoresComp.5 -0.489877   0.200645   -2.44   0.0348
ex$scoresComp.6  1.762076   0.443363    3.97   0.0026

Residual standard error: 0.289 on 10 degrees of freedom
Multiple R-Squared: 0.995,      Adjusted R-squared: 0.993
F-statistic:  367 on 6 and 10 degrees of freedom,       p-value: 3.94e-11
```

Are the principal component scores ordered in terms of their importance in predicting the response? Now for later comparison, we have the regression on just first PC.

```
> g <- lm(cy ~ ex$scores[,1] -1)
> summary(g)
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
ex$scores[, 1]   0.0251     0.0034    7.38  2.3e-06

Residual standard error: 1.63 on 15 degrees of freedom
Multiple R-Squared: 0.784,      Adjusted R-squared: 0.77
F-statistic: 54.5 on 1 and 15 degrees of freedom,       p-value: 2.28e-06
```

Now we compute the first component of the partial least squares regression:

```
> b1 <- numeric(6)
> for(i in 1:6){
+ b1[i] <- crossprod(cx[,i],cy)/crossprod(cx[,i],cx[,i])
+ }
> b1
[1] 0.315966 0.034752 0.018885 0.023078 0.484878 0.716512
> ncx <- sweep(cx,2,b1,"*")
> t1 <- apply(ncx,1,mean)
```

Here we have a chosen an equal weighting for the variables. Now see how well this component predicts the response:

```
> gpls1 <- lm(cy ~ t1 -1)
> summary(gpls1)
Coefficients:
   Estimate Std. Error t value Pr(>|t|)
t1   1.3108     0.0959    13.7  7.1e-10

Residual standard error: 0.957 on 15 degrees of freedom
Multiple R-Squared: 0.926,      Adjusted R-squared: 0.921
F-statistic:  187 on 1 and 15 degrees of freedom,       p-value: 7.11e-10
```

Compare this to the result above for one principal component.

An alternative weighting scheme assigns the weights proportional to the variances of the variables:

```
> varx <- apply(cx,2,var)
> vncx <- sweep(ncx,2,varx,"*")
> t1a <- apply(vncx,1,sum)/sum(varx)
> gpls1a <- lm(cy ~ t1a -1)
> summary(gpls1a)
Coefficients:
    Estimate Std. Error t value Pr(>|t|)
t1a    1.605      0.154    10.4  2.8e-08
```

```
Residual standard error: 1.22 on 15 degrees of freedom
Multiple R-Squared: 0.879,      Adjusted R-squared: 0.871
F-statistic:  109 on 1 and 15 degrees of freedom,       p-value: 2.81e-08
```

Compare this to the other output. Now we compute the second component of the PLS. We need to regress out the effect of the first component and then use the same computational method as above.

```
> cx2 <- matrix(0,16,6)
> for(i in 1:6){
+ cx2[,i] <- lm(cx[,i] ~ t1-1)$res
+ }
> cy2 <- lm(cy ~ t1 -1)$res
> b2 <- numeric(6)
> for(i in 1:6){
+ b2[i] <- crossprod(cx2[,i],cy2)/crossprod(cx2[,i],cx2[,i])
+ }
> ncx2 <- sweep(cx2,2,b2,"*")
> t2 <- apply(ncx2,1,mean)
```

Notice the correlation of the components:

```
> cor(t1,t2)
[1] 9.0843e-19
```

Now add t2 to the regression:

```
> gpls2 <- lm(cy ~ t1+t2 -1)
> summary(gpls2)
Coefficients:
   Estimate Std. Error t value Pr(>|t|)
t1   1.3108     0.0749   17.49  6.5e-11
t2  10.9309     3.3658    3.25   0.0058

Residual standard error: 0.748 on 14 degrees of freedom
Multiple R-Squared: 0.958,      Adjusted R-squared: 0.952
F-statistic:  158 on 2 and 14 degrees of freedom,       p-value: 2.44e-10
```

Compare the coefficient of t1 with that above. Now compare this fit to the two component PCR.

```
> g <- lm(cy ~ ex$scores[,1:2] -1)
> summary(g)
Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
ex$scores[, 1:2]Comp.1  0.02510    0.00243   10.34  6.2e-08
ex$scores[, 1:2]Comp.2  0.01404    0.00358    3.93   0.0015

Residual standard error: 1.16 on 14 degrees of freedom
Multiple R-Squared: 0.897,      Adjusted R-squared: 0.883
F-statistic: 61.2 on 2 and 14 degrees of freedom,       p-value: 1.2e-07
```

Which one is superior in explaining y?
Notes:

- The tricky part is choosing how many components are required. Crossvalidation is a possible way of selecting the number of components.
- There are other faster versions of the algorithm described above but these generally provide less insight into the method.
- PLS has been criticized as an algorithm that solves no well-defined modeling problem.
- PLS has the biggest advantage over ordinary least squares and PCR when there are large numbers of variables relative to the number of case. It does not even require that $n \geq p$.

PCR and PLS compared

PCR attempts to find linear combinations of the predictors that explain most of the variation in these predictors using just a few components. The purpose is dimension reduction. Because the principal components can be linear combinations of all the predictors, the number of variables used is not always reduced. Because the principal components are selected using only the X-matrix and not the response, there is no definite guarantee that the PCR will predict the response particularly well although this often happens. If it happens that we can interpret the principal components in a meaningful way, we may achieve a much simpler explanation of the response. Thus PCR is geared more towards explanation than prediction.

In contrast, PLS finds linear combinations of the predictors that best explain the response. It is most effective when ther are large numbers of variables to be considered. If successful, the variablity of prediction is substantially reduced. On the other hand, PLS is virtually useless for explanation purposes.

9.4 Collinearity

If X^TX is singular, i.e. some predictors are linear combinations of others, we have (exact) collinearity and there is no unique least squares estimate of β. If X^TX is close to singular, we have (approximate) collinearity or multicollinearity (some just call it collinearity). This causes serious problems with the estimation of β and associated quantities as well as the interpretation. Collinearity can be detected in several ways:

1. Examination of the correlation matrix of the predictors will reveal large *pairwise* collinearities.
2. A regression of x_i on all other predictors gives R_i^2. Repeat for all predictors. R_i^2 close to one indicates a problem — the offending linear combination may be found.
3. Examine the eigenvalues of X^TX - small eigenvalues indicate a problem. The condition number is defined as
$$\kappa = \sqrt{\frac{\lambda_1}{\lambda_p}}$$
where $\kappa \geq 30$ is considered large. κ is called the condition number. 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.

Collinearity makes some of the parameters hard to estimate. Define
$$S_{x_jx_j} = \sum_i (x_{ij} - \bar{x})^2$$

then

$$\text{var}\,\hat{\beta}_j = \sigma^2 \left(\frac{1}{1-R_j^2} \right) \frac{1}{S_{x_j x_j}}$$

We can see that if x_j doesn't vary much then the variance of $\hat{\beta}_j$ will be large. As an aside, the variance of the first principal component is maximized and so the variance of the corresponding regression coefficient will tend to be small. Another consequence of this equation is that it tells us what designs will minimize the variance of the regression coefficients if we have the ability to place 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, most would put some design points in the middle of the range to allow checking of the fit.

If R_j^2 is close to one then the *variance inflation factor* $\frac{1}{1-R_j^2}$ will be large. For orthogonal designs and principal components, $R_j^2 = 0$, so in these case, we see that the regression coefficient estimation suffers no additional penalty in terms of precision.

Collinearity leads to

1. imprecise estimates of β — the signs of the coefficients may be misleading.
2. t-tests which fail to reveal significant factors
3. missing importance of predictors

The Longley dataset is a good example of collinearity:

```
> g <- lm(Employed ~ ., longley)
> summary(g)
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.48e+03   8.90e+02   -3.91  0.00356
GNP.deflator  1.51e-02   8.49e-02    0.18  0.86314
GNP          -3.58e-02   3.35e-02   -1.07  0.31268
Unemployed   -2.02e-02   4.88e-03   -4.14  0.00254
Armed.Forces -1.03e-02   2.14e-03   -4.82  0.00094
Population   -5.11e-02   2.26e-01   -0.23  0.82621
Year          1.83e+00   4.55e-01    4.02  0.00304

Residual standard error: 0.305 on 9 degrees of freedom
Multiple R-Squared: 0.995,      Adjusted R-squared: 0.992
F-statistic:  330 on 6 and 9 degrees of freedom,        p-value: 4.98e-10
```

Recall that the response is number employed. Three of the predictors have large p-values but all are variables that might be expected to affect the response. Why aren't they significant? Check the correlation matrix first (rounding to 3 digits for convenience)

```
> round(cor(longley[,-7]),3)
             GNP deflator   GNP Unemployed Armed Forces Population  Year
GNP deflator        1.000 0.992      0.621        0.465      0.979 0.991
```

```
         GNP        0.992 1.000      0.604        0.446      0.991 0.995
  Unemployed        0.621 0.604      1.000       -0.177      0.687 0.668
Armed Forces        0.465 0.446     -0.177        1.000      0.364 0.417
  Population        0.979 0.991      0.687        0.364      1.000 0.994
        Year        0.991 0.995      0.668        0.417      0.994 1.000
```

There are several large pairwise correlations. Now we check the eigendecomposition:

```
> x <- as.matrix(longley[,-7])
> e <- eigen(t(x) %*% x)
> e$val
[1] 6.6653e+07 2.0907e+05 1.0536e+05 1.8040e+04 2.4557e+01 2.0151e+00
> sqrt(e$val[1]/e$val)
[1]    1.000   17.855   25.153   60.785 1647.478 5751.216
```

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 out the variance inflation factors. For the first variable this is

```
> summary(lm(x[,1] ~ x[,-1]))$r.squared
[1] 0.99262
> 1/(1-0.99262)
[1] 135.5
```

which is large - the VIF for orthogonal predictors is 1. Now we compute all the VIF's in one go:

```
> vif(x)
[1]  135.5324 1788.5135   33.6189    3.5889  399.1510  758.9806
```

There's definitely a lot of variance inflation! For example, we can interpret $\sqrt{(}1788) \approx 42$ as telling us that the standard error for GNP is 42 times larger than it would have been without collinearity. How can we get rid of this problem? One way is to throw out some of the variables. Examine the full correlation matrix above. Notice that variables 3 and 4 do not have extremely large pairwise correlations with the other variables so we should keep them and focus on the others for candidates for removal:

```
> cor(x[,-c(3,4)])
             GNP.deflator     GNP Population    Year
GNP.deflator      1.00000 0.99159    0.97916 0.99115
GNP               0.99159 1.00000    0.99109 0.99527
Population        0.97916 0.99109    1.00000 0.99395
Year              0.99115 0.99527    0.99395 1.00000
```

These four variables are strongly correlated with each other - any one of them could do the job of representing the other. We pick year arbitrarily:

```
> summary(lm(Employed ~ Armed.Forces + Unemployed + Year,longley))
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.80e+03   6.86e+01  -26.18  5.9e-12
```

```
Armed.Forces -7.72e-03   1.84e-03   -4.20   0.0012
Unemployed   -1.47e-02   1.67e-03   -8.79  1.4e-06
Year          9.56e-01   3.55e-02   26.92  4.2e-12

Residual standard error: 0.332 on 12 degrees of freedom
Multiple R-Squared: 0.993,      Adjusted R-squared: 0.991
F-statistic:  555 on 3 and 12 degrees of freedom,       p-value: 3.92e-13
```

Comparing this with the original fit, we see that the fit is very similar but only three rather than six predictors are used.

One final point - extreme collinearity can cause problems in computing the estimates - look what happens when we use the direct formula for $\hat{\beta}$.

```
> x <- as.matrix(cbind(1,longley[,-7]))
> solve(t(x) %*% x) %*% t(x) %*% longley[,7]
Error: singular matrix 'a' in solve
```

R , like most statistical packages, uses a more numerically stable method for computing the estimates in `lm()`. Something more like this:

```
> solve(t(x) %*% x , t(x) %*% longley$Emp, tol = 1e-12)
             [,1]
[1,] -3.4822e+03
[2,]  1.5061e-02
[3,] -3.5818e-02
[4,] -2.0202e-02
[5,] -1.0332e-02
[6,] -5.1110e-02
[7,]  1.8291e+00
```

Collinearity can be interpreted geometrically. Imagine a table — as two diagonally opposite legs are moved closer together, the table becomes increasing unstable.

The effect of collinearity on prediction depends on where the prediction is to be made. The greater the distance from the observed data, the more unstable the prediction. Distance needs to be considered in a Mahalanobis sense rather than Euclidean.

One cure for collinearity is amputation — too many variables are trying to do the same job of explaining the response. When several variables which are highly correlated are each associated with the response, we have to take care that we don't conclude that the variables we drop have nothing to do with the response.

9.5 Ridge Regression

Ridge regression makes the assumption that the regression coefficients (after normalization) are not likely to be very large. It is appropriate for use when the design 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 estimates of β are then given by

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y$$

The ridge constant λ is usually selected from the range $[0,1]$.

The use of ridge regression can be motivated in two ways. Suppose we take a Bayesian point of view and put a prior (multivariate normal) distribution on β that expresses the belief that smaller values of β are more likely than larger ones. Large values of λ correspond to a belief that the β are really quite small whereas smaller values of λ correspond to a more relaxed belief about β. This is illustrated in Figure 9.5.

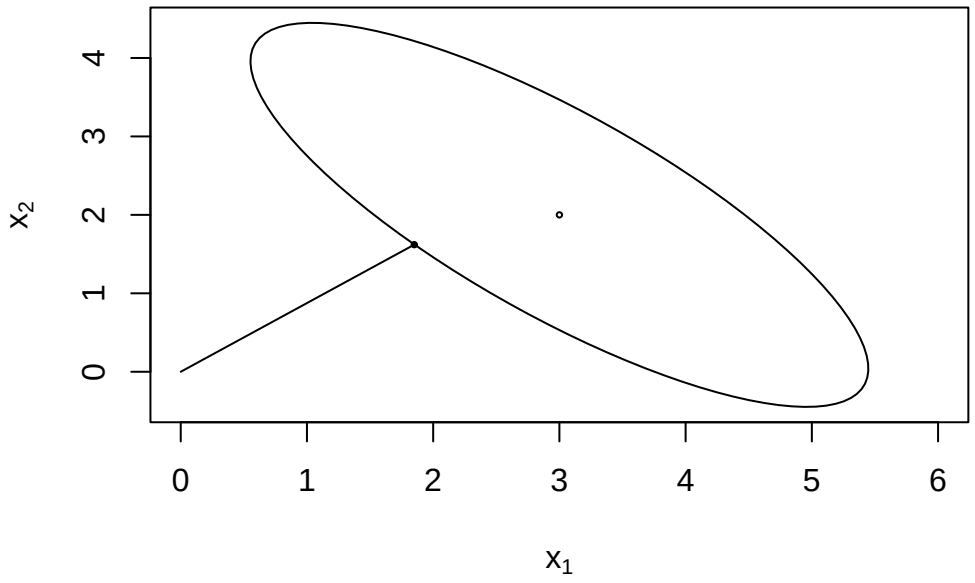

Figure 9.5: Ridge regression illustrated. The least squares estimate is at the center of the ellipse while the ridge regression is the point on the ellipse closest to the origin. The ellipse is a contour of equal density of the posterior probability, which in this case will be comparable to a confidence ellipse. λ controls the size of the ellipse - the larger λ is, the larger the ellipse will be

Another way of looking at is to suppose we place to some upper bound on $\beta^T\beta$ and then compute the least squares estimate of β subject to this restriction. Use of Lagrange multipliers leads to ridge regression. The choice of λ corresponds to the choice of upper bound in this formulation.

λ may be chosen by automatic methods but it is probably safest 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 Longley data. $\lambda = 0$ corresponds to least squares while we see that as $\lambda \to \infty$, $\hat{\beta} \to 0$.

```
> library(MASS)
> gr <- lm.ridge(Employed ~ .,longley,lambda = seq(0,0.1,0.001))
> matplot(gr$lambda,t(gr$coef),type="l",xlab=expression(lambda),
                  ylab=expression(hat(beta)))
> abline(h=0,lwd=2)
```

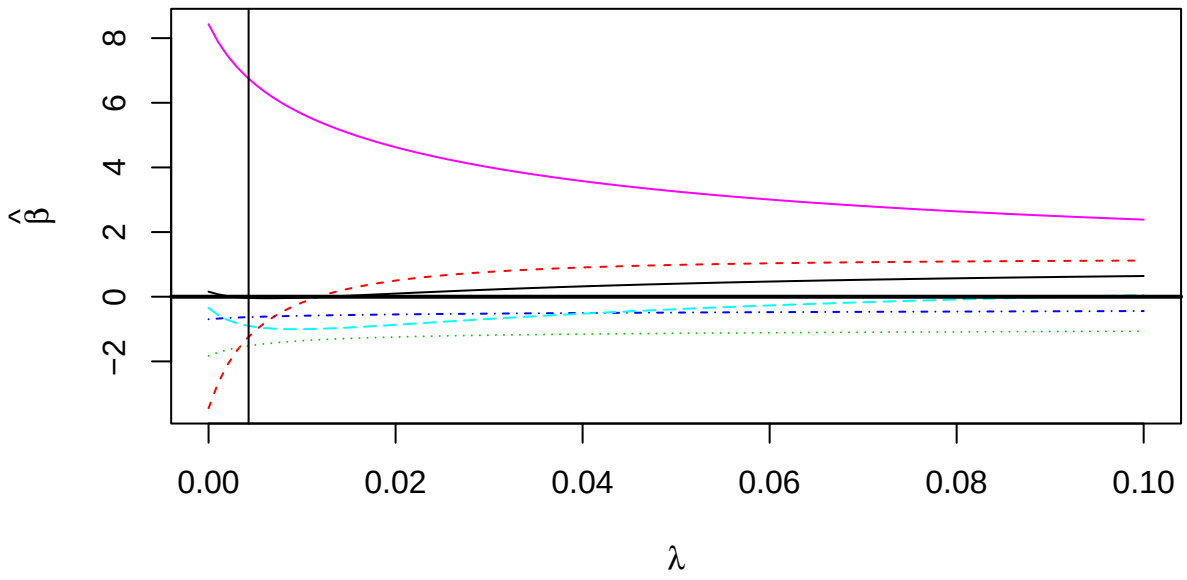

Figure 9.6: Ridge trace plot for the Longley data. The vertical line is the Hoerl-Kennard choice of λ. The topmost curve represent the coefficient for year. The dashed line that starts well below zero but ends above is for GNP.

The ridge trace plot is shown in Figure 9.5.
Various automatic selections for λ are available

```
> select(gr)
modified HKB estimator is 0.0042754
modified L-W estimator is 0.032295
smallest value of GCV  at 0.003
> abline(v=0.00428)
```

The Hoerl-Kennard (the originators of ridge regression) choice of λ has been shown on the plot but I would prefer a larger value of 0.03. For this choice of λ, the $\hat{\beta}$'s are

```
> gr$coef[,gr$lam == 0.03]
GNP.deflator       GNP   Unemployed Armed.Forces  Population     Year
     0.22005   0.76936     -1.18941     -0.52234    -0.68618  4.00643
```

in contrast to the least squares estimates of

```
> gr$coef[,1]
GNP.deflator       GNP  Unemployed Armed.Forces  Population     Year
     0.15738  -3.44719    -1.82789     -0.69621    -0.34420  8.43197
```

Note that these values are based on centered and scaled predictors which explains the difference from previous fits. Consider the change in the coefficient for GNP. For the least squares fit, the effect of GNP is negative on the response - number of people employed. This is counter-intuitive since we'd expect the effect to be positive. The ridge estimate is positive which is more in line with what we'd expect.

Ridge regression estimates of coefficients are biased. Bias is undesirable but there are other considerations. The mean squared error 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 mean-squared error of an estimate can be represented as the square of the bias plus the variance. Sometimes a large reduction in the variance may obtained at the price of an increase in the bias. If the MSE is 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.

Chapter 10

Variable Selection

Variable selection is intended to select the "best" subset of predictors. But why bother?

1. We want to explain the data in the simplest way — redundant predictors should be removed. 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 is best.

2. Unnecessary predictors will add noise to the estimation of other quantities that we are interested in. Degrees of freedom will be wasted.

3. Collinearity is caused by having too many variables trying to do the same job.

4. Cost: if the model is to be used for prediction, we can save time and/or money by not measuring redundant predictors.

Prior to variable selection:

1. Identify outliers and influential points - maybe exclude them at least temporarily.

2. Add in any transformations of the variables that seem appropriate.

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 be removed from the model before higher order terms in the same variable. There two common situations where this situation arises:

- Polynomials models. Consider the model

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

  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 removed the x term, our reduced model would then become

$$y = \beta_0 + \beta_2 x^2 + \varepsilon$$

but suppose we then made a scale change $x \to x+a$, then the model would become

$$y = \beta_0 + \beta_2 a^2 + 2\beta_2 ax + \beta_2 x^2 + \varepsilon.$$

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 good. 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 $x = 0$. Often 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.

- Models with interactions. Consider the second order response surface model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2$$

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 linear one. Just removing the x_1x_2 term would correspond to a surface that is aligned with the coordinate axes. This is hard 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 Stepwise Procedures

Backward Elimination

This 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.

1. Start with all the predictors in the model
2. Remove the predictor with highest p-value greater than α_{crit}
3. Refit the model and goto 2
4. Stop when all p-values are less than α_{crit}.

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 15-20% cut-off may work best, although methods designed more directly for optimal prediction should be preferred.

10.2.1 Forward Selection

This just reverses the backward method.

1. Start with no variables in the model.
2. For all predictors not in the model, check their p-value if they are added to the model. Choose the one with lowest p-value less than α_{crit}.
3. Continue until no new predictors can be added.

10.2.2 Stepwise Regression

This 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.

Stepwise procedures are relatively cheap computationally but they do have some drawbacks.

1. Because of the "one-at-a-time" nature of adding/dropping variables, it's 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 these variables are unrelated to the response, it's 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.

We illustrate the variable selection methods on some data on the 50 states - the variables are population estimate as of July 1, 1975; per capita income (1974); illiteracy (1970, percent of population); life expectancy in years (1969-71); murder and non-negligent manslaughter rate per 100,000 population (1976); percent high-school graduates (1970); mean number of days with min temperature < 32 degrees (1931-1960) in capital or large city; and land area in square miles. The data was collected from US Bureau of the Census. We will take life expectancy as the response and the remaining variables as predictors - a fix is necessary to remove spaces in some of the variable names.

```
> data(state)
> statedata <- data.frame(state.x77,row.names=state.abb,check.names=T)
> g <- lm(Life.Exp ~ ., data=statedata)
> summary(g)
Coefficients:
             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
```

```
Residual standard error: 0.745 on 42 degrees of freedom
Multiple R-Squared: 0.736,      Adjusted R-squared: 0.692
F-statistic: 16.7 on 7 and 42 degrees of freedom,       p-value: 2.53e-10
```

Which predictors should be included - can you tell from the p-values? Looking at the coefficients, can you see what operation would be helpful? Does the murder rate decrease life expectancy - that's obvious a priori, but how should these results be interpreted?

We illustrate the backward method - at each stage we remove the predictor with the largest p-value over 0.05:

```
> g <- update(g, . ~ . - Area)
> summary(g)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.10e+01   1.39e+00   51.17  < 2e-16
Population   5.19e-05   2.88e-05    1.80    0.079
Income      -2.44e-05   2.34e-04   -0.10    0.917
Illiteracy   2.85e-02   3.42e-01    0.08    0.934
Murder      -3.02e-01   4.33e-02   -6.96  1.5e-08
HS.Grad      4.85e-02   2.07e-02    2.35    0.024
Frost       -5.78e-03   2.97e-03   -1.94    0.058

> g <- update(g, . ~ . - Illiteracy)
> summary(g)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.11e+01   1.03e+00   69.07  < 2e-16
Population   5.11e-05   2.71e-05    1.89    0.066
Income      -2.48e-05   2.32e-04   -0.11    0.915
Murder      -3.00e-01   3.70e-02   -8.10  2.9e-10
HS.Grad      4.78e-02   1.86e-02    2.57    0.014
Frost       -5.91e-03   2.47e-03   -2.39    0.021

> g <- update(g, . ~ . - Income)
> summary(g)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.10e+01   9.53e-01   74.54  < 2e-16
Population   5.01e-05   2.51e-05    2.00   0.0520
Murder      -3.00e-01   3.66e-02   -8.20  1.8e-10
HS.Grad      4.66e-02   1.48e-02    3.14   0.0030
Frost       -5.94e-03   2.42e-03   -2.46   0.0180

> g <- update(g, . ~ . - Population)
> summary(g)
```

```
Coefficients:
             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   0.0020
Frost       -0.00691    0.00245   -2.82   0.0070

Residual standard error: 0.743 on 46 degrees of freedom
Multiple R-Squared: 0.713,      Adjusted R-squared: 0.694
F-statistic:   38 on 3 and 46 degrees of freedom,       p-value: 1.63e-12
```

The final removal of the Population variable is a close call. We may want to consider including this variable if interpretation is aided. 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.

10.3 Criterion-based procedures

If there are p potential predictors, then there are 2^p possible models. We fit all these models and choose the best one according to some criterion. Clever algorithms such as the "branch-and-bound" method can avoid actually fitting all the models — only likely candidates are evaluated. Some criteria are

1. The Akaike Information Criterion (AIC) and the Bayes Information Criterion (BIC) are some other commonly used criteria. In general,

   $$AIC = -2\texttt{log} - \texttt{likelihood} + 2p$$

   while

   $$BIC = -2\texttt{log} - \texttt{likelihood} + p\log n$$

   For linear regression models, the -2log-likelihood (known as the *deviance* is $n\log(RSS/n)$. We want to minimize AIC or BIC. Larger models will fit better and so have smaller RSS but use more parameters. Thus the best choice of model will balance fit with model size. BIC penalizes larger models more heavily and so will tend to prefer smaller models in comparison to AIC. AIC and BIC can be used as selection criteria for other types of model too.

   We can apply the AIC (and optionally the BIC) to the state data. 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 with the advantage that no dubious p-values are used.

```
> g <- lm(Life.Exp ~ ., data=statedata)
> step(g)
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
- 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
   7.10e+01     5.01e-05    -3.00e-01     4.66e-02    -5.94e-03
```

The sequence of variable removal is the same as with backward elimination. The only difference is the the Population variable is retained.

2. Adjusted R^2 — called R^2_a. 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 so R^2 by itself is not a good criterion because it would always choose the largest possible model.

$$R^2_a = 1 - \frac{RSS/(n-p)}{TSS/(n-1)} = 1 - \left(\frac{n-1}{n-p}\right)(1-R^2) = 1 - \frac{\hat{\sigma}^2_{model}}{\hat{\sigma}^2_{null}}$$

Adding a predictor will only increase R^2_a if it has some value. Do you see the connection to $\hat{\sigma}^2$? Minimizing the standard error for prediction means minimizing $\hat{\sigma}^2$ which in term means maximizing R^2_a.

3. Predicted Residual Sum of Squares (PRESS) is defined as $\sum_i \hat{\varepsilon}^2_{(i)}$ where the $\hat{\varepsilon}_{(i)}$ are the residuals calculated without using case i in the fit. The model with the lowest PRESS criterion is then selected. This tends to pick larger models (which may be desirable if prediction is the objective).

4. Mallow's C_p Statistic. A good model should predict well so average MSE of prediction might be a good criterion:

$$\frac{1}{\sigma^2}\sum_i E(\hat{y}_i - Ey_i)^2$$

which can be estimated by the C_p statistic

$$C_p = \frac{\text{RSS}_p}{\hat{\sigma}^2} + 2p - n$$

where $\hat{\sigma}^2$ is from the model with all predictors and RSS_p indicates the RSS from a model with p parameters.

(a) C_p is easy to compute

(b) It is closely related to R_a^2 and the AIC.

(c) For the full model $C_p = p$ exactly.

(d) If a p predictor model fits then $E(\text{RSS}_p) = (n-p)\sigma^2$ 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.

Now we try the C_p and R_a^2 methods for the selection of variables in the State dataset. The default for the `leaps()` function is the Mallow's Cp criterion:

```
> library(leaps)
> x <- model.matrix(g)[,-1]
> y <- statedata$Life
> g <- leaps(x,y)
> Cpplot(g)
```

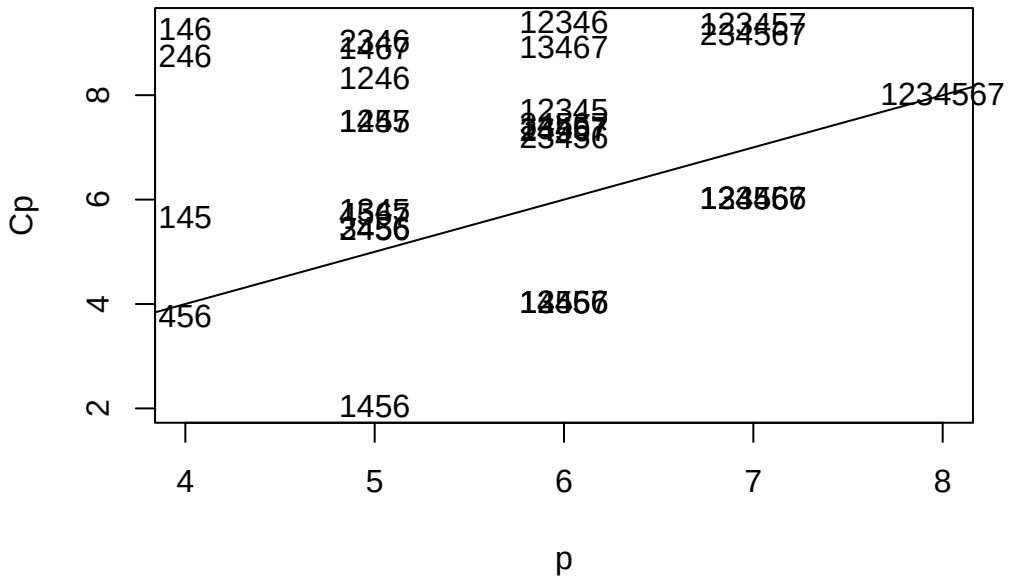

Figure 10.1: The Cp plot for the State data

The models are denoted by indices for the predictors. The competition is between the "456" model i.e. the Frost, HS graduation and Murder model 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. Smaller models with 1 or 2 predictors are not shown on this plot because their C_p plots are so large.

Now let's see which model the adjusted R^2 criterion selects.

```
> adjr <- leaps(x,y,method="adjr2")
> maxadjr(adjr,8)
  1456  12456  13456  14567 123456 134567 124567    456
 0.713  0.706  0.706  0.706  0.699  0.699  0.699  0.694
```

We see that the Population, Frost, HS graduation and Murder model has the largest R^2_a. The best three predictor model is in eighth place but the intervening models are not attractive since they use more predictors than the best model.

Variable selection methods are sensitive to outliers and influential points. Let's check for high leverage points:

```
> h <- hat(x)
> names(h) <- state.abb
> rev(sort(h))
      AK       CA       HI       NV       NM       TX       NY       WA
0.809522 0.408857 0.378762 0.365246 0.324722 0.284164 0.256950 0.222682
```

Which state sticks out? Let's try excluding it (Alaska is the second state in the data).

```
> l <- leaps(x[-2,],y[-2],method="adjr2")
> maxadjr(l)
 12456   1456 123456
 0.710  0.709  0.707
```

We see that area now makes it into the model. Transforming the predictors can also have an effect: Take a look at the variables:

```
> par(mfrow=c(3,3))
> for(i in 1:8) boxplot(state.x77[,i],main=dimnames(state.x77)[[2]][i])
```

In Figure 10.3, we see that Population, Illiteracy and Area are skewed - we try transforming them:

```
> nx <- cbind(log(x[,1]),x[,2],log(x[,3]),x[,4:6],log(x[,7]))
```

And now replot:

```
> par(mfrow=c(3,3))
> apply(nx,2,boxplot)
```

which shows the appropriately transformed data.

Now try the adjusted R2 method again.

```
> a <- leaps(nx,y,method="adjr2")
> maxadjr(a)
 1456 12456 13456
0.717 0.714 0.712
```

This changes the "best" model again to log(Population), Frost, HS graduation and Murder.

The adjusted R^2 is the highest models we have seen so far.

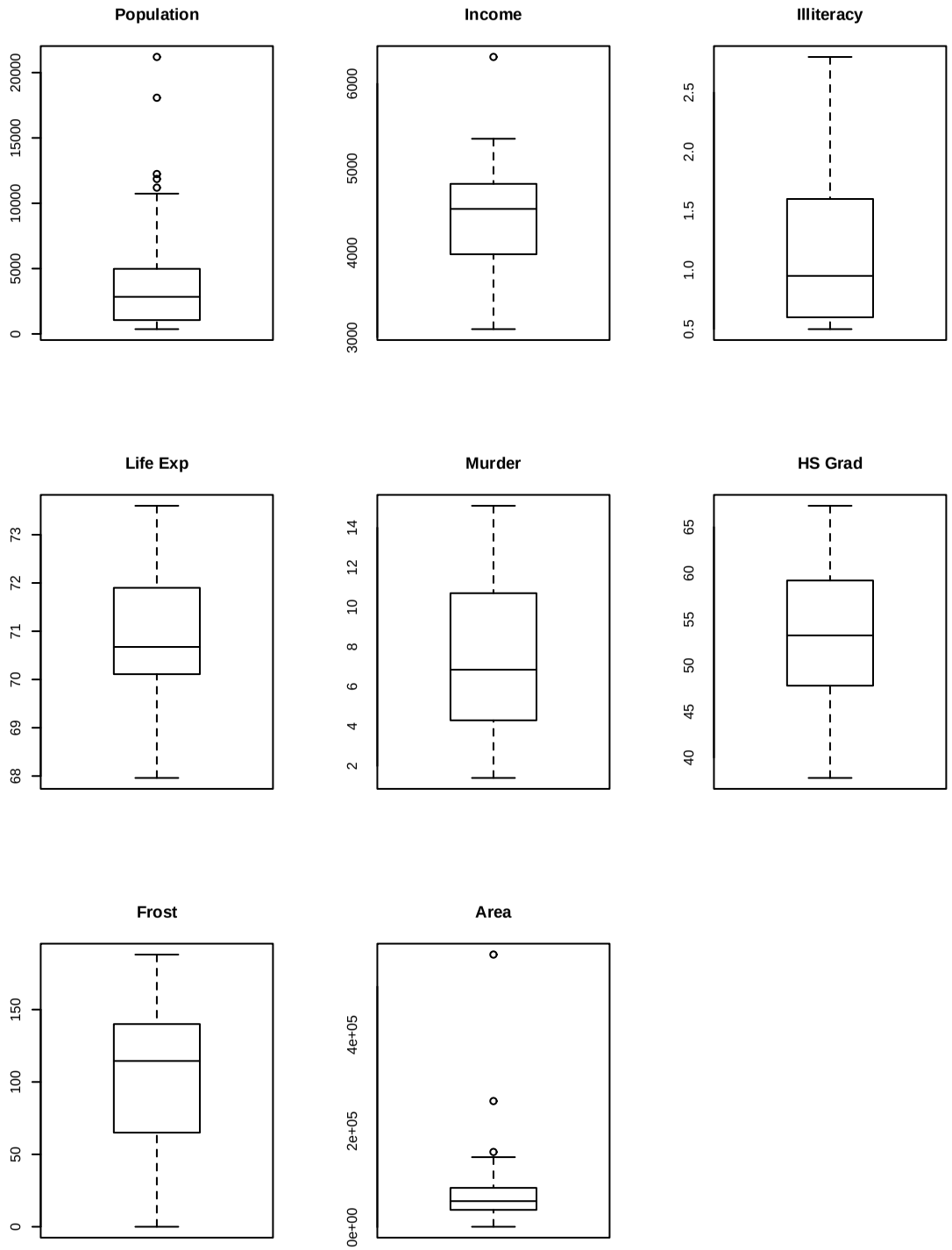

Figure 10.2: Boxplots of the State data

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.

Stepwise methods use a restricted search through the space of potential models and use a dubious hypothesis testing based method for choosing between models. Criterion-based methods typically involve a wider search and compare models in a preferable manner. For this reason, I 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 equally as good but lead to quite different conclusions then it is clear that the data cannot answer the question of interest unambiguously. Be alert to the danger that a model contradictory to the tentative conclusions might be out there.

Chapter 11

Statistical Strategy and Model Uncertainty

11.1 Strategy

Thus far we have learnt various tactics

1. *Diagnostics:* Checking of assumptions: constant variance, linearity, normality, outliers, influential points, serial correlation and collinearity.

2. *Transformation:* Transforming the response — Box-Cox, transforming the predictors — tests and polynomial regression.

3. *Variable selection:* Stepwise and criterion based methods

What order should these be done in? Should procedures be repeated at later stages? When should we stop?

I would recommend *Diagnostics* $\rightarrow$ *Transformation* $\rightarrow$ *Variable Selection* $\rightarrow$ *Diagnostics* as a rudimentary strategy. However, regression analysis is a search for structure in data and there are no hard-and-fast rules about how it should be done. Regression analysis requires some skill. You must be alert to unexpected structure in the data. Thus far, no one has implemented a computer program for conducting a complete analysis. Because of the difficulties in automating the assessment of regression graphics in an intelligent manner, I do not expect that this will accomplished soon. The human analyst has the ability to assess plots in light of contextual information about the data.

There is a danger of doing too much analysis. The more transformations and permutations of leaving out influential points you do, the better fitting model you will find. Torture the data long enough, and sooner or later it will confess. Remember that fitting the data well is no guarantee of good predictive performance or that the model is a good representation of the underlying population. So

1. Avoid complex models for small datasets.

2. Try to obtain new data to validate your proposed model. Some people set aside some of their existing data for this purpose.

3. Use past experience with similar data to guide the choice of model.

Data analysis is not an automatic process. Analysts have personal preferences in their choices of methodology, use software with varying capabilities and will interpret the same graphical display differently. In comparing the competing analyses of two statisticians, it may sometimes be possible to determine that one

analysis is clearly superior. However, in most cases, particularly when the analysts are experienced and professionally trained, a universally acceptable judgment of superiority will not be possible.

The same data may support different models. Conclusions drawn from the models may differ quantitatively and qualitatively. However, except for those well-known datasets that circulate endlessly through textbooks and research articles, most data is only analyzed once. The analyst may be unaware that a second independent look at the data may result in quite different conclusions. We call this problem *model multiplicity*. In the next section, we describe an experiment illustrating the depth of this problem.

11.2 Experiment

In Fall 1996, I taught a semester length masters level course in applied regression analysis to 28 students. Towards the end of the semester, I decided to set an assignment to test the students ability in building a regression model for the purposes of prediction. I generated regression data with a response y and five uncorrelated predictors and $n = 50$ from a model known only to me which was:

$$1/y = x_1 + 0.57x_1^2 + 4x_1x_2 + 2.1\exp(x_4) + \varepsilon$$

where $x_1 \sim U(0,1)$,$x_2 \sim N(0,1)$,$1/x_3 \sim U(0,1)$,$x_4 \sim N(1,1)$, $x_5 \sim U(1,3)$ and $\varepsilon \sim N(0,1)$.

I asked students to predict the mean response at 10 values of the predictors which I specified. I also asked them to provide a standard error for each of their predictions. The students understood and were reminded of the distinction between the standard error for the mean response and for a future observed value. The students were told that their score on the assignment would depend only on the closeness of their predicted values and the true values and on how closely their standard errors reflected the difference between these two quantities. Students were told to work independently.

For a given student's input, let p_i be their prediction, t_i be the true value and s_i be the standard error where $i = 1,\ldots 10$. To assess their prediction accuracy, I used

$$\sum_{i=1}^{10}\left(\frac{p_i - t_i}{t_i}\right)^2$$

whereas to measure the "honesty" of their standard errors, I used

$$\frac{1}{10}\sum_{i=1}^{10}\left|\frac{p_i - t_i}{s_i}\right|.$$

We'd expect the predicted value to differ from the true value by typically about one standard error if the latter has been correctly estimated. Therefore, the measure of standard error honesty should be around one.

1.12	1.20	1.46	1.46	1.54	1.62	1.69
1.69	1.79	3.14	4.03	4.61	5.04	5.06
5.13	5.60	5.76	5.76	5.94	6.25	6.53
6.53	6.69	10.20	34.45	65.53	674.98	37285.95

Table 11.1: Prediction accuracy

The prediction accuracy scores for the 28 students are shown in Table 11.1. We see that one student did very poorly. An examination of their model and some conversation revealed that they neglected to backtransform their predictions to the original scale when using a model with a transform on the response.

Three pairs of scores are identical in the table but an examination of the models used and more digits revealed that only one pair was due to the students using the same model. This pair of students were known associates. Thus 27 different models were found by 28 students.

The scores for honesty of standard errors are shown in Table 11.2. The order in which scores are shown correspond to that given in Table 11.1.

0.75	7.87	6.71	0.59	4.77	8.20	11.74
10.70	1.04	17.10	3.23	14.10	84.86	15.52
80.63	17.61	14.02	14.02	13.35	16.77	12.15
12.15	12.03	68.89	101.36	18.12	2.24	40.08

Table 11.2: Honesty of standard errors - order of scores corresponds to that in Table 11.1.

We see that the students' standard errors were typically around an order of magnitude smaller than they should have been.

11.3 Discussion

Why was there so much model multiplicity? The students were all in the same class and used the same software but almost everyone chose a different model. The course covered many of the commonly understood techniques for variable selection, transformation and diagnostics including outlier and influential point detection. The students were confronted with the problem of selecting the order in which to apply these methods and choosing from several competing methods for a given purpose.

The reason the models were so different was that students applied the various methods in different orders. Some did variable selection before transformation and others the reverse. Some repeated a method after the model was changed and others did not. I went over the strategies that several of the students used and could not find anything clearly wrong with what they had done. One student made a mistake in computing their predicted values but there was nothing obviously wrong in the remainder. The performance on this assignment did not show any relationship with that in the exams.

The implications for statistical practice are profound. Often a dataset is analyzed by a single analyst who comes up with a single model. Predictions and inferences are based on this single model. The analyst may be unaware that the data support quite different models which may lead to very different conclusions. Clearly one won't always have a stable of 28 independent analysts to search for alternatives, but it does point to the value of a second or third independent analysis. It may also be possible to automate the components of the analysis to some extent as in Faraway (1994) to see whether changes in the order of analysis might result in a different model.

Another issue is raised by the standard error results. Often we use the data to help determine the model. Once a model is built or selected, inferences and predictions may be made. Usually inferences are based on the assumption that the selected model was fixed in advance and so only reflect uncertainty concerning the parameters of that model. Students took that approach here. Because the uncertainty concerning the model itself is not allowed for, these inferences tend to be overly optimistic leading to unrealistically small standard errors. Methods for realistic inference when the data is used to select the model have come under the heading of *Model Uncertainty* — see Chatfield (1995) for a review. The effects of model uncertainty often overshadow the parametric uncertainty and the standard errors need to be inflated to reflect this. Faraway (1992) developed a bootstrap approach to compute these standard errors while Draper (1995) is an example of a Bayesian approach. These methods are a step in the right direction in that they reflect the uncertainty in

model selection. Nevertheless, they do not address the problem of model multiplicity since they proscribe a particular method of analysis that does not allow for differences between human analysts.

Sometimes the data speak with a clear and unanimous voice — the conclusions are incontestable. Other times, differing conclusions may be drawn depending on the model chosen. We should acknowledge the possibility of alternative conflicting models and seek them out.

Chapter 12

Chicago Insurance Redlining - a complete example

In a study of insurance availability in Chicago, the U.S. Commission on Civil Rights attempted to examine charges by several community organizations that insurance companies were redlining their neighborhoods, i.e. canceling policies or refusing to insure or renew. First the Illinois Department of Insurance provided the number of cancellations, non-renewals, new policies, and renewals of homeowners and residential fire insurance policies by ZIP code for the months of December 1977 through February 1978. The companies that provided this information account for more than 70% of the homeowners insurance policies written in the City of Chicago. The department also supplied the number of FAIR plan policies written an renewed in Chicago by zip code for the months of December 1977 through May 1978. Since most FAIR plan policyholders secure such coverage only after they have been rejected by the voluntary market, rather than as a result of a preference for that type of insurance, the distribution of FAIR plan policies is another measure of insurance availability in the voluntary market.

Secondly, the Chicago Police Department provided crime data, by beat, on all thefts for the year 1975. Most Insurance companies claim to base their underwriting activities on loss data from the preceding years, i.e. a 2-3 year lag seems reasonable for analysis purposes. the Chicago Fire Department provided similar data on fires occurring during 1975. These fire and theft data were organized by zip code.

Finally the US Bureau of the census supplied data on racial composition, income and age and value of residential units for each ZIP code in Chicago. To adjust for these differences in the populations size associated with different ZIP code areas, the theft data were expressed as incidents per 1,000 population and the fire and insurance data as incidents per 100 housing units.

The variables are

`race` racial composition in percent minority

`fire` fires per 100 housing units

`theft` theft per 1000 population

`age` percent of housing units built before 1939

`volact` new homeowner policies plus renewals minus cancellations and non renewals per 100 housing units

`involact` new FAIR plan policies and renewals per 100 housing units

income median family income

The data comes from the book by Andrews and Herzberg (1985). We choose the involuntary market activity variable (the number getting FAIR plan insurance) as the response since this seems to be the best measure of those who are denied insurance by others. It is not a perfect measure because some who are denied insurance may give up and others still may not try at all for that reason. The voluntary market activity variable is not as relevant.

Furthermore, we do not know the race of those denied insurance. We only know the racial composition in the corresponding zip code. This is an important difficulty and brings up the following topic:

Ecological Correlation

When data is 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)
> 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. We can see a clear negative correlation.

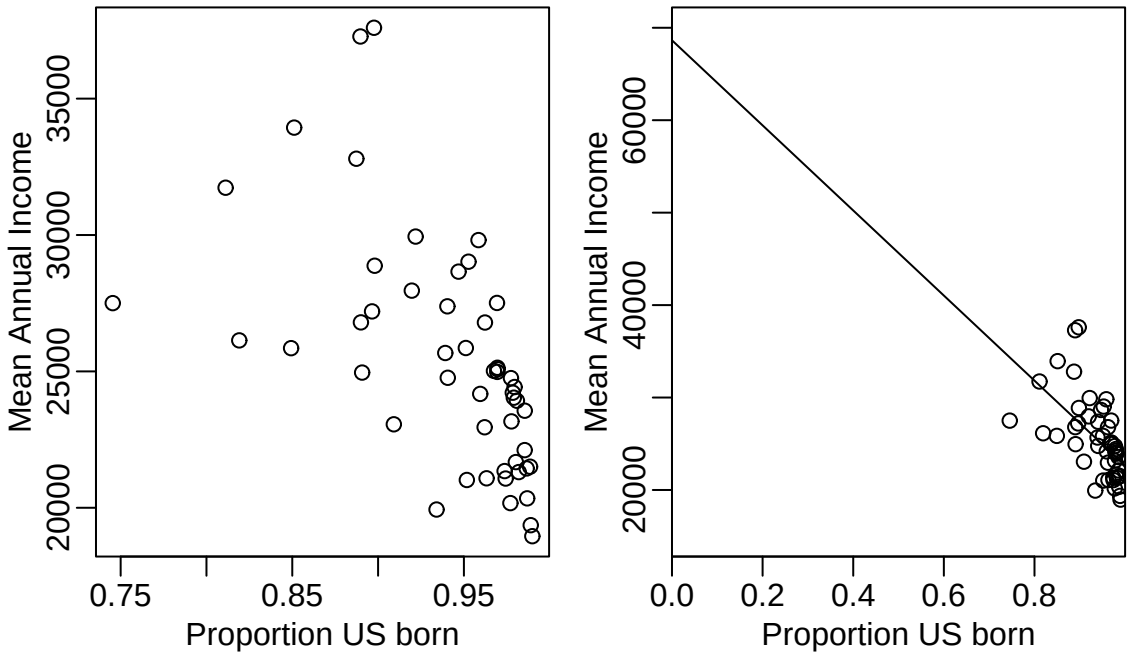

Figure 12.1: 1998 annual per capita income and proportion US born for 50 states plus DC. Plot on the right is the same data as on the left but with an extended scale and the least squares fit shown

We can fit a regression line and show the fitted line on an extended range:

```
> g <- lm(income ~ usborn, eco)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    68642       8739    7.85  3.2e-10
usborn        -46019       9279   -4.96  8.9e-06

Residual standard error: 3490 on 49 degrees of freedom
Multiple R-Squared: 0.334,      Adjusted R-squared: 0.321
F-statistic: 24.6 on 1 and 49 degrees of freedom, p-value: 8.89e-06

> plot(income ~ usborn, data=eco, xlab="Proportion US born",
  ylab="Mean Annual Income",xlim=c(0,1),ylim=c(15000,70000),xaxs="i")
> abline(g$coef)
```

We see that there is a clear statistical significant relationship between 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 are three times wealthier than US born citizens. In truth, information 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 this data because immigrants are naturally attracted to wealthier states.

This is also relevent 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.

We will focus on the relationship between race and the response although similar analyses might be done for the income variable.

Start by reading the data in and examining it:

```
> data(chicago)
> chicago
      race fire theft  age volact involact income
60626 10.0  6.2    29 60.4    5.3      0.0  11744
60640 22.2  9.5    44 76.5    3.1      0.1   9323
etc.
60645  3.1  4.9    27 46.6   10.9      0.0  13731
```

Rescale the income variable and omit volact

```
> ch <- data.frame(chicago[,1:4],involact=chicago[,6],income=chicago[,7]/1000)
> ch
      race fire theft  age involact income
60626 10.0  6.2    29 60.4      0.0 11.744
60640 22.2  9.5    44 76.5      0.1  9.323
etc.
60645  3.1  4.9    27 46.6     0.0       13.731
```

Summarize:

```
> summary(ch)
      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   :35.00   Mean   :12.30   Mean   : 32.4   Mean   :60.3
 3rd Qu.:57.60   3rd Qu.:16.10   3rd Qu.: 38.0   3rd Qu.:77.3
 Max.   :99.70   Max.   :39.70   Max.   :147.0   Max.   :90.1
    involact         income
 Min.   :0.000   Min.   : 5.58
 1st Qu.:0.000   1st Qu.: 8.45
 Median :0.400   Median :10.70
 Mean   :0.615   Mean   :10.70
 3rd Qu.:0.900   3rd Qu.:12.00
 Max.   :2.200   Max.   :21.50
```

We see that there is a wide range in the `race` variable with some zip codes being 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 homogenous, we would never be able to discover an effect from this aggregated data. We also note some skewness in the `theft` and `income` variables. The response `involact` has a large number of zeroes. This is not good for the assumptions of the linear model but we have little choice but to proceed.

Now make some graphical summaries:

```
> par(mfrow=c(2,3))
> for(i in 1:6) hist(ch[,i],main=names(ch)[i])
> for(i in 1:6) boxplot(ch[,i],main=names(ch)[i])
> pairs(ch)
```

Only the boxplots are shown in Figure 12.

An examination of the data using xgobi would also be worthwhile.

Now look at the relationship between involact and race:

```
> summary(lm(involact ~ race,data=ch))
Coefficients:
            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

Residual standard error: 0.449 on 45 degrees of freedom
Multiple R-Squared: 0.509,      Adjusted R-squared: 0.499
F-statistic: 46.7 on 1 and 45 degrees of freedom,       p-value: 1.78e-08
```

We can clearly see that homeowners in zip codes with high % minority are being denied insurance at 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? For example, we see that % minority is correlated

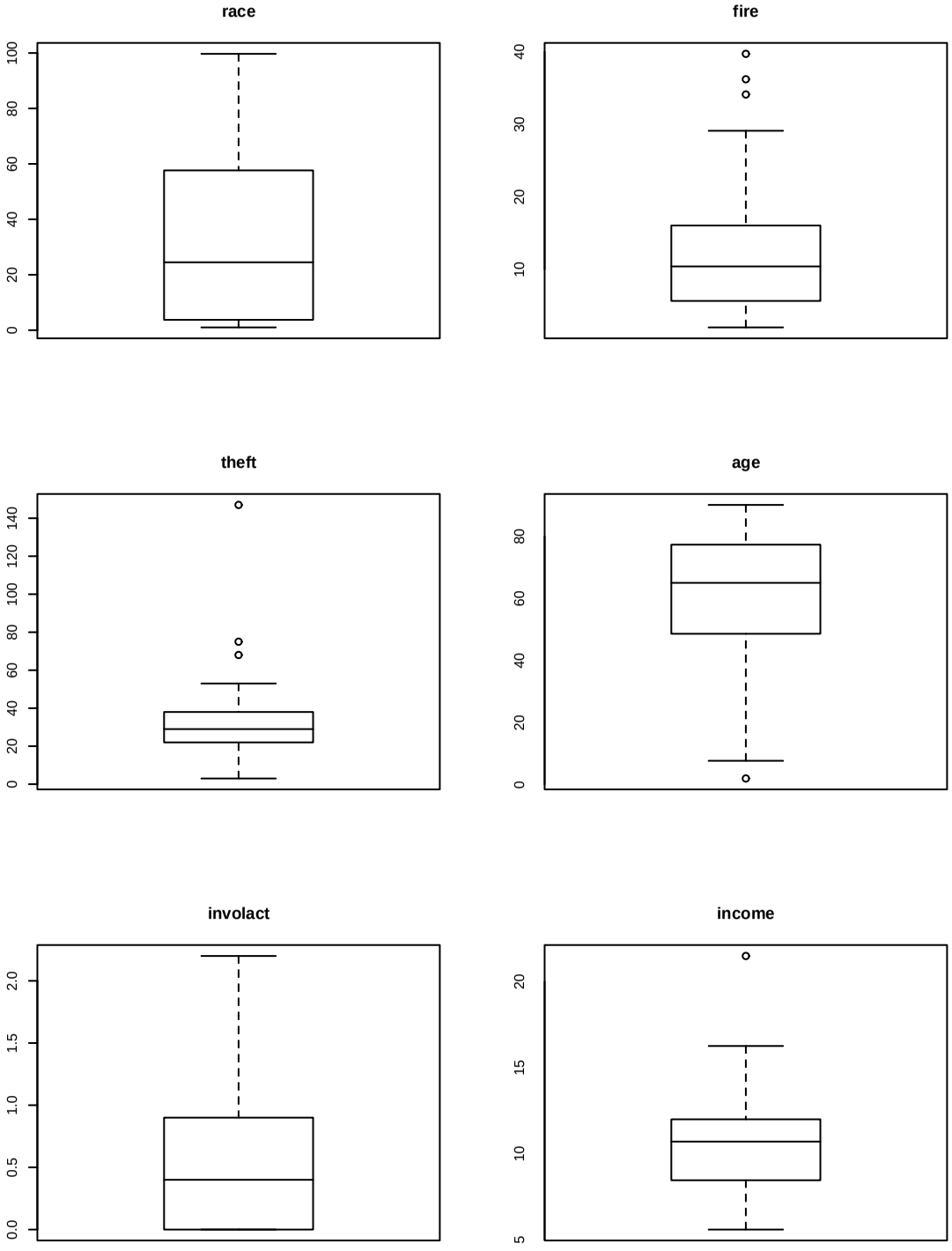

Figure 12.2: Boxplots of the Chicago Insurance data

with the fire rate from the plots. The insurance companies could say that they were denying insurance in neighborhoods where they had sustained large fire-related losses and any discriminatory effect was a by-product of (presumably) legitimate business practice. What can regression analysis tell us about this claim?

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 US are generally lower than those of men. However, if one adjusts for various factors 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 factors be used to make the adjustment?

In this example, suppose that if 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 here just to see what happens.

I use log(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.

We start with the full model:

```
> g <- lm(involact ~ race + fire + theft + age + log(income), data = ch)
Coefficients:
             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  4.8e-05
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

Residual standard error: 0.335 on 41 degrees of freedom
Multiple R-Squared: 0.752,      Adjusted R-squared: 0.721
F-statistic: 24.8 on 5 and 41 degrees of freedom,       p-value: 2.01e-11
```

Before we start making any conclusions, we should check the model assumptions.

```
> plot(g$fit,g$res,xlab="Fitted",ylab="Residuals",
  main="Residual-Fitted plot")
> abline(h=0)
> qqnorm(g$res)
```

These two diagnostic plots are shown in Figure 12.

The diagonal streak in the residual-fitted plot is caused by the large number of zero response values in the data. When $y =$, the residual $\hat{\varepsilon} = \hat{y} = -x^T\hat{\beta}$, hence the line. Turning a blind eye to this feature, we see no particular problem. The Q-Q plot looks fine too.

Now let's look at influence - what happens if points are excluded? We'll use a function `qqnorml()` that I wrote that labels the points in a Q-Q plot with the case numbers Plot not shown but cases 6 and 24 seem to stick out.

```
> gi <- lm.influence(g)
> for(i in 1:5) qqnorml(gi$coef[,i+1],main=names(ch)[-5][i])
```

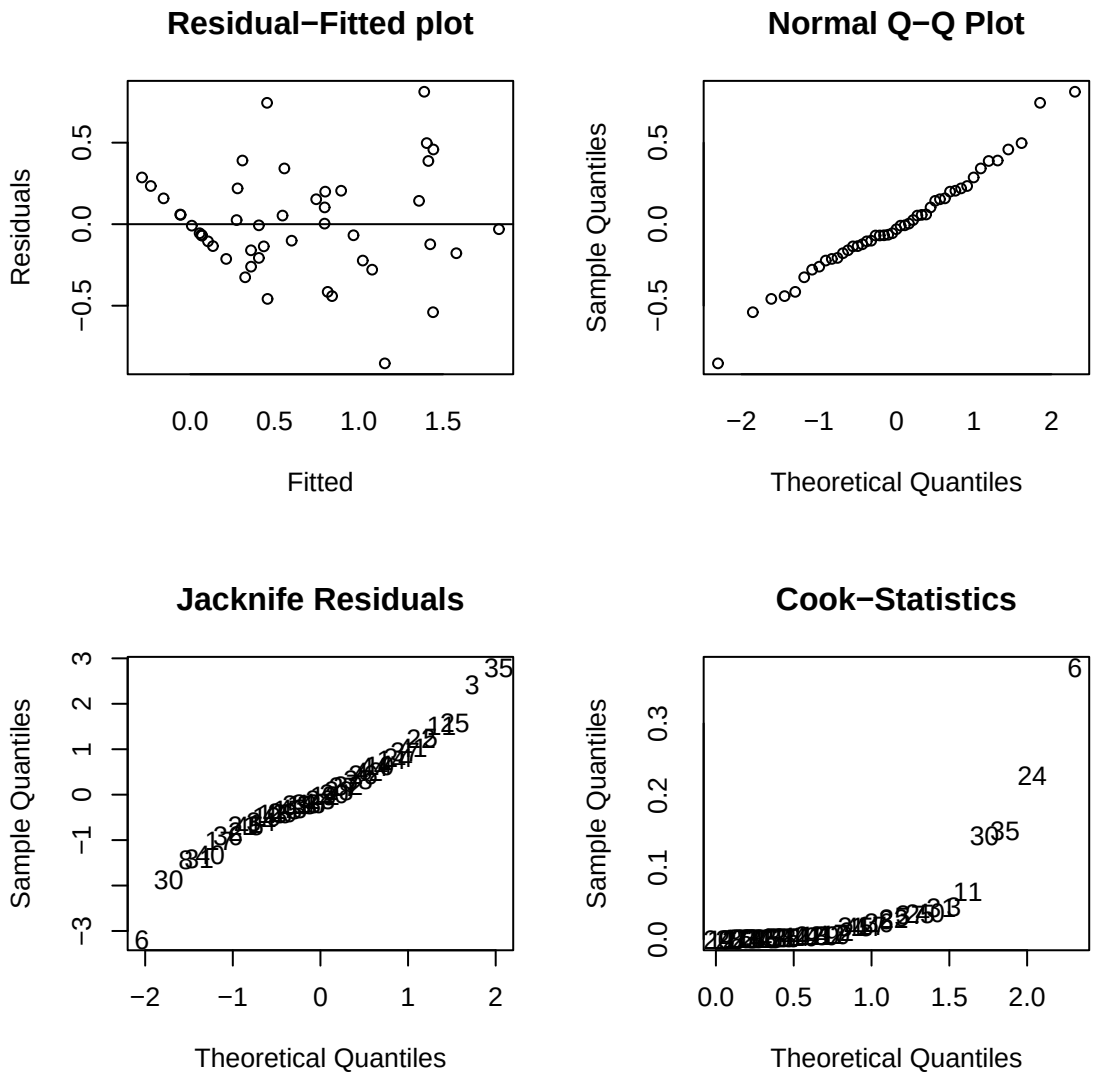

Figure 12.3: Diagnostic plots of the Chicago Insurance data

Check out the jacknife residuals:

```
> qqnorml(rstudent(g),main="Jacknife Residuals")
> qt(0.05/(2*47),47-6-1)
[1] -3.529468
```

Nothing too extreme - now look at the Cook statistics using the `halfnorm()` function that I wrote:

```
> halfnorm(cooks.distance(g),main="Cook-Statistics")
```

Cases 6 and 24 stick out again. Let's take a look at these two cases:

```
> ch[c(6,24),]
      race fire theft  age involact income
```

```
60610 54.0 34.1    68 52.6      0.3  8.231
60607 50.2 39.7   147 83.0      0.9  7.459
```

These are high theft and fire zip codes. See what happens when we exclude these points:

```
> g <- lm(involact ~ race + fire + theft + age + log(income),ch,
              subset=(1:47)[-c(6,24)])
> summary(g)
Coefficients:
            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

Residual standard error: 0.303 on 39 degrees of freedom
Multiple R-Squared: 0.804,      Adjusted R-squared: 0.779
F-statistic:   32 on 5 and 39 degrees of freedom,       p-value: 8.19e-13
```

`theft` and `age` are no longer significant at the 5% level. We now address the question of transformations - because the response has some zero values and for interpretational reasons we will not try to transform it. Similarly, since the race variable is the primary predictor of interest we won't try transforming it either so as to avoid interpretation difficulties. We try fitting a polynomial model with quadratic terms in each of the predictors:

```
> g2 <- lm(involact ~ race + poly(fire,2) + poly(theft,2) + poly(age,2)
       + poly(log(income),2), ch, subset=(1:47)[-c(6,24)])
> anova(g,g2)
Analysis of Variance Table

Model 1: involact ~ race + fire + theft + age + log(income)
Model 2: involact ~ race + poly(fire, 2) + poly(theft, 2) + poly(age,
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     39       3.59
2     35       3.20  4   0.38    1.04    0.4
```

Seems that we can do without the quadratic terms. A check of the partial residual plots also reveals no need to transform. We now move on to variable selection. We are not so much interested in picking one model here because we are mostly interested in the dependency of involact on the race variable. So $\hat{\beta}_1$ is the thing we want to focus on. The problem is that collinearity with the other variables may cause $\hat{\beta}_1$ to vary substantially depending on what other variables are in the model. We address this question here. `leaps()` is bit picky about its input format so I need to form the x and y explicitly:

```
> y <- ch$inv[cooks.distance(g) < 0.2]
> x <- cbind(ch[,1:4],linc=log(ch[,6]))
> x <- x[cooks.distance(g) < 0.2,]
```

Removing all points with Cook's Statistics greater than 0.2 takes out cases 6 and 24.
We make the Cp plot.

```
> library(leaps)
> a <- leaps(x,y)
> Cpplot(a)
```

See Figure 12.

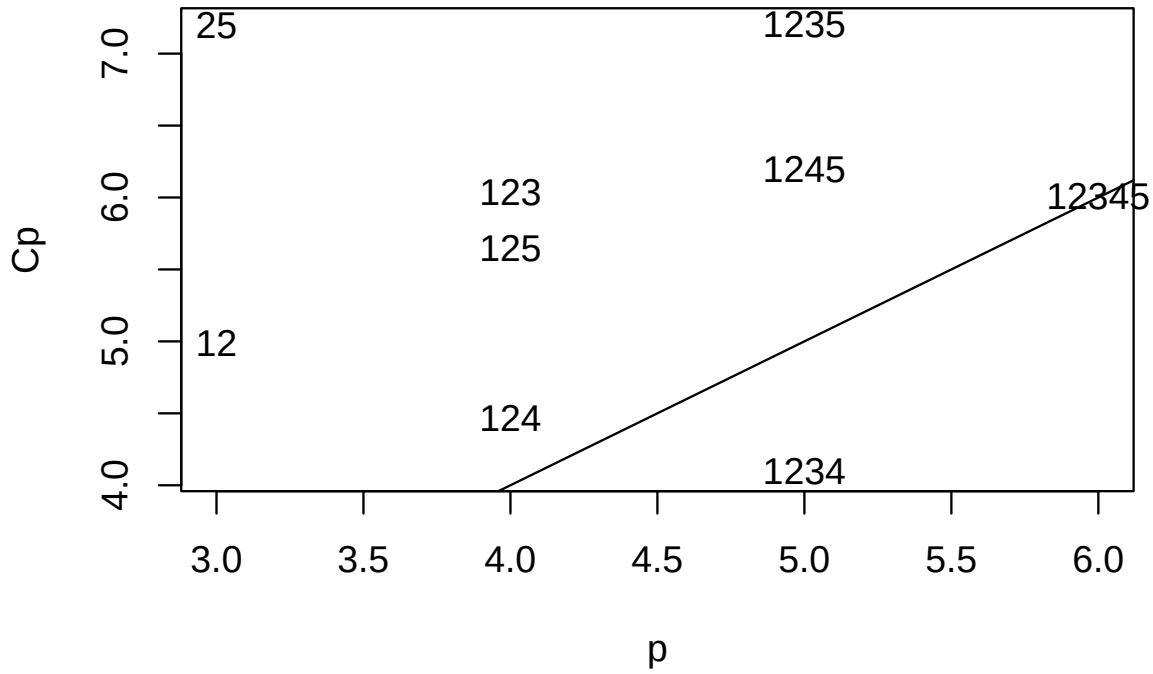

Figure 12.4: Cp plot of the Chicago Insurance data

The best model seems to be this one:

```
> g <- lm(involact ~ race + fire + theft + age, ch, subset=(1:47)[-c(6,24)])
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.26787    0.13967   -1.92   0.0623
race         0.00649    0.00184    3.53   0.0011
fire         0.04906    0.00823    5.96  5.3e-07
theft       -0.00581    0.00373   -1.56   0.1271
age          0.00469    0.00233    2.01   0.0514

Residual standard error: 0.3 on 40 degrees of freedom
Multiple R-Squared: 0.804,      Adjusted R-squared: 0.784
F-statistic: 40.9 on 4 and 40 degrees of freedom,       p-value: 1.24e-13
```

The fire rate is also significant and actually has higher t-statistics. Thus, we have verified that there is a positive relationship between involact and race while controlling for a selection of the other variables.

How robust is the conclusion? Would other analysts have come to the same conclusion? One alternative model is

```
> galt <- lm(involact ~ race+fire+log(income),ch,subset=(1:47)[-c(6,24)])
> summary(galt)
Coefficients:
            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

Residual standard error: 0.309 on 41 degrees of freedom
Multiple R-Squared: 0.786,      Adjusted R-squared: 0.77
F-statistic: 50.1 on 3 and 41 degrees of freedom,       p-value: 8.87e-14
```

In this model, we see that `race` is not statistically significant. The previous model did fit slightly better but it is important that there exists a reasonable model in which `race` is not significant since although the evidence seems fairly strong in favor of a `race` effect, it is not entirely conclusive. Interestingly enough, if `log(income)` is dropped:

```
> galt <- lm(involact ~ race+fire,ch,subset=(1:47)[-c(6,24)])
> summary(galt)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.19132    0.08152   -2.35   0.0237
race         0.00571    0.00186    3.08   0.0037
fire         0.05466    0.00784    6.97  1.6e-08

Residual standard error: 0.31 on 42 degrees of freedom
Multiple R-Squared: 0.779,      Adjusted R-squared: 0.769
F-statistic: 74.1 on 2 and 42 degrees of freedom,       p-value: 1.7e-14
```

we find `race` again becomes significant which raises again the question of whether income should be adjusted for since it makes all the difference here.

We now return to the two left-out cases. Observe the difference in the fit when the two are re-included on the best model. The quantities may change but the qualitative message is the same. It is better to include all points if possible, especially in a legal case like this where excluding points might lead to criticism and suspicion of the results.

```
> g <- lm(involact ~ race + fire + theft + age, data=ch)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.24312    0.14505   -1.68  0.10116
race         0.00810    0.00189    4.30  0.00010
fire         0.03665    0.00792    4.63  3.5e-05
theft       -0.00959    0.00269   -3.57  0.00092
age          0.00721    0.00241    2.99  0.00460
```

```
Residual standard error: 0.334 on 42 degrees of freedom
Multiple R-Squared: 0.747,      Adjusted R-squared: 0.723
F-statistic:   31 on 4 and 42 degrees of freedom,       p-value: 4.8e-12
```

The main message of the data is not changed - we should check the diagnostics. I found no trouble. (Adding back in the two points to the `race+fire+log(income)` model made `race` significant again. So it looks like there is some good evidence that zip codes with high minority populations are being "red-lined" - that is improperly denied insurance. While there is evidence that some of the relationship between race and involact can be explained by the fire rate, there is still a component that cannot be attributed to the other variables.

However, there is some doubt due to the response not being a perfect measure of people being denied insurance. It is an aggregate measure which raises the problem of ecological correlations. We have implicitly assumed that the probability that 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 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.

Another point to be considered is the size of the effect. The largest value of the response is only 2.2% and most cases are much smaller. Even assuming the worst, the number of people affected is small.

There is also the problem of a potential latent variable that might be the true cause of the observed relationship, but it is difficult to see what that variable might be. Nevertheless, this always casts a shadow of doubt on our conclusions.

There are some special difficulties in presenting this during a court case. With scientific enquiries, there is always room for uncertainty and subtlety in presenting the results, but this is much more difficult in the court room. The jury may know no statistics and lawyers are clever at twisting words. A statistician giving evidence as an expert witness would do well to keep the message simple.

Another issue that arises in cases of this nature is how much the data should be aggregated. For example, I divided the data using a zip code map of Chicago into north and south. Fit the model to the south of Chicago:

```
> data(chiczip)
> g <- lm(involact ~ race + fire + theft +age, subset=(chiczip == "s"), ch)
> summary(g)
Coefficients:
            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

Residual standard error: 0.351 on 17 degrees of freedom
Multiple R-Squared: 0.743,      Adjusted R-squared: 0.683
F-statistic: 12.3 on 4 and 17 degrees of freedom,       p-value: 6.97e-05
```

and now to the north.

```
> g <- lm(involact ~ race + fire + theft +age, subset=(chiczip == "n"), ch)
> summary(g)
Coefficients:
            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

Residual standard error: 0.343 on 20 degrees of freedom
Multiple R-Squared: 0.756,      Adjusted R-squared: 0.707
F-statistic: 15.5 on 4 and 20 degrees of freedom,       p-value: 6.52e-06
```

What differences do you see? 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 often a point of contention in legal cases.

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 thing about Statistics in relation to how it helps us reason in the face of uncertainty.

Chapter 13

Robust and Resistant Regression

When the errors are normal, least squares regression is clearly best but when the errors are nonnormal, other methods may be considered. A particular concern is long-tailed error distributions. One approach is to remove the largest residuals as outliers and still use least squares but this may not be effective when there are several large residuals because of the leave-out-one nature of the outlier tests. Furthermore, the outlier test is an accept/reject procedure that is not smooth and may not be statistically efficient for the estimation of β. Robust regression provides an alternative.

There are several methods. M-estimates choose β to minimize

$$\sum_{i=1}^{n} \rho\left(\frac{y_i - x_i^T\beta}{\sigma}\right)$$

Possible choices for ρ are

1. $\rho(x) = x^2$ is just least squares
2. $\rho(x) = |x|$ is called least absolute deviations regression (LAD). This is also called L_1 regression.
3. $$\rho(x) = \begin{cases} x^2/2 & \text{if } |x| \le c \\ c|x| - c^2/2 & \text{otherwise} \end{cases}$$

   is called Huber's method and is a compromise between least squares and LAD regression. c can be an estimate of σ but not the usual one which is not robust. Something $\propto$ median$|\hat{\varepsilon}_i|$ for example.

Robust regression 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:

$$\sum_{i=1}^{n} w_i x_{ij}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j) = 0 \quad j = 1, \ldots p$$

Now differentiating the M-estimate criterion with respect to β_j and setting to zero we get

$$\sum_{i=1}^{n} \rho'\left(\frac{y_i - \sum_{j=1}^{p} x_{ij}\beta_j}{\sigma}\right) x_{ij} = 0 \quad j = 1, \ldots p$$

Now let $u_i = y_i - \sum_{j=1}^{p} x_{ij}\beta_j$ to get

$$\sum_{i=1}^{n} \frac{\rho'(u_i)}{u_i} x_{ij}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j) = 0 \quad j = 1, \ldots p$$

so we can make the identification of

$$w(u) = \rho'(u)/u$$

and we find for our choices of ρ above:

1. LS: $w(u)$ is constant.
2. LAD: $w(u) = 1/|u|$ - note the asymptote at 0 - this makes a weighting approach difficult.
3. Huber:

$$w(u) = \begin{cases} 1 & \text{if } |u| \leq c \\ c/|u| & \text{otherwise} \end{cases}$$

There are many other choices that have been used. Because the weights depend on the residuals, an iteratively reweighted least squares approach to fitting must be used. We can sometimes get standard errors by vâr $\hat{\beta} = \hat{\sigma}^2(X^T W X)^{-1}$ (use a robust estimate of σ^2 also).

We demonstrate the methods on the Chicago insurance data. Using least squares first.

```
> data(chicago)
> g <- lm(involact ~ race + fire + theft + age + log(income),chicago)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.57398    3.85729   -0.93  0.35958
race         0.00950    0.00249    3.82  0.00045
fire         0.03986    0.00877    4.55  4.8e-05
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

Residual standard error: 0.335 on 41 degrees of freedom
Multiple R-Squared: 0.752,      Adjusted R-squared: 0.721
F-statistic: 24.8 on 5 and 41 degrees of freedom,       p-value: 2.01e-11
```

Least squares works well when there are normal errors but can be upset by long-tailed errors. A convenient way to apply the Huber method is to apply the `rlm()` function which is part of the MASS (see the book Modern Applied Statistics in S+) which also gives standard errors. The default is to use the Huber method but there are other choices.

```
> library(MASS)
> g <- rlm( involact ~ race + fire + theft + age + log(income), chicago)
Coefficients:
            Value  Std. Error t value
(Intercept) -2.926  3.397     -0.861
race         0.008  0.002      3.583
```

```
fire         0.046  0.008      5.940
theft       -0.010  0.002     -3.912
age          0.006  0.002      2.651
log(income)  0.283  0.352      0.803

Residual standard error: 0.249 on 41 degrees of freedom
```

The R^2 and F-statistics are not given because they cannot be calculated (at least not in the same way). The numerical values of the coefficients have changed a small amount but the general significance of the variables remains the same and our substantive conclusion would not be altered. 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.

Another method that can be used is Least Trimmed Squares(LTS). Here one minimizes $\sum_{i=1}^{q} \hat{\varepsilon}^2_{(i)}$ 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. The Huber and L_1 methods will still fail if some $\varepsilon_i \to \infty$. LTS is an example of a *resistant* regression method. Resistant methods are good at dealing with data where we expect there to be a certain number of "bad" observations that we want to have no weight in the analysis.

```
> library(lqs)
> g <- ltsreg(involact ~ race + fire + theft + age + log(income),chicago)
> g$coef
(Intercept)        race        fire       theft         age log(income)
 -1.6950187   0.0037348   0.0549117  -0.0095883   0.0018549   0.1700325
> g <- ltsreg(involact ~ race + fire + theft + age + log(income),chicago)
> g$coef
(Intercept)        race        fire       theft         age log(income)
  2.2237795   0.0050697   0.0423565  -0.0084868   0.0008755  -0.2398183
```

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. I repeated the command twice and you will notice that the results are somewhat different. This is because the default genetic algorithm used to compute the coefficients is non-deterministic. An exhaustive search method can be used

```
> g <- ltsreg(involact ~ race + fire + theft + age + log(income),chicago,
                nsamp="exact")
> g$coef
(Intercept)        race        fire       theft         age log(income)
-1.12093591  0.00575147  0.04859848 -0.00850985  0.00076159  0.11251547
```

This takes about 20 minutes on a 400Mhz Intel Pentium II processor. For larger datasets, it will take much longer so this method might be impractical.

The most notable difference from LS for the purposes of this data is the decrease in the race coefficient - if the same standard error applied then it would verge on insignificance. However, we don't have the standard errors for the LTS regression coefficients. We now use a general method for inference which is especially useful when such theory is lacking - the Bootstrap.

To understand how this method works, think about how we might empirically determine the distribution of an estimator. 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 don't know the true model. The Bootstrap emulates the simulation procedure above except instead of sampling from the true model, it samples from the observed data itself. Remarkably, this technique is often effective. It sidesteps the need for theoretical calculations that may be extremely difficult or even impossible. The Bootstrap may be the single most important innovation in Statistics in the last 20 years.

To see how the bootstrap method compares with simulation, let's spell out the steps involved. In both cases, we consider X fixed.

Simulation

In general 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 case, 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.
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 for 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 don't know the true error distribution and we don't know the true β.

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 itself.

Bootstrap

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

This time, we use only quantities that we know. For small n, it is possible to compute $\hat{\beta}^*$ for every possible sample from $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_n$, 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=T)
 [1] 7 9 9 2 5 7 4 1 8 9
```

and hence a random sample (with replacement) of RTS residuals is:

```
> g$res[sample(47,rep=T)]
    60639     60641     60634     60608     60608     60612     60651     60620
 0.091422 -0.039899  0.013526  0.342344  0.342344 -0.022214  0.255031  0.333714
```

(rest deleted

You will notice that there is a repeated value even in this small snippet. We now execute the bootstrap - first we make a matrix to save the results in and then repeat the bootstrap process 1000 times: (This takes about 6 minutes to run on a 400Mhz Intel Pentium II processor)

```
> x <- model.matrix(~ race+fire+theft+age+log(income),chicago)[,-1]
> bcoef <- matrix(0,1000,6)
> for(i in 1:1000){
+ newy <- g$fit + g$res[sample(47,rep=T)]
+ brg <- ltsreg(x,newy,nsamp="best")
+ bcoef[i,] <- brg$coef
+ }
```

It is not convenient to use the `nsamp="exact"` since that would require 1000 times the 20 minutes it takes to make original estimate. That's about two weeks, so I 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. It is notable that this calculation was the only one in this book that did not take a negligible amount of time. You typically do not need the latest and greatest computer to do statistics on the size of datasets encountered in this book.

To test the null hypothesis that $H_0 : \beta_{race} = 0$ against the alternative $H_1 : \beta_{race} > 0$ we may figure what fraction of the bootstrap sampled β_{race} were less than zero:

```
> length(bcoef[bcoef[,2]<0,2])/1000
[1] 0.019
```

So our p-value is 1.9% and we reject the null at the 5% level.

We can also make a 95% confidence interval for this parameter by taking the empirical quantiles:

```
> quantile(bcoef[,2],c(0.025,0.975))
      2.5%      97.5%
0.00099037 0.01292449
```

We can get a better picture of the distribution by looking at the density and marking the confidence interval:

```
> plot(density(bcoef[,2]),xlab="Coefficient of Race",main="")
> abline(v=quantile(bcoef[,2],c(0.025,0.975)))
```

See Figure 13.1. We see that the distribution is approximately normal with perhaps so longish tails.

This would be more accurate if we took more than 1000 bootstrap resamples. The conclusion here would be that the race variable is significant but the effect is less than that estimated by least squares. Which is better? This depends on what the "true" model is which we will never know but since the QQ plot did not indicate any big problem with non-normality I would tend to prefer the LS estimates. However, this does illustrate a general problem that occurs when more than one statistical method is available for a given dataset.

Summary

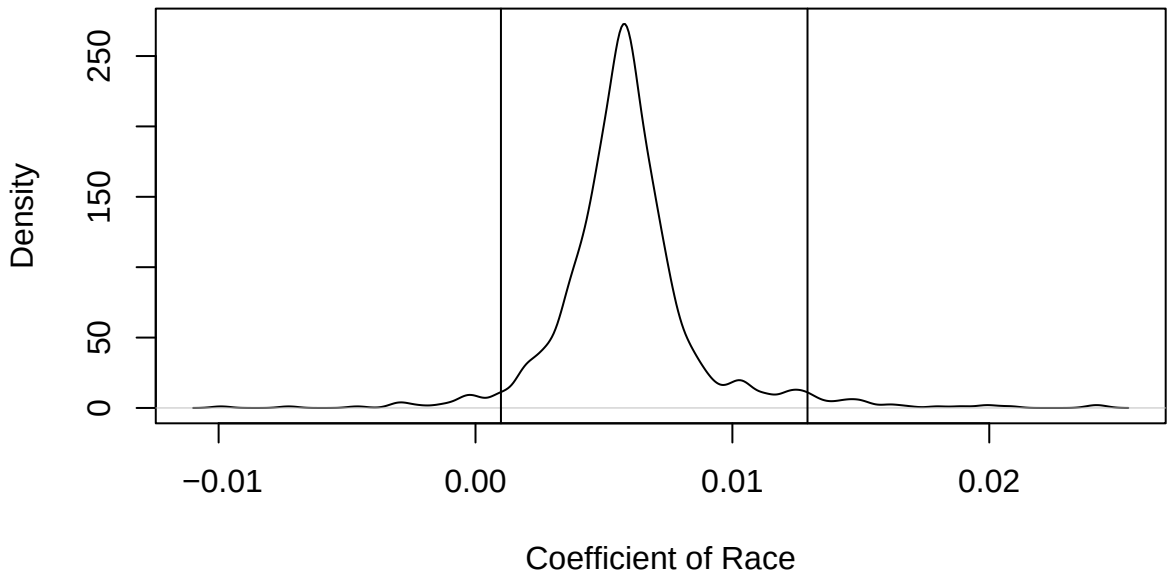

Figure 13.1: Bootstrap distribution of $\hat{\beta}_{race}$ with 95% confidence intervals

1. Robust estimators provide protection against long-tailed errors but they can't overcome problems with the choice of model and its variance structure. This is unfortunate because these problems are more serious than non-normal error.

2. Robust estimates just give you $\hat{\beta}$ and possibly standard errors without the associated inferential methods. Software and methodology for this inference is not easy to come by. The bootstrap is a general purpose inferential method which is useful in these situations.

3. Robust methods can be used in addition to LS as a confirmatory method. You have cause to worry if the two estimates are far apart.

Chapter 14

Missing Data

Missing data is the situation where some values of some cases are missing. This is not uncommon. Dealing with missing data is time consuming. In my experience, fixing up problems caused by missing data sometimes takes longer than the analysis itself.

What can be done? Obviously, finding the missing values is the best option but this is not always possible. Next ask why the data are missing. If the reason for a datum being missing is non-informative, then a fix is easier. For example, if a data point is missed because it was large then this could cause some bias and a simple fix is not possible. Patients may drop out of a drug study because they feel their treatment is not working - this would cause bias.

Here are several fix-up methods to use when data are missing for noninformative reasons:

1. Delete the case with missing observations. This is OK if this only causes the loss of a relatively small number of cases. This is the simplest solution.

2. Fill-in or *impute* the missing values. Use the rest of the data to predict the missing values. Simply replacing the missing value of a predictor with the average value of that predictor is one easy method. Using regression on the other predictors is another possibility. It's not clear how much the diagnostics and inference on the filled-in dataset is affected. Some additional uncertainty is caused by the imputation which needs to be allowed for.

3. Missing observation correlation. Consider just (x_i, y_i) pairs with some observations missing. The means and SDs of x and y can be used in the estimate even when a member of a pair is missing. An analogous method is available for regression problems.

4. Maximum likelihood methods can be used assuming the multivariate normality of the data. The EM algorithm is often used here. We will not explain the details but the idea is essentially to treat missing values as nuisance parameters.

Suppose some of the values in the Chicago Insurance dataset were missing. I randomly declared some the observations missing in this modified dataset. Read it in and take a look:

```
> data(chmiss)
> 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
```

There are 20 missing observations denoted by NA here. It's important to know what the missing value code is for the data and/or software you are using. What happens if we try to fit the model?

```
> g <- lm(involact ~ .,chmiss)
> summary(g)
Coefficients:
            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

Residual standard error: 0.338 on 21 degrees of freedom
Multiple R-Squared: 0.791,      Adjusted R-squared: 0.741
F-statistic: 15.9 on 5 and 21 degrees of freedom,       p-value: 1.59e-06
```

Any case with at least one missing value is omitted from the regression. You can see there are now only 21 degrees of freedom - almost half the data is lost. We can fill in the missing values by their variable means as in:

```
> cmeans <- apply(chmiss,2,mean,na.rm=T)
> cmeans
    race     fire    theft      age involact   income
35.60930 11.42444 32.65116 59.96905  0.64773 10.73587
> mchm <- chmiss
> for(i in c(1,2,3,4,6)) mchm[is.na(chmiss[,i]),i] <- cmeans[i]
```

We don't fill in missing values in the response because this is the variable we are trying to model. Now refit:

```
> g <- lm(involact ~ ., data=mchm)
> summary(g)
Coefficients:
             Value Std. Error t value Pr(>|t|)
(Intercept)  0.0707  0.5094     0.1387  0.8904
       race  0.0071  0.0027     2.6307  0.0122
       fire  0.0287  0.0094     3.0623  0.0040
      theft -0.0031  0.0027    -1.1139  0.2723
        age  0.0061  0.0032     1.8954  0.0657
     income -0.0271  0.0317    -0.8550  0.3979

Residual standard error: 0.3841 on 38 degrees of freedom
Multiple R-Squared: 0.6819
F-statistic: 16.3 on 5 and 38 degrees of freedom, the p-value is 1.41e-08
```

Compare with the previous results - what differences do you see? Different statistical packages have different ways of handling missing observations. For example, the default behavior in S-PLUS would refuse to fit the model at all.

The regression coefficients are now all closer to zero. The situation is analogous to the error 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.

We can also use regression methods to predict the missing values of the covariates. Let's try to fill-in the missing race values:

```
> gr <- lm(race ~ fire+theft+age+income,chmiss)
> chmiss[is.na(chmiss$race),]
      race fire theft  age involact income
60646   NA  5.7    11 27.9      0.0 16.250
60651   NA 15.1    30 89.8      0.8 10.510
60616   NA 12.2    46 48.0      0.6  8.212
60617   NA 10.8    34 58.0      0.9 11.156
> predict(gr,chmiss[is.na(chmiss$race),])
  60646   60651   60616   60617
-17.847  26.360  70.394  32.620
```

Can you see a problem with filling these values in? Obviously we would need to put more work into the regression models used to fill-in the missing values. One trick that can be applied when the response is bounded between 0 and 1 is the logit transformation:

$$y \to \log(y/(1-y))$$

This transformation maps to the whole real line. We define the logit function and its inverse:

```
> logit <- function(x) log(x/(1-x))
> ilogit <- function(x) exp(x)/(1+exp(x))
```

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:

```
> gr <- lm(logit(race/100) ~ fire+theft+age+income,chmiss)
> ilogit(predict(gr,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:

```
> data(chicago)
> chicago$race[is.na(chmiss$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.

Like the mean fill-in method, regression fill-in will also introduce a bias towards zero in the coefficients while tending to reduce the variance also. 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.

For situations where there is a substantial proportion of missing data, I recommend that you investigate more sophisticated methods, likely using the EM algorithm. Multiple imputation is another possibility. The fill-in methods described above will be fine when only a few cases need to be filled but will become less reliable as the proportion of missing cases increases.

Chapter 15

Analysis of Covariance

Predictors that are qualitative in nature, like for example eye color, are sometimes called *categorical* or *factors*. How can these predictors be incorporated into a regression analysis? Analysis of Covariance refers to regression problems where there is a mixture of quantitative and qualitative predictors.

Suppose we are interested in the effect of a medication on cholesterol level - we might have two groups - one of which receives the medication and the other which does not. However, we could not treat this as a simple two sample problem if we knew that the two groups differed with respect to age and this would affect the cholesterol level. See Figure 15 for a simulated example. For the patients who received the medication, the mean reduction in cholesterol level was 0% while for those who did not the mean reduction was 10%. So superficially it would seem that it would be better not to be treated. However, the treated group ranged in age from 50 to 70 while those who were not treated ranged in age between 30 and 50. We can see that once age is taken into account, the difference between treatment and control is again 10% but this time in favor of the treatment.

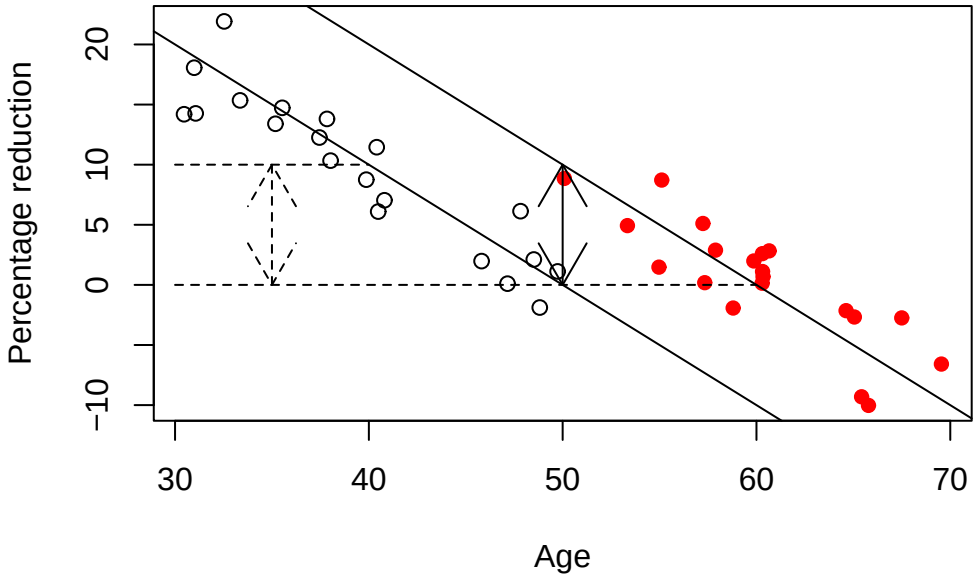

Figure 15.1: Simulated example showing the confounding effect of a covariate. The patients who took the medication are marked with a solid dot while those who did not are marked with an empty dot

Analysis of covariance is a method for adjusting the groups for the age difference and then seeing the

effect of the medication. It can also be used when there are more than two groups and more than one covariate.

> Our strategy is to incorporate the qualitative predictors within the $y = X\beta + \varepsilon$ framework. We can then use the estimation, inferential and diagnostic techniques that we have already learnt.

This avoids have to learn a different set of formulae for each new type of qualitative predictor configuration which is the approach taken by many texts. To put qualitative predictors into the $y = X\beta + \varepsilon$ form we need to code the qualitative predictors. Let's consider a specific example:

- y = change in cholesterol level
- x = age
- $$d = \begin{cases} 0 & \text{did not take medication} \\ 1 & \text{took medication} \end{cases}$$

A variety of linear models may be considered here:

1. The same regression line for both groups — $y = \beta_0 + \beta_1 x + \varepsilon$ or in `R` `y ~ x`
2. Separate regression lines for each group but with the same slope — $y = \beta_0 + \beta_1 x + \beta_2 d + \varepsilon$ or in `R` `y ~ x + d`. In this case β_2 represents the distance between the regression lines i.e. the effect of the drug.
3. Separate regression lines for each group $y = \beta_0 + \beta_1 x + \beta_2 d + \beta_3 x.d + \varepsilon$ or in `R` `y ~ x + d + d:x` or `y ~ x*d`. Any interpretation of the effect of the drug will now depend on age also. To form the slope interaction term $x.d$ in the X-matrix, simply multiply x by d elementwise.

Estimation and testing works just as it did before. Interpretation is much easier if we can eliminate the slope interaction term.

Other codings of d are possible, for instance

$$d = \begin{cases} -1 & \text{did not take medication} \\ 1 & \text{took medication} \end{cases}$$

is used by some. This coding enables β_2 and β_3 to be viewed as differences from a response averaged over the two groups. Any other coding that assigned a different number to the two groups would also work but interpretation of the estimated parameters might be more difficult.

15.1 A two-level example

The data for this example consist of `x`= nave height and `y` = total length in feet for English medieval cathedrals. Some are in the Romanesque (r) style and others are in the Gothic (g) style. Some cathedrals have parts in both styles and are listed twice. We wish to investigate how the length is related to height for the two styles. Read in the data and make a summary on the two styles separately.

```
> data(cathedral)
> cathedral
             style   x   y
Durham           r  75 502
Canterbury       r  80 522
....etc....
Old.St.Paul      g 103 611
Salisbury        g  84 473
> lapply(split(cathedral,cathedral$style),summary)
$g
  style        x               y
 g:16   Min.   : 45.0   Min.   :182
 r: 0   1st Qu.: 60.8   1st Qu.:299
        Median : 73.5   Median :412
        Mean   : 74.9   Mean   :397
        3rd Qu.: 86.5   3rd Qu.:481
        Max.   :103.0   Max.   :611

$r
  style       x              y
 g:0   Min.   :64.0   Min.   :344
 r:9   1st Qu.:70.0   1st Qu.:425
       Median :75.0   Median :502
       Mean   :74.4   Mean   :475
       3rd Qu.:80.0   3rd Qu.:530
       Max.   :83.0   Max.   :551
```

Now plot the data — see Figure 15.1.

```
> plot(cathedral$x,cathedral$y,type="n",xlab="Nave height",ylab="Length")
> text(cathedral$x,cathedral$y,as.character(cathedral$s))
```

Now fit the separate regression lines model. `y ~ x*style` is equivalent.

```
> g <- lm(y ~ x+style+x:style, data=cathedral)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    37.11      85.68    0.43   0.6693
x               4.81       1.11    4.32   0.0003
style         204.72     347.21    0.59   0.5617
x.style        -1.67       4.64   -0.36   0.7227

Residual standard error: 79.1 on 21 degrees of freedom
Multiple R-Squared: 0.541,      Adjusted R-squared: 0.476
F-statistic: 8.26 on 3 and 21 degrees of freedom,       p-value: 0.000807
```

Because style is non-numeric, R automatically treats it as a qualitative variables and sets up a coding - but which coding?

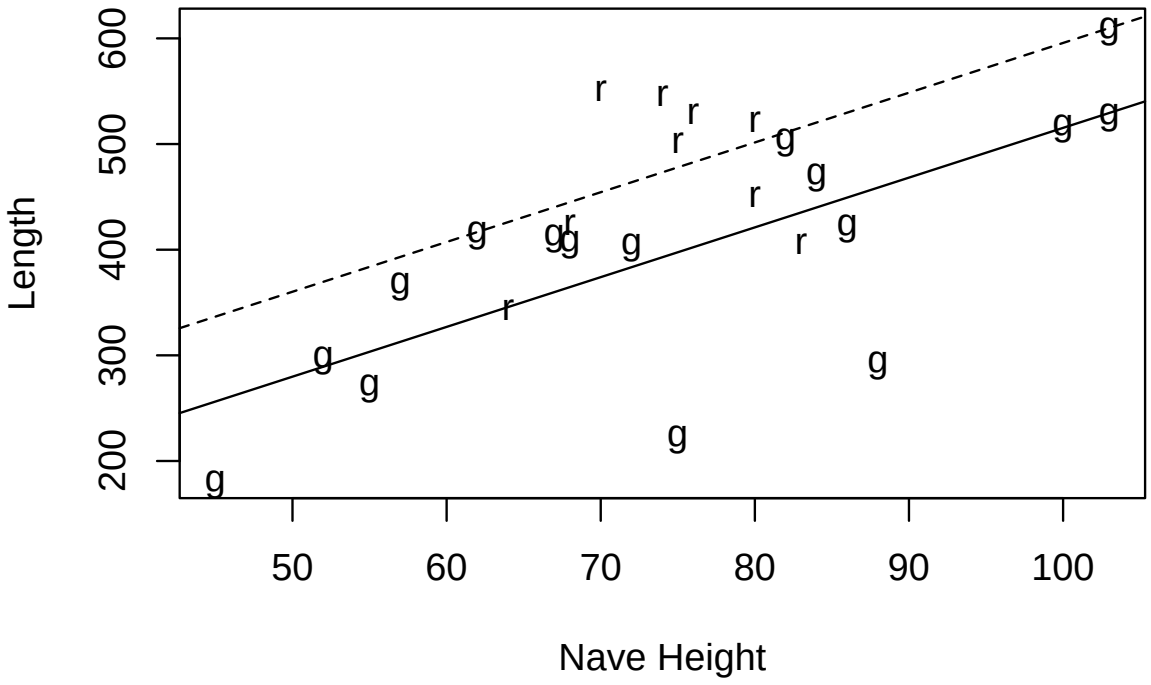

Figure 15.2: A comparison of Romanesque (r) and Gothic (g) Cathedrals

```
> model.matrix(g)
             (Intercept)   x style x.style
Durham                 1  75     1      75
Canterbury             1  80     1      80
...etc...
Old.St.Paul            1 103     0       0
Salisbury              1  84     0       0
```

We see that the model can be simplified to

```
> g <- lm(y ~ x+style, cathedral)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    44.30      81.65    0.54   0.5929
x               4.71       1.06    4.45   0.0002
style          80.39      32.31    2.49   0.0209

Residual standard error: 77.5 on 22 degrees of freedom
Multiple R-Squared: 0.538,      Adjusted R-squared: 0.496
F-statistic: 12.8 on 2 and 22 degrees of freedom,       p-value: 0.000203
```

Put the two parallel regression on the plot:

```
> abline(44.30,4.71)
> abline(44.30+80.39,4.71,lty=2)
```

A check on the diagnostics reveals no particular problems.

Our conclusion is that for cathedrals of the same height, Romanesque ones are 80.39 feet longer. For each extra foot in height, both types of cathedral are about 4.7 feet longer. Gothic cathedrals are treated as the *reference level* because "g" comes before "r" in the alphabet. We can change this:

```
> cathedral$style <- relevel(cathedral$style,ref="r")
> g <- lm(y ~ x+style, cathedral)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   124.69      82.92    1.50   0.1469
x               4.71       1.06    4.45   0.0002
style         -80.39      32.31   -2.49   0.0209

Residual standard error: 77.5 on 22 degrees of freedom
Multiple R-Squared: 0.538,      Adjusted R-squared: 0.496
F-statistic: 12.8 on 2 and 22 degrees of freedom,       p-value: 0.000203
```

Although the coefficients have different numerical values, this coding leads to the same conclusion as before.

Notice that in this case the two groups have about the same average height — about 74 feet. The difference in the lengths is 78 feet on average which is similar to the 80 feet from the fit. This is in contrast to the cholesterol example above where the two groups had very different means in their predictors.

15.2 Coding qualitative predictors

There is no unique coding for a two-level factor — there are even more choices with multi-level predictors. For a k-level predictor, $k-1$ *dummy* variables are needed for the representation. One parameter is used to represent the overall mean effect or perhaps the mean of some reference level and so only $k-1$ variables are needed rather than k.

These dummy variables cannot be exactly collinear but otherwise there is no restriction. The choice should be based on convenience.

Treatment coding

Consider a 4 level factor that will be coded using 3 dummy variables. This table describes the coding:

	Dummy coding		
	1	2	3
1	0	0	0
levels 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. R assigns levels to a factor in alphabetical order by default. The columns are orthogonal and the corresponding dummies will be too. The dummies won't be orthogonal to the intercept. Treatment coding is the default choice for R

Helmert Coding

	Dummy coding		
	1	2	3
1	-1	-1	-1
levels 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 the each other and the intercept. This coding is not so nice for interpretation. It is the default choice in S-PLUS.

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 effect the $\hat{\beta}$ and you do need to know what the coding is before making conclusions about $\hat{\beta}$.

15.3 A Three-level example

Here's an example with a qualitative predictor with more than one level. The data for this example come from a 1966 paper by Cyril Burt entitled "The genetic determination of differences in intelligence: A study of monozygotic twins reared apart". The data consist of IQ scores for identical twins, one raised by foster parents, the other by the natural parents. We also know the social class of natural parents (high, middle or low). We are interested in predicting the IQ of the twin with foster parents from the IQ of the twin with the natural parents and the social class of natural parents. Let's read in and take a look at the data:

```
> data(twins)
> twins
   Foster Biological Social
 1     82         82   high
 2     80         90   high
etc.
26    107        106    low
27     98        111    low
> plot(twins$B,twins$F,type="n",xlab="Biological IQ",ylab="Foster IQ")
> text(twins$B,twins$F,substring(as.character(twins$S),1,1))
```

See Figure 15.3 — what model seems appropriate? The most general model we'll consider is the seperate lines model:

```
> g <- lm(Foster ~ Biological*Social, twins)
> summary(g)
Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             -1.8720    17.8083   -0.11     0.92
Biological               0.9776     0.1632    5.99    6e-06
Sociallow                9.0767    24.4487    0.37     0.71
Socialmiddle             2.6881    31.6042    0.09     0.93
Biological.Sociallow    -0.0291     0.2446   -0.12     0.91
Biological.Socialmiddle -0.0050     0.3295   -0.02     0.99

Residual standard error: 7.92 on 21 degrees of freedom
```

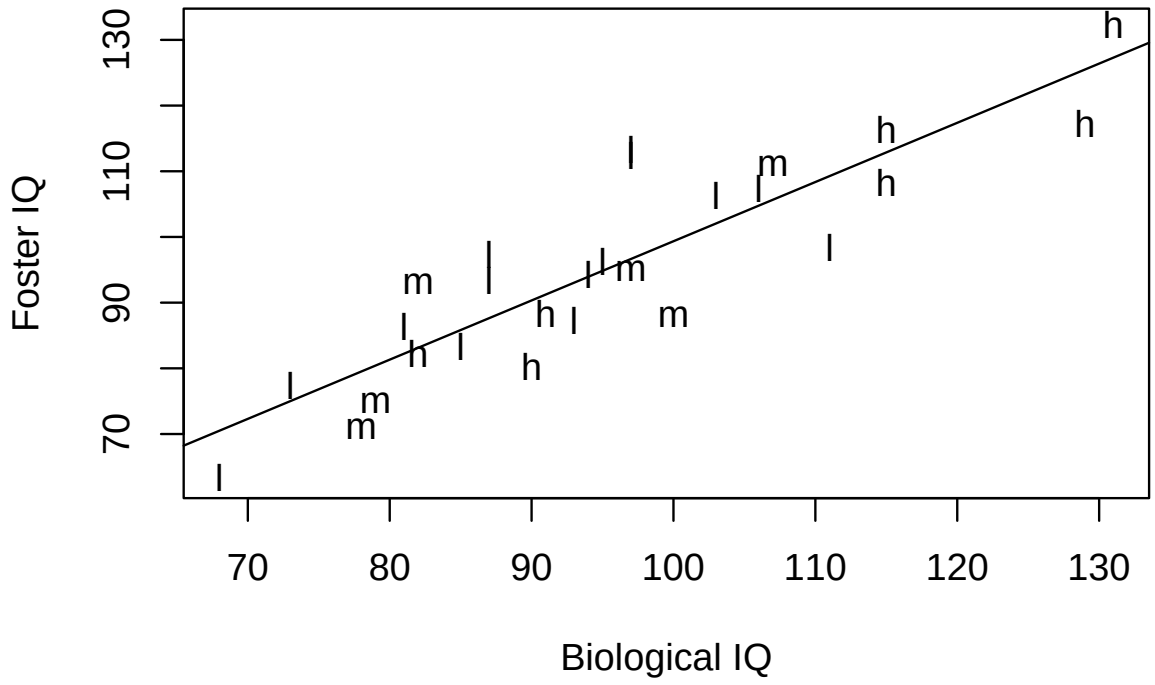

Figure 15.3: Burt twin data, l=low class, m=middle class and h= high class, regression fit shown

```
Multiple R-Squared: 0.804,      Adjusted R-squared: 0.757
F-statistic: 17.2 on 5 and 21 degrees of freedom,       p-value: 8.31e-07
```

The reference level is high class, being first alphabetically. We see that the intercept for low class line would be -1.872+9.0767 while the slope for the middle class line would be 0.9776-0.005. Check the design matrix for the gory details

```
> model.matrix(g)
```

Now see if the model can be simplified to the parallel lines model:

```
> gr <- lm(Foster ~ Biological+Social, twins)
> anova(gr,g)
Analysis of Variance Table

Model 1: Foster ~ Biological + Social
Model 2: Foster ~ Biological + Social + Biological:Social
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     23       1318
2     21       1317  2      1    0.01    1.0
```

Yes it can. The sequential testing can be done in one go:

```
> anova(g)
Analysis of Variance Table
```

```
Response: Foster
                  Df Sum Sq Mean Sq F value  Pr(>F)
Biological         1   5231    5231   83.38 9.3e-09
Social             2    175      88    1.40    0.27
Biological:Social  2      1 4.7e-01    0.01    0.99
Residuals         21   1317      63
```

We see that a further reduction to a single line model is possible:

```
> gr <- lm(Foster ~ Biological, twins)
```

Plot the regression line on the plot:

```
> abline(gr$coef)
```

A check of the diagnostics shows no cause for concern. The (almost) final model:

```
> summary(gr)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.2076     9.2999    0.99     0.33
Biological    0.9014     0.0963    9.36  1.2e-09

Residual standard error: 7.73 on 25 degrees of freedom
Multiple R-Squared: 0.778,      Adjusted R-squared: 0.769
F-statistic: 87.6 on 1 and 25 degrees of freedom,       p-value: 1.2e-09
```

The icing on the cake would be a further simplification of this model to the line y=x (the IQ's are equal). The model has no parameters at all so it has $RSS = \sum_i (y_i - x_i)^2$ and degrees of freedom equal to the sample size. We compute the F-test and p-value:

```
> sum(gr$resˆ2)
[1] 1493.5
> sum((twins$F-twins$B)ˆ2)
[1] 1557
> ((1557-1493.5)/2)/(1493.5/25)
[1] 0.53147
> 1-pf(0.53147,2,25)
[1] 0.59423
```

So the null is not rejected.

Burt was interested in demonstrating the importance of heredity over environment in intelligence and this data certainly point that way. (Although it would be helpful to know the social class of the foster parents)

However, before jumping to any conclusions, you may be interested to know that there is now considerable evidence that Cyril Burt invented some of his data on identical twins. In light of this, can you see anything in the above analysis that might lead one to suspect this data?

Chapter 16

ANOVA

Predictors are now all categorical/ qualitative. The name ANOVA stands for Analysis of Variance is used because the original thinking was to try to partition the overall variance in the response to that due to each of the factors and the error. Predictors are now typically called factors which have some number of levels. The parameters are now often called *effects*. We shall first consider only models where the parameters are considered fixed but unknown — called *fixed-effects* models but *random-effects* models are also used where parameters are taken to be random variables.

16.1 One-Way Anova

16.1.1 The model

Given a factor α occurring at $i = 1, \ldots, I$ levels, with $j = 1, \ldots, J_i$ observations per level. We use the model

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \quad i = 1, \ldots, I \quad j = 1, \ldots, J_i$$

As it stands not all the parameters are identifiable and some restriction is necessary:

1. Set $\mu = 0$ and use I different dummy variables.
2. Set $\alpha_1 = 0$ — this corresponds to treatment contrasts
3. Set $\sum_i J_i \alpha_i = 0$ which leads to the least squares estimates

$$\hat{\mu} = \bar{y}_{\cdot} \qquad \hat{\alpha}_i = \bar{y}_{i\cdot} - \bar{y}_{\cdot}$$

where $\cdot$ indicates which index or indices the mean is taken over.

This last method is the most commonly recommended for manual calculation in older textbooks although it is harder to represent within in the $y = X\beta + \varepsilon$ framework. The first two are easier to implement for computations. As usual, some preliminary graphical analysis is appropriate before fitting. A side-by-side boxplot is often the most useful plot. Look for equality of variance, transformations, outliers (influence is not relevant here since leverages won't differ unless the design is very unbalanced).

16.1.2 Estimation and testing

The effects can be estimated using direct formulae as above or by using the least squares approach (the outcome is the same). The first test of interest is whether there is a difference in the levels of the factor. We compare

- $H_0 : \alpha_i = 0 \qquad \forall i$
- H_a : at least one α_i is non zero.

We 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 the null is accepted then we are done (subject to an investigation of transformation and outliers). If we reject the null, we must investigate which levels differ.

16.1.3 An example

The example dataset we will use is a set of 24 blood coagulation times. 24 animals were randomly assigned to four different diets and the samples were taken in a random order. This data comes from Box, Hunter, and Hunter (1978).

```
> data(coagulation)
> coagulation
   coag diet
1    62    A
2    60    A
...etc...
23   63    D
24   59    D
```

The first step is to plot the data - boxplots are useful:

```
> plot(coag ~ diet, data=coagulation)
```

See the first panel of Figure 16.1.3. We are hoping *not* to see

1. Outliers — these will be apparent as separated points on the boxplots. The default is to extend the *whiskers* of the boxplot no more than one and half times the interquartiles range from the quartiles. Any points further away than this are plotted separately.
2. Skewness — this will be apparent from an asymmetrical form for the boxes.
3. Unequal variance — this will be apparent from clearly unequal box sizes. Some care is required because often there is very little data be used in the construction of the boxplots and so even when the variances truly are equal in the groups, we can expect a great deal of variability

In this case, there are no obvious problems. For group C, there are only 4 distinct observations and one is somewhat separated which accounts for the slightly odd looking plot.

Now let's fit the model.

```
> g <- lm(coag ~ diet, coagulation)
> summary(g)
Coefficients:
             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       -1.00e-14   1.45e+00 -7.4e-15  1.00000
```

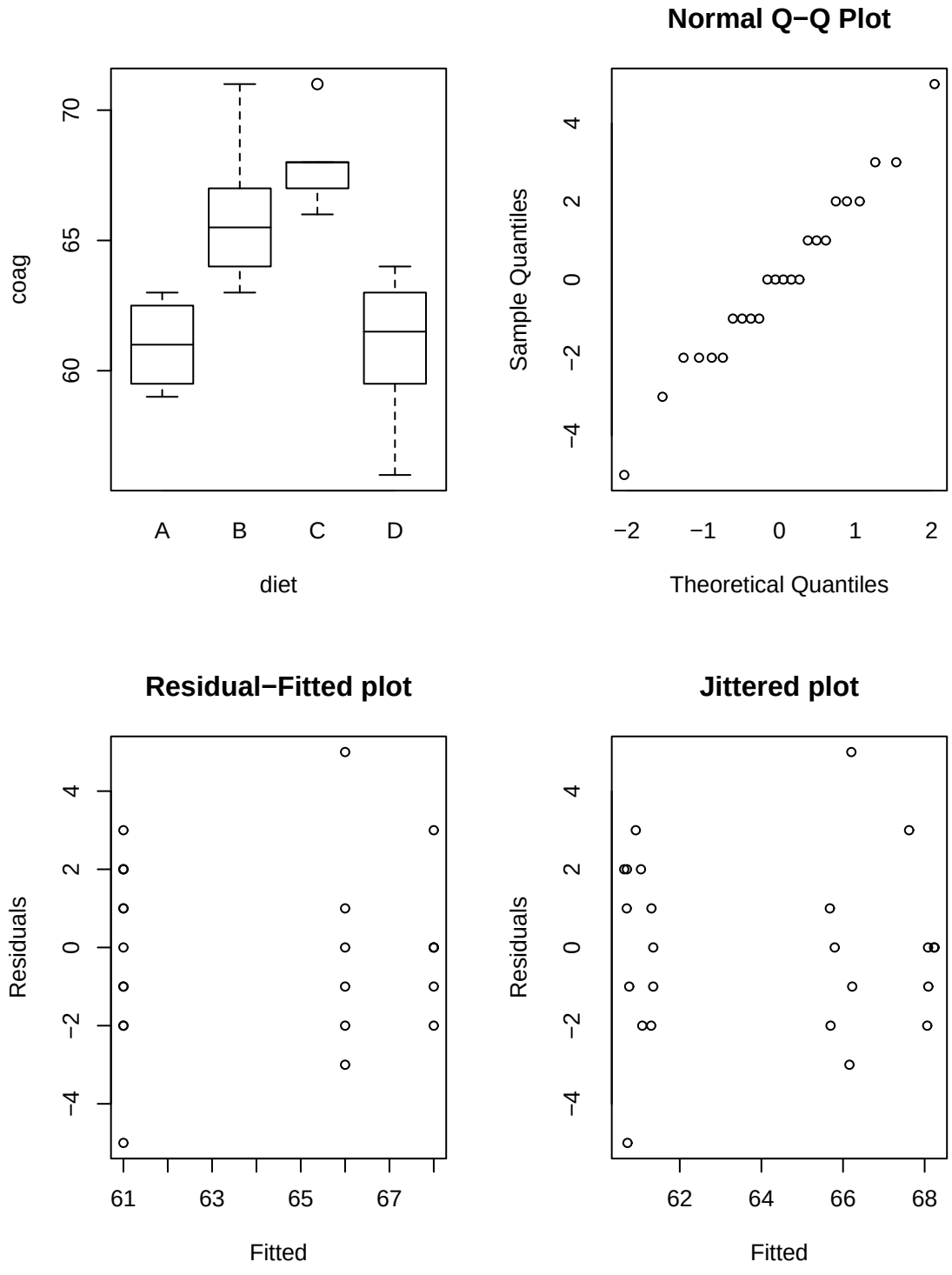

Figure 16.1: One way anova plots

```
Residual standard error: 2.37 on 20 degrees of freedom
Multiple R-Squared: 0.671,      Adjusted R-squared: 0.621
F-statistic: 13.6 on 3 and 20 degrees of freedom,       p-value: 4.66e-05
```

We conclude from the small p-value for the F-statistic that there is some difference between the groups? Group A is the reference level and has a mean of 61, groups B, C and D are 5, 7 and 0 seconds larger on average. Examine the design matrix to understand the coding:

```
> model.matrix(g)
```

We can fit the model without an intercept term as in

```
> gi <- lm(coag ~ diet -1, coagulation)
> summary(gi)
Coefficients:
      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

Residual standard error: 2.37 on 20 degrees of freedom
Multiple R-Squared: 0.999,      Adjusted R-squared: 0.999
F-statistic: 4.4e+03 on 4 and 20 degrees of freedom,    p-value:    0
```

We can directly read the level means but the tests are not useful since they involve comparisons with zero. Note the miscalculation of R^2.

16.1.4 Diagnostics

Remember to plot the residuals/fitted values and do the QQ plot. Influential points and transforming the predictors are not an issue although it is reasonable to consider transforming the response if the situation demands it.

See the last three panels of Figure 16.1.3.

```
> qqnorm(g$res)
> plot(g$fit,g$res,xlab="Fitted",ylab="Residuals",
  main="Residual-Fitted plot")
> plot(jitter(g$fit),g$res,xlab="Fitted",ylab="Residuals",
  main="Jittered plot")
```

Because the data are integers and the fitted values turn out to integers also, some discreteness is obvious in the Q-Q plot. Of course, discrete data can't be normally distributed. However, here it is approximately normal and so we can go ahead with the inference without any qualms. The discreteness in the residuals and fitted values shows up in the residual-fitted plot because we can see fewer points than the sample size. This is because of overplotting of the point symbols. There are several ways round this problem. One simple solution is to add a small amount of noise to the data. This is called *jittering*. Sometimes you have to tune the amount of noise but the default setting is adequate here.

16.1.5 Multiple Comparisons

After detecting some difference in the levels of the factor, interest centers on which levels or combinations of levels are different. Note that it does not make sense to ask whether a particular level is significant since this begs the question, "significantly different from what". Any meaningful test must involve a comparison of some kind.

It is important to ascertain whether the comparison made were decided on before or after examining the data. After fitting a model, one might decide to test only those differences that look large. To make such a decision, you also have to examine the small differences. Even if you do not actually test these small differences, it does have an effect on the inference.

If the comparisons were decided on prior to examining the data, there are three cases:

1. Just one comparison — use the standard t-based confidence intervals that we have used before.
2. Few comparisons — use the Bonferroni adjustment for the t. If there are m comparisons, use α/m for the critical value.
3. Many comparisons — Bonferroni becomes increasingly conservative as m increases. At some point it is better to use the Tukey or Scheffé or related methods described below.

It is difficult to be honest and be seen to be honest when using pre-data comparisons. Will people really believe that you only planned to make certain comparisons? Although some might make a distinction between pre and post-data comparisons, I think it is best to consider all comparisons as post-data.

If the comparisons were decided on after examining the data, you must adjust the CI to allow for the possibility of all comparisons of the type to be made.

There are two important cases:

1. Pairwise comparisons only. Use the Tukey method.
2. All contrasts i.e. linear combinations. Use the Scheffé method.

We consider pairwise comparisons first. A simple C.I. for $\alpha_i - \alpha_j$ is

$$\hat{\alpha}_i - \hat{\alpha}_j \pm t_{n-I}^{\alpha/2}\hat{\sigma}\sqrt{\frac{1}{J_i}+\frac{1}{J_j}}$$

A test for $\alpha_i = \alpha_j$ amounts to seeing whether zero lies in this interval or not. This is fine for just one test but suppose we do all possible pairwise tests when $\alpha = 5\%$ and the null hypothesis is in fact true. In Table 16.1, we see effects of multiple comparisons on the true error rates.

I	2	3	4	5	6
Nominal Type I error	5%	5%	5%	5%	5%
Actual overall Type I error	5%	12.2%	20.3%	28.6%	36.6%

Table 16.1: True error rates for multiple comparisons

We see that the true type I error can get quite high. Using the t-based CI for multiple comparisons is called least significant differences or LSD but this one is a bad trip. When comparisons are only made after the overall F-test shows a difference, it's called Fisher's LSD — this one isn't quite so bad but the type I error will still be too big.

We can use a simulation to illustrate the issues involved in multiple comparisons. Because random numbers are random, your results may differ but the message will be the same.

Suppose we have a factor of 6 levels with 4 observations per level:

```
> x <- factor(rep(LETTERS[1:6],rep(4,6)))
> x
 [1] A A A A B B B B C C C C D D D D E E E E F F F F
Levels:  A B C D E F
```

and suppose the response has no relationship to the factor (i.e. the null hypothesis holds):

```
> g <- lm(rnorm(24) ~ x)
> gs <- summary(g)
```

Here are the coefficients:

```
> g$coef
(Intercept)          xB          xC          xD          xE          xF
   0.221638    0.331200    0.058631   -0.536102    0.295339    0.067889
```

The t-statistic for testing whether level A = level B is $(\hat{\alpha}_A - \hat{\alpha}_B)/se(\hat{\alpha}_A - \hat{\alpha}_B)$ where $se((\hat{\alpha}_A - \hat{\alpha}_B)) = \hat{\sigma}\sqrt{1/4+1/4} = \hat{\sigma}/\sqrt{(2)}$

```
> g$coef[2]*sqrt(2)/gs$sig
     xB
0.41881
```

This would (in absolute value) need exceed this t-critical value for significance at the 5% level:

```
> qt(0.975,24-6)
[1] 2.1009
```

Out of all the possible pairwise comparisons, we may compute the maximum t-statistic as

```
> range(c(0,g$coef[-1]))
[1] -0.5361  0.3312
> rg <- range(c(0,g$coef[-1]))
> (rg[2]-rg[1])*sqrt(2)/gs$sig
[1] 1.0967
```

which just fails to meet significance. Now let's repeat the experiment 1000 times.

```
> res <- matrix(0,1000,2)
> for(i in 1:1000){
g <- lm(rnorm(24) ~ x)
gs <- summary(g)
res[i,1] <- abs(g$coef[2]*sqrt(2)/gs$sig)
rg <- range(c(0,g$coef[-1]))
res[i,2] <- (rg[2]-rg[1])*sqrt(2)/gs$sig
}
```

Now see how many of the test statistics for comparing level A and level B were significant at the 5% level:

```
> sum(res[,1] > 2.1)/1000
[1] 0.045
```

Just a shade under the 5% it should be. Now see how many times the maximum difference is significant at the 5% level.

```
> sum(res[,2] > 2.1)/1000
[1] 0.306
```

About 1/3 of the time. So in cases where there is no difference between levels of the factor, about 1/3 of the time, an investigator will find a statistically significant difference between some levels of the factor. Clearly there is a big danger that one might conclude there is a difference when none truly exists.

We need to make the critical value larger so that the null is rejected only 5% of the time. Using the simulation results we estimate this value to be:

```
> quantile(res[,2],0.95)
   95%
3.1627
```

It turns out that this value may be calculated using the ”Studentized Range distribution”:

```
> qtukey(0.95,6,18)/sqrt(2)
[1] 3.1780
```

which is close to the simulated value.

Now let’s take a look at the densities of our simulated t-statistics:

```
> dmax <- density(res[,2],from=0,to=5)
> d2 <- density(res[,1],from=0,to=5)
> matplot(d2$x,cbind(dmax$y,d2$y),type="l",xlab="Test statistic",
  ylab="Density")
> abline(h=0)
> abline(v=2.1,lty=2)
> abline(v=3.178)
```

We see the result in Figure 16.2. We see that the distribution of the maximum t-statistic has a much heavier tail than the distribution for a prespecified difference. The true density for the prespecified difference is a the upper half of a t-distribution — the maximum in the estimated distribution does not occur at zero because boundary error effects in the density estimator.

Now we return to our real data. We’ve found that there is a significant difference among the diets but which diets can be said to be different and which diets are not distinguishable. Let’s do the calculations for the difference between diet B and diet C which is 2. First we do the LSD calculation:

```
> qt(1-.05/2,20)*2.366*sqrt(1/6+1/6)
[1] 2.8494
> c(2-2.85,2+2.85)
[1] -0.85  4.85
```

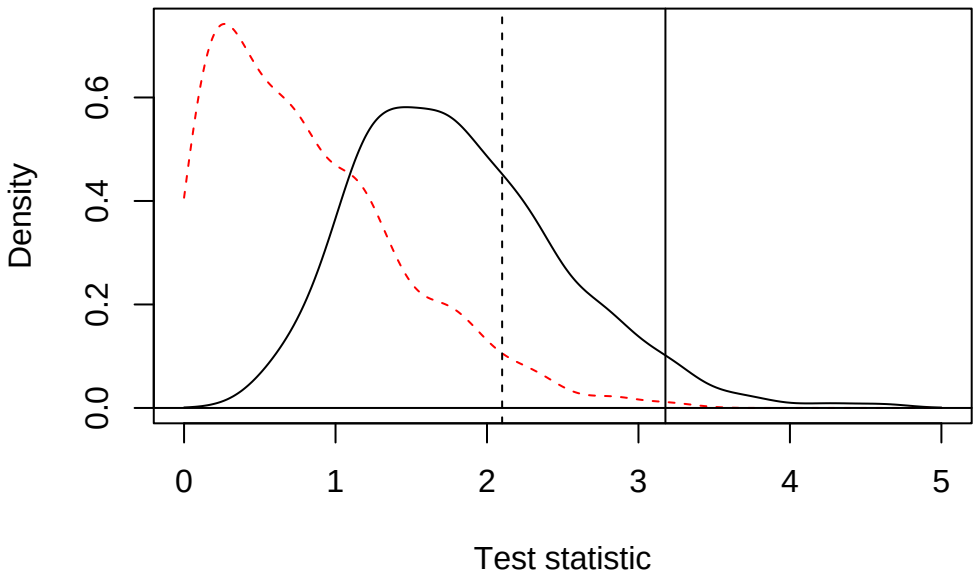

Figure 16.2: Estimated densities of the maximum t-statistic (solid line) and t-statistic for a prespecified difference (dashed line). The corresponding theoretical 95% quantiles are marked with vertical lines

An alternative way we can make the same calculation is to recode the factor with B as the reference level and refit the model:

```
> coagulation$diet <- relevel(coagulation$diet,ref="B")
> g <- lm(coag ~ diet, coagulation)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   66.000      0.966   68.32  < 2e-16
dietA         -5.000      1.528   -3.27  0.00380
dietC          2.000      1.366    1.46  0.15878
dietD         -5.000      1.278   -3.91  0.00086

Residual standard error: 2.37 on 20 degrees of freedom
Multiple R-Squared: 0.671,      Adjusted R-squared: 0.621
F-statistic: 13.6 on 3 and 20 degrees of freedom,       p-value: 4.66e-05
```

We can read the B vs. C difference directly from the output now as 2 and compute the width of the confidence band using the corresponding standard error:

```
> qt(0.975,20)*1.366
[1] 2.8494
```

We can verify that the standard error of 1.366 can also be obtained directly as

```
> 2.366*sqrt(1/6+1/6)
[1] 1.366
```

As can be seen, the result is the same as before.

Suppose two comparisons were pre-planned, then critical value is now this, using the Bonferroni correction.

```
> qt(1-.05/4,20)*2.37*sqrt(1/6+1/6)
[1] 3.3156
> c(2-3.32,2+3.32)
[1] -1.32  5.32
```

Tukey's Honest Significant Difference (HSD) is designed for all pairwise comparisons and depends on the studentized range distribution. Let $X_1, \ldots, X_n$ 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 $q_{n,\nu}$ where ν is the number of degrees of freedom used in estimating σ.

The Tukey C.I.'s are

$$\hat{\alpha}_i - \hat{\alpha}_j \pm q_{I,n-I} \frac{\hat{\sigma}}{\sqrt{2}} \sqrt{\frac{1}{J_i} + \frac{1}{J_j}}$$

When the sample sizes J_i are very unequal, Tukey's HSD may be too conservative but in general they are narrower than those produced by Scheffé's theorem. There are several other methods for multiple comparisons — the Tukey method tends to be more conservative than most because it takes the rather pessimistic approach based on the maximum difference. Not all the differences will be as large as the maximum and so some competing methods take advantage of this to get tighter intervals.

For future reference, a more general form for the Tukey intervals is

$$(\text{difference}) \pm (q_{l,df}/\sqrt{2}) \times (\text{se of difference})$$

where l is the number of levels of the factor on which we are making multiple comparisons and df is the degrees of freedom for the error.

We compute the Tukey HSD bands for the diet data. First we need the critical value from the studentized range distribution.

```
> qtukey(0.95,4,20)
[1] 3.9583
```

and the interval is:

```
> (3.96/sqrt(2))*2.37*sqrt(1/6+1/6)
[1] 3.8315
> c(2-3.83,2+3.83)
[1] -1.83  5.83
```

A convenient way to obtain all the intervals is

```
> TukeyHSD(aov(coag ~ diet, coagulation))
  Tukey multiple comparisons of means
    95% family-wise confidence level

$diet
           diff      lwr      upr
```

```
A-B -5.0000e+00  -9.2754 -0.72455
C-B  2.0000e+00  -1.8241  5.82407
D-B -5.0000e+00  -8.5771 -1.42291
C-A  7.0000e+00   2.7246 11.27545
D-A -1.4211e-14  -4.0560  4.05604
D-C -7.0000e+00 -10.5771 -3.42291
```

The Bonferroni based bands would have been just slightly wider:

```
> qt(1-.05/12,20)*2.37*sqrt(1/6+1/6)
[1] 4.0052
```

We divide by 12 here because there are 6 possible pairwise differences and we want a two-sided confidence interval: $6 \times 2 = 12$. With a bit of work we find that only the A-D and B-C differences are not significant.

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. For a detailed description of the many availabe alternatives see Hsu (1996). Some other pairwise comparison tests may be found in the R package `ctest`.

16.1.6 Contrasts

A contrast among the effects $\alpha_1, \ldots, \alpha_I$ is a linear combination $\sum_i c_i \alpha_i$ where the c_i are known and $\sum_i c_i = 0$. For example

1. $\alpha_1 - \alpha_2$ is a contrast with $c_1 = 1, c_2 = -1$ and the other $c_i = 0$. All pairwise differences are contrasts.
2. $(\alpha_1 + \alpha_2)/2 - (\alpha_3 + \alpha_4)/2$ with $c_1 = c_2 = 1/2$ and $c_3 = c_4 = -1/2$ and the other $c_i = 0$. This contrast is directly interpretable.

16.1.7 Scheffé's theorem for multiple comparisons

An estimable function of the parameters is one that can be estimated given the data we have. More precisely, a linear combination $\psi = c^T\beta$ is estimable if there exists an $a^T y$ such that $Ea^T y = c^T\beta$. Contrasts are estimable but something like α_i is not because it will depend on the coding used. Now $\hat{\psi} = a^T y$ and $\text{var}\,\hat{\psi} = \sigma^2 a^T a$ which can be estimated by $\hat{\sigma}^2 a^T a$. Suppose we let the dimension of the space of possible c be q and the $\text{rank}(X) = r$. ($r = p$ if we have complete identifiability.)

Scheffé's theorem

A $100(1-\alpha)\%$ simultaneous confidence interval for all estimable ψ is

$$\hat{\psi} \pm \sqrt{qF^{\alpha}_{q,n-r}}\sqrt{\hat{\text{var}}\,\hat{\psi}}$$

Example: Simultaneous confidence interval for the regression surface:

$$x^T\hat{\beta} \pm \sqrt{pF^{\alpha}_{p,n-p}}\hat{\sigma}\sqrt{x^T(X^TX)^{-1}x}$$

We can illustrate this with corrosion data used in the lack of fit chapter. We compute the usual t-based pointwise bands along with the simultaneous Scheffé bands:

```
> data(corrosion)
> gf <- lm(loss ~ Fe, corrosion)
> grid <- seq(0,3,by=0.1)
> p <- predict(gf,data.frame(Fe=grid),se=T)
> fmult <- sqrt(2*qf(0.95,2,11))
> tmult <- qt(0.975,11)
> matplot(grid,cbind(p$fit,p$fit-fmult*p$se,p$fit+fmult*p$se,
  p$fit-tmult*p$se,p$fit+tmult*p$se),type="l",lty=c(1,2,2,5,5),
  ylab="loss",xlab="Iron Content")
> points(corrosion$Fe,corrosion$loss)
```

The plot is shown in Figure 16.3. The bands form a 95% simultaneous confidence region for the true regression line. These bands are slightly wider than the t-based pointwise confidence bands described in Chapter 3. This is because they hold over the whole real line and not just a single point.

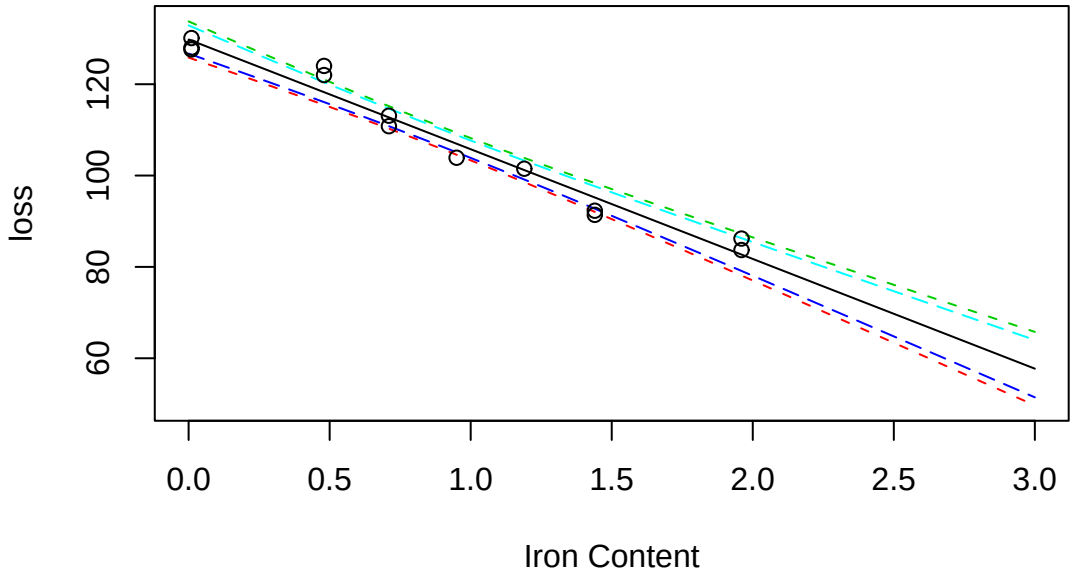

Figure 16.3: Scheffé 95% simultaneous confidence bands are shown as dotted lines surrounding the least squares fit. The interior (dashed) lines represent the pointwise confidence intervals.

Example: One-way anova: Consider $\psi = \sum_i c_i \alpha_i$, a contrast which is therefore estimable. We compute

$$\hat{\text{var}}\ \hat{\psi} = \hat{\sigma}^2 \sum_{i=1}^{I} \frac{c_i^2}{J_i}$$

and the SCI for ψ is then

$$\sum_i c_i \hat{\alpha}_i \pm \sqrt{(I-1)F^{\alpha}_{I-1,n-I}}\hat{\sigma}\sqrt{\sum_{i=1}^{I}\frac{c_i^2}{J_i}}$$

Here we apply the Scheffé method for (B+C)/2 -(A+D)/2 so that $c_2 = c_3 = 1/2$ and $c_1 = c_4 = -1/2$

```
> sqrt(3*qf(0.95,3,20))*2.37*sqrt(1/4+1/6+1/6+1/8)/2
[1] 3.0406
```

```
> (5+7)/2-(0+0)/2
[1] 6
> c(6-3.04,6+3.04)
[1] 2.96 9.04
```

We see that this difference is significantly different from 0 and so we may conclude that there is significant difference between the average of B and C and the average of A and D despite the fact that we may have chosen to test this difference after seeing the data.

16.1.8 Testing for homogeneity of variance

This can be done using Levene's test. Simply compute the absolute values of the residuals and use these as the response in a new one-way anova. A significant difference would indicate non constant variance.

There are other tests but this one is quite insensitive to non-normality and is simple to execute. Most tests and CI's 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:

```
> summary(lm( abs(g$res) ~ coagulation$diet))
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)          1.500      0.716    2.10    0.049
coagulation$dietB    0.500      0.924    0.54    0.594
coagulation$dietC   -0.500      0.924   -0.54    0.594
coagulation$dietD    0.500      0.877    0.57    0.575

Residual standard error: 1.43 on 20 degrees of freedom
Multiple R-Squared: 0.0956,     Adjusted R-squared: -0.0401
F-statistic: 0.705 on 3 and 20 degrees of freedom,      p-value: 0.56
```

Since the p-value is large, we conclude that there is no evidence of a non-constant variance.

16.2 Two-Way Anova

Suppose we have two factors, α at I levels and β at J levels. Let n_{ij} be the number of observations at level i of α and level j of β and let those observations be $y_{ij1}, y_{ij2}, \ldots$. A complete layout has $n_{ij} \geq 1$ for all i, j. The most general model that may be considered is

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

The interaction effect $(\alpha\beta)_{ij}$ is interpreted as that part of the mean response not attributable to the additive effect of α_i and β_j. For example, you may enjoy strawberries and cream individually, but the combination is superior. In contrast, you may like fish and ice cream but not together.

A balanced layout requires that $n_{ij} = n$. Not all the parameters are identifiable but if the main effects α and β are coded appropriately and the interaction effects coding is then derived from the product of these codings, then every contrast of parameters can be estimated.

16.2.1 One observation per cell

When $n_{ij} = 1$ we would have as many observation as parameters if we tried to fit the full model as above. The parameters could be estimated but no further inference would be possible.

We can assume $(\alpha\beta)_{ij} = 0$ to free up degrees of freedom to make some tests and CI's. This assumption can be checked graphically using an interaction plot - plot the cell means on the vertical axis and the factor α on the horizontal. Join points with same level of β. The role of α and β can be reversed. Parallel lines on the plot are a sign of a lack of interaction. Tukey's non-additivity test provides another way of investigating 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 and that it makes the assumption that the interaction effect is multiplicative in a form which seems somewhat tenuous.

Barring any trouble with interaction, because of the balanced design, the factors are orthogonal and their significance can be tested in the usual way.

16.2.2 More than one observation per cell

When $n_{ij} = n$ i.e. the same number of observations per cell, we have orthogonality. Orthogonality can also occur if the row/column cell numbers are proportional. Orthogonality is desirable and experiments are usually designed to ensure it.

With more than one observation per cell we are now free to fit and test the model:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

The interaction effect may be tested by comparison to the model

$$y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ijk}$$

and computing the usual F-test. If the interaction effect is found to be significant, do not test the main effects even if they appear not to be significant. The estimation of the main effects and their significance is coding dependent when interactions are included in the model.

If the interaction effect is found to be insignificant, then test the main effects but use RSS/df from the full model in the denominator of the F-tests — this has been shown to maintain the type I error better. So the F-statistic used is

$$\frac{F = RSS_{small} - RSS_{large}/(df_{small} - df_{large})}{\hat{\sigma}^2_{full}}$$

16.2.3 Interpreting the interaction effect

No interactions You can do pairwise comparisons on α without regard to β and vice versa.

Interaction present A comparison of the levels of α will depend on the level of β. Interpretation is not simple. Consider the following two layouts of $\hat{\mu}_{ij}$ in a 2x2 case:

	Male	Female	Male	Female
drug 1	3	5	2	1
drug 2	1	2	1	2

The response is a measure of performance. In the case on the left, we can say that drug 1 is better than drug 2 although the interaction means that its superiority over drug 2 depends on the gender. In the case on the right, which drug is best depends on the gender. We can also plot this as in Figure 16.4. We see that neither case are the lines parallel indicating interaction but the superiority of drug 1 is clear in the first plot and the ambiguous conclusion is clear in the second. I recommend making plots like this when you want to understand an interaction effect.

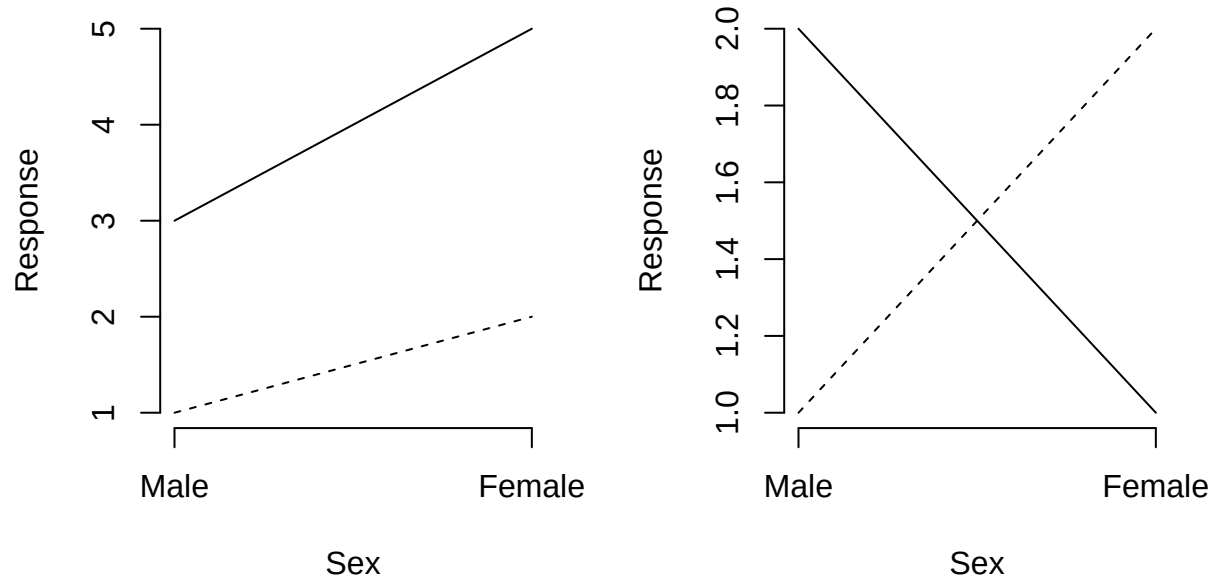

Figure 16.4: Two 2x2 tables with the response plotted by the factors, sex on the horizontal axis and drug 1 as the solid line and drug 2 as the dotted line.

When the interaction is significant, the main effects cannot be defined in an obvious and universal way. For example, we could the gender effect as the effect for females, the effect for males, the effect for the average males and females or something else. If there was no interaction effect, the gender effect could be defined unambiguously.

When you have a significant inteaction, you can fit a model

$$y_{ijk} = \mu_{ijk} + \varepsilon_{ijk}$$

and then treat the data as a one-way anova with IJ levels. Obviously this makes for more complex comparisons but this is unavoidable when interactions exist.

Here is a two-way anova design where there are 4 replicates. As part of an investigation of toxic agents, 48 rats were allocated to 3 poisons (I,II,III) and 4 treatments (A,B,C,D). The response was survival time in tens of hours. The Data:

```
          A       B       C       D
I         0.31    0.82    0.43    0.45
          0.45    1.10    0.45    0.71
          0.46    0.88    0.63    0.66
          0.43    0.72    0.76    0.62

II        0.36    0.92    0.44    0.56
```

```
         0.29    0.61    0.35    1.02
         0.40    0.49    0.31    0.71
         0.23    1.24    0.40    0.38

III      0.22    0.30    0.23    0.30
         0.21    0.37    0.25    0.36
         0.18    0.38    0.24    0.31
         0.23    0.29    0.22    0.33
```

We make some plots:

```
> data(rats)
> plot(time ~ treat + poison, data=rats)
```

Some evidence of skewness can be seen, especially since it appears that variance is in some way related to the mean response. We now check for an interaction using graphical methods:

```
> interaction.plot(rats$treat,rats$poison,rats$time)
> interaction.plot(rats$poison,rats$treat,rats$time)
```

Do these look parallel? The trouble with interaction plots is that we expect there to be some random variation regardless so it is difficult to distinguish true interaction from just noise. Fortunately, in this case, we have replication so we can directly test for an interaction effect.

Now fit the full model and see the significance of the factors:

```
> g <- lm(time ~ poison*treat, rats)
> anova(g)
Analysis of Variance Table

Response: time
             Df Sum Sq Mean Sq F value  Pr(>F)
poison        2  1.033   0.517   23.22 3.3e-07
treat         3  0.921   0.307   13.81 3.8e-06
poison:treat  6  0.250   0.042    1.87    0.11
Residuals    36  0.801   0.022
```

We see that the interaction effect is not significant but the main effects are. We check the diagnostics:

```
> qqnorm(g$res)
> plot(g$fitted,g$res,xlab="Fitted",ylab="Residuals")
```

Clearly there's a problem - perhaps transforming the data will help. Try logs first:

```
> g <- lm(log(time) ~ poison*treat,rats)
> plot(g$fitted,g$res,xlab="Fitted",ylab="Residuals",main="Log response")
```

Not enough so try the reciprocal:

```
> g <- lm(1/time ~ poison*treat,rats)
> plot(g$fitted,g$res,xlab="Fitted",ylab="Residuals",
  main="Reciprocal response")
```

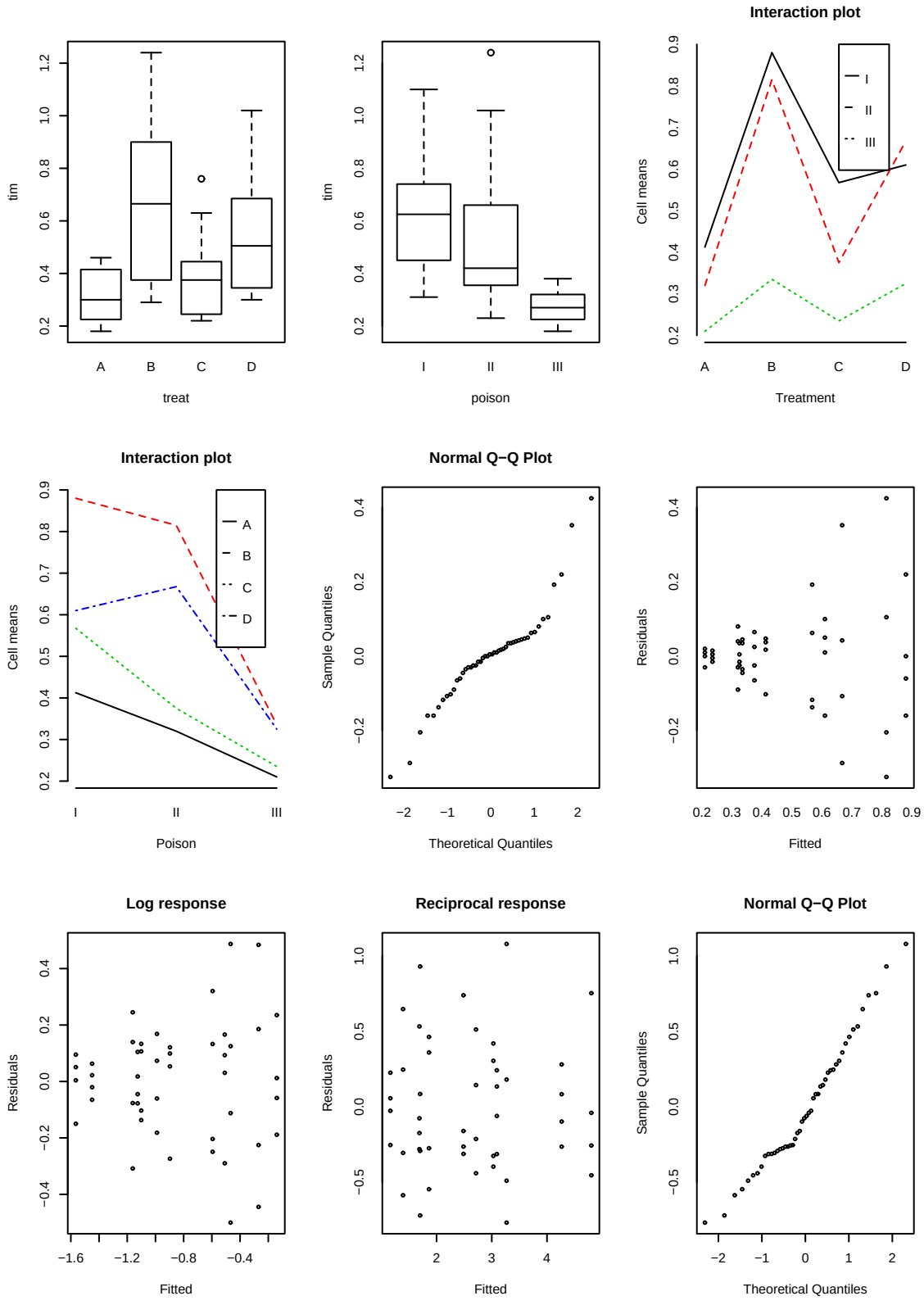

Figure 16.5: Two way anova plots

Looks good - the reciprocal can be interpreted as the rate of dying. Better check the Q-Q plot again:

```
> qqnorm(g$res)
```

This looks better than the first Q-Q plot. We now check the ANOVA table again, find the interaction is not significant, simplify the model and examine the fit:

```
> anova(g)
Analysis of Variance Table

Response: 1/time
             Df Sum Sq Mean Sq F value  Pr(>F)
poison        2   34.9    17.4   72.63 2.3e-13
treat         3   20.4     6.8   28.34 1.4e-09
poison:treat  6    1.6     0.3    1.09    0.39
Residuals    36    8.6     0.2

> g <- lm(1/time ~ poison+treat, rats)
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.698      0.174   15.47  < 2e-16
poisonII       0.469      0.174    2.69   0.0103
poisonIII      1.996      0.174   11.45  1.7e-14
treatB        -1.657      0.201   -8.23  2.7e-10
treatC        -0.572      0.201   -2.84   0.0069
treatD        -1.358      0.201   -6.75  3.3e-08

Residual standard error: 0.493 on 42 degrees of freedom
Multiple R-Squared: 0.844,      Adjusted R-squared: 0.826
F-statistic: 45.5 on 5 and 42 degrees of freedom,       p-value: 6.66e-16
```

Let's construct pairwise confidence intervals for the treatment factor using the Tukey method. Because of the balance, the CI's will all be the same width. First the standard error for a pairwise difference may be obtained from the output as 0.201. We then compute the width of the interval as

```
> qtukey(0.95,4,42)*0.201/sqrt(2)
[1] 0.53767
```

So the bands will be difference plus or minus 0.54. All the bands except the B-D do not include 0 so we can conclude that all these other pairs of treatments are significantly different. The treatment reduces the rats survival time the most is A since it is the reference level and all other treatments reduce the response (rate of dying). Can you distinguish between the poisons?

16.2.4 Replication

It's important that the observations observed 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, imagine that the experiment above involved the reaction times of human subjects under two factors. We need to distinguish between an experiment that uses 48 subjects and one that uses 12 subjects where each subject repeats their assigned factor combination 4 times. In the latter case, the responses will not be independent and a repeated measures style of analysis will be necessary.

16.3 Blocking designs

In completely randomized designs (CRD) like the one and two-way anova, the treatments are assigned to the experimental units at random. This is appropriate when the units are homogenous. Sometimes, we may suspect that the units are heterogenous, but we can't describe the form it takes - for example, we may know 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 reduce bias, but the real justification lies in the randomization test discussed earlier for regression. The usual testing argument may be applied. 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. The normal-based inference is much quicker so we might prefer to use that.

When the experimental units are heterogenous in a known way and can be arranged into *blocks* where the intrablock variation is ideally small but the interblock variation is large, a *block design* can be more efficient than a CRD.

The contrast between the two designs is shown in Figure 16.6.

Examples:

Suppose we want to compare 4 treatments and have 20 patients available. We might be able divide the patients in 5 blocks of 4 patients each where the patients in each block have some relevant similarity. We would then randomly assign the treatments within each block.

Suppose we want to test 3 crop varieties on 5 fields. Divide each field into 3 strips and randomly assign the crop variety.

Note: We prefer to have block size equal to the number of treatments. If this is not done or possible, an *incomplete* block design must be used.

Notice that under the randomized block design the randomization used in assigning the treatments to the units is restricted relative to the full randomization used in the CRD. This has consequences for the inference.

16.3.1 Randomized Block design

We have one factor (or treatment) at t levels and one blocking variable at r levels. The model is

$$y_{ij} = \mu + \tau_i + \rho_j + \varepsilon_{ij}$$

The analysis is then very similar to the two-way anova with one observation per cell. We can check for interaction and 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

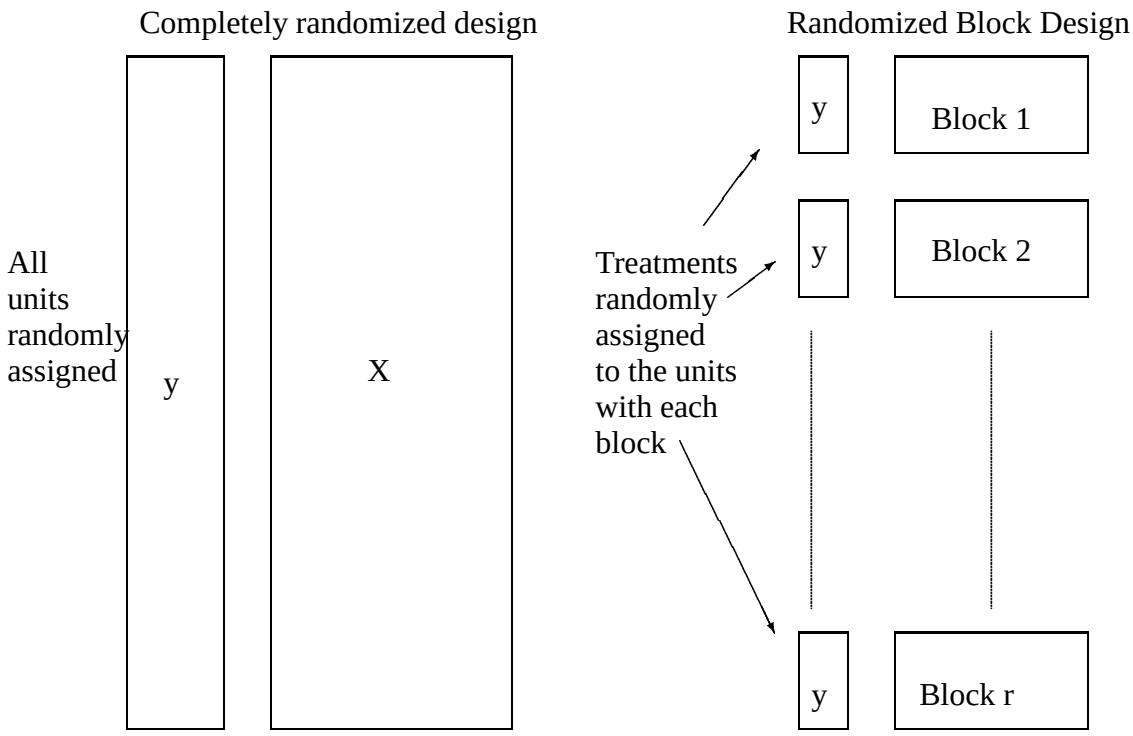

Figure 16.6: Completely randomized design vs. Randomized Block design

the treatments. This means that we cannot regain the degrees of freedom devoted to blocking even if the blocking effect turns out not to be significant. The randomization test-based argument means that we must judge the magnitude of the treatment effect within the context of the restricted randomization that has been used.

We illustrate with an experiment to compare 4 processes, A,B,C,D for the production of penicillin. These are the treatments. The raw material, corn steep liquor, is quite variable and can only be made in blends sufficient for 4 runs. Thus a randomized complete block design is definitely suggested by the nature of the experimental units. The data is:

	A	B	C	D
Blend 1	89	88	97	94
Blend 2	84	77	92	79
Blend 3	81	87	87	85
Blend 4	87	92	89	84
Blend 5	79	81	80	88

We start with some graphical checks:

```
> data(penicillin)
> plot(yield ~ blend+treat,data=penicillin)
```

See the first two panels of Figure 16.3.1

Did you see any problems? Now check for interactions:

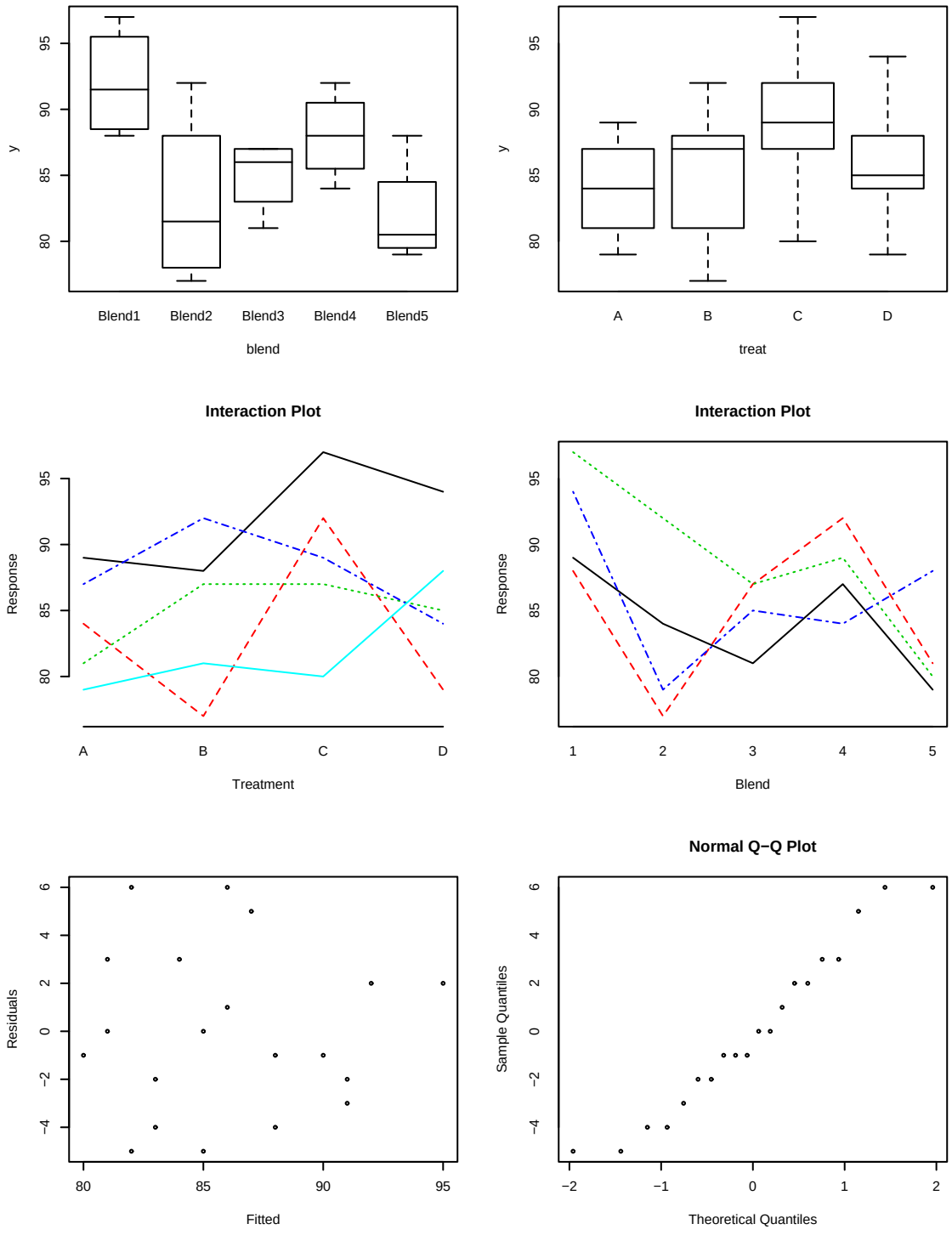

Figure 16.7: RCBD plots for the penicillin data

```
> interaction.plot(penicillin$treat,penicillin$blend,penicillin$yield)
> interaction.plot(penicillin$blend,penicillin$treat,penicillin$yield)
```

What do you think? It is hard to tell — interaction plots are only suggestive, not definitive.

Regardless, we now fit the model:

```
> g <- lm(yield ~ treat+blend,penicillin)
> anova(g)
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3   70.0    23.3    1.24  0.339
blend      4  264.0    66.0    3.50  0.041
Residuals 12  226.0    18.8
```

We see no significant treatment effect but the block effect is, as suspected, significant. The analysis of variance table corresponds to a sequential testing of models, here corresponding to the sequence

```
y ~ 1
y ~ treat
y ~ treat + blend
```

So here the p-value 0.339 corresponds to a comparison of the first two models in this list, while the p-value of 0.041 corresponds to the test comparing the second two. One small point to note is that the denominator in both F-test is the mean square from the full model, here 18.8

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 ~ blend+treat,penicillin))
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
blend      4  264.0    66.0    3.50  0.041
treat      3   70.0    23.3    1.24  0.339
Residuals 12  226.0    18.8
```

By way of comparison, see what happens if we omit the first observation in the dataset — this might happen in practice if this run was lost:

```
> anova(lm(yield ~ blend+treat,penicillin[-1,]))
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
blend      4  266.5    66.6    3.27  0.054
treat      3   59.7    19.9    0.98  0.439
Residuals 11  224.3    20.4
```

```
> anova(lm(yield ~ treat+blend,penicillin[-1,]))
Analysis of Variance Table

Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3   91.8    30.6    1.50  0.269
blend      4  234.4    58.6    2.87  0.075
Residuals 11  224.3    20.4
```

Notice that now the order does matter. If we want to test for a treatment effect, we would prefer the first table since in that version the blocking factor `blend` 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.

Check the diagnostics:

```
> plot(g$fitted,g$res,xlab="Fitted",ylab="Residuals")
> qqnorm(g$res)
```

And that might be the end of the story except for that worrying interaction effect possibility. We execute the Tukey non-additivity test:

```
> summary(g)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    90.00       2.74   32.79  4.1e-13
treatB          1.00       2.74    0.36   0.7219
treatC          5.00       2.74    1.82   0.0935
treatD          2.00       2.74    0.73   0.4802
blendBlend2    -9.00       3.07   -2.93   0.0125
blendBlend3    -7.00       3.07   -2.28   0.0416
blendBlend4    -4.00       3.07   -1.30   0.2169
blendBlend5   -10.00       3.07   -3.26   0.0068

Residual standard error: 4.34 on 12 degrees of freedom
Multiple R-Squared: 0.596,      Adjusted R-squared: 0.361
F-statistic: 2.53 on 7 and 12 degrees of freedom,       p-value: 0.0754

> alpha <- c(0,g$coef[2:4])
> alpha
       treatB treatC treatD
     0      1      5      2
> beta <- c(0,g$coef[5:8])
> beta
            blendBlend2 blendBlend3 blendBlend4 blendBlend5
          0          -9          -7          -4         -10
> ab <- rep(alpha,5)*rep(beta,rep(4,5))
> h <- update(g,.~.+ab)
> anova(h)
Analysis of Variance Table
```

```
Response: yield
          Df Sum Sq Mean Sq F value Pr(>F)
treat      3   70.0    23.3    1.15  0.374
blend      4  264.0    66.0    3.24  0.055
ab         1    2.0     2.0    0.10  0.760
Residuals 11  224.0    20.4
```

Because the p-value of the treat times block effect is .76 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.

We can do multiple comparisons for the treatment effects, using the Tukey or the Scheffé method as appropriate:

$$\hat{\tau}_i - \hat{\tau}_j \pm q_{t,(t-1)(r-1)}\hat{\sigma}/\sqrt{r}$$

or

$$\sum_i c_i\hat{\tau}_i \pm \sqrt{(t-1)F_{t-1,(t-1)(r-1)}}\hat{\sigma}\sqrt{\sum_i c_i^2/r}$$

Now, just for the sake of the example, we compute the Tukey pairwise confidence intervals for the treatment effects:

```
> qtukey(0.95,4,12)
[1] 4.1987
```

The standard errors for the differences are

```
> 4.34*sqrt(1/5+1/5)
[1] 2.7449
```

Can you find this in the output above? The bands are difference plus or minus this:

```
> 4.2*2.745/sqrt(2)
[1] 8.1522
```

How does this compare with the observed difference in the treatment effects?

16.3.2 Relative advantage of RCBD over CRD

We can measure precision by considering var $\hat{\tau}$ (or equivalently $\hat{\sigma}^2$). Compare the $\hat{\sigma}^2$ for designs with the same sample size. We define *relative efficiency* as

$$\frac{\hat{\sigma}^2_{CRD}}{\hat{\sigma}^2_{RCBD}}$$

where the quantities can be computed by fitting models with and without the blocking effect. For example, suppose $\hat{\sigma}^2_{CRD} = (226+264)/(12+4) = 30.6$ and $\hat{\sigma}^2_{RCBD} = 18.8$, as it is in the example above, then the relative efficiency is 1.62. The $\hat{\sigma}^2_{CRD}$ numbers come from combining the sums of squares and degrees of freedom for the residuals and the blend in the first anova table we made for this data. An alternative method would be to simply fit the model `yield ~ treat` and read off $\hat{\sigma}_{CRD}$ from the output.

The interpretation is that a CRD would require 62% more observations to obtain the same level of precision as a 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.

16.4 Latin Squares

These are 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 4 production methods (the treatment) — A, B, C, and D. We have available 4 machines 1, 2, 3, and 4, and 4 operators, I, II, III, IV. A Latin square design is

	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 16.2: Latin Square

- Each treatment is assigned to each block once and only once.
- The design and assignment of treatments and blocks should be random.

We use the model

$$y_{ijk} = \mu + \tau_i + \beta_j + \gamma_k + \varepsilon_{ijk} \qquad i, j, k = 1, \ldots, t$$

To test for a treatment effect simply fit a model without the treatment effect and compare using the F-test. The Tukey pairwise CI's are

$$\hat{\tau}_l - \hat{\tau}_m \pm q_{t,(t-1)(t-2)} \hat{\sigma} \sqrt{1/t}$$

- The Latin square can be even more efficient than the RCBD provided that the blocking effects are sizable.
- 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.
- The Latin square can be used for comparing 3 treatment factors. Only t^2 runs are required compared to the t^3 required if all combinations were run. (The downside is that you can't estimate the interactions if they exist). This is an example of a *fractional factorial*.
- The Latin square can be replicated if more runs are available.
- When there are 3 blocking variables, a Graeco-Latin square may be used but these rarely arise in practice.

An engineer wants to compare the qualities of raw materials from four suppliers, A, B, C, D. The raw material is used to produce a component whose breaking strength is measured. It takes an operator a whole day to make one component and there are 4 operators and 4 days on which the experiment will take place. A Latin square design is appropriate here where the operator and the day are the blocking effects.

```
> data(breaking)
> breaking
      y operator  day supplier
```

```
 1  810      op1 day1        B
 2 1080      op1 day2        C
...etc...
15 1025      op4 day3        D
16  900      op4 day4        C
```

We can check the Latin square structure:

```
> matrix(breaking$supplier,4,4)
     [,1] [,2] [,3] [,4]
[1,] "B"  "C"  "D"  "A"
[2,] "C"  "D"  "A"  "B"
[3,] "A"  "B"  "C"  "D"
[4,] "D"  "A"  "B"  "C"
```

Plot the data:

```
> plot(y ~ operator + day + supplier, breaking)
```

Examine the boxplots in Figure 16.4. There appear to be differences in suppliers but not in the two blocking variables. No outlier, skewness or unequal variance is apparent.

Now fit the Latin squares model:

```
> g <- lm(y ~ operator + day + supplier, breaking)
> anova(g)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value Pr(>F)
operator   3   7662    2554    0.41 0.7510
day        3  17600    5867    0.94 0.4759
supplier   3 371138  123712   19.93 0.0016
Residuals  6  37250    6208
```

Does it make a difference if we change the order of fitting? Let's see:

```
> anova(lm(y ~ day + supplier + operator, breaking))
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value Pr(>F)
day        3  17600    5867    0.94 0.4759
supplier   3 371137  123712   19.93 0.0016
operator   3   7662    2554    0.41 0.7510
Residuals  6  37250    6208
```

They are the same because of the balanced design. We see that there is clear supplier effect but no evidence of an effect due to day or operator.

Now check the diagnostics

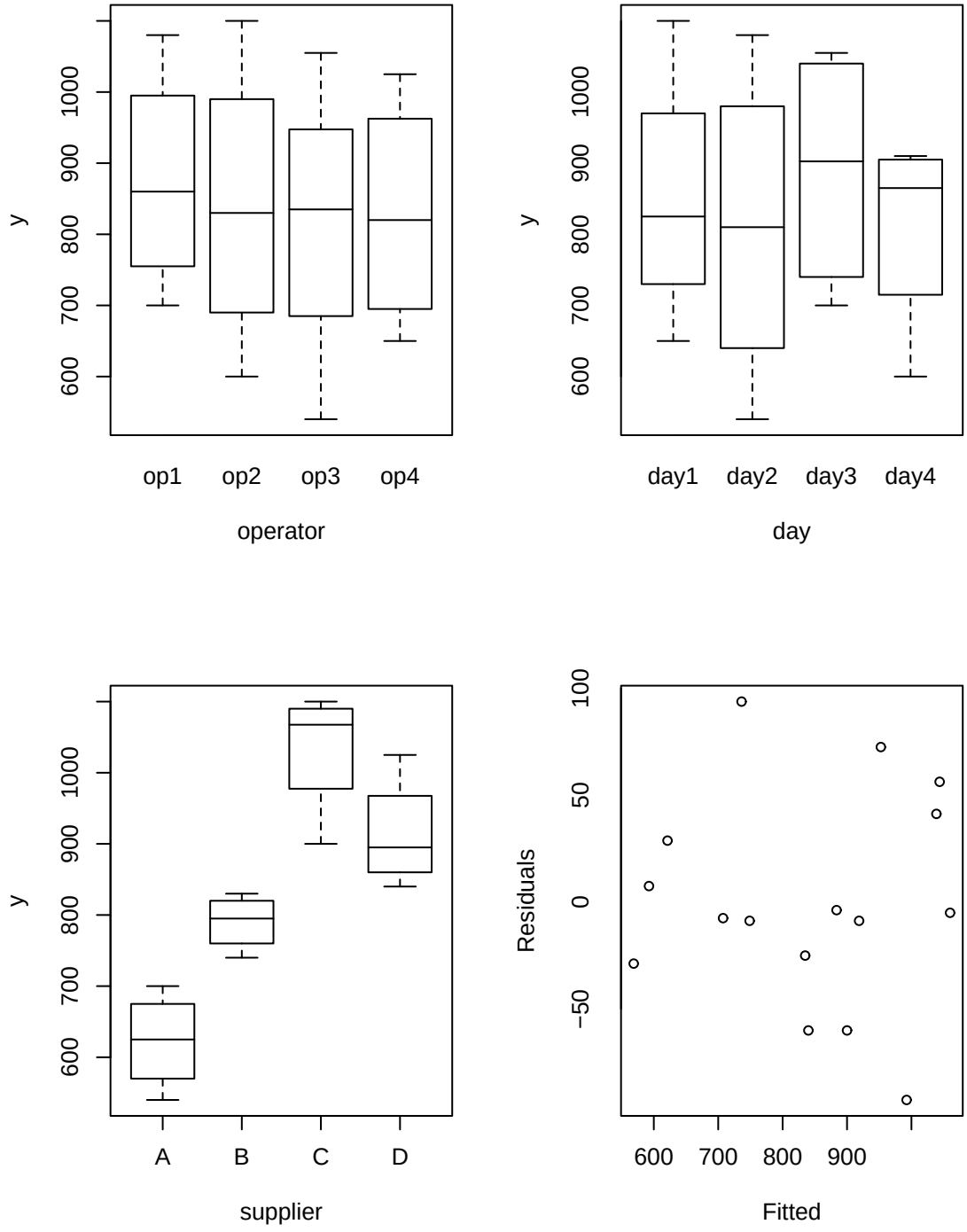

Figure 16.8: Latin square analysis

```
> plot(g$fit,g$res,xlab="Fitted",ylab="Residuals")
> qqnorm(g$res,ylab="Residuals")
```

I show only the residual-fitted plot which is fine as was the Q-Q plot. Now look at the estimates of the effects:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    667.5       62.3   10.72  3.9e-05
operatorop2    -35.0       55.7   -0.63  0.55302
operatorop3    -58.7       55.7   -1.05  0.33227
operatorop4    -46.2       55.7   -0.83  0.43825
dayday2        -40.0       55.7   -0.72  0.49978
dayday3         40.0       55.7    0.72  0.49978
dayday4        -40.0       55.7   -0.72  0.49978
supplierB      167.5       55.7    3.01  0.02381
supplierC      411.2       55.7    7.38  0.00032
supplierD      291.2       55.7    5.23  0.00196

Residual standard error: 78.8 on 6 degrees of freedom
Multiple R-Squared: 0.914,      Adjusted R-squared: 0.785
F-statistic: 7.09 on 9 and 6 degrees of freedom,        p-value: 0.0135
```

We see that Supplier C looks best followed by D. Is the difference significant though? Which suppliers in general are significantly better than others? We need the Tukey pairwise intervals to help determine this. The width of the bands calculated in the usual manner:

```
> qtukey(0.95,4,6)*55.7/sqrt(2)
[1] 193
```

The width of the interval is 193 - what can we say about differences between suppliers? We can make a handy table of the supplier differences:

```
> scoefs <- c(0,g$coef[8:10])
> outer(scoefs,scoefs,"-")
                supplierB supplierC supplierD
           0.00   -167.50   -411.25   -291.25
supplierB 167.50      0.00   -243.75   -123.75
supplierC 411.25    243.75      0.00    120.00
supplierD 291.25    123.75   -120.00      0.00
```

We see that the (A,B), (B,D) and (D,C) differences are not significant at the 5% level. Notice that it would not be reasonable to include that A is no different from C by chaining these comparisons together because each comparison is made using a statistical test where doubt exists about the conclusion and not a logical and definite assertion of equality.

If maximizing breaking strength is our aim, we would pick supplier C but if supplier 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.

How much more (or less) efficient is the Latin square compared to other designs? First compare to the completely randomized design:

```
> gr <- lm(y ~ supplier,breaking)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 0.8391
```

We see that the LS is 16% less efficient than the CRD. Now compare to the blocked designs:

```
> gr <- lm(y ~ supplier+operator,breaking)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 0.98166
> gr <- lm(y ~ supplier+day,breaking)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 0.8038
```

We see that the Latin square turned out to be a bad choice of design because there was very little if any difference between the operators and days but we did not know that until after the experiment! Next time we will know better.

16.5 Balanced Incomplete Block design

For 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 penicillin example, suppose 6 production processes were to be compared but each batch of material was only sufficient for four runs.

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 design*, all the pairwise differences are identifiable and have the same standard error. Pairwise differences are more likely to be interesting than other contrasts. Here is an example:

We have 4 treatments (t=4) 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

1	A	B	C
2	A	B	D
3	A	C	D
4	B	C	D

Table 16.3: BIB design

Each pair of treatments appears in the same block $\lambda = 2$ times — this feature enables simple pairwise comparison. 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 integer, a BIB design is not always possible even when the first two conditions are satisfied. For example, consider $r = 4, t = 3, b = 6, k = 2$ 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). BIB's 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:

$$y_{ij} = \mu + \tau_i + \rho_j + \varepsilon_{ij}$$

A nutrition specialist studied the effects of six diets, a, b, c, d, e, and f on weight gain of domestic rabbits. When rabbits reached 4 weeks of age they were assigned to a diet. It was decided to block on litters of rabbits but from past experience about sizes of litters, it was felt that only 3 uniform rabbits could be selected from each available litter. There were ten litters available forming blocks of size three. Each pair of diets appear in the same block twice. Examine the data.

```
> data(rabbit)
> rabbit
   block treat gain
 1    b1     f 42.2
 2    b1     b 32.6
etc.
30   b10     a 37.3
```

We can see the BIB structure:

```
> matrix(rabbit$treat,nrow=3)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] "f"  "c"  "c"  "a"  "e"  "b"  "d"  "a"  "d"  "f"
[2,] "b"  "a"  "f"  "e"  "c"  "f"  "a"  "e"  "b"  "d"
[3,] "c"  "b"  "d"  "c"  "d"  "e"  "b"  "f"  "e"  "a"
```

Now plot the data:

```
> plot(gain ~ block + treat, rabbit)
```

See Figure 16.5. What do you conclude? Now fit the model:

```
> g <- lm(gain ~ block+treat,data=rabbit)
> anova(g)
Analysis of Variance Table

Response: gain
          Df Sum Sq Mean Sq F value  Pr(>F)
block      9    730      81    8.07 0.00025
treat      5    159      32    3.16 0.03817
Residuals 15    151      10
```

Changing the order of treatment and block:

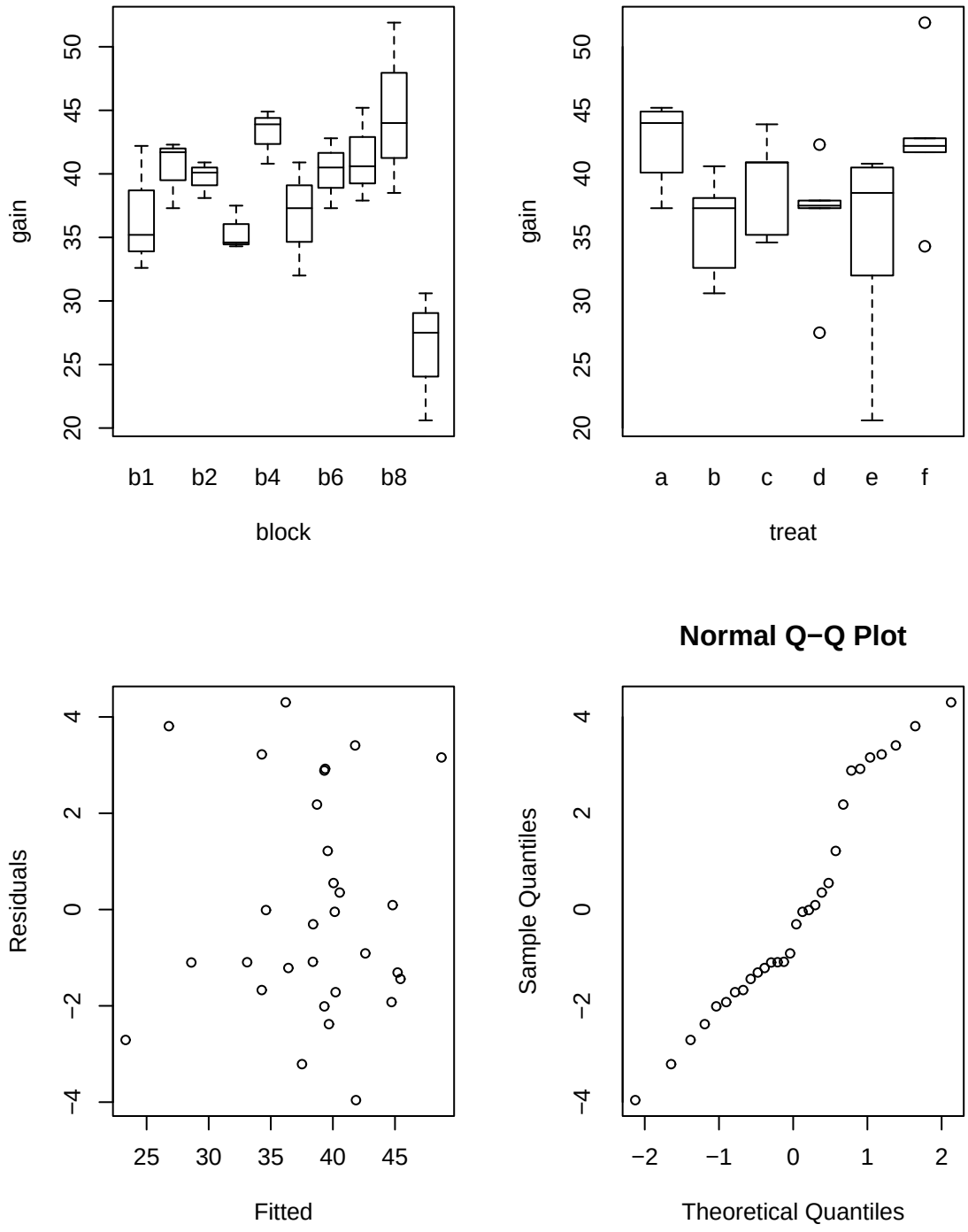

Figure 16.9: Balanced incomplete block analysis

```
> anova(lm(gain ~ treat+block,data=rabbit))
Analysis of Variance Table

Response: gain
          Df Sum Sq Mean Sq F value  Pr(>F)
treat      5    293      59    5.84 0.00345
block      9    596      66    6.59 0.00076
Residuals 15    151      10
```

Does make a difference because the design is not orthogonal because of the incompleteness. Which table is appropriate for testing the treatment effect or block effect? The first one, because we want to test for a treatment effect after the blocking effect has been allowed for.

Now check the diagnostics

```
> plot(g$fitted,g$res,xlab="Fitted",ylab="Residuals")
> qqnorm(g$res)
```

Which treatments differ? We need to do pairwise comparisons. Tukey pairwise confidence intervals are easily constructed:

$$\hat{\tau}_l - \hat{\tau}_m \pm \frac{q_{t,n-b-t+1}}{\sqrt{2}}\sqrt{\frac{2k}{\lambda t}}\hat{\sigma}$$

First we figure out the difference between the treatment effects:

```
> tcoefs <- c(0,g$coef[11:15])
> outer(tcoefs,tcoefs,"-")
                 treatb   treatc    treatd treate  treatf
       0.000000  1.7417 -0.40000 -0.066667 5.2250 -3.3000
treatb -1.741667  0.0000 -2.14167 -1.808333 3.4833 -5.0417
treatc  0.400000  2.1417  0.00000  0.333333 5.6250 -2.9000
treatd  0.066667  1.8083 -0.33333  0.000000 5.2917 -3.2333
treate -5.225000 -3.4833 -5.62500 -5.291667 0.0000 -8.5250
treatf  3.300000  5.0417  2.90000  3.233333 8.5250  0.0000
```

Now we want the standard error for the pairwise comparisons:

```
> summary(g)
Coefficients:
            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

Residual standard error: 3.17 on 15 degrees of freedom
Multiple R-Squared: 0.855,      Adjusted R-squared: 0.72
F-statistic: 6.32 on 14 and 15 degrees of freedom,      p-value: 0.000518
```

We see that the standard error for the pairwise comparison is 2.24. This can also be obtained as $\sqrt{\frac{2k}{\lambda t}}\hat{\sigma}$:

```
> sqrt((2*3)/(2*6))*3.17
[1] 2.2415
```

Notice that all the treatment standard errors are equal because of the BIB. If the roles of blocks and treatments were reversed, we see that the design would not be balanced and hence the unequal standard errors for the blocks.

Now compute the Tukey critical value:

```
> qtukey(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:

```
> abs(outer(tcoefs,tcoefs,"-")) > 7.27
             treatb treatc treatd treate treatf
       FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
treatb FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
treatc FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
treatd FALSE  FALSE  FALSE  FALSE  FALSE  FALSE
treate FALSE  FALSE  FALSE  FALSE  FALSE   TRUE
treatf FALSE  FALSE  FALSE  FALSE   TRUE  FALSE
```

Only the e-f difference is significant.

How much better is this blocked design than the CRD? We compute the relative efficiency:

```
> gr <- lm(gain ~ treat,rabbit)
> (summary(gr)$sig/summary(g)$sig)^2
[1] 3.0945
```

Blocking was well worthwhile here.

16.6 Factorial experiments

Suppose we have

- Factors $\alpha, \beta, \gamma \ldots$
- with levels $l_\alpha, l_\beta, l_\gamma \ldots$

A *full* factorial experiment has at least one run for each combination of the levels. The number of combinations is $l_\alpha l_\beta l_\gamma \ldots$ which could easily be very large. The biggest model for a full factorial contains all possible interaction terms which may be of quite high order.

Advantages of factorial designs

1. 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.
2. Factorial experiments are efficient — it is often better to use replication for investigating another factor instead. For example, instead of doing a 2 factor experiment with replication, it is often better to use that replication to investigate another factor.

Disadvantage of factorial designs

Experiment may be too large and so cost too much time or money.

Analysis

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 because the experimental material is limited and only a few runs can be made. It is often possible to estimate the lower order effects with just a fraction. Consider an experiment with 7 factors, each at 2 levels

	mean	main	2-way	3-way	4	5	6	7
no. of pars	1	7	21	35	35	21	7	1

Table 16.4: No. of parameters

If we are going to assume that higher order interactions are negligible then we don't really need $2^7 = 128$ runs to estimate the remaining parameters. We could run only a quarter of that, 32, and still be able to estimate main and 2-way effects. (Although, in this particular example, it is not possible to estimate all the two-way interactions uniquely. This is because, in the language of experimental design, there is no available resolution V design, only a resolution IV design is possible.)

A Latin square 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 little data as possible. This means there is often not many degrees of freedom left over. 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. In the social sciences and medicine,

the experimental materials, often human or animal, are much less homgenous and less controllable so σ^2 tends to be larger. In such cases, fractional factorials are of no value.

Fractional factorials are popular 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.

Speedometer cables can be noisy because of shrinkage in the plastic casing material, so an experiment was conducted to find out what caused shrinkage. The engineers started with 15 different factors: liner O.D., 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, labelled a through o. Response is percentage shrinkage per specimen. There were two levels of each factor. A full factorial would take 2^{15} runs, which is highly impractical so a design with only 16 runs was used where the particular runs have been chosen specially so as to estimate the the mean and the 15 main effects. We assume that there is no interaction effect of any kind. The purpose of such an experiment is to screen a large number of factors to identify which are important. Examine the data. The + indicates the high level of a factor, the - the low level. The data comes from Box, Bisgaard, and Fung (1988)

Read in and check the data.

```
> data(speedo)
> 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
```

Fit and examine a main effects only model:

```
> g <- lm(y ~ .,speedo)
> summary(g)
Residuals:
ALL 16 residuals are 0: no residual degrees of freedom!

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.582500
h           -0.062188
d           -0.060938
```

```
l           -0.027188
b            0.055937
j            0.000938
f           -0.074062
n           -0.006562
a           -0.067813
i           -0.042813
e           -0.245312
m           -0.027813
c           -0.089687
k           -0.068438
g            0.140312
o           -0.005937

Residual standard error: NaN on 0 degrees of freedom
Multiple R-Squared:    1,       Adjusted R-squared:  NaN
F-statistic:  NaN on 15 and 0 degrees of freedom,       p-value:  NaN
```

Why are there no degrees of freedom? Why do we have so many ”NA”’s in the display? Because there are as many parameters as cases.

It’s important to understand the coding here, so look at the X-matrix.

```
> model.matrix(g)
   (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 0 and “-” is coded as 1. This unnatural ordering is because of their order in the ASCII alphabet.

We don’t have any degrees of freedom so we can’t make the usual F-tests. We need a different method. Suppose there were no significant effects and the errors are 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. The `qqnorm()` function is not suitable because we want to label the points.

```
> coef <- g$coef[-1]
> i <- order(coef)
> plot(qnorm(1:15/16),coef[i],type="n",xlab="Normal Quantiles",
   ylab="Effects")
> text(qnorm(1:15/16),coef[i],names(coef)[i])
```

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’d prefer what ever “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 is better for detecting extreme points. This plots the sorted absolute values against $\Phi^{-1}((n+i)/(2n+1))$. Thus it compares the absolute values of the data against the upper half of a normal distribution. We don’t particularly care if the coefficients are not normally distributed, it’s just the extreme

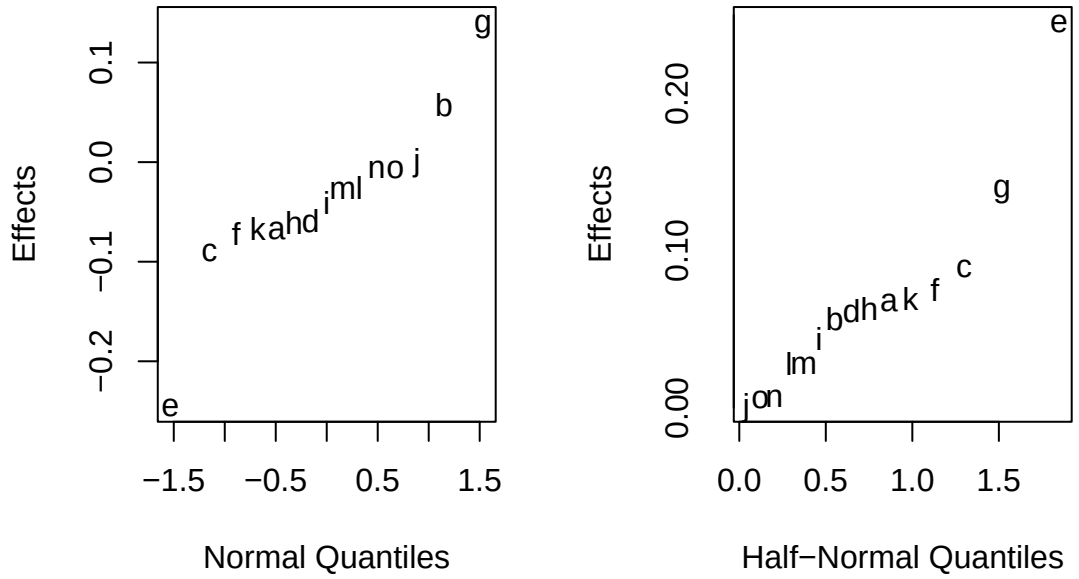

Figure 16.10: Fractional Factorial analysis

cases we want to detect. Because the half-normal folds over the ends of a QQ plot it “doubles” our resolution for the detection of outliers.

```
> coef <- abs(coef)
> i <- order(coef)
> plot(qnorm(16:30/31),coef[i],type="n",xlab="Half-Normal Quantiles",
 ylab="Effects")
> text(qnorm(16:30/31),coef[i],names(coef)[i])
```

We might now conduct another experiment focusing on the effect of ”e” and ”g”.

Appendix A

Recommended Books

A.1 Books on R

There are currently no books written specifically for R , although several guides can be downloaded from the R web site.

R is very similar to S-plus so most material on S-plus applies immediately to R . I highly recommend Venables and Ripley (1999). Alternative introductory books are Spector (1994) and Krause and Olson (2000). You may also find Becker, Chambers, and Wilks (1998) and Chambers and Hastie (1991), useful references to the S language. Ripley and Venables (2000) is a more advanced text on programming in S or R .

A.2 Books on Regression and Anova

There are many books on regression analysis. Weisberg (1985) is a very readable book while Sen and Srivastava (1990) contains more theoretical content. Draper and Smith (1998) is another well-known book. One popular textbook is Kutner, Nachtschiem, Wasserman, and Neter (1996). This book has everything spelled out in great detail and will certainly strengthen your biceps (1400 pages) if not your knowledge of regression.

Appendix B

R functions and data

R may be obtained from the R project web site at `www.r-project.org`.

This book uses some functions and data that are not part of base R . You may wish to download these functions from the R web site. The additional packages used are

```
MASS leaps ggobi ellipse nlme
```

MASS and nlme are part of the "recommended" R installation so depending on what installation option you choose, you may already have these without additional effort. Use the command

```
> library()
```

to see what packages you have. The MASS functions are part of the VR package that is associated with the book Venables and Ripley (1999). The ggobi data visualization application may also need to be installed. This may be obtained from `www.ggobi.org` This is not essential so don't worry if you can't install it. In addition, you will need the splines, mva and lqs packages but these come with basic R installation so no extra work is necessary.

I have packaged the data and functions that I have used in this book as an R package that you may obtain from my web site — `www.stat.lsa.umich.edu/~faraway`. The functions available are

```
halfnorm        Half normal plot
Cpplot          Cp plot
qqnorml         Case-labeled Q-Q plot
maxadjr         Models with maximum adjusted R^2
vif             Variance Inflation factors
prplot          Partial residual plot
```

In addition the following datasets are used:

```
breaking        Breaking strengths of material by day, supplier, operator
cathedral       Cathedral nave heights and lengths in England
chicago         Chicago insurance redlining
chiczip         Chicago zip codes north/south
chmiss          Chicago data with some missing values
coagulation     Blood coagulation times by diet
corrosion       Corrosion loss in Cu-Ni alloys
eco             Ecological regression example
gala            Species diversity on the Galapagos Islands
```

```
odor            Odor of chemical by production settings
pima            Diabetes survey on Pima Indians
penicillin      Penicillin yields by block and treatment
rabbit          Rabbit weight gain by diet and litter
rats            Rat survival times by treatment and poison
savings         Savings rates in 50 countries
speedo          Speedometer cable shrinkage
star            Star light intensities and temperatures
strongx         Strong interaction experiment data
twins           Twin IQs from Burt
```

Where add-on packages are needed in the text, you will find the appropriate `library()` command. However, I have assumed that the `faraway` library is always loaded. You can add a line reading `library(faraway)` to your Rprofile file if you expect to use this package in every session. Otherwise you will need to remember to type it each time.

I set the following options to achieve the output seen in this book

```
> options(digits=5,show.signif.stars=FALSE)
```

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 2 or 3 significant digits in a displayed table is usually unnecessary and more important, distracting.

Appendix C

Quick introduction to R

C.1 Reading the data in

The first step is to read the data in. You can use the `read.table()` or `scan()` function to read data in from outside R . You can also use the `data()` function to access data already available within R .

```
> data(stackloss)
> stackloss
   Air.Flow Water.Temp Acid.Conc. stack.loss
1        80         27         89         42
2        80         27         88         37
... stuff deleted ...
21       70         20         91         15
```

Type

```
> help(stackloss)
```

We can check the dimension of the data:

```
> dim(stackloss)
[1] 21  4
```

C.2 Numerical Summaries

One easy way to get the basic numerical summaries is:

```
> summary(stackloss)
    Air.Flow      Water.Temp     Acid.Conc.     stack.loss
 Min.   :50.0   Min.   :17.0   Min.   :72.0   Min.   : 7.0
 1st Qu.:56.0   1st Qu.:18.0   1st Qu.:82.0   1st Qu.:11.0
 Median :58.0   Median :20.0   Median :87.0   Median :15.0
 Mean   :60.4   Mean   :21.1   Mean   :86.3   Mean   :17.5
 3rd Qu.:62.0   3rd Qu.:24.0   3rd Qu.:89.0   3rd Qu.:19.0
 Max.   :80.0   Max.   :27.0   Max.   :93.0   Max.   :42.0
```

We can compute these numbers seperately also:

```
> stackloss$Air.Flow
 [1] 80 80 75 62 62 62 62 62 58 58 58 58 58 58 50 50 50 50 50 56 70
> mean(stackloss$Ai)
[1] 60.429
> median(stackloss$Ai)
[1] 58
> range(stackloss$Ai)
[1] 50 80
> quantile(stackloss$Ai)
  0%  25%  50%  75% 100%
  50   56   58   62   80
```

We can get the variance and sd:

```
> var(stackloss$Ai)
[1] 84.057
> sqrt(var(stackloss$Ai))
[1] 9.1683
```

We can write a function to compute sd's:

```
> sd <- function(x) sqrt(var(x))
> sd(stackloss$Ai)
[1] 9.1683
```

We might also want the correlations:

```
> cor(stackloss)
           Air.Flow Water.Temp Acid.Conc. stack.loss
Air.Flow    1.00000    0.78185    0.50014    0.91966
Water.Temp  0.78185    1.00000    0.39094    0.87550
Acid.Conc.  0.50014    0.39094    1.00000    0.39983
stack.loss  0.91966    0.87550    0.39983    1.00000
```

Another numerical summary with a graphical element is the stem plot:

```
> stem(stackloss$Ai)

  The decimal point is 1 digit(s) to the right of the |

  5 | 000006888888
  6 | 22222
  7 | 05
  8 | 00
```

C.3 Graphical Summaries

We can make histograms and boxplot and specify the labels if we like:

```
> hist(stackloss$Ai)
> hist(stackloss$Ai,main="Histogram of Air Flow",
  xlab="Flow of cooling air")
> boxplot(stackloss$Ai)
```

Scatterplots are also easily constructed:

```
> plot(stackloss$Ai,stackloss$W)
> plot(Water.Temp ~ Air.Flow,stackloss,xlab="Air Flow",
  ylab="Water Temperature")
```

We can make a scatterplot matrix:

```
> plot(stackloss)
```

We can put several plots in one display

```
> par(mfrow=c(2,2))
> boxplot(stackloss$Ai)
> boxplot(stackloss$Wa)
> boxplot(stackloss$Ac)
> boxplot(stackloss$s)
> par(mfrow=c(1,1))
```

C.4 Selecting subsets of the data

Second row:

```
> stackloss[2,]
  Air.Flow Water.Temp Acid.Conc. stack.loss
2       80         27         88         37
```

Third column:

```
> stackloss[,3]
 [1] 89 88 90 87 87 87 93 93 87 80 89 88 82 93 89 86 72 79 80 82 91
```

The 2,3 element:

```
> stackloss[2,3]
[1] 88
```

c() is a function for making vectors, e.g.

```
> c(1,2,4)
[1] 1 2 4
```

Select the first, second and fourth rows:

```
> stackloss[c(1,2,4),]
  Air.Flow Water.Temp Acid.Conc. stack.loss
1       80         27         89         42
2       80         27         88         37
4       62         24         87         28
```

The : operator is good for making sequences e.g.

```
> 3:11
[1]  3  4  5  6  7  8  9 10 11
```

We can select the third through sixth rows:

```
> stackloss[3:6,]
  Air.Flow Water.Temp Acid.Conc. stack.loss
3       75         25         90         37
4       62         24         87         28
5       62         22         87         18
6       62         23         87         18
```

We can use ”-” to indicate ”everthing but”, e.g all the data except the first two columns is:

```
> stackloss[,-c(1,2)]
   Acid.Conc. stack.loss
1          89         42
2          88         37
... stuff deleted ...
21         91         15
```

We may also want select the subsets on the basis of some criterion e.g. which cases have an air flow greater than 72.

```
> stackloss[stackloss$Ai > 72,]
  Air.Flow Water.Temp Acid.Conc. stack.loss
1       80         27         89         42
2       80         27         88         37
3       75         25         90         37
```

C.5 Learning more about R

While running R you can get help about a particular commands - eg - if you want help about the `stem()` command just type `help(stem)`.

If you don’t know what the name of the command is that you want to use then type:

```
help.start()
```

and then browse. You may be able to learn the language simply by example in the text and refering to the help pages.

You can also buy the books mentioned in the recommendations or download various guides on the web — anything written for S-plus will also be useful.

Bibliography

Andrews, D. and A. Herzberg (1985). *Data : a collection of problems from many fields for the student and research worker*. New York: Springer-Verlag.

Becker, R., J. Chambers, and A. Wilks (1998). *The new S language: A Programing Environment for Data Analysis and Graphics* (revised ed.). CRC.

Belsley, D. A., E. Kuh, and R. E. Welsch (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. New York: Wiley.

Box, G. P., S. Bisgaard, and C. Fung (1988). An explanation and critque of taguchi's contributions to quality engineering. *Quality and reliability engineering international 4*, 123–131.

Box, G. P., W. G. Hunter, and J. S. Hunter (1978). *Statistics for Experimenters*. New York: Wiley.

Carroll, R. and D. Ruppert (1988). *Transformation and Weighting in Regression*. London: Chapman Hall.

Chambers, J. and T. Hastie (1991). *Statistical Models in S*. Chapman and Hall.

Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *JRSS-A 158*, 419–466.

Draper, D. (1995). Assessment and propagation of model uncertainty. *JRSS-B 57*, 45–97.

Draper, N. and H. Smith (1998). *Applied Regression Analysis* (3rd ed.). New York: Wiley.

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. Springer Verlag.

Hsu, J. (1996). *Multiple Comparisons Procedures: Theory and Methods*. London: Chapman Hall.

Ihaka, R. and R. Gentleman (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics 5*(3), 299–314.

Johnson, M. P. and P. H. Raven (1973). Species number and endemism: The galápagos archipelago revisited. *Science 179*, 893–895.

Krause, A. and M. Olson (2000). *The basics of S and S-Plus* (2nd ed.). New York: Springer-Verlag.

Kutner, M., C. Nachtschiem, W. Wasserman, and J. Neter (1996). *Applied Linear Statistical Models* (4th ed.). McGraw-Hill.

Longley, J. W. (1967). An appraisal of least-squares programs from the point of view of the user. *Journal of the American Statistical Association 62*, 819–841.

Ripley, B. and W. Venables (2000). *S Programming*. New York: Springer Verlag.

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.

Spector, P. (1994). *Introduction to S and S-Plus*. Duxbury.

Venables, W. and B. Ripley (1999). *Modern Applied Statistics with S-PLUS* (3rd ed.). Springer.

Weisberg, S. (1985). *Applied Linear Regression* (2nd ed.). New York: Wiley.