H. Bar, N. Ravishanker, G. Asha

Statistical Practice for Data Science: with Hands-on Illustrations using R (draft)

Contents

1

Introduction to R and RStudio

See the material from the R bootcamp given by Chiranjit Dutta Aug 8-11, 2022. Anyone with this link will have access to R bootcamp materials, including the lecture materials and Webex recording links.

`https://gitfront.io/r/user-3679719/H8v5hTnvHh2U/R-Bootcamp-Course/`

Webex recording links

Recording link Day-1 : `https://uconn-cmr.webex.com/uconn-cmr/ldr.php?RCID=e36877d66ff57b90539915c6e64687c2`

Recording link Day-2 : `https://uconn-cmr.webex.com/uconn-cmr/ldr.php?RCID=6a39d6cbf1e3820375144b5c453102e8`

Recording link Day-3 : `https://uconn-cmr.webex.com/uconn-cmr/ldr.php?RCID=7ecdd012e6872bc4b3eaf059b01b80b9`

Recording link Day-4 : `https://uconn-cmr.webex.com/uconn-cmr/ldr.php?RCID=7138caff42cf86fcbb48ce5b304a1a46`

2

Data Visualization

In this chapter, we discuss visualization tools for qualitative and quantitative data, to get enhanced plots for independent and dependent multivariate data samples. We also describe techniques for assessing whether a random sample comes from a normal population.

2.1 Introduction

Data visualization is often the first step in any data analysis. Pictorial representations give useful initial insights about the data and also become useful in model diagnostics. In this chapter, we discuss a few popular data visualization tools for univariate and multivariate data. Histograms, boxplots and stem-and-leaf plots are popularly used for the exploration of numerical data. By numerical data, we mean *quantitative data* that may be either integer-valued or real-valued. These data are represented using different graphical methods than *qualitative data*, which usually occur as categories or labels and are usually represented using barplots, pie diagrams, etc. The R software has wide graphic capabilities; use demo(graphics) to explore the various possibilities. In many situations that arise in the book, we must assess whether a random sample could have been generated from a normal population.The normal quantile-quantile, or Q-Q plot is a graphical way to do this, and is often preferred to significance tests that enable the assessment.

2.2 Essential Terminology and Notation

✓ The **theoretical cumulative distribution function** (c.d.f.) of a continuous random variable X is a continuous, nondecreasing function defined by

$$F(x) = P(X \leq x) \quad \text{for all real } x. \tag{2.1}$$

Note that $0 \leq F(x) \leq 1$, and $P(a < X < b) = F(b) - F(a)$.

✓ The **probability density function** (p.d.f.) of a continuous random variable X is a function $f(x)$ which assigns area to a small interval surrounding x. Given the c.d.f.

$F(x)$, we can obtain the p.d.f. as

$$f(x) \quad = \quad \frac{d}{dx}F(x). \tag{2.2}$$

✓ A **kernel density plot** is a variation of a histogram that uses kernel density estimation (or, kernel smoothing) to smooth out noise, leading to a smoother distribution than a histogram.

✓ The **p.d.f. of a normal random variable** $X \sim N(\mu, \sigma^2)$ is

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad -\infty < x < \infty, \tag{2.3}$$

where $\mu \in \mathbb{R}$ is the mean (location parameter) and $\sigma > 0$ is the standard deviation (scale parameter) of the distribution. All the distributional properties of X are entirely characterized by its mean μ and variance σ^2.

✓ Probabilities and areas under the curve for the normal p.d.f. in (2.3) are given by the **normal cumulative distribution function (c.d.f.)**

$$F(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-(u-\mu)^2/2\sigma^2} du. \tag{2.4}$$

✓ If $X \sim N(\mu, \sigma^2)$, then the random variable $Z = (X - \mu)/\sigma$ is called the **standard normal variable**. Its mean is $E(Z) = 0$, and its variance is $Var(Z) = 1$. We say $Z \sim N(0, 1)$ and all normal probabilities may be calculated in terms of the standard normal probabilities. The p.d.f. and c.d.f. of Z are

$$\begin{aligned} \phi(z) &= \frac{1}{\sqrt{2\pi}} e^{-z^2/2}, \text{ and} \\ \Phi(z) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-u^2/2} du. \end{aligned} \tag{2.5}$$

✓ **Empirical Rule.** For data drawn from ny normal distribution, the empirical rule says that approximately 68%, 95% and 99.9% of the data lie respectively within one, two and three standard deviations of the sample mean.

✓ The normal family is an example of the **location-scale family of distributions**, with the distribution of the standard normal variable Z being the standard distribution in the family. This means that each member of this family will have the same bell-shaped curve implied by the empirical rule. If the value of μ changes while σ^2 is held constant, the distribution shifts to the right or left, retaining its bell shape and size. If the value of σ^2 changes while μ is held constant, the distribution's center and bell shape stay the same, but the size expands or contracts. If the values of μ and σ^2 change, the distribution retains its bell shape, but can shift to the right or left and expand or contract in size.

✓ Suppose $F(x)$ denotes the c.d.f. of a specified probability distribution, and suppose it is strictly increasing and continuous. For $0 < p < 1$, the **theoretical pth quantile** of $F(\cdot)$ is a unique real value $Q_T(p)$ which satisfies

$$F(Q_T(p)) = p, \text{ or } Q_T(p) = F^{-1}(p).$$

✓ **Order statistics.** The ith order statistic of a random sample $X_1, \ldots, X_n$ is the ith smallest value when the observations are arranged in ascending order. Order statistics are denoted by $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$, or by $X_{1:n} \leq X_{2:n} \leq \ldots \leq X_{n:n}$.

✓ **Empirical quantiles.** Let $X_1, \ldots, X_n$ be a random sample from some population $F(x)$. Consider the proportions $p_i = (i - 0.5)/n$. The ith empirical (data) quantile $Q_E(p_i)$ corresponding to $p_i = (i - 0.5)/n$ is the ith order statistic $X_{(i)}$, so that a proportion p_i of the sample data lies below this value.

✓ Given the order statistics $X_{(i)}$ corresponding to a random sample of size n, the **empirical c.d.f.** $F_n(x)$ is defined as the proportion of sample observations that are $\leq x$:

$$F_n(x) = \begin{cases} 0 & \text{if } x < X_{(1)} \\ i/n & \text{if } X_{(i)} \leq x < X_{(i+1)} \\ 1 \text{ if } x > X_{(n)}. \end{cases} \tag{2.6}$$

The empirical c.d.f. is a step function with jumps at $X_{(i)},\ i = 1, \ldots, n$, each jump being of size $1/n$.

✓ The **multinomial distribution** is the distribution of a discrete random variable useful for modeling categorical data, and corresponds to a multinomial experiment. A random vector $\boldsymbol{y} = (Y_1, \ldots, Y_k)'$ has a multinomial distribution with n trials and cell probabilities $p_1, \ldots, p_k$ if its joint p.m.f. is

$$P(Y_1 = C_1, \ldots, Y_k = C_k) = n! \prod_{i=1}^{k} \frac{p_i^{C_i}}{C_i!} \tag{2.7}$$

on the set of frequencies $(C_1, \ldots, C_k)$ such that each C_i is a nonnegative integer and $\sum_{i=1}^{k} C_i = n$. Note that $\sum_{i=1}^{k} p_i = 1$, so that only $k - 1$ of these probabilities can vary freely between 0 and 1, and the kth probability is determined as $p_k = 1 - \sum_{i=1}^{k-1} p_i$.

2.3 Data Sets

Before embarking on a statistical analysis, it is important to explore the data we are dealing with. It is good practice to identify the variables, study the data structure and handle the data appropriately. This helps us to identify suitable statistical tools for data analysis. The function str() helps us to understand the structure of data sets stored in R. Data types include character, numeric, integer, complex, or logical. The function typeof(x) will exhibit the data type of a variable called `x`. Factors are normally used to group variables into categories or levels. The function as.factor() coerces its arguments to a factor. The function levels() shows all the possible levels of a factor variable.

Trees

Example 2.3.1 (Trees) The trees data is available in the *datasets* package in R. It provides measurements of the girth, height and volume of timber in 31 felled black cherry trees. Note that girth is the diameter of the tree (in inches) measured at 4ft 6in above the ground. The data is stored as a data frame with 31 observations (rows) and 3 numeric variables (columns), `Girth`, `Height`, `Volume`.

Try running the following code in R, making sure that the file trees.csv is in the working directory.

```
trees <- read.csv('Data/trees.csv', header=TRUE)
str(trees)
```

The output shows the structure of trees data.

```
'data.frame':        31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
 $ Height: int  70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

Crab mating

Example 2.3.2 (Crab Mating) (Source: Agresti (2007)Agresti, A.) Each female horseshoe crab in this study had a male crab attached to her in her nest. The study investigated some factors that were thought to affect whether the female crab had any other males residing nearby, called satellites. These factors included the color, spine condition, weight (kg), and carapace width (cm) of the female crabs. Color has four levels (1: light medium; 2: medium; 3: dark medium; 4: dark). Spine condition has three levels (1: both good; 2: one worn or broken; 3: both worn or broken). The crabs data is available in the R package *glmbb*. The spine condition levels are coded as good, bad, middle. The data has 173 observations on six variables of which two are categorical while the rest are quantitative. Use the function levels() on the data object through the command sapply() to see the levels of the variables. The command factor() can be used to encode a vector as a factor.

```
crabs <- read.csv('Data/crabs.csv', header=TRUE)
# show the levels of all the variables in the crabs dataset:
sapply(crabs, levels)
levels(crabs$color)
```

The output from the last line is

```
[1] "dark"   "darker" "light"  "medium"
```

It is possible to change the order of levels in a factor. For example, to change the order of `color`, we can run

```
crabs$color <- factor(crabs$color, levels=c("light", "medium", "dark",
    "darker"))
```

The `spine` variable should also be reordered (bad, middle, good).

Birdkeeping

Example 2.3.3 (Birdkeeping) (Source: Ramsey and Schafer (2013)) A 1972–1981 health survey in The Hague, The Netherlands, discovered an association between keeping pet birds and increased risk of lung cancer. To investigate this, researchers conducted a case-control study in 1985 of patients at four hospitals in The Hague (population 450,000). They identified 49 cases of lung cancer among the patients who were registered with a general practice, who

were aged 65 or younger and who had resided in the city since 1965. They also selected 98 controls from a population of residents having the same general age structure. They collected data on the variables `LC` (an indicator whether the subject has lung cancer), `FM`, (an indicator for gender of the subject), `SS` (socioeconomic status, determined by occupation of the household's principal wage earner), `BK` (indicator of bird keeping), `AG` (age of the subject in years), `YR` (years of smoking prior to diagnosis or examination) and `CD` (average rate of smoking in number of cigarettes per day). This data is available in the R package *Sleuth3*.

```
birds <- read.csv('Data/birds.csv', header=TRUE)
birds$LC <- relevel(as.factor(birds$LC), ref = "NoCancer")
str(birds)
```

The command relevel(x, ref) reorders the levels of the factor variable LC so that the level specified in relevel() becomes the reference or first level.

Tensile strength

Example 2.3.4 (Tensile Strength) (Source: Montgomery (2017)) An engineer wants to see whether the cotton weight percentage in a synthetic fiber affects its tensile strength. The cotton weight percentage(`CWP`) is fixed at five different levels, i.e., 15%, 20%, 25%, 30%, and 35%. Each percentage level is assigned five experimental units and the tensile strength of the fabric is measured on each of them. The randomization is specified in the column `Run Number`. The goal of the engineer is to investigate whether `Tensile Strength` is the same across all levels of cotton weight percentage. The (`CWP`)is interpreted by R to be of integer type, but we may want to treat it as a factor, in which case we will use the as.factor command to coerce `CWP` to a factor variable.

```
tensile <- read.csv('Data/tensile.csv', header=TRUE)
str(tensile$CWP) # int [1:25] 20 30 20 35 30 15 25 20 25 30 ...
tensile$CWP <- as.factor(tensile$CWP)
str(tensile$CWP) # Factor w/ 5 levels "15","20","25",..: 2 4 2 5 4 1 3 2 3 4
    ...
```

Connecticut towns

Example 2.3.5 (Connecticut towns) The Connecticut Data Collaborative website[1] contains many datasets about the State's demographic, economy, housing, health, safety, etc. From this website we downloaded the total households by town for 2015-2019, and the dataset contains rows (towns), and three columns: Town, Federal Information Processing Standards (FIPS) code, and Value - the total number of households in the town.

2.4 Visualization for Quantitative Data

2.4.1 Histograms

A histogram is a pictorial representation of numerical or quantitative data. It uses rectangles to show the frequencies or relative frequencies of data items in successive intervals. A

[1] http://data.ctdata.org/dataset/total-households-by-town

histogram is constructed by dividing the entire range of data values into a series of intervals called bins, which are represented along the x-axis, while the frequencies (or relative frequencies of data items) are shown as rectangles on the y-axis. The shape of a histogram provides useful information such as the location, spread, skewness, outliers and data density.

A histogram can be plotted using the command hist(x, breaks=, freq=NULL, right=TRUE, density=NULL, angle=45) where x is the vector of values for which the histogram is to be plotted. We can specify the number of bins by using the parameter breaks. When freq=FALSE, probability densities are plotted. For right=TRUE, the cells in the histogram will be right-closed (left open) intervals. The parameter density provides the density of shading lines, in lines per inch, the default value of density=NULL giving no shading lines. The option angle gives the slope of shading lines. The option main allows us to set the heading of the histogram. Setting main=NA disables the display of the heading.

For the trees data in Example 2.3.1, we can plot a histogram of the tree heights using the hist function. We may also add a kernel density plot to the histogram using lines and density. One should not confuse the density option in the hist function (which controls the density of the filled rectangles) with the density function (which is used to compute the density of a distribution.) Similarly, we can overlay the histogram with a normal distribution, which helps us see the extent to which the histogram is similar to data from a normal distribution. For this, we use the dnorm function to calculate the normal probability density function with mean and standard deviation set to be the same as in the observed data. Pulling these together in one plot, we get Figure 2.1 by running the following code:

```
hist(trees$Height, main=NA, breaks=8, col='lightgreen', prob=T,
    border="white", xlab="height")
lines(density(trees$Height), col="blue", lwd=2)
x <- seq(min(trees$Height), max(trees$Height), 0.01)
curve(dnorm(x, mean=mean(trees$Height), sd=sd(trees$Height)), add=T,
    col="red", lty=2, lwd=2)
legend("topleft", c("Kernel Density", "Normal"), col=c("blue", "red"),
    lty=c(1,2), cex=0.8, bty = "n")
```

Note how the legend function was used to add a legend to the plot. The cex option controls the font size of the legend items, and the bty controls the box type. In this case, it is set to have no box.

2.4.2 Boxplots

A box and whisker plot, also known as a boxplot, is a popular way of displaying the distribution of numerical data based on the five number summary which includes the minimum, first quartile, median, third quartile, and maximum. Boxplots display information about the shape of the data distribution as well as its central value and variability. The box represents the middle 50 percent of the data. Boxplots also have lines extending vertically from the boxes; these are called whiskers and indicate variability beyond the upper and lower quartiles. The spacing between the different parts of the box indicates the degree of dispersion (spread) and skewness in the data and also reveals outliers. To draw a boxplot of the data, use the command

boxplot(x, range=, width=, notch=, outline=, names, horizontal=).

The option range provides a distance to which the whiskers extend out from the box. The

option width allows us to set the width of the boxes. The option outline=T determines whether the outliers are to be shown. By default, the boxplots are vertical. The option horizontal=T gives horizontal boxplots. The option notch=T provides notches on each side of the box. If the notches in two boxplots do not overlap, we can conclude that the two medians differ.

We demonstrate this function using the tensile strength data in Example 2.3.4. In the following code we show the overall distribution (line 2), and a break-down by cotton weight percentage (line 3). In line 3 we use a formula as the first argument to the boxplot function, so a separate boxplot of tensile strength is drawn for each level of CWP. The side-by-side boxplots, or parallel boxplots are very useful to show differences between groups (Figure 2.2).

```
par(mfrow=c(1,2))
boxplot(tensile$strength, xlab="cotton weight percentage", ylab="strength",
    border = "blue", col="lightblue")
boxplot(tensile$strength ~ tensile$CWP, xlab="cotton weight percentage",
    ylab="strength", border = "blue", col="lightblue")
```

Except maybe for CWP=15%, the distributions of the strength appear rather skewed. There are three outliers (at 25% and 35%). However, we must keep in mind that there are only five observations per group. What is perhaps most striking about this figure, is that the strength of the tensile increases when cotton weight percentage increases from 15% to 30%, but increasing CWP further causes a sharp drop in tensile strength.

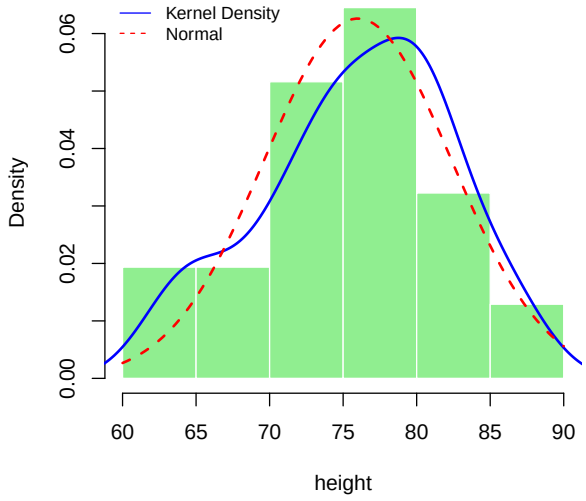

FIGURE 2.1: The trees data – a histogram of height, with a kernel density plot (solid, blue), and a normal density plot (dashed, red).

2.4.3 Stem-and-Leaf Plots

A stem-and-leaf plot is a textual graph of quantitative data. Each data value is split into a stem and a leaf. The plot shows the frequency with which certain classes of values occur. For cases where there are many leaves per stem, we may split the data corresponding to one stem into several rows to enhance readability. We can use a stem-and-leaf plot for

- showing the frequency corresponding to observed data values,
- checking skewness of the data, and
- detecting outliers.

We can use the command stem(x, scale=, width=) where x is a numeric vector. The values assigned to scale and width control the length and width of the plot respectively. The default is scale=1 and width=80.

Using the trees data from Example 2.3.1, the stem-and-leaf plot of `Volume` is obtained as follows:

```
stem(trees$Volume)
```

```
The decimal point is 1 digit(s) to the right of the |

1 | 00066899
2 | 00111234567
3 | 24568
4 | 3
5 | 12568
6 |
7 | 7
```

We observe that the distribution of `Volume` is right skewed. Most of the data values lie in

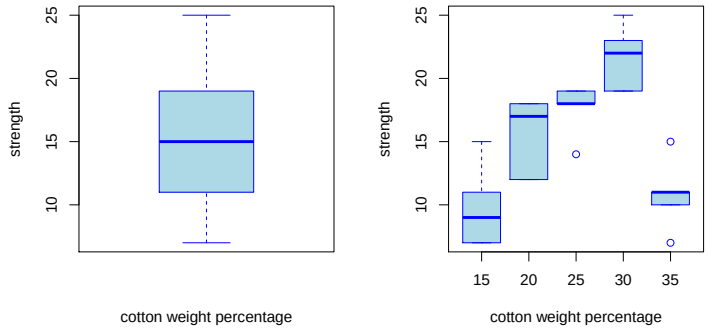

FIGURE 2.2: The tensile strength data: Left - a boxplot using all the data; Right - side-by-side boxplot of tensile strength for each level of `CWP`.

the range 15-25. Further, the last row of the stem plot (data value=77) may be considered an outlier.

2.4.4 Dot Plot or Dot Chart

A dot plot or a dot chart, which is sometimes referred to as a one-dimensional scatterplot, is a simple and compact way to describe univariate data. It consists of a plot of the data along an axis labeled according to the measurement scale. We can plot a dot chart using the command dotchart(x, labels=NULL, groups=NULL, gdata=NULL). Here x is a vector or matrix of numeric values. The option labels provides labels for each point. Grouping the elements of x can be done using the option groups. The option gdata provides the data values for the groups which is usually a summary statistic such as mean or median of each group.

We demonstrate it with the crab mating data (Example 2.3.2). The options lty= and lwd= specifies line type and line width, the default in each case being 1. See for example, Figure 2.3. Try changing the pch option to any number between 1 and 25. Use ?points to see the possible shapes for points in R.

```
with(crabs, {
  y <- table(color, spine)
  dotchart(y, color = c("grey60","grey45","grey30","black"), pch=17,
    main=NA, xlab="Counts")
})
```

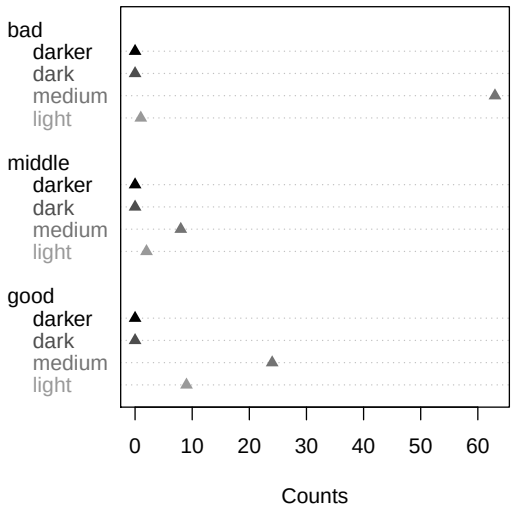

FIGURE 2.3: Dot plot of color of crabs categorized by spine condition

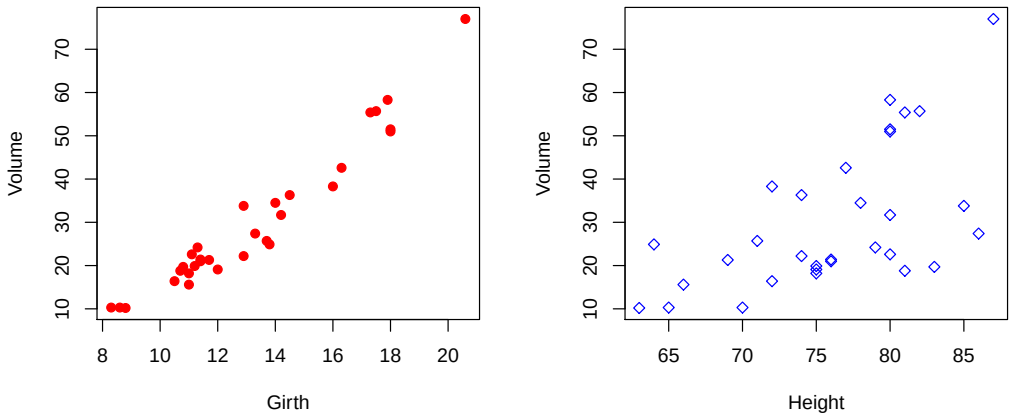

FIGURE 2.4: Scatterplots for the trees data.

2.4.5 Scatterplots

A scatterplot is a graph in which each data point is plotted in a two-dimensional coordinate system. If the points show a tendency to lie about a line with either positive or negative slope we can suspect a linear relationship. To draw a scatterplot of data on two variables x and y, use the command plot(x,y).

Again using the trees data (Example 2.3.1), the variables `Volume` and `Girth` are associated with the same tree and so we expect a relation between them. Likewise, `Volume` and `Height` may also be associated. A scatterplot of `Volume` against `Girth` and a scatterplot of `Volume` against `Height` can be constructed as follows:

```
with(trees, {
  plot(Girth, Volume, main=NA, xlab= "Girth", ylab="Volume", pch=20, col=2,
    cex=1.5)
  plot(Height, Volume, main=NA, xlab= "Height", ylab="Volume", pch=23, col=4)
})
```

From the scatterplot on the left in Figure 2.4, we suspect a linear relationship between `Volume` and `Girth` whereas the points are more randomly scattered in the scatterplot on the right, indicating that a strong linear relationship between `Volume` and `Height` is less likely. Try changing pch and col to different values to see changes in the pattern and color of plotted points.

It is also possible to create a matrix scatterplot (also referred to as scatterplot matrix) in order to visualize relationships between three or more variables. The plot consists of pairwise scatterplots of the variables in a matrix format. Kernel densities (by default) of the variables are shown along the main diagonal. It can be constructed in a single figure by using the scatterplotMatrix() function in the *car* library.

2.4.6 Jittered, and Smoothed Scatterplots

When there are multiple observations with the same x and y values, a jittered scatterplot may improve visualization of the data. This may occur, for example, if the data has been rounded, or if the natural units of observations are integers. By adding a small amount of random noise to each data point, we can avoid plotting different observations on top of one another. Jittering in R is achieved by using the jitter(z) function which modifies the numeric data, z, by a value drawn from a uniform distribution, u$\sim$ Uniform[-a,a]. The default value for a is $d/5$ where d is the smallest difference between pairs of z values. Note that this ensures that the ranking of the original points does not change after applying jitter(). When creating a scatterplot, we may jitter the data in both dimensions using plot(jitter(x), jitter(y)), or just in one (e.g., plot(x, jitter(y))).

When the dataset is very large, a scatterplot may be too 'crowded', and it will be impossible to observe individual points. In such cases, we may want to use the smoothScatter() function in R. It produces a smooth density representation of the scatterplot, showing areas of high density in darker shades than areas of low density.

To see how both features (jittering and smoothing) work, we use simulated data:

```
n <- 10000
x1  <- matrix(round(rnorm(n), digits=0), ncol = 2)
x2  <- matrix(round(rnorm(n, mean = 3, sd = 1.5), digits=0), ncol = 2)
x   <- rbind(x1, x2)
par(mfrow=c(1,3))
plot(x, xlab="x", ylab="y", pch=19, col=4)
smoothScatter(x, xlab="x", ylab="y")
smoothScatter(jitter(x,amount = 0.5), xlab="x", ylab="y")
```

The left plot in Figure 2.5 shows the scatterplot of the data which are normally distributed, but rounded to integers). Note that the true (simulated) distribution is a mixture of two bivariate normal distributions, but because the points overlap, it is impossible to see it in the plot. In the middle plot, we used smoothScatter which recovers the original shape to some extent. The plot on the right is obtained by first jittering the data by a small amount ($a = 0.5$), then, in order to address the fact that there are 10,000 data points to be drawn, we used the smoothScatter command. The resulting plot recovers the true shape of the original data (before rounding), and is not too crowded. Note that in this example, we jittered the data by an amount greater than $d/5$, so the order of the points is not preserved. However, the added noise is random, so the overall properties of the data, such as the shape and the means of the two mixture components are not affected.

2.5 Q-Q plots

Quantile-quantile (Q-Q) plots are an effective visual way to check whether two samples come from the same distribution. They can also be used in order to check if a data set is likely to follow a certain distribution. In this book, we mostly use it to check if the data is from a normal distribution, but the method can be applied to any distribution.

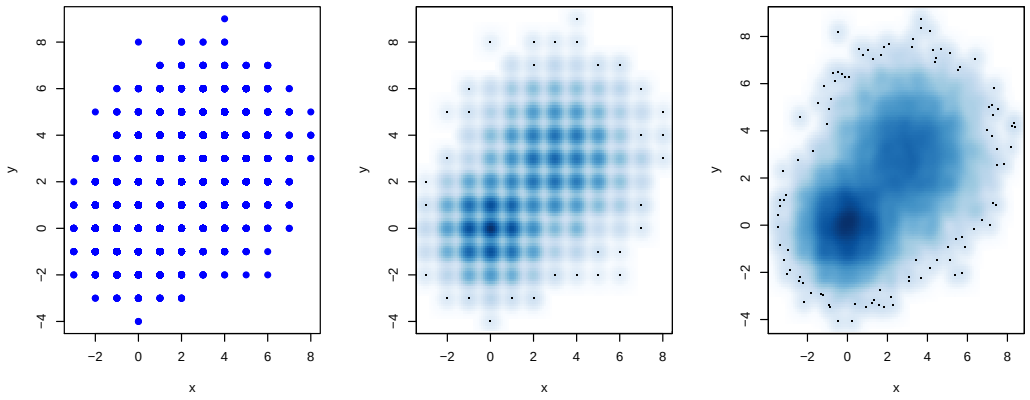

FIGURE 2.5: Simulated data. Left: a scatterplot of the original data (normally distributed, but rounded to integer-valued). Middle: using smoothScatter. Right: using jitter, then smoothScatter.

2.5.1 Empirical Q-Q Plots

Empirical quantile-quantile (Q-Q) plots are useful graphical tools that enable us to compare two data sets via their distribution functions. The empirical Q-Q plot is constructed by plotting the sample quantiles (or the quantiles of the empirical distribution) of one data against the corresponding quantiles (for the same proportion p) of the other. For example, the median and quartiles of one set are plotted against the median and quartiles of the second set.

This plot is extremely simple to construct in the situation where both data sets have the same number of observations. In this case, the empirical Q-Q plot is obtained by plotting the order statistics (empirical quantiles) of one data against the order statistics (empirical quantiles) of the second data. If the two samples have different sizes, interpolation is necessary.

If the two empirical distributions coincide, all of the points in the empirical Q-Q plot would lie exactly on the 45 degree line, y=x; otherwise, departures from this line provide information on how the two distributions differ. If the points lie on *any* straight line, the two data sets have the same shape. If the points show large, systematic departures from a straight line, the two data distributions differ in shape. We use the command qqplot(x, y, plot.it = T, xlab, ylab).

For example, using the crab mating data (Example 2.3.2) we can see that the empirical Q-Q plot (Figure 2.6, left) has no large systematic departures from a straight line, that the distribution of `width` is approximately the same for crabs in good and bad `spine` conditions. The empirical c.d.f. plot can also be used to show that overall, the two distributions are quite similar (Figure 2.6, right). For that, we use the ecdf function. The code used to produce these plots is listed here:

```
with(crabs, {
  par(mfrow=c(1,2))
  qqplot(width[spine=="good"], width[spine=="bad"], plot.it = TRUE, xlab=
```

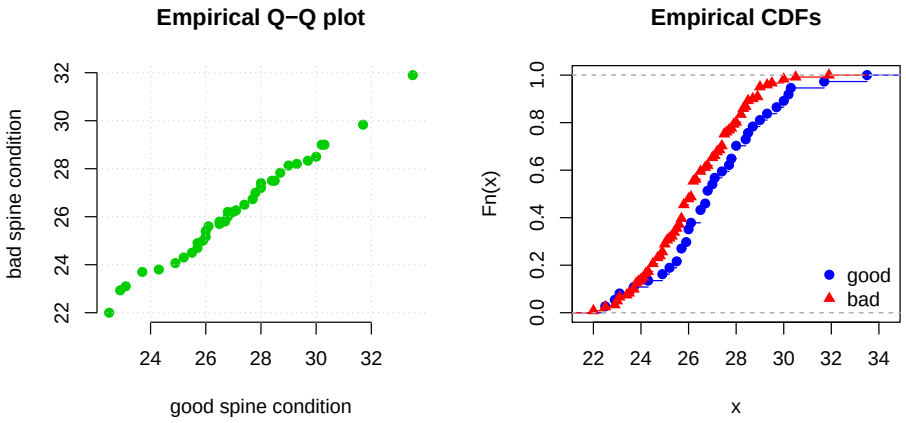

FIGURE 2.6: Left: Empirical Q-Q plot of `width` for crabs with good and bad `spine` condition. Right: Empirical c.d.f. plots for the width in the two groups.

```
    "good spine condition", ylab="bad spine condition", main="Empirical Q-Q
    plot", pch=19, col=3, axes=F )
  axis(1); axis(2); grid()
  plot(ecdf(width[spine=="good"]), col=4, main="Empirical c.d.f.s")
  lines(ecdf(width[spine=="bad"]), col=2, pch=17)
  legend("bottomright",c("good","bad"),col=c(4,2), pch=c(19,17),bty="n")
  par(mfrow=c(1,1))
})
```

Note that we have used the axes=F option to suppress the enclosing box of the Q-Q plot, and then added only the x and y axes by using axis(1) and axis(2).

2.5.2 Normal Q-Q Plot

The normal quantile-quantile, or normal Q-Q plot is routinely used for checking the assumption that a random data sample is generated from a normal population, and is a standard graphical feature in almost any statistical software. Rather than plotting the quantiles of two groups as we did in the previous subsection, in a normal Q-Q plot we show the quantiles of one given sample versus the corresponding quantiles of a theoretical distribution, but the procedure is the same, and so is the interpretation.

Let $X_1, \ldots, X_n$ denote the observations in a random sample, and let $x_{(1)} \leq x_{(2)} \ldots \leq x_{(n)}$ denote the ordered values or empirical quantiles $Q_E(p_i)$ corresponding to $p_i = (i - 0.5)/n$. Our interest lies in assessing whether the empirical data distribution $F_n(x)$ matches the standard normal c.d.f. $\Phi(z)$. To do this, we use the normal Q-Q plot which plots the sample or empirical quantiles for the data against the corresponding theoretical quantiles from a standard normal distribution (i.e., for the same p_i).

If the sample is generated from a standard normal population with c.d.f. $\Phi(z)$, the n points on this Q-Q plot should lie approximately along the $y = x$ line, i.e., a straight line

through the origin and with a slope of 1. In this case, the empirical distribution matches the standard normal, $N(0,1)$ distribution.

Suppose the points fall along the $y = c_1 + c_2 x$ line, where $c_1 \neq 0$ and $c_2 \neq 1$. Then, the $N(c_1, c_2^2)$ distribution is a match for the empirical distribution. In other words, the data matches a normal distribution with mean c_1 and standard deviation c_2. This follows because the normal distribution is in the location-scale family of distributions.

Patterns of departures of the points from linearity indicate the nature of departure of the empirical distribution from normality. In general, departures from linearity in the normal Q-Q plot can indicate properties such as skewness, heavy-tailed behavior, or multi-modality in the empirical distribution.

For example, we will use Example 2.3.5 in order to check whether number of households per town in the state of Connecticut has a normal distribution. In the following code, we use the *car* package to draw a normal Q-Q plot with a 95% confidence interval.

```
CTtowns <- read.csv('Data/total-households-town-2019.csv', header=TRUE)
library("car")
par(mfrow=c(1,2))
qqPlot(CTtowns$Value, main="A", ylab="Households", cex=0.6,pch=19,
    col="red", col.lines = "orange")
qqPlot(log10(CTtowns$Value), main="B", ylab="log10(Households)",
    cex=0.6,pch=19, col="green", col.lines = "lightblue")
par(mfrow=c(1,1))
```

In panel A in Figure 2.7 (left) we see that the empirical quantiles and the normal quantiles do not line up, which means that the number of households in towns in Connecticut are not normally distributed. However, after a log transformation, the points in the normal Q-Q plot lie very neatly along a straight line. This means that the number of households very likely follows a log-normal distribution. Notice that in panel B of Figure 2.7 (right), the mean of the standard normal distributions (0) matches approximately 3.7 in the distribution of the log-transformed (base 10) number of households, which means that the median number of households in Connecticut is approximately $10^{3.7} \approx 5000$.

Remark 1. A theoretical Q-Q plot graphically assesses whether the empirical data distribution $F_n(x)$ matches *any* user-specified theoretical probability distribution with c.d.f. $F(x)$ (not necessarily the standard normal c.d.f.) by plotting the empirical quantiles $Q_E(p_i)$ of the random sample (its order statistics) against the corresponding theoretical quantiles $Q_T(p_i)$ from the hypothesized probability distribution. If the hypothesized theoretical distribution is a close approximation to the empirical data distribution, the data quantiles should closely match the theoretical quantiles, and the points in the plot should lie on, or reasonably close to, the $y = x$ line. This line, which passes through the origin with slope 1 can be regarded as the null, or reference configuration (when the empirical and theoretical distributions match exactly). Departures of points from this line indicate lack of fit of the assumed theoretical distribution to the observed data.

Remark 2. We can simulate data from other distributions such as a chi-squared distribution with ν d.f., or Student-t distributions with different values for the d.f. ν. In each case, we can construct the normal Q-Q plot using the qqnorm() function. Using the R function qqplot(), we can also construct Q-Q plots corresponding to the generating distributions instead of the normal distribution.

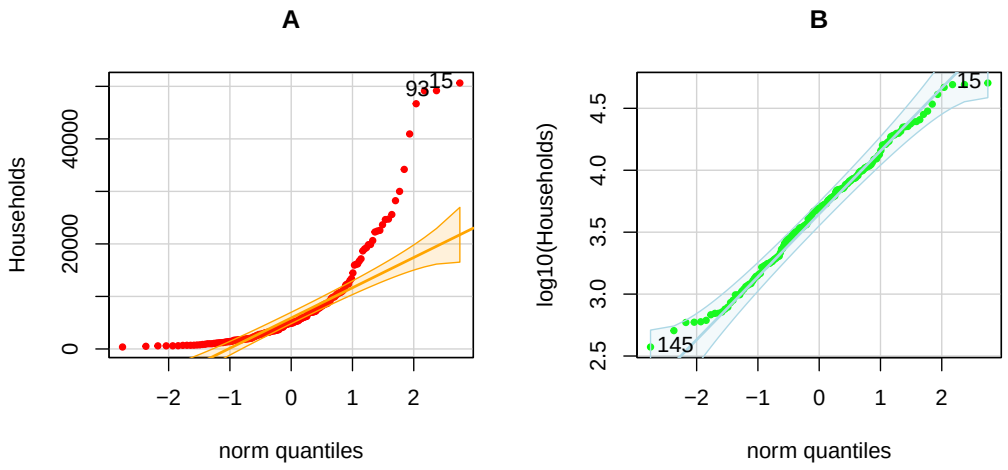

FIGURE 2.7: Left: Normal Q-Q plot for the number of households in towns in Connecticut. Right: Normal Q-Q plot for the log transformed number of households in towns in Connecticut.

2.6 Visualization of Qualitative Data

We describe the pie diagram or pie chart for one qualitative variable followed by a segmented bar graph to represent two qualitative variables.

2.6.1 Pie Diagram

A pie chart or a pie diagram is often used to represent the frequency of each categorical data value as a percentage of the total number of observations. We construct a circle (which spans 360° at the center), then divide the circle into sectors, one for each distinct category value, with the size of any given sector being proportional to the percentage (or relative frequency) of that category. Note that the complete circle represents the total number of measurements. We determine the angle of each slice by multiplying the relative frequency by 360°.

```
with(crabs, {
  spinetable <- table(spine)
  pie(spinetable, labels = rownames(spinetable),
    col=rainbow(length(spinetable)), main="A")
})
```

The left plot in Figure 2.8 shows the pie chart for spine condition of the crabs (Example 2.3.2). The spine condition for most of the crabs appears to be either worn or broken. Try heat.colors(n), terrain.colors(n), topo.colors(n), and colors(n) instead of using the rainbow option. The command colors() returns the 657 color names available in R. To see these

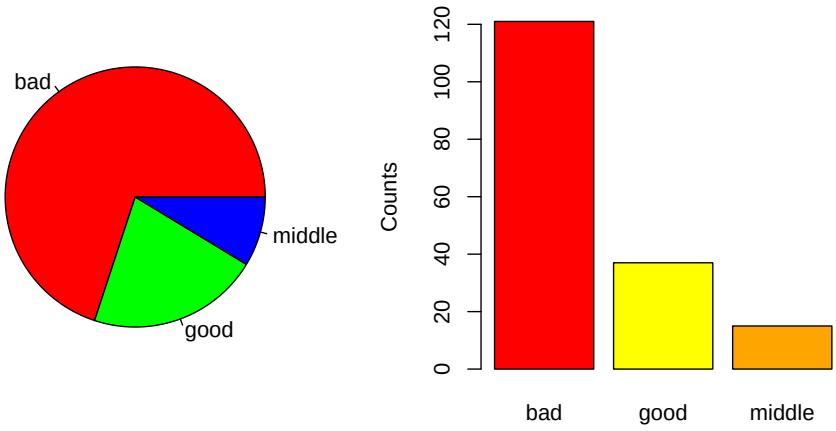

FIGURE 2.8: Left: Pie diagram showing the crabs categorized by `spine` condition. Right: Barplot of crabs categorized by `spine` condition.

names, refer to the color chart in Glynn (2005). The dot plot or the bar graph (below) are preferred by users over the pie chart (Cleveland (1985), p. 264).

2.6.2 Barplots

A barplot or a barchart is a simple graphical tool for visually summarizing qualitative data, i.e., data that may be classified into one of several, say K categories. The *frequency* of each category is the number of measurements in that category. The *relative frequency* of each category is the proportion of data in that category, and is the the ratio of the frequency of the category to the total number of measurements. To construct a barchart, we show categories on the x-axis, place corresponding frequencies (or relative frequencies) on the y-axis, and construct vertical bars of equal width, one for each category. The height of a bar is proportional to the frequency (or relative frequency) of that category. There are many variations of the barchart, for instance, the bars may be displayed horizontally or vertically.

Figure 2.8, right, shows the distribution of the `spine` condition of the crabs as a bar chart (Example 2.3.2).

```
with(crabs, {
  spinetable <- table(spine)
  barplot(spinetable, beside = F, col=c("red","yellow","orange"), ylab =
    "Counts", main ="B")
})
```

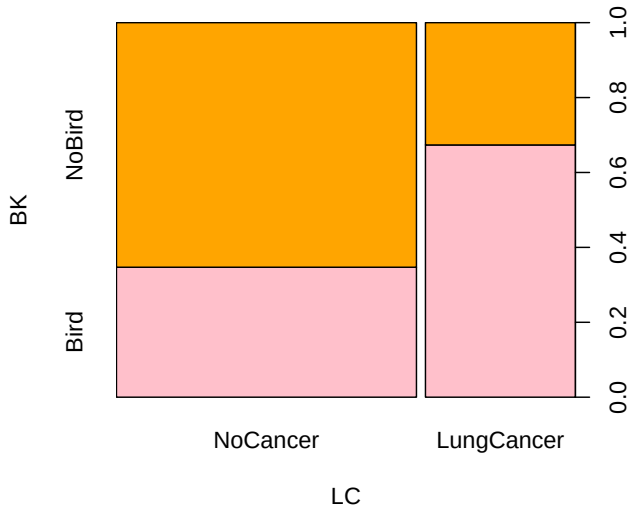

FIGURE 2.9: Spine plot of bird keeping versus lung cancer

2.6.3 Spine Plots and Spinograms

The relationship between two categorical variables can be explored using spine plots. For a spine plot, both quantities are approximated by the corresponding empirical relative frequencies. When x is numerical, spineplot(x, y) gives spinograms. For the spinogram, we first discretize x by using hist() with the breaks option, and then compute empirical relative frequencies. Spinograms may be viewed as extended stacked histograms. Spine plots may also be viewed as generalizations of stacked barplots, where the widths of the bars (and not their heights) correspond to the relative frequencies of x. The heights of the bars then correspond to the conditional relative frequencies of y in every x group. The partitions within the bars are the conditional relative frequencies of y in each x group which are labeled on the left axis and quantified on the right axis. To draw a spine plot, use the commands spineplot(x,y) or spineplot(y∼x), where y is categorical and x can be categorical or numerical.

For example, using the bird keeping data (Example 2.3.3), we draw a spine plot to investigate whether bird keeping is related to lung cancer:

```
with(birds, {
  y <- table(LC, BK)
  spineplot(y, col = c("pink","orange"), main=NA)
})
```

In Figure 2.9, the partitions within the bars are the proportions of a given status of bird keeping having lung cancer or not. It appears that individuals who have not kept birds are less prone to lung cancer.

2.7 Assessing Goodness of Fit

We saw some visualization methods which show us how data from a sample are distributed, or allow us to compare two or more populations. We will often need to check more formally whether some assumption or hypothesis we make is supported by the data. To do that, there are many so-called 'goodness-of-fit' tests.

Most statistical software include several significance tests for normality. In each of these tests, the null hypothesis H_0 is that the data come from a normal distribution. While the alternate hypothesis negates H_0 and claims that the data is not from a normal distribution (without specifying an alternative distribution that could have generated the data). Thus, after rejecting H_0, the user resorts to graphical methods to see how the data departs from normality in order to either transform the data to normality, or to seek an alternative distribution that might better fit the data. The significance tests share this disadvantage that, if the sample size n is sufficiently large, the test may detect even trivial departures from the null hypothesis. In such situations, although there may be some statistically significant effect to claim that the data departs from normality, the departure may be irrelevant to be of practical significance. A normal Q–Q plot may be preferred in such cases. The plot, which we covered in detail in 2.5, will also show us how many points depart from the straight line pattern, and in what way they depart. The overall takeaway message is not to rely on any one approach to assess normality of a given data set. Visually understanding the data distribution, the normal Q-Q plot and powerful significance tests must be used in conjunction. As we saw in Figure 2.7B with the Connecticut Towns data (2.3.5), after a suitable transformation the data lie along a straight line, supporting the normality assumption.

The Shapiro-Wilk test is based on a comparison of ordered sample values with their expected locations under the null hypothesis of normality Shapiro and Wilk (1965). If the data are normal, we expect that the ordered observations $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$ are linearly related to the means of an ordered sample from a standard normal distribution. The test statistic SW can be expressed as the ratio of two estimates of the variance of a normal distribution based on the sample data, and SW is roughly a measure of the straightness of the normal Q-Q plot. Hence, the closer SW is to one, the more normal the sample is; otherwise, we reject the normality assumption for the data.

```
shapiro.test(log(CTtowns$Value))
```

Based on the observed p-value of 0.2966, we conclude that the data supports the normality assumption.

```
Shapiro-Wilk normality test

data:  log(CTtowns$Value)
W = 0.9902, p-value = 0.2966
```

Another class of tests called EDF tests for normality is based on the empiricial c.d.f. function of the random sample, and include the Kolmogorov-Smirnov, Anderson-Darling and Cramér-von Mises tests (D'Agostino and Stephens (1986)). For instance, the Kolmogorov-Smirnov goodness of fit test statistic is defined as the maximum vertical distance between

the empirical c.d.f. $F_n(x)$ of the sample data and the c.d.f. of a normal distribution with estimated mean $\overline{Y}$ (the sample mean) and standard deviation s (the sample standard deviation). We reject normality if the test statistic is large. This class of tests is more general than the Shapiro-Wilk test, and can be used with any continuous distribution, not just the normal distribution.

```
ks.test(log(CTtowns$Value), pnorm,
    mean(log(CTtowns$Value)),sd(log(CTtowns$Value)))
```

```
  Asymptotic one-sample Kolmogorov-Smirnov test

  data:  log(CTtowns$Value)
  D = 0.038439, p-value = 0.9641
  alternative hypothesis: two-sided
```

We use the R package *nortest* to use the Anderson-Darling and Cramér-von Mises tests for normality.

```
library(nortest)
ad.test(log(CTtowns$Value))
cvm.test(log(CTtowns$Value))
```

```
  Anderson-Darling normality test

  data:  log(CTtowns$Value)
  A = 0.28396, p-value = 0.6275


  Cramer-von Mises normality test

  data:  log(CTtowns$Value)
  W = 0.036922, p-value = 0.7361
```

The chi-squared goodness of fit test procedure is useful for testing whether a data sample comes from a population with a specific distribution. It can be employed to *any* univariate distribution whose c.d.f. $F(x)$ is known.

We group the sample data into k groups and then test the null hypothesis H_0 that the grouped sample data follows a multinomial distribution with k categories defined by (2.7). The alternate hypothesis is that the data do not follow the distribution specified under the null hypothesis. To carry out the test, we first calculate the expected frequencies corresponding to each of the k groups under the multinomial distribution specified under H_0 as

$$E_i = np_i, \quad i = 1, \ldots, k. \tag{2.8}$$

The chi-squared goodness-of-fit test statistic is

$$\chi^2_{gof} = \sum_{i=1}^{k} \frac{(C_i - E_i)^2}{E_i}, \tag{2.9}$$

which follows an approximate χ^2_{k-r} distribution under some conditions Snedecor and Cochran (1989). Here, r denotes the number of unknown parameters that must be estimated from the data. A large test statistic leads to rejection of the null hypothesis and the normality assumption.

The code and output shown below confirm that `log(Value)` satisfies the normality assumption.

```
pearson.test(log(CTtowns$Value))
```

```
  Pearson chi-square normality test

  data:  log(CTtowns$Value)
  P = 8.3254, p-value = 0.8218
```

2.8 Final Remarks

To conclude this chapter, it is important to note that there are many options for summarizing and visualization to choose from, and new approaches and functions are created frequently. While it is not possible to cover all the available plotting tools, it is very important to emphasize the importance of generating intuitive, appealing plots. Doing this often requires trying many things, changing settings such as scale, colors, and so forth. For further reading and resources, we recommend the comprehensive book *R Graphics* (second edition), by Paul Murrell, which is available at `https://www.stat.auckland.ac.nz/~paul/RG2e/`. We also recommend use of the *ggplot2* package which has several flexible options. The use of this package is explored in the exercises, and more extensively in the following chapters. For a quick reference, see the online cheatsheet, at `https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf`.

Note that all the graphics can be saved as files with commonly used file types such as eps, pdf, jpg, svg, etc. For a more elaborate illustration of base R's capabilities for data visualization, refer to Rahlf (2017).

3

Two Sample Inference

This chapter describes how to compare characteristics from two populations. By characteristics, we mean location parameters such as the means or medians, or dispersion parameters such as variances, or even the entire data distributions. We describe parametric and nonparametric statistical procedures that will help us answer research questions of interest about these parameters. We show R code for carrying out data analysis for matched pair samples and independent samples.

3.1 Introduction

Visualization gives a good initial understanding of the data. In many situations, an important next step is to formalize and validate the assessments obtained from data visualization using statistical inferential procedures, including estimation and hypothesis testing. There are two general approaches for inference, namely *parametric* and *nonparametric*. If one chooses to *assume* that in the entire population, a variable of interest has a certain distribution which is determined by a set of parameters, it is said to be a parametric modeling framework. The parameters are unknown, but if they can be estimated accurately from a finite sample and if the assumed distribution is the correct one, then hypotheses can be expressed and tested by using the explicit form of the distribution. In a nonparametric modeling framework, there is no mathematical specification for the probability distribution of the data and the only information that is available is that the data is a random sample from a population with perhaps certain descriptive characteristics, such as symmetry, say. The parametric framework is usually more powerful in the sense that one can detect more subtle effects. However, it relies on rather strong assumptions about the exact form of the distribution in the population. Quite often, the choice of the assumed distribution is reasonable, but in order to use this inferential approach, we need to check the validity of the assumption. If the assumptions required for employing the parametric approaches seem to be invalid, suitable non-parametric approaches may be used.

In this chapter, we consider the question of comparing two populations. This can mean many things – for example, do the two populations have the same *center* (e.g., mean, median)? Do they have the same *spread* (e.g., variance)? These are the questions we discuss in this chapter, but we may also ask more general questions and, for example, compare the populations in terms of differences in some percentiles.

3.2 Essential Terminology and Notation

✓ A **population parameter** is a numerical descriptive measure of a population, usually unknown, and must be estimated from the sample data. It is usually denoted by a Greek letter. For example, if a random variable Y follows a normal population, $Y \sim N(\mu, \sigma^2)$, the population mean μ and the population variance σ^2 (or standard deviation σ) are parameters.

✓ A **sample statistic** is a numerical descriptive measure calculated based on sample observations and is used to estimate an unknown population parameter. It is denoted by a Latin letter. For example, the sample mean $\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$, the sample variance $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \overline{Y})^2$, and the sample standard deviation S, are sample statistics, which are based on a random sample $Y_i,\ i = 1, 2, \ldots, n$, drawn from the population of interest.

✓ The **sampling distribution** of a sample statistic based on n sample observations is the probability distribution of the sample statistic. For example, the sampling distribution of the sample mean $\overline{Y}$ based on a sample of size n, is $N(\mu, \sigma^2/n)$. Commonly used sampling distributions are the Student's t, chi-square, and F-distributions.

✓ The **standard error** of a sample statistic is the standard deviation of the sampling distribution of that statistic.

✓ A **point estimate** of an unknown population parameter is a rule or formula that enables us to obtain a numeric value from the sample data as an estimate for that parameter. For example, a point estimate of the population mean μ is the sample mean $\overline{Y}$.

✓ If the mean of the sampling distribution of the sample statistic is equal to the corresponding population parameter, the sample statistic is called an **unbiased** estimate of the population parameter. Otherwise, it is called a **biased** estimate. The sample mean $\overline{Y}$ is an unbiased estimate of μ and the sample variance S^2 is an unbiased estimate of σ^2. However, S is a biased estimate of σ.

✓ A **statistical hypothesis** is a claim or statement about some population characteristic.

✓ A **hypothesis test** is a statistical method to determine which of two contradictory hypotheses is supported by the data. It is a formal way to distinguish between two hypotheses based on available data.

✓ A **null hypothesis** is denoted by H_0, and is the hypothesis of no difference, a claim asserting no change or no effect in the state of nature. Unless data provides convincing evidence that it is false, H_0 is not rejected. A null hypothesis has the form H_0: population parameter = hypothesized value.

✓ An **alternate or alternative hypothesis** is a statement that contradicts H_0 and is denoted by H_1 or H_a. It is accepted only if data provides convincing evidence of its truth. An alternative hypothesis may be one-sided, H_1: population parameter > hypothesized value (upper-tailed), or H_1: population parameter < hypothesized value (lower-tailed),

or two-sided, H_1: population parameter $\neq$ hypothesized value.
For example, suppose we wish to test the equality of means of two populations,

$$H_0 : \mu_1 = \mu_2 \text{ or } H_0 : \mu_1 - \mu_2 = 0$$

versus a two-sided alternative hypothesis

$$H_1 : \mu_1 \neq \mu_2 \text{ or } H_1 : \mu_1 - \mu_2 \neq 0.$$

We can also consider a one sided alternative hypothesis, i.e.,

$$H_1 : \mu_1 < \mu_2 \text{ or } H_1 : \mu_1 > \mu_2.$$

✓ Under the Neyman-Pearson paradigm in hypothesis testing, we decide whether or not to reject the null hypothesis H_0 based on the evidence presented by a random sample through a sample statistic, say T (sample mean, sample variance, etc.). We compute a test statistic and determine a rejection region.

✓ A **test statistic** is a quantity computed from the sample and is used to make a decision in a test of hypothesis.

✓ A **rejection region** (or critical region) consists of all values of the test statistic T for which H_0 will be rejected, i.e., values that provide strong evidence in favor of the alternative hypothesis H_1.

✓ Two types of error may occur in implementing a hypothesis test. The **type I error** is the error from rejecting H_0 when H_0 is true, i.e., the error of rejecting a true null hypothesis.

✓ The probability of type I error is P(Reject H_0 when H_0 is true) and is the **level of significance** of the test, denoted by α (which lies between 0 and 1 and usually takes a value like 0.05 or 0.01).

✓ The **type II error** is the error of not rejecting H_0 when H_0 is false, i.e., the error of not rejecting a false null hypothesis. The probability of type II error is P(Do not Reject H_0 when H_0 is false), and is usually denoted by β.

✓ The **power** of a statistical test is P(Reject H_0 when H_0 is false) and is denoted by $1-\beta$. Although one would like both α and β to be close to zero, this is impossible except in trivial situations, and α and β are inversely related.

✓ The **significance level** α is fixed in advance at a small value (e.g., $\alpha = 0.05$ or 0.01), and a test is constructed to yield a small value of β, or a large power $1 - \beta$.

✓ The ***p*-value** of a test is the smallest level of significance at which H_0 can be rejected.

✓ A **confidence interval** (C.I.) estimate may also be used to make a decision on a hypothesis test. A confidence interval estimate consists of an interval of values computed from the sample data that estimates a population parameter. Associated with an interval estimate is a **confidence coefficient**, $(1 - \alpha)$, which represents the confidence we place on the estimator and is the probability that an interval estimator contains the population parameter if this estimator is obtained repeatedly a very large number of

times. The **confidence level** is the confidence coefficient expressed as a percentage. So, if $\alpha = 0.05$, then $(1 - \alpha) = 0.95$, and the confidence level is 95%. Note that in general, the confidence interval estimate for any population parameter is computed using this general formula: Point Estimate of the parameter $\pm$ (critical value) $\times$ (Standard error of the Point Estimate).

✓ **Relation between C.I. and hypothesis test.** If the $100(1 - \alpha/2)\%$ C.I. for the true parameter does not include the value hypothesized under H_0 (say 0), we conclude that the data provides evidence to reject H_0 at the level of significance α. On the other hand, if the C.I. includes the value hypothesized under H_0, we conclude that the data does not provide evidence to reject H_0 at the $100\alpha\%$ level of significance.

✓ The number of **degrees of freedom**, (d.f.), is the number of observations in a sample that are independently available to estimate an unknown parameter of the population from which that sample is drawn. For example, suppose we have $n = 20$ observations in the sample. For estimating the population mean μ by the sample mean $\overline{Y}$, we must use all the n observations. However, for estimating the population variance σ^2 using the sample variance S^2, we lose one degree of freedom for estimating the mean μ by $\overline{Y}$, and only have $n - 1 = 19$ independent observations. The d.f. associated with S^2 is $n - 1$.

✓ **Effect size** tells us how meaningful the relationship between variables or the difference between groups is, i.e., it indicates the practical significance of a research outcome. A large effect size means that a research finding has practical significance, while a small effect size indicates limited practical applications.

3.3 Continuous Variable – Matched Pairs

U.S. Crime

Consider the following example:

Example 3.3.1 (U.S. Violent Crime Data) The Federal Bureau of Investigation (FBI) website on Uniform Crime Reporting[1] contains historical data about crime in the United States, from 1995 to the present (at the time of writing this book the last year covered is 2019). The data consists of rates of violent crimes per 100,000 inhabitants in each US state.

As stated on their website, "the FBI's interactive Crime Data Explorer tool serves as the digital front door for UCR data, enabling law enforcement and the general public to more easily use and understand the massive amounts of UCR data currently collected". The UCR Program consists of four data collections: The National Incident-Based Reporting System (NIBRS), the Summary Reporting System (SRS), the Law Enforcement Officers Killed and Assaulted (LEOKA) Program, and the Hate Crime Statistics Program. The UCR Program publishes annual reports for each of these data collections and a preliminary semiannual report of summary data each winter, as well as special compilations on cargo theft, human trafficking, and NIBRS topical studies.

An annual publication for more than eight decades, the Crime in the United States report contains a compilation of the volume and rate of violent and property crime offenses for the nation and by state using Summary Reporting System data and summarized data

[1] `https://ucr.fbi.gov/crime-in-the-u.s/`

from the National Incident-Based Reporting System (NIBRS). Data at the level of local law enforcement agencies are also provided for those contributors supplying 12 months of complete offense data. This report includes arrests, clearances, trends, and law enforcement employee data. The NIBRS website gives crime data for several years on many different categories.

In this discussion, we consider data for all the US states from two years, 2010 and 2019, on rates corresponding to two categories, i.e., Violent Crime (which consists of four subcategories - murder, rape, robbery, and aggravated assault) and Property Crime (with three subcategories). The table also gives the estimated population of each state.

Research question #1

Is there an overall difference in violent crime rate between 2010 and 2019?

As mentioned earlier, we have to state the question more precisely. We may ask whether the *mean* (or *median*) violent crime rate in 2019 is different than the mean (or median) violent crime rate in 2010. Furthermore, we have to state our assumptions, which will determine the inferential procedure that will be used. For example, are we willing to assume that

- the data from two years are independent within each state?
- violent crime rates in different states are independent?
- the violent crime rate has a normal distribution in both years?
- the dispersion of the data is the same in both years?

In this case, common sense suggests that the crime rates in different years within each state are *not* independent. One reason for this assertion is that the population itself, and therefore most characteristics of the population in each state remain about the same from one year to the next. For example, population size, median income, poverty rate, and police force size, are things that change slowly and gradually over time.

Across states we may *assume* that violent rates are independent, but in reality things may be more complicated because criminals can cross state lines. As opposed to cyber crimes which are not bound by state lines or require any travel, we may argue that it is reasonable to *assume* that most violent crimes occur near the perpertrator's residence, and so violent crime rates are independent across states.

Since crime rates in different years within each state are dependent, we treat the data as a *paired sample*, meaning that from each state we obtain a matching pair of observations, Y_1 and Y_2. Our research question can then be expressed in terms of the differences between the years,

$$d_j = Y_{2,j} - Y_{1,j}$$

where $j = 1, \ldots, 50$ is the state index. In doing so, we have now obtained an independent sample with a single observation from each state. Intuitively, if the answer to our research question is "no", that is, there is no difference in violent crime rate between the two years, then we expect the values of d_j to be centered around 0. In other words, it is equally likely to observe a state with a higher violent crime rate in 2010 as it is to observe a state with a higher violent crime rate in 2019.

Having stated the research question and the assumptions, our next step is to calculate

a sample statistic, state the formal null and alternative hypotheses in terms of an underlying parameter, and use the statistic's probabilistic properties to perform the inferential procedure and draw our conclusion.

Whether we are willing to make the assumption that the d_j are normally distributed or not determines which methods may be used. The following table contains three possible methods[2]: Fisher's sign procedure and Wilcoxon's signed-rank procedure may be used when we do not make the normality assumption, and the t-procedure may be used when we do. The methods appear in the table according to their statistical power (increasing from left to right). All three methods use the differences between cohorts, d_j, but with each method a different sample statistic is defined, each with a different sampling distribution. In the table, n is the sample size (e.g., $n = 50$ states) and the following notation is used:

$$\begin{aligned} \overline{d} &= \frac{\sum_{j=1}^{n} d_j}{n} \\ S_d &= \frac{\sum_{j=1}^{n} (d_j - \overline{d})^2}{n-1} \\ \psi_i &= 1 \text{ if } d_i > 0, \text{ and 0 otherwise} \\ R_j &= \text{rank of } |d_j|. \end{aligned}$$

Note that for R_j we order the absolute differences in ascending order and assign ranks from 1 to n, or assign average rank in case of ties.

Method	Fisher's sign test	Wilcoxon's signed-rank test	t-test
Assumptions	None	Symmetry	$d_j \overset{iid}{\sim} N(0, \sigma^2)$
Inference about	Median (θ_d)	Median (θ_d)	Mean (μ_d)
Statistic	$B = \sum_{i=1}^{n} \psi_i$	$T^+ = \sum_{i=1}^{n} \psi_i R_i$	$t = \frac{\overline{d}}{S_d/\sqrt{n}}$
Sampling distribution	$\frac{B - n/2}{\sqrt{n/4}} \approx N(0,1)$	$\frac{T^+ - n(n+1)/4}{\sqrt{n(n+1)(2n+1)/24}} \approx N(0,1)$	Student's t_{n-1}
Null hypothesis	$H_0 : \theta_d = 0$	$H_0 : \theta_d = 0$	$H_0 : \mu_d = 0$
R function	BSDA::SIGN.test	wilcox.test	t.test

Key formulas

The alternative hypotheses in the case of the U.S. violent crime data are $H_0 : \theta_d \neq 0$ for the two non-parametric procedures, and $H_0 : \mu_d \neq 0$ for the parametric procedure.

The U.S. violent crime data from Example 3.3.1 were made available by the FBI in the form of Excel files, so we will use the *readxl* package to read them. Note that as is often the case in the life of a data scientist, loading the data requires looking at it first, and then some manipulations may be needed. In this case there are several things to note.

1. Each file begins with a few lines which describe the content, but contain no data to be extracted. Similarly, the last few lines contain notes from the data provider. These are important details which should be considered in the analysis and the interpretation of the results, but are not loaded into a data table. To account for these lines, we use the 'range' option of the read_excel() command (see line 2 in the code below).

[2] All these methods assume a continuous distribution

2. The column names may span over multiple lines. For example, the column we are interested in (Violent crime) has a newline character in it. This may cause problems when we refer to column names in a data table in R, so we replace the newline with and underscore (see the usage of the gsub() function in line 3).

3. Also note that the column and row names are not necessarily the same across years! In the 2019 data, the fifth colum in the data contains a superscript. Also, the 2019 file contains data for Puerto Rico, which means that the range we use to read in the data must be changed accordingly.

4. Many rows are not used in our analysis. We extract the rows for which the Area column contains 'State Total' (see the usage of the which() function in line 6 in the code).

5. The state names appear in a different row than the U.S. violent crime data. We have to match the state name to its total violent crimes value. Therefore, we first get all the state names (line 4) and omit all the blank rows. We note that the ninth value is 'District of Columbia' and we choose to omit it (line 5). Note that for D.C. the Area column contains 'Total', and not 'State Total', so when we use the criterion in line 6 to obtain violent crime values, we only get them for the 50 states. Similarly, Puerto Rico is included in the 2019 data, but it is also excluded when we use the 'State Total'.

```
library("readxl")
ViolentCrime2010 <- read_excel("Data/Violent Crime-by
    state-2010-table-5.xls", sheet=1, range = "A4:M507", trim_ws = T)  #
    comment
colnames(ViolentCrime2010) = gsub("\n","_",colnames(ViolentCrime2010))
States <- as.character(na.omit(ViolentCrime2010$State))
States <- States[c(1:8,10:51)]
VCrimeRate2010 <- ViolentCrime2010$Violent_crime[which(ViolentCrime2010$Area
    == "State Total")]
```

Statistical methods for comparing paired data, i.e., the Fisher's sign, Wilcoxon signed-rank, and paired t-procedures are described in the appendix. Before choosing a method to check whether violent crime rates have changed from 2010 to 2019, we must look at the data distribution. Figure 3.1, left, shows the histogram of violent crime rates in the 50 states in 2010. It is clearly skewed, which renders both the Wilcoxon and t-procedures invalid (since former requires symmetry, and the latter requires normality). After a log transformation, the data appear to be normally distributed, as shown in Figure 3.1, middle. A Q-Q plot is useful to check if the normality assumption is reasonable, and indeed, Figure 3.1, right, shows that after a log transformation all the points lie close to a straight line, suggesting that the normality assumption is valid. We can also perform the Shapiro-Wilk test for normality, which gives a p-value of 5.9e-9 for the original data, and 0.68 for the log-transformed data, leading to the conclusion that the hypothesis that the original data are normally distributed should be rejected, while there is no evidence to reject the normality of the log transformed data. The code used to produce Figure 3.1 and the Shapiro-Wilk test is as follows.

```
par(mfrow = c(1, 3))
hist(VCrimeRate2010, main = NA, xlab = "Violent crimes", col = "grey",
    border = "white")
hist(log10(VCrimeRate2010), main = NA, xlab = "Violent crimes (log10)", col
    = "grey", border = "white")
```

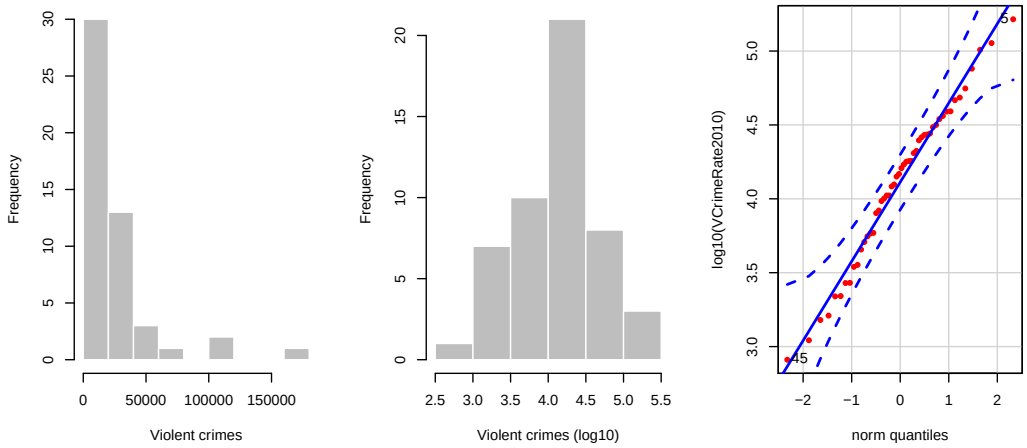

FIGURE 3.1: Left: the distribution of the original violent crime data. Middle: the distribution of the log-transformed violent crime data. Right: Q-Q plot of the log-transformed violent crime data.

```
qqPlot(log10(VCrimeRate2010), main = NA, pch = 19, col = 2, cex = 0.7)
shapiro.test(VCrimeRate2010) # p=5.901e-09
shapiro.test(log10(VCrimeRate2010)) # p=0.6758
```

Figure 3.2 shows the log-transformed violent crime rates in the 50 states in 2010 on the x-axis and 2019 on the y-axis. The points lie very close to the 45° line, indicating that the crime rates were approximately the same in both years.

To carry out Fisher's sign procedure, we look at the differences $d_i = Y_{i,1} - Y_{i,2}$ of the paired data, for $i = 1, \ldots, n$. Let θ_d be the median of the population of differences. We define indicator variables $\psi_i = 1$ if $d_i > 0$ and $\psi_i = 0$ if $d_i < 0$ and obtain the sum of positive signs:

$$B = \sum_{i=1}^{n} \psi_i. \tag{3.1}$$

If H_0 were true, the test statistic B should follow a binomial distribution with success proportion $1/2$. The criteria to test $H_0 : \theta_d = 0$ are shown in Table 3.1. See the code below.

```
SIGN.test(log10(VCrimeRate2019), log10(VCrimeRate2010)) # exactly the same
    as with no transformation
```

The Wilcoxon Signed-Rank test uses both (a) the signs of the differences, ψ_i and (b) the ranks of the magnitudes of the differences, R_i. The test statistic is the sum of the positive signed ranks:

$$T^+ = \sum_{i=1}^{n} \psi_i R_i. \tag{3.2}$$

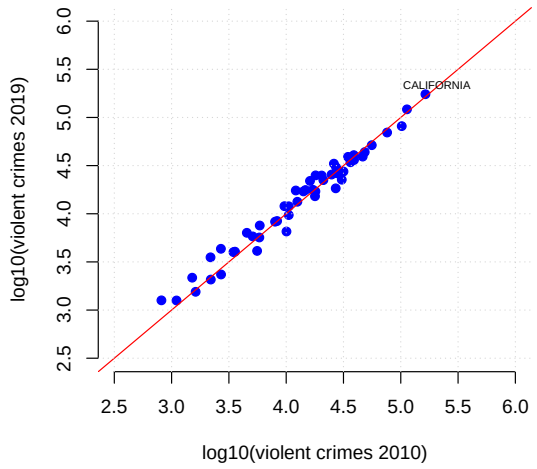

FIGURE 3.2: The violent crime data in 2010 vs. 2019 (log-transformed).

The decision rules based on T^+ for testing $H_0 : \theta_d = 0$ are given in Table 3.2, and the R code is shown below.

```
wilcox.test(log10(VCrimeRate2019), log10(VCrimeRate2010), paired = T)
```

For the paired t-procedure, we assume that the differences $d_1, \ldots, d_n$ come from a normal population with mean μ_d and variance σ_d^2, and we use the paired sample data to test the null hypothesis $H_0 : \mu_d = 0$. This enables us to compare the true means of the paired samples using the test statistic

$$t = \frac{\overline{d}}{S_d/\sqrt{n}}. \tag{3.3}$$

If H_0 is true, the paired t-statistic follows a Student's t-distribution with $n - 1$ degrees of freedom. Table 3.3 summarizes the rejection regions for different alternative hypotheses. The R code is shown below.

```
t.test(log10(VCrimeRate2019), log10(VCrimeRate2010), paired = T)
```

While the paired t-procedure is the best procedure when the normality assumption is met, and the Wilcoxon signed-rank procedure is a resistant alternative to the paired t-procedure and is more powerful than the sign procedure (but requires more assumptions on the data), Fisher's sign procedure requires the least assumptions and is also least powerful. The procedures covered in this section all lead to the same conclusion. The p-values from Fisher's sign-test, Wilcoxon's signed-rank procedure, and the t-procedure are 0.2, 0.09, and 0.08, respectively, which means that at the 5% level we cannot reject the hypothesis that the median and mean of the log-transformed crime rates in 2010 and 2019 are the same.

3.4 Continuous Variable – Independent Samples

Blood lead

Consider the following example:

Example 3.4.1 (Blood Lead Data) This data set presents observations corresponding to blood lead levels for 33 children of parents who had worked in a lead related factory and 33 other children from their neighborhood (treated as control cases) Pruzek and Helmreich (2009). Although the data set is available in the R package *PairedData* presumably to serve as an example of matched paired data, we believe that a careful reading of these setup indicates that is more reasonable to treat the data as two independent samples.

Research question #2

Is the average lead level higher in exposed children than in the control group?

Based on the collected data, a question of interest is whether the *mean* (or *median*) lead level in the exposed group of children is higher than the corresponding level in the control group. As we have seen before, it is important to state our assumptions, which will determine the inferential procedure that will be used. For instance, are we willing to assume that

- the data from the two groups may be regarded as two independent samples from different populations?
- the dispersions of the lead levels are the same in both groups?
- the shapes of the data distributions are the same in both groups?
- the lead level data has a normal distribution in both groups?

The data collection mechanism offers us a guide to treating the data $Y_{1,j},\ j = 1, \ldots, 33$ and $Y_{2,j},\ j = 1, \ldots, 33$ as independent samples from two populations. Although the sample sizes are the same, in order to be paired data, the children must have been matched on several characteristics other than just being from the same region, and there is no evidence from the data description that this was the case. Given that we have data from two independent samples, our research question consists of asking whether on average, the lead levels are higher among children in the exposed group compared with the control group. Note that similar to the previous discussion, the averages may refer to means when the assumption that data come from two normal populations is valid. On the other hand, if the assumption of normal populations is questionable, but the data distributions nevertheless have the same shape, we may seek to compare the medians in the two groups. Note that it is by chance that we have the same number of children in the two groups. Often, with data from independent samples, the sample sizes may be different due to several reasons.

Our next step is to state the formal null and alternative hypotheses in terms of underlying parameters, calculate a sample statistic and its probabilistic properties and then carry out relevant inference and draw conclusions about our research question. For comparing the differences in population location parameters (means or medians, as relevant) based on two independent samples, we discuss three procedures: the two-sample pooled t-procedure,

the Welch's t-procedure and the Wilcoxon rank-sum procedure. The first two are parametric procedures, while the third is a nonparametric procedure. We can use the two-sample pooled t-procedure if the data supports our assumptions that both samples comes from normal populations with the same variance. The Welch's t-procedure is useful when both samples satisfy the normality assumption, but their variances are very different. Finally, when one or both samples may not come from normal populations, but their distributions have the same shape, the nonparametric rank-sum procedure (which is equivalent to the Mann-Whitney procedure that is available in some software) may be appropriate. When the most stringent assumptions of normality of both populations and equal variances are satisfied by the data, then the two-sample pooled t-procedure will be most powerful, followed by Welch's t-procedure, and finally the Wilcoxon rank-sum procedure.

The following table shows details for the three methods including the sample statistics along with the sampling distributions. In the table, n_1 and n_2 denote the two sample sizes (here, $n_1 = n_2 = 33$ children), and for $i = 1, 2$, μ_i and η_i are the means and medians respectively of the two populations, $\overline{Y}_i$ and S_i^2 are the means and variances of the two samples, S_P^2 is the pooled estimate of the common variance, while R_j, $j = 1, \ldots, n_1$ denote the ranks computed in the combined sample of the observations corresponding to the first sample:

$$\begin{aligned}
\overline{Y}_i &= \frac{\sum_{j=1}^{n_i} Y_{i,j}}{n_i},\ i = 1, 2 \\
S_i^2 &= \frac{(Y_{i,j} - \overline{Y}_i)^2}{n_i - 1},\ i = 1, 2 \\
S_P^2 &= \{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2\}/(n_1 + n_2 - 2) \\
W &= \sum_{j=1}^{n_1} R_j
\end{aligned}$$

Method	Rank-sum test	Welch's t test	Pooled t test
Assumptions	Same shape	Normality	Normality and equal variances
Inference about	$\eta_1 - \eta_2$	$\mu_1 - \mu_2$	$\mu_1 - \mu_2$
Statistic	$W = \sum_{j=1}^{n_1} R_j$	$T_W = \frac{\overline{Y}_1 - \overline{Y}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$	$t_P = \frac{(\overline{Y}_1 - \overline{Y}_2)}{S_P\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$
Sampling distribution	$\frac{W - E_0(W)}{\sqrt{Var_0(W)}} \approx N(0, 1)$	Student's $t_{\widetilde{\nu}}$	Student's $t_{n_1+n_2-2}$
Null hypothesis	$H_0 : \eta_1 = \eta_2$	$H_0 : \mu_1 = \mu_2$	$H_0 : \mu_1 = \mu_2$
R function	wilcox.test	t.test	t.test

Key formulas

The alternative hypothesis is $H_1 : \eta_1 > \eta_2$ for the nonparametric rank-sum procedure, and $H_1 : \mu_1 > \mu_2$ for the two parametric procedures. Although we show upper-tailed alternatives that are relevant to the examples in this chapter, H_1 can be a lower-tailed or a two-tailed alternative, as appropriate.

The Blood Lead data from Example 3.4.1 is available in the *PairedData* package. We have downloaded it as an rda file, and load it using the command load(). The command str() shows us a description of the variables in the data frame BloodLead, which is in the 'wide format'.

```
load("Data/BloodLead.rda")
str(BloodLead)
head(BloodLead)
```

```
>str(BloodLead)
'data.frame':        33 obs. of  3 variables:
 $ Pair   : Factor w/ 33 levels "P01","P02","P03",..: 1 2 3 4 5 6 7 8 9 10  ...
 $ Exposed: int  38 23 41 18 37 36 23 62 31 34 ...
 $ Control: int  16 18 18 24 19 11 10 15 16 18 ...
>head(BloodLead)
  Pair Exposed Control
1  P01      38      16
2  P02      23      18
3  P03      41      18
```

For some of the following steps, it will be more convenient to work with a 'long format'. We convert from wide to long format using the command stack() and save the data into lead. This can also be done by using the melt() function from the *reshape2* package. The data lead has two columns called `values` and `ind`, which we rename to `Lead` and `Group`, respectively.

```
lead <- stack(BloodLead[, 2:3])
colnames(lead) <- c("Lead", "Group")
head(lead, 3)
```

```
> head(lead, 3)
  Lead   Group
1   38 Exposed
2   23 Exposed
3   41 Exposed
```

We can visually verify whether the spreads in the two populations ("Exposed" and "Control") and are equal by checking whether the side-by-side boxplots of the two samples of the variable `Lead`, grouped by `Group`, overlap. We can also construct normal Q-Q plots and carry out the Shapiro-Wilk test to check the normality assumption in both groups. Notice that unlike the paired procedure, we must keep the two samples separate when we make these plots. The following code uses the *ggplot2* package for creating the boxplots and the Q-Q plots. Note that *ggplot2* allows us to create the Q-Q plots by group within the same plot rather easily, by using the 'aes' option. The package *gridExtra* allows us to place multiple plots in the same display, via the grid.arrange() function.

```
library(gridExtra)
library(ggplot2)
pb <- ggplot(lead, aes(x = Group, y = Lead, fill = Group)) +
    geom_boxplot(show.legend = F)
pq <- ggplot(lead, aes(sample = Lead, shape = Group, color = Group)) +
    stat_qq() + stat_qq_line()
grid.arrange(pb, pq, ncol = 2)
#Test for normality
```

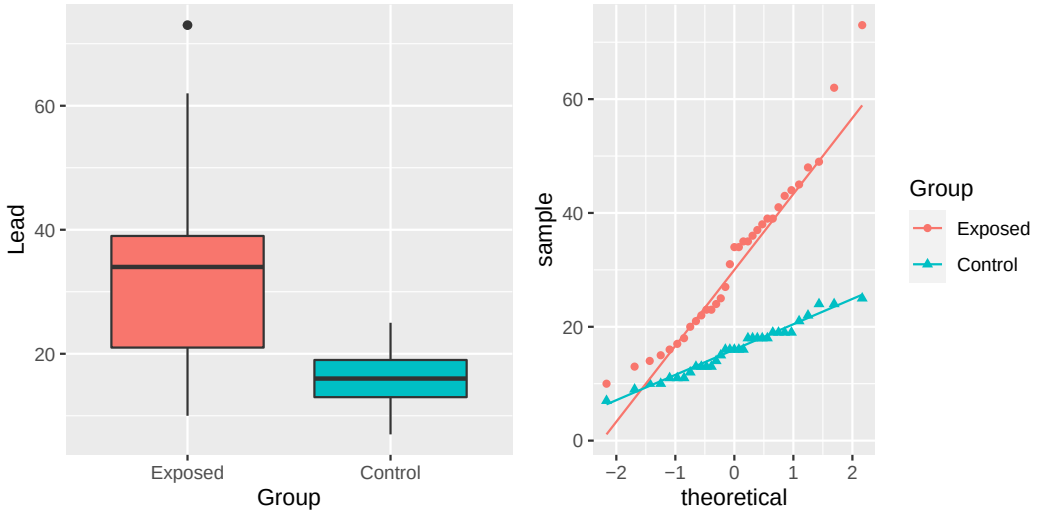

FIGURE 3.3: Left: Side-by-side boxplots of lead level by group (exposed or control). Right: Q-Q plot for the two groups.

```
shapiro.test(BloodLead[, 2])$p.value   # p-value=0.110
shapiro.test(BloodLead[, 3])$p.value   # p-value=0.577
```

The boxplots show that the spreads in the two groups are quite different. The normal Q-Q plot for each group shows that the points fall approximately on a straight line, while the Shapiro-Wilk significance test gives p-values of 0.110 and 0.577. These results support the assumption of normality in both populations.

If the normality assumption holds, we may use the F-test to verify whether the variances of the two populations are the same, thus using a significance test to corroborate the information given by the side-by-side boxplots. The F-statistic for testing $H_0 : \sigma_1^2 = \sigma_2^2$ versus $H_1 : \sigma_1^2 \neq \sigma_2^2$ is the ratio of the sample variances,

$$F = S_1^2/S_2^2 \sim F_{n_1-1,n_2-1}. \tag{3.4}$$

We use the R function var.test() to obtain the F-statistic. Not rejecting H_0 indicates that the equal variance assumption is satisfied.

```
var.test(BloodLead[, 2], BloodLead[, 3]) # p-value = 3.121e-09
```

Since the F-statistic, which is the ratio of sample variances, is 10.071 with a very small p-value of 3.12e-09, we can conclude that the data provides sufficient evidence to reject H_0, and the assumption of equal variances may be unreasonable. Therefore, to test whether the means are different between the two groups, Welch's t-test is appropriate, and not the pooled t-test (which relies on the equal variance assumption).

The Welch's t-statistic T_W and the pooled t-statistic t_P are used for testing $H_0 : \mu_1 = \mu_2$

versus $H_1 : \mu_1 > \mu_2$, and have the following forms:

$$T_W = \frac{\overline{Y}_1 - \overline{Y}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}, \text{ and} \tag{3.5}$$

$$t_P = \frac{(\overline{Y}_1 - \overline{Y}_2)}{S_P\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}. \tag{3.6}$$

T_W in (3.5) has an approximate Student t-distribution with d.f. given by the value of $\widetilde{\nu}$ rounded up to the nearest integer, where

$$\widetilde{\nu} = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{S_1^4/[n_1^2(n_1-1)] + S_2^4/[n_2^2(n_2-1)]}; \tag{3.7}$$

In the formula for t_P in (3.6), S_P denotes the pooled standard deviation defined as the square root of

$$S_P^2 = \{(n_1-1)S_1^2 + (n_2-1)S_2^2\}/(n_1+n_2-2). \tag{3.8}$$

The pooled t-statistic t_P has a Student-t distribution with n_1+n_2-2 d.f. when H_0 is true. To use this test, the data must satisfy two assumptions: normality of both samples, and equality of population variances.

The following code shows how to obtain both the Welch's and pooled t-tests, using the t.test function. Although the conclusion is the same from both, i.e., we reject the null hypothesis that the two means are equal, the correct method to use here is the one which does not rely on the equal variance assumption. Note that with the pooled t-test, the degrees of freedom parameter is 64, while with the Welch's method it is 38.29 (rounded to 38). It is always true that $df_W \leq df_P$, and thus, the p-value from the pooled t-test is always less than or equal to the one from Welch's t-test. Therefore, when the variances are not equal, the two tests can lead to different conclusions.

The samples means in the "Exposed" and "Control" groups are 31.848 and 15.879, as shown below and the difference between the sample means is 15.970.

```
# Welch t-test
tw <- t.test(Lead ~ Group, data = lead, alternative = c("greater"))
# pooled t-test
tp <- ttest <- t.test(Lead ~ Group, data = lead, alternative = c("greater"),
    var.equal = TRUE)
# p-values
tw$p.value # 2.2e-07
tp$p.value # 3.8e-08
# d.f. for the t-statistics
tw$parameter # Welch: 38.29
tp$parameter # Pooled: 64
```

The test statistic can be extracted directly from the `tw` variable, as follows, or it can be computed via the estimates for the two groups and the standard error:

```
tw$statistic  # 6.0732
# which is the same as
(tw$estimate[1] - tw$estimate[2]) / tw$stderr
```

In conclusion, there is strong evidence for rejecting the null hypothesis of equal means in the two groups and concluding that the mean lead level is significantly higher in the "Exposed" group.

Cohen's d is useful for estimating *effect size* when we compare two means, and is defined as the difference between the two sample means divided by an estimate of the standard deviation, i.e.,

$$\text{Cohen's d} = \frac{\overline{Y}_1 - \overline{Y}_2}{S}. \tag{3.9}$$

S is calculated using the formula for S_P in (3.8), except for using a divisor of n_1+n_2 instead of n_1+n_2-2. An effect size of $d=1.5$ means that the value of the average observation in the first group is 1.5 standard deviations above the average observation in the second group. We can use the R code cohen.d() in the R package *effsize*.

```
>cohen.d(BloodLead[, 2],BloodLead[, 3])

Cohen's d
d estimate: 1.495109 (large)
95 percent confidence interval:
    lower     upper
0.9388195 2.0513992
```

To illustrate the Wilcoxon rank-sum procedure (equivalently, the Mann-Whitney procedure), we consider a different example. As we have seen earlier, this procedure is useful when the normality assumption is not satisfied, perhaps not even after a suitable data transformation (such as a log transformation).

Cars

Example 3.4.2 (Cars Data) The mtcars dataset contains information about 32 cars, including their gasoline consumption (mpg), transmission type (am=automatic/manual), number of cylinders (cyl), etc. The information was collected from the 1974 Motor Trend US magazine.

Research question #3

Are cars with manual transmissions more economical (giving higher median mpg) than cars with automatic transmissions?

First, let us read and set up the data. In the following code we create a categorical variable called `AM` corresponding to the transmission type. We also create two numeric variables for the mpg measurement by transmission type (called `a.mpg` and `m.mpg`).

```
data(mtcars)
str(mtcars)
AM <- factor(mtcars$am, labels = c("Automatic", "Manual"))
table(AM) # 19 automatic, 13 manual
a.mpg <- mtcars$mpg[which(AM == "Automatic")]
m.mpg <- mtcars$mpg[which(AM == "Manual")]
```

Before using the rank-sum procedure to test whether the median mileages in the two groups are different, we look at side-by-side boxplots to visually compare medians and

spreads in the two groups, and an empirical Q-Q plot of mileage per gallon for cars with manual transmission versus mileages for cars with automatic transmission. Recall from Chapter 2 that (a) if the data distributions are exactly the same, the points in an empirical Q-Q plot would lie along a 45° line, implying that the two populations have the same shape and the same medians, so that we expect $\Delta = 0$; (b) If the points lie along a line that is parallel to the 45° line, then the two groups have the same shape, but $\Delta \neq 0$. If the points fall above the 45° line, then we expect $\Delta > 0$, while the points falling below the 45° line implies $\Delta < 0$; (c) however, if the points are not on the 45° line or parallel to it, then the two populations do not have the same shape.

The following code is used to create the side-by-side boxplots by transmission type (Figure 3.4, left), and the empirical Q-Q plot (Figure 3.4, right) which is constructed using the command qqplot(). Note that we save the result of qqplot() in a variable called `qqp` so that we can fit the mpg values of the two transmission types (the red dashed line in the plot). We also use the abline(0,1) line to draw a 45° line. Note that we use the xlim and ylim options to make sure that the scale for the mpg of the two transmission types is the same.

```
cols <- c("green", "lightblue")
boxplot(mpg ~ AM, ylab = "mpg", xlab = "Transmission", main = "A", data =
    mtcars, col = cols, border = "gray33")
qqp <- qqplot(a.mpg, m.mpg,  xlab = "mpg - automatic", ylab = "mpg -
    manual", main = "B", pch = 19, col = 4, xlim=c(min(mtcars$mpg),
    max(mtcars$mpg)), ylim=c(min(mtcars$mpg), max(mtcars$mpg)))
abline(0, 1, col = "gray", lwd = 2) # x=y line
abline(lm(qqp$y ~ qqp$x), lty=2, col=2)
```

In Figure 3.4 (left), we see that both the median efficiency (mpg) and the spread for cars with manual transmission are higher than the corresponding summaries for cars with automatic transmission, although the boxes overlap, so that the disparity in spreads is not too much. The empirical Q-Q plot shows that the points lie on a straight line which is approximately parallel to the 45° line.

If we let η_1 and η_2 be the two population medians, let $\Delta = \eta_1 - \eta_2$ be their difference. We assume that the populations have the same shape, so they may be considered to be similar except for the difference caused by a non-zero Δ value. For testing $H_0 : \Delta = 0$ versus the alternative $H_1 : \Delta > 0$, we can use the Wilcoxon rank-sum statistic constructed as follows. We order the $N = n_1 + n_2$ observations of the combined sample in ascending order; if two or more observations are tied, we assign to each the midrank (i.e., the average of the ranks). We denote by R_j the rank of the observations in the first sample. The rank-sum statistics is

$$W = \sum_{j=1}^{n_1} R_j, \tag{3.10}$$

which is the sum of the ranks assigned to the observations in the first sample. We use the rank-sum procedure on the data using the wilcox.test command with the `paired=F` and `alternative="greater"` options.

```
wilcox.test(m.mpg, a.mpg, paired = F, alternative = "greater")
```

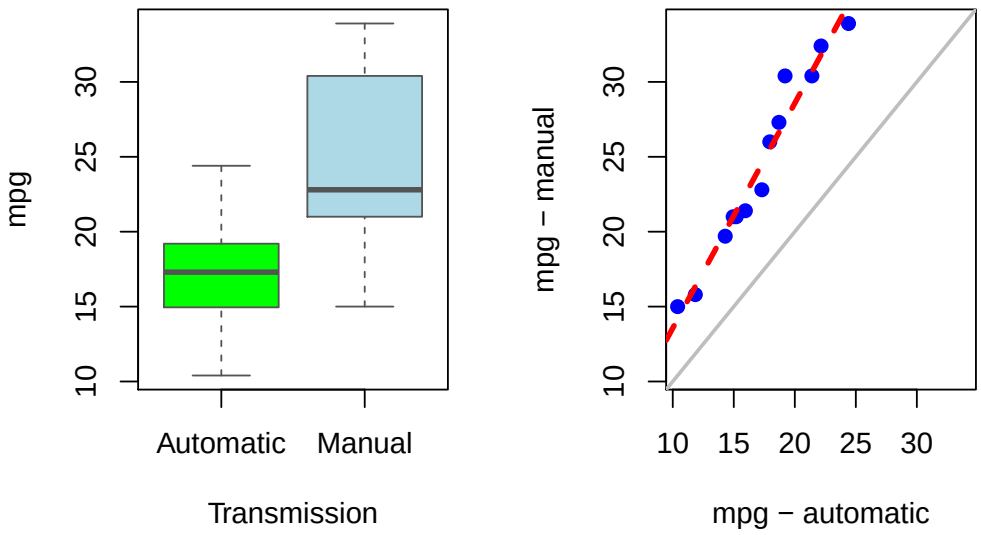

FIGURE 3.4: Side-by-side boxplots of mileage per gallon by transmission mode (left plot) and empirical Q-Q plot of manual versus automatic transmission with the x=y line (right plot).

The Wilcoxon rank-sum statistic with continuity correction is $W = 205$, with a p-value of 0.0009, so the data overwhelmingly supports the hypothesis that the fuel efficiency (mpg) of manual cars is significantly higher than for the automatic transmission cars.

We end this section by illustrating how we can use the nonparametric two-sample Kolmogorov-Smirnov procedure to compare the miles per gallon data *distributions* between cars using the two types of transmission. The null hypothesis is H_0: the c.d.f.'s of the two populations are the same, versus the alternative H_1: the two c.d.f.'s are not the same, i.e.,

$$\begin{aligned} H_0 &: F(u) = G(u) \text{ for all } u \text{ versus} \\ H_1 &: F(u) \neq G(u) \text{ for at least one } u. \end{aligned}$$

Let $\widehat{F}_{n_1}(u)$ and $\widehat{G}_{n_2}(u)$ denote the respective empirical c.d.f.'s of the two samples. Let d denote the greatest common divisor (gcd) of n_1 and n_2. The Kolmogorov-Smirnov statistic (see (3.15)) represents the value of the *largest* vertical distance between $\widehat{F}_{n_1}(x)$ and $\widehat{G}_{n_2}(x)$, which must be small under H_0. We use the ks.test() function, as shown below. The output shows that the observed test statistic is $D = 0.636$ and the p-value is 0.004, indicating that the data supports the hypothesis of different distributions.

```
ks.test(m.mpg, a.mpg)
```

In practice, it is possible that we first run the Kolmogorov-Smirnov procedure to see whether there is a difference in the distributions, followed by the rank-sum procedure to see if the difference is caused by a location shift, and the Ansari-Bradley procedure to check equality of spreads *if the medians are the same*. Here, since the rank-sum procedure showed that the medians are different, so we do not run the Ansari-Bradley procedure (see the appendix for details on this procedure).

3.5 Table Analysis of Two Categorical Variables

Arthritis

Consider the following example:

Example 3.5.1 (Arthritis Data) The Arthritis dataset contains observations from a double-blind clinical trial of treatments for rheumatoid arthritis Koch and Edwards (1988). The data is available in the package *vcd* in R. It consists of 84 observations with five variables. The effectiveness of the treatment is recorded under the ordinal factor variable `improved`. This variable has three levels – None, Some, and Marked improvement. Other predictors of interest are the age and sex of the patient, but these will be considered in later chapters.

Research question #4

Is the treatment effective for rheumatoid arthritis?

If the treatment is effective, we would expect the proportion of markedly improved patients to be significantly greater among treated individuals than among placebo recepients. We may also interpret the question by considering the proportions of patients who had no

improvement; again, we would expect that if a treatment is effective, there would be a smaller proportion of patients showing no improvement in the treatment group than in the placebo cohort. If the treatment is not effective, we would expect to see no difference between the two groups in the proportions of any of the three improvement categories.

Similar to the previous subsections, we are interested in a comparison between two groups (treatment versus placebo). However, in this case the outcome is categorical, rather than numeric. To analyze such data, we first arrange it as a frequency table, so that the rows correspond to the treatment categories, and the columns correspond to the outcome. More generally, there can be r levels for the treatment, and c different outcome categories, and we arrange the data in an $r \times c$ matrix of counts. Such matrices are called two-way tables. The null hypothesis is that the r groups are not different from each other, and if that is true, then in each row of the two-way table, we expect to have the same proportions of outcome levels. That is, the probability of the outcome being $j \in \{1, \ldots, c\}$ is the same in each row ($i \in \{1, \ldots, r\}$).

We illustrate a typical analysis using Example 3.5.1. We use the following code to load the dataset, and the command str to show the structure of the dataset, along with the first ten observations of each variable.

```
library(vcd)
str(Arthritis)
```

```
'data.frame':        84 obs. of  5 variables:
$ ID       : int  57 46 77 17 36 23 75 39 33 55 ...
$ Treatment: Factor w/ 2 levels "Placebo","Treated": 2 2 2 2 2 2 2 2 2 2 ...
$ Sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
$ Age      : int  27 29 30 32 46 58 59 59 63 63 ...
$ Improved : Ord.factor w/ 3 levels "None"<"Some"<..: 2 1 1 3 3 3 1 3 1 1 ...
```

We use the xtabs() function to cross-tabulate the frequencies of the two variables, `Treatment` and `Improved`.

```
(A.table <- xtabs( ~ Treatment + Improved, data = Arthritis))
```

```
          Improved
Treatment None Some Marked
  Placebo   29    7      7
  Treated   13    7     21
```

The code and putput below show the values in the 2×3 table as proportions and percentages.

```
> prop.table(A.table) # proportions
          Improved
Treatment       None       Some     Marked
  Placebo 0.34523810 0.08333333 0.08333333
  Treated 0.15476190 0.08333333 0.25000000
> prop.table(A.table)*100 # percentages
          Improved
Treatment      None      Some    Marked
  Placebo 34.523810  8.333333  8.333333
  Treated 15.476190  8.333333 25.000000
```

The categorical variable `Improved` has three levels: `None`, `Some` and `Marked`. We see the frequency, proportion and percentage of patients corresponding to each level of the variable `Improved` as follows:

```
(I.table <- table(Arthritis$Improved))

  None   Some Marked
    42     14     28
> prop.table(I.table) # proportions

     None      Some    Marked
0.5000000 0.1666667 0.3333333
> prop.table(I.table)*100 # percentages

    None     Some   Marked
50.00000 16.66667 33.33333
```

We wish to see whether the distribution of the variable `Improved` (with three levels) varies by `Treatment` (with two levels), the data is shown again below.

```
Improved
Treatment None Some Marked
Placebo   29    7      7
Treated   13    7     21
```

Clearly, the distribution of improvement status is quite different in the two treatment groups. Figure 3.5 shows this data graphically, obtained using the following code:

```
mosaicplot(A.table, color = c("gray", "lightgreen", "slateblue"), main =
    "Arthritis data")
```

The function margin.table() is used to obtain row or column totals. To show the frequencies of the improvement status within each treatment, use the following (the option index =1 refers to the first variable in the xtabs() statement):

```
margin.table(A.table, 1) # row sums
```

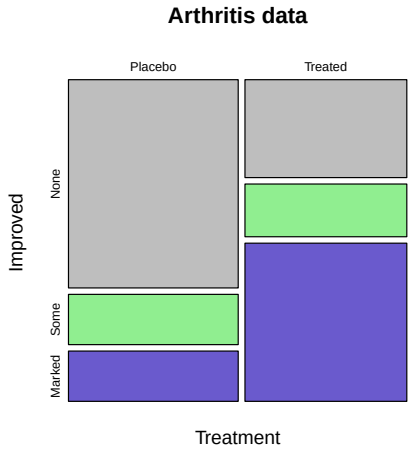

FIGURE 3.5: A mosaic plot showing the distribution of improvement status by treatment.

```
Treatment
Placebo Treated
43      41
```

Note 1 The table() function ignores missing values (NA) by default. To include NA as a valid category in the frequency counts, use NA="ifany".

Note 2 In addition to table() and xtabs(), it is possible to create two-way tables using the CrossTable() function in the *gmodels* package.

We wish to test whether the improvement status is significantly associated with treatment status (control, or treatment), i.e., test H_0: Treatment status and Improvement status are independent versus H_1: Treatment status and Improvement status are dependent.

Let $C(i, j)$ denote the *cell count* in the (i, j)th cell of the two-way table with r rows and c columns. We compute the expected count in the (i, j)th cell under H_0 as

$$E(i,j) = \frac{(\text{sum of row } i) \times (\text{sum of column } j)}{\text{total sample size}}.$$

If the observed cell counts are significantly different from the expected counts, we would conclude that there is a relationship between the two variables, i.e., they are not independent. The chi-squared test statistic is defined by

$$\chi^2 = \sum_{i,j} \frac{[C(i,j) - E(i,j)]^2}{E(i,j)}. \tag{3.11}$$

If all the observed counts are close to the expected counts, the chi-squared statistic will be small, suggesting weak evidence against the null hypothesis. where r and c are the number

of rows and columns in the two-way table. Provided $E(i,j) > 5$ for all i, j, we can reject H_0 in favor of H_1 at level of significance α if $\chi^2 > \chi^2_{(r-1)(c-1),\alpha}$, a critical value from the $\chi^2_{(r-1)(c-1)}$-distribution. We use the command chisq.test():

```
(cstest <- chisq.test(A.table))
```

```
  Pearson's Chi-squared test

  data:  A.table
  X-squared = 13.055, df = 2, p-value = 0.001463
```

The small p-value (0.0015) suggests that we can reject the null hypothesis at the 5% level and conclude that the variables `Treatment` and `Improved` are not independent. Note that the `cstest` variable contains also the observed frequencies table, the expected counts, and the residuals. Try using names(cstest) to see all the possibilities.

In this example, the expected counts in each cell are all greater than 5. (Try cstest*cstest*expected to see that.)

```
  > cstest$expected
           Improved
  Treatment None     Some   Marked
    Placebo 21.5 7.166667 14.33333
    Treated 20.5 6.833333 13.66667
```

Nevertheless, we can also try the non-parametric approach for illustration, which is valid when the counts in some cells are not sufficiently large. For Fisher's exact test, use:

```
(fishertest <- fisher.test(A.table))
```

```
  Fisher's Exact Test for Count Data

  data:  A.table
  p-value = 0.001393
  alternative hypothesis: two.sided
```

Since the p-value is 0.001393, we reject the null hypothesis at the 5% level and conclude that `Treatment` and `Improvement` are dependent.

We can also compare proportions of interest. Consider the difference between two proportions amongst patients who underwent treatment: proportion of patients who showed some or marked improvement and proportion of patients who showed no improvement. We can construct the 95% confidence interval estimate for the difference in true proportions (see (3.18) in the appendix) using the prop.test function as shown below. The interval $(0.1336521, 0.5806336)$ does not include zero, showing that the two proportions are significantly different.

```
A1.table <- data.frame(A.table)
T.freq <- A1.table[A1.table$Treatment=="Treated",]$Freq
grpcounts <- c(T.freq[2]+T.freq[3],T.freq[1])  # c(28,13)
totals <- c(csums[2]+csums[3],csums[1])   # c(42,42)
# Chi-square test for comparing proportions and C.I. for difference in
    proportions
prop.test(grpcounts, totals)
```

```
  2-sample test for equality of proportions with continuity correction

  data:  grpcounts out of totals
  X-squared = 9.3386, df = 1, p-value = 0.002244
  alternative hypothesis: two.sided
  95 percent confidence interval:
   0.1336521 0.5806336
  sample estimates:
     prop 1    prop 2
  0.6666667 0.3095238
```

Appendix Chapter 3 -Statistical Details

Procedures for Paired Samples: Statistical Details

Paired Sign Procedure

Fisher's sign procedure is a useful nonparametric procedure under very minimal assumptions in the paired two sample situation. Let Y_1 and Y_2 denote the variables of interest in the two populations and let their difference $D = Y_1 - Y_2$ follow an *arbitrary continuous* distribution with median θ_d. We wish to test $H_0 : \theta_d = 0$ using a random sample of size n from both populations. Let d_i be the ith difference. We define indicator variables $\psi_i = 1$ if $d_i > 0$ and $\psi_i = 0$ if $d_i < 0$ and obtain the sum of positive signs (see (3.1)):

$$B = \sum_{i=1}^{n} \psi_i.$$

Under H_0, ψ_i are independent Bernoulli random variables with $p = 1/2$. Hence, $P(\psi_i = 1) = P(\psi_i = 0) = 1/2$, so that its expectation and variance under H_0 are $E_0(\psi_i) = 1/2$ and $\text{Var}_0(\psi_i) = 1/4$. Then,

$$\begin{aligned} E_0(B) &= E_0\left(\sum_{i=1}^{n} \psi_i\right) = n/2 \text{ and} \\ \text{Var}_0(B) &= \text{Var}_0\left(\sum_{i=1}^{n} \psi_i\right) = n/4. \end{aligned}$$

TABLE 3.1: Criteria and Decision Rules for the Sign Procedure

Alternative Hypothesis	Criterion
$H_1 : \theta_d > 0$	$B_{\text{obs}} \geq b(\alpha, n, 1/2)$
$H_1 : \theta_d < 0$	$B_{\text{obs}} \leq n - b(\alpha, n, 1/2)$
$H_1 : \theta_d \neq 0$	$B_{\text{obs}} \geq b(\alpha_2, n, 1/2)$ or $B \leq n - b(\alpha_1, n, 1/2)$, where $\alpha = \alpha_1 + \alpha_2$.

Under H_0, the distribution of B is symmetric about $n/2$. Hence, for $u = 0, 1, \ldots, n/2$,

$$P_0(B \geq u) = P_0(B \leq n - u).$$

For a given level of significance α, we choose $b(\alpha, n, 1/2)$ as the smallest integer such that

$$\sum_{b=b(\alpha,n,1/2)}^{n} \binom{n}{b} (1/2)^n \leq \alpha.$$

The criteria to test H_0 are shown in Table 3.1.

For large n, the statistic

$$B^* = \frac{B - E_0(B)}{[\text{Var}_0(B)]^{1/2}} = \frac{B - n/2}{\sqrt{n/4}} \tag{3.12}$$

has an asymptotic $N(0, 1)$ distribution, which can be used to construct an approximate z-procedure for testing H_0. Thus, for instance, for a two-sided test, we reject H_0 at level α if $|B^*| > z_{\alpha/2}$.

Paired Wilcoxon Signed-Rank Procedure

The Wilcoxon Signed-Rank test uses both (a) the signs of the differences and (b) the ranks of the magnitudes of the differences. It is a resistant alternative to the paired t-procedure, and is more powerful than the sign procedure (but requires more assumptions on the data).

Again, let Y_1 and Y_2 denote the variables of interest in the two populations and let their difference $D = Y_1 - Y_2$ follow a *symmetric, continuous* distribution with median θ_d. We wish to test $H_0 : \theta_d = 0$ using a random sample of size n from both populations. Let d_i be the ith difference.

After verifying that the differences come from a symmetric population (using a histogram or boxplot, say), we compute the absolute differences, $|d_i|$, $i = 1, \ldots, n$. We order the absolute differences in ascending order and assign ranks from 1 to n, or assign average rank in case of ties. Let R_i denote the ranks. Define indicator variables $\psi_i = 1$ if $d_i > 0$ and $\psi_i = 0$ if $d_i < 0$. Let

$$T^+ = \sum_{i=1}^{n} \psi_i R_i$$

be the sum of the positive signed ranks. Note that $\psi_i R_i$ is equal to 0 if $d_i < 0$, and is equal to the rank of $|d_i|$ if $d_i > 0$. In this procedure, we must know the signs of d_i's and their magnitudes.

Under H_0, ψ_i are independent Bernoulli random variables with $p = 1/2$, so that

$$\begin{aligned} E_0(T^+) &= n(n+1)/4 \text{ and} \\ \text{Var}_0(T^+) &= n(n+1)(2n+1)/24. \end{aligned}$$

TABLE 3.2: Criteria and Decision Rules for the Wilcoxon Signed-Rank Procedure

Alternative Hypothesis	Criterion
$H_1 : \theta_d > 0$	$T^+_{\text{obs}} \geq t^+(\alpha, n)$
$H_1 : \theta_d < 0$	$T^+_{\text{obs}} \leq n(n+1)/2 - t^+(\alpha, n)$
$H_1 : \theta_d \neq 0$	$T^+\text{obs} \geq t^+(\alpha_2, n)$ or $T^+ \leq n(n+1)/2 - t^+(\alpha_1, n)$, where $\alpha = \alpha_1 + \alpha_2$.

The distribution of T^+ is symmetric about $n(n+1)/4$, and we reject H_0 for large values of T^+. At a level of significance α, let $t^+(\alpha, n)$ be the critical value which satisfies $P_0\{T^+ \geq t^+(\alpha, n)\} = \alpha$, obtained based on the distribution of T^+ under H_0. The test criteria and decision rules are given in Table 3.2.

Under H_0, the statistic

$$T^* = \frac{T^+ - E_0(T^+)}{[\text{Var}_0(T^+)]^{1/2}} = T^+ - \{n(n+1)/4\}\{n(n+1)(2n+1)/24\}^{1/2} \tag{3.13}$$

has an asymptotic $N(0,1)$ distribution, which can be used to construct an approximate z-test when n is large. The R function for the Wilcoxon signed-rank procedure is wilcox.test(x, y, paired = TRUE).

Paired t-Procedure

Let y_{i1} and y_{i2} be observations on the same unit (subject) i, and let $d_i = y_{i1} - y_{i2}$, $i = 1, \ldots, n$ denote the n differences. We assume (and verify using the normal Q-Q plot) that the differences $d_1, \ldots, d_n$ come from a normal population. Suppose the normal population has mean μ_d and variance σ_d^2. Using the paired samples to test the null hypothesis $H_0 : \mu_d = 0$ enables us to compare the true means of the paired samples. The alternative hypothesis is determined by the question of interest and can be

(i) $H_1 : \mu_d \neq 0$, a two-tailed alternative; or

(ii) $H_1 : \mu_d > 0$, an upper-tailed alternative; or

(iii) $H_1 : \mu_d < 0$, a lower-tailed alternative.

Before doing the test, we choose a level of significance of the test as $\alpha = 0.05$, say. We compute the sample mean and standard deviation of the d_i's as $\overline{d}$ and S_d respectively, and construct the paired t-statistic

$$t = \frac{\overline{d}}{S_d/\sqrt{n}}.$$

If H_0 is true, the paired t-statistic follows a Student's t-distribution with $n-1$ degrees of freedom. If the test statistic (3.3) falls in the rejection region, we say the data provides evidence to reject H_0 at level α. Equivalently, we reject H_0 if the observed p-value is less than α. Table 3.3 summarizes the rejection region and the observed p-value corresponding to each alternative hypothesis H_1.

Let Y_1 and Y_2 be two dependent random variables with respective means μ_1 and μ_2 and respective variances σ_1^2 and σ_2^2. Since they are dependent, we assume they are correlated and $\text{Corr}(Y_1, Y_2) = \rho_{12}$. Let $D = Y_1 - Y_2$ denote their difference. We assume that the random

variable D follows a normal distribution with mean μ_d and standard deviation σ_d, where $\mu_d = \mu_1 - \mu_2$ and $\sigma_d^2 = \sigma_1^2 + \sigma_2^2 + 2\rho_{12}\sigma_1\sigma_2$.

Under the assumption that the sampled differences $d_1, \ldots, d_n$ come from a normal population with mean μ_d and variance σ_d^2, the paired t-statistic in (3.3) is the ratio of a standard normal random variable divided by the square root of an independent chi-squared variable divided by its degrees of freedom, which has a Student's t-distribution with $n-1$ degrees of freedom. We choose a level of significance of the test $\alpha = P[t \geq t(\alpha, n-1)] = 0.05$, say. We can then obtain the rejection regions shown in Table 3.3.

TABLE 3.3: Criteria and Decisions for the Paired t-test

Alternative Hypothesis	Criterion	p-value
$H_1 : \mu_d > 0$	$t_{\text{obs}} > t(\alpha, n-1)$	$P(t > t_{\text{obs}})$
$H_1 : \mu_d < 0$	$t_{\text{obs}} < -t(\alpha, n-1)$	$P(t < t_{\text{obs}})$
$H_1 : \mu_d \neq 0$	$\lvert t_{\text{obs}} \rvert > t(\alpha/2, n-1)$,	$2P(t > \lvert t_{\text{obs}} \rvert)$

Procedures for Independent Samples: Statistical Details

Wilcoxon Rank-Sum Procedure

The Wilcoxon rank-sum procedure Wilcoxon (1949) is a nonparametric procedure for comparing the location parameters of two populations with only minimal assumptions on the data. Let Y_1 and Y_2 be variables from two populations, let η_1 and η_2 be the population medians and let $\Delta = \eta_1 - \eta_2$. We assume that the populations have the same shape, so they may be considered to be similar except for the difference caused by a non-zero Δ value. Suppose that we wish to test the null hypothesis $H_0 : \Delta = 0$ versus the alternative $H_1 : \Delta > 0$. In other situations, we may wish to test the null against different alternatives, say $H_1 : \Delta < 0$, or $H_1 : \Delta \neq 0$. In most cases, one population is a *treatment* population, while the second is a *control* population, and we seek to test the effect of the treatment.

Given data from two independent random samples $Y_{1,1}, \ldots, Y_{1,n_1}$ from the first population, and $Y_{2,1}, \ldots, Y_{2,n_2}$ from the second population, we can also understand the procedure by formulating the following model:

$$\begin{aligned} Y_{1,j} &= \varepsilon_j + \Delta,\ j = 1, \ldots, n_1 \text{ and} \\ Y_{2,\ell} &= \varepsilon_{n_1+\ell},\ \ell = 1, \ldots, n_2, \end{aligned}$$

where the $N(= n_1 + n_2)$ ε_j's are mutually independent random variables from the same continuous population (same shape), and Δ is an unknown parameter, which represents a location shift due to the treatment.

To compute Wilcoxon's rank-sum statistic, we order the $N = n_1 + n_2$ observations of the combined sample in ascending order. If two or more observations are tied, we assign to each the midrank (i.e., the average of the ranks). Let R_j denote the rank of the observations in the first sample, $Y_{1,j}$. The statistic

$$W = \sum_{j=1}^{n_1} R_j$$

is the sum of the ranks assigned to $Y_{1,j}$'s. Being based only on ranks in the combined sample,

TABLE 3.4: Criteria for rejection of H_0 by the Wilcoxon rank-sum test

Alternative Hypothesis	Criterion
$H_1 : \Delta > 0$	$W \geq w(\alpha, n_1, n_2)$,
$H_1 : \Delta < 0$	$W \leq \{n(n_1 + n_2 + 1) - w(\alpha, n_1, n_2)\}$,
$H_1 : \Delta \neq 0$	$W \geq w(\alpha_2, n_1, n_2)$ or $W \leq \{n_2(n_1 + n_2 + 1) - w(\alpha_1, n_1, n_2)\}$, where $\alpha = \alpha_1 + \alpha_2$.

this procedure is useful in situations where we do not know the actual magnitudes of the observations, but only know the relative ranks.

Provided there are no ties, we can show that under H_0,

$$\begin{aligned} E_0(W) &= n_1(n_1 + n_2 + 1)/2, \text{ and} \\ \text{Var}_0(W) &= n_1 n_2(n_1 + n_2 + 1)/12. \end{aligned}$$

The null distribution of W is symmetric about $E_0(W)$. Hence, for $x = n_1(n_1 + 1)/2, \ldots, n_1(2n_1 + n_2 + 1)/2$, we have

$$P_0(W \geq x) = P_0\{W \leq n_1(n_1 + n_2 + 1) - x\}.$$

If there are ties, let g denote the number of tied groups, and let t_j be the size of the jth tied group. Then,

$$\text{Var}_0(W) = \frac{n_1 n_2}{12} \left\{ (n_1 + n_2 + 1) - \left[\sum_{j=1}^{g} t_j(t_j^2 - 1)]/[(n_1 + n_2)(n_1 + n_2 - 1) \right] \right\}.$$

If there are no ties, note that $g = N$, $t_j = 1$, $j = 1, \ldots, N$. At level of significance α and for a constant $w(\alpha, n_1, n_2)$ such that $P_0\{W \geq w(\alpha, n_1, n_2)\} = \alpha$, the rejection criteria are given in Table 3.4.

A large sample approximation of the test statistic is useful in some cases. As $\min(n_1, n_2) \to \infty$,

$$W^* = \frac{W - E_0(W)}{\{Var_0(W)\}^{1/2}} = \frac{W - n_1(n_1 + n_2 + 1)/2}{\{n_1 n_2(n_1 + n_2 + 1)/12\}^{1/2}} \sim N(0, 1).$$

The R code for the rank-sum test is wilcox.test(x, y, paired = FALSE).

The rank-sum statistic is related to the Mann-Whitney test statistic proposed by Mann and Whitney (1947)

$$\begin{aligned} U &= \sum_{j=1}^{n_1} \sum_{\ell=1}^{n_2} \phi(Y_{1,j}, Y_{2,\ell}), \text{ where} \\ \phi(Y_{1,j}, Y_{2,\ell}) &= \begin{cases} 1 & \text{if } Y_{1,j} < Y_{2,\ell}, \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

For each pair $(Y_{1,j}, Y_{2,\ell})$, assign a score of 1 if $Y_{1,j} < Y_{2,\ell}$, and a score of 0 otherwise. Then U is the sum of all the $n_1 n_2$ scores. If there are no ties, the relationship between U and W is

$$W = U + n_1(n_1 + 1)/2.$$

Welch t-Procedure

Let $Y_{1,1}, \ldots, Y_{1,n_1}$ be a random sample from a $N(\mu_1, \sigma_1^2)$ population, and let $Y_{2,1}, \ldots, Y_{2,n_2}$ be an independent random sample from a $N(\mu_2, \sigma_2^2)$ population. Suppose that the population means μ_1 and μ_2 as well as the population variances σ_1^2 and σ_2^2 are unknown parameters. Suppose that we wish to test $H_0 : \mu_1 = \mu_2$ versus $H_1 : \mu_1 > \mu_2$ (upper-tailed test), when $\sigma_1^2 \neq \sigma_2^2$. In other situations, we may alternatively test $H_1 : \mu_1 < \mu_2$ (lower-tailed test) or $H_1 : \mu_1 \neq \mu_2$ (a two-tailed test).

Let $\overline{Y}_1$ and $\overline{Y}_2$ respectively denote the means of the first and second samples; these are point estimates of μ_1 and μ_2. Let S_1^2 and S_2^2 respectively denote the variances of the first and second samples. For $i = 1, 2$, $\overline{Y}_i$ have independent $N(\mu_i, \sigma_i^2)$ distributions; S_i^2 have independent $\{\sigma_i^2/(n_i - 1)\}\chi^2_{n_i-1}$ distributions, and are independent of the distributions of the sample means.

The Welch's t-statistic (Welch (1947) is

$$T_W = \frac{\overline{Y}_1 - \overline{Y}_2}{\sqrt{(S_1^2/n_1 + S_2^2/n_2)}}.$$

Let

$$\widetilde{\nu} = \frac{(S_1^2/n_1 + S_2^2/n_2)^2}{S_1^4/[n_1^2(n_1 - 1)] + S_2^4/[n_2^2(n_2 - 1)]};$$

then, T_W has an approximate Student t-distribution with d.f. given by the value of $\widetilde{\nu}$ rounded up to the nearest integer. We use the R code t.test().

Pooled t-Procedure

Suppose the same setup as we had under the Welch's t-procedure, except that we additionally assume that both populations have the same variance, i.e., $\sigma_1^2 = \sigma_2^2$. Suppose that we again wish to test $H_0 : \mu_1 = \mu_2$ versus $H_1 : \mu_1 > \mu_2$. The means and variances from both samples have the same forms and distributions defined under Welch's t-procedure.

Under the equal variance assumption, we do not estimate each population variance using data from its own sample. Instead, we pool data from both samples to construct the pooled estimate of the common variance σ^2 as

$$S_P^2 = \{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2\}/(n_1 + n_2 - 2).$$

The two-sample pooled t-statistic for testing H_0 is

$$t_P = \frac{(\overline{Y}_1 - \overline{Y}_2) - (\mu_1 - \mu_2)}{S_P\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

and has a Student-t distribution with $n_1 + n_2 - 2$ d.f. when H_0 is true. We use the R code t.test with the *var.equal = TRUE* option to implement the pooled t test, for which the data must satisfy two assumptions: normality of both samples, and equality of population variances. The latter assumption can be checked by comparing side-by-side boxplots or by using the F-test shown below.

Cohen's d is useful for estimating effect size when we compare two means, and is defined as the difference between the two sample means divided by an estimate of the standard deviation, i.e.,

$$\text{Cohen's d} = \frac{\overline{Y}_1 - \overline{Y}_2}{S_P} \tag{3.14}$$

We can use the R code cohen.d() in the R package *effsize*.

F-test for Equality of Two Variances

Assuming the same setup with two independent samples from two populations as we have seen above, we can show that

$$\frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} \sim F_{n_1-1,n_2-1}$$

under $H_0 : \sigma_1^2 = \sigma_2^2$, i.e., $\frac{\sigma_1^2}{\sigma_2^2} = 1$. The F-statistic is the ratio of the sample variances,

$$F = S_1^2/S_2^2 \sim F_{n_1-1,n_2-1}.$$

If $H_1 : \sigma_1^2 \neq \sigma_2^2$, then at level of significance α, we reject H_0 if $F > F_{n_1-1,n_2-1,\alpha}$, where $P(F > F_{n_1-1,n_2-1,\alpha}) = \alpha$. If we do not reject H_0, then the equal variance assumption is satisfied.

Ansari-Bradley Rank Test

We discuss a nonparametric procedure for comparing the spreads of two populations under the assumption that their medians η_1 and η_2 are the same, i.e., $\eta_1 = \eta_2$, and further, the populations only differ in their scales (or spreads). Let $Y_{1,1}, \ldots, Y_{1,n_1}$ and $Y_{2,1}, \ldots, Y_{2,n_2}$ denote independent samples from the two populations. Suppose

$$\begin{aligned} Y_{1,j} &= \sigma_1 \varepsilon_j + \eta, \ j = 1, \ldots, n_1 \\ Y_{2,\ell} &= \sigma_2 \varepsilon_{n_1+\ell} + \eta, \ \ell = 1, \ldots, n_2, \end{aligned}$$

where η is the common median of the two populations, and is treated as a nuisance parameter (i.e., a parameter that we are not interested in, but is unknown and must be estimated in order to carry out inference on the parameters of interest). Here, the parameter of interest is $\gamma = \sigma_1/\sigma_2$, the ratio of the scale parameters, or γ^2. Assume that ε_j, $j = 1, \ldots, n_1 + n_2$ are independent random variables from a distribution with median 0. If the variances of the two populations exist (they need not!), then γ^2 is the ratio of the variances. The assumption of equal medians is essential. If not, we at least need that the two medians are known, in which case we can obtain the data from each sample with its median subtracted out.

Suppose we wish to test the null hypothesis $H_0 : \gamma^2 = 1$ versus either $H_1 : \gamma^2 > 1$, or $H_1 : \gamma^2 < 1$, or $H_1 : \gamma^2 \neq 1$. To derive the Ansari-Bradley statistic, we order the $N = n_1 + n_2$ observations of the combined sample in ascending order. If two or more observations are tied, we assign to each the midrank (i.e., the average of the ranks). Assign rank 1 to the smallest and the largest observation. Assign rank 2 to the second smallest and second largest observation, etc. If N is even, the array of ranks will be $(1, 2, 3, \ldots, N/2, N/2, \ldots, 3, 2, 1)$. If N is odd, the array of ranks is $(1, 2, 3, \ldots, (N-1)/2, (N+1)/2, (N-1)/2, \ldots, 3, 2, 1)$. Let R_j denote the rank of $Y_{1,j}$ (i.e., rank of the observations in the first sample), $j = 1, \ldots, n_1$. The statistic is

$$AB = \sum_{j=1}^{n_1} R_j.$$

Provided there are no ties, under H_0,

(a) If N is even:

$$\begin{aligned} E_0(AB) &= n_1(n_1+n_2+2)/4, \text{ and} \\ Var_0(AB) &= n_1n_2(n_1+n_2+2)(n_1+n_2-2)/48(n_1+n_2-1). \end{aligned}$$

(b) If N is odd:

$$\begin{aligned} E_0(AB) &= n_1(n_1+n_2+1)^2/4(n_1+n_2), \text{ and} \\ Var_0(AB) &= n_1n_2(n_1+n_2+1)[3+(n_1+n_2)^2]/48(n_1+n_2)^2. \end{aligned}$$

The decision rule follows. Suppose $H_1 : \gamma^2 > 1$. At level of significance α, we reject H_0 if $AB \geq c_\alpha$, where c_α is a constant such that $P_0\{AB \geq c_\alpha\} = \alpha$. If $AB < c_\alpha$, we do not reject H_0. If $H_1 : \gamma^2 < 1$, we reject H_0 at level α if $AB \leq c_{1-\alpha} - 1$, and if $AB > c_{1-\alpha} - 1$, we do not reject H_0. If $H_1 : \gamma^2 \neq 1$, we reject H_0 at level α if $AB \geq c_{\alpha_1}$ or $AB \leq c_{\alpha_2-1}$, where $\alpha = \alpha_1 + \alpha_2$. There is a large sample approximation of the Ansari-Bradley statistic which holds when N is even or odd. As $\min(n_1, n_2) \to \infty$,

$$AB^* = \frac{AB - E_0(AB)}{\{Var_0(AB)\}^{1/2}} \sim N(0,1).$$

For a discussion of the AB statistic under ties, see Hollander et al. (2013). The test can be implemented in R using ansari.test(y1, y2), where y1 and y2 are the two samples from populations that have the same medians.

Two-sample Kolmogorov-Smirnov Procedure

We next describe a distribution-free procedure for comparison of the c.d.f.'s of two populations. Let $Y_{1,1}, \ldots, Y_{1,n_1}$ and $Y_{2,1}, \ldots, Y_{2,n_2}$ denote independent samples from the two populations with respective c.d.f.'s $F(.)$ and $G(.)$. Let $N = n_1 + n_2$. Suppose we wish to test

$$\begin{aligned} H_0 &: F(u) = G(u) \text{ for all } u \text{ versus} \\ H_1 &: F(u) \neq G(u) \text{ for at least one } u. \end{aligned}$$

Let $\widehat{F}_{n_1}(u)$ and $\widehat{G}_{n_2}(u)$ denote the respective empirical c.d.f.'s of the two samples, i.e.,

$$\begin{aligned} \widehat{F}_{n_1}(u) &= \frac{1}{n_1}\mathbf{1}(Y_{1,j} \leq u), \text{ and} \\ \widehat{G}_{n_2}(u) &= \frac{1}{n_2}\mathbf{1}(Y_{2,\ell} \leq u), \end{aligned}$$

where $\mathbf{1}(Y_{1,j} \leq u)$ denotes the number of observations in the first sample $\leq u$, and $\mathbf{1}(Y_{2,\ell} \leq u)$ is the number of observations in the first sample $\leq u$. Let d denote the greatest common divisor (gcd) of n_1 and n_2.

Consider the statistic

$$D = \frac{n_1n_2}{d} \sup_{-\infty<u<\infty} \{|\widehat{F}_{n_1}(u) - \widehat{G}_{n_2}(u)|\}, \tag{3.15}$$

where sup denotes supremum. Suppose we plot $\widehat{F}_{n_1}(x)$ versus $\widehat{G}_{n_2}(x)$; then D represents the value of the *largest* vertical distance between $\widehat{F}_{n_1}(x)$ and $\widehat{G}_{n_2}(x)$, which must be small under H_0. At level of significance α, we reject H_0 if $D \geq d_\alpha$, where $P_0(D \geq d_\alpha) = \alpha$. For details, see Hollander et al. (2013) or Conover (1999). In R, we can use the command ks.test(y1, y2).

Table Analysis: Statistical Details

Chi-Squared Test for Independence

The chi-squared test for independence can be used to determine whether or not the distribution of one variable is independent of the distribution of another variable. For two categorical variables A and B, we wish to test

$$H_0\text{: A and B are independent versus } H_1\text{: A and B are dependent.}$$

Let $C(i,j)$ denote the *cell count* in the (i,j)th cell of the two-way table for A and B. Under H_0, the expected count in the (i,j)th cell is

$$E(i,j) = \frac{(\text{sum of row } i) \times (\text{sum of column } j)}{\text{total sample size}}.$$

If the observed cell counts are significantly different from the expected counts, we would conclude that there is a relationship between the two variables, i.e., they are not independent. We compute a test statistic using the observed counts $C(i,j)$ and expected counts $E(i,j)$:

$$\chi^2 = \sum_{i,j} \frac{[C(i,j) - E(i,j)]^2}{E(i,j)}.$$

Note that if all the observed counts are close to the expected counts, the chi-squared statistic will be small, suggesting weak evidence against the null hypothesis. When H_0 is true, and the expected cell counts are sufficiently large (i.e., $E(i,j) > 5$ for all i,j), the statistic has a chi-squared distribution with $(r-1)(c-1)$ degrees of freedom, (i.e., $\chi^2_{(r-1)(c-1)}$-distribution), where r and c are the number of rows and columns in the two-way table. We reject H_0 in favor of H_1 at level of significance α if

$$\chi^2 > \chi^2_{(r-1)(c-1),\alpha},$$

where $\chi^2_{\alpha,(r-1)(c-1)}$ is such that $P(\chi^2 > \chi^2_{(r-1)(c-1),\alpha} | H_0 \text{ is true}) = \alpha$. Alternately, the decision can be made by checking whether the p-value of the test is smaller than α.

Fisher's Exact Test

The chi-squared test requires that the expected counts $E(i,j)$ are greater than 5. When that is not the case, we can use a nonparametric test, known as Fisher's exact test. In the simplest case of a 2×2 table (i.e., $r = c = 2$), Fisher's exact test is based on the hypergeometric distribution, and hence, statistical significance can be determined by a closed-form formula. In the more genral case, there is no known distribution, and statistical analysis has to rely on simulations ('Monte Carlo' methods.) In such simulations, a large number of matrices with the same row- and column-sums ('marginals') as the observed table are drawn randomly, and the likelihood of the observed matrix is determined numerically. For each simulated table, we count the number of times it was drawn, and obtain the empirical probability of each table (with the given marginals). Then, the p-value is the sum of the probabilities of the observed table and all the tables which occurred even less frequently in the simulations. When the dimensions of the table are large, this approach requires a very large number of simulations and becomes computationally challenging.

	Response		
Group	Yes	No	Total
Group 1	n_{11}	n_{12}	n_1
Group 2	n_{21}	n_{22}	n_2

TABLE 3.5: A 2×2 table.

Comparing Two Proportions

For simplicity, consider two groups, 'Group 1' and 'Group 2'. For instance, these could denote a treatment group and a control group. In each group, we observe counts of a categorical variable of interest, say Response, with two levels, 'Yes' and 'No'. This information can be represented by a 2×2 table shown in Table 3.5.

Let $\hat{\pi}_{1|1} = n_{11}/n_1$ denote the proportion of 'Yes' counts in Group 1, where 'Yes' refers to the first level of the Response. Similarly, let $\hat{\pi}_{1|2} = n_{21}/n_2$ denote the proportion of 'Yes' counts in Group 2. We assume that the samples from the two groups are independent.

The sample proportions $\hat{\pi}_{1|1}$ and $\hat{\pi}_{1|2}$ are estimates of the true proportions of 'Yes' $\pi_{1|1}$ and $\pi_{1|2}$ in the two groups. We can show that

$$E(\hat{\pi}_{1|1} - \hat{\pi}_{1|2}) = \pi_{1|1} - \pi_{1|2}, \tag{3.16}$$

$$\text{Var}(\hat{\pi}_{1|1} - \hat{\pi}_{1|2}) = \frac{\pi_{1|1}(1-\pi_{1|1})}{n_1} + \frac{\pi_{1|2}(1-\pi_{1|2})}{n_2}. \tag{3.17}$$

Provided the conditions $n_1\hat{\pi}_{1|1} > 5$, $n_1(1-\hat{\pi}_{1|1}) > 5$, $n_2\hat{\pi}_{1|2} > 5$, and $n_2(1-\hat{\pi}_{1|2}) > 5$ are satisfied, $\hat{\pi}_{1|1} - \hat{\pi}_{1|2}$ has an approximate normal distribution whose mean and variance are given in (3.17).

We can construct an approximate $100(1-\alpha)\%$ C.I. estimate for $\pi_{1|1} - \pi_{1|2}$ as the z-interval

$$\hat{\pi}_{1|1} - \hat{\pi}_{1|2} \pm z_{\alpha/2}\text{SE}(\hat{\pi}_{1|1} - \hat{\pi}_{1|2}), \tag{3.18}$$

where

$$\text{SE}(\hat{\pi}_{1|1} - \hat{\pi}_{1|2}) = \sqrt{\frac{\hat{\pi}_{1|1}(1-\hat{\pi}_{1|1})}{n_1} + \frac{\hat{\pi}_{1|2}(1-\hat{\pi}_{1|2})}{n_2}}. \tag{3.19}$$

4

Fixed Effects Analysis of Variance Models

This chapter describes the well-known Analysis of Variance (ANOVA) models which enable us to compare the mean responses at different levels of one or more categorical variables, called factors in designed experiments. In an experiment involving a single Factor A with a levels, an overall F-test enables us to compare whether all a means are equal, while multiple comparison procedures help us to further determine which of the pairwise mean differences may have led to reject the overall hypothesis of equal means. We look at nonparametric alternatives to the F-test and also extend the ideas to experiments involving two or more factors, describing models with and without interactions between the levels of the factors.

4.1 Introduction

We compared characteristics of two populations in Chapter 3. In many designed experiments and observational studies, we need to compare more than two populations. For instance, an experiment might involve a response comparison of the average between several levels of a treatment (factor), such as several doses of a drug. Another experiment might involve comparison of the distribution of the heights of seedlings as a function of several levels of a pesticide (Factor A) and several levels of fertilizer (Factor B). The factors are categorical variables with different levels. This chapter describes methods for testing hypotheses about the equality of the different distributions, or characteristics of these populations such as their means, or medians, or spreads.

In a one-factor analysis of variance (ANOVA) model, we assume that the response variable Y is normally distributed but may have a different mean for each of the a levels of a single factor. The one-factor ANOVA F-test enables us to check whether the mean responses are the same for the a populations or not, under the assumptions of normality, independence, and equal variances (or, homoscedasticity) of the response distributions, and is an extension of the two-sample pooled t-procedure to $a > 2$ groups. Similar to Chapter 3, we check for normality using normal Q-Q plots or significance tests like the Shapiro-Wilk test. The Bartlett test is used for checking whether the variances are the same in the a populations, and is an a-sample extension of the two-sample F-test for equal variances that we saw in Chapter 3. Like the two-sample F-test, Bartlett's test also requires normality and independence of the samples from the a populations. On the other hand, Levene's test for equal variances is more robust to departures from normality. Following the overall one-way ANOVA F-test, if the null hypothesis of equal means in a populations is rejected, multiple

TABLE 4.1: Data from a Single-factor Unbalanced Design

Treatment (level)	Observations				Totals	Averages
1	Y_{11}	Y_{12}	...	Y_{1n_1}	$Y_{1.}$	$\overline{Y}_{1.}$
2	Y_{21}	Y_{22}	...	Y_{2n_2}	$Y_{2.}$	$\overline{Y}_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a	Y_{a1}	Y_{a2}	...	Y_{an_a}	$Y_{a.}$	$\overline{Y}_{a.}$
					$Y_{..}$	$\overline{Y}_{..}$

comparison procedures enable us to see which means are significantly different from the others.

When two or more factors are varied at different levels, and we seek to model their effects on a response variable, we have multi-factor or multi-way ANOVA models. In this chapter, we describe statistical analysis for two-factor models, again resulting in ANOVA F-tests. In a two-factor experiment, we can distinguish between an additive model and a model with an interaction between the two factors.

When one or more assumptions for the parametric procedure, i.e., the ANOVA F-test, are not satisfied by the data, we discuss nonparametric procedures that are rank-based, the Kruskal-Wallis and Friedman procedures, that are similar to the rank-based procedures for two samples that we saw in Chapter 3. For a detailed discussion and design implications underlying these models, see Montgomery (2017) or Kutner et al. (2005).

4.2 Essential Terminology and Notation

✓ **Factors** are categorical explanatory variables consisting of at least two levels. We are interested in comparing the average responses at different levels of factors.

✓ **Designed experiments** are controlled studies in which one or more treatments are applied to experimental units or subjects in order to observe the effect of different levels of these treatments on a response variable of interest.

✓ In an **unbalanced design**, not all levels of a factor have the same number of observations. This can be by design, or because some observations become missing while the experiment is conducted. In Table 4.1, there are n_i observations at the ith level of a single factor, Factor A, $i = 1, \ldots, a$, leading to unbalanced data. Here, $N = \sum_{i=1}^{a} n_i$ denotes the total number of observations.

✓ A **balanced design** consists of the same number of observations at each level of a factor. For example, in a single-factor model, Factor A can have n observations in each of its a levels, for a total of $N = an$ observations.

✓ A **full factorial experimental design** consists of two or more factors, each with a set of discrete levels, and whose experimental units take on all possible combinations of

these levels across all such factors. For example, an experiment with two factors each at two levels is called a 2^2-factorial design. A design with 3 factors each at 2 levels is a 2^3 design, etc.

✓ **Analysis of variance (ANOVA)** is a statistical method to compare the mean responses between a different normal populations, all having the same variance. It can be regarded as an extension of the pooled two-sample t-procedure to more than two populations. An ANOVA model is a linear model which relates a continuous-valued response to effects due to different levels of factors (treatments).

The phrase 'Analysis of variance' is a bit misleading. The name may lead us to think that our focus is on estimating or testing variances in a linear model. However, we are in fact making inference (estimation, test) about mean effects *using* variance-like quantities for constructing suitable test statistics and obtaining critical values. The inference is based on the ratio between two estimates of the error variance, which are only equal if the means of all the groups are the same.

✓ A **single-factor ANOVA model** is a linear model of a response variable Y on mean effects from a single qualitative predictor, Factor A, with a levels, denoted by $\mu_1, \mu_2, \ldots, \mu_a$, say. It is usual to represent μ_i as $\mu + \tau_i$ for $i = 1, \ldots, a$, where μ is an overall effect common to all a levels and τ_i is an individual level-specific effect beyond μ.

✓ **Multiple comparisons**, also referred to as multiple testing, or multiplicity, consists of looking at a set of statistical inferences simultaneously on several treatment means, say, $\mu_1, \mu_2, \ldots, \mu_a$ in an ANOVA context. It is usually used to determine which pairwise mean differences $\mu_i - \mu_\ell$ may have caused rejection of the overall ANOVA F-test.

✓ **A contrast** is a linear combination of the means in an ANOVA model of the form $c_1\mu_1 + c_2\mu_2 + \ldots + c_a\mu_a$ such that the constant coefficients sum to zero, i..e, $c_1 + c_2 + \ldots + c_a = 0$.

✓ **Two-factor additive ANOVA models** are useful in experiments or observational studies consisting of two factors, Factor A at a levels, and Factor B at b levels. In a *cross-classified model*, every level of Factor A can be studied with every level of Factor B, and units are observed at all possible combinations of all the levels across all the factors, and main effects of these on the response Y can be estimated from the data. A balanced two factor model has n replicates for each of the a levels of Factor A and b levels of Factor B, and the total number of observations is $N = abn$. See Table 4.2 for a layout of this balanced design. If there are n_{ij} observations in the (i, j)th cell, the unbalanced design will have $N = \sum_{i=1}^{a} \sum_{j=1}^{b} n_{ij}$ observations.

✓ **Two-factor ANOVA models with interactions** are useful in situations where the difference in response between the levels of Factor A is not the same at all levels of Factor B, and we include an interaction effect between the two factors in addition to their main effects. Data for this model also looks like Table 4.2.

✓ **Two-factor nested ANOVA models** are useful in cases where some, but not all of the levels of Factor B occur within each level of Factor A. For example, in a study of effects of different looms and different operators on fabric strength, the manufacturer may be unable to assign all operators to all the looms due to logistical constraints; each

TABLE 4.2: Data from a Two-factor Balanced Design

Factor A	Factor B			
	1	2	...	b
1	$Y_{111}, Y_{112}, \ldots, Y_{11n}$	$Y_{121}, Y_{122}, \ldots, Y_{12n}$	...	$Y_{1b1}, Y_{1b2}, \ldots, Y_{1bn}$
2	$Y_{211}, Y_{212}, \ldots, Y_{21n}$	$Y_{221}, Y_{222}, \ldots, Y_{22n}$	...	$Y_{2b1}, Y_{2b2}, \ldots, Y_{2bn}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a	$Y_{a11}, Y_{a12}, \ldots, Y_{a1n}$	$Y_{a21}, Y_{a22}, \ldots, Y_{a2n}$	...	$Y_{ab1}, Y_{ab2}, \ldots, Y_{abn}$

operator is therefore not cross-classified with each loom, but rather only runs one loom, say.

4.3 One-factor ANOVA model

To illustrate the ideas in this chapter we use the U.S. violent crime data set, and our research question is stated as follows. We use the U.S. violent crime data from Example 3.3.1 to answer the following question:

Research question #1

Does the number of violent crimes depend on the size of the state?

We are asking whether the number of violent crimes depend on the size of the state. In subsequent chapters, we will see how to answer this question when state size is considered a continuous variable. Here, we define five levels based on the quintiles (20, 40, 60, and 80 percentiles), with ten states in each level, and use the ANOVA method which we discuss in this chapter. To create the categorical variable, we use the cut function, as follows:

```
ViolentCrime2010 <- read_excel("Data/Violent Crime-by
    state-2010-table-5.xls", sheet=1, range = "A4:M507", trim_ws = TRUE)
colnames(ViolentCrime2010) = gsub("\n", "_", colnames(ViolentCrime2010))
States <- as.character(na.omit(ViolentCrime2010$State))
States <- States[c(1:8,10:51)]
# read the total violent crime numbers, and population sizes:
VCrime2010 <- ViolentCrime2010$Violent_crime[which(ViolentCrime2010$Area ==
    "State Total")]
Population <-
    as.numeric(ViolentCrime2010$Population[which(ViolentCrime2010$Area ==
    "State Total")])
# Create a categorical variable with 5 levels of state sizes:
stateSize <- cut(Population,breaks =
    quantile(Population,c(0,0.2,0.4,0.6,0.8,1)), right = T, include.lowest =
    T, labels = c("A","B","C","D","E"))
UScrime <- data.frame(logCrime=log10(VCrime2010), Population, stateSize)
```

The categorized `stateSize` variable has $a = 5$ levels. In each level, there are $n = 10$

states. Let the response Y_{ij} be the number of violent crimes in the jth state within the ith level, where $j = 1, \ldots, 10$ and $i = 1, \ldots, 5$. The one-factor ANOVA model expresses the response Y_{ij} as the sum of an overall mean parameter, μ, an effect due to the ith level of `stateSize`, τ_i, and a random error ε_{ij}:

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij}, \ j = 1, 2, \ldots, n, \ i = 1, 2, \ldots, a. \tag{4.1}$$

Let the total number of observations be$N = an$. In general, the number of observations within each of the a levels can be different, say, $n_i, \ i = 1, \ldots, a$. That is, in (4.1), $j = 1, 2, \ldots, n_i$. Then, $N = \sum_{i=1}^{a} n_i$. One of the main tasks in an ANOVA procedure is to see whether these means are different, and if so, which of them are different.

In order for the ANOVA method to be valid, we assume that

- the observations are independent,
- the distribution of the observations is normal,
- the variance of the observations is the same in all five categories of the `stateSize` variable.

In Chapter 3, we saw that the distribution of the number of violent crimes is skewed, but the normality assumption seems reasonable after a log-transformation. In the ANOVA method, we usually verify the normality assumption on the residuals after fitting the model to the data.

We use the method of least squares (LS) to solve for the unknown parameters $\mu, \tau_1, \ldots, \tau_a$ and σ^2 in terms of data summaries. The LS approach consists of minimizing the sum of squares

$$S(\mu, \tau_1, \ldots, \tau_a) = \sum_{i=1}^{a} \sum_{j=1}^{n} \varepsilon_{ij}^2 = \sum_{i=1}^{a} \sum_{j=1}^{n} (Y_{ij} - \mu - \tau_i)^2.$$

The fitted values for responses in level $i, \ i = 1, \ldots, a$ are

$$\widehat{Y}_{ij} = \hat{\mu}_i = \mu^0 + \tau_i^0 = \overline{Y}_{i\cdot}, \ j = 1, \ldots, n. \tag{4.2}$$

The residuals are defined by

$$e_{ij} = Y_{ij} - \widehat{Y}_{ij} = Y_{ij} - \overline{Y}_{i\cdot}, \ j = 1, \ldots, n, \ i = 1, \ldots, a, \tag{4.3}$$

and are estimates of the unobserved random errors ε_{ij}'s.

To check the validity of the equal variance assumption, we draw side-by-side boxplots of the logarithm (base 10) of the 2010 violent crimes, and we draw a Q-Q plot of the residuals from the fitted ANOVA model (Fig. 4.1). We use the *ggplot2* package in the following code:

```
# Create boxplots by group, and a Q-Q plot of the residuals from the ANOVA
    model
pb <- ggplot(UScrime, aes(x=stateSize,  y=logCrime, fill=stateSize)) +
    geom_boxplot(show.legend = F)
aovmod <- aov(logCrime ~ stateSize, data = UScrime)
aovres <- residuals(aovmod)
pq <- qplot(sample = aovres)
grid.arrange(pb, pq, ncol=2)
```

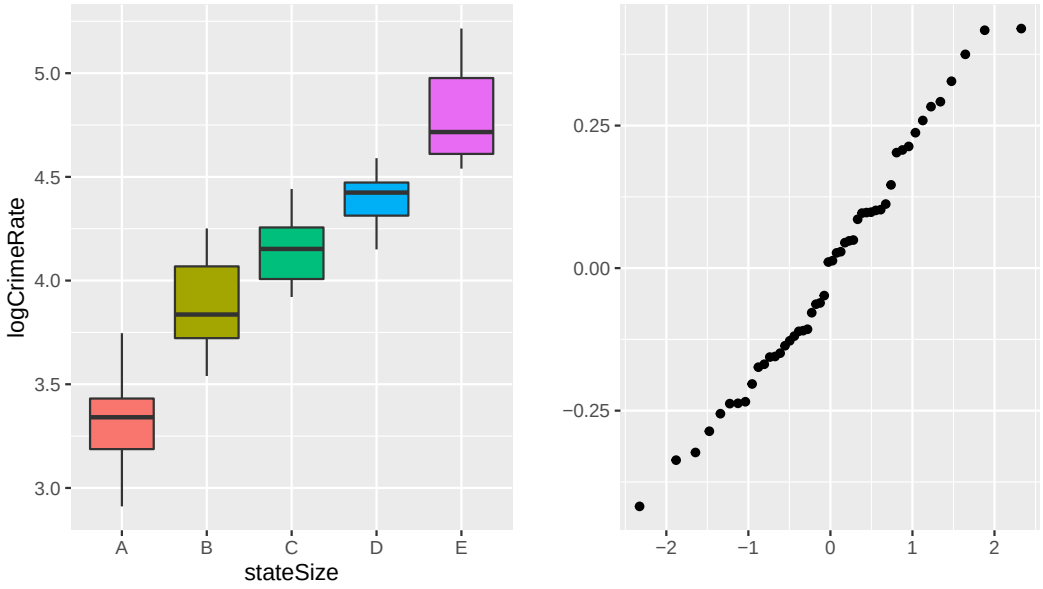

FIGURE 4.1: Left: Side-by-side boxplots of log of the number of violent crimes by state size. Right: Q-Q plot for the residuals from the ANOVA model.

The spreads of the five boxplots appear to be similar, which supports the assumption of equal variance. We can also test this formally, using Levene's test:

```
# Use Levene's Test to check equality of variance
leveneTest(logCrime ~ stateSize, data = UScrime)
```

which gives the following output:

```
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  4  1.1837 0.3309
      45
```

The p-value is 0.33, which means that we cannot reject the null hypothesis of equal variance.

To check whether the data are normally distributed we obtain the Q-Q plot of the residuals from the ANOVA model (Fig. 4.1, Right). The fact that the points lie close to a straight line suggests that the normality assumption is reasonable. We can also use the Shapiro-Wilk test:

```
shapiro.test(aovres)
```

which yields a p-value of 0.53, so indeed, we cannot reject the null hypothesis that the residuals are from a normal distribution.

Now that we have verified that the ANOVA assumptions hold, we may use the model output to answer our research question:

```
summary(aovmod)
```

This yields the following output:

```
             Df Sum Sq Mean Sq F value Pr(>F)
stateSize     4 12.132  3.0329   65.16 <2e-16 ***
Residuals    45  2.095  0.0465
```

Note that since there are five state sizes, the degrees of freedom associated with the model are four. The p-value is very small, which means that we can reject the null hypothesis that the logarithm of the number of violent crimes is the same in all five state size categories.

The output shown above follows from the ANOVA decomposition, which consists of partitioning the total variability in the response Y into the *between-group* or between-level variation and the *within-group* or due to error (residual) variation. Table 4.3 shows the general form of the ANOVA table in a one-factor ANOVA model.

TABLE 4.3: Analysis of Variance (ANOVA) Table

Source	DF	Sums of Squares	Mean Squares	F-stat	Prob $> F$
Model	$a-1$	SSA	MSA	$F = \frac{\text{MSA}}{\text{MSE}}$	p-value
Error	$N-a$	SSE	MSE		
C Total	$N-1$	SST			

In the table, MSA and MSE are respectively the treatment mean square and error mean square given by

$$\text{MSA} = \frac{\text{SSA}}{a-1} \text{ and } \text{MSE } = \frac{\text{SSE}}{N-a}. \tag{4.4}$$

The F-test enables us to see whether the means of the a groups are equal by comparing the ratio of MSA to MSE. The null and alternative hypotheses can be expressed in terms of comparing the τ_i's:

$$\begin{aligned} H_0 : &\quad \tau_1 = \tau_2 = \ldots = \tau_a \text{ versus} \\ H_1 : &\quad \text{not all treatment effects } \tau_i \text{ are equal.} \end{aligned} \tag{4.5}$$

The one-factor ANOVA F-statistic has the form

$$F = \frac{\text{MSA}}{\text{MSE}}\,. \tag{4.6}$$

Provided the null hypothesis is true, $F \sim F_{a-1,N-a}$, i.e., an F-distribution with numerator d.f. $a-1$ and denominator d.f. $N-a$. We reject the null hypothesis H_0 if

$$\begin{aligned} F &\geq F_{1-\alpha,a-1,N-a}, \text{ or,} \\ \text{p-value} &= P(F_{a-1,N-a} \geq F) \leq \alpha, \end{aligned} \tag{4.7}$$

where α is a prespecified level of significance. If the data does not provide evidence to reject H_0, we conclude that the levels of the factor have no effect on the response means.

The F-test does not tell us which levels of the factor are different - only that not all are the same. The boxplots imply that the number of violent crimes may be very different for any pair of groups, but to test it formally we need to use a method which accounts for multiple testing. Multiple comparison procedures can enable us to see which of the a means are different, while controlling the probability of false positives.

To perform all pairwise comparisons of the five means, we can run the following test called Tukey's Honest Significant Difference post-hoc test. This procedure is based on the idea that: if we determine a critical value for the difference between the largest and the smallest sample means, then any other pair of sample means that differ by at least this critical value would also imply a difference in the corresponding level means.

```
TukeyHSD(aovmod)
```

At the 5% level, only the comparison between levels C and D is not significant, after accounting for multiple testing. The positive differences in all other comparisons are in agreement with what we observed in the boxplots, namely that the number of violent crimes increases as the size of the state increases.

```
  Tukey multiple comparisons of means
   0.95  family-wise confidence level

Fit: aov(formula = logCrime ~ stateSize, data = UScrime)

$stateSize
         diff          lwr       upr     p adj
B-A 0.5474749  0.273319688 0.8216301 0.0000091
C-A 0.8294512  0.555296016 1.1036064 0.0000000
D-A 1.0585911  0.784435918 1.3327463 0.0000000
E-A 1.4660286  1.191873428 1.7401838 0.0000000
C-B 0.2819763  0.007821143 0.5561315 0.0410530
D-B 0.5111162  0.236961045 0.7852714 0.0000324
E-B 0.9185537  0.644398554 1.1927089 0.0000000
D-C 0.2291399 -0.045015283 0.5032951 0.1410715
E-C 0.6365774  0.362422226 0.9107326 0.0000004
E-D 0.4074375  0.133282324 0.6815927 0.0010498
```

Putting the numbers in context, the difference of 1.466 between the ten largest states and the ten smallest ones means that $\log_{10}(y_E) - \log_{10}(y_A) = \log_{10}(y_E/y_A) = 1.466$, so, transforming back to the original scale, we get $y_E/y_A = 10^{1.466} = 29.2$. That is, the number of violent crimes in the ten largest states is 29 times greater than for the ten smallest ones. Because crime data is known to be related to population size, quite often in the literature concerning crimes the data is transformed in to rate, or rate per 100,000 people: TotalCrimes/(Population/100000). Try the following:

```
plot(VCrime2010/(Population/100000) ~ stateSize)
```

Notice that even when looking at the rate per 100,000, there appears to be a significant difference between levels A and E of the state size factor.

We can also use other types of multiple-testing adjustments when we compare all the means. For example, we may use Bonferroni's method, which requires fewer assumptions than Tukey's method, but is more conservative (meaning that it yields p-values which are greater than or equal to ones produced by Tukey's adjustment). To use Bonferroni's method, we use the pairwise.t.test function:

```
pairwise.t.test(UScrime$logCrime, UScrime$stateSize, p.adjust.method =
    "bonf")
```

The output is presented in a form of a table which contains the adjusted p-values for all possible pairs:

```
        Pairwise comparisons using t tests with pooled SD

data:  UScrime$logCrime and UScrime$stateSize

  A       B       C       D
B 9.5e-06 -       -       -
C 4.7e-10 0.0542  -       -
D 2.6e-13 3.4e-05 0.2188  -
E < 2e-16 2.4e-11 4.0e-07 0.0012

P value adjustment method: bonferroni
```

Finally, we could also use a non-parametric procedure instead of the ANOVA method, which can be particularly useful when the data do not satisfy the ANOVA assumptions. This ranks based method is an extension of the Wilcoxon rank-sum test to more than two populations. The Kruskal-Wallis statistic tests the null hypothesis that a level effects have identical distributions against the alternative hypothesis that not all the a distributions are the same. To perform the non-parametric Kruskal-Wallis test, by invoking the following code, but in this case the results are qualitatively the same as the ones obtained from ANOVA.

```
kruskal.test(logCrime ~ stateSize, data = UScrime)
```

4.4 Two-Factor Model

Cycling

Example 4.4.1 (Cycling to school) The cycling data has been introduced in Muralidharan and Prakash (2017) who studied the effect of a program in Bihar, India, which aimed to reduce the gender gap in secondary school enrollment. Girls who were scheduled to enroll in secondary school were provided bicycles, in the hope that it will facilitate getting to school and back in a timely manner, and therefore increase enrollment and graduation levels. The Government of Bihar provided funds for bicycles for 160,000 girls to be distributed in 2007–2008. The enrollment dataset contains 75,744 observations, of which 61,920 are from Bihar and 13,824 from Jharkhand. The data contains 24 columns, and

we will use the following: `enrollment`, `female` (indicator, 0 for male, 1 for female), `year`, `class` (grade), and `statecode`. Other data files associated with this study are available online (`https://nishithprakash.com/published-papers/`)

Research question #2

Did girls' enrollment in secondary school in Bihar increase as a result of the program which provided bicycles?

Data collection took place around 18 months after the Cycle program was launched. The typical age in which students enter 9th grade is 14-15, then children aged 16-17 at the time of the launch of the program were not slated to receive bicycles when they had to decide whether to enroll in a secondary school. So, the 14-15 year old children at the time of the launch are considered the 'treated' group, while the 16-17 age group is the 'control'. Therefore, we define a variable called `n_year` which represents the treatment, and is a categorical (factor) variable. It is also possible to look at `n_year` as a continuous predictor and study the effect it may cause on enrollment; we discuss such regression analyses in Chapter 5.

There may be other events or trends in the region which may affect enrollment. To control for such possible effects, we also look at boys' enrollment in the same period. Since boys did not get bicycles, it makes sense to use the male student cohort as a control group. If we see a significant and equal increase in enrollment of boys, then we can conclude that the bicycle program did not have a big impact. If, on the other hand, the increase in girls' enrollment over that time period is greater than the increase for boys, then it may well be because of the program. (Note that the authors of the paper went further, and compared Bihar with Jharkhand, to ensure that a greater increase in enrollment of girls versus boys, did not occur in a province which did not have a similar program.)

In our analysis, we will consider two factors: the year (four levels, corresponding to four years prior to the program), and the gender (two levels.) The interaction between the two factors is of interest. If it turns out to be significant, than the rate of change in enrollment for girls is not the same as that for boys, and thus may be attributed to the Cycle program. The data is provided in Stata format, which requires us to use the haven package and the read_dta function. Note that the function returns an object of type 'tibble', which we convert to a data frame. We take the logarithm of the enrollment, as is done in the paper, because the population base is different in the two provinces. We convert the `female` and `year` variables to factors. In the following analysis (Table 1A in the paper) we restrict the dataset to the Bihar province (state code =1) and to students entering ninth grade in the years 2003-2006. Finally, we exclude any NA or -infinite enrollment values. The total number of observations in the `dat_table1` data frame is 20,266.

```
library(haven) # to import data from Stata
lenrolldat <- data.frame(read_dta("Data/bh_enroll_data_reg.dta"))
lenrolldat$lenrollment <- log(lenrolldat$enrollment)
lenrolldat$female <- as.factor(lenrolldat$female)
lenrolldat$n_year <- as.factor(lenrolldat$year - 2002)
dat_table1 <- lenrolldat[which(lenrolldat$class==9 & lenrolldat$statecode
    ==1 &  !is.na(lenrolldat$lenrollment) & lenrolldat$lenrollment > -Inf),]
```

The factor `female` has $a = 2$ levels and the factor `year` has $b = 4$ levels. We denote

by $Y_{ij\ell}$ the ℓth log enrollment (response) for the ith level of gender and jth level of year. The two-factor ANOVA model which includes main effects due to `female` and `year` and an interaction between the two factors is

$$Y_{ij\ell} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ij\ell},\ \ell = 1, \ldots, n,\ i = 1, \ldots, a,\ j = 1, \ldots, b. \tag{4.8}$$

Similar to the one-factor model, μ is the overall mean effect, τ_i is the effect due to the ith level of Factor A (here, `female`), and $\varepsilon_{ij\ell}$ are i.i.d. $N(0, \sigma^2)$ variables. Additionally, we have β_j which denotes the effect due to the jth level of Factor B (here, `year`), and $(\tau\beta)_{ij}$ which is the effect due to the interaction between the ith level of Factor A and the jth level of Factor B. The total number of model parameters is $p = 1 + a + b + ab$, and the total number of observations is $N = nab$. If we write the mean response in the (i, j)th cell as μ_{ij}, where

$$\mu_{ij} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij},$$

we can interpret the interaction effect $(\tau\beta)_{ij} = \mu_{ij} - \mu - \tau_i - \beta_j$ as the left-over effect from additivity, i.e., $\mu_{ij} = \mu + \tau_i + \beta_j$ (as discussed below).

We fit an interaction model using the lm() function as follows:

```
panelA <- lm(lenrollment ~ female*n_year, data = dat_table1)
summary(aov(panelA))
summary(panelA)
```

This gives the following output:

```
                Df Sum Sq Mean Sq  F value   Pr(>F)
female           1   2768  2768.2 3619.069  < 2e-16 ***
n_year           3    318   106.1  138.673  < 2e-16 ***
female:n_year    3     17     5.7    7.479 5.32e-05 ***
Residuals    20258  15495     0.8
```

The interaction term is significant. The formulas for the ANOVA decomposition for the two-factor model with interaction is shown in Table Table 4.4. As we saw in the one-factor model, the mean squares MSA, MSB, MSAB, and MSE are obtained by dividing the respective sums of squares by the corresponding d.f. shown in Table 4.4.

TABLE 4.4: ANOVA table for a two-factor model with interaction

Source	DF	SS	MS	F_0
A	$a-1$	SSA	MSA$= \frac{\text{SSA}}{a-1}$	$\frac{\text{MSA}}{\text{MSE}}$
B	$b-1$	SSB	MSB$= \frac{\text{SSB}}{b-1}$	$\frac{\text{MSB}}{\text{MSE}}$
AB	$(a-1)(b-1)$	SSAB	MSAB$= \frac{\text{SSAB}}{(a-1)(b-1)}$	$\frac{\text{MSAB}}{\text{MSE}}$
Error	$ab(n-1)$	SSE	MSE$= \frac{\text{SSE}}{ab(n-1)}$	
Total	$abn-1$	SST		

The forms of the F-test statistics for the different hypotheses of interest are shown in Table 4.5. We also show the F-distribution that each test statistic follows if the corresponding null hypotheis were true.

TABLE 4.5: F-tests in a two-factor model with interaction

Hypotheses	Test statistic	Null distribution
$H_{AB}: (\tau\beta)_{ij} = 0\ \forall i,j$ versus at least one $(\tau\beta)_{ij}$ is 0	$F_{AB} = \frac{SS_{AB}/(a-1)(b-1)}{SSE/ab(n-1)}$	$F_{(a-1)(b-1),ab(n-1)}$
$H_B: \beta_j + \sum_{i=1}^{a}(\tau\beta)_{ij}/a$ equal $\forall j$	$F_B = \frac{SS_B/(b-1)}{SSE/ab(n-1)}$	$F_{(b-1),ab(n-1)}$
$H_A: \tau_i + \sum_{j=1}^{b}(\tau\beta)_{ij}/b$ equal $\forall i$	$F_A = \frac{SS_A/(a-1)}{SSE/ab(n-1)}$	$F_{(a-1),ab(n-1)}$

The output gives more details about the effect of each combination of year (treatment) and gender as shown below. These are the least squares solutions of the model parameters; for details, see the appendix.

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     4.32159    0.01784 242.224  < 2e-16 ***
female1        -0.82483    0.02475 -33.331  < 2e-16 ***
n_year2         0.07883    0.02514   3.136  0.00171 **
n_year3         0.17235    0.02513   6.858 7.18e-12 ***
n_year4         0.25287    0.02514  10.059  < 2e-16 ***
female1:n_year2  0.06947    0.03487   1.992  0.04639 *
female1:n_year3  0.10758    0.03485   3.087  0.00203 **
female1:n_year4  0.16015    0.03484   4.597 4.31e-06 ***
---
Residual standard error: 0.8746 on 20258 degrees of freedom
Multiple R-squared:  0.1669,        Adjusted R-squared:  0.1666
F-statistic: 579.6 on 7 and 20258 DF,  p-value: < 2.2e-16
```

The p-values in the table of coefficients indicate that the coeffcients are all significant. These are based on marginal t-tests, and do not account for multiple testing. To do that, we can use Tukey's procedure for testing hypotheses of interest $H_0: \mu_i = \mu_\ell,\ i \neq \ell$, where $i, \ell = 1, \ldots, a$.

```
library(lsmeans) # to perform pairwise comparisons
marginal <- lsmeans(panelA, pairwise ~ female:n_year, adjust="tukey")
marginal$contrasts
```

```
 contrast  estimate     SE    df t.ratio p.value
 0 1 - 1 1   0.8248 0.0247 20258  33.331 <.0001
 0 1 - 0 2  -0.0788 0.0251 20258  -3.136 0.0365
 0 1 - 1 2   0.6765 0.0247 20258  27.424 <.0001
 0 1 - 0 3  -0.1723 0.0251 20258  -6.858 <.0001
 0 1 - 1 3   0.5449 0.0246 20258  22.110 <.0001
 0 1 - 0 4  -0.2529 0.0251 20258 -10.059 <.0001
 0 1 - 1 4   0.4118 0.0246 20258  16.730 <.0001
 1 1 - 0 2  -0.9037 0.0246 20258 -36.661 <.0001
 1 1 - 1 2  -0.1483 0.0242 20258  -6.135 <.0001
 1 1 - 0 3  -0.9972 0.0246 20258 -40.464 <.0001
 1 1 - 1 3  -0.2799 0.0241 20258 -11.592 <.0001
 1 1 - 0 4  -1.0777 0.0247 20258 -43.718 <.0001
 1 1 - 1 4  -0.4130 0.0241 20258 -17.126 <.0001
 0 2 - 1 2   0.7554 0.0246 20258  30.741 <.0001
 0 2 - 0 3  -0.0935 0.0250 20258  -3.736 0.0046
 0 2 - 1 3   0.6237 0.0245 20258  25.410 <.0001
 0 2 - 0 4  -0.1740 0.0250 20258  -6.950 <.0001
 0 2 - 1 4   0.4906 0.0245 20258  20.013 <.0001
 1 2 - 0 3  -0.8489 0.0246 20258 -34.555 <.0001
 1 2 - 1 3  -0.1316 0.0241 20258  -5.469 <.0001
 1 2 - 0 4  -0.9294 0.0246 20258 -37.820 <.0001
 1 2 - 1 4  -0.2647 0.0240 20258 -11.013 <.0001
 0 3 - 1 3   0.7173 0.0245 20258  29.226 <.0001
 0 3 - 0 4  -0.0805 0.0250 20258  -3.216 0.0284
 0 3 - 1 4   0.5842 0.0245 20258  23.832 <.0001
 1 3 - 0 4  -0.7978 0.0245 20258 -32.497 <.0001
 1 3 - 1 4  -0.1331 0.0240 20258  -5.543 <.0001
 0 4 - 1 4   0.6647 0.0245 20258  27.109 <.0001

P value adjustment: tukey method for comparing a family of 8 estimates
```

At the 5% level, all the comparisons are significant. For example, the difference in enrollment for girls between year 1 and year 4 appears in the line which starts with '1 1 - 1 4', and the value is -0.43 (an increase of 0.43 if we subtract year 1 from year 4.), For the boys, the change from year 1 to year 4 is denoted by '0 1 - 0 4', and it is -0.25. So, enrollment went up for both groups, but more so for girls.

A convenient way to visualize the results is via an interaction plot. R has several functions to generate such plots, and we use the one in the phia package, which gives the plots in Figure 4.2.

```
library(phia) # for the interactionMeans function
IM <- interactionMeans(panelA)
plot(IM)
```

The top-left panel shows the overall difference in log-enrollment between girls and boys. Clearly, the enrollment for boys is much higher. The top-right panel shows the change in enrollment over the years for boys (black-solid line) and girls (dashed-red). Although the enrollment of boys is higher throughout, the difference between the genders decreases over time, which is seen by the steeper slope for girls. The lower-left plot shows a similar picture, but here the gender is on the x-axis and we see how the difference between boys and girls has decreased from year 1 to year 4. The bottom-right plot shows the overall difference between years. When considering the two cohorts together, there is a steady increase in enrollment.

From this detailed analysis, we can conclude that the enrollment in Bihar has increased

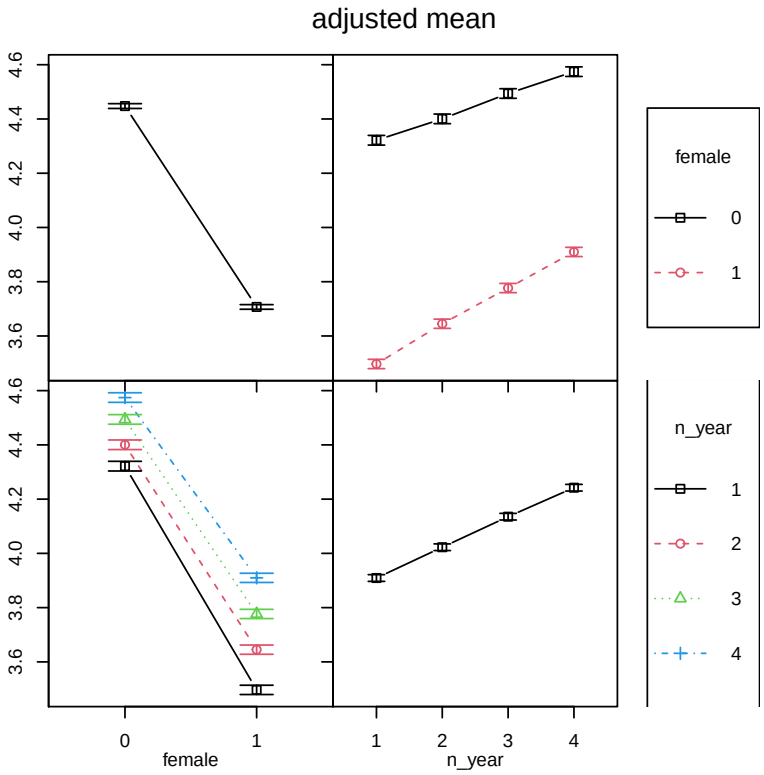

FIGURE 4.2: An interaction plot for the cycling data.

during the period in the study, but much more so for girls. Since we accounted for the overall effect of time on all children in the province, a plausible conclusion from the interaction model is that the faster increase for girls may, indeed, be due to the Cycle program.

Remark 1. If the ANOVA output had indicated that there is no significant interaction between the two factors, we could fit a two-factor additive ANOVA model with the form

$$Y_{ij\ell} = \mu + \tau_i + \beta_j + \varepsilon_{ij\ell},\ \ell = 1, \ldots, n,\ i = 1, \ldots, a,\ j = 1, \ldots, b. \tag{4.9}$$

Details of the least squares solutions and ANOVA decomposition for this model are shown in the Appendix.

Appendix Chapter 4 -Statistical Details

Statistical details - one-factor ANOVA model

The one-factor ANOVA model expresses Y_{ij} as the sum of an overall mean parameter, μ, an effect due to the ith level of Factor A, τ_i, and random errors ε_{ij} shown in (4.1) for the balanced case with the same number of observations, n, in each level. The unbalanced one-factor model with a levels is

$$Y_{ij} = \mu + \tau_i + \varepsilon_{ij},\ j = 1, 2, \ldots, n_i,\ i = 1, 2, \ldots, a. \tag{4.10}$$

Let $N = \sum_{i=1}^{a} n_i$. We assume that $\varepsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma^2)$, which includes assumptions of independence, equal variances, and normality. The assumptions on the errors also imply that the observations Y_{ij} are mutually independent normal random variables with the same variance σ^2,

$$Y_{ij} \sim N(\mu + \tau_i, \sigma^2)\ j = 1, \ldots, n_i,\ i = 1, 2, \ldots, a, \tag{4.11}$$

where, the mean response in the ith group is $\mu_i = \mu + \tau_i,\ i = 1, \ldots, a$, and may be different in the different groups. One of the main tasks in an ANOVA procedure is to see whether these means are different, and if so, which of them are different. We start with solving for the unknown parameters $\mu, \tau_1, \ldots, \tau_a$ and σ^2 in terms of data summaries.

Least Squares Approach

The method of least squares (LS) consists of minimizing the sum of squares

$$S(\mu, \tau_1, \ldots, \tau_a) = \sum_{i=1}^{a} \sum_{j=1}^{n_i} \varepsilon_{ij}^2 = \sum_{i=1}^{a} \sum_{j=1}^{n_i} (Y_{ij} - \mu - \tau_i)^2.$$

Setting the first derivatives of $S(\mu, \tau_1, \ldots, \tau_a)$ with respect to $\mu, \tau_1, \ldots, \tau_a$ equal to zero leads to $p = a + 1$ normal equations:

$$\left.\frac{\partial S}{\partial \mu}\right|_{\mu^0, \tau_i^0} = 0, \quad \text{and} \quad \left.\frac{\partial S}{\partial \tau_i}\right|_{\mu^0, \tau_i^0} = 0, \quad i = 1, \cdots, a,$$

which have the form

$$N\mu^0 + \sum_{i=1}^{a} n_i\tau_i^0 = Y_{\cdot\cdot} \quad \text{and} \quad n_i(\mu^0 + \tau_i^0) = Y_{i\cdot}, \quad i = 1, \cdots, a.$$

Since the sum of the last a normal equations is equal to the first, the $a+1$ normal equations are linearly dependent and hence no unique solution exists in the *overparametrized* model (4.1). We can impose one constraint on the group effects τ_i, such as $\sum_{i=1}^{a} n_i\tau_i = 0$ and then solve the normal equations to get

$$\begin{aligned} \mu^0 &= \overline{Y}_{\cdot\cdot}, \\ \tau_i^0 &= \overline{Y}_{i\cdot} - \overline{Y}_{\cdot\cdot}, \ i = 1, \ldots, a; \end{aligned} \tag{4.12}$$

that is, the estimate of the overall mean μ is the overall average of all N observations, while the estimate of the ith group effect τ_i is the difference between the average of the sample observations under the ith treatment and the overall average. The LS estimate of the error variance is

$$\widehat{\sigma}^2 = \frac{SSE}{(N-a)}, \tag{4.13}$$

where the error sum of squares (or residual sum of squares) is

$$SSE = \sum_{i=1}^{a}\sum_{j=1}^{n_i}(Y_{ij} - \overline{Y}_{i\cdot})^2, \tag{4.14}$$

and is one component in the ANOVA decomposition of the total variation in the response Y.

The fitted values for responses in group i, $i = 1, \ldots, a$ are

$$\widehat{Y}_{ij} = \hat{\mu}_i = \mu^0 + \tau_i^0 = \overline{Y}_{i\cdot}, \ j = 1, \ldots, n_i. \tag{4.15}$$

The residuals are defined by

$$e_{ij} = Y_{ij} - \widehat{Y}_{ij} = Y_{ij} - \overline{Y}_{i\cdot}, \ j = 1, \ldots, n_i, \ i = 1, \ldots, a, \tag{4.16}$$

and are estimates of the unobserved random errors ε_{ij}'s. We will use standardized residuals for checking adequacy of the model fit.

ANOVA Decomposition and F-test

The ANOVA decomposition consists of partitioning the total variability in Y into the *between-group* or between-treatment variation and the *within-group* or due to error (residual) variation:

$$\text{SST} = \text{SSA} + \text{SSE}, \tag{4.17}$$

where SST is the Total Sum of Squares, SSA is the Sum of Squares due to Factor A and SSE is the Sum of Squares due to Error, given by

$$\begin{aligned} \text{SST} &= \sum_{i=1}^{a}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{..})^2, \\ \text{SSA} &= \sum_{i=1}^{a} n_i(\overline{Y}_{i.}-\overline{Y}_{..})^2, \\ \text{SSE} &= \sum_{i=1}^{a}\sum_{j=1}^{n_i}(Y_{ij}-\overline{Y}_{i.})^2. \end{aligned} \tag{4.18}$$

The corresponding breakdown in the degrees of freedom associated with these sums of squares is

$$\begin{array}{ccccc} \text{d.f. of SST} & = & \text{d.f. of SSA} & + & \text{d.f. of SSE} \\ \Updownarrow & & \Updownarrow & & \Updownarrow \\ N-1 & = & (a-1) & + & (N-a) \end{array}$$

The analysis of variance is usually presented in an ANOVA table (see Table 4.3). In the table, MSA and MSE are respectively the treatment mean square and error mean square given by (see (4.4)) $\text{MSA} = \frac{\text{SSA}}{a-1}$ and $\text{MSE} = \frac{\text{SSE}}{N-a}$.

The one-factor ANOVA F-procedure is used to test whether the means of the a populations are equal by comparing the ratio of MSA to MSE. The null and alternative hypotheses are shown in (4.5) as

$$\begin{aligned} H_0 &: \quad \tau_1 = \tau_2 = \ldots = \tau_a \text{ versus} \\ H_1 &: \quad \text{not all treatment effects } \tau_i \text{ are equal,} \end{aligned}$$

and the one-factor ANOVA F-statistic has the form shown in (4.6) as

$$F = \frac{\text{MSA}}{\text{MSE}}\,.$$

Under the null hypothesis, $F \sim F_{a-1,N-a}$, i.e., an F-distribution with numerator d.f. $a-1$ and denominator d.f. $N-a$, so that, at a prespecified level of significance α, we reject the null hypothesis H_0 if (see (4.19))

$$\begin{aligned} F &\geq F_{1-\alpha,a-1,N-a}, \text{ or,} \\ p-\text{value} &= P(F_{a-1,N-a} \geq F) \leq \alpha. \end{aligned}$$

If the data does not provide evidence to reject H_0, we conclude that the levels of the factor have no effect on the response means.

Note that we can also express the hypotheses in terms of comparing the μ_i's instead of τ_i's, i.e.,

$$\begin{aligned} H_0' &: \quad \mu_1 = \mu_2 = \ldots = \mu_a \text{ versus} \\ H_1' &: \quad \text{not all } \mu_i \text{ are equal (or, at least one } \mu_i \text{ is different).} \end{aligned} \tag{4.19}$$

We can easily carry out inference about individual means. The $100(1-\alpha)\%$ confidence interval for μ_i is given by

$$\hat{\mu}_i \pm t_{1-\alpha/2,N-a}\frac{\hat{\sigma}}{\sqrt{n_i}} = \overline{Y}_{i.} \pm t_{1-\alpha/2,N-a}\frac{\sqrt{\text{MSE}}}{\sqrt{n_i}}. \tag{4.20}$$

Multiple Comparisons

Suppose the overall F-test rejects the null hypothesis that all the treatment effects are equal, so there is evidence that some of the $\binom{a}{2} = a(a-1)/2$ pairs of means are significantly different; however, the test cannot tell which pairs are significantly different. Finding such pairs requires testing equality for each pair. This type of multiple testing is known as a multiple comparison. Since it is done *after* an overall ANOVA analysis has indicated the potential existence of significantly different pairs, it is called a *post hoc* analysis. However, if the goal from the beginning of the data analysis is to identify significantly different pairs, then multiple comparisons can be done regardless of other statistical tests on the data, and thus, need not be *post hoc*. For *post hoc* multiple comparisons, the effects to be compared must be planned in advance.

Many issues that arise in multiple testing are due to the increasing difficulty of controlling the Type I error as more hypotheses are simultaneously tested. By definition, a Type I error, also known as a false positive (FP), is the incorrect rejection of a null hypothesis that is in fact true. As an example, suppose we conduct independent tests on $m = 200$ null hypotheses, each at the same significance level $\alpha = 0.05$. Suppose at least half of these are true. Then the probability of at least one FP is

$$\begin{aligned} P(\text{at least one FP}) &= 1 - P(\text{no FP in the tests on the true nulls}) \\ &\geq 1 - P(\text{no FP in the test on the first true null})^{100} \\ &= 1 - (1-\alpha)^{100} = 0.994. \end{aligned}$$

If the probability is to be controlled at a level no more than 5%, then α has to be less than 5.13×10^{-4}. Such a small significance level makes it difficult to reject any false nulls, resulting in lower power of testing. Thus, balancing Type I error control and power requires more than adjusting a single significance level α. In high-throughput domains such as genomics or business, several thousand (or even more) statistical tests are performed, and multiple testing is like finding needles in a haystack. It becomes critical to get a good tradeoff between Type I error control and power. There are several multiple comparison procedures of which we consider only the Least significant difference, Bonferroni and Tukey procedures. For details and other procedures, please see Montgomery (2017) or Kutner et al. (2005).

Least Significant Difference (LSD) Procedure

Fisher's *least significance difference* (LSD) procedure is useful for making pairwise comparisons among a set of a treatment means. The idea is that if we know the standard error (the denominator in the t-test), we can find the smallest number that has to be in the numerator (the difference between the means of two groups) so that the ratio is large enough to be statistically significant. For any pair $i \neq \ell$,

$$\frac{\hat{\mu}_i - \hat{\mu}_\ell}{\hat{\sigma}\sqrt{1/n_i + 1/n_\ell}} \sim t_{N-a}.$$

For a given level α, the LSD for comparing μ_i and μ_ℓ in a one-factor ANOVA model is

$$LSD_{i\ell} = t_{1-\alpha/2,N-a}\hat{\sigma}\sqrt{1/n_i + 1/n_\ell},$$

and μ_i and μ_ℓ are declared significantly different if

$$|\hat{\mu}_i - \hat{\mu}_\ell| > LSD_{i\ell} \quad \text{or} \quad \frac{|\hat{\mu}_i - \hat{\mu}_\ell|}{\hat{\sigma}\sqrt{1/n_i + 1/n_\ell}} > t_{1-\alpha/2,N-a}.$$

where $t_{1-\alpha/2,N-a}$ is the $100(1-\alpha/2)$th quantile of the t-distribution with $N-a$ degrees of freedom. The LSD procedure is not adjusted for multiplicity, and the procedure is generally criticized for inflating the Type I error rate. By this, we mean that the overall probability of incorrectly declaring some pair of treatment means significantly different, when they are in fact equal, is substantially higher than the specified α value.

In practice, the LSD procedure is used too often, because of its simplicity and ease of implementation. To address this, Fisher's protected LSD recommends that the LSD procedure be used only after the overall F-test for treatments is seen to be significant. Some simulation studies show that the error rate for the protected LSD procedure is controlled on an experimentwise basis at a level that is approximately equal to α.

Bonferroni Procedure

The Bonferroni procedure is based on the following first-order Bonferroni inequality:

$$P\Big(\bigcup_{i=1}^{m} A_i\Big) \leq \sum_{i=1}^{m} P(A_i). \tag{4.21}$$

We use this inequality for the events

$$A_{i\ell} = \text{reject } H_0\text{: } \mu_i = \mu_\ell \text{ when } H_0 \text{ is true.} \tag{4.22}$$

In order to maintain an overall significance level α, we must have

$$P\Big(\bigcup_{1\leq i<\ell\leq a} A_{i\ell}\Big) \leq \alpha. \tag{4.23}$$

If we require that for each $i < \ell$

$$P(A_{i\ell}) \leq \frac{\alpha}{\binom{a}{2}}, \tag{4.24}$$

then we maintain the overall level of significance to be less than α.

For each pair $1 \leq i < \ell \leq a$, compare the observed absolute difference $|\overline{Y}_{i\cdot} - \overline{Y}_{\ell\cdot}|$ with

$$B(n_i, n_\ell) = t_{1-\frac{\alpha}{a(a-1)},N-a}\sqrt{\text{MSE}\left(\frac{1}{n_i}+\frac{1}{n_\ell}\right)}. \tag{4.25}$$

If

$$|\overline{Y}_{i\cdot} - \overline{Y}_{\ell\cdot}| > B(n_i, n_\ell)\,, \tag{4.26}$$

we conclude that the corresponding treatment means are significantly different from each other.

Tukey's Procedure

Also called the Tukey's Honest Significant Difference post-hoc test, this procedure is useful when a researcher wants to make all possible pairwise comparisons of means while controlling the overall significance level at level α. The procedure is based on the distribution of the *Studentized range*, which is defined for balanced data as

$$Q = \frac{\overline{Y}_{\max} - \overline{Y}_{\min}}{\sqrt{\frac{\text{MSE}}{n}}},$$

where $\overline{Y}_{\max}$ and $\overline{Y}_{\min}$ are the largest and the smallest sample means. Let $q_{1-\alpha;a,\nu}$ denote the critical value of the Studentized range distribution with ν degrees of freedom, and let

$$T_\alpha = q_{1-\alpha;a,\nu}\sqrt{\text{MSE}/n}. \tag{4.27}$$

For $1 \le i < \ell \le a$, if

$$|\overline{Y}_{i\cdot} - \overline{Y}_{\ell\cdot}| > T_\alpha, \tag{4.28}$$

the procedure concludes difference in the treatment means. If the design is not balanced, the Tukey-Kramer procedure based on $W_{i\ell}$ (rather than T_α) is useful, where

$$W_{i\ell} = q_{1-\alpha;a,\nu}\sqrt{0.5\text{MSE}\Big(\frac{1}{n_i} + \frac{1}{n_\ell}\Big)}. \tag{4.29}$$

Assumptions	$y_{ij} \overset{iid}{\sim} N(\mu_i, \sigma^2)$		
Hypotheses	$H_0 : \mu_i = \mu_\ell$ vs. $H_1 : \mu_i \neq \mu_\ell$		
Statistic	$d_{i\ell} = \overline{Y}_{i\cdot} - \overline{Y}_{\ell\cdot}$		
Rejection Criterion	$\lvert d_{i\ell}\rvert > C_\alpha\sqrt{MSE\left(\frac{1}{n_i} + \frac{1}{n_\ell}\right)}$		
Confidence Interval	$d_{i\ell} \pm C_\alpha\sqrt{MSE\left(\frac{1}{n_i} + \frac{1}{n_\ell}\right)}$		
Method	Bonferroni	LSD	Tukey-Kramer
C_α	$t_{N-a,1-\frac{\alpha}{a(a-1)}}$	$t_{N-a,\alpha/2}$	$q_{1-\alpha;a,\nu}\sqrt{0.5}$
Objective	Pre-planned comparisons	Post planned comparisons upon rejection of the F test for equality of all means	Comparison of all pairs of means
R function	p.adjust, method=bonferroni	agricolae::LSD.test	TukeyHSD*

* Tukey's method requires equal sample sizes, but the function TukeyHSD 'incorporates an adjustment for sample size that produces sensible intervals for mildly unbalanced designs.'

Key formulas

In general (although not always true), we have

$$LSD(n_i, n_\ell) \le T_\alpha \le B(n_i, n_\ell). \tag{4.30}$$

Thus, a pair of means declared to be significantly different by the Bonferroni procedure will be significantly different under both the LSD and Tukey procedures.

Checking Model Assumptions

As we mentioned earlier, the one-factor ANOVA F-test is an a-sample extension of the two-sample pooled t-test. The assumptions needed for these two tests are similar, consisting of normality of the a populations, equality of the a population variances, and independence of the samples from these populations. While the last assumption follows from the design of the experiment, we discuss below how we check the normality and equal variances assumptions.

Checking normality. Recall that for $j = 1, \ldots, n_i$, Y_{ij} are i.i.d. from $N(\mu_i, \sigma^2)$. Provided n_i are not too small for any i, we can use normal Q-Q plots of the data in a samples. If the sample points in each group fall approximately along a straight line, normality may be assumed. However, in many designed experiments, n_i are likely to be small and we may be unable to check normality based on responses in each group. Instead, we first fit the one-factor model, obtain model residuals using (4.3), and then construct a single normal Q-Q using all N residuals.

Checking equal variances. We can construct side-by-side boxplots of the responses grouped by the levels of Factor A. If the boxes (including the whiskers) have approximately the same size, we may conclude that the a groups have equal spreads. The **spread-versus-level plot** is another graphical tool that is useful when there are a moderate to large number of groups. Assume that the underlying distributions of the response in the a groups have similar shapes, but they have different levels and spreads. A spread-versus level plot can help us detect whether the spreads for the a groups (represented by the logarithms of the interquartile range $\log(\text{IQR})_i$) vary with their levels (represented by the logarithms of their medians $\log(\text{M}_i)$), through a plot of $\log(\text{IQR})_i$ versus $\log(\text{M}_i)$ for $i = 1, \ldots, a$. If the slope of this line is zero, the spreads do not change with levels. A positive (or negative) slope indicates that the spreads increase (or decrease) with the levels.

Bartlett's test. We can also use a significance test called Bartlett's test for checking equality of variances. We set

$$\begin{aligned} H_0 &: \sigma_i^2 = \sigma^2,\ i = 1, \ldots, a, \text{ versus} \\ H_1 &: \text{ all not } \sigma_i^2 \text{ are equal.} \end{aligned} \tag{4.31}$$

Let

$$S_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (Y_{ij} - \overline{Y}_{i\cdot})^2,\ i = 1, \ldots, a, \tag{4.32}$$

with d.f. $= \nu_i = n_i - 1$. Compute

$$\overline{S}^2 = \frac{\sum_{i=1}^a \nu_i S_i^2}{\sum_{i=1}^a \nu_i}.$$

Define

$$\begin{aligned} M &= \sum_{i=1}^a \nu_i \log \overline{S}^2 - \sum_{i=1}^a \nu_i \log S_i^2, \text{ and} \\ C &= 1 + \frac{1}{3(a-1)} \left[\sum_{i=1}^a \frac{1}{\nu_i} - \frac{1}{\sum_{i=1}^a \nu_i} \right]. \end{aligned}$$

Under H_0, the quantity $\frac{M}{C}$ is approximately distributed as a χ^2_{a-1} variable. Note that the chi-square approximation is less satisfactory if most of the associated degrees of freedom ν_i are less than 5. Bartlett's test is also very sensitive to the normality assumption, and would reject H_0 too often if observations come from a heavy-tailed distribution.

Levene's test. This is an approximate test for the equality of variances from a populations,

which is somewhat less sensitive to departures from normality. This is because this test uses the average of the absolute deviations instead of mean square deviations as a measure of variation within a group. As a consequence of the absence of "squaring"' terms, this test is less sensitive to heavy-tailed distributions. We define the quantity

$$W_i^2 = \frac{1}{n_i} \left| Y_{ij} - \overline{Y}_{i\cdot} \right| \tag{4.33}$$

instead of the S_i^2 defined in (4.32). We then compute

$$\overline{W}^2 = \frac{1}{a} \sum_{i=1}^{a} W_i^2$$

in the balanced model and

$$\overline{W}^2 = \frac{\sum_{i=1}^{a} \nu_i W_i^2}{\sum_{i=1}^{a} \nu_i}$$

in the unbalanced case. The statistic for Levene's test is then derived in a manner similar to Bartlett's test.

Nonparametric Kruskal-Wallis procedure

In situations where the normality assumption is not justified, the Kruskal-Wallis procedure (Kruskal and Wallis (1952)) is used as an alternative to the one-factor ANOVA F-test. This is an extension of the Wilcoxon rank-sum test to more than two populations. The Kruskal-Wallis statistic tests the null hypothesis that a treatment effects have identical distributions against the alternative hypothesis that not all the a distributions are the same.

We first rank the observations $Y_{ij},\ j = 1, \ldots, n_i,\ i = 1, \ldots, a$ in ascending order. We then replace each Y_{ij} by its rank, say R_{ij}.The smallest observation will have a rank of one. In case of ties, we assign the average rank to each of the tied observations. Let $R_{i\cdot}$ denote the sum of the ranks under the ith treatment. The Kruskal-Wallis test statistic is

$$KW = \frac{1}{S^2} \left[\sum_{i=1}^{a} \frac{R_{i\cdot}^2}{n_i} - \frac{N(N+1)^2}{4} \right], \tag{4.34}$$

where

$$S^2 = \frac{1}{N-1} \left[\sum_{i=1}^{a} \sum_{j=1}^{n_i} R_{ij}^2 - \frac{N(N+1)^2}{4} \right]. \tag{4.35}$$

is the variance of the ranks R_{ij}. If there are no ties, it is easily shown that $S^2 = N(N+1)/12$. In this case, the test statistic simplifies as

$$KW = \frac{12}{N(N+1)} \sum_{i=1}^{a} \frac{R_{i\cdot}^2}{n_i} - 3(N+1). \tag{4.36}$$

If the n_i are reasonably large, i.e., $n_i \geq 5$, say, then under H_0, KW has an approximate chi-square distribution with $a-1$ d.f. Hence, if $KW > \chi^2_{a-1,\alpha}$, we reject the null hypothesis at

level of significance α. Conover and Iman (1981) showed that the usual F-statistic computed for the ranks, R_{ij} (instead of the Y_{ij}) is

$$F_R = \frac{(N-a)KW}{(a-1)(N-1-KW)}, \tag{4.37}$$

and its behavior parallels that of KW, so that the Kruskal-Wallis test is equivalent to the usual F-test applied to the ranks of the data.

Statistical Details - Two-factor ANOVA model

Suppose an experiment or observational study has two factors, Factor A at a levels, and Factor B at b levels. Assume that we can observe the response Y at all possible combinations of the a levels of Factor A and b levels of Factor B, denoted by Y_{ij} in the (i,j)th combination, for $i = 1, \ldots, a$ and $j = 1, \ldots, b$. The data structure for a balanced two factor model with n replicates corresponding to levels of Factor A and b levels of Factor B is shown in Table 4.2.

In some experiments, we may find that the difference in response between the levels of Factor A is not the same at all levels of Factor B (or vice versa). In this case, we say there is an *interaction* between the two factors, and an adequate model must represent this feature. Assuming an interaction between the factor levels, we model the response Y as follows (see (4.8)):

$$Y_{ij\ell} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij} + \varepsilon_{ij\ell},\ \ell = 1, \ldots, n,\ i = 1, \ldots, a,\ j = 1, \ldots, b. \tag{4.38}$$

Similar to the one-factor model, μ is the overall mean effect, τ_i is the effect due to the ith level of Factor A, and $\varepsilon_{ij\ell}$ are i.i.d. $N(0, \sigma^2)$ variables. Additionally, here we have β_j which denotes the effect due to the jth level of Factor B, and $(\tau\beta)_{ij}$ which is the effect due to the interaction between the ith level of Factor A and the jth level of Factor B. The total number of model parameters is $p = 1 + a + b + ab$ and the total number of observations is $N = nab$. We can write the mean response in the (i,j)th cell as $\mu_{ij} = \mu + \tau_i + \beta_j + (\tau\beta)_{ij}$, and interpret the interaction effect $(\tau\beta)_{ij} = \mu_{ij} - \mu - \tau_i - \beta_j$ as the left-over effect from additivity, where we would assume that $\mu_{ij} = \mu + \tau_i + \beta_j$ (as discussed below). ables to factors. For simplicity, we have described the statistical methods for a balanced two-factor model. In practice, it is possible that, either by design or due to unforeseen events, (a) some observations in some cells become unavailable, or (b) there are zero observations in some cells, leading to an unbalanced data structure with $n_{ij} \geq 0$ observations in the (i,j)th cell. Statistical software routinely handle modeling with unbalanced data structures. For theory, we refer the reader to Kutner et al. (2005).

Least Squares Approach

Let $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_a)'$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_b)'$, and $(\boldsymbol{\tau\beta}) = ((\tau\beta)_{11}, \ldots, (\tau\beta)_{ab})'$. Under the least squares (LS) approach, minimizing the error sum of squares

$$S(\mu, \boldsymbol{\tau}, \boldsymbol{\beta}, (\boldsymbol{\tau\beta})) = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{\ell=1}^{n}(Y_{ij\ell} - \mu - \tau_i - \beta_j - (\tau\beta)_{ij})^2$$

with respect to the model parameters leads to a set of $p = 1+a+b+ab$ normal equations of which only the last ab equations are linearly independent, and the remaining equations can

be derived from these, by adding over i, or over j, or both. By imposing the required constraints on the normal equations (similar to the one-factor model), we obtain the following LS solutions:

$$\begin{aligned}
\mu^0 &= \overline{Y}_{...} \\
\tau_i^0 &= \overline{Y}_{i..} - \overline{Y}_{...},\ i = 1, \ldots, a \\
\beta_j^0 &= \overline{Y}_{.j.} - \overline{Y}_{...},\ j = 1, \ldots, b, \text{ and} \\
(\tau\beta)_{ij}^0 &= \overline{Y}_{ij.} - \overline{Y}_{i..} - \overline{Y}_{.j.} + \overline{Y}_{...},\ i = 1, \ldots, a, j = 1, \ldots, b; \qquad (4.39)
\end{aligned}$$

see Kutner et al. (2005) for details. The fitted values and residuals from the model (4.8) can be obtained by using the above formulas as

$$\begin{aligned}
\widehat{Y}_{ij\ell} &= \mu^0 + \tau_i^0 + \beta_j^0 + (\tau\beta)_{ij}^0, \text{and} \qquad (4.40) \\
e_{ij\ell} &= Y_{ij\ell} - \widehat{Y}_{ij\ell}. \qquad (4.41)
\end{aligned}$$

ANOVA Decomposition and F-Tests

We partition the total variability in Y as

$$\text{SST} = \text{SSA} + \text{SSB} + \text{SSAB} + \text{SSE}, \qquad (4.42)$$

where

$$\begin{aligned}
SST &= \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{\ell=1}^{n}(Y_{ij\ell} - \overline{Y}_{...})^2 \\
SSA &= bn\sum_{i=1}^{a}(\overline{Y}_{i..} - \overline{Y}_{...})^2 \\
SSB &= an\sum_{j=1}^{b}(\overline{Y}_{.j.} - \overline{Y}_{...})^2 \\
SSAB &= n\sum_{i=1}^{a}\sum_{j=1}^{b}(\overline{Y}_{ij.} - \overline{Y}_{i..} - \overline{Y}_{.j.} + \overline{Y}_{...})^2 \\
SSE &= \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{\ell=1}^{n}(Y_{ij\ell} - \overline{Y}_{ij.})^2 \qquad (4.43)
\end{aligned}$$

The ANOVA table was shown in Table 4.4, showing the formulas for the mean squares MSA, MSB, MSAB, and MSE, as well as the F-statistics. The sampling distributions of the test statistics if the null hypotheses were true were shown in Table 4.5. Some details are shown below.

Remark 1. We first test for no interaction effects H_{AB}. If the F-statistic $F_{AB} > F_{(a-1)(b-1),ab(n-1),\alpha}$, we reject H_{AB} at the $100\alpha\%$ level of significance, and conclude that there is a significant interaction effect between some level(s) Factor A and some level(s) of Factor B, which contribute to μ_{ij}, and not the corresponding *additive model*.

Remark 2. Following the rejection decision from testing H_{AB}, we next test H_B. It is important to note that this *does not* imply testing $\beta_1 = \ldots = \beta_b$; we are testing whether the effects due to the levels of Factor B are equal, *averaged over all levels of Factor A*.

Remark 3. Similarly, H_A tests whether effects due to the levels of Factor A are equal, averaged over all levels of Factor B.

Remark 4. The F-statistics discussed above rely on the assumptions of normality, independence, and homoscedasticity in the linear model. Checking model assumptions is similar to what we discussed for the one-factor model; since the number of observations in each cell can be quite small, we recommend checking the assumptions on the N residuals we get from the model fit.

Two-factor Additive Model

For completeness, we describe the simpler two-factor additive model which assumes that there is no interaction between the levels of Factor A and the levels of Factor B in (4.8), and expresses $Y_{ij\ell}$ as the sum of an overall mean parameter, μ, an effect due to the ith level of Factor A, τ_i, an effect due to the jth level of Factor B, β_j and a random error $\varepsilon_{ij\ell}$. Here, the mean response in the (i,j)th cell is $\mu_{ij} = \mu + \tau_i + \beta_j$. The model is shown in (4.9). It is easy to show that the LS solutions under the additive model are

$$\begin{aligned} \mu^0 &= \overline{Y}_{...}, \\ \tau_i^0 &= \overline{Y}_{i..} - \overline{Y}_{...},\ i = 1, \ldots, a; \\ \beta_j^0 &= \overline{Y}_{.j.} - \overline{Y}_{...},\ j = 1, \ldots, b. \end{aligned} \tag{4.44}$$

That is, the estimate of the overall mean μ is the overall average of all N observations, while the estimate of the ith group effect τ_i is the difference between the average of the sample observations under the ith level of Factor A across all levels of Factor B and the overall average, and the estimate of the jth group effect β_j is the difference between the average of the sample observations under the jth level of Factor B across all levels of Factor A and the overall average. The LS estimate of the error variance σ^2 is

$$\widehat{\sigma}^2 = \frac{SSE}{N - a - b + 1}, \tag{4.45}$$

where the error sum of squares (or residual sum of squares) is

$$SSE = \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{\ell=1}^{n_{ij}} (Y_{ij\ell} - \overline{Y}_{i..} - \overline{Y}_{.j.})^2. \tag{4.46}$$

Again, the fitted values and residuals from the additive model are

$$\begin{aligned} \widehat{Y}_{ij\ell} &= \mu^0 + \tau_i^0 + \beta_j^0, \text{and}, \\ e_{ij\ell} &= Y_{ij\ell} - \widehat{Y}_{ij\ell}. \end{aligned}$$

The ANOVA decomposition reduces to partitioning the total variability in Y into variation due to levels of Factor A and Factor B and due to error (residual) variation (no interaction induced variability):

$$\text{SST} = \text{SSA} + \text{SSB} + \text{SSE}. \tag{4.47}$$

The details are similar to what we showed above and are not repeated here; see Kutner et al. (2005) for more details.

5

Linear Regression Analysis

This chapter describes the setup and analysis of the widely used linear regression model. The multiple linear regression (MLR) model explains the relationship between a response variable and one or more explanatory variables via a linear functional form that involves some unknown parameters called partial regression coefficients. The method of least squares enables us to estimate these coefficients optimally. We discuss inference on these coefficients to assess goodness of model fit, as well as graphical and quantitative methods to assess whether the model assumptions are not violated. We then show how the fitted MLR model can be used for making predictions of the response at new predictor values. A special case of the MLR model corresponds to a situation with a single predictor, leading to the simple linear regression (SLR) model.

5.1 Introduction

Regression analysis is a popular statistical tool which helps to study meaningful relationships, if any, between two or more variables. Sir Francis Galton Galton (1886) first introduced the term 'regression' in a study of the nature of offspring to tend not towards the height of the parents, but rather to regress to some average. A regression model is a mechanism that enables us to describe the relationship between a response variable and one or more explanatory variables via a functional form that involves some unknown parameters called the regression coefficients. A model formulation enables us to conceptualize how the responses are generated under some probabilistic mechanism. If the functional form is linear in the regression parameters, we call it a linear regression model. This technique is now widely used to build empirical models in many disciplines in the physical and social sciences including but not limited to business, economics, education, engineering, psychology, etc.

A simple linear regression (SLR) model studies the relationship between a response variable and a *single* predictor variable. This model generalizes to a multiple linear regression (MLR) model which studies the relationship between a response variable and $p > 1$ predictor variables. We will focus on questions such as when to use a linear regression model, how to fit and interpret the fitted model, as well as assess adequacy of fit. We will introduce the notion of cross-validation.

5.2 Essential Terminology and Notation

✓ The **linear regression model** is a statistical model which explains the mean of a response variable Y as a linear function of $p \geq 1$ predictors $X_1, \ldots, X_p$.

✓ When $p = 1$, we have a **simple linear regression (SLR)** model which formulates a straight line relationship between Y and a single predictor variable X based on a random sample of pairs (Y_i, X_i), $i = 1, \ldots, n$:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i. \tag{5.1}$$

Here, β_0 is the unknown intercept and β_1 is unknown the slope of the straight line, while $\epsilon_i = Y_i - (\beta_0 + \beta_1 X_i)$ is the random error which is assumed to have a $N(0, \sigma^2)$ distribution.

✓ If X and Y are continuous-valued, the **sample correlation coefficient** is used to measure of the strength of their *linear relationship*:

$$r_{xy} = \frac{S_{xy}}{\sqrt{S_{yy} S_{xx}}}, \tag{5.2}$$

where, the sample means, sample variances and sample covariance are

$$\begin{aligned} S_{xx} &= \sum_{i=1}^{n} (X_i - \overline{X})^2 / (n-1) \ , \ S_{yy} = \sum_{i=1}^{n} (Y_i - \overline{Y})^2 / (n-1) \ , \\ S_{xy} &= \sum_{i=1}^{n} (X_i - \overline{X})(Y_i - \overline{Y}) / (n-1) \ , \\ \overline{X} &= \sum_{i=1}^{n} X_i / n, \ \overline{Y} = \sum_{i=1}^{n} Y_i / n. \end{aligned} \tag{5.3}$$

✓ The **correlation matrix** between Y, $X_1, \ldots, X_p$ is a $(p+1) \times (p+1)$ symmetric matrix whose off-diagonal elements contain pairwise correlations between the variables. It is obtained using the cor() in R.

✓ When $p > 1$, and data $(Y_i, X_{i1}, X_{i2}, \ldots, X_{ip})$, $i = 1, \ldots, n$, we can fit a **multiple linear regression (MLR)** model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{ip} + \epsilon_i, \tag{5.4}$$

where X_{ij} is the ith observation on the jth predictor variable and β_j is the jth partial regression coefficient, for $j = 1, \ldots, p$. The phrase 'multiple linear' implies that the partial regression coefficients β_j enter linearly into the specification on the right side. The jth coefficient measures the expected change in Y per unit increase in X_j while the remaining variables $X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p$ are held constant. The random errors in the model are represented by $\epsilon_i, \ i = 1, \ldots, n$. We can also express the model in (5.4) as

$$E(Y_i) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip}, \tag{5.5}$$

stating that the mean response on the ith case, $E(Y_i)$ is explained by the linear function $\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip}$. In the SLR model, this can be represented as

$$E(Y_i) = \beta_0 + \beta_1 X_i. \tag{5.6}$$

The MLR model can be expressed in vector-matrix form by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{5.7}$$

where $\mathbf{y} = (Y_1, Y_2, \ldots, Y_n)'$ is an n-dimensional response vector, $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$ is a $(p+1)$-dimensional vector of partial regression coefficients, $\mathbf{X} = \{X_{ij}\}$ is an $n \times (p+1)$ matrix of predictors whose first column is an n-dimensional vector of 1's and $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)'$ is an n-dimensional error vector. We assume $n > p$.

✓ We make the following **assumptions:** linearity, equal variances (homoscedasticity), independence, and normality. Linearity implies that the right side of (5.4) involves only linear terms in $\beta_0, \beta_1, \ldots, \beta_p$. The equal variance assumption implies that $\text{Var}(\epsilon_i) = \sigma^2$, a constant for all i, implying that $\text{Var}(Y_i) = \sigma^2$ as well. The independence and normality assumptions imply that the errors ϵ_i are i.i.d. $N(0, \sigma^2)$ variables, leading to Y_i being independently (but not identically) distributed as $N(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip}, \sigma^2)$ variables. In addition, we will require that the matrix $\mathbf{X}$ has full column rank, i.e., $\text{rank}(\mathbf{X}) = p + 1$.

✓ The **method of least squares** is a standard approach in regression analysis to approximate the solution of sets of equations by minimizing the sum of the squares of the errors (an error is the difference between an observed value and the fit under a true model).

✓ The **coefficient of determination** (also called the *multiple correlation coefficient*), is a goodness of fit measure in the MLR model and is defined as

$$R^2 = \frac{SSR}{SST} = \frac{SST - SSE}{SST}, \tag{5.8}$$

where SSR and SST are respectively the model sum of squares and the corrected total sum of squares. This measure represents the proportion of variability in the response that is explained by the fitted model. It is often expressed as a percentage, $100R^2\%$. Similar to the pairwise correlations between Y and each X_j, R^2 is also used to measure effect size in a regression model.

✓ **Inference on partial regression coefficients** consists of confidence interval estimation of β_j and hypothesis testing consisting of $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$, for $j = 0, \ldots, p$. The intervals are $100(1-\alpha)\%$ t-intervals, while the tests are t-tests. If the t-interval for β_j does not include zero, or equivalently, the p-value of the t-test statistic for H_0 exceeds α, we conclude that X_j has a significant effect (in a linear model) on the response Y.

✓ The **F-test** in an MLR model tests the hypothesis $H_0 : \beta_1 = 0, \beta_2 = 0, \ldots, \beta_p = 0$ versus H_1 : at least one of these β_j is non-zero. Let MSR and MSE respectively denote the model mean squares and the error (residual) mean squares. A large value of the F-statistic defined by F-stat = MSR/MSE points to an adequate model where at least one of the p predictors is useful in explaining the mean response, while a small value indicates an inadequate model.

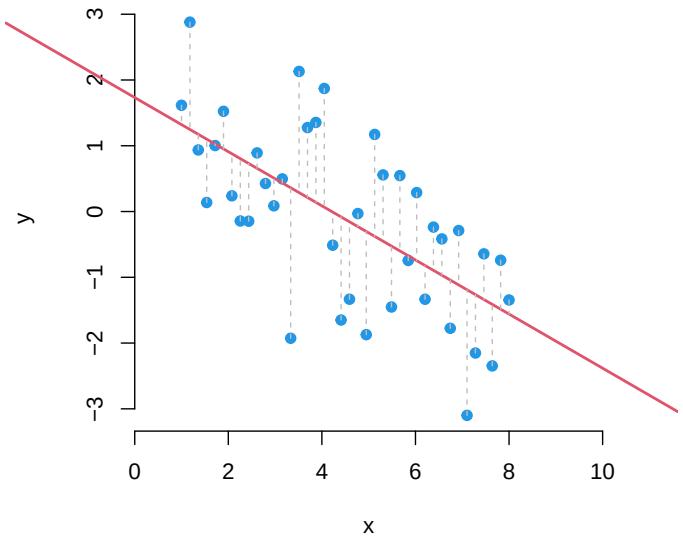

FIGURE 5.1: A graphical example of simple linear regression.

✓ **Out-of-sample prediction** consists of entering new values for the independent variables into the fitted MLR model equation in order to predict the response variable. As long as the new values of the predictors, say $\mathbf{x}_0 = (X_{1,0}, \ldots, X_{p,0})'$ are not too far away from what we observed in the training sample, the predictions will be reasonable. We must however be cautious about using the fitted regression model to predict an unknown Y at a values of any predictor X that may be an outlier in the X-space.

Figure 5.1 illustrates the ideas and concepts graphically. To simplify things, we use the simple linear regression case, with one predictor, X, which is linearly related to the outcome variable, Y. The observed data is shown as blue dots, and the fitted regression line appears in red. In this case the slope can be seen to be negative (as X increases, Y decreases.) The intercept is the point where the regression line (in red) intersects the y-axis, and it represents the value we expect to observe when all the predictors (in this case, just one) are set to zero. The grey dashed lines depict the residuals – the differences between the observed values, and the ones predicted by the regression formula. These residuals can be regarded as estimates of the errors in the MLR model; if the model fit is adequate, we expect that they will be normally distributed with zero mean and a variance σ^2 which does not depend on any of the predictors. These are the values which are used in the calculation of R^2, for example. In fact, the regression line is found by minimizing the sum of the squares of the residuals.

5.3 Multiple Linear Regression

The multiple linear regression (MLR) model is useful for explaining a numerical response variable as a *linear* function of predictors. The form of the multiple linear regression (MLR) model was shown in (5.4); the response variable is Y, while $X_1, \ldots, X_p$ are the predictors. As we will see below, these predictors can either be original variables, or new variables created as products of the original variables, or new variables created by transforming the original variables.

We use the cycling data which was introduced in Chapter 4 to explain the ideas.

Cycling

Example 5.3.1 (Cycling to school) We are interested in knowing whether girls' enrollment in the secondary school in Bihar has increased as a result of the program which provided bicycles. The response variable is `enrollment`. The code for implementing the MLR model is very similar to the one used in the previous chapter, except that this time, in addition to gender, we treat `year` as a continuous predictor, rather than a categorical one:

```
lenrolldat$n_year <- lenrolldat$year - 2002
fittedlm <- lm(lenrollment ~ female*n_year, data = dat_table1)
confint(fittedlm)
```

Using the lm() function gives the following output:

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.234536   0.021794 194.298  < 2e-16 ***
female1       -0.870071   0.030223 -28.789  < 2e-16 ***
n_year         0.085223   0.007945  10.726  < 2e-16 ***
female1:n_year 0.051822   0.011012   4.706 2.54e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8745 on 20262 degrees of freedom
Multiple R-squared:  0.1669,        Adjusted R-squared:  0.1667
F-statistic:  1353 on 3 and 20262 DF,  p-value: < 2.2e-16
```

The estimates for `Intercept`, `female1`, `n_year` and the interaction `female1:n_year` shown in the output above are the least squares estimates of the partial regression coefficients $\beta_0, \beta_1, \beta_2$, and β_3.

Given n observations (cases) and p predictors, the ordinary least squares (OLS) estimates of the partial regression coefficients β_j's in the MLR model (5.4) (or, (5.7) in vector-matrix form) are obtained by minimizing the sum of squares of the errors expressed as a function of the β_j's, i.e.,

$$S(\beta_0, \beta_1, \ldots, \beta_p) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2. \tag{5.9}$$

The estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ that are obtained by the method of least squares can also be obtained using differential calculus by setting to zero the partial derivatives of $S(\beta_0, \beta_1, \ldots, \beta_p)$

with respect to $\beta_j,\ j = 0, 1, \ldots, p$, and solving the resulting set of $(p+1)$ *normal equations*. We can denote them as a vector $\hat{\boldsymbol{\beta}}$. The output also gives the standard errors of the estimates. The appendix gives details on the formulas.

The F-statistic has the value 1353 on 3 and 20262 d.f., with a p-value $< 2.2e-16$. Similar to the F-statistic in Chapter 4, this statistic is useful to assess the usefulness of all the predictors taken together, for explaining the response variable in an MLR model. Specifically, the F-statistic tests

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0 \text{ versus}$$
$$H_1 : \beta_j \neq 0 \text{ for at least one } j \in \{1, 2, \ldots, p\}.$$

Note that H_1 does not state that *all* the β_j must be non-zero. It just says that at least one of the β_j is non-zero, or in other words, that not all the β_j are zero. If H_0 is true, the F-statistic will have an F-distribution with p numerator d.f. and $n-p-1$ denominator d.f. In this output, the data provides sufficient evidence to reject H_0 and claim that `female1`, `n_year` and `female1:n_year` together explain textttenrollment.

Note that the coefficient of determination from the fitted MLR model is $R^2 = 0.1669$ (see (5.8)). This tells us that, while significant based on the F-statistic, the fitted model explains only 17% of the variability in the response.

We can also test each coefficient separately for each j, i.e., test

$$H_0 : \beta_j = 0 \text{ versus } H_1 : \beta_j \neq 0$$

via a t-test. The test statistic is

$$t_0 = \hat{\beta}_j / se(\hat{\beta}_j) = \hat{\beta}_j / \hat{\sigma}\sqrt{C_{jj}}, \tag{5.10}$$

where C_{jj} is the jth diagonal element of $(\mathbf{X}'\mathbf{X})^{-1}$. The statistic t_0 has a Student t-distribution with $n-p-1$ degrees of freedom under H_0. We see that the interaction term `female1:n_year` is positive (0.051822) and significant (with a p-value from the t-test of 2.54e-06), so the rate of change in enrollment is different among girls and boys. The 'reference' group here is boys (`female`=0), and for them, the log-enrollment increased by 0.085 every year. For females, the enrollment decreased by -0.87+0.0518=-0.8182 every year. The overall enrollment of girls still goes down, but the research question is whether that *rate* changes over time (presumably, as a result of the bicycle program), or whether it stays the same. Inference regarding this question is done via the interaction term, and we see a positive effect. This means that although the log-enrollment for girls is still less than that for boys, the gap is closing by 0.052 every year. The intercept (4.234536) is the expected log-enrollment for boys (the reference group) at time 0 (the year 2002, since we centered the data).

We use the function confint() to obtain the confidence interval estimates of β_j:

```
confint(fittedlm)
```

The output is shown below:

```
                    2.5 %      97.5 %
(Intercept)     4.19181802  4.27725400
female1        -0.92931021 -0.81083207
n_year          0.06964958  0.10079625
female1:n_year  0.03023802  0.07340513
```

These confidence intervals are t-intervals. Specifically, in the MLR model (5.4), the $100(1-\alpha)\%$ confidence interval for $\beta_j,\ j = 0, 1, \ldots, p$ is

$$\hat{\beta}_j \pm t_{\alpha/2,n-k-1}\ \mathrm{SE}(\hat{\beta}_j). \tag{5.11}$$

As we have seen earlier, if a C.I. includes the value 0, we conclude that the data supports the null hypothesis $H_0 : \beta_j = 0$ and not the alternative hypothesis $H_1 : \beta_j \neq 0$.

Let $\hat{Y}_i$ denote the ith in-sample fitted value given by replacing each β_j by its OLS estimate $\hat{\beta}_j$:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \ldots + \hat{\beta}_p X_{ip}, \tag{5.12}$$

and let e_i be the ith OLS residual

$$e_i = Y_i - \hat{Y}_i. \tag{5.13}$$

Table 5.1 shows a few modifications of these OLS residuals e_i that use other quantities from the MLR fit, such as

(i) h_{ii}, the ith diagonal element of the hat matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ (in some texts, the hat matrix is called the projection matrix and is denoted by $\mathbf{P}$);

(ii) $\hat{\sigma} = \sqrt{\sum_{i=1}^{n} e_i^2/(n-p-1)}$, the OLS estimate of the error standard deviation σ; and

(iii) $\hat{\sigma}_{(i)}$, the estimate of σ using the same formula as in (ii), but after fitting the model to $n-1$ observations dropping the ith.

In the language of Chapter 4, $\hat{\sigma} = \sqrt{\mathrm{MSE}}$, where $\mathrm{MSE} = \mathrm{SSE}/(n-p-1)$.

TABLE 5.1: Types of residuals

Type of Residual	Formula
Normalized residual	$a_i = \frac{e_i}{\sqrt{\sum_{i=1}^{n} e_i^2}}$
Standardized residual	$b_i = \frac{e_i}{\hat{\sigma}}$
Internally Studentized residual	$r_i = \frac{e_i}{\hat{\sigma}(1-h_{ii})^{1/2}}$
Externally Studentized residual	$r_i^* = \frac{e_i}{\hat{\sigma}_{(i)}(1-h_{ii})^{1/2}}$

To check model adequacy, we plot the externally Studentized residuals r_i^* versus the fitted values $\hat{Y}_i$. If the model is adequate, the residuals should be spread equally around 0, with no discernable pattern; for example, if the residuals increase (or decrease) as the predicted values increase, then we have evidence that the errors are not independent of the prediction, and thus are probably not independent from all the predictors. This would violate an assumption of the linear model. The externally Studentized residuals are preferred over the other types shown in Table 5.1 because it is least affected by (i.e., is most resistant to) outlying observations.

Figure 5.2 is obtained by using the following code. Notice that because the year variable contains only integers, we get only eight possible predicted values (for the combinations of four years and two groups); therefore, to improve the plot, (i) we use the jitter() function, and (ii) we use different colors for the two groups (blue for the girls and green for the boys).

The plot shows that the model appears to be reasonable. However, the magnitude of the residuals may be higher than we would like – we expect 95% of the Studentized residuals to be between -2 and 2, but we see that some residuals are outside this range. We saw

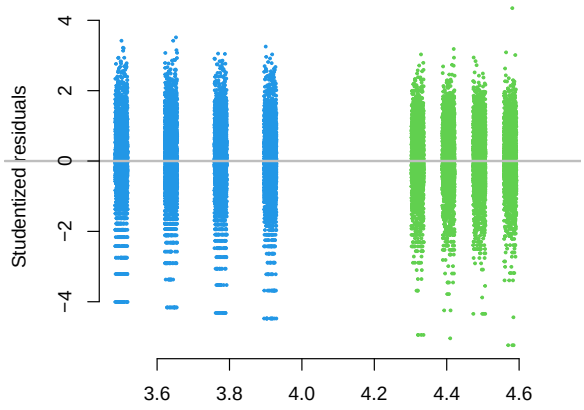

FIGURE 5.2: The Studentized residual vs. predicted values plot for the bicycle data.

previously that the R^2 value was only 0.17, which also suggests a less-than-desired model-fit. This could be because other important variables may be necessary in order to explain the log-enrollment.

```
plot(jitter(fittedlm$fitted.values), rstudent(fittedlm), pch=19,
     col=ifelse(dat_table1$female == 0, 3, 4),cex=0.3, xlab="",
     ylab="Studentized residuals", axes=F)
abline(h=0,col="grey", lwd=2)
axis(1);axis(2)
```

While the in-sample fitted values are obtained using (5.12), we can also use the fitted model to *predict* out-of-sample enrollment of boys and girls. For example, we can try to predict the outcome in years 5-14 by using the predict() function, as shown in the code below.

```
newdat <- data.frame(female=as.factor(c(rep(1,10), rep(0,10))),
                     n_year=rep(c(5:14),2))
plot(newdat$n_year, predict(fittedlm, newdata = newdat),
     col=ifelse(newdat$female == 0, 3, 4),
     pch=ifelse(newdat$female == 0, 18, 17), xlab="years",
     ylab="log-enrollment", axes=F, ylim=c(3.5,5.5))
axis(1); axis(2)
```

It is easy to obtain the predicted response from the fitted MLR model at a new set of predictors $\mathbf{x}_0 = (1, X_{1,0}, \ldots, X_{p,0})$ as

$$\hat{Y}_0 = \mathbf{x}_0'\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \hat{\beta}_1 X_{1,0} + \ldots + \hat{\beta}_p X_{p,0}. \tag{5.14}$$

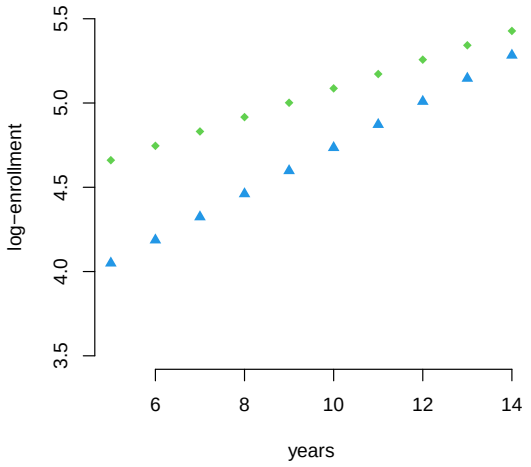

FIGURE 5.3: Predicted log-enrollment values for boys (green diamonds) and girls (blue triangles).

The SE of prediction is obtained by taking the square root of

$$\text{Var}(\hat{Y}_0) = \sigma^2 \mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0, \tag{5.15}$$

from which we can construct the $100(1-\alpha)\%$ t-interval on $n-p-1$ d.f. for the unknown response Y_0.

The resulting Figure 5.3 shows that the enrollment of girls is lower than that of boys (the residual plot also shows quite effectively the big difference between boys and girls), but the gap between enrollments of girls and boys is expected to decrease. We must be very careful with such statements, though; we extrapolated the data to years which are all outside the range of what we have in the data! There is no guarantee that the trend we observed during the four years of the study will continue in the future.

5.4 Polynomial Regression

Polynomial regression models are useful when a polynomial function of the predictor variables, rather than a linear function, is a good approximation to the mean response Y. A polynomial regression model in a single predictor X, which is also called curvilinear regression of Y on X can be written as

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \ldots + \beta_q X_i^q + \epsilon_i, \tag{5.16}$$

where X_i^j is the jth power of X_i for $j = 1, \ldots, q$, β_j is the corresponding jth partial regression coefficient, for $j = 1, \ldots, q$, and β_0 is the intercept. Note that the q predictors in (5.16) are constructed as polynomial functions of a single variable X. Now, suppose we have an additional predictor X_2 which has a linear effect on Y. We can then write the MLR

model as

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \ldots + \beta_q X_i^q + \beta_{q+1} X_2 + \epsilon_i, \tag{5.17}$$

with $p = q + 1$. Since the polynomial regression model is an example of the MLR model shown in (5.4), we can use the methods discussed in the previous sections, as shown in the following example.

Ozone

Example 5.4.1 (Ozone) High ground-level ozone (O_3) concentrations levels are hazardous, and are therefore monitored regularly by authorities, such as the United States Environmental Protection Agency (EPA). The O3.RData file contains ozone measurements from 2019, collected at six monitoring stations: two in London, two in Washington D.C., and two in New York. In the analysis we perform here, we use only the US data. In addition to the location, we also have the date for each measurement, which we convert to a variable called `doy` (day of year). Several other factors may contribute to ground-level ozone concentration, but here we focus on the location and the day of year, in order to demonstrate polynomial regression. The units of the response is ppm = part per million, and is denoted by `Value`.

Research question #2

Is ground-level ozone concentration related to the time of year? If so, how?

We fit two models to predict the mean ozone concentration level:

1. Linear: E(`Value`) = `City` + `doy`
2. Quadratic: E(`Value`) = `City` + `doy` + `doy`^{2}

Notice that since there are two cities, the output resulting from the code below will contain an estimate only for one of the cities, and the estimated mean ozone level in the reference city is the intercept.

```
load("Data/O3.RData")
OzoneAll <- OzoneAll[which(OzoneAll$City != "London"),]
# linear model:
linearfit <- lm(Value ~ City + doy, data=OzoneAll)
summary(linearfit)
# quadratic model, with squared doy
quadraticfit <- lm(Value ~ City + doy + I(doy^2), data=OzoneAll)
summary(quadraticfit)
```

The output from the Linear model appears below. Notice that there is a small difference between the cities (Washington D.C. has a higher mean concentration levels of ozone), and the `doy` is not significant. Overall, the model provides a very poor fit, with $R^2 = 0.0036$, so less than 0.5% of the variability is explained by the model. With a p-value of 0.07, the model as a whole is not significant at the 5% level of significance.

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.500e-02  8.011e-04  43.692   <2e-16 ***
CityWashington  1.625e-03  7.134e-04   2.278   0.0229 *
doy            -7.241e-07  3.411e-06  -0.212   0.8319
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01353 on 1435 degrees of freedom
Multiple R-squared:  0.003639,        Adjusted R-squared:  0.00225
F-statistic:  2.62 on 2 and 1435 DF,  p-value: 0.07312
```

The output from the Quadratic model appears below, and it can be seen the the F-test is significant ($< 2.2e-16$), and the R^2 is much higher, with 46% of the variability explained by the model. All three predictors are significant.

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.437e-02  8.376e-04  17.157  < 2e-16 ***
CityWashington  1.524e-03  5.259e-04   2.897  0.00382 **
doy             3.376e-04  1.006e-05  33.557  < 2e-16 ***
I(doy^2)       -9.282e-07  2.673e-08 -34.731  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.009971 on 1434 degrees of freedom
Multiple R-squared:  0.4588,        Adjusted R-squared:  0.4577
F-statistic: 405.3 on 3 and 1434 DF,  p-value: < 2.2e-16
```

We can perform a sequential F-test, using the anova() function, like this (see the appendix for details on the sequential and partial F-tests, which are examples of the extra sum of squares F-test):

```
anova(quadraticfit, linearfit)
```

We obtain the following output, which says that the addition of the quadratic terms makes a very significant difference in the goodness of fit, with $p < 2.2e-16$.

```
Analysis of Variance Table

Model 1: Value ~ City + doy + I(doy^2)
Model 2: Value ~ City + doy
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1   1434 0.14257
2   1435 0.26250 -1  -0.11993 1206.3 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

We also plot the Studentized residuals from each model, versus the `doy` variable, using the following code. The results are shown in Figure 5.4. The plot for the Linear model shows a clear quadratic relationship between the predictor and the residuals, while there is no obvious trend in the second model. Recall that the regression framework relies on an assumption that

the residuals are independent from any predictor, so clearly the Linear model violates this assumption. For the Quadratic model, the residuals are spread approximately equally around zero, and most of the externally Studentized residuals fall between -2 and 2, which is in agreement with the assumptions that the Studentized residuals have a $N(0,1)$ distribution.

```
par(mfrow=c(1,2))
plot(OzoneAll$doy, rstudent(linearfit),cex=0.5, pch=ifelse(OzoneAll$City ==
    "Washington", 19, 22), col=ifelse(OzoneAll$City == "Washington", 3, 4),
    xlab="day of year", ylab="Studentized residuals", axes=F,main="Linear
    model", cex.main=1, ylim=c(-3.5,3.5))
axis(1); axis(2); abline(h=0,lwd=3,col="orange")
plot(OzoneAll$doy, rstudent(quadraticfit), pch=ifelse(OzoneAll$City ==
    "Washington", 19, 22),  cex=0.5, col=ifelse(OzoneAll$City ==
    "Washington", 3, 4), xlab="day of year", ylab="Studentized residuals",
    axes=F,main="Quadratic model", cex.main=1, ylim=c(-3.5,3.5))
axis(1); axis(2); abline(h=0,lwd=3,col="orange")
par(mfrow=c(1,1))
```

We can conclude that the Quadratic model is much more appropriate than the Linear model, and that there is a strong quadratic relationship between `doy` and ozone levels:

$$E(\text{Value}) = -9.282e-07\text{doy}^2 + 3.376e-04\text{doy} + 1.437e-02 + 1.524e-03\text{I[Washington]}\,.$$

From this relationship we can compute the day of the year in which we expect the maximum concentration level of ozone:

$$\arg\max(O_3) = \frac{3.376e-04}{2 * 9.282e-07} \approx 182,$$

which is very close to the summer solstice in the northern hemisphere. It is known that ozone levels increase with the amount of day light, and our analysis shows this relationship quite clearly. It should be noted that since the length of day is periodic, so is the function of ozone concentration, and in fact, the well-fitting Quadratic model actually represents a cosinusoidal trend from which we only observe one cycle. (Recall that the second-order Taylor expansion of the cosine function is a quadratic function). There are other factors which are known affect ozone levels but are not included in our dataset. Obtaining the values of such predictors will likely increase the goodness of fit, but even with just the `doy` predictor, we are able to provide reasonable predictions, if we recognize the fact that a polynomial model is more adequate than a linear one.

Note that, after we fit the Linear model $E(O_3) = City + doy$, we can construct two enhanced residual plots which show us whether or not to include the $\texttt{doy}^2$ and run a Quadratic model. Figure 5.5 (right plot) shows the partial residual plot for `doy` using the crPLots() function in the R package *car*. We first fit the Linear model, pull out the residuals which we plot against the predictor `doy`. If polynomial terms are useful for the model, the plot would show curvature, else the points will lie along a straight line. The curvature in the right plot in 5.5 shows that including $\texttt{doy}^2$ in the model will be useful for explaining `Value`. The plot on the left shows parallel boxplots of the partial residuals for the levels of the factor variable `City` (a feature of crPlots).

```
library(car)
crPlots(linearfit)
```

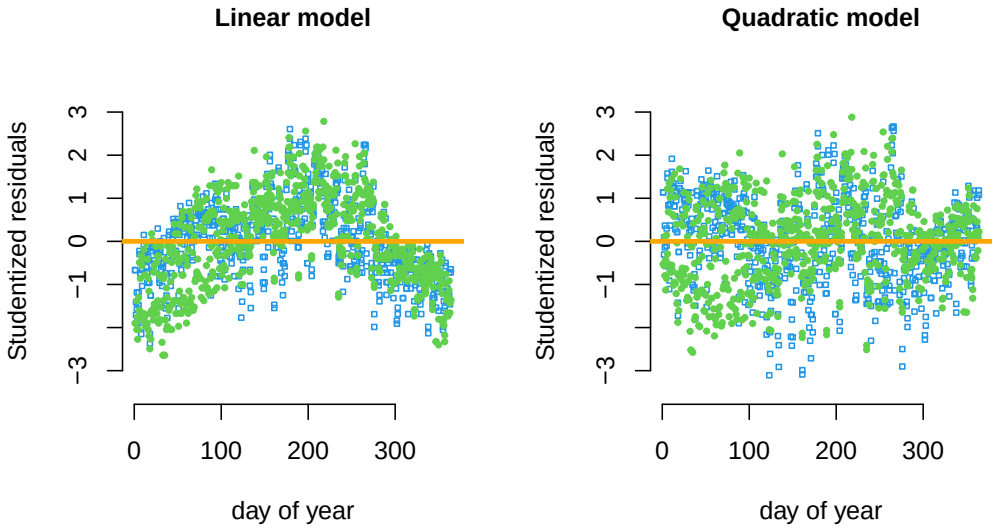

FIGURE 5.4: Studentized residuals vs. day of year. Left: linear model, Right: Quadratic model. Filled, green circles are for Washington, DC, and empty blue squares for New York, NY.

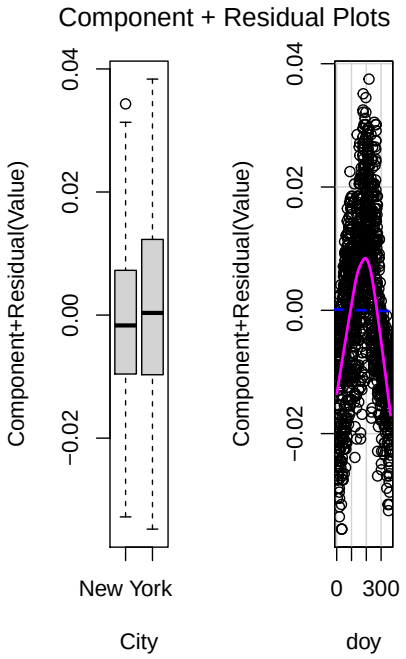

FIGURE 5.5: Partial residual plots from the Linear model for the Ozone data.

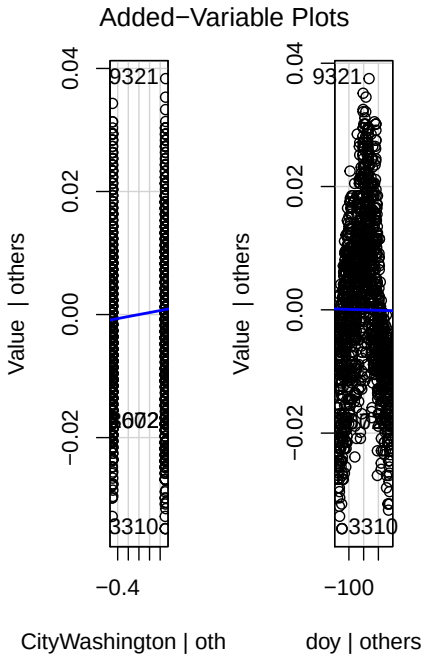

FIGURE 5.6: Added variable (partial regression) plots for the Linear model for the Ozone data.

Figure 5.6 (right plot) shows added variable plots which are also called partial regression plots for the Linear model using the avPlots() function. Again, we fit the Linear model, pull out the residuals. Next, we regress $\texttt{doy}^2$ on `doy` and pull out the residuals from this second regression. The added variable plot is a plot of the first set of residuals against the second set. Since we remove any linear effect between $\texttt{doy}^2$ and `doy`, the added variable plot is considered to be more useful than the partial residual plot. for investigating the usefulness of including $\texttt{doy}^2$. Again, the plot on the left shows parallel boxplots of the partial residuals for the levels of the factor variable `City`.

```
avPlots(linearfit)
```

Together with the significant t-statistic for $\texttt{doy}^2$, these plots confirm that the Quadratic model gives a better fit. Now, we can repeat the analysis by seeing whether a Cubic model, which additionally includes $\texttt{doy}^3$ is useful. For getting the partial residual and partial regression plots, we will replace 'linearfit' by 'quadraticfit' in the code above. The resulting plots (not shown here) show no curvature, indicating that higher order polynomial terms in `doy` may not be useful for explaining `Value`.

Remark 1. We can easily write an MLR model which includes polynomial terms in more than one variable. For example, a second-order model on two variables X_1 and X_2 is

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_{11} X_{i1}^2 + \beta_{22} X_{i2}^2 + \beta_{12} X_{i1} X_{i2} + \epsilon_i, \tag{5.18}$$

where $X_{i1}X_{i2}$ is called an interaction term with coefficient β_{12}. We can fit rich MLR models of this form when we have more predictor variables with polynomial rather than linear effects on the response Y.

5.5 Cross-validation

In order to evaluate the performance of a fitted model on data, it is important to quantify how well the predictions that we obtain from the fitted model match the observed data. While it is useful to make this comparison on in-sample data that we used for fitting the model, it is often useful to do this on out-of-sample data, i.e., on 'test' data that was not used for model fitting.

One way to cross-validate is to randomly split the data set into a training portion and a test portion or the hold-out portion. Usually, we set aside 80% of the data for training the models, the remaining 20% serving as the test portion. We fit models on the training data, we then use the fitted model to predict the response using values of the predictors from the test data. Then, we compute a criterion such as MSE between the observed and predicted cases in the test data.

Another commonly used method for cross-validating is known as k-fold cross-validation, which includes the following steps.

Step 1. Randomly divide the data into k groups (or, "folds"), of roughly equal size. Usually, $k = 5$, or 10.

Step 2. Choose one of the k folds to be the hold-out set. Fit the model on the remaining $k - 1$ folds. Calculate selected criteria, such as MSE, on the observations in the fold that was held out (i.e., not used for model fitting). Call this the "test MSE".

Step 3. Repeat this process k times, using a different set as the holdout set each time.

Step 4. Calculate the overall test MSE as the average of the k test MSE's.

We can use the `trainControl()` function in the *caret* library in R to do k-fold cross-validation. As we mentioned earlier, setting $k = 1$ consists of dividing the data randomly into train and test portions and computing the test MSE once on the hold-out data. We will discuss cross-validation in the following chapters.

Appendix Chapter 5-Statistical Details

Least Squares Estimation and Fit

We fit the MLR model (5.4) by the method of least squares. Given n observations (cases) and p predictors, the ordinary least squares (OLS) estimates of the partial regression coefficients β_j's are obtained by minimizing the sum of squares of the errors expressed as a function of

the β_j's

$$S(\beta_0, \beta_1, \ldots, \beta_p) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2. \tag{5.19}$$

The least squares estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ can be obtained using differential calculus by setting to zero the partial derivatives of $S(\beta_0, \beta_1, \ldots, \beta_p)$ with respect to $\beta_j,\ j = 0, 1, \ldots, p$, and solving the resulting set of $(p+1)$ *normal equations*

$$\frac{\partial S}{\partial \beta_j} = 0,\ j = 0, 1, \ldots, p.$$

We can concisely express the sum of squares of the errors and the OLS estimate of $\boldsymbol{\beta}$ in vector-matrix notation as

$$S(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}), \tag{5.20}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{5.21}$$

respectively. The partial regression coefficient corresponding to each predictor variable indicates the expected change (increase or decrease depending on whether the coefficient is positive or negative) in the response variable for a unit increase in that predictor variable, holding all other predictor variables constant.

While the least squares approach does not require the assumption of normality for constructing these point estimates of the regression coefficients shown in (5.21) or (5.27), we *will* need the normality assumption to develop inference (i.e., tests and confidence intervals) for the parameters.

For $i = 1, \ldots, n$, the fitted values under the MLR model are defined by plugging in the estimated coefficients in the model form and are

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \ldots + \hat{\beta}_p X_{ip}. \tag{5.22}$$

If the model is a good fit, we expect that $\hat{Y}_i$ will be close to Y_i, and that the residuals defined by

$$e_i = Y_i - \hat{Y}_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \ldots + \hat{\beta}_p X_{ip}) \tag{5.23}$$

will be close to zero. Let $\hat{\mathbf{y}} = (\hat{Y}_1, \ldots, \hat{Y}_n)'$ be the n-dimensional vector of fits and let $\mathbf{e} = (e_1, \ldots, e_n)'$ be the corresponding residual vector. We can show that the random vector $\mathbf{e}$ has mean $\mathbf{0}$ and variance $\sigma^2(\mathbf{I} - \mathbf{H})$, where

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' = \{h_{ij}\} \tag{5.24}$$

is called the hat matrix or the projection matrix. The function lm() gives us least squares estimates $\hat{\beta}_j,\ j = 0, \ldots, p$ as well as the fitted values and residuals.

The least squares estimate of the error variance σ^2 is then obtained as the Mean Squared Error,

$$\hat{\sigma}^2 = \text{MSE} = \frac{\text{SSE}}{n - (p+1)} = \frac{\sum_{i=1}^{n} e_i^2}{n - (p+1)}. \tag{5.25}$$

We can recover the standard errors of $\widehat{\beta}_j,\ j = 0, 1, \ldots, p$ from the square roots of the diagonal elements of

$$\text{Cov}(\hat{\boldsymbol{\beta}} = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}, \tag{5.26}$$

while the off-diagonal elements of this $(p+1)\times(p+1)$ matrix shows the covariances between the estimated coefficients.

Special case: SLR model. Consider the **SLR model** (5.1), where the least squares procedure consists of constructing the "best fitting" line through the points (Y_i, X_i), $i = 1, \ldots, n$. The intercept and slope of this line are the OLS estimates of β_0 and β_1, which can be obtained in closed form as

$$\begin{aligned} \widehat{\beta}_1 &= \frac{\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^n (X_i - \overline{X})^2} \text{ and} \\ \widehat{\beta}_0 &= \overline{Y} - \widehat{\beta}_1 \overline{X}. \end{aligned} \tag{5.27}$$

Parameter	Correlation	Intercept	Slope
Assumptions	Normal errors	Normal errors	Normal errors
Estimate	$r_{xy} = \frac{S_{xy}}{\sqrt{S_{yy}S_{xx}}}$	$\hat{\beta}_0 = \overline{Y} - \widehat{\beta}_1 \overline{X}$	$\hat{\beta}_1 = \frac{\sum_{i=1}^n (X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^n (X_i - \overline{X})^2}$
Statistic	$T_{xy} = r_{xy}\sqrt{\frac{n-2}{1-r_{xy}^2}}$	$t_0 = \frac{\hat{\beta}_0}{se(\hat{\beta}_0)}$	$t_1 = \frac{\hat{\beta}_1}{se(\hat{\beta}_1)}$
Null hypothesis	$H_0 : \rho_{xy} = 0$	$H_0 : \beta_0 = 0$	$H_0 : \beta_1 = 0$

Key formulas

For the SLR model, the fits and residuals have the following simpler forms:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i, \tag{5.28}$$

$$e_i = Y_i - (\hat{\beta}_0 + \hat{\beta}_1 X_i). \tag{5.29}$$

From the residuals, we can compute the error or residual sum of squares as

$$\text{SSE} = \sum_{i=1}^n e_i^2. \tag{5.30}$$

The least squares estimate of the error variance σ^2 is the Mean Squared Error,

$$\hat{\sigma}^2 = \text{MSE} = \frac{\text{SSE}}{n-2} = \frac{\sum_{i=1}^n e_i^2}{n-2}. \tag{5.31}$$

F-test in the MLR Model

A test for significance of the overall regression model fit helps to test the existence of a linear relationship between the response variable and the set of predictors and is referred to as test for model adequacy or test for goodness of fit of the regression. The null hypothesis is

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$$

and the alternative hypothesis is

$$H_1 : \beta_j \neq 0 \text{ for at least one } j \in \{1, 2, \ldots, p\}.$$

TABLE 5.2: Analysis of Variance Table

Source	DF	Sums of Squares	Mean Squares	F-stat	Prob $> F$
Regression	p	SSR $= \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$	MSR $= \frac{\text{SSR}}{p}$	$F_0 = \frac{\text{MSR}}{\text{MSE}}$	p-value
Residual	$n-p-1$	SSE$=\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$	MSE $= \frac{\text{SSE}}{n-p-1}$		
Total	$n-1$	SST$=\sum_{i=1}^{n}(Y_i - \bar{Y})^2$			

Let $\bar{Y}$ denote the mean response. The corrected Total Sum of Squares, corrected Regression Sum of Squares and the Error (Residual) Sum of Squares are:

$$\text{SST} = \sum_{i=1}^{n}(Y_i - \bar{Y})^2, \tag{5.32}$$

$$\text{SSR} = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2, \tag{5.33}$$

$$\text{SSE} = \sum_{i=1}^{n} e_i^2. \tag{5.34}$$

SST is the total variation in the response variable Y corrected for the mean, SSR is the corrected variability explained by the fitted MLR model and SSE is the variation unexplained by the fitted MLR model. The ANOVA decomposition partitions the corrected Total Sum of Squares as follows:

$$\text{SST} = \text{SSR} + \text{SSE}.$$

This is summarized in the ANOVA table shown in Table 5.2. If the null hypothesis is true, the F-statistic F_0 has an F-distribution with degrees of freedom p and $n-p-1$.

Marginal Tests and Confidence Intervals for Coefficients

We test $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$ separately for each j via a t-test. The test statistic is

$$t_0 = \hat{\beta}_j / se(\hat{\beta}_j) = \hat{\beta}_j / \hat{\sigma}\sqrt{C_{jj}}, \tag{5.35}$$

where C_{jj} is the jth diagonal element of $(\mathbf{X}'\mathbf{X})^{-1}$. The statistic t_0 has a Student t-distribution with $n-p-1$ degrees of freedom under H_0. The $100(1-\alpha)\%$ confidence interval for β_j, $j = 0, 1, \ldots, p$, can be obtained by inverting the t-statistic in (5.35) and is

$$\hat{\beta}_j \pm t_{\alpha/2, n-p-1}\ se(\hat{\beta}_j).$$

In the SLR model (where $p = 1$), these forms for the slope β_1 and intercept β_0 are shown below. The statistic for testing $H_0 : \beta_1 = 0$ against the alternative hypothesis $H_1 : \beta_1 \neq 0$ is

$$t_{\beta_1} = \frac{\hat{\beta}_1}{se(\hat{\beta}_1)}, \text{ where,}$$

$$se(\hat{\beta}_1) = \sqrt{\frac{\text{MSE}}{\sum_{i=1}^{n}(X_i - \overline{X})^2}}. \tag{5.36}$$

The $100(1-\alpha)\%$ confidence interval for the slope β_1 is

$$\hat{\beta}_1 \pm t_{\alpha/2,n-2}\ se(\hat{\beta}_1). \tag{5.37}$$

For the intercept β_0, these formulas are respectively given by

$$\begin{aligned} t_{\beta_0} &= \frac{\hat{\beta}_0}{se(\hat{\beta}_0)}, \text{ where,} \\ se(\hat{\beta}_0) &= \sqrt{\frac{\text{MSE}}{(\frac{1}{n} + \frac{\bar{X}^2}{\sum_{i=1}^{n}(X_i-\overline{X})^2})}}, \end{aligned} \tag{5.38}$$

and

$$\hat{\beta}_0 \pm t_{\alpha/2,n-2} se(\hat{\beta}_0). \tag{5.39}$$

Extra Sum of Squares F-Tests

In many situations, we may wish to assess the "extra" contribution of a subset of predictors to the corrected Regression Sum of Squares. The *Extra Sum of Squares* F-test enables such analyses. The Extra Sum of Squares principle refers to a general idea in hypothesis testing, in which a test of hypothesis is formulated as a problem of comparing two models for the response variable Y, a full model and a reduced model, where the reduced model is a subset of (or is nested within) the full model. The full model is a general model that adequately describes the data, for instance the MLR model in (5.4) with p predictors $X_1, \ldots, X_p$, or the fixed effects two-way ANOVA model with interactions between Factor A and Factor B that we discussed in Chapter 4. The reduced model is a special case (subset) of the full model and is obtained by imposing on the full model the restrictions of a null hypothesis H_0 which constrains a few of the full model parameters to zero. Examples are an MLR model where the last set of $p_2 < p$ predictors are hypothesized to be zero, or the fixed effects two-way additive ANOVA model where the interaction terms are hypothesized to be zero.

To compute the Extra Sum of Squares, we need to fit the full model and reduced model separately, and extract information about the Error (Residual) Sum of Squares and corresponding degrees of freedom from each output. Let SSE(full) and SSE(red.) respectively denote the full and reduced model Error (Residual) Sums of Squares, and let d.f.(full) and d.f.(red.) denote the respective degrees of freedom. Let MSE(full) denote the MSE in the full model. The Extra Sum of Squares is computed as the difference of SSE(full) from SSE(red.):

$$\text{Extra SS} = \text{SSE(red.) - SSE(full)}$$

with d.f.

$$\text{Extra d.f.} = \text{Error d.f.(red.) - Error d.f.(full).}$$

The Extra Sum of Squares F-statistic is

$$\text{Extra SS } F\text{-stat} = \frac{\text{Extra SS/Extra d.f.}}{\text{MSE(full)}}. \tag{5.40}$$

Consider the MLR model in (5.4) with p predictors. Let $\mathbf{X}_1$ and $\mathbf{X}_2$ denote the matrices corresponding to the first p_1 predictors and the last p_2 predictors respectively, with

$p_1 + p_2 = p$. Suppose $\boldsymbol{\beta}$ is partitioned conformably, so that $\boldsymbol{\beta}_1 = (\beta_1, \ldots, \beta_{p_1})'$ and $\boldsymbol{\beta}_2 = (\beta_{p_1+1}, \ldots, \beta_p)'$. Suppose we wish to test $H_0 : \boldsymbol{\beta}_2 = \mathbf{0}$ versus $H_1 : \boldsymbol{\beta}_2 \neq \mathbf{0}$. The full model is the MLR model with p predictors and an intercept, and the reduced model is the MLR model with the first p_1 predictors and an intercept. The Extra Sum of Squares F-statistic has the form in (5.40) and follows an $F_{p_2,n-p-1}$ distribution under H_0. If the Extra SS F-stat $> F_{p_2,n-p-1,\alpha}$, we reject the null hypothesis, since the extra variation explained by including $\mathbf{X}_2$ in the model which already has $\mathbf{X}_1$ and an intercept is greater than what we would attribute to chance.

Note that we can also express the Extra Sum of Squares in terms of the Regression Sums of Squares in the full and reduced models:

$$\text{Extra SS} = \text{SSR(full) - SSR(red.)}$$

with d.f.

$$\text{Extra d.f.} = \text{Regression d.f.(full) - Regression d.f.(red.).}$$

Partial F-tests

The Extra Sum of Squares F-statistic is called a partial F-statistic when only a single regression coefficient is tested, i.e., when $p_1 = p - 1$ and $p_2 = 1$. Partial F-tests can be used to test the significance of each of the p coefficients individually. For instance, the partial (extra) sum of squares for testing $H_0 : \beta_1 = 0$ in a full model containing all p predictors is the difference between the Error Sum of Squares in the reduced model without β_1 and the Error Sum of Squares in the full model with all p parameters including β_1:

$$\text{Partial SS for } \beta_1 = \text{SSE}(\beta_0, \beta_2, \ldots, \beta_p) - \text{SSE}(\beta_0, \beta_1, \beta_2, \ldots, \beta_p)$$

and the resulting partial F-statistic has an $F_{1,n-p-1}$ distribution under H_0. The partial F-statistic for testing for the other β_j's have similar forms. Note that these partial sums of squares for the p coefficients need not add up to SSR in the full model.

Sequential F-tests

We can represent a partition of SSR into single degree of freedom contributions from explanatory variables that are added sequentially one at a time to a model with just an intercept. Denote the sequential Regression Sum of Squares (SS) for β_1 as

$$\text{SS}(\beta_1|\beta_0) = \text{SSR}(\beta_1, \beta_0) - \text{SSR}(\beta_0).$$

Similarly, the sequential Regression SS for β_2 is

$$\text{SS}(\beta_2|\beta_1, \beta_0) = \text{SSR}(\beta_2, \beta_1, \beta_0) - \text{SSR}(\beta_1, \beta_0),$$

etc., and the the sequential Regression SS for β_p is

$$\text{SS}(\beta_p|\beta_{p-1}, \ldots, \beta_1, \beta_0) = \text{SSR}(\beta_p, \ldots, \beta_1, \beta_0) - \text{SSR}(\beta_{p-1}, \ldots, \beta_1, \beta_0).$$

Then,

$$\text{S}SR = \text{SS}(\beta_1|\beta_0) + \text{SS}(\beta_2|\beta_1, \beta_0) + \ldots + \text{SS}(\beta_p|\beta_{p-1}, \ldots, \beta_1, \beta_0).$$

Sequential SS partitioning enables us to assess the contribution of each explanatory variable individually. The corresponding test statistics are called sequential F-statistics and enable us to test the significance of the contribution of an explanatory variable in a model containing the previous variables. Note that the order of entry of the variables into the model will affect the results.

Prediction in the MLR Model

It is useful to discuss predicting an out-of-sample (a) mean response, or (b) individual Y value. Which we use depends on the nature of the question asked in the analysis.

Predicting the Mean Response $\mu(Y|\mathbf{x}_0)$

Let $\mathbf{x}_0 = (1, X_{1,0}, \ldots, X_{p,0})$ denote the $(p+1)$-dimensional vector corresponding a new set of values of the predictors. The predicted response is

$$\hat{Y}_0 = \mathbf{x}_0'\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \hat{\beta}_1 X_{1,0} + \ldots + \hat{\beta}_p X_{p,0}, \tag{5.41}$$

with

$$\text{Var}(\hat{Y}_0) = \sigma^2 \mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0. \tag{5.42}$$

The $100(1-\alpha)\%$ confidence interval for the true mean value of Y at $\mathbf{x}_0$ is given by

$$\hat{Y}_0 \pm t_{n-p-1,\alpha/2}\ \hat{\sigma}\sqrt{\mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0}. \tag{5.43}$$

In the SLR model, let $\mu(Y|X_0)$ denote the mean response at a new value X_0 of the independent variable. The point prediction of $\mu(Y|X_0)$ is

$$\hat{Y}_0 = \hat{\beta}_0 + \hat{\beta}_1 X_0, \tag{5.44}$$

and the $100(1-\alpha)\%$ confidence interval for $\mu(Y|X_0)$ is

$$\begin{aligned} &\hat{Y}_0 \pm se(\hat{\mu}(Y|X_0)),\ \text{where,} \\ &se(\hat{\mu}(Y|X_0)) = \sqrt{MSE}\left(\frac{1}{n} + \frac{(X_0 - \overline{X})^2}{\sum(X_i - \overline{X})^2}\right)^{1/2} \end{aligned} \tag{5.45}$$

Predicting an Individual Response

If we wish to predict an individual Y response given $\mathbf{x}_0$, the point estimate is again $\hat{Y}_0$ given by (5.41), while the $100(1-\alpha)\%$ prediction interval is

$$\hat{Y}_0 \pm t_{n-p-1,\alpha/2}\ \hat{\sigma}\sqrt{1 + \mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0}. \tag{5.46}$$

As discussed in Section 5.5, the prediction interval in (5.46) is wider than the C.I. for the mean response in (5.43).

In the SLR model, suppose we wish to predict an individual response $(Y|X_0)$ at a new value of the predictor X_0. The predicted value of an individual response is the same as the prediction of the mean response, and is given by (5.44). The corresponding $100(1-\alpha)\%$ confidence interval of $(Y|X_0)$ is however wider than the interval obtained in (5.45):

$$\begin{aligned} &\hat{Y}_0 \pm se(\hat{Y}_0|X_0),\ \text{where,} \\ &se(\hat{Y}_0|X_0) = \sqrt{MSE}\left(1 + \frac{1}{n} + \frac{(X_0 - \overline{X})^2}{\sum(X_i - \overline{X})^2}\right)^{1/2} \end{aligned} \tag{5.47}$$

6

Linear Regression – More Topics

This chapter describes additional approaches for fitting a multiple linear regression model. We discuss numerical and graphical diagnostics to detect violation of regression assumptions, including outlying, high leverage, or influential cases, or the problem of multicollinearity in the predictors. The chapter also discusses remedies for detected problems. The last section describes variable selection using classical methods such as best subset regression and stepwise regression, as well as modern approaches such as regularized regression such as lasso, and elasticnet.

6.1 Introduction

Regression diagnostics refer to graphical, informal numerical procedures, or statistical inferential methods that enable us to assess the validity of a fitted regression model – to assess whether model assumptions are satisfied, to study the effect due to collinearity in the explanatory variables, or study the influence of one or more cases (observations) on different aspects of the fitted model.

Diagnostic measures are used for detecting outliers, high leverage cases, or influential cases. A case may be influential to the regression fit because it is an outlier, or a high-leverage point, or both. Belsley et al. (1980) give an extensive description of these topics, while a more theoretical discussion is found in Chatterjee and Hadi (1988).

Multicollinearity implies that two or more predictors are highly linearly dependent. This issue can crop up in several examples, especially as the predictor dimension increases. Diagnostics for multicollinearity only depend on the predictor matrix $\mathbf{X}$, and remedies range from judiciously omitting problematic predictors from the model to early computer-age methods such as ridge regression, or principal components regression.

Variable selection in MLR models enables us to identify predictors that best explain the response. We discuss classical approaches such as best subsets regression and stepwise regression, as well as more recent approaches such as regularized regression (lasso, or elastic net), random forest regression, gradient boosting, etc.

6.2 Essential Terminology and Notation

✓ The **eigenvalues** $\lambda_1, \ldots, \lambda_k$ of a symmetric $k \times k$ matrix $\mathbf{A}$ are solutions to the equation $|\mathbf{A} - \lambda\mathbf{I}| = 0$. Then, the determinant of $\mathbf{A}$ is the product of the eigenvalues.

✓ The **eigenvectors** $\mathbf{q}_1, \ldots, \mathbf{q}_k$ of a symmetric $k \times k$ matrix $\mathbf{A}$ with eigenvalues $\lambda_j,\ j = 1, \ldots, k$ satisfy the equations $\mathbf{A}\mathbf{q}_j = \lambda_j \mathbf{q}_j,\ j = 1, \ldots, k$.

✓ A square matrix $\mathbf{A}$ is an **orthogonal matrix** if its transpose is its inverse, i.e., $\mathbf{A}' = \mathbf{A}^{-1}$. It follows that $\mathbf{A}\mathbf{A}' = \mathbf{A}'\mathbf{A} = \mathbf{I}$.

✓ A square matrix $\mathbf{A}$ is a **singular matrix** if $|\mathbf{A}| = 0$, and is nearly singular if $|\mathbf{A}|$ is very close to 0.

✓ The **spectral decomposition** of a symmetric matrix $\mathbf{A}$ is

$$\mathbf{A} = \mathbf{Q}\boldsymbol{\Delta}\mathbf{Q}' \tag{6.1}$$

where $\mathbf{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_k)$ is a matrix whose columns are the orthogonal eigenvectors of $\mathbf{A}$ corresponding to its ordered eigenvalues denoted by $\lambda_{(1)} \leq \lambda_{(2)} \leq \ldots \leq \lambda_{(k)}$, and $\boldsymbol{\Delta} = \text{diag}(\lambda_{(1)}, \ldots, \lambda_{(k)})$.

✓ Let $\overline{X}_j = \frac{1}{n}\sum_{i=1}^n X_{ij}$, $S_{jh} = \sum_{i=1}^n (X_{ij} - \overline{X}_j)(X_{ih} - \overline{X}_h)$, for $j, h = 1, \ldots, p$, and $S_{yy} = \sum_{i=1}^n (Y_i - \overline{Y})^2$. The **centered and scaled form of the MLR model** is

$$Y_i^* = \beta_1^* X_{i1}^* + \beta_2^* X_{i2}^* + \ldots + \beta_p^* X_{ip}^* + \epsilon_i, \tag{6.2}$$

where for $i = 1, \ldots, n$ and $j = 1, \ldots, p$,

$$\begin{aligned} X_{ij}^* &= \frac{X_{ij} - \overline{X}_j}{\sqrt{S_{jj}}}, \\ Y_i^* &= \frac{Y_i - \overline{Y}}{\sqrt{S_{yy}}}, \text{ and,} \\ \beta_j^* &= \beta_j (S_{jj}/S_{yy})^{1/2}. \end{aligned}$$

Let

$$\mathbf{X}^* = \begin{pmatrix} X_{11}^* & X_{12}^* & \ldots & X_{1p}^* \\ X_{21}^* & X_{22}^* & \ldots & X_{2p}^* \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1}^* & X_{n2}^* & \ldots & X_{np}^* \end{pmatrix};$$

then $\mathbf{X}^{*\prime}\mathbf{X}^*$ denotes the $p \times p$ sample correlation matrix of the explanatory variables (since the intercept column becomes zero by the centering). The MLR model in (6.29) can be written in vector-matrix notation as

$$\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta}^* + \boldsymbol{\epsilon}. \tag{6.3}$$

✓ **Regression diagnostics** refer to graphical or informal numerical procedures or statistical inferential methods that enable us to assess the validity of a fitted regression model. The diagnostics help us assess whether model assumptions are satisfied through a careful look at the model residuals, or to study the influence of one or more cases (observations) on different aspects of the fitted model.

✓ An **outlier** is an anomalous observation that does not reasonably fit the assumed model so that the observed response deviates considerably from the fitted response. Simply stated, an outlying case is one with an unusually large absolute residual.

✓ The **predicted residual** for the ith case is defined as

$$e_{i(i)} = Y_i - \hat{Y}_{i(i)} = \frac{e_i}{1 - h_{ii}}, \tag{6.4}$$

where for $i = 1, \ldots, n$, $\hat{Y}_{i(i)}$ is the fitted value for the ith response when the ith case is omitted in fitting the model (5.4).

✓ A **branch-and-bound algorithm** is a computing paradigm used for solving combinatorial optimization problems, such as selecting a best subset of predictors from a set of p predictors in an MLR model. The set of candidate solutions forms a tree, whose root consists of all the predictors. This algorithm efficiently searches the branches of this tree by using bounds on the objective function to prune large parts of the tree.

6.3 Regression Diagnostics

Choosing a model in order to draw conclusions about the relationship between predictors and some variable, y, or in order to make accurate prediction is only the first step in the data analysis process. After we collect data and fit the model we must check whether the model is adequate. If not, we must check why and see if there are steps we can take which will remediate some problems in our model. We discussed some notions of goodness of fit in previous chapters, and now we provide a more detailed approach which is especially tailored for regression models. We call this process "regression diagnostics", and we use it to check validity of the assumptions of the MLR model, and to investigate whether the data contains any "unusual" points (e.g. outliers in the data set, or points which have a large influence on the results from the modeling.)

In this chapter we still focus on the linear regression framework, so it is good to recall the underlying assumptions, which are the linearity of the relationship between the mean response and predictors, the equal error variance for all cases, and independence and normality of the errors and responses. As usual, we explain the process with a motivating example. In this case, we introduce an interesting dataset which wil allow us to investigate factors contributing to deforestation in Western China.

Deforestation

Example 6.3.1 (Deforestation) Agricultural populations whose livelihood heavily depends on natural resources such as timber production may be affected by deforestation which results from the increasing demand for timber. The data in this example was collected from four mountainous provinces in Western China, (Deng et al. (2011)). The data

file deforest.csv has observations from 405 counties, and $p = 17$ predictors which may help explain changes in the forest cover. The variable `fcover` in our data is the observed cover divided by 1000. The predictors included in this dataset are: total population per square km (`tpop`), agricultural population per square km (`apop`), per capita primary industry added value 10,000 yuan (`gdp1`), per capita secondary industry added value 10,000 yuan (`gdp2`), per capita third industry added value 10,000 yuan (`gdp3`), distance to provincial capital (`distpvc`), distance to port or open city (`distport`), length of highway (`hwaylen`), length of natural express (`explen`), proportion of plain area (`plain`), average elevation (`elev`), average precipitation for many years (`prec`) average slope in degrees (`slope`), average temperature in degrees Celsius (`temp`), accumulated temperature greater than 0 degrees Celsius (`temp0`), total land area (`area`), and an indicator for important reforestation policies (`policy`).

Our research question is stated as follows:

Research question

Is deforestation in the four mountainous provinces in Western China related to any of the predictors in the data?

This is the general question, but we will see that getting the answer is not as simple as fitting a linear model with all 17 predictors. In order to answer it properly, we must go through several steps in which we will identify, and try to remediate some problems.

We read the data from deforest.csv into `data1`. There are 16 continuous predictors, and one binary predictor. The response is the square-root of `fcover`.

```
data1 <- read.csv("/Data/deforest.csv", header = TRUE)
data1$sqrtfcover <- sqrt(data1$fcover)
```

We do an 80-20 split of `data1` into a training (calibration, or fitting) set, and a test (hold-out) set using code shown below. That is, starting with a random seed, we randomly sample without replacement 80% of the rows of `data1` into a training set denoted by `train.set`, while the remaining 20% go into a test set, denoted by `test.set`. We will build a regression model on the training data and validate the fit on the test data.

```
set.seed(123457)
train.prop <- 0.80
trnset <- sort(sample(1:nrow(data1), ceiling(nrow(data1)*train.prop)))
# create the training and test sets
train.set <- data1[trnset, ]
test.set  <- data1[-trnset, ]
```

Next, we standardize the train and test sets. To do this correctly, we must use the mean and standard deviation of the training set to standardize the continuous variables in *both* the training and test sets using the R package *caret*, as the code below shows. Note that we do not standardize the binary predictior `policy` is binary. The data sets `data.train` and `data.test` contain the standardized continuous predictors, the binary predictor and the response (square root of `fcover`).

```
library(caret)
```

```
contpredcols <- 5:20
# Find mean and std dev of train.set.1
normParam <- preProcess(train.set[,contpredcols],
                        method = c("center", "scale"))
# standardize the training set based on its mean and std
data.train <- cbind(train.set[,c("sqrtfcover", "policy")],
                    predict(normParam, train.set[,contpredcols]))
# standardize the test set based on the above mean and std dev of the
    training set
data.test <- cbind(test.set[,c("sqrtfcover", "policy")],
                   predict(normParam, test.set[,5:20]))
```

Note that it would be incorrect to (a) use the mean and standard deviation of the entire data to standardize both sets, or (b) use the mean and standard deviation of the test data to standardize the test data set.

Remark 1. If the binary or categorical variable is character-valued (and not factors), then the function preProcess() will find the mean and scale only for continuous variables.

We regress the response (square root of forest cover) on the predictors in `pred`, which has $n = 324$ rows.

```
mod.1 <- lm(sqrtfcover ~., data = data.train)
summary(mod.1)
plot(mod.1)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.33554    0.18571  61.037  < 2e-16 ***
policy       -0.34658    0.25149  -1.378  0.16918
tpop          1.24817    0.60013   2.080  0.03837 *
apop         -1.60689    0.58919  -2.727  0.00675 **
gdp1          0.02893    0.12092   0.239  0.81107
gdp2          0.03107    0.12316   0.252  0.80097
gdp3         -0.12546    0.10933  -1.148  0.25207
distpvc       0.64993    0.34667   1.875  0.06177 .
distport     -1.63072    0.19628  -8.308 3.19e-15 ***
hwaylen      -0.69049    0.34755  -1.987  0.04785 *
explen       -0.06788    0.11886  -0.571  0.56837
plain        -0.33745    0.13519  -2.496  0.01308 *
elev         -0.02341    0.32213  -0.073  0.94210
prec          0.38424    0.21509   1.786  0.07502 .
slope         2.09290    0.18867  11.093  < 2e-16 ***
temp         11.82725    1.50847   7.841 7.51e-14 ***
temp0       -10.35239    1.48779  -6.958 2.10e-11 ***
area          4.48566    0.18103  24.779  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.856 on 306 degrees of freedom
Multiple R-squared:  0.8428,        Adjusted R-squared:  0.8341
F-statistic:  96.5 on 17 and 306 DF,  p-value: < 2.2e-16
```

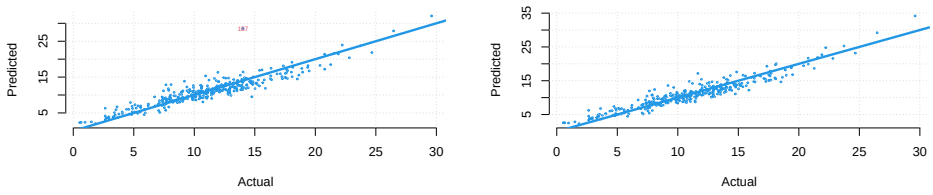

FIGURE 6.1: Fitted versus actual plot from a least squares fit of all standardized predictors on $\sqrt{\text{fcover}}$ for the deforestation data. Left plot uses all training data, right plot excludes the two outliers.

Based on the p-values, we can see that some variables are significant in explaining $\sqrt{\text{fcover}}$, while others seem to be ineffective. There are two anomalous cases, denoted by `extpts`, cases 187 and 152. The fitted versus actual plot for the training data in the left panel of Figure 6.1 shows reasonably good fit, and indicates an outlier, case 187. Note that in an MLR model, we classify the ith case (or observation) as an outlier if the magnitude of the raw residual, internally Studentized residual r_i, and/or the externally Studentized residual r_i^* (see Table 5.1) is large when compared with the rest of the cases in the data set. Note that the plot(mod.1) function can produce a set of five useful plots.

```
plot(data.train$sqrtfcover, predict(mod.1,newdata = data.train),
     col=4, cex=0.3, xlab="Actual", ylab="Predicted", axes=FALSE)
extpts <- which(abs(residuals(mod.1)) > 3*sd(residuals(mod.1)))
text(data.train$sqrtfcover[extpts],
     predict(mod.1,newdata = data.train)[extpts],
     rownames(data.train)[extpts], cex=0.5, col=2)
axis(1); axis(2); grid(); abline(0,1, col=4, lwd=3)
```

The presence of outliers may seriously bias parameter estimation and inference in the MLR model discussed in Chapter 5. We can look at results from fitting an MLR model to the training data without the two outliers. We can compare the fitted versus actual plot based on the smaller training data set with the one in Figure 6.1.

```
data.train.2 <- data.train[-extpts,]
mod.2 <- lm(sqrtfcover ~., data = data.train.2)
summary(mod.2)
## diagnostic plots:
plot(mod.2)
## fitted versus actual plot:
plot(data.train.2$sqrtfcover, predict(mod.2,newdata = data.train.2),
     col=4, cex=0.3, xlab="Actual", ylab="Predicted", axes=FALSE)
axis(1); axis(2); grid(); abline(0,1, col=4, lwd=3)
```

6.3.1 High leverage cases

Let $\mathbf{x}_i = (1, X_{i1}, \ldots, X_{ip})'$ be the vector of explanatory variables for the ith case and let $\overline{\mathbf{x}} = (1/n, \overline{X}_1, \ldots, \overline{X}_p)'$ be the average of the columns of the $\mathbf{X}$ matrix. The diagonal elements

h_{ii} of the hat (or projection) matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ have some useful properties:

$$0 \leq h_{ii} \leq 1 \text{ and } \sum_{i=1}^{n} h_{ii} = p + 1. \tag{6.5}$$

A large value of h_{ii} indicates that $\mathbf{x}_i$ is far removed from the average $\overline{\mathbf{x}}$, i.e., the ith case is an *outlier in the X space*. The reciprocal $1/h_{ii}$ is the effective or equivalent number of observations that determines $\widehat{Y}_i$ (Huber (1981)). We may also interpret the off-diagonal elements of $\mathbf{H}$, i.e., h_{ij} as the amount of *leverage* each Y_j has on determining the ith fit $\widehat{Y}_i$, irrespective of its actual value (Hoaglin and Welsch (1978)). Since

$$\widehat{Y}_i = h_{ii}Y_i + \sum_{j \neq i} h_{ij}Y_j, \tag{6.6}$$

a large value of h_{ii} implies that Y_i important in determining $\widehat{Y}_i$, and that the residual e_i is small (since $\text{Var}(e_i) = \sigma^2(1 - h_{ii})$). When $h_{ii} = 1$, we can show that $h_{ij} = 0$ for $j \neq i$, so that $\widehat{Y}_i = Y_i$, indicating that $e_i = 0$, and $\text{Var}(e_i) = 0$. Huber (1981) suggested the following rules of thumb to identify the ith case as a high leverage point: (i) $h_{ii} \geq \frac{2(p+1)}{n}$, or (ii) $h_{ii} > 0.5$.

```
n <- nrow(data.train)
p <- ncol(data.train)-1
(hilev <- which(influence(mod.1)$hat > max(2*(p+1)/n,0.5)))
```

An L-R plot is useful for distinguishing between outlier and high leverage points. This plot combines the leverage values and residuals in a single scatter plot of leverages h_{ii} versus squared normalized residuals $\widehat{a}_i^2$, where $\widehat{a}_i = e_i/\sqrt{\mathbf{e}'\mathbf{e}}$, $i = 1, \ldots, n$. The points in this plot must lie within the triangle satisfying the following conditions: (i) $0 \leq h_{ii} \leq 1$, (ii) $0 \leq \widehat{a}_i^2 \leq 1$, and (iii) $h_{ii} + \widehat{a}_i^2 \leq 1$. Points that fall in the lower right corner of the L-R plot are outliers, while points that fall in the upper left corner are high leverage points. The plot is constructed using the code below and is shown in Figure 6.2.

```
par(mfrow=c(1,1))
plot(rstandard(mod.1)^2, influence(mod.1)$hat, pch =19, cex=0.5, col="blue",
     xlab = "squared residual", ylab = "leverage")
inf0 <- which(influence(mod.1)$hat > 0.5)
text(rstandard(mod.1)[hilev]^2, influence(mod.1)$hat[hilev], labels = inf0,
    cex = 0.9, font =2, pos =1)
```

6.3.2 Influential cases

We say that an observation (or case) is influential, if its deletion from the data set would noticeably change a result in the fitted MLR model. Neither outliers nor high-leverage points are necessarily influential. We can choose to study the influence of the ith case on the properties of $\widehat{\boldsymbol{\beta}}$, or the fitted values $\widehat{\mathbf{y}}$, or individual estimated coefficients $\widehat{\beta}_j$ $j = 1, \ldots, p$. Here, we discuss three widely used measures of influence Huber (1981)), i.e., Cook's Distance, Welsch-Kuh Distance (DFFITS), and DFBETAS. These measures follow from an application

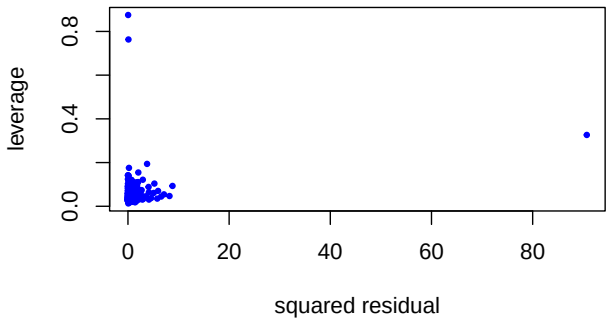

FIGURE 6.2: L-R plot for the deforestation data.

of the influence function to the MLR model (Cook and Weisberg (1983); Chatterjee and Hadi (1988)) and are summarized in Table 6.1.

We need the following notation. For $j = 1, \ldots, p$, if $\mathbf{x}_j$ denotes the column for the jth predictor, let $\mathbf{X}_{(j)}$ be the predictor matrix without the jth column; then, let $\mathbf{H}_{(j)} = \mathbf{X}_{(j)}(\mathbf{X}'_{(j)}\mathbf{X}_{(j)})^{-1}\mathbf{X}'_{(j)}$, and $\mathbf{W}_j = (\mathbf{I} - \mathbf{H}_{(j)})\mathbf{x}_j$.

TABLE 6.1: Influence diagnostics

Diagnostic	Formula	Influence of the ith case on what
Cook's distance	$C_i = \frac{1}{p}\frac{h_{ii}}{(1-h_{ii})}r_i^2$	on least squares estimates and fits
DFFITS	$WK_i = \lvert r_i^* \rvert \sqrt{\frac{h_{ii}}{1-h_{ii}}}$	on the ith fitted response $\hat{Y}_i$
DFBETAS	$r_i^* \frac{w_{ij}}{\sqrt{\mathbf{W}_j'\mathbf{W}_j}} \frac{1}{\sqrt{1-h_{ii}}}$	on the jth coefficient $\hat{\beta}_j$

Cook's Distance

Cook's Distance C_i is an influence diagnostic which can be calculated from a single MLR model fit using the formula shown in Table 6.1, as a function of the leverage h_{ii} and Studentized residual r_i^2 for the ith case. Note that C_i is large if h_{ii} is large, or r_i^2 is large, or both. Although it has been suggested that each C_i be compared to $F_{p,n-p,1-\alpha}$, we can use a boxplot of $C_i,\ i = 1, \ldots, n$ to answer the question 'how large is large?'. Two other useful interpretations of C_i are given in the appendix.

Welsch-Kuh Distance (DFFITS)

The Welsch-Kuh distance (DFFITS) measures the influence of the ith case on the ith fitted response $\hat{Y}_i$ by the change in the ith fitted value when the ith case is omitted, relative to

the se($\hat{Y}_i$):

$$WK_i = \frac{|\hat{Y}_i - \hat{Y}_{i(i)}|}{\hat{\sigma}_{(i)}\sqrt{h_{ii}}} \tag{6.7}$$

While cut-off points like $2\sqrt{p/(n-p)}$ have been recommended to decide whether WK_i is large, we can also look at a boxplot of these values and see whether some cases are outliers.

DFBETAS

It is also possible to measure the influence of the ith case on a single regression coefficient $\hat{\beta}_j$ using DFBETAS$_{ij}$ shown in Belsley et al. (1980) suggested that absolute values of $DFBETAS_{ij}$ exceeding $2/\sqrt{n}$ are influential on $\hat{\beta}_j$. For the deforestation data, we can obtain the DFBETAS for each β_j using the code below.

```
dfbetas.all  <- as.data.frame(dfbetas(mod.1))
dfbet.fun=function(x){
  which(abs(x) > 2/sqrt(n))
}
hidfbeta=apply(dfbetas.all, 2, dfbet.fun)
```

We can also look at boxplots of the residuals, leverages, Cook's D, DFFITS, and selected DFBETAS (plots not shown here).

```
# Boxplots for residuals, leverages, Cook's D and DFFITS
par(mfrow=c(2,2))
boxplot (rstudent (mod.1), sub =" Stud . resid ")
boxplot (influence(mod.1)$hat , sub=" leverages ")
boxplot(cooks.distance(mod.1), sub = "Cook's D")
boxplot(dffits(mod.1), sub = "DFFITS")
par(mfrow = c(1,2))
boxplot(dfbetas.all[,3], sub ="apop")
boxplot(dfbetas.all[,4], sub ="gdp1")
```

6.4 Multicollinearity

As we saw in Chapter 5, an MLR model fit is useful if the response variable Y is highly correlated with the set of explanatory variables, as indicated by a high R^2 value. However, it is necessary that the explanatory variables are not highly correlated *among themselves*. The problem of multicollinearity exists when there is high correlations or 'near-dependency' between the columns of $\mathbf{X}$, resulting in the matrix $\mathbf{X}'\mathbf{X}$ being nearly singular. Then, from (5.21) and (5.26), we see that the least squares estimates as well as their variances and covariances can become unstable. In such cases, the data/model pair is said to be ill-conditioned.

Although multicollinearity does not affect the fitted values much, a high degree of multicollinearity tends to cause the following problems in an MLR fit:

(i) The standard errors of the regression coefficients will be very large, resulting in small associated t-statistics , leading to the conclusion that truly useful explanatory variables are insignificant in explaining the regression.

(ii) The sign of regression coefficients may be the opposite of what a mechanistic understanding of the problem would suggest.

(iii) Deleting a row or a column of the $\mathbf{X}$ matrix could cause large changes in the estimates corresponding to coefficients of other variables; in general, the regression fit can be unstable.

Several approaches have been suggested in the literature for the detection of multicollinearity as well as to mitigate or remedy its effect.

Deforestation

Example 6.4.1 (Deforestation) We continue to use the deforestation data from the deforestation.csv file in order to investigate factors driving deforestation in four mountainous provinces in Western China. We continue to use the standardized continuous predictors and square root of forest cover.

Our research question is now stated as follows:

Research question #2

Are any of the 17 predictors linearly dependent on any of the others? If this is the case, how can we detect which predictors are? How can we remedy this and fit a regression to explain `sqrtfcover`?

6.4.1 Detecting Multicollinearity

Assess correlations among predictors

A simple approach consists of assessing the simple, multiple, and partial correlations among the predictors. A commonly used rule is that if a correlation coefficient exceeds 0.95 or 0.99, there is a possibility of collinearity between the corresponding pair of predictors.

```
pred.df <- data.train[,-1] #data frame of predictors only
cor.pred <- cor(pred.df)
off.diag <- function(x) x[col(x) > row(x)]
v <- off.diag(cor.pred)
table(v >=0.95)
table(v >=0.99)
```

While two correlations exceed 0.95, one correlation exceeds 0.99. However, pairwise correlations do not give much insight into interrelationships between three or more predictors.

Another rule suggests multicollinearity if a simple correlation exceeds the multiple correlation from the MLR fit as shown below.

The code below shows that three simple correlations exceed the R^2 value shown in the output `mod.1` shown in Section 6.3.

```
table(v > summary(mod.1)$r.squared)
```

We can also compute the pairwise *partial* correlation coefficients between the predictors. One suggested rule of thumb is a partial correlation coefficient greater than 0.5. However, the cut-off value of 0.5 is quite arbitrary, and further partial correlations only capture linear relationships between pairs of variables, and would not be useful in detecting multicollinearity that is a result of a more complex dependency. For example, X_1 may not be highly correlated individually with X_2 or X_3, but it may be highly correlated with some linear combination of X_2 and X_3.

```
library(ppcor)
pcor1 <- pcor(pred.df)
vp <- off.diag(pcor1$estimate)
table(vp >= 0.5)
```

Results from the code above shows that three partial correlations exceed 0.5. Another indication of multicollinearity is that the F-statistic from the MLR model fit is large implying that the predictors taken together explain the mean response well, but almost all the marginal t-statistics for the individual coefficients are small. In such cases, it is difficult to decide whether the coefficient a for predictor is insignificant due to multicollinearity or simply because the predictor does not explain the response well. Also, this procedure would not indicate which explanatory variable is highly correlated with other variables. Variance inflation factors (VIFs) are helpful diagnostics forthis purpose.

Variance Inflation Factor (VIF)

For $j = 1, \ldots, p$, let R_j^2 denote the coefficient of determination of a linear regression of the explanatory variable X_j on the remaining $p-1$ explanatory variables, i.e.,

$$X_{i,j} = \gamma_0 + \sum_{\ell=1}^{j-1} \gamma_\ell X_{i,\ell} + \sum_{\ell=j+1}^{p} \gamma_\ell X_{i,\ell} + \epsilon_i, \ i = 1, \ldots, n.$$

A large value of R_j^2 close to 1 indicates that X_j is collinear with at least one of the other $p-1$ explanatory variables. The quantity

$$VIF_j = \frac{1}{1 - R_j^2} \tag{6.8}$$

denotes the variance inflation factor for X_j. If X_j is uncorrelated with (or orthogonal to) the other $p-1$ explanatory variables, $R_j^2 = 0$, so that $VIF_j = 1$. As R_j^2 increases from zero, VIF_j increases as well. For example, if $R_j^2 = 0.8$, then $VIF_j = 5.0$, while if $R_j^2 = 0.99$, then $VIF_j = 100$.

General guidelines exist in the literature on how large VIF_j should be to indicate multicollinearity. One suggestion is that any VIF_j greater than 10 (or, greater than 30) indicates multicollinearity. Alternatively, we can compute the average of the variance inflation factors, i.e., $\overline{VIF} = \frac{1}{p}\sum_{j=1}^{p} VIF_j$, and if $\overline{VIF}$ considerably exceeds 1, multicollinearity is indicated.

The code and results below show that the VIF's corresponding to `temp` and `temp0` exceed

200. Although the values are much lower, `tpop`, `apop`, `distpvc` and `hwaylen` also have VIF's bigger than 10. The results suggest evidence of multicollinearity.

```
car::vif(mod.1)
```

```
>car::vif(mod.1)
    policy       tpop       apop       gdp1       gdp2       gdp3    distpvc   distport
  1.409765  33.773757  32.553432   1.371252   1.422323   1.120951  11.269900   3.612788
   hwaylen     explen      plain       elev       prec      slope       temp      temp0
 11.327585   1.324763   1.713787   9.730837   4.338333   3.338234 213.387048 207.574291
      area
  3.073224
```

Condition Indices and Condition Number

Let the (real) eigenvalues of $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ be $\lambda_j,\ j = 1, \ldots, p$; $0 \leq |\mathbf{X}^{*\prime}\mathbf{X}^{*}| \leq 1$. Since $|\mathbf{X}^{*\prime}\mathbf{X}^{*}| = \prod_{j=1}^{p} \lambda_j$, if one or more eigenvalues are close to zero, $|\mathbf{X}^{*\prime}\mathbf{X}^{*}|$ will be close to zero as well. Let $d_j = \lambda_j^2,\ j = 1, \ldots, p$. The k *condition indices* are defined as

$$\eta_j = \frac{d_{\max}}{d_j},\ j = 1, \ldots, p. \tag{6.9}$$

A value of d_j which is relatively close to zero will be associated with a large condition index. If d_j is exactly equal to zero, there is an exact linear relationship among the p explanatory variables.

The code below shows how to construct the condition indices. Note that unlike the VIF's, the 18 values shown below must not be associated drectly with the intercept and 17 predictors (see the appendox for some details).

```
library(olsrr)
(mod.condind <- ols_eigen_cindex(mod.1)[,2])
```

```
 [1]  1.000000  1.672370  1.764450  1.932458  2.373442  2.448060  2.632633  2.723273
 [9]  3.128310  3.651675  4.341692  4.968210  5.477556  6.430869  8.484049 11.364146
[17] 20.191643 50.895901
```

The *condition number* is defined as

$$C = \frac{d_{\max}}{d_{\min}}, \tag{6.10}$$

and always exceeds 1. A large condition number, say $C > 15$ (or $C > 30$), indicates evidence of strong multicollinearity and a need for corrective action. The code below enables us to compute the condition number of 50.8959 for this data.

```
(mod.condnum <- max(mod.condind)/min(mod.condind))
```

6.4.2 Remedies for Multicollinearity

We discuss a few usual remedies for handling multicollinearity in MLR models.

Dropping predictors from the model

Once detected, an obvious remedy could be to drop the variables that are highly correlated with the others. However, if the dropped variable is potentially valuable in an understanding of the response, we would get absolutely no information about it. Moreover, it may not always be clear as to how the omission of a variable will affect the estimates of the remaining model parameters. A second, but not a very promising, solution to the problem is to include additional data, if available, to the analysis; in some cases, this might resolve the multicollinearity. Formal and more rigorous statistical procedures for dealing with the multicollinearity problem include ridge regression, and principal components regression.

We refit a model by excluding one of the predictors with the largest VIF; here, we can drop the variable `temp`, and refit the MLR model and obtain the VIF's.

```
data.train.1 <- subset(data.train, select = -c(temp))
mod.droptemp <- lm(sqrtfcover ~., data = data.train.1)
summary(mod.droptemp)
anova(mod.droptemp)
 car::vif(mod.droptemp)
```

```
> car::vif(mod.droptemp)
   policy      tpop      apop      gdp1      gdp2      gdp3   distpvc  distport   hwaylen
 1.396241 33.509798 32.528053  1.356162  1.420866  1.120451 10.329263  3.375474 10.409090
   explen     plain      elev      prec     slope     temp0      area
 1.311268  1.701280  9.713470  4.224137  3.052512  6.356137  2.704171
```

The VIF's are now much smaller than 200, and the VIF for `temp0` is only 6.35. We now see two variables, `tpop` and `apop`, with VIF's exceeding 30, and two variables `distpvc` and `hwaylen` with VIF's just above 10. We can refit an MLR by dropping `tpop`, and get the VIF's shown below.

```
data.train.2 <- subset(data.train.1, select = -c(tpop))
mod.droptpop <- lm(sqrtfcover ~ ., data = data.train.2)
summary(mod.droptpop)
anova(mod.droptpop)
car::vif(mod.droptpop)
```

```
> car::vif(mod.droptpop)
   policy      apop      gdp1      gdp2      gdp3   distpvc  distport   hwaylen    explen
 1.394528  1.901877  1.351789  1.418626  1.118372 10.006197  3.286475 10.191837  1.308195
    plain      elev      prec     slope     temp0      area
 1.674786  9.713219  4.182711  3.022363  6.336799  2.695253
```

The LS estimates from this model after dropping `temp` and `tpop` are shown below.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.20193    0.20421  54.854  < 2e-16 ***
policy      -0.12904    0.27608  -0.467 0.640528
apop        -0.14494    0.15719  -0.922 0.357201
gdp1         0.14740    0.13252   1.112 0.266880
gdp2         0.07559    0.13576   0.557 0.578068
gdp3        -0.13046    0.12054  -1.082 0.279968
distpvc      1.27179    0.36054   3.527 0.000484 ***
distport    -1.93938    0.20663  -9.386  < 2e-16 ***
hwaylen     -1.33245    0.36387  -3.662 0.000295 ***
explen       0.01025    0.13036   0.079 0.937412
plain       -0.38120    0.14750  -2.584 0.010216 *
elev        -0.13467    0.35523  -0.379 0.704867
prec         0.16914    0.23311   0.726 0.468637
slope        2.47577    0.19815  12.494  < 2e-16 ***
temp0        1.17265    0.28692   4.087 5.58e-05 ***
area         4.02095    0.18712  21.488  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.048 on 308 degrees of freedom
Multiple R-squared:  0.8072,        Adjusted R-squared:  0.7978
F-statistic: 85.98 on 15 and 308 DF,  p-value: < 2.2e-16
```

We can look at the p-values from the marginal t-tests in the above output to see which variables are significant in explaining `sqrtfcover`. We can look at the residual diagnostic plots from this model by the command plot(mod.droptpop).

Remark 1. We can also use the *dplyr* package for dropping the variable `temp`; see the sample code below.

```
data.train.1 <- select(data.train, -c(temp))
```

Ridge Regression

Ridge regression is a procedure which addresses multicollinearity in an MLR model. Instead of minimizing the error sum of squares in the centered and scaled MLR model defined in (6.20), we minimize the error sum of squares subject to a penalty function denoted by $\lambda \parallel \boldsymbol{\beta} \parallel^2]$. We can denote the ridge estimates as $\widehat{\boldsymbol{\beta}}^*(\lambda) = (\widehat{\beta}_1^*(\lambda), \ldots, \widehat{\beta}_p^*(\lambda))'$. When the penalty parameter $\lambda = 0$, the ridge estimates coincide with the LS estimates of the parameters. More details are given in the appendix. A plot of $\widehat{\beta}_j^*(\lambda)$ versus λ is known as the *ridge trace*, and can be useful for selecting a suitable value of $\lambda \neq 0$ such that the ridge estimates of β_j^* will stabilize to reasonable values. We can also choose λ using a k-fold cross-validation approach, as we show below.

For the deforestation data, ridge regression estimates can be obtained using the *glmnet* package. The function maximizes a penalized likelihood function (see Section 6.6 for more details in the context of regularized regression). This function does not work with data frames, so we must create a numeric matrix for the predictors (training features) and a vector of responses (target values).

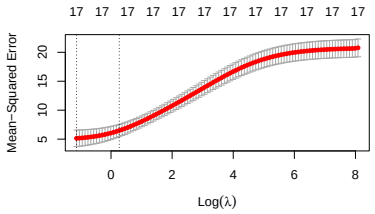

FIGURE 6.3: MSE versus log(λ) using the *glmnet* package

```
pred.mat <- as.matrix(pred.df)
resp <- data.train$sqrtfcover
```

We can use the glmnet() function and manually select the value of λ as follows.

```
mod.ridge.1=glmnet(pred, resp, alpha = 0, nlambda = 100, family = "gaussian",
          standardize=FALSE, intercept = TRUE)
```

We can get the same output by just using the simpler code below, allowing all other options to be at their default settings.

```
mod.ridge.1=glmnet(pred, resp, alpha = 0, standardize=FALSE)
```

If we choose the last of the 100 λ values, we can obtain the ridge estimates at this $\lambda = 0.3240187$ using the code below.

```
(lambda.m  <- mod.ridge.1$lambda[100])
coef(mod.ridge.1, s= lambda.m)
```

The *glmnet* package allows us to select the best λ value using k-fold cross-validation, by minimizing a user selected criterion, such as MSE. We use 10-fold validation below. Figure 6.3 plots the criterion, MSE versus log(λ).

```
cvfit.ridge <- cv.glmnet(pred.mat,resp,alpha=0, standardize=FALSE,
                         type.measure = "mse", nfolds = 10)
plot(cvfit.ridge)
```

We can select the λ value which minimizes the mean cross-validated error.

```
cvfit.ridge$lambda.min
```

```
[1] 0.3240187
[1] 1.308069
```

Or, we can select the λ value which gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

```
cvfit.ridge$lambda.1se
```

```
[1] 1.308069
```

We can obtain the ridge estimates for lambda.min and lambda.1se, as shown below.

```
(all.coef <- cbind(coef(cvfit.ridge, s = "lambda.min"),
                   coef(cvfit.ridge, s = "lambda.1se")))
```

```
(Intercept) 11.17102307 11.14515660
policy      -0.07871911 -0.03660486
tpop         0.17234495 -0.02564813
apop        -0.27277357 -0.12012392
gdp1         0.02156346 -0.09547346
gdp2         0.01684329 -0.06999666
gdp3        -0.16293999 -0.16744178
distpvc      0.58672995  0.48782723
distport    -1.53486017 -1.04680556
hwaylen     -0.33502190  0.18424987
explen       0.05094388  0.10235664
plain       -0.30424267 -0.21065072
elev         0.39327040  0.43910665
prec         0.27549262  0.28368329
slope        1.97680655  1.27846803
temp         0.99010972  0.42703827
temp0        0.24887686  0.25125813
area         3.32437651  2.29857012
```

Note that while the coefficients are shrunk towards zero, no sparsity is achieved since of none of the coefficients becomes exactly zero.

Ridge regression estimates of the standardized coefficient vector $\boldsymbol{\beta}$ can also be obtained using the lm.ridge() function in the *MASS* package.

Another remedy for multicollinearity, called principal components regression, is sometimes used. This method invokes the idea of reducing the dimension of the predictor space by regressing Y on a set of orthogonal linear combinations of the original standardized predictors.

6.5 Variable Selection

In MLR modeling, variable selection consists of using statistical procedures to choose a subset of all available explanatory variables (or predictors or features) for inclusion in the final model (possibly with reduced dimension) in order to best explain and predict the response Y. Starting with p available predictors, suppose we wish to fit an MLR model (5.4) which includes q of these variables, using suitable methods. In general, variable selection seeks the *principle of parsimony*, i.e., if two models fit the data equally well, the simpler (smaller) model is the better model. We describe a few procedures that differ in their approaches and computational complexity.

We continue discussing the deforestation data. Our research question is stated as follows:

Research question #2

Can we explain and predict `sqrtfcover` well using only a select few out of the $p = 17$ predictors? How can we select these variables?

6.5.1 Best Subsets Regression

Given an MLR model (5.4) with p predictors, our goal is to select the best subset $X_1, X_2, \ldots, X_q$, where $q \leq p$, according to some criterion. The *best subsets regression* approach compares all the $2^p + 1$ models based on suitable variable selection criteria. Criteria such as SSE, MSE, R^2 and adjusted R^2 which we defined in Chapter 5 for the MLR model (5.4) are useful criteria for variable selection. We show these, along with a few other useful criteria in the table below.

Criterion	Formula	Best value
SSE_p	$\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$	Smallest
MSE_p	$\frac{\text{SSE}_p}{n-p}$	Smallest
R^2_p	$1 - \frac{\text{SSE}_p}{\text{SST}}$	Largest
$R^2_{\text{adj},p}$	$1 - \left(\frac{n-1}{n-p}\right)\frac{\text{SSE}_p}{\text{SST}}$	Largest
Mallows C_p	$\frac{SSE_p}{MSE} - (n - 2p)$	Small, closest to p
PRESS_p	$\sum_{i=1}^{n} e^2_{i(i)} = \sum_{i=1}^{n}(Y_i - \widehat{Y}_{i(i)})^2$	Smallest
AIC_p	$n\log(\text{SSE}_p) - n\log n + 2p$	Smallest
BIC_p	$n\log(\text{SSE}_p) - n\log n + p\log(n)$	Smallest

Key formulas

Remark 1. When selecting the best model, we look for models with the largest values of R^2 or adjusted R^2, and smallest values of SSE, MSE, PRESS, AIC, or BIC.

Remark 2. Unlike SSE_p, MSE_p does not decrease monotonically as p increases, and serves as a reasonable variable selection criterion.

Remark 3. While R^2_p increases monotonically as p increases and does not penalize the model for increasing dimension, adjusted R^2 penalizes the model for increasing dimension.

Remark 4. Mallows (1973) showed that if a fitted model is adequate and does not suffer from lack of fit, we expect that $C_p \approx p$, while models with substantial bias will tend to fall considerably above the line $C_p = p$.

Remark 5. In general, the BIC_p criterion favors more parsimonious models compared to the AIC_p criterion.

The data with standardized response and predictors is in the data frame `data.train`. The regsubsets() function in the R package *leaps* performs an exhaustive search for the best subsets of the predictors to explain the response Y in an MLR model, using an efficient branch-and-bound algorithm and one of several criteria such as R^2, Mallows C_p, or BIC. When there are p possible predictors, and any subset of the predictors is allowed to specify a model, there will be $2^p + 1$ possible models; this includes a model with just the intercept and none of the predictors.

In the code below, the int = TRUE option includes an intercept in each model. The nvmax = ncol(pred.df) option says that the maximum size of subsets to examine is $p = 17$; the default is 8. The nbest =1 option asks for the best model of each size to report; while the default is also 1, if we set it as 2, then two subsets of each size will be reported. We can also obtain a criterion, such as the Bayesian Information Criterion (BIC).

```
library(leaps)
out <- regsubsets(x = pred.df, y = resp, int = TRUE, nvmax = ncol(pred.df),
    nbest =1)
(bic <- summary(out)$bic)
```

```
 [1] -218.8674 -270.1163 -426.4575 -495.6111 -532.4471 -540.6090 -537.6201 -535.4083
 [9] -532.6811 -529.7310 -525.4680 -522.5265 -517.9942 -512.5991 -506.9044 -501.1818
[17] -495.
```

The following code outputs a logical matrix (output not shown) indicating which elements are in each model, which includes p predictors, $p = 1, \ldots, 17$. We can now pull out more information about the best model based on the smallest BIC (i.e., -505.6398). We write out the estimated coefficients from the best model. The selected predictors are `apop`, `distport`, `slope`, `temp`, `temp0`, and `area`.

```
(elmts <- summary(out)$which)
new <- as.data.frame(cbind(vars, bic))
min.bic <- min(new$bic)
rw <- which(new$bic==min.bic)
coef(out, rw)
```

```
 (Intercept)        apop    distport       slope        temp       temp0        area
 11.1226740  -0.4778413  -1.5891113   2.1007565  12.2341041 -10.4065223   4.4836042
```

In the above analysis, we used the default method = "exhaustive" option. Other available options are forward selection, backward selection or sequential replacement. Instead of the BIC criterion, we can use the rsq, adjr2 or cp options corresponding to the R^2, adjusted R^2, or Mallow's C_p. When $p > 50$, we must use the really.big = TRUE option.

6.5.2 Stepwise Regression

Stepwise regression procedures seek to reduce the computing complexity of best subsets regression by sequentially selecting variables based on *partial* F-tests. A popular approach is *stepwise selection* which includes forward selection and backward elimination as special cases. Stepwise selection uses partial F-statistics (equivalently, t-statistics) and associated $p-$values in order to decide whether to include or exclude each of the p predictors into the model.

Stepwise selection for the deforestation data uses the following code and selects a model whose coefficients are shown below. We see that this method selects a model with 12 out of the 17 predictors. Note that `direction = "both"` implies both forward and backward selection.

```
fit.step <- lm(sqrtfcover ~., data = data.train)
mod.step <- step(fit.step, direction = "both", trace = 0)
summary(mod.step)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  11.3614     0.1770  64.178  < 2e-16 ***
policy       -0.3887     0.2349  -1.654  0.09904 .
tpop          1.2477     0.5936   2.102  0.03635 *
apop         -1.5965     0.5783  -2.761  0.00611 **
distpvc       0.6154     0.3249   1.894  0.05912 .
distport     -1.6484     0.1748  -9.431  < 2e-16 ***
hwaylen      -0.6938     0.3394  -2.044  0.04177 *
plain        -0.3306     0.1331  -2.485  0.01349 *
prec          0.3825     0.1885   2.030  0.04325 *
slope         2.0688     0.1750  11.821  < 2e-16 ***
temp         11.8463     1.4824   7.992 2.63e-14 ***
temp0       -10.3715     1.4628  -7.090 9.05e-12 ***
area          4.4830     0.1681  26.676  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.846 on 311 degrees of freedom
Multiple R-squared:  0.8419,        Adjusted R-squared:  0.8358
F-statistic:   138 on 12 and 311 DF,  p-value: < 2.2e-16
```

How does the stepwise selection work? To illustrate the idea, suppose $K = 4$ predictors are available, denoted by X_1, X_2, X_3, and X_4. Our goal is to select the best MLR model of the form (5.4) involving $p \leq K$ predictors. The algorithm consists of the following steps.

Step 0. Specify two different levels of significance α_{enter} and α_{stay}. These are respectively the probability of a Type I error for including a predictor into the current MLR model and the probability of a Type I error for retaining in the model a predictor variable that was previously entered. Recommended values are $0.05, 0.10$ or 0.15. All the models are fit to cases $i = 1, \ldots, n$.

Step 1. Fit K single predictor models for $j = 1, \ldots, K$ denoted by

$$Y_i = \beta_0 + \beta_1 X_{ij} + \epsilon_i. \tag{6.11}$$

Compute the partial F-statistic (or t-statistic) and the corresponding p-value for testing $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ under each of the K models. Note that under H_0, the model becomes

$$Y_i = \beta_0 + \epsilon_i. \tag{6.12}$$

(i) If the p-values exceed α_{enter} for all K models, the procedure stops, i.e., no variables can enter the model (6.12), and we assume that Y can be best explained by just the intercept, estimated by $\overline{Y}$.

(ii) If at least one of the p-values is less than or equal to α_{enter}, then the predictor with the smallest such p-value (i.e., the largest partial F-statistic value) enters the model (6.12). Denote this variable by $X_{[1]}$, so the model at the end of Step 1 is

$$Y_i = \beta_0 + \beta_1 X_{i[1]} + \epsilon_i. \tag{6.13}$$

In the illustration, suppose X_3 enters the model in Step 1, while X_1, X_2 and X_4 do not; then $X_{[1]} = X_3$.

Step 2. Fit $K - 1$ models with two predictors, of which $X_{[1]}$ was selected in Step 1, while a second variable must be selected from the remaining $K - 1$ predictors:

$$Y_i = \beta_0 + \beta_1 X_{i[1]} + \beta_2 X_{ij} + \epsilon_i. \tag{6.14}$$

For each model, compute the partial F-statistic and p-value for testing $H_0 : \beta_2 = 0$ versus $H_1 : \beta_2 \neq 0$. Let $X_{[2]}$ denote the predictor corresponding to the smallest p-value (or largest t-statistic). The variable $X_{[2]}$ will be included in the model (6.13) if its p-value is smaller than α_{enter}, to give

$$Y_i = \beta_0 + \beta_1 X_{i[1]} + \beta_2 X_{i[2]} + \epsilon_i. \tag{6.15}$$

The stepwise procedure then checks to see whether or not $X_{[1]}$ should continue to remain in the model. If the p-value corresponding to $H_0 : \beta_1 = 0$ versus $H_1 : \beta_1 \neq 0$ in the model (6.15) is smaller than α_{stay}, $X_{[1]}$ is retained in the model and the model at the end of Step 2 is (6.15); otherwise, it is dropped from the model and the current one-variable model after Step 2 is

$$Y_i = \beta_0 + \beta_2 X_{i[2]} + \epsilon_i. \tag{6.16}$$

We now are back to the position at the start of Step 2, and must search for another varible that is significant and can be included in the model. In the illustration, suppose X_1 enters the model in Step 2, so the model now includes $X_{[1]} = X_3$ and $X_{[2]} = X_1$.

This method continues with the steps of adding predictor variables into the model, one at a time. At each step, a variable is added to the model only if it has the smallest p-value (or largest t-statistic) among all variables that are not in the model, and also, it is significant at the α_{enter} level. Once a variable is added to the model, the procedure checks all the variables in the model and removes any variable that is not significant at the α_{stay} level, doing this deletion step before attempting to add another *new* variable. The procedure terminates when all the predictor variables not included in the model are insignificant at the α_{enter} level, or when the variable to be added into the model is the one that was just removed.

Forward selection is a simpler version of stepwise selection, where a variable that has entered the model stays in the model and is not removed. For *backward elimination*, we start with a rich model that includes all K predictors and uses partial F-statistics and corresponding p-values to eliminate variables that are not significant in the presence of other variables; once removed, a variable may not reenter the model.

Remark 1. We can obtain results from forward selection for the deforestation data by running the code below (output is not shown here). changing direction = "both" to direction = "forward". We can also use the regsubsets() function in the *leaps* package.

```
out.forward <- regsubsets(x = pred.df, y = resp, int = TRUE, nbest = 1,
    method = "forward")
```

Remark 2. For backward elimination, use direction = "backward"

6.6 Regularized Regression

Recall that we obtain the least squares regression estimates coefficients of the MLR model in (5.21) to minimize the sum of squared errors in (5.20) (a loss function). In practice, we split the rows of the data frame randomly into training data and test data. We will build a model on the training dataset, holding out the test data, i.e., not using the test data for model fitting. We will then evaluate performance of the fitted model on the test dataset. This is called the *holdout-validation* approach for evaluating model performance.

If the coefficients of the MLR model are large, they can lead to over-fitting on the training dataset, and the fitted model will not generalize well on the unobserved test data. Regularization can help to overcome this shortcoming by penalizing large coefficients. That is, in regularized regression, we seek to estimate coefficients which minimize a *penalized* error sum of squares.

6.6.1 Lasso regression

The most popular variant of regularized regression is *Lasso regression* which minimizes

$$S(\beta_0, \beta_1, \ldots, \beta_p, \lambda) = \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{6.17}$$

where, $\lambda \sum_{j=1}^{p} |\beta_j|$ is the penalty term, and λ is called the *shrinkage penalty* parameter which we must select. Lasso is an acronym for least absolute shrinkage and selection operator, and the term $\sum_{j=1}^{p} |\beta_j| = \|\boldsymbol{\beta}\|_1$ is called the ℓ_1-norm. By using the ℓ_1-norm constraint, we force some of the regression coefficients to zero, and the corresponding less important predictors will be discarded from the fitted model. The predictors (features) corresponding to non-zero coefficients will remain in the fitted MLR model. By reducing the coefficients of the less important predictors (features) to zero, lasso regression is used for *feature selection*. This is a useful technique, especially when p is large.

We can use the R package *glmnet* to fit Lasso regression to the deforestation data. The regularization path is computed for the penalty (regularization) parameter λ at a grid of values (on the log scale). The code to get Lasso estimates is similar to what we used to carry out ridge regression, except for replacing `alpha=0` by `alpha =1`, which is also the default in the *glmnet* package. We can again use 10-fold cross-validation to select the λ value which minimizes the mean cross-validated error, or the λ value which gives the most regularized model such that the cross-validated error is within one standard error of the minimum.

```
cvfit.lasso <- cv.glmnet(pred.mat, resp,alpha=1,
                         standardize=FALSE, type.measure = "mse", nfolds =
    10)
plot(cvfit.lasso)
```

```
>cvfit.lasso$lambda.min
[1] 0.0006820283
> (cvfit.lasso$lambda.1se)
[1] 0.2181991
```

We can then obtain the lasso estimates corresponding to these two λ values (shown below). When we use the ℓ_2-norm constraint $\|\boldsymbol{\beta}\|_2$ instead of the ℓ_1-norm in (6.17), the regularized regression becomes the ridge regression we saw earlier. Note that while Lasso regression shrinks some of the regression coefficients to zero, ridge regression retains all the model predictors.

6.6.2 Elastic net

Elastic net is a widely used regularized regression approach which is a combination of lasso and ridge regression. Similar to ridge regression, Lasso pertains to a convex optimization problem. However, unlike ridge regression, Lasso is not always strictly convex, and therefore, it need not always have a unique solution. Zou and Hastie (2005) defined "elastic net", which is always strictly convex, and combines the predictive properties of ridge regression with the sparsity properties of Lasso.

The elastic net penalty

$$P_\alpha(\boldsymbol{\beta}) = \alpha\|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2 \tag{6.18}$$

is a convex combination of the Lasso penalty and the ridge regression penalty, $\alpha \in (0, 1)$. The elastic net estimates minimize the criterion

$$\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|_2^2 + \lambda\{\alpha \| \boldsymbol{\beta} \|_1 + (1-\alpha) \| \boldsymbol{\beta} \|_2^2\},$$

where $\lambda > 0$ is an unknown penalty parameter. The elastic net estimates are also obtained using the R package *glmnet*, and can be viewed as stabilized Lasso estimates of the coefficients. The code below is similar to what we used for ridge and Lasso regressions, except that we let $\alpha = 0.5$.

```
cvfit.enet <- cv.glmnet(x, y,alpha=0.5, standardize=FALSE, type.measure =
    "mse", nfolds = 10)
```

```
>cvfit.enet$lambda.min
[1] 0.00149705
> (cvfit.enet$lambda.1se) # (Default)
[1] 0.4363981
```

By writing the regression estimates side-by-side, we can compare the output from (left to right) OLS, ridge, lasso and elastic net estimates.

```
(all.coef.1 <- cbind(mod.1$coef, coef(cvfit.ridge, s = "lambda.1se"),
coef(cvfit.lasso, s = "lambda.1se"), coef(cvfit.enet, s = "lambda.1se")))
```

```
(Intercept)  11.33554099 11.14515660 11.122673986 11.1226740
policy       -0.34657744 -0.03660486  .            .
tpop          1.24816817 -0.02564813  .            .
apop         -1.60688702 -0.12012392  .            .
gdp1          0.02893167 -0.09547346  .            .
gdp2          0.03107325 -0.06999666  .            .
gdp3         -0.12545952 -0.16744178  .            .
distpvc       0.64993428  0.48782723  0.211369389  0.3931055
distport     -1.63071675 -1.04680556 -1.444079017 -1.3527115
hwaylen      -0.69049129  0.18424987  .            .
explen       -0.06787578  0.10235664  .            .
plain        -0.33744629 -0.21065072 -0.001878076  .
elev         -0.02341371  0.43910665  .            .
prec          0.38423546  0.28368329  .            .
slope         2.09290274  1.27846803  1.969709185  1.7016063
temp         11.82725272  0.42703827  0.857053569  0.5672179
temp0       -10.35239292  0.25125813  .            .
area          4.48566486  2.29857012  3.311560175  2.9558426
```

Appendix Chapter 6-Statistical Details

Centered and Scaled Regression Model

The centered and scaled multiple linear regression model (it does not have an intercept) is

$$Y_i^* = \beta_1^* X_{i1}^* + \beta_2^* X_{i2}^* + \ldots + \beta_p^* X_{ip}^* + \epsilon_i, \tag{6.19}$$

where $\beta_j^* = \beta_j (S_{jj}/S_{yy})^{1/2},\ j = 1, \ldots, p$ and

$$X_{ij}^* = \frac{X_{ij} - \overline{X}_j}{\sqrt{S_{jj}}}, \quad \text{and} \quad Y_i^* = \frac{Y_i - \overline{Y}}{\sqrt{S_{yy}}}, \quad \text{for } i = 1, \ldots, n;\ j = 1, \ldots, p. \tag{6.20}$$

Let

$$\mathbf{X}^* = \begin{pmatrix} X_{11}^* & X_{12}^* & \ldots & X_{1p}^* \\ X_{21}^* & X_{22}^* & \ldots & X_{2p}^* \\ \vdots & \vdots & \vdots & \vdots \\ X_{n1}^* & X_{n2}^* & \ldots & X_{np}^* \end{pmatrix};$$

then $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ denotes the $p \times p$ sample correlation matrix of the explanatory variables (since the intercept column becomes zero by the centering).

Cook's Distance

Two other interpretations of Cook's distance C_i are useful.

(i) We can interpret C_i as the scaled distance between the centers of the joint confidence region for $\boldsymbol{\beta}$ under two setups (Cook and Weisberg (1983)). The least squares estimate $\hat{\boldsymbol{\beta}}$ is the center of the $100(1-\alpha)\%$ joint (ellipsoidal) confidence region for $\boldsymbol{\beta}$ when all n observations are used to fit the MLR model. When the ith observation is omitted, the center of the the joint confidence region changes to $\hat{\boldsymbol{\beta}}_{(i)}$. Cook's Distance is defined as

$$C_i = \frac{(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)})'(\mathbf{X}'\mathbf{X})(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)})}{(p+1)\hat{\sigma}^2}, \; i = 1, \ldots, n. \tag{6.21}$$

(i) We can also interpret C_i as the scaled distance between $\hat{\mathbf{y}}$ and $\hat{\mathbf{y}}_{(i)}$, i.e., the fitted response vector from the MLR model based on n observations, and the fitted model after deleting the ith case respectively:

$$C_i = \frac{(\hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)})'(\hat{\mathbf{y}} - \hat{\mathbf{y}}_{(i)})}{(p+1)\hat{\sigma}^2}, \; i = 1, \ldots, n. \tag{6.22}$$

Condition Indices and Condition Number

Suppose $\boldsymbol{X}^*$ is the $n \times p$ standardized predictor matrix.

The condition number C of $\mathbf{X}^{*\prime}\mathbf{X}^{*}$ is defined by

$$C = \sqrt{\frac{\max(\lambda_1, \ldots, \lambda_p)}{\min(\lambda_1, \ldots, \lambda_p)}}, \tag{6.23}$$

We can use the following approach for detecting multicollinearity, based on column-equilibrating the matrix $\mathbf{X}$ by dividing each column of $\mathbf{X}$ by the sum of squares of its elements. Let $\mathbf{X}_E$ denote the column-equilibrated matrix, in which the sum of squares of each column will be unity. We find the singular value decomposition of $\mathbf{X}_E$, i.e., $\mathbf{X}_E = \mathbf{U}\mathbf{D}\mathbf{V}'$, where $\mathbf{U}$ is $n \times p$, $\mathbf{D}$ is $p \times p$, and $\mathbf{V}$ is $p \times p$, $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{V}\mathbf{V}' = \mathbf{I}_p$. Let $\mathbf{D} = \text{diag}(d_1, \ldots, d_p)$, where the non-negative d_j's are called the singular values of $\mathbf{X}_E$.

Now

$$\mathbf{X}_E'\mathbf{X}_E = \mathbf{V}\mathbf{D}\mathbf{U}'\mathbf{U}\mathbf{D}\mathbf{V}' = \mathbf{V}\mathbf{D}^2\mathbf{V}' \tag{6.24}$$

gives the spectral decomposition of the symmetric matrix $\mathbf{X}_E'\mathbf{X}_E$, so that $\mathbf{D}^2 = diag(c_1, \ldots, c_p)$, where $c_j = d_j^2$ are the eigenvalues of $\mathbf{X}_E'\mathbf{X}_E$ while $\mathbf{V} = \{v_{ij}\}$ is the corresponding orthogonal eigenvector matrix.

The colldiag implement procedures Belsley et al. (1980)to examine the 'conditioning' of the matrix of independent variables by computing the condition indexes of the matrix. If the condition index larger than 30 it indicates collinearity problems. All large condition indexes may be worth investigating. If a large condition index is associated two or more variables with large variance decomposition proportions,(50%) these variables may be causing collinearity problems.

Ridge Regression

Consider the centered and scaled form of the MLR model defined in (6.20):

$$\mathbf{y}^* = \mathbf{X}^*\boldsymbol{\beta}^* + \boldsymbol{\epsilon}.$$

For $\lambda \geq 0$, ridge regression estimates of coefficients $\beta_j^*,\ j = 1, \ldots, p$ are obtained as

$$\widehat{\boldsymbol{\beta}}^*(\lambda) = (\mathbf{X}^{*\prime}\mathbf{X}^* + \lambda \mathbf{I}_p)^{-1}\mathbf{X}^{*\prime}\mathbf{y}^*. \tag{6.25}$$

Although this was not explicitly done by Hoerl and Kennard (1970), we can describe the ridge regression estimate $\widehat{\boldsymbol{\beta}}^*(\lambda)$ as a penalized maximum likelihood estimator, i.e.,

$$\widehat{\boldsymbol{\beta}}^*(\lambda) = \arg\min_{\boldsymbol{\beta}}[\| \mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|^2]. \tag{6.26}$$

When the penalty parameter $\lambda = 0$, (6.26) leads to the OLS estimate of $\boldsymbol{\beta}^*$. Values of $\lambda > 0$ penalize choices of $\boldsymbol{\beta}^*$ that lead to large values of $\| \boldsymbol{\beta} \|^2$, biasing $\widehat{\boldsymbol{\beta}}^*(\lambda)$ towards the origin (shrinkage).

The predictors in an MLR model can be numerical or categorical. When there are two or more continuous predictors, it is possible that they have widely different scales. In this case, it is important to consider standardizing them before including in the MLR model. Standardizing X_j consists of subtracting the mean and dividing by the standard deviation, i.e.,

$$X_{ij}^* = \frac{X_{ij} - \overline{X}_j}{\sqrt{S_{jj}}},\ i = 1, \ldots, n. \tag{6.27}$$

Failure to do this may result in Standardizing such variables can also help us accurately determine which variables are important in explaining the response. One way to decide whether or not to standardize the predictors is to see whether or not the spreads in the centered (mean-subtracted) variables are very different. If the spreads are more or less the same, standardizing may not be necessary. Also, if the predictors have nearly equal means, it may be enough to just scale the variables, without mean-subtraction in (6.27). Note that we may choose to standardize (or scale) the response Y, or not; the standardized response can be denoted by

$$Y_i^* = \frac{Y_i - \overline{Y}}{\sqrt{S_{yy}}},\ i = 1, \ldots, n. \tag{6.28}$$

We do not standardize categorical predictors; for ease of notation, we can just set $X_{ij}^* = X_{ij}$ for categorical predictors.

The MLR model with standardized response and continuous predictors can be represented as follows:

$$Y_i^* = \beta_0^* + \beta_1^* X_{i1}^* + \beta_2^* X_{i2}^* + \ldots + \beta_p^* X_{ip}^* + \epsilon_i. \tag{6.29}$$

If we do not standardize Y, we set $Y_i^* = Y_i$ in (6.29). The standardized coefficients β_j^* are related to the original coefficients β_j by

$$\beta_j^* = \beta_j (S_{jj}/S_{yy})^{1/2},\ j = 1, \ldots, p. \tag{6.30}$$

β_0^* denotes the intercept in the model. We recommend to always include an intercept in a model, except when the science of the domain precludes an intercept (i.e., requires fitting a regression through the origin). Once we fit the MLR model in (6.29), we can obtain the original coefficients inverting the relation in (6.30), i.e.,

$$\beta_j = \beta_j^* (S_{yy}/S_{jj})^{1/2},\ j = 1, \ldots, p. \tag{6.31}$$

7

Generalized Linear Models (GLIM)

This chapter explains the framework of the generalized linear model (GLIM) which is extensively used for regression analysis involving some types of non-normal responses such as counts, or categories. A GLIM uses a link function to connect the mean of a response variable to a linear function of one or more explanatory variables. A Fisher scoring (FS) algorithm enables us to estimate the unknown partial regression coefficients in the model. We discuss inference on these coefficients to assess goodness of model fit. We then show how the fitted GLIM can be used for making predictions of the response at new predictor values. We illustrate these for two examples, binary regression and regression for count responses.

7.1 Introduction

Recall from Chapter 5 that a linear regression model formulation enables us to adequately describe the relationship between the mean $\mu_i = E(Y_i)$ of a normally distributed response variable Y_i and one or more explanatory variables via a linear functional form, $\beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} = \eta_i$, say. That is, the MLR model was written in (5.5) as

$$E(Y_i) = \mu_i = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_p X_{ip} = \eta_i, \text{ say,}$$

for modeling the real-valued mean response μ_i by a real-valued linear predictor η_i, which we can regard as a *systemtic component*. In many applications, there is a need to model a response Y which is a count, or categorical (i.e., falling into one of two or more groups) variable. In these cases, the mean of Y_i is no longer real-valued; instead, μ_i is a positive real number for counts, and a value in the interval $[0, 1]$ for categories. Then, the linear model is no longer valid, since there is a mismatch between the μ_i (which is not on the real line) and the η_i (which is on the real line). Arguing that an MLR model is thus unsuitable for modeling *non-normal* responses such as counts or categories, McCullagh and Nelder (1989) developed the framework of generalized linear models (GLIMs). Rather than directly relating the mean response μ_i to a linear predictor η_i as we did in the MLR model (see Chapter 5), a generalized linear model (GLIM) relates η_i to a function $g(\mu_i)$ of the mean response, for a suitably selected *link function* $g(.)$. In other words, a GLIM does not assume a linear relationship between the response variable and the explanatory variables. Instead, a function $g(.)$ specifies the link between the mean response μ_i and the non-random (systematic) component η_i, and thus specifies how μ_i relates to the explanatory variables $X_{i1}, \ldots, X_{ip}$.

A GLIM is a flexible generalization of linear regression that allows for the response variable to have an error distribution other than the normal distribution, and provides a

rich class of statistical models for analyzing special types of non-Gaussian responses and is now ubiquitous in data analysis in virtually all domains of inquiry. Two popular instances of the GLIM are (i) models for count responses and (ii) binary response models. We will explain the GLIM setup including the underlying probability mechanism describing the response, the link functions, algorithms for estimating model parameters, model assessment and selection, and prediction.

When the response Y is continuous, following a normal distribution and the link function is the identity function, the mean response is a linear function of $\eta = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$ in the multiple linear regression model which is fit using least squares, as we have seen in Chapter 6. In this chapter, we discuss GLIM's for count and binary responses. Other approaches for binray responses, extensions of binary responses to more than two categories (levels), and generalized additive models are discussed in the next chapter.

7.2 Essential Terminology and Notation

✓ In GLIMs, the mean μ_i of a response variable Y_i is related to the linear predictor (systemtic component) η_i through a *link* function $g(.)$, i.e.,

$$g(\mu_i) = \eta_i = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_p X_{ip}. \tag{7.1}$$

✓ A **binary response variable** Y can assume a value of 0 or 1, with $P[Y = 1]{=}\pi$ and $P[Y = 0]{=}1 - \pi$, for $0 < \pi < 1$.

✓ A binary random variable Y has a **Bernoulli**(π) distribution if its probability mass function (p.m.f.) is

$$p(y; \pi) = \pi^y (1 - \pi)^{1-y},\ y = 0 \text{ or } 1; 0 < \pi < 1. \tag{7.2}$$

Then, $E(Y) = \pi$ and $Var(Y) = \pi(1 - \pi)$.

✓ An integer-valued random variable Y has a **Binomial**(m, π) distribution if its probability mass function (p.m.f.) is

$$p(y; m, \pi) = \binom{m}{y} \pi^y (1 - \pi)^{m-y},\ y = 0, 1, \ldots, m; 0 \le \pi \le 1. \tag{7.3}$$

Then, $E(Y) = m\pi$ and $Var(Y) = m\pi(1 - \pi)$. When $m = 1$, this is the Bernoulli(π) distribution.

✓ A **count response variable** Y can assume an integer value in the set $\{0, 1, 2, \ldots\}$, with mean $E(Y) = \lambda > 0$.

✓ A count random variable Y has a **Poisson**(λ) distribution if its probability mass function (p.m.f.) is

$$p(y; \lambda) = \exp(-\lambda)\lambda^y / y! \tag{7.4}$$

Then, $E(Y) = \lambda$ and $Var(Y) = \lambda$.

✓ A count random variable Y has a **negative binomial**(λ, κ) distribution if its probability mass function (p.m.f.) is

$$p(y; \lambda, \kappa) = \tag{7.5}$$

Then, $E(Y) = \lambda$ and $Var(Y) = \kappa\lambda > \lambda$ when $\kappa > 1$.

✓ When $Var(Y) = E(Y)$, then Y is said to be **nominally dispersed**. An example is the Poisson random variable.

✓ If $Var(Y) > E(Y)$, then Y is said to be **overdispersed**, for example, the negative binomial random variable. In some cases, due to clustering of events, or some contaminating influences, there is variation in the responses that does not coincide with that implied by the Poisson distribution where $Var(Y) = E(Y) = \mu$. Consider the situation where the dispersion of the data exceeds that implied by the Poisson model, i.e., $Var(Y) > E(Y)$.

✓ **Sensitivity**, or true positive (TP) is the probability that a binary response is predicted as a 1 (or, "yes"), given that it is an event (or,"yes").

✓ **Specificity**, or true negative (TN) is the probability that a binary response is predicted as a 0 (or, "no"), given that it is a non-event (or,"no").

✓ The **Receiver Operating Characteristics ROC)** curve is a metric used to evaluate the prediction accuracy in binary and multi-class classification. It quantifies the trade-off between the sensitivity or true positive rate (TPR) and specificity or false positive rate (FPR) of a prediction.

7.3 Loglinear Models for Counts

Suppose the responses Y_i are counts with means $\lambda_i,\ i = 1, \ldots, n$, and we wish to model the responses as functions of p independent variables that are collected in the data frame $X_{i,j},\ j = 1, \ldots, p$ for each i. The Poisson distribution is the most popular distribution for explaining counts. This distribution has nominal dispersion, i.e., the variance is equal to the mean. In situations where this is not the case, and there is *overdispersion*, another useful distribution is the negative binomial distribution whose p.m.f. was shown in (7.5). We refer to such models for count responses as loglinear models.

Snails

Example 7.3.1 (Snail counts) The snails data was described in Bloch and Willig (2006), for studying snail (gastropod) abundance in the Luquillo Forest Dynamics Plot (LFDP; 18°20 N, 65°49 W), a16-ha grid in the northwest of the Luquillo Experimental Forest (LEF) in the Luquillo Mountains of northeastern Puerto Rico. Although 17 species of gastropods are known to live in the Luquillo Forest Dynamics Plot, the dataset shows counts of the three most abundant species, *Caracolus caracolla* and *Gaeotis nigrolineata*, and *Nenia tridens*, at 40 different sites in the wet season in 1995. We denote these by `carcar`, `gaeig`, and `nentri` in the dataset. Note that snail abundance is determined based on the minimum number known to be alive (MKNA) at each site during each season (i.e., the maximum number of individuals captured within a season at a site).

The dataset also has several predictors that can affect snail abundance. `Elevation` and `Slope` (unitless) are continuous variables, ranging from 333 m asl to 428 m asl, and 0.7 to 65.1, respectively, whereas `Aspect` is a categorical variable that represents the compass directions and is aggregated into four levels. `Soil type` is a categorical variable with three levels, *Zarzal =1*, *Cristal =2* and *Prieto=3*. There are three canopy cover (`CC`) classes; CC level 1 (0–49% cover) experienced the most intensive logging and agriculture prior to 1934; CC level 2 (50–80% cover) was used for shade-coffee cultivation and other small scale mixed agriculture before 1934; and CC level 3 (80–100% cover) was lightly and selectively logged up to the 1950s. With Canopy openness (CO), higher numbers represent greater canopy openness. Litter cover (`LC`) measures the mean leaf litter cover at a point calculated as levels 1-5. Plant apparency quantifies the aerial density of all living vegetation at heights up to 3 m above the forest floor. Apparency is estimated separately for *Prestoea acuminata*, the sierra palm by `PAsp` and for all other plant species by `PAothers`.

Research question #2

Can we quantify how the environmental and habitat variables affect the abundance of a snail species?

We read the snails data.

```
snails <- read.csv("Data/snails.csv", header = TRUE)
```

We combine Aspect levels (1 and 2) and (5 and 6), and save the categorical variables as factor variables.

```
snails$Aspect[snails$Aspect==6]=5
snails$Aspect[snails$Aspect==2]=1
snails$Aspect <- as.factor(snails$Aspect)
snails$Soil <- as.factor(snails$Soil)
snails$CC <- as.factor(snails$CC)
snails$LC <- as.factor(snails$LC)
```

Let us see the structure of levels in these four predictors. The levels are shown in the order in which they appear in the data frame.

```
> unique(snails$Aspect)
[1] 5 7 8 1
> unique(snails$Soil)
[1] 1 4 6
> unique(snails$CC)
[1] 1 3 2
> unique(snails$LC)
[1] 3 1 2 4 5
```

Remark 1. To convert the categorical variables into factors, we could also use the following more compact code which uses the lapply() function.

```
cat.id <- which(colnames(snails) %in% c("Aspect","Soil","CC","LC"))
snails[,cat.id] <- lapply(snails[,cat.id],as.factor)
```

Remark 2. To create indicator (or dummy) variables corresponding to the levels of a factor, recall that we can use the ifelse() function.

```
Asp1 <- ifelse(Aspect == 1, 1, 0)
Asp5 <- ifelse(Aspect == 5, 1, 0)
Asp7 <- ifelse(Aspect == 7, 1, 0)
Asp8 <- ifelse(Aspect == 8, 1, 0)
```

Or, we can use packages such as *dummy* or *FastDummies*. These indicators may be included as predictors in glm() instead of the factor variables (the latter is simpler).

7.3.1 Poisson Loglinear Model

The Poisson loglinear model is a popular example of a GLIM for explaining Y_i:

$$Y_i|\lambda_i \sim \text{Poisson}(\lambda_i). \tag{7.6}$$

In (7.6), the sampling distribution for the responses is the Poisson distribution whose p.m.f. is given in (7.4). The logarithmic link function relates the ith mean response λ_i to the regression fit η_i, i.e.,

$$\begin{aligned} \log(\lambda_i) &= \eta_i, \text{ where,} \\ \eta_i &= \beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j} = \mathbf{x}_i'\boldsymbol{\beta}. \end{aligned} \tag{7.7}$$

Here, the first element of $\mathbf{x}_i$ is 1, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$. Since the mean response $\lambda_i > 0$, whereas η_i is real-valued, we need the log function to link the two in a correct way.

We use the glm() function to fit a Poisson loglinear GLIM to `Carcar` counts, using all the predictors; we call this the full model. The parameter estimates are obtained using the iterative Fisher scoring (FS) algorithm (sometimes called the iteratively reweighted least squares algorithm) which is discussed briefly in (7.14). The full model coefficients are shown below the code. The output shows that convergence was achieved after five FS iterations.

```
carcar.pf <- glm(Carcar~Elevation+Slope+Aspect+Soil+CC+LC+PA.sp+PA.other,
    family='poisson', data=snails)
```

```
>summary(carcar.pf)
Call:
glm(formula = Carcar ~ Elevation + Slope + Aspect + Soil + CC +
    LC + PA.sp + PA.other, family = "poisson", data = snails)

Deviance Residuals:
    Min       1Q   Median       3Q
-3.7901  -1.3740  -0.2230   0.8666
    Max
 4.8005
 Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.8037825  1.0392198   1.736 0.082615 .
Elevation   -0.0006231  0.0026609  -0.234 0.814847
Slope        0.0128226  0.0035358   3.626 0.000287 ***
Aspect5      0.6417406  0.1103234   5.817 5.99e-09 ***
Aspect7      0.2338650  0.0981155   2.384 0.017146 *
Aspect8     -0.0030597  0.0955173  -0.032 0.974446
Soil4        0.4990691  0.0702176   7.107 1.18e-12 ***
Soil6        1.0632136  0.1557836   6.825 8.80e-12 ***
CC2         -0.2814573  0.0969687  -2.903 0.003701 **
CC3         -0.2943464  0.0756138  -3.893 9.91e-05 ***
LC2          0.4036784  0.1074070   3.758 0.000171 ***
LC3          0.3750953  0.1102054   3.404 0.000665 ***
LC4         -0.0122835  0.1205093  -0.102 0.918812
LC5         -0.1519346  0.2053417  -0.740 0.459354
PA.sp       -0.0090556  0.0045251  -2.001 0.045371 *
PA.other     0.0011689  0.0014541   0.804 0.421490
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 717.45  on 150  degrees of freedom
Residual deviance: 445.72  on 135  degrees of freedom
AIC: 1047.7

Number of Fisher Scoring iterations: 5
```

The output shows the Poisson regression coefficients for each of the variables along with the standard errors, z-test statistics, and p-values. The estimated coefficient for `Elevation` is -0.0006231 with a SE of 0.0026609. This means that the decrease in expected log count $\log \lambda_i$ for one unit increase in `Elevation` is 0.0006231. Exponentiating this value gives 0.9993771, so that the multiplicative effect on expected count λ_i due to a unit increase in `Elevation` is 0.9993771. For `Slope`, these values are 0.0128226 and $\exp(0.0128226) = 1.012905$ respectively.

From the output, we see that for each factor variable, the glm() function has treated the lowest level as the baseline or reference level, and we do not see an estimated β_j for that level. Examples are `Aspect1`, `Soil1`, etc. The coefficient 0.6417406 for the level `Aspect5` is the difference in expected log count between `Aspect5` and the baseline `Aspect1`; then, $\exp(0.6417406) = 1.899785$ is the multiplicative effect on expected count of `Aspect5` over and above the baseline `Aspect1`. The corresponding values for `Aspect7` are 0.2338650 and 1.263474.

The output shows values for *null deviance* and *residual deviance*. The **null deviance** tells us how well the response variable can be predicted by a model with only an intercept

term, i.e., $\eta_i = \beta_0$, and the fitted mean response is $\hat{\lambda}_i = \overline{Y}$ (the sample mean). See (7.24) for the null deviance calculation in the Poisson model. The **residual deviance** tells us how well the response variable can be predicted by a model with the intercept and p predictor variables; see the appendix for more details. When comparing different models for the same response, a lower residual deviance or a larger value of the difference (null deviance - fitted model residual deviance) will imply a better model.

We see whether the fitted model is adequate for the data by comparing the deviance from the fitted model with the deviance from the *saturated model*, a notion unique to GLIM. In the saturated model, the predicted values are identical to the observed responses; see the appendix for more details. If this chi-squared test statistic with $n - p$ d.f. is significant, it would indicate that the data do not fit the model well - perhaps due to omitted predictors, violation of the log-linearity assumption, or the issue of over-dispersion (discussed later in this section).

```
> with(carcar.pf, cbind(deviance = deviance, df = df.residual,
+                 p = pchisq(deviance, df.residual, lower.tail=FALSE)))
     deviance  df            p
[1,] 445.7201 135 7.276196e-35
```

We can also fit the null model (with intercept only) using the code below.

```
carcar.pn <- glm(Carcar~1,
                 family='poisson', data=snails)
summary(carcar.pn)
```

```
Call:
glm(formula = Carcar ~ 1, family = "poisson", data = snails)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.2753  -1.7071  -0.7383   0.9025   8.0908

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.21256    0.02692   82.19   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 717.45  on 150  degrees of freedom
Residual deviance: 717.45  on 150  degrees of freedom
AIC: 1289.5

Number of Fisher Scoring iterations: 5
```

The output above shows the coefficient for the intercept with its SE, z-statistic and p-value. Note that the null deviance and residual deviance have the same value with the same 150 d.f. Note that the null deviance is the same value as we saw in the full model above.

We can compare the full model residual deviance with the null deviance in order to see whether all the predictors taken together are useful for explaining log counts. Here, we are

comparing the null model which is nested within the full model. Let

$$\begin{aligned}\text{Extra deviance} &= \text{null deviance - fitted model residual deviance},\\ \text{Extra d.f.} &= \text{d.f.(null deviance) - d.f.(fitted model residual deviance)},\\ \text{Extra deviance/Extra d.f.} &\sim \chi^2_{\text{Extra d.f.}}\end{aligned}$$

```
with(carcar.pf, cbind(deviance = null.deviance-deviance,
                      df = df.null-df.residual,
                      p = pchisq(null.deviance-deviance,
                      df.null-df.residual,
                      lower.tail=FALSE)))
```

The output below shows the Extra deviance, Extra d.f. and p-value from the test. The very small p-value shows that the data prefers the full model to the null model.

```
     deviance df           p
[1,] 271.7329 15 4.056617e-49
```

We can obtain the same result using the anova() function; we get the same p-value.

```
> an <- anova(carcar.pn, carcar.pf, test="Chisq")
> an$`Pr(>Chi)`
```

We can also compare the Akaike Information Criterion (AIC) from both models, the better model being the one which gives the smaller AIC.

```
> AIC(carcar.pn, carcar.pf)
          df      AIC
carcar.pn  1 1289.482
carcar.pf 16 1047.749
```

Finally, we compute the dispersion parameter from the fitted full model as the residual deviance divided by its d.f. If the dispersion parameter is close to 1, then the Poisson regression is appropriate for the data. If the dispersion parameter estimate exceeds one, then the data is said to be overdispersed, and we seek models that can better fit overdispersed data.

```
(disp.est <- carcar.pf$deviance/carcar.pf$df.residual)
[1] 3.30163
```

Remark 1. Suppose we wish to assign level 8 of `Aspect` as the reference level, instead of level 1. We may wish to do this for interpretation in the application domain. It should not alter the numerical stability or statistical understanding of the results. We can make this reference level change using the relevel() function, and see that there is a coefficient for `Aspect1` below, and none for `Aspect8`.

```
>snails$Aspect <- relevel(snails$Aspect, ref="8")
# Fit the Poisson loglinear model with "8" as baseline  level for Aspect
>carcar.Asp8baseline <- glm(Carcar~Elevation+Slope+Aspect+Soil+CC+LC+PA.sp+PA.other,
 family='poisson',data=snails)
>summary(carcar.Asp8baseline)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.8007228  1.0400779   1.731 0.083392 .
Elevation   -0.0006231  0.0026609  -0.234 0.814847
Slope        0.0128226  0.0035358   3.626 0.000287 ***
Aspect1      0.0030597  0.0955173   0.032 0.974446
Aspect5      0.6448003  0.0838275   7.692 1.45e-14 ***
Aspect7      0.2369247  0.0693733   3.415 0.000637 ***
Soil4        0.4990691  0.0702176   7.107 1.18e-12 ***
Soil6        1.0632136  0.1557836   6.825 8.80e-12 ***
CC2         -0.2814573  0.0969687  -2.903 0.003701 **
CC3         -0.2943464  0.0756138  -3.893 9.91e-05 ***
LC2          0.4036784  0.1074070   3.758 0.000171 ***
LC3          0.3750953  0.1102054   3.404 0.000665 ***
LC4         -0.0122835  0.1205093  -0.102 0.918812
LC5         -0.1519346  0.2053417  -0.740 0.459354
PA.sp       -0.0090556  0.0045251  -2.001 0.045371 *
PA.other     0.0011689  0.0014541   0.804 0.421490
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 717.45  on 150  degrees of freedom
Residual deviance: 445.72  on 135  degrees of freedom
AIC: 1047.7

Number of Fisher Scoring iterations: 5
```

Overdispersion

If we fit a Poisson regression to overdispersed responses (where the variance exceeds the mean), we may not get an adequate fit. Although the estimate of $\boldsymbol{\beta}$ will still be approximately unbiased, the standard errors of the coefficients are likely to be smaller than they should be, leading to larger t-statistics. There are a few simple ways to check for overdispersion.

1. We can compare the sample variances to the sample averages for groups of responses with identical predictor values.

2. We can examine the deviance goodness of fit test after fitting a rich model with several predictors to the data.

3. We can examine the deviance residuals to see if outliers may lead to a large deviance statistic.

7.3.2 Quasi-Poisson Loglinear Model

We can fit the quasi-Poisson model to the overdispersed `Carcar` counts using the code shown below. Parameter estimation is done by maximizing the quasi-likelihood function.

```
carcar.qpf <- glm(Carcar~Elevation+Slope+Aspect+Soil+CC+LC+PA.sp+PA.other,
 family=quasipoisson,data=snails)
```

```
summary(carcar.qpf)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.8037825  1.8976492   0.951 0.343539
Elevation   -0.0006231  0.0048589  -0.128 0.898147
Slope        0.0128226  0.0064565   1.986 0.049059 *
Aspect5      0.6417406  0.2014541   3.186 0.001795 **
Aspect7      0.2338650  0.1791621   1.305 0.194002
Aspect8     -0.0030597  0.1744177  -0.018 0.986030
Soil4        0.4990691  0.1282196   3.892 0.000155 ***
Soil6        1.0632136  0.2844659   3.738 0.000273 ***
CC2         -0.2814573  0.1770680  -1.590 0.114277
CC3         -0.2943464  0.1380732  -2.132 0.034833 *
LC2          0.4036784  0.1961287   2.058 0.041491 *
LC3          0.3750953  0.2012387   1.864 0.064503 .
LC4         -0.0122835  0.2200539  -0.056 0.955567
LC5         -0.1519346  0.3749606  -0.405 0.685971
PA.sp       -0.0090556  0.0082629  -1.096 0.275061
PA.other     0.0011689  0.0026552   0.440 0.660489
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 3.334395)

    Null deviance: 717.45  on 150  degrees of freedom
Residual deviance: 445.72  on 135  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

The dispersion parameter is estimated to be 3.334395, which is numerically very close to the value we estimated above as the residual deviance divided by the d.f. from the full Poisson model. We show below that the quasi-Poisson model with all predictors fits the data much better than the null model.

```
carcar.qpn <- glm(Carcar~1,
                  family=quasipoisson,data=snails)
summary(carcar.qpn)
an.qp<- anova(carcar.qpn, carcar.qpf, test="Chisq")
> an.qp$‘Pr(>Chi)‘
[1]           NA 3.719877e-11
```

7.3.3 Negative Binomial Regression

To address overdispersion, we can consider a negative binomial sampling distribution instead of the Poisson distribution (since unlike the Poisson case, the variance exceeds the mean for a negative binomial random variable). The and null negative binomial regression model fit is shown below.

```
library(MASS)
carcar.nbn <- glm.nb(Carcar~1, data = snails)
summary(carcar.nbn)
```

```
Call:
glm.nb(formula = Carcar ~ 1, data = snails, init.theta = 2.433887816,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7550  -0.8566  -0.3504   0.3991   2.9856

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   2.2126     0.0587   37.69   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(2.4339) family taken to be 1)

    Null deviance: 159.89  on 150  degrees of freedom
Residual deviance: 159.89  on 150  degrees of freedom
AIC: 952.72

Number of Fisher Scoring iterations: 1
              Theta:  2.434
          Std. Err.:  0.351

 2 x log-likelihood:  -948.715
```

The code and output below correspond to the full negative binomial regression model.

```
carcar.nbf <-
    glm.nb(Carcar~Elevation+Slope+Aspect+Soil+CC+LC+PA.sp+PA.other,
    data=snails)
summary(carcar.nbf)
```

```
Call:
glm.nb(formula = Carcar ~ Elevation + Slope + Aspect + Soil +
    CC + LC + PA.sp + PA.other, data = snails, init.theta = 4.472787009,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7747  -0.8639  -0.1554   0.5063   2.1908

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.0140986  1.7940177   1.123 0.261576
Elevation   -0.0012500  0.0045842  -0.273 0.785105
Slope        0.0117504  0.0063166   1.860 0.062851 .
Aspect5      0.6106842  0.1983958   3.078 0.002083 **
Aspect7      0.2252464  0.1671430   1.348 0.177778
Aspect8      0.0025504  0.1595054   0.016 0.987243
Soil4        0.4937690  0.1300269   3.797 0.000146 ***
Soil6        1.1118251  0.3266687   3.404 0.000665 ***
CC2         -0.1934611  0.1700038  -1.138 0.255128
CC3         -0.2145249  0.1347576  -1.592 0.111400
LC2          0.3999864  0.1725364   2.318 0.020434 *
LC3          0.3622314  0.1774664   2.041 0.041238 *
LC4          0.0204368  0.1911156   0.107 0.914841
LC5         -0.2019140  0.3025747  -0.667 0.504568
PA.sp       -0.0033830  0.0081395  -0.416 0.677686
PA.other    -0.0004075  0.0026892  -0.152 0.879552
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(4.4728) family taken to be 1)

    Null deviance: 241.35  on 150  degrees of freedom
Residual deviance: 160.93  on 135  degrees of freedom
AIC: 919.61

Number of Fisher Scoring iterations: 1
              Theta:  4.473
          Std. Err.:  0.811

 2 x log-likelihood:  -885.605
```

Each coefficient and its significance is interpreted in the same way as we showed for the Poisson regression. The dispersion parameter θ is estimated as 4.473 with a SE of 0.811, which is much bigger than 1 (the nominal dispersion under a Poisson regression). Note that this value is different from the dispersion parameter we estimated from the full Poisson regression, although both indicate overdispersion. The predictors are useful for explaining the counts as the very small p-value below shows.

```
> an.nb <- anova(carcar.nbn, carcar.nbf, test="Chisq")
> an.nb$‘Pr(Chi)‘
[1]           NA 7.306626e-08
```

We explore this model fit in more detail. We can fit a reduced model by dropping the predictors `Elevation`, `CC`, `PA.sp`, and `PA.other` from the full model, as shown below.

```
Call:
glm.nb(formula = Carcar ~ Slope + Aspect + Soil + LC, data = snails,
    init.theta = 4.260223364, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7957  -0.8499  -0.2145   0.5141   2.4233

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.323821   0.253884   5.214 1.85e-07 ***
Slope        0.011203   0.005887   1.903 0.057028 .
Aspect5      0.636035   0.192248   3.308 0.000938 ***
Aspect7      0.257699   0.166679   1.546 0.122085
Aspect8      0.030143   0.158078   0.191 0.848774
Soil4        0.471368   0.123072   3.830 0.000128 ***
Soil6        1.185382   0.302415   3.920 8.87e-05 ***
LC2          0.414730   0.173663   2.388 0.016934 *
LC3          0.427884   0.176237   2.428 0.015187 *
LC4          0.065769   0.187854   0.350 0.726257
LC5         -0.249444   0.303676  -0.821 0.411409
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(4.2602) family taken to be 1)

    Null deviance: 234.12  on 150  degrees of freedom
Residual deviance: 159.68  on 140  degrees of freedom
AIC: 912.99

Number of Fisher Scoring iterations: 1
              Theta:  4.260
          Std. Err.:  0.751

 2 x log-likelihood:  -888.989
```

The estimate of θ is slightly different from the value estimated from the full model. We will see which of the two models is preferred by the data. We can compare the residual deviance between the full model and the reduced model (which is a subset of the full model) via a deviance difference test (also called the drop in deviance test) to see which model is better supported by the data. The model postulated by the null hypothesis is the reduced model. In the code below, the positive-valued deviance difference test statistic is the difference between the residual deviance from the reduced model and the residual deviance from the full model. If H_0 is true, the deviance difference statistic has a chi-squared distribution whose degrees of freedom are the number of parameters we test (or the difference between the number of coefficients in the full model and the number of coefficients in the reduced model). The test statistic value is 3.383563 with a p-value of 0.6410745, so we do not have enough evidence to reject H_0. In other words, the data supports the reduced model.

```
> (comp.nb <- anova(carcar.nbf, carcar.nbr, test="Chisq"))
Likelihood ratio tests of Negative Binomial Models

Response: Carcar
                                                                    Model    theta Resid. df    2 x log-lik.
1                                       Slope + Aspect + Soil + LC 4.260223       140       -888.9887
2 Elevation + Slope + Aspect + Soil + CC + LC + PA.sp + PA.other 4.472787       135       -885.6052
    Test    df LR stat.   Pr(Chi)
1
2 1 vs 2     5 3.383563 0.6410745
> comp.nb$`Pr(Chi)`
[1]        NA 0.6410745
```

From the output for the reduced model, we can assess whether a coefficient β_j is significant by testing $H_0 : \beta_j = 0$ versus $H_1 : \beta_j \neq 0$. For large n, the Wald z-statistic

$$z = \hat{\beta}_j / \mathrm{SE}(\hat{\beta}_j) \tag{7.8}$$

has a N(0,1) distribution provided H_0 is true. If $|z| > z_{\alpha/2}$, we conclude that the jth predictor has a significant effect on the response. Alternately, we can check whether the p-value is smaller than the chosen level of significance $\alpha = 0.05$. From the output we see that `Aspect7`, `Aspect8`, `LC4` and `LC5` are not significant for explaining `Carcar` counts.

We can also construct $100(1-\alpha)\%$ Wald C.I. estimates for β_j as

$$\hat{\beta}_j \pm z_{\alpha/2}\mathrm{SE}(\hat{\beta}_j), \tag{7.9}$$

and conclude that a predictor is useful for explaining `carcar` counts if the C.I. does not include the value 0. For example, the 95% C.I. for the coefficient corresponding to `Aspect5` is

$$0.636035 \pm (1.96)(0.192248) = (0.2592289, 1.012841),$$

which does not include zero, hence `Slope` is a significant predictor.

Also, for $i = 1, \ldots, n$, using the estimates $\hat{\beta}_j,\ j = 0, \ldots, p$, we can compute

$$\hat{\eta}_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j X_{ij},$$

and then

$$\hat{\lambda}_i = \exp(\hat{\eta}_i).$$

The $\hat{\lambda}_i$ are the fitted mean responses from the full model, which we can recover as follows.

```
> carcar.nbr.fit <- fitted(carcar.nbr)
> summary(carcar.nbr.fit)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.510   6.624   7.901   9.126  10.904  26.410
```

The summaries of the fitted values are reasonably close to the summaries of the observed `Carcar` counts.

```
> summary(snails$Carcar)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   4.500   7.000   9.139  12.000  43.000
```

For residual diagnostics, we can use deviance residuals (see (7.21) for a definition) from the full model, which are obtained using the code below.

```
> carcar.nbr.res <- resid(carcar.nbr, type = "deviance")
s> ummary(carcar.nbr.res)
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.7957 -0.8500 -0.2144 -0.1721  0.5141  2.4233
```

Remark 1. A normal Q-Q plot of these residuals can be constructed using the qqnorm() function. We can also view a plot of these residuals versus the fitted values.

Remark 2. If we set type = "pearson", we can recover Pearson residuals (see the appendix for a definition).

7.4 Binary Response Models

We assume that a binary response Y_i assumes either the value 1 (usually denoting 'Success'), or the value 0 (usually denoting 'Failure'); Y_i follows a Bernoulli(π_i) distribution with

$$P(Y_i = 1) = \pi_i = 1 - P(Y_i = 0),\ E(Y_i) = \pi_i,$$

and p.m.f. in (7.2). The mean response is $E(Y_i) = \pi_i$, which lies in the interval $[0, 1]$, while $\text{Var}(Y_i) = \pi_i(1-\pi_i)$. Our goal is to model the binary responses as functions of p independent variables denoted by $X_{i,j},\ j = 1, \ldots, p$ for each i.

7.4.1 Logit regression

The binary logit (or, logistic regression) model is a generalized linear model (GLIM) for explaining binary responses Y_i:

$$Y_i|\pi_i \sim \text{Bernoulli}(\pi_i). \tag{7.10}$$

In (7.10), the logit link function relates the ith mean response π_i to the regression fit η_i:

$$\begin{aligned} \text{logit}(\pi_i|\mathbf{x}_i) &= \eta_i, \text{ where,} \\ \eta_i &= \beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j} = \mathbf{x}_i'\boldsymbol{\beta}, \end{aligned} \tag{7.11}$$

with $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$, and $\mathbf{x}_i = (1, X_{i,1}, \ldots, X_{i,p})'$. Since the mean response $0 < \pi_i < 1$, whereas η_i is real-valued, we need a function such as the logit function to link the two in a correct way. By inverting the logit link function, we can also write the binary logit model as

$$\pi_i = P(Y_i = 1|\mathbf{x}_i) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j})}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j})}. \tag{7.12}$$

Example 7.4.1 (BANK) The BANK data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls, and

BANK

more than one contact to the same client was often required in order to access whether the product (bank term deposit) would be (or not) subscribed. Data details are described in Moro et al. (2011); see `http://hdl.handle.net/1822/14838`. The data consists of information on $n = 2466$ clients of whom $n_1 = 521$ subscribed to a term deposit (with `y`=1), while the remaining $n_0 = 1945$ did not (`y`=0). The data set also has 15 potential predictor variables: age, job (categorical with 12 levels), marital status (categorical with 3 levels), education (categorical with 4 levels), default (credit default, binary), average yearly balance (in euros), has a housing loan (yes,no), has personal loan (yes,no), contact communication type (categorical with 3 levels), last contact day of the month, last contact month of year (categorical with 12 levels) last contact duration (in seconds), number of contacts performed during this campaign and for this client, number of days that passed by after the client was last contacted from a previous campaign (-1 means the client was not previously contacted), number of contacts performed before this campaign and for this client, and outcome of the previous marketing campaign (categorical with 4 levels).

Research question #2

Are these variables useful for predicting whether the client will subscribe a term deposit?

We load the data into a data frame called `bank.data`. The binary response is the variable `y`.

```
bank.data <- read.csv("Data/Bankdata.csv")
str(bank.data)
# Check the relative frequency of client's subscription of term-deposit
table(bank.data$y)
# Change y to a factor variable
bank.data$y <- as.factor(ifelse(bank.data$y=="yes",1,0))
```

We convert the response and character predictors to factors.

```
bank.data$y <- as.factor(ifelse(bank.data$y=="yes",1,0))
# Look up the classes of each column. and determine indexes of character
    columns
col_class <- sapply(1:ncol(bank.data), function(x) class(bank.data[,x]))
col_id <- which(col_class == "character")
# Change character columns into factors
for(i in 1:length(col_id)){
  bank.data[,col_id[i]] <- as.factor(bank.data[,col_id[i]])
}
```

As we did in Chapter 6, we do an 80-20 train-test split of the data. We will train the logit regression model on the train data (consisting of 1972 cases) and use the test data (with 494 cases) to check accuracy. It is generally useful to see whether the train and test data reflect the structure of the full data in terms of distributions of the variables. We show an alternate way to the random splitting we saw in earlier chapters:

```
set.seed (123457)
train.prop <- 0.80
strats <- bank.data$y
```

```
rr <- split(1:length(strats), strats)
idx <- sort(as.numeric(unlist(sapply(rr, function(x) sample(x,
    length(x)*train.prop )))))
bank.data.train <- bank.data[idx, ]
bank.data.test <- bank.data[-idx, ]
```

We can see whether the proportions of the two levels of the response `y` are the same in the train, test, and the entire data.

```
> summary(bank.data.train$y)/nrow(bank.data.train)
        0         1
0.7890467 0.2109533
> summary(bank.data.test$y)/nrow(bank.data.test)
       0         1
0.7874494 0.2125506
> summary(bank.data$y)/nrow(bank.data)
        0         1
0.7887267 0.2112733
```

We can use the glm() function to fit a binary regression with a logit link (see (7.10)) to the training data in `bank.data.train`. The response is `y` and the model includes all the predictors; we can call it the *full model*.

Recall that the output provides the null deviance (tells how well the response variable can be predicted by a model with only an intercept term) and the residual deviance (tells how well the response variable can be predicted by a model with the intercept and the selected predictor variables). The larger the difference between the null deviance and residual deviance, better the model fit.

```
full.logit <- glm(y ~ . ,data = bank.data.train, family = binomial(link =
    "logit"))
summary(full.logit)
```

```
Call:
glm(formula = y ~ ., family = binomial(link = "logit"), data = bank.data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9998  -0.4571  -0.2844  -0.1304   2.7544

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.946e+00  7.575e-01  -2.569 0.010186 *
age                 8.228e-04  9.221e-03   0.089 0.928899
jobblue-collar     -4.228e-01  3.169e-01  -1.334 0.182184
jobentrepreneur     1.556e-01  5.135e-01   0.303 0.761839
jobhousemaid       -1.064e-01  5.000e-01  -0.213 0.831499
jobmanagement       1.901e-01  3.159e-01   0.602 0.547398
jobretired          5.909e-01  4.098e-01   1.442 0.149279
jobself-employed    1.106e-01  4.677e-01   0.237 0.813020
jobservices        -2.385e-01  3.539e-01  -0.674 0.500460
jobstudent          2.405e-01  4.983e-01   0.483 0.629303
jobtechnician       1.565e-01  3.027e-01   0.517 0.605092
jobunemployed      -6.981e-01  5.434e-01  -1.285 0.198887
jobunknown          5.856e-01  7.231e-01   0.810 0.418069
maritalmarried     -2.362e-01  2.329e-01  -1.014 0.310503
maritalsingle       1.743e-01  2.668e-01   0.653 0.513470
educationsecondary  8.574e-02  2.556e-01   0.335 0.737318
educationtertiary   1.705e-01  3.003e-01   0.568 0.570299
educationunknown   -3.347e-01  4.626e-01  -0.723 0.469404
defaultyes          5.069e-01  5.856e-01   0.866 0.386712
balance            -2.945e-06  2.055e-05  -0.143 0.886045
housingyes         -2.882e-01  1.750e-01  -1.647 0.099561 .
loanyes            -7.053e-01  2.681e-01  -2.631 0.008517 **
contacttelephone   -3.365e-01  3.082e-01  -1.092 0.274898
contactunknown     -1.354e+00  2.751e-01  -4.923 8.54e-07 ***
monthaug           -8.488e-01  3.212e-01  -2.642 0.008232 **
monthdec           -2.309e-01  9.079e-01  -0.254 0.799270
monthfeb           -1.038e-01  3.586e-01  -0.289 0.772328
monthjan           -1.409e+00  4.843e-01  -2.910 0.003618 **
monthjul           -8.047e-01  3.224e-01  -2.496 0.012570 *
monthjun            2.867e-01  3.620e-01   0.792 0.428318
monthmar            1.953e+00  5.296e-01   3.687 0.000227 ***
monthmay           -8.631e-01  2.989e-01  -2.888 0.003880 **
monthnov           -1.141e+00  3.578e-01  -3.188 0.001433 **
monthoct            1.524e+00  4.395e-01   3.469 0.000522 ***
monthsep            1.471e-01  5.523e-01   0.266 0.789908
duration            5.072e-03  3.008e-04  16.861  < 2e-16 ***
campaign           -1.044e-01  3.541e-02  -2.949 0.003189 **
pdays              -4.486e-04  1.314e-03  -0.341 0.732800
previous            2.458e-02  4.937e-02   0.498 0.618493
poutcomeother       4.880e-01  3.459e-01   1.411 0.158325
poutcomesuccess     2.863e+00  4.092e-01   6.996 2.63e-12 ***
poutcomeunknown    -2.521e-01  4.176e-01  -0.604 0.546040
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2032.0  on 1971  degrees of freedom
Residual deviance: 1228.2  on 1930  degrees of freedom
AIC: 1312.2

Number of Fisher Scoring iterations: 6
```

We see from the results that 12 variables are significant for explaining incidence of `yes` to subscribing to a deposit. The *null model* is a model with the intercept only, as shown below.

```
null.logit <- glm(y~1, data = bank.data.train, family = binomial(link =
    "logit"))
summary(null.logit)
```

```
Call:
glm(formula = y ~ 1, family = binomial(link = "logit"), data = bank.data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6884  -0.6884  -0.6884  -0.6884   1.7642

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.3192     0.0552   -23.9   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2032  on 1971  degrees of freedom
Residual deviance: 2032  on 1971  degrees of freedom
AIC: 2034

Number of Fisher Scoring iterations: 4
```

The following steps show how we can use both backward and forward selection to choose predictors that best explain the response `y`, using the option direction ="both" in the step() function.

```
both.logit <- step(null.logit, list(lower=formula(null.logit),upper=formula(full.logit)),
                   direction="both",trace=0, data = bank.data.train)
formula(both.logit)
y ~ duration + poutcome + month + contact + campaign + loan +
    housing + marital
```

The detailed output follows.

```
Call:
glm(formula = y ~ duration + poutcome + month + contact + campaign +
    loan + housing + marital, family = binomial(link = "logit"),
    data = bank.data.train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9641  -0.4598  -0.2957  -0.1367   2.7060

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -1.7924853  0.3775685  -4.747 2.06e-06 ***
duration          0.0049609  0.0002915  17.019  < 2e-16 ***
poutcomeother     0.5438411  0.3340242   1.628  0.10349
poutcomesuccess   2.8328691  0.3936268   7.197 6.16e-13 ***
poutcomeunknown  -0.2355325  0.2306337  -1.021  0.30714
monthaug         -0.6504756  0.3114308  -2.089  0.03674 *
monthdec         -0.0720328  0.8845766  -0.081  0.93510
monthfeb         -0.0524152  0.3496039  -0.150  0.88082
monthjan         -1.3317116  0.4718827  -2.822  0.00477 **
monthjul         -0.7362716  0.3166115  -2.325  0.02005 *
monthjun          0.3461652  0.3563932   0.971  0.33140
monthmar          2.1416515  0.5212080   4.109 3.97e-05 ***
monthmay         -0.8079778  0.2911019  -2.776  0.00551 **
monthnov         -1.0305560  0.3492869  -2.950  0.00317 **
monthoct          1.6921774  0.4307629   3.928 8.55e-05 ***
monthsep          0.2913004  0.5428927   0.537  0.59156
contacttelephone -0.2759768  0.2909213  -0.949  0.34281
contactunknown   -1.3902240  0.2700373  -5.148 2.63e-07 ***
campaign         -0.1044155  0.0347161  -3.008  0.00263 **
loanyes          -0.6994935  0.2644761  -2.645  0.00817 **
housingyes       -0.3990506  0.1661952  -2.401  0.01635 *
maritalmarried   -0.2583238  0.2277553  -1.134  0.25670
maritalsingle     0.1662755  0.2445333   0.680  0.49652
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2032.0  on 1971  degrees of freedom
Residual deviance: 1248.3  on 1949  degrees of freedom
AIC: 1294.3

Number of Fisher Scoring iterations: 6
```

is suppressed, but the residual deviance is shown below.

```
> both.logit$deviance
[1] 1248.251
```

We can assess how well the models full.logit and both.logit fit the response from the test data.

```
pred.both <- predict(both.logit, newdata = bank.data.test, type="response")
pred.full <- predict(full.logit, newdata = bank.data.test, type="response")
```

We can compute and compare the confusion matrices using the code below.

```
> (table.both <- table(pred.both > 0.5, bank.data.test$y))

         0   1
  FALSE 368  54
  TRUE   21  51
> (table.full <- table(pred.full > 0.5, bank.data.test$y))

         0   1
  FALSE 369  55
  TRUE   20  50
```

We can then compute prediction accuracy for the test data as percentages.

```
> (accuracy.both <- round((sum(diag(table.both))/sum(table.both))*100,2))
[1] 84.82
> (accuracy.full <- round((sum(diag(table.full))/sum(table.full))*100,2))
[1] 84.82
```

Another useful metric is area under the receiver operating characteristics (ROC) curve. We see that the area under the curve (AUC) is very similar for the test data under the two model fits. We can also use the *ROCR* package for this purpose (not shown).

```
> library(pROC)
> roc.both <- roc(bank.data.test$y, pred.both, levels=c(1,0))
Setting direction: controls > cases
> auc(bank.data.test$y, pred.both)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Area under the curve: 0.8811
> roc.full <- roc(bank.data.test$y, pred.full, levels=c(1,0))
Setting direction: controls > cases
> auc(bank.data.test$y, pred.full)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
Area under the curve: 0.8886
```

The confusionMatrix() function in the package *caret* is also useful for looking at several criteria for assessing the predictions, including the ones we showed above. We show the code below, but not the detailed output.

```
library(caret)
b <- ifelse(pred.both > 0.5,1,0)
cm.both <- confusionMatrix(reference=as.factor(bank.data.test$y),
    data=as.factor(b),
                           mode="everything")
f <- ifelse(pred.full > 0.5,1,0)
cm.full <- confusionMatrix(reference=as.factor(bank.data.test$y),
    data=as.factor(f),
                           mode="everything")
```

Remark 1. Using code similar to what we showed for the test data, we can also predict the train data and assess accuracy under both models. We see that the two models give similar performance. The accuracy on both.logit and full.logit are 86.21% and 86.05% respectively, and their respective AUC's are 0.908 and 0.9099.

Remark 2. Instead of stepwise selection of predictors, the code for variable selection using only backward elimination is shown below.

```
backwards <- step(full.logit, trace = 0)  # suppress details of each
    iteration
# backwards <- step(full.logit)  # to show all details
formula(backwards)
summary(backwards)
```

Remark 3. The code for variable selection using only forward selection is shown below.

```
forwards = step(null.logit, trace=0,
      scope=list(lower=formula(null.logit),upper=formula(full.logit)),
    direction="forward")
formula(forwards)
summary(forwards)
```

Remark 4. Using the code beow, we can fit a model which also includes second-order interactions between all the predictors, carry out stepwise selection of variables, then predict from the best model.

```
variables <- c("job","campaign","duration","contact")
interaction.var <- combn(variables,2,FUN = function(x) paste(x,collapse=":"))
formula.inter <- as.formula(paste("y ~ . +",paste(interaction.var,collapse =
    "+")))
all.logit <- glm(formula = formula.inter , family = binomial(link = "logit")
                 ,data = bank.data.train)
```

Remark 5. Starting with the standard normal c.d.f $\Phi(z)$ which lies in the interval $[0, 1]$, the probit (or inverse normal c.d.f.) link assumes that $\Phi^{-1}(\pi_i) = \eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, so that

$$\pi_i = \Phi(\mathbf{x}_i'\boldsymbol{\beta})$$

The Fisher scoring algorithm is again used to obtain MLE's of the regression coefficients. The basic code for the full model is shown below.

```
full.probit <- glm(y ~ . ,data = bank.data.train, family = binomial(link =
    "probit"))
summary(full.probit)
```

Appendix Chapter 7: Statistical Details

Components of a GLIM

Let $Y_1, Y_2, \ldots, Y_n$ be independent response variables, each associated with the same number of known predictors $X_{i1}, \ldots, X_{ip}$. Let $\mathbf{y} = (Y_1, Y_2, \ldots, Y_n)'$ and $\mathbf{x}_i = (1, X_{i1}, \ldots, X_{ip})'$, $i = 1, \ldots, n$. A GLIM for the data $(Y_i, X_{i1}, \ldots, X_{ip})$, $i = 1, \ldots, n$, is specified by the following three components:

1. **Random component.** Each Y_i has a p.m.f. or p.d.f. from an exponential family of distributions, taking the form (7.13). Let $\mu_i = E(Y_i)$.

   A random varibale Y belongs to the **exponential family of distributions** if its probability (density or mass) function is

   $$f(y; \theta, \phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right\}, \tag{7.13}$$

   where both θ and $\phi > 0$ are scalar parameters and a, b, and c are known functions. We assume that $a(\phi)=\phi/m$ where m is a known weight (often $m_i = 1$ for all observations $i = 1, \ldots, n$) and $c = c(y, \phi/m)$, and ϕ may be either known or unknown. In the models we will consider, a, b, and c will determine a particular family in the class, such as the normal, Bernoulli, binomial, Poisson, negative binomial, etc.

2. **Systematic component.** Consider a vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$, where the β_j's are unknown coefficients. The systemaic component is $\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}$.

3. **Link function.** The link function $g(\mu_i)$ is a continuous, one-to-one function such that $g(\mu_i) = \eta_i$. The **link function** $g(.)$ provides the relationship between the linear predictor η_i and the mean response μ_i in a GLIM. The choice of the link function is based on matching the domain of the link function to the range of the mean response. The table below shows commonly used link functions in GLIMs.

Distribution/ Support	Link name	Link function	Mean function
Normal real line: $(-\infty, \infty)$	identity	$\eta = \mu$	$\mu = \eta$
Poisson integers: $0, 1, 2, \ldots$	log	$\eta = \log(\mu)$	$\mu = e^{\eta}$
Binomial integers: $0, 1, \ldots, m$	logit	$\eta = \log(\frac{\mu}{1-\mu})$	$\mu = \frac{e^{\eta}}{1+e^{\eta}}$
	probit	$\eta = \Phi^{-1}(\mu)$	$\mu = \Phi(\eta)$

Key formulas

Fisher scoring algorithm

The iterative Fisher scoring (FS) algorithm is a variant of the Newton-Raphson algorithm and is used for obtaining MLE's of the regression parameters in a GLIM. Each iteration of the FS algorithm has the form

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} + [\mathbf{I}(\boldsymbol{\beta}^{(m)})]^{-1}\mathbf{S}(\boldsymbol{\beta}^{(m)}), \tag{7.14}$$

where $\mathbf{S}(\boldsymbol{\beta}^{(m)})$ and $\mathbf{I}(\boldsymbol{\beta}^{(m)}$ are respectively the score function (vector) and the Fisher information (matrix) evaluated at the mth iterate $\boldsymbol{\beta}^{(m)}$. The converged values are the MLE's of the regression coefficients $\beta_j,\ j = 0, 1, \ldots, p$. Provided n is sufficiently large, the MLE β_j has a normal distribution with mean β_j and variance obtained from the (j, j)th diagonal element of the inverse of the Fisher information matrix.

Goodness of fit in GLIMs

In the GLIM, it is useful to think about three model classes.

1. **Null (or constant) model**. Assume that $\mu_i = \mu_0\ i = 1, \ldots, n$, a constant, so that $E(Y_i)$ does not depend on any predictors. Then, $\widehat{\mu}_i = \widehat{\mu}_0 = \overline{Y}$, the sample mean of the responses.

2. **Fitted model with p predictors**. Suppose $g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$, where $\mathbf{x}_i$ is a $(p + 1)$-dimensional vector, the first element being 1. Then, $\widehat{\mu}_i = g^{-1}(\mathbf{x}_i'\widehat{\boldsymbol{\beta}}_{ML})$.

3. **Saturated model**. Assume that $\boldsymbol{\mu}$ has n free parameters and is completely free of constraints. In this case, the log-likelihood is maximized at $\widehat{\boldsymbol{\mu}} = (\widehat{\mu}_1, \ldots, \widehat{\mu}_n)'$, where $\widehat{\mu}_i = Y_i$, i.e., $\widehat{\boldsymbol{\mu}} = \mathbf{y}$, the observed response vector.

Let $d_i,\ i = 1, \ldots, n$ denote the discrepancy of fit between the saturated model and the model with p predictors. For the log-linear Poisson model, this discrepancy is

$$d_i = 2Y_i \log(Y_i/\widehat{\lambda}_i). \tag{7.15}$$

while, for the binary response model with logit link, it is

$$d_i = 2[Y_i \log(Y_i/\widehat{\pi}_i) + (1 - Y_i) \log[(1 - Y_i)/(1 - \widehat{\pi}_i)], \tag{7.16}$$

The **deviance** under the fitted model is defined as the following function of the data:

$$D(\mathbf{y}; \widehat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} d_i \tag{7.17}$$

It is straightforward to compute the deviance for different models. For example,

$$D(\mathbf{y}; \widehat{\boldsymbol{\mu}}) = \begin{cases} \sum_{i=1}^{n}(Y_i - \widehat{\mu}_i)^2 \text{ or } SSE & \text{(normal model)} \\ 2\sum_{i=1}^{n}[Y_i \log(Y_i/\widehat{\mu}_i) + (1 - Y_i) \log[(1 - Y_i)/(1 - \widehat{\mu}_i)] & \\ & \text{(binary logit model)} \\ 2\sum_{i=1}^{n} Y_i \log(Y_i/\widehat{\mu}_i) & \text{(Poisson log-linear model)} \end{cases}$$

Note that the deviance is a non-decreasing function of p (the number of predictors included in the model).

The **scaled deviance** is the deviance divided by the dispersion ϕ. When $\phi = 1$, the scaled deviance is the deviance. The scaled deviance has an asymptotic χ^2_{n-p} distribution.

Another criterion that enables us to assess discrepancy in fit is the generalized Pearson X^2 statistic defined by

$$X^2 = \sum_{i=1}^{n} \frac{(Y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)}. \tag{7.18}$$

Let $a_i(\phi) = \phi$ for $i = 1, \ldots, n$. The scaled generalized Pearson X^2 statistic is defined as

$$X^2_s = \frac{X^2}{\phi}. \tag{7.19}$$

For the normal model, the X^2 statistic is SSE, while for the binomial and Poisson models, it is the usual Pearson X^2 statistic. For the Poisson model, $\sum_{i=1}^{n} r^2_{i,P} = X^2$, Pearson's goodness of fit statistic, which leads to the construction of $r_{i,P}$.

Residuals in GLIMs

Recall that for the MLR models, the ith residual was defined as the difference between the observed response Y_i and the fitted response $\widehat{Y}_i$. We cannot obtain residuals in a similar way in GLIMs. Let $\widehat{\mu}_i$ be the mean value of Y_i determined by $g(\widehat{\mu}_i) = \widehat{\eta}_i = \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$. We define Pearson and deviance residuals.

For $i = 1, \ldots, n$, the ith Pearson residual is defined by

$$r_{i,P} = \frac{Y_i - \widehat{\mu}_i}{\sqrt{V(\widehat{\mu}_i)}}, \tag{7.20}$$

while the ith **deviance residual** is defined by

$$r_{i,D} = \text{sign}(Y_i - \widehat{\mu}_i)\sqrt{d_i}, \tag{7.21}$$

where d_i was defined above as the discrepancy of fit between the saturated model and the model with p predictors, and the **sign or signum** function is defined as

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x = 0 \end{cases} \tag{7.22}$$

For example, for the Poisson model,

$$r_{i,D} = \text{sign}(Y_i - \widehat{\mu}_i)[2(Y_i \log(Y_i/\widehat{\mu}_i) - Y_i + \widehat{\mu}_i)]^{1/2}.$$

The residual deviance for the Poisson model is

$$\text{Dev}_{\text{resid}} = 2\sum_{i=1}^{n} Y_i \log(Y_i/\hat{\lambda}_i) = 2\sum_{i=1}^{n} Y_i \log(Y_i/\exp(\hat{\eta}_i). \tag{7.23}$$

The null deviance for the Poisson model is

$$\text{Dev}_{\text{null}} = 2\sum_{i=1}^{n} Y_i \log(Y_i/\overline{Y}). \tag{7.24}$$

Likelihood Ratio Test (LRT)

We can use the LRT to compare two models, $\mathcal{M}_1 \subset \mathcal{M}_2$. For instance, $\mathcal{M}_2$ could be the "full model" with all p β coefficients, while $\mathcal{M}_1$ is a model reduced by a null hypothesis H_0 which tests whether some of the β coefficients are zero. Suppose the maximized likelihood under $\mathcal{M}_1$ is

$$\max_{\boldsymbol{\beta} \in \mathcal{M}_1} L(\boldsymbol{\beta}; \mathbf{y}) = L(\widehat{\boldsymbol{\beta}}_{1,ML}; \mathbf{y})$$

and the maximized likelihood under $\mathcal{M}_2$ is

$$\max_{\boldsymbol{\beta} \in \mathcal{M}_2} L(\boldsymbol{\beta}; \mathbf{y}) = L(\widehat{\boldsymbol{\beta}}_{2,ML}; \mathbf{y}).$$

The LRT statistic is

$$\Lambda = \frac{L(\widehat{\boldsymbol{\beta}}_{1,ML}; \mathbf{y})}{L(\widehat{\boldsymbol{\beta}}_{2,ML}; \mathbf{y})}. \tag{7.25}$$

Under the usual regularity conditions,

$$-2 \log \Lambda \sim \chi^2_\nu \text{ approx. under } H_0,$$

where $\nu = \dim(\mathcal{M}_2) - \dim(\mathcal{M}_1)$.

Drop-in-deviance test

The widely used drop-in-deviance test, also called the deviance difference test, is used to compare two nested GLIMs using the deviance functions from both models. This is similar to the use of the Extra SS F-test in the normal MLR model. As before, the bigger model with more predictors is called the *full model*, while a smaller model with only a subset of predictors from the full model is called the reduced model. The reduced model is said to be nested in the full model.

Suppose the *full model* has q coefficients, $\beta_1, \ldots, \beta_q$ corresponding to including q (out of the p) predictors. Let the *reduced model* have $m < q$ coefficients, by dropping $q - m$ predictors from the full model. Let $\widehat{\boldsymbol{\mu}}_F$ and $\widehat{\boldsymbol{\mu}}_R$ denote the fitted vectors under the full and reduced models respectively, obtained using the FS algorithm. The deviance difference statistic which allows us to test whether the data prefers the reduced model (null behavior) or the full model (alternative hypothesis) is

$$D(\mathbf{y}; \widehat{\boldsymbol{\mu}}_R) - D(\mathbf{y}; \widehat{\boldsymbol{\mu}}_F) \sim \chi^2_{q-m} \text{ approx. under } H_0. \tag{7.26}$$

8

More on GLIM and Related Methods

This chapter describes useful extensions of the generalized linear regression models that we described in Chapter 7. For binary responses, we describe methods such as classification and regression trees (CART), random forest (RF), and gradient boosting (GB). Then, we discuss models for categorical responses which have more than 2 levels; we describe multinomial logit (or generalized logit) models for nominal categorical variables and cumulative logit (or proportional odds) models for ordinal categorical variables.

8.1 Introduction

In Chapter 7, we discussed GLIM's for counts and binary-valued responses using loglinear and logistic regressions. Other approaches are possible, as we see below. Other more modern approaches for binary responses include tree based models including CART and random forest as well as gradient boosting methods. Many examples have categorical responses with more than two levels. In some situations, these levels are not ordered, and we can fit multinomial (or generalized) logits models to explain the response as a function of p predictors. In other examples, the levels of the categorical response are ordered, and we then fit cumulative logits (proportional odds) models for explaining the response.

8.2 Essential Terminology and Notation

✓ The **Gini index** is an impurity measure used in the context of decision trees for classifying a response with J categories. It is defined by

$$\text{Gini index} = 1 - \sum_{j=1}^{J} p_j^2, \tag{8.1}$$

where $p_j = P(Y \in \text{class} j)$, $j = 1, \ldots, J$. Gini index lies in $[0, 1]$; 0 denotes a pure classification where all the cases belong to a single class, while 1 indicates a random distribution of cases across the J classes. A Gini index of 0.5 shows an equal distribution of cases over some classes.

✓ **Entropy** is an alternate impurity measure defined by

$$\text{Entropy} = \sum_{j=1}^{J} -p_j \log_2(p_j). \tag{8.2}$$

Entropy also lies in $[0, 1]$.

8.3 Trees, Random Forests, and Gradient Boosting

Binary responses can be analyzed using methods beyond logistic regression. In this section, we discuss useful methods such as classification and regression trees (CART), random forest (RF) and gradient boosting (GB).

8.3.1 Methods Based on Regression Trees

While logit or probit regression can be a standard choice for 'classical' binary response modeling of low-dimensional datasets (where the number of predictors is small compared to the sample size), nonparametric methods based on regression (decision) trees such as the classification and regression trees (CART) (Breiman et al. (2017)), the random forest (RF) (Breiman (2001)), and boosting algorithms are being increasingly used in many domains for analyzing the effect of predictors on a binary response. Although regression trees are more computer intensive and may be less efficient than GLIM's, they are popular because they enable us to model complex datasets with minimal assumptions Hastie et al. (2009).

A decision tree algorithm, which is a supervised learning algorithm, is a top down "greedy" approach which partitions a dataset into smaller subsets. The top of the decision tree is known as the root node, which corresponds to the best predictor variable, called a feature in this setup. At each decision node, the predictors (features) are split into two branches, and this process of binary splitting is repeated until the leaf nodes are reached, which is used to make the final prediction. This branching is called *recursive partitioning*.

CART Algorithm

Classification and regression trees (CART) is a decision tree algorithm (Breiman et al. (2017)). We can use CART to predict a binary response Y as a function of predictors (features) $X_1, X_2, \ldots, X_p$. A decision tree is (really) an inverted tree which divides a data set into different classes, with the root node at the top. A CART model uses a greedy algorithm for selecting features (predictors) and split points (cut points or nodes) until a a predetermined termination criterion is reached and a suitable tree is constructed. A root node represents a single feature, say X_j, and a split point on that feature. A binary tree is constructed by repeatedly splitting a node into two child nodes. That is, at each internal node in the tree, the algorithm applies a test to one of the features, say X_j, and branches either to the left or to the right depending on the outcome; i.e., the split is made based on the predictor (feature) which results in the largest possible reduction in heterogeneity of the binary response variable. Eventually we come to a leaf node where we can stop growing the tree and decide to make a prediction for the response Y. Each terminal leaf node of the tree

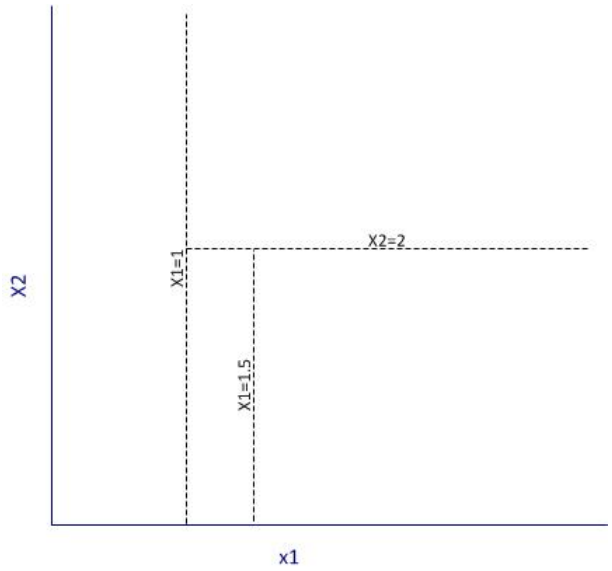

FIGURE 8.1: Example to show how rectangular regions in CART partitioning.

represents a cell of the partition, and has a simple model attached to it. The splitting rules partition the dataset (decision region) into rectangular regions, each of which correspond to a split. For example, see Figure 8.1 (source: `https://eight2late.files.wordpress.com/2016/02/cart_partitioning.jpg`).

The tree can be shallow or deep, depending on how the stratification happens. Each node of the tree consists of a feature which is the cause for further splitting in the downward direction. Useful questions when we build the tree are: (i) how does the algorithm decide which feature should be at the root node, (ii) which feature(s) must be at the internal or leaf nodes, and (iii) how can the accuracy of splitting the tree be measured, and when should the algorithm stop based on a predefined stopping criterion, such as a minimum number of training instances assigned to each leaf node of the tree. CART uses a greedy algorithm in order to minimize a cost function. All features and all possible split points are evaluated and chosen in a greedy manner, i.e., the very best split point is chosen each time.

BANK

Example 8.3.1 (BANK) We continue with the BANK data, using the train-test split we created earlier. We use the R package *rpart* to implement CART, using default options. We grow the tree using the rpart() function with all the predictors on the train data.

```
library(rpart)
fit.allp <- rpart(y~.,method="class", data=bank.data.train,
                  control=rpart.control(minsplit=1, cp=0.001))
```

The *rpart* package uses the Gini index (see (8.1)) impurity and minimizes a cost

$$\text{Cost}_{Cp}(\text{Tree}) = \text{Error}(\text{Tree}) + Cp\,\mathcal{N}(\text{Tree}),$$

where, Error(Tree) is the fraction of misclassified cases and $\mathcal{N}$(Tree) is the number of leaf

nodes in the tree. Features possessing the least value of the Gini index would be preferred at each step. The *root node error* is the percent of incorrectly classified cases at the first (root) splitting node. Here, it is the proportion of subscriptions in the dataset. The *rel error* (relative error) is the error in predicting the training observations (similar to MSE in linear regression). It decreases as the tree grows and becomes more and more adjusted to the data. However, this may not mean that we will predict the test data well; larger trees tend to overfit the training data, and may not work well on test data. By contrast, *xerror* is the error on the observations from cross validation (test) data (similar to the PRESS statistic in linear regression), and is a more realistic estimate of the performance of the tree on new test samples. It is obtained by the rpart() function via an internal cross validation process. The complexity parameter Cp is the minimum improvement in the model needed at each node. When $Cp = 0$, the algorithm will return the original fully grown tree. As Cp increases, we incur a penalty proportional to the number of leaf nodes. We select the value of Cp which gives the subtree resulting in the smallest cross-validated prediction error (xerror) using the results below and Figure 8.2.

```
> printcp(fit.allp) # display the results

Classification tree:
rpart(formula = y ~ ., data = bank.data.train, method = "class",
    control = rpart.control(minsplit = 1, cp = 0.001))

Variables actually used in tree construction:
 [1] age       balance   campaign  contact   default   duration  education housing   job       marit
[11] month     pdays     poutcome  previous

Root node error: 416/1972 = 0.21095
n= 1972

          CP nsplit rel error  xerror     xstd
1  0.0913462      0 1.0000000 1.00000 0.043552
2  0.0600962      2 0.8173077 0.82212 0.040417
3  0.0480769      3 0.7572115 0.78125 0.039604
4  0.0240385      4 0.7091346 0.71635 0.038233
5  0.0112179      5 0.6850962 0.70192 0.037914
6  0.0096154      9 0.6394231 0.73317 0.038599
7  0.0084135     10 0.6298077 0.74279 0.038804
8  0.0080128     12 0.6129808 0.74279 0.038804
9  0.0072115     15 0.5889423 0.75240 0.039008
10 0.0060096     20 0.5504808 0.75240 0.039008
11 0.0054087     24 0.5264423 0.75000 0.038957
12 0.0048077     28 0.5048077 0.76683 0.039308
13 0.0044071     39 0.4495192 0.76683 0.039308
14 0.0040064     46 0.4182692 0.78606 0.039702
15 0.0036058     50 0.4014423 0.79567 0.039895
16 0.0034341     72 0.3100962 0.80288 0.040039
17 0.0032051     79 0.2860577 0.80048 0.039991
18 0.0024038     82 0.2764423 0.79808 0.039943
19 0.0018029    153 0.1057692 0.84615 0.040877
20 0.0016026    158 0.0961538 0.86779 0.041281
21 0.0013736    174 0.0697115 0.87260 0.041370
22 0.0012019    181 0.0600962 0.91106 0.042061
23 0.0010000    224 0.0072115 0.91106 0.042061
```

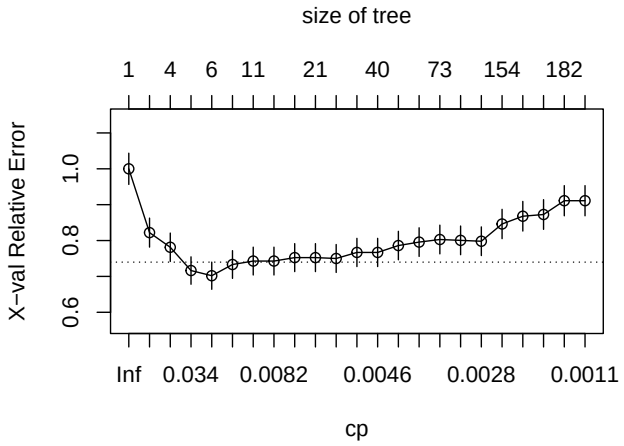

FIGURE 8.2: Cross-validation results under CART for the BANK data.

From the table above, we can see that the smallest value of *xerror* is 0.70192 at a *Cp* value of 0.0112179. We can also pull these out using the code below.

```
> (cp= fit.allp$cptable[which.min(fit.allp$cptable[,"xerror"]),"CP"])
[1] 0.01121795
> (xerr = fit.allp$cptable[which.min(fit.allp$cptable[,"xerror"]),"xerror"])
[1] 0.7019231
```

Figure 8.2 shows a plot of the cross-validation results obtained using the code:

```
plotcp(fit.allp)
```

We can obtain all the results and a summary of fit.allp, although the output can be quite large (output not shown here).

```
summary(fit.allp) # see detailed summary of splits - can be a lot of output
# We can also print all results (may be a large output)
print(fit.allp)
```

We can generate an attractive plot of the fitted tree using the *rpart.plot* package, see Figure 8.3. The tree has a large number of leaf nodes!

```
library(rpart.plot)
rpart.plot(fit.allp, extra = "auto")
```

We can calculate the base level accuracy on the test data (note that we must make it a data frame first), including the confusion matrix, sensitivity, specificity, and misclassification error rate. Recall that sensitivity, or true positive (TP) is the probability that a binary response is predicted as a 1 (or, "yes"), given that it is an event (or,"yes"). Specificity, or

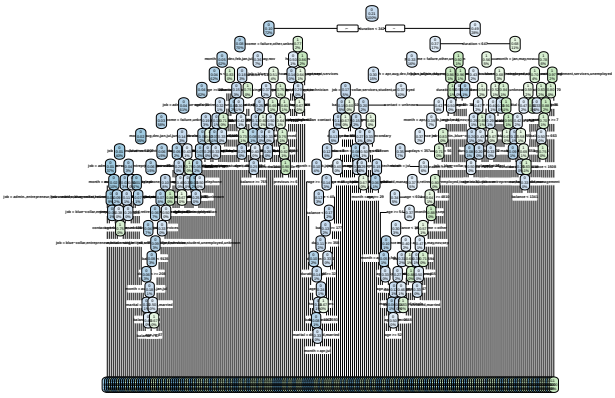

FIGURE 8.3: Fitted tree using CART for the BANK data.

true negative (TN) is the probability that a binary response is predicted as a 0 (or, "no"), given that it is a non-event (or,"no").

```
> test_df <- data.frame(actual=bank.data.test$y,pred=NA)
> test_df$pred <- predict(fit.allp, newdata = bank.data.test, type = "class")
> (conf_matrix_base <- table(test_df$actual,test_df$pred)) #confusion matrix

     0   1
  0 333  56
  1  41  64
> library(caret)
> sensitivity(conf_matrix_base)
[1] 0.8903743
> specificity(conf_matrix_base)
[1] 0.5333333
> (mis.rate <- conf_matrix_base[1,2] + conf_matrix_base[2,1])/sum(conf_matrix_base)
[1] 0.1963563
```

The confusion matrix shows that 333 cases are correctly classified as 0, while 64 were correctly classified as 1. The rest were misclassified. The sensitivity is 0.89, the specificity is 0.53, while the overall misclassification rate is about 0.20.

Remark 1. It is recommended to start with a small Cp value, which we can do using the *control* option, i.e., control=rpart.control(cp=0.0001).

Remark 2. The CART algorithm makes a locally optimal decision by making the best possible choice at each stage, without considering whether those choices remain optimal in future stages.

Remark 3. The algorithm does not find a globally optimal tree. Also, the algorithm grows a tree in an unrestricted way without restricting to shallow trees, and then prunes back using an appropriate criterion.

Remark 4. We can use the entropy as the impurity measure instead of the Gini index.

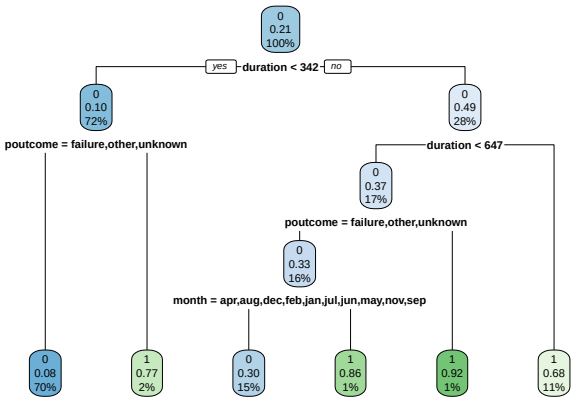

FIGURE 8.4: Fitted pruned tree using CART for the BANK data.

It has been shown that any difference in results between the two impurity measures can occur only in about 2% of the cases. Since we must compute a logarithm, entropy may be slower to compute (see (8.2)).

The function prune() can be used to select a subtree of the tree obtained with rpart() if we think (by looking at the *xerror* estimates) that we would fit the data better by pruning. The code and output below show how we find the Cp corresponding to the smallest *xerror*, and use the best Cp to prune the tree to avoid overfitting, i.e., select a tree size that minimizes the cross-validation error (*xerror*) to get shallow rather than deep trees.

```
pfit.allp <- prune(fit.allp, cp=
    fit.allp$cptable[which.min(fit.allp$cptable[,"xerror"]),"CP"])
rpart.plot(pfit.allp,extra = "auto")
```

We can again use summary(pfit.allf) to see details of what happens at each node as we prune the tree (output may still be large).

Recall that the *root node error* is the percent of incorrectly sorted cases at the first (root) splitting node, *rel error* is the error in predicting the training observations, while *xerror* is the error on the observations from cross validation (test) data. We can use these to calculate two measures of predictive performance:

$$\begin{aligned} \text{resubstitution error rate} &= \text{Root Node Error} \times \text{rel error} \\ \text{cross-validation error rate} &= \text{Root Node Error} \times \text{xerror}, \end{aligned} \tag{8.3}$$

which are respectively error rates computed on the training sample and the test sample.

```
> (rnerr <- pfit.allf$frame[1, 'dev']/pfit.allf$frame[1, 'n'])
[1] 0.4819945
> (min.xerror <- min(pfit.allf$cptable[,"xerror"]))
[1] 0.01340996
>
> # Misclassification rate for prediction = 100(Root node error x min xerror)
> (misclass.rate <- 100*rnerr*min.xerror)
[1] 0.6463527
> (correctclass.rate <- 100 - misclass.rate)
[1] 99.35365
```

We can again calculate the confusion matrices for the training data and test data predictions. When the pruned tree differs from the tree prior to pruning, these numbers reflect better predictive accuracy. We can calculate the accuracy on the test data of the pruned tree.

```
> test_df <- data.frame(actual=bank.data.test$y,pred=NA)
> test_df$pred <- predict(pfit.allp, newdata = bank.data.test, type = "class")
> (conf_matrix_pruned_tree <- table(test_df$actual,test_df$pred)) #confusion matrix

     0   1
  0 370  19
  1  54  51
> library(caret)
> sensitivity(conf_matrix_pruned_tree)
[1] 0.8726415
> specificity(conf_matrix_pruned_tree)
[1] 0.7285714
> # Missclassification error rate:
> (conf_matrix_pruned_tree[1,2] + conf_matrix_pruned_tree[2,1])/sum(conf_matrix_pruned_tree)
[1] 0.1477733
```

Random Forest

The random forest (RF) is an *ensemble learning* method which consists of aggregating a large number of decision trees Breiman (2001) to avoid overfitting and build a better classification model. The word *random* appears because in training the data, predictors are chosen randomly from the full set of predictors. The word *forest* is used because output from multiple trees are used to make a decision. That is, two types of randomnesses go into constructing a random forest: (i) each tree is built on a random sample from the dataset, and (ii) at each tree node, a subset of features are randomly selected to generate the best split. RF is a non-linear classification algorithm. Figure 8.5 represents a RF pictorially (source: `https://community.tibco.com/wiki/random-forest-template-tibco-spotfirer-wiki-page`). Since we expect that a large number of uncorrelated trees (models) operating together will outperform individual trees (models), an RF algorithm is usually more accurate than a single decision tree algorithm.

The training algorithm for a random forest applies the technique of *bootstrap aggregating, or bagging*, to tree learners. That is, the training dataset is bagged repeatedly B times to select a random sample with replacement. We then fit decision trees to these samples as follows. For $b = 1, \ldots, B$:

(i) generation of bootstrap samples with replacement, to get n training observations from

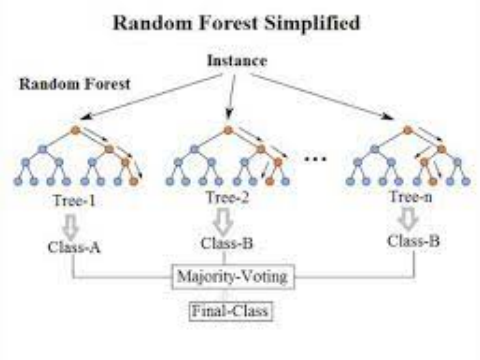

FIGURE 8.5: Random Forest Explained.

the training dataset. Sampling with replacement means that if an observation is chosen in the first random draw, it still remains in the original basket for choosing in another random draw that may follow with an equal probability.

(ii) train a decision tree on each of the training examples in (i).

Out-of-bag (OOB) observations from the first bootstrap sample are those observations in the training sample that did not enter the first bootstrap sample. Similarly, we will have OOB observations corresponding to each bootstrap sample (decision tree).

After the different decision tree models have been trained, the leftover or OOB samples will be given as unseen data to decision trees that did not include them. The decision trees will predict these as either 1 or 0. The out-of-bag (OOB) error is the average error for each training observation calculated using predictions from the trees that did not contain this training observation in their respective bootstrap sample. This allows the RandomForestClassifier to be fit and validated while being trained. After training, predictions for test data samples are obtained by taking the majority vote in the case of decision trees.

BANK

Example 8.3.2 (BANK) We continue with the BANK data, using the 80-20 train-test split we created earlier. We use the R package *ranger* to implement the random forest algorithm, with the Gini index as the impurity measure for classification.

```
library(ranger)
fit.rf.ranger <- ranger(y ~ ., data=bank.data.train,
                  importance='impurity', mtry=3)
```

```
> print(fit.rf.ranger)
Ranger result

Call:
 ranger(y ~ ., data = bank.data.train, importance = "impurity",      mtry = 3)

Type:                             Classification
Number of trees:                  500
Sample size:                      1972
Number of independent variables:  15
Mtry:                             3
Target node size:                 1
Variable importance mode:         impurity
Splitrule:                        gini
OOB prediction error:             15.31 \%
```

After training a RF, we would like to understand which variables have the most predictive power. Variables with high importance will have a significant impact on the binary outcomes, while we may consider dropping variables with low importance from the model (leading to a more parsimonious model). We can use the vi() function in the R package *vip* to extract and print a tibble of variable importance scores. We can also construct a variable importance plot using the vip(fit.rf.ranger) function (see Figure 8.6).

```
library(vip)
(v1 <- vip(fit.rf.ranger))
> (v1 <- vi(fit.rf.ranger))
# A tibble: 15 x 2
   Variable  Importance
   <chr>          <dbl>
 1 duration      204.
 2 balance        65.4
 3 age            62.8
 4 month          53.1
 5 pdays          35.2
 6 job            32.6
 7 poutcome       28.8
 8 campaign       26.0
 9 education      20.3
10 previous       18.6
11 contact        17.7
12 marital        15.7
13 housing        14.2
14 loan            5.92
15 default         2.08
```

For each variable, we can use the above table and Figure 8.6 to see how important it is in classifying the data. The vip plot shows the variables from high to low importance.

We can now use the fitted RF model to predict responses from the test data, and construct the confusion matrix. We see that 372 0's are predicted as 0, while 54 1's are predicted as 1.

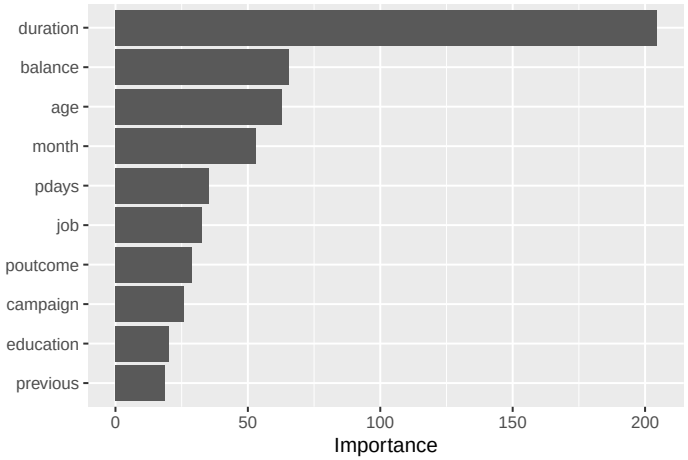

FIGURE 8.6: Variable Importance plot from RF for the BANK data.

```
pred <- predict(fit.rf.ranger, data = bank.data.test)
test_df <- data.frame(actual=bank.data.test$y,pred=NA)
test_df$pred <- pred$predictions
(conf_matrix_rf <- table(test_df$actual,test_df$pred)) #confusion matrix

      0   1
  0 372  17
  1  51  54
```

We can use the R package *caret* to find sensitivity, specificity and misclassification error rate.

```
> library(caret)
> sensitivity(conf_matrix_rf)
[1] 0.8794326
> # Specificity
> specificity(conf_matrix_rf)
[1] 0.7605634
> # Missclassification error rate:
> (conf_matrix_rf[1,2] + conf_matrix_rf[2,1])/sum(conf_matrix_rf)
[1] 0.1376518
```

Remark 1. Other R packages such as *randomForestSRC* and *randomForest* can also be used to fit RF models to binary respoonse data.

8.3.2 Gradient Boosting

Like random forests, *boosting* is an out-of-the box learning algorithm that enjoys good predictive performance for the response usually in high-dimensional settings, with a large number of features. In contrast to random forests which build an ensemble of independent deep trees, gradient boosting algorithms (GBMs) successively build an ensemble of shallow

trees, each tree learning from the previous tree. When these trees are combined, these trees provide a highly accurate predictive algorithm.

For binary response modeling, the idea of boosting was introduced to improve the performance of *weak learners*. This was done by resampling the training data responses, giving more weight to the misclassified ones, thereby leading to a refined classifier (binary model) which would boost feature performance, especially in ambiguous areas of the feature space. A popular variant is the *gradient boosting algorithm* , and XGBoost (acronym for eXtreme Gradient Boosting) is an increasingly popular tool for predictive modeling which we can fit using the *xgboost* package. This is sometimes referred to as regularized gradient boosting, and is thought to be faster and more accurate thgan using a package like *gbm*. The following link is useful: `https://cran.r-project.org/web/packages/xgboost/vignettes/discoverYourData.html`

We use the code below to do dummy variable or indicator encoding of the categorical variables in the train and test data sets. This procedure is called one-hot encoding on the machine learning domain.

```
# Transform the predictor matrix using or dummy (or indictor or one-hot)
    encoding
matrix_predictors.train <- as.matrix(sparse.model.matrix(y ~ ., data =
    bank.data.train))[,-1]
matrix_predictors.test <- as.matrix(sparse.model.matrix(y ~ ., data =
    bank.data.test))[,-1]
```

The *xgboost* package requires a specific type of data setup. Specifically, we set up the predictors (or features) and the response (or label) in a Dmatrix format. The resulting matrices are `dtrain` and `dtest` for the train and test data sets respectively.

```
# Train dataset
pred.train.gbm <- data.matrix(matrix_predictors.train) # predictors only
bank.data.train.gbm <- as.numeric(as.character(bank.data.train$y)) #convert
    factor to numeric
dtrain <- xgb.DMatrix(data = pred.train.gbm, label=bank.data.train.gbm)
# Test dataset
pred.test.gbm <- data.matrix(matrix_predictors.test) # predictors only
bank.data.test.gbm <- as.numeric(as.character(bank.data.test$y)) #convert
    factor to numeric
dtest <- xgb.DMatrix(data = pred.test.gbm, label=bank.data.test.gbm)
```

Then, set up two lists, i.e., `watchlist` and `param` as follows.

```
watchlist <- list(train=dtrain, test=dtest)
param <- list(max_depth = 2, eta = 1, verbose = 0, nthread = 2,
              objective = "binary:logistic", eval_metric = "auc")
```

We fit the XGBoost model and display the accuracy (AUC) for the training and test data at each of `nround =2` rounds.

```
> model.xgb <- xgb.train(param, dtrain, nrounds = 2, watchlist)
[17:43:05] WARNING: amalgamation/../src/learner.cc:627:
Parameters: { "verbose" } might not be used.

  This could be a false alarm, with some parameters getting used by language bindings but
  then being mistakenly passed down to XGBoost core, or some parameter actually being used
  but getting flagged wrongly here. Please open an issue if you find any such cases.


[1]        train-auc:0.797338        test-auc:0.785384
[2]        train-auc:0.858964        test-auc:0.853679
```

We can use the fitted model to make predictions on the train data as follows, and check the accuracy.

```
> pred.y.train <- predict(model.xgb, pred.train.gbm)
> prediction.train <- as.numeric(pred.y.train > 0.5)
> # Measure prediction accuracy on train data
> (tab<-table(bank.data.train.gbm,prediction.train))
                  prediction.train
bank.data.train.gbm    0    1
                  0 1505   51
                  1  270  146
> #;tab
> sum(diag(tab))/sum(tab)
[1] 0.8372211
```

Similarly, we can predict and test accuracy on the test data.

```
> pred.y = predict(model.xgb, pred.test.gbm)
> prediction <- as.numeric(pred.y > 0.5)
> print(head(prediction))
[1] 0 0 0 0 1 1
> # Measure prediction accuracy on test data
> (tab1<-table(bank.data.test.gbm,prediction))
                 prediction
bank.data.test.gbm   0   1
                 0 375  14
                 1  60  45
> sum(diag(tab1))/sum(tab1)
[1] 0.8502024
```

8.4 Models for Categorical Responses

Categorical (or polytomous) responses have more than two levels. The responses can be of two types, *nominal*, or *ordinal*. With nominal (or nominally scaled) data, the levels are not ordered. Examples include marital status (single, married, widowed, divorced, separated), or toothpaste brand (Colgate, Crest, Sensodyne, Other). By contrast, the levels of ordinal responses are ordered. Examples include response to a drug (no improvement, slight improvement, marked improvement), or satisfaction level (low, medium, high), etc. While

proportional odds and partial proportional odds models are usually used for analyzing ordinal data, generalized logit models may also be used. For nominal data, we use generalized logit models.

8.4.1 Nominal Categorical Response Modeling

Glass identification

Suppose the dataset consists of responses Y_i from each of n subjects, where Y_i is a categorical variable with $L > 2$ levels (or categories). Assume that there is no ordering in the levels, i.e., Y is nominally scaled. The data also consists of p predictors $X_j,\ j = 1, \ldots, p$.

Example 8.4.1 (Glass identification data) The data obtained from the website `https://archive.ics.uci.edu/ml/datasets/glass+identification` is based on a study motivated by criminological investigation of classification of the type of glass left at the scene of a crime. The glass can be used as evidence, provided it is correctly identified based on several predictors. The predictors are the refractive index (RI), and the weight percent in the corresponding oxide in the following elements: sodium (NA), magnesium (Mg), aluminum (Al), silicon (Si), potassium (K), calcium (Ca), barium (Ba), and iron (Fe). There are six types of glass, which are type 1: building windows float processed; type 2: building windows non-float processed; type 3: vehicle windows float processed; type 4: vehicle windows non-float processed (none in this database); type 5: containers; type 6: tableware; and type 7: headlamps.

Research question #2

Can we predict the glass type from a knowledge of the predictors such as the refractive index, and percent weight of chemical elements?

We read the data from the file glass.csv. The columns Ba and Fe are mostly zeroes, and we drop them. We also relabel the glass types into three nominal (unordered) categories.

```
dat <- read.csv("Data/glass.csv")
rm_id <- which(colnames(dat) %in% c("Ba","Fe"))
glass.dat <- dat[,-rm_id]  #drop predictors Ba and Fe
glass.dat$Type <- ifelse(glass.dat$Type>=3,3,glass.dat$Type) #three levels
    in the response
```

We do an 80-20 split of the data into training and test sets.

```
# Do 80-20 glass.train-glass.test split of the data dat - random split
set.seed(123457)
strats <- as.factor(glass.dat$Type)
rr <- split(1:length(strats),strats)
p<- 0.8
idx <- sort(as.numeric(unlist(sapply(rr, function(x) sample(x, length(x) *
    p)))))
glass.train <- glass.dat[idx,]
table(glass.train$Type)/nrow(glass.train)
glass.test <- glass.dat[-idx,]
```

```
> table(glass.test$Type)/nrow(glass.test)

        1         2         3
0.3181818 0.3636364 0.3181818
```

We extend the logit regression model discussed in Section 7.4 to explain the effect of predictors on the categorical response Y with $L = 3$ levels. Given a data frame with n cases, corresponding to case i, there will be L binary responses $Y_{i,1}, \ldots, Y_{i,L}$, where

$$Y_{i,\ell} = \begin{cases} 1 & \text{if case } i \text{ response is in category } \ell \\ 0 & \text{otherwise,} \end{cases} \tag{8.4}$$

with $\sum_{\ell=1}^{L} Y_{i,\ell} = 1$, since the events are mutually exclusive (i.e., only one of the L events can occur for case i). Let $\pi_{i,\ell} = P(Y_{i,\ell} = 1)$ denote the probability that category ℓ is selected for case i. Let n_ℓ cases correspond to $Y_i = \ell,\ \ell = 1, \ldots, L$. For $i = 1, \ldots, n$, we can write $\mathbf{y}_i = (Y_{i,1}, \ldots, Y_{i,L})'$ as an L-dimensional data vector which is assumed to follow a multinomial probability distribution with p.m.f.

$$P(Y_{i,1} = n_1, Y_{i,2} = n_2, \ldots, Y_{i,L} = n_L) = \frac{n!}{\prod_{\ell=1}^{L} n_\ell!} \prod_{\ell=1}^{L} p_{i,\ell}^{n_{i,\ell}},\ \sum_{\ell=1}^{L} n_i = n; \sum_{\ell=1}^{L} p_{i,\ell} = 1. \tag{8.5}$$

We define *generalized logits* as shown below, choosing one of the L levels as the baseline level. Suppose we choose the last category L to denote the baseline level. We consider $L-1$ comparisons to this reference category. Then, for $\ell = 1, \ldots, L-1$, let

$$\text{logit}_{i,\ell} = \log\left(\frac{\pi_{i,\ell}}{\pi_{i,L}}\right) = \beta_{\ell,0} + \sum_{j=1}^{p} \beta_{\ell,j} X_{i,j}, \tag{8.6}$$

where $X_{i,j}\ j = 1, \ldots, p$ are p predictors on the ith response, while $\beta_{\ell,0}, \beta_{\ell,1}, \ldots, \beta_{\ell,p}$ denote the intercept and partial regression coefficients for the ℓth logit. Note that for each logit, there are different intercept and partial regression coefficients. Similar to the binary case, we can use an inverse transformation to go from the generalized logits to the category probabilities, i.e., for $\ell = 1, \ldots, L-1$, we have

$$\pi_{i,\ell} = \frac{\exp(\beta_{\ell,0} + \sum_{j=1}^{p} \beta_{\ell,j} X_{i,j})}{1 + \sum_{k=1}^{L-1} \exp(\beta_{k,0} + \sum_{j=1}^{p} \beta_{k,j} X_{i,j})}. \tag{8.7}$$

Then, $\pi_{i,L} = 1 - \sum_{\ell=1}^{L-1} \pi_{i,\ell}$.

Suppose for example that $L = 3$, the generalized logits are

$$\text{logit}_{i,1} = \log\left(\frac{\pi_{i,1}}{\pi_{i,3}}\right) = \beta_{1,0} + \sum_{j=1}^{p} \beta_{1,j} X_{i,j} \tag{8.8}$$

$$\text{logit}_{i,2} = \log\left(\frac{\pi_{i,2}}{\pi_{i,3}}\right) = \beta_{2,0} + \sum_{j=1}^{p} \beta_{2,j} X_{i,j}. \tag{8.9}$$

Suppose we wish to compare probabilities for Category 1 with Category 2.

$$\log\left(\frac{\pi_{i,1}}{\pi_{i,2}}\right) = \log\left(\frac{\pi_{i,1}}{\pi_{i,3}} \times \frac{\pi_{i,3}}{\pi_{i,2}}\right) = \log\left(\frac{\pi_{i,1}}{\pi_{i,3}}\right) - \log\left(\frac{\pi_{i2}}{\pi_{i3}}\right) \tag{8.10}$$

$$= (\beta_{1,0} + \sum_{j=1}^{p} \beta_{1,j} X_{i,j}) - (\beta_{2,0} + \sum_{j=1}^{p} \beta_{2,j} X_{i,j}). \tag{8.11}$$

We can use the multinom() function to fit a generalized logit (also called the multinomial logit) model to the $L = 3$ glass types, the code and output are shown below.

```
fit.gl <- multinom(as.factor(Type) ~ ., data = glass.train)
```

```
> summary(fit.gl)

Call:
multinom(formula = as.factor(Type) ~ ., data = glass.train)

Coefficients:
  (Intercept)       RI        Na        Mg        Al        Si         K        Ca
2    194.0741 108.2017 -2.921222 -4.554711 0.4024951 -3.751140 -3.695590 -3.462326
3    197.6751 112.2804 -1.887646 -5.955096 0.4189000 -3.947215 -3.871303 -4.129335

Std. Errors:
  (Intercept)         RI        Na        Mg       Al        Si        K        Ca
2  0.02490786 0.04051535 0.5071799 0.6310351 1.177974 0.1365748 1.367178 0.4698985
3  0.02981848 0.04868940 0.6007941 0.6610636 1.296439 0.1586926 1.407175 0.5189273

Residual Deviance: 230.6129
AIC: 262.6129
```

Note that level 1 is treated as a baseline level. Intercepts and partial regression coefficients corresponding to levels $\ell = 2, 3$ are shown above, along with the standard errors of these coefficients. The residual deviance from the fitted model and AIC are also shown.

We can evaluate the fitted model on the test data denoted by glass.test. We predict the held-out responses in glass.test using the code shown below, using the option `type="class"` in the predict() function. We then use the ctable() function on the observed and predicted responses in the test data to construct a classification table. The table helps us to calculate the model fitting accuracy as the sum of the diagonal elements divided by the total number of observations in glass.test.

```
glass.test$pred <- predict(fit.gl, newdata = glass.test, type="class")
table=cbind(glass.test$Type,glass.test$pred)
```

```
> (ctable.pred <- table(glass.test$Type, glass.test$pred)) #classification table

    1  2  3
  1  9  5  0
  2  1 12  3
  3  4  5  5

> round((sum(diag(ctable.pred))/sum(ctable.pred))*100,2) #accuracy
[1] 59.09
```

We can also compute the model accuracy in the dataset glass.training by replacing glass.test by glass.train in the code above, as we show below. The accuracy is about 70%, while the accuracy is the test set was only about 60%.

```
glass.train$pred <- predict(fit.gl, newdata = glass.train, type="class")
table=cbind(glass.train$Type,glass.train$pred)
```

```
> (ctable.pred <- table(glass.train$Type, glass.train$pred)) #classification table

    1  2  3
  1 41 15  0
  2 18 36  6
  3  5  8 41
> round((sum(diag(ctable.pred))/sum(ctable.pred))*100,2) #accuracy
[1] 69.41
```

8.4.2 Ordinal Categorical Response Modeling

When the response variable is ordinal, we can model *cumulative logits* (instead of generalized or multinomial logits defined in (8.6)). Suppose there are L ordinal levels of Y, and there are p predictors, $X_j,\ j = 1, \ldots, p$. Then,

$$P(Y \leq \ell) = P(Y = 1) + P(Y = 2) + \ldots + P(Y = \ell) \tag{8.12}$$

is the cumulative probability of Y being less than or equal to a specific category ℓ, where $\ell = 1, 2, \ldots, L$; of course, $P(Y > \ell) = 1 - P(Y \leq \ell)$,

$P(Y \leq L) = 1$, and $P(Y > L) = 1 - P(Y \leq L) = 0$. The generalized odds of $Y \leq \ell$ are

$$\text{Odds that } Y \leq \ell = \frac{P(Y \leq \ell)}{P(Y > \ell)} = \frac{P(Y \leq \ell)}{1 - P(Y \leq \ell)},\ \ell = 1, \ldots, L-1. \tag{8.13}$$

The generalized log odds or logits of $Y \leq \ell$ are then

$$\text{logit}(P(Y \leq \ell)) = \log\left(\frac{P(Y \leq \ell)}{P(Y > \ell)}\right),\ \ell = 1, \ldots, L-1. \tag{8.14}$$

For $\ell = 1, \ldots, L-1$, the logits will be used as link functions for modeling Y as a function of predictors.

Crash Severity

Example 8.4.2 (Crash severity) The crashseverity.csv file contains data related to

pedestrian crashes of different severity levels on different road segments in California during a selected week. Each row corresponds to a distinct crash. This data was created by merging two data sets, one related to crashes and the other related to road characteristics. The categorical response variable `severity` has five levels (1 – 5), indicating the severity level of a crash: property damage only, complaint of pain, other visible injury, severe injury, or fatal injury. There is a clear ordering in the levels of `severity` from least severe to most severe. The categorical predictors include `light` with two levels, DL (daylight), or NDL (no daylight); `mediantype` with two levels, UD (undivided), or D (divided); `numlanes` denoting the number of lanes with four levels, 2 (two lanes), 3 (three lanes), 4 (four lanes) and > 4 (five or more lanes); and `speed` denoting speed limit with three levels, L (25, 30, or 35 mph), I (40, 45, 40, or 55 mph), and H (60, 65, or 70 mph). There are three numerical predictors, `medianwidth`, `AADT` denoting traffic volume, and `numvehs` denoting the number of vehicles involved in the crash.

Research question #1

Do the crash level and roadway segment level predictors explain crash severity?

We read the data from the file crashseverity.csv. We can look at the details on the categorical variables.

```
data=read.csv("Data/crashseverity.csv", sep = ",", header= TRUE)
lapply(data[, c("severity","numlanes", "mediantype",
                "speed","ruralurban","light")], table)
```

We reorder the response variable, so that 1 corresponds to the lowest severity (PDO) and 5 corresponds to the highest severity (Fatal). We also combine F, SI, and OVI into one category '3'.

```
data$severity = recode(data$severity, "1='F';2='SI';3='OVI';4='CoP';5='PDO'")
data$severity = recode(data$severity, "'F'=5;'SI'=4;'OVI'=3;'CoP'=2;'PDO'=1")
data$severity = recode(data$severity, "1=1;2=2;else='3'")
```

We create dummy variables for the categorical predictors using the model.matrix() function.

```
ex.cat=model.matrix(~-1+numlanes+mediantype+ruralurban+light+speed,
    data=data)
ex.cat=ex.cat[,-1] #removing category >4 in numlanes to avoid singularity
head(ex.cat)
crash.new=cbind(data[,c(6,8,11,13)],ex.cat)  # cbind response and all
    predictors
```

We also standardize the traffic volume, and make severity a factor variable. We will model the data in `crash.new`.

```
crash.new$volume=scale(crash.new$volume)
crash.new$severity <- as.factor(crash.new$severity)
```

The cumulative logits model (or, proportional odds (PO) model) for the nominally scaled categorical response Y is

$$\text{logit}(P(Y_i \leq \ell)) = \beta_{\ell,0} + \sum_{j=1}^{p} \beta_j X_{i,j} = \beta_{\ell,0} + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \ldots + \beta_p X_{i,p}, \qquad (8.15)$$

for $\ell = 1, \ldots, L-1$. The model parameters are $\beta_{\ell,0},\ \ell = 1, \ldots, L-1$ and $\beta_j,\ j = 1, \ldots, p$. While this model assumes different intercept parameters $\beta_{\ell,0}$ for the $L-1$ cumulative logits, it has the same partial regression coefficients $\beta_j,\ j = 1, \ldots, p$ for all the levels $\ell = 1, \ldots, L-1$. Since these logits (log odds) only differ by a constant (intercept) and the term $\sum_{j=1}^{p} \beta_j X_{i,j}$ is the same for all levels ℓ, the log odds are proportional, and we call this model the proportional odds logistic regression model. The PO model, which has $(L-1)+p$ parameters is a more parsimonious model than the multinomial logit model which has $(L-1)(p+1)$ parameters.

We do an 80-20 split of the data into a training portion and test portion.

```
set.seed(123457)
strats <- as.factor(crash.new$severity)
rr <- split(1:length(strats),strats)
p<- 0.8
idx <- sort(as.numeric(unlist(sapply(rr, function(x) sample(x, length(x) *
    p)))))
crash.train <- crash.new[idx,]
summary(crash.train$severity)/nrow(crash.train)
summary(crash.new$severity)/nrow(crash.new)
crash.test <- crash.new[-idx,]
summary(crash.test$severity)/nrow(crash.test)
```

We fit the cumulative logits (or proportional odds) model using the polr() function in the *MASS* package. One thing is to notice is that the polr() function fits the model

$$\text{logit}(P(Y_i \leq \ell)) = \beta_{\ell,0} - \sum_{j=1}^{p} \eta_j X_{i,j} = \beta_{\ell,0} - \eta_1 X_{i,1} - \eta_2 X_{i,2} - \ldots - \eta_p X_{i,p} \qquad (8.16)$$

for $\ell = 1, \ldots, L-1$. That is, comparing (8.15) and (8.16), we must remember that $\eta_j = -\beta_j$.

The code and output for fitting the full model with all predictors is shown below. Since the function polr() does not provide p-values for each coefficient, the code below shows how we can compute the p-values and append them to the output.

```
crash.fit1 <- polr(as.factor(severity) ~ ., data = crash.train)
summary(crash.fit1)
(ctable1 <- coef(summary(crash.fit1))) # p-values not given
```

```
p <- pnorm(abs(ctable1[, "t value"]), lower.tail = FALSE) * 2
(ctable1 <- cbind(ctable1, "p value" = p))
 p <- pnorm(abs(ctable1[, "t value"]), lower.tail = FALSE) * 2
> (ctable1 <- cbind(ctable1, "p value" = p))
                  Value  Std. Error    t value      p value
numvehs       0.179532334 0.045442129  3.9507905 7.789348e-05
medwidth      0.001566615 0.001583905  0.9890843 3.226219e-01
volume       -0.256410553 0.056511679 -4.5373019 5.697850e-06
numlanes2    -0.210119590 0.246619331 -0.8519997 3.942143e-01
numlanes3     0.209763569 0.378248858  0.5545650 5.791923e-01
numlanes4     0.135457259 0.119977899  1.1290184 2.588901e-01
mediantypeUD  0.401706808 0.229557437  1.7499185 8.013238e-02
ruralurbanR   0.390225692 0.220765330  1.7676041 7.712712e-02
ruralurbanU   0.393890385 0.216709144  1.8175993 6.912540e-02
lightNDL      0.174302811 0.069707525  2.5004877 1.240224e-02
speedI        0.289579445 0.135068663  2.1439425 3.203749e-02
speedL        0.536237069 0.271392714  1.9758713 4.816936e-02
1|2           1.649259892 0.244158392  6.7548769 1.429565e-11
2|3           3.043743772 0.248709649 12.2381411 1.944349e-34
```

From the coefficients, we can easily compute the odds ratios as follows.

```
> exp(coef(crash.fit1))
     numvehs     medwidth       volume    numlanes2    numlanes3    numlanes4 mediantypeUD  ruralurb
   1.1966576    1.0015678    0.7738242    0.8104873    1.2333864    1.1450603    1.4943731    1.4773
ruralurbanU     lightNDL       speedI       speedL
   1.4827380    1.1904160    1.3358656    1.7095618
```

We can fit a null model with intercepts only.

```
crash.fit2=polr(as.factor(severity)~1, data=crash.train)
(ctable2 <- coef(summary(crash.fit2)))
p <- pnorm(abs(ctable2[, "t value"]), lower.tail = FALSE) * 2
```

```
> (ctable2 <- cbind(ctable2, "p value" = p))
       Value Std. Error  t value      p value
1|2 0.6834306 0.03482293 19.62588 9.296093e-86
2|3 2.0443391 0.05161235 39.60950 0.000000e+00
```

We can carry out variable selection using a stepwise procedure using the smallest AIC as the criterion (via the stepAIC() function).

```
vs.s = polr(as.factor(severity)~1, data=crash.train)
mod.s=stepAIC(vs.s, scope = ~ numvehs + medwidth + volume + numlanes2 +
    numlanes3 + numlanes4 +
                mediantypeUD + speedI + speedL + ruralurbanU + ruralurbanR +
    lightNDL, trace=FALSE,
              direction = "both")
```

```
> summary(mod.s)
Call:
polr(formula = as.factor(severity) ~ volume + numvehs + lightNDL +
    speedI + speedL, data = crash.train)

Coefficients:
           Value Std. Error t value
volume   -0.2990    0.04067  -7.352
numvehs   0.1779    0.04485   3.968
lightNDL  0.1800    0.06941   2.594
speedI    0.3035    0.12206   2.486
speedL    0.4425    0.26015   1.701

Intercepts:
    Value   Std. Error t value
1|2  1.1806  0.1076    10.9667
2|3  2.5719  0.1160    22.1758

Residual Deviance: 6203.962
AIC: 6217.962
```

The following code facilitates residual diagnostics (details not shown here).

```
autoplot.polr(crash.fit1) #q-q plot of residuals
#residual vs  volume
crash.res1=resids(crash.fit1)
scatter.smooth(crash.train$volume, crash.res1, lpars = list(lwd = 2, col =
    "red"),
               xlab = expression(x), ylab = "Surrogate residual",
               main = "Correct model")
plot(gof(crash.fit1, nsim = 50))  #gof plot
```

To check model accuracy on the test data set using the best model we got via the stepAIC() function, we can use the code below. We again calculate accuracy as the sum of diagonal elements divided by total number of observations.

```
crash.test$pred3 <- predict(mod.s, newdata = crash.test, type="class")
ctable.pred3 <- table(crash.test$severity, crash.test$pred3) # classification table
round((sum(diag(ctable.pred3))/sum(ctable.pred3))*100,2) # accuracy
[1] 66.45
```

9

Some Extensions to ANOVA Models

This chapter describes two extensions of the models described in 4. The extensions are random-effects models and mixed-effects models.

9.1 Introduction

In some experimental situations, we may be interested in a factor which has a very large number of possible levels, and the a levels selected into the data frame only form a random sample of these population levels. We refer to such a factor as a random factor and to the effects corresponding to its a levels as random effects. That is, a factor effect is a *random effect* if the specific levels included in the experiment and statistical analysis are not themselves of any particular interest, but can be regarded as a random subset of a much larger (possibly infinite) number of possible population of factor level effects, and statistical analysis makes inference about the variability in the population of levels. Examples of random factors may include days in a year, workers in a large organization, machines in a large factory, batches of material on a production line, etc. ANOVA models which only include random effects for explaining the response are referred to as *random-effects ANOVA models*. Estimating in the context of random effects primarily involves estimating the varibaility in the population of levels via *variance components*.

A multi-factor experiment may often include factors with fixed levels (effects) and factors with random levels (effects). For example, an experiment in a manufacturing plant may involve $a = 4$ temperature settings in which we have specific interest, whereas $b = 10$ workers may be randomly selected from a large pool of hundreds of workers. In this case we may consider the temperature effect as fixed, while the worker effect is assumed to be random. ANOVA models which include factors with fixed and random effects are called *mixed-effects ANOVA models*. In mixed-effects models, we will estimate the fixed effects similar to Chapter 4, while we estimate the variance components corresponding to random effects by a few different approaches such as the ANOVA method, and maximum and restricted maximum likelihood methods.

9.2 Essential Terminology and Notation

✓ In a **fixed-effects model**, we are interested in comparing average responses at the specific treatment levels to see whether they are significantly different. For example, an experiment might involve a comparison of average responses between different doses of a drug, and we are interested in the efficacy of these specific doses.

✓ In a **random-effects model**, the levels of a treatment are regarded as a random sample from a large population of treatments, and we wish to see whether the variation in the population of levels is significantly different from zero. For example, an engineer may suspect existence of considerable variation in fabric strength woven by a large number of looms, and select a few looms (levels) to carry out an experiment. Interest is in not in the randomly selected looms, but we want to see if the selection of looms introduces significant variability, in addition to the random error which occurs in every experiment.

✓ A factor effect is a **random effect** if the specific levels included in the experiment and statistical analysis are not themselves of any particular interest, but can be regarded as a random subset of a much larger (even infinite) number of possible population of factor levels. Random effects are parameters that are themselves random variables.

9.3 Random-Effects (RE) ANOVA Model

Suppose we are interested in analyzing the effect of some factor (A) on a response, where A has a large number of possible levels. We randomly select a levels from the population of levels. Let Y_{ij} denote the jth replicate of the response that we observe for the ith level of Factor A, $i \in 1, \ldots, a$.

The unbalanced one-factor random-effects ANOVA model expresses Y_{ij} as the sum of an overall mean parameter, μ, an effect due to the ith level of the random Factor A which we denote by γ_i, and random errors ε_{ij}:

$$Y_{ij} = \mu + \gamma_i + \varepsilon_{ij},\ j = 1, 2, \ldots, n_i,\ i = 1, 2, \ldots, a. \tag{9.1}$$

Let $N = \sum_{i=1}^{a} n_i$. Similar to Section **??**, we assume that $\varepsilon_{ij} \overset{i.i.d.}{\sim} N(0, \sigma_\varepsilon^2)$, which includes assumptions of independence, equal variances, and normality. Unlike Section **??**, here we use the notation γ_i instead of τ_i to denote are random effects which are i.i.d. $N(0, \sigma_\gamma^2)$ variables, also assumed to be independent of ε_{ij}.

TNC

Example 9.3.1 (TNC usage) The usage of Transportation Network Companies (including Uber, Lyft, and Via) in New York City (NYC) has increased dramatically in recent year. Not long ago, the notion of TNC did not even exist, and the most common modes of transportation in the city were subways, busses, and taxis. The data contains TNC usage (TNC) in 105 taxi zones in the city (`Zoneid`), over a period of two years (104 weeks). We select 10 random zones, and 10 random weeks, and assume that TNC usage for these weeks are independent. The data is in the file tnc-randeff.csv.

Our research question is now stated as follows:

Research question #2

Is there significant difference in TNC usage across different taxi zones?

We read the TNC data file. We convert the variable `Zoneid` into a factor variable `taxi.zone` with $a = 10$ levels.

```
tnc <- read.csv("Data/tnc-randeff.csv", header = TRUE)
attach(tnc.data)
taxi.zone <- factor(Zoneid)
```

The levels correspond to taxi zones with these ID's: "ID250" "ID113" "ID26" "ID97" "ID62" "ID188" "ID112" "ID143" "ID41" "ID100". For each of the a levels (taxi zones), the variable `TNC` contains observations onTNC usage for $n = 10$ different weeks. Let Y_{ij}, $j = 1, \ldots, n, i = 1, \ldots, a$ denote the TNC usage for the jth week (replicate) in taxi zone i. The effect of the factor `taxi.zone` on TNC is treated as a random effect, so the data structure is that of an unbalanced one-factor ANOVA model with a random Factor A (taxi zone). The dataset also shows the population in the different taxi zones. Depending on the objective, we can treat either the TNC usage or the population adjusted TNC usage as the response variable.

Figure 9.1 show parallel boxplots of TNC usage for the $a = 10$ taxi zones.

```
boxplot(TNC~taxi.zone, xlab="taxi zones",ylab="TNC")
```

We can do an ANOVA decomposition similar to what we showed in Chapter 4, see (4.17):

$$\text{SST} = \text{SSA} + \text{SSE}.$$

The sums of squares were defined in (4.18) and degrees of freedom (d.f.) and mean squares (MS) were shown in Table 4.3. This decomposition helps with our primary goal of testing whether the variability in taxi zones, σ^2_γ, is significantly greater than zero. Specifically, we test

$$H_0 : \sigma^2_\gamma = 0 \text{ versus } H_1 : \sigma^2_\gamma > 0.$$

We use the following code to fit the model and see the ANOVA table and F-statistic.

```
tnc.anova <- aov(TNC~taxi.zone, data = tnc.data)
```

```
> summary(tnc.anova)
             Df    Sum Sq   Mean Sq F value Pr(>F)
taxi.zone     9 7.254e+09 805970259   32.69 <2e-16 ***
Residuals   244 6.015e+09  24652433
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, which means that we cannot reject the null hypothesis, and

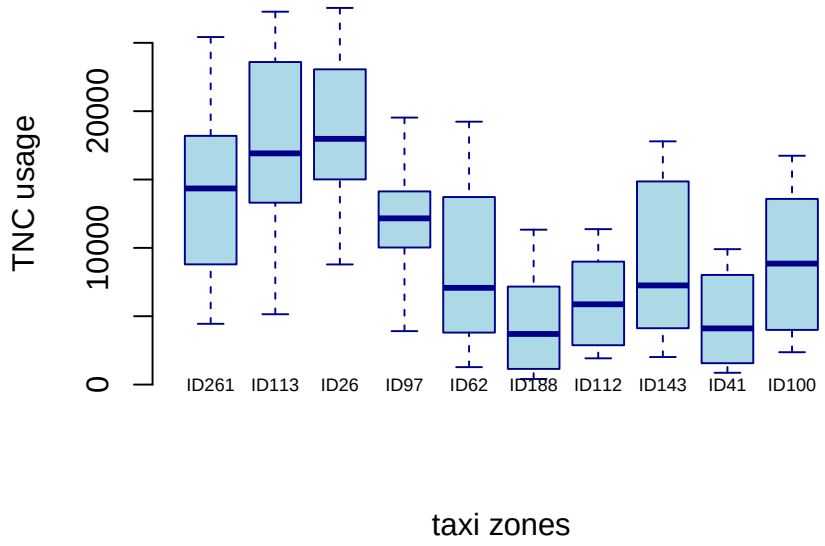

FIGURE 9.1: Parallel boxplots for TNC usage in randomly selected weeks from $a = 10$ randomly seelcted taxi zones in NYC.

thus variability in the population of taxi zone levels is significantly different than zero. The idea behind the F-statistic for the one-factor random-effects model by comparing the expected mean squares, i.e., MSA and MSE, is shown in the appendix.

The restricted maximum likelihood approach is primarily used in random-effect and mixed-effect models for estimating the unknown parameters. We can use lmer() function in the R package *lme4* as shown below.

```
library(lme4)
tnc.reml <- lmer(TNC~1|taxi.zone, data=tnc.data)
```

```
> summary(tnc.reml)

inear mixed model fit by REML ['lmerMod']
Formula: TNC ~ 1 | taxi.zone
   Data: tnc.data

REML criterion at convergence: 5061.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.36592 -0.68916 -0.08093  0.73847  2.29868

Random effects:
 Groups    Name        Variance Std.Dev.
 taxi.zone (Intercept) 31904134 5648
 Residual              24653252 4965
Number of obs: 254, groups:  taxi.zone, 10

Fixed effects:
            Estimate Std. Error t value
(Intercept)     9809       1813   5.409
```

The estimate of the fixed mean μ is 9809 with a SE of 1813. The estimated random effects for the $a = 10$ taxi zones can be recovered using ranef(), which extracts the conditional modes (also the conditional means) of the random effects.

```
> ranef(tnc.reml)
$taxi.zone
      (Intercept)
ID100   4200.7460
ID112   7084.4213
ID113   8579.9168
ID143   2083.4613
ID188  -1292.3571
ID250  -8662.7632
ID26   -5320.7818
ID41    -772.5462
ID62   -5159.3111
ID97    -740.7858
with conditional variances for \taxi.zone"
```

Using a normal Q-Q plot of the residuals from the REML fit in Figure 9.2, we verify that the data supports the normality assumption.

```
tnc.resid <- resid(tnc.reml)
qqnorm(tnc.resid)
```

Remark 1. We can use lmer() with `REML = FALSE` to obtain the maximum likelihood estimates (MLEs) of the variance components, and recover the random effects using ranef().

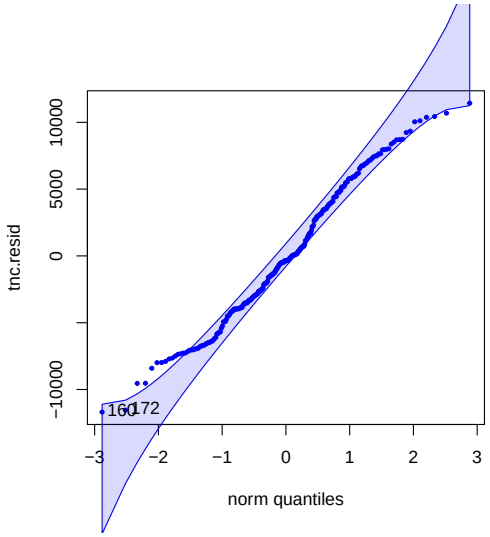

FIGURE 9.2: Normal Q-Q plot for residuals from the one-factor random-effects REML fit for TNC usage.

9.4 Linear Mixed-Effects Models (LMMs)

Linear mixed-effects models (LMMs) allow us to include both fixed effects and random effects for explaining a response variable. For example, students can be randomly sampled from fixed classrooms, or taxi zones in NYC can be randomly sampled from the city boroughs. These are examples of nested designs, where a random factor is nested within a fixed factor. Alternately, we can have a crossed design setup with one of the factors being fixed and the second factor being random. For example, Factor A can be the median age of TNC users in NYC and Factor B can be random taxi zones. Yet another design could be that Factor B (taxi zones) is random and is nested within Factor C (boroughs of NYC). The crossing factor could be Factor A (median age of TNC users). A LMM presents a framework for analyzing the effect of fixed and random factors on a response variable of interest. We show a few examples. Detailed statistical details for LMM are shown in the appendix.

9.4.1 Nested model with a random nested factor

Consider a model with two factors. Suppose Factor A is fixed with a levels, while Factor B is a random factor with b_i levels nested within the ith level of Factor A. Suppose there are n_{ij} replicates in the (i, j)th cell. Let Y_{ijk} denote the kth replicate of the observed response in the (i, j)th cell.

TNC

Example 9.4.1 (TNC nested) We continue analyzing patterns of TNC usage in NYC. Let Factor A denote the four boroughs in NYC (Manhattan, Bronx, Brooklyn, and Queens), with $a = 4$ fixed levels. Let Factor B denote taxi zones that are nested within Factor A, the boroughs. We assume that Factor B is a random factor, where the taxi zones shown in the data set are random samples selected from all the zones within each borough.

Research question #2

Are there significant differences in average TNC between the NYC boroughs? Is there significant variation in TNC between taxi zones nested within boroughs?

We read the data from the file tnc-nested.cvs using the code below.There are $a = 4$ levels in `Borough`.

```
tnc.nested <- read.csv("/Data/tnc-nested.csv", header = TRUE)
unique(tnc.nested$Borough)
```

```
[1] "Manhattan" "Brooklyn"  "Queens"    "Bronx"
```

We convert `Zoneid` into a factor variable `taxi.zone`. The number of taxi zones within each borough can also be obtained using the code shown below, as $b_1 = 11, b_2 = 10, b_3 = 6, b_4 = 4$.

```
taxi.zone <- factor(tnc.nested$Zoneid)
tnc.nested1 <- cbind(tnc.nested, tnc.nested$taxi.zone)
attach(tnc.nested1)
length(unique(subset(tnc.nested1,Borough==as.character("Manhattan"))$Zoneid))
length(unique(subset(tnc.nested1,Borough==as.character("Brooklyn"))$Zoneid))
length(unique(subset(tnc.nested1,Borough==as.character("Bronx"))$Zoneid))
length(unique(subset(tnc.nested1,Borough==as.character("Queens"))$Zoneid))
```

Consider the linear model

$$Y_{ijk} = \mu + \tau_i + \theta_{j(i)} + \varepsilon_{ijk}, \tag{9.2}$$

for $k = 1, \ldots, n_{ij}$, $j = 1, \ldots, b_i$, $i = 1, \ldots, a$. Here, μ and τ_i, $i = 1, \ldots, a$ are fixed effects. The random effects $\theta_{j(i)}$'s are i.i.d. $N(0, \sigma^2_\theta)$ variables for $j = 1, \ldots, n_i$ and $i = 1, \ldots, a$. The random errors ε_{ijk} are assumed to be $N(0, \sigma^2_\varepsilon)$ and are assumed to be uncorrelated with $\theta_{j(i)}$. This is the setup for a general unbalanced mixed-effects nested model. If $n_{ij} = n$ and $b_i = b$ for $i = 1, \ldots, a$ for all j and i, we get the balanced data setup.

Here, we fit a mixed-effects model where `taxi.zone` is a random effect nested within `Borough` with $a = 4$ fixed levels. We use the lme4() function in the *lmer* package.

```
library(lme4)
nested.fit <- lmer(Tnc~ 0 + Borough + (1|Borough:taxi.zone), data =
    tnc.nested1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Tnc ~ 0 + Borough + (1 | Borough:taxi.zone)
   Data: tnc.nested1

REML criterion at convergence: 18321.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.4393 -0.4004  0.0592  0.4950  3.5905

Random effects:
 Groups            Name        Variance Std.Dev.
 Borough:taxi.zone (Intercept) 26483365 5146
 Residual                      20105218 4484
Number of obs: 930, groups:  Borough:taxi.zone, 31

Fixed effects:
                 Estimate Std. Error t value
BoroughBronx         1253       2127   0.589
BoroughBrooklyn      5321       1648   3.229
BoroughManhattan    14421       1571   9.179
BoroughQueens        4544       2606   1.744

Correlation of Fixed Effects:
            BrghBrn BrghBrk BrghMn
BorghBrklyn 0.000
BorghMnhttn 0.000   0.000
BoroughQuns 0.000   0.000   0.000
```

We can estimate the random effects as shown below.

```
> ranef(nested.fit)
$`Borough:taxi.zone`
                 (Intercept)
Bronx:ID241       684.321806
Bronx:ID254       276.020722
Bronx:ID259      -216.450399
Bronx:ID32       -247.433036
Bronx:ID51        -45.509473
Bronx:ID94       -450.949620
Brooklyn:ID108  -4342.881356
Brooklyn:ID181   9999.706095
Brooklyn:ID21   -3363.205837
Brooklyn:ID210  -3328.744562
Brooklyn:ID22   -2819.302889
Brooklyn:ID257  -2863.484844
Brooklyn:ID33    2094.387219
Brooklyn:ID65    1332.923091
Brooklyn:ID80    6317.291322
Brooklyn:ID85   -3026.688239
Manhattan:ID107  5711.867610
Manhattan:ID148  7062.456981
Manhattan:ID164  7604.051674
Manhattan:ID211   492.122371
Manhattan:ID239  2443.410772
Manhattan:ID243 -9243.317764
Manhattan:ID261 -7667.234618
Manhattan:ID42  -6660.250164
Manhattan:ID48   8835.619605
Manhattan:ID75  -7048.914823
Manhattan:ID87  -1529.811643
Queens:ID145     3417.233638
Queens:ID258    -2480.033265
Queens:ID82        -3.275447
Queens:ID92      -933.924926

with conditional variances for \Borough:taxi.zone"
```

9.4.2 Three-factor mixed-effects model with a random nested factor

TNC

Example 9.4.2 (TNC 3factor) We continue to analyze patterns of TNC usage in NYC. Factor A and Factor B are the same as those in Example **??**. Additionally, Factor C is `median.age` of TNC users with three fixed levels: [20, 25], (25, 40], and (40, 55]. Factor A and Factor C are crossed fixed effects, while Factor B is a random factor which is nested within Factor A. The data is in the file tnc-3facme.csv.

Research question #2

Are there significant differences in average TNC between the NYC boroughs? Are there significant differences in average TNC between the median ages of users? Is there evidence of an interaction between Factor A and Factor C? Is there significant variation in TNC between taxi zones?

We read the data from the file

```
tnc.3facme <- read.csv("Data/tnc-3facme.csv", header = TRUE)
```

We then check the levels of the predictors and make them into factor variables. We then bind these new factor variables `taxi.zone` and `age.cat` into the data frame `tnc.3facme1`.

```
unique(tnc.3facme$Borough)
unique(tnc.3facme$AgeCategory)
boro <- factor(tnc.3facme$Borough)
taxi.zone <- factor(tnc.3facme$Zoneid)
age.cat <- factor(tnc.3facme$AgeCategory)
tnc.3facme1 <- cbind(tnc.3facme, taxi.zone, age.cat)
```

We can fit the following linear model to the response `TNC`.

$$Y_{ijk\ell} = \mu + \tau_i + \gamma_k + (\tau\gamma)_{ik} + \theta_{j(i)} + \varepsilon_{ijk\ell}, \tag{9.3}$$

for $\ell = 1, \ldots, n_{ijk}$, $k = 1, \ldots, c$, $j = 1, \ldots, b_i$, $i = 1, \ldots, a$.

We use the lmer() function in the *lme4* package.

```
fit.3facme <- lmer(TNC~ 0 + boro + age.cat + boro*age.cat + (1|boro:taxi.zone), data = tnc.3facme1)
fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients

> summary(fit.3facme)

 summary(fit.3facme)
Linear mixed model fit by REML ['lmerMod']
Formula: TNC ~ 0 + boro + age.cat + boro * age.cat + (1 | boro:taxi.zone)
   Data: tnc.3facme1

REML criterion at convergence: 15136.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2812 -0.3812  0.0613  0.4694  3.9034

Random effects:
 Groups         Name        Variance Std.Dev.
 boro:taxi.zone (Intercept) 36209362 6017
 Residual                   22239206 4716
Number of obs: 769, groups:  boro:taxi.zone, 31

Fixed effects:
                      Estimate Std. Error t value
boroBronx               1245.9     4304.4   0.289
boroBrooklyn            2968.4     6279.7   0.473
boroManhattan          14056.7     6100.8   2.304
boroQueens              5254.3     6147.9   0.855
age.cat2                -538.9     5338.7  -0.101
age.cat3                1960.7     5744.6   0.341
boroBrooklyn:age.cat2   5422.0     3666.4   1.479
boroManhattan:age.cat2  1413.9     8357.4   0.169
boroManhattan:age.cat3  2087.1    10356.4   0.202

Correlation of Fixed Effects:
            brBrnx brBrkl brMnht borQns ag.ct2 ag.ct3 brB:.2 brM:.2
boroBroklyn  0.685
boroManhttn  0.000  0.000
boroQueens   0.700  0.670  0.000
age.cat2    -0.806 -0.772  0.000 -0.868
age.cat3    -0.749 -0.915  0.000 -0.732  0.843
brBrklyn:.2  0.000 -0.447  0.000  0.117 -0.135  0.339
brMnhttn:.2  0.515  0.493 -0.730  0.555 -0.639 -0.539  0.086
brMnhttn:.3  0.416  0.507 -0.589  0.406 -0.468 -0.555 -0.188  0.729
fit warnings:
fixed-effect model matrix is rank deficient so dropping 3 columns / coefficients
```

Ch9-Appendix: Statistical Details

F-test

Consider the ANOVA decomposition shown in (4.17),

$$\text{SST} = \text{SSA} + \text{SSE},$$

with the sums of squares defined in (4.18) and degrees of freedom (d.f.) and mean squares (MS) shown in Table 4.3. We can compute the expected mean squares (EMS) formulas using

$$\begin{aligned} E(\text{SSA}) &= \left(N - \sum_{i=1}^{a} n_i^2/N\right)\sigma_\gamma^2 + (a-1)\sigma_\varepsilon^2, \text{ and} \quad (9.4) \\ E(\text{MSA}) &= (N-a)\sigma_\varepsilon^2. \quad (9.5) \end{aligned}$$

In the balanced case, $n_i = n,\ i = 1, \ldots, a$, and the expected MS become

$$E(\text{MSE}) = \sigma_\varepsilon^2 \text{ and } E(\text{MSA}) = \sigma_\varepsilon^2 + n\sigma_\gamma^2$$

Note that MSE is an unbiased estimator of σ_ε^2, while MSA is an unbiased estimator of σ_ε^2 only when $\sigma_\gamma^2 = 0$. To test

$$H_0 : \sigma_\gamma^2 = 0 \text{ versus } H_1 : \sigma_\gamma^2 > 0,$$

we use the test statistic

$$F = \text{MSA/MSE}, \quad (9.6)$$

Provided the null hypothesis is true, $F \sim F_{a-1,N-a}$ distribution. For a prespecified level of significance α, we reject H_0 if

$$\begin{aligned} F &\geq F_{1-\alpha,a-1,N-a}, \text{ or,} \\ \text{p-value} &= P(F_{a-1,N-a} \geq F) \leq \alpha, \quad (9.7) \end{aligned}$$

If the data does not provide evidence to reject H_0, we conclude that σ_γ^2 is zero, and there is no variation in the true levels of the factor to be informative on the response Y.

Estimating the Mean and Variance Components

For simplicity, we show formulas for the ANOVA method for the balanced model. We equate the expected mean squares (EMS) in (9.5) to their observed counterparts in Table 4.3 for $n_i = n,\ i = 1, \ldots, a$ (recall method of moments), and solve for the variance components as

$$\begin{aligned} \hat{\sigma}^2 &= \text{MSE, and} \\ \hat{\sigma}_\gamma^2 &= \frac{\text{MSA} - \text{MSE}}{n}. \quad (9.8) \end{aligned}$$

One drawback is that the estimate of σ_γ^2 in (9.8) may become negative in some situations. We can resolve this by replacing a negative estimate by 0. We do not show derivations of ML and REML estimates, since these are beyond the scope of this text.

10

Models for Dependent Data

This chapter describes regression models for dependent data. Specifically, we discuss time series regression modeling.

10.1 Introduction

We discussed multiple linear regression (MLR) models in Chapters 5 and 6, based on independent observations on n subjects. In many situations, we wish to use predictors to explain a response variable over time. In this chapter, we describe MLR models with autocorrelated errors to handle regression for time series.

10.2 Linear Regression Models with Autocorrelated Errors (ARMAX Models)

We describe regression with autocorrelated errors. Let Y_t denote the observed response time series. Instead of being independent for $t = 1, \ldots, n$, the responses at different times are autocorrelated, i.e., $\text{Corr}(Y_t, Y_{t+h})$, $h = \pm 1, \pm 2, \ldots$ may not all be zero. Let $Z_{t,j}$, $j = 1, \ldots, p$ the time series serving as explanatory variables (predictors).

Cardiovascular mortality

Example 10.2.1 (Cardiovascular mortality) We consider data on average weekly cardiovascular mortality in Los Angeles County, which was discussed in . The data ia available in the R package *astsa* and consists of $n = 508$ six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979. The dataset also contains a time series of length $n = 508$ on temperatures and particulate matter in Los Angeles County over the same time period corresponding to cardiovascular mortality.

Research question #2

Can we explain and predict cardiovascular mortality using its historical values and values of temperature and particulate matter in Los Angeles county?

We source the data from the R package *astsa* and save them as time series objects using the function ts().

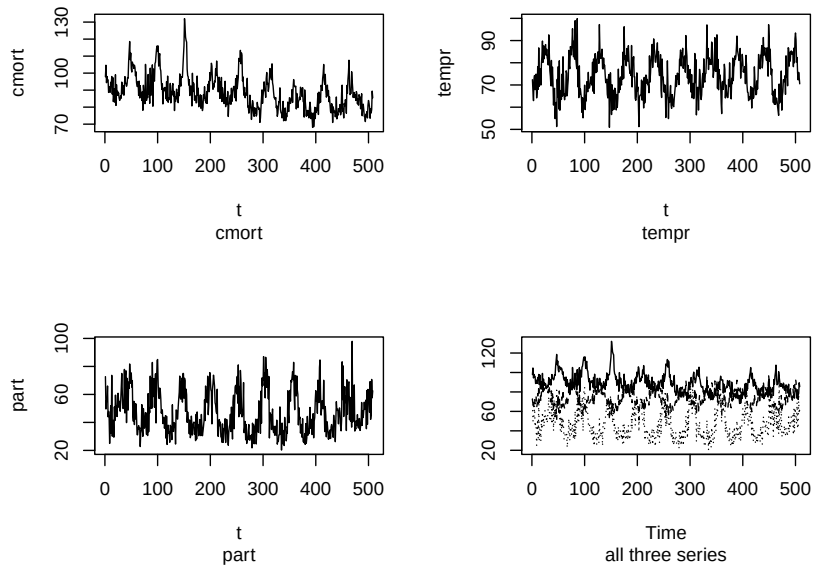

FIGURE 10.1: Time series of cardiovascular mortality, temperature, and particulates in LA county.

```
library(astsa)
library(nlme)
data(cmort); cmort=ts(cmort)
data(tempr); tempr=ts(tempr)
data(part); part=ts(part)
```

We can plot the three time series separately using the ts.plot() function. The `cmort` time series shows a slight downward linear trend over time, while `tempr` and `part` do not show any trend over time. We can also column bind all three series and plot them together as well. The plots are shown in Figure 10.1.

```
par(mfrow=c(2,2))
ts.plot(cmort, main ="Time series in LA county", sub="cmort", xlab ="t")
ts.plot(tempr, sub="tempr", xlab ="t")
ts.plot(part, sub="part",xlab ="t")
all=cbind(cmort,tempr,part)
ts.plot(all, sub="all three series",lty=1:3)
```

We can also construct a matrix scatterplot to understand the relationships between `cmort, tempr, part`. There is some contemporaneous correlations between the series. Figure 10.2 shows a quadratic relationship between `cmort` and `tempr` and a positive linear relationship between `cmort` and `part`.

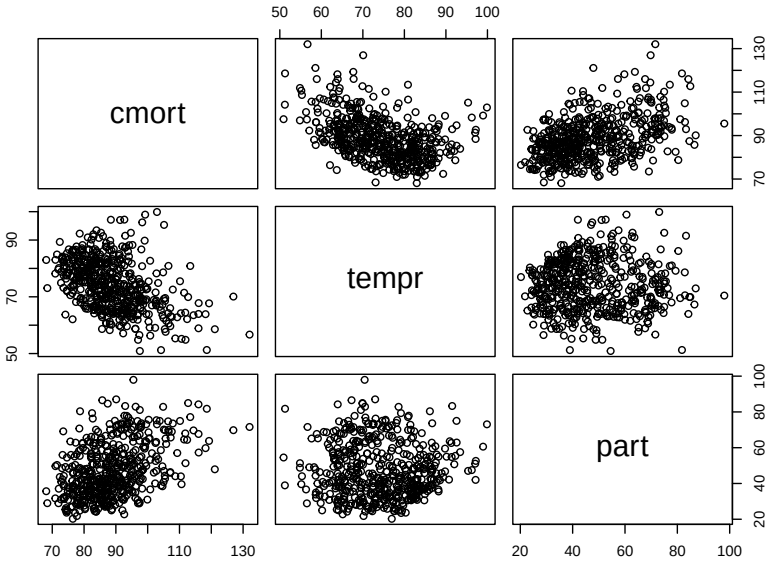

FIGURE 10.2: Matrix scatterplot for cardiovascular mortality, temperature, and particulates in LA county.

```
pairs(cbind(cmort,tempr,part))
```

We transform the `tempr` series by centering the series by subtracting the mean of the series, to get `temp`. We also form the square of `temp` to form `temp2`.

```
temp <- tempr-mean(tempr)  # center tempr t.s.
temp2 <- temp^2  # form a new var temp^2 squared temp
```

We also create a variable for the time trend $t = 1, \ldots, n$, denoted by `trend`.

```
trend <- time(cmort)
```

We can alternately use the command below to set up a trend vector.

```
trend <- c(1:length(cmort))
```

We bind the response and predictor variables into `alldat`.

```
alldat <- cbind(cmort,trend,temp,temp2,part)
```

We form a train-test split of the data, so that the training data corresponds to the first n.train = 496 observations, while the last n.test = 12 observations are held-out as test observations. We also make the train data as a data frame

```
n.train <- 496 #length(train set)
n.test <- 12 # length(test set)
train.alldat <- data.frame(alldat[1:n.train,])
test.alldat <- data.frame(alldat[(n.train+1):nrow(alldat),])
```

Suppose we decide to use the lm() to fit the MLR model for Y_t:

$$Y_t = \beta_0 + \beta_1 Z_{t,1} + \beta_2 Z_{t,2} + \ldots + \beta_p Z_{t,p} + \varepsilon_t, \tag{10.1}$$

where Y_t is cmort_t, $Z_{t,1}$ is trend_t, $Z_{t,2}$ is temp_t, $Z_{t,3}$ is temp_t^2, and $Z_{t,4}$ is part_t. Using the command lm() assumes that the errors ε_t are i.i.d. $N(0, \sigma^2)$ errors, as we did in Chapters 5 and 6.

```
regfit <- lm(cmort ~ trend +temp+temp2 +part, data = train.alldat)
```

```
> summary(regfit)

Call:
lm(formula = cmort ~ trend + temp + temp2 + part, data = train.alldat)

Residuals:
     Min       1Q   Median       3Q      Max
-19.0583  -4.3376  -0.5092   3.8657  29.1691

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 81.345481   1.131999   71.86  < 2e-16 ***
trend       -0.026370   0.002029  -13.00  < 2e-16 ***
temp        -0.469648   0.032209  -14.58  < 2e-16 ***
temp2        0.022415   0.002892    7.75 5.34e-14 ***
part         0.259149   0.019192   13.50  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.424 on 491 degrees of freedom
Multiple R-squared:  0.5909,        Adjusted R-squared:  0.5875
F-statistic: 177.3 on 4 and 491 DF,  p-value: < 2.2e-16
```

We can also summarize the ANOVA decomposition of the MLR model fit.

```
> summary(aov(regfit))

             Df Sum Sq Mean Sq F value Pr(>F)
trend         1   9764    9764   236.6 <2e-16 ***
temp          1   8755    8755   212.2 <2e-16 ***
temp2         1   3218    3218    78.0 <2e-16 ***
part          1   7523    7523   182.3 <2e-16 ***
Residuals   491  20260      41
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can use the Akaike information criterion (AIC) as an in-sample model selection criterion.

```
> AIC(regfit)
[1] 3259.672
```

When data is observed over time, the response time series Y_t and regression errors X_t are dependent over time (unlike the examples we saw in Chapter 5. We build the MLR model with autocorrelated errors or the ARMAX(P,Q) model for Y_t as a function of predictors $Z_{t,1}, Z_{t,2}, Z_{t,p}$:

$$\begin{aligned} Y_t &= \beta_0 + \beta_1 Z_{t,1} + \beta_2 Z_{t,2} + \ldots + \beta_p Z_{t,p} + X_t, \text{ where,} \\ X_t &= \sum_{j=1}^{P} \phi_j X_{t-j} + \sum_{j=1}^{Q} \theta_j \varepsilon_{t-j} + \varepsilon_t \\ &= \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_P X_{t-P} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} \ldots + \theta_Q \varepsilon_{t-Q} + \varepsilon_t \quad (10.2) \end{aligned}$$

where, ε_t are $N(0, \sigma^2)$ errors. The equation

$$X_t = \sum_{j=1}^{P} \phi_j X_{t-j} + \sum_{j=1}^{Q} \theta_j \varepsilon_{t-j} + \varepsilon_t \quad (10.3)$$

in the second line of (10.2) represents the well-known time series model called the autoregressive moving average model of order P and Q, i.e., the ARMA(P,Q) model. Here, P and Q are respectively the autoregressive (AR) and moving average (MA) orders. Of course, if the regression errors are independent, then $P = 0$ and $Q = 0$ and X_t becomes ε_t, so that (10.2) becomes the MLR model we saw in Chapters 5. Note that in the time series literature, P and Q are usually denoted by p and q. Here, we instead use P and Q so there is no confusion with p, which denotes the number of predictors $(Z_1, \ldots, Z_p)$.

When $Q = 0$ in (10.2) or (10.3), we get the simpler AR(P) model given by

$$X_t = \sum_{j=1}^{P} \phi_j X_{t-j} + \varepsilon_t. \quad (10.4)$$

We can use the auto.arima function in the R package *forecast* to fit the model (10.2), by automatically selecting the best values for the nonnegative integers P and Q.

```
library(forecast)
library(Metrics)
```

```
regmat.train <- as.matrix(train.alldat[,2:5])
armaxfit <- auto.arima(ts(train.alldat$cmort),xreg = regmat.train)
```

```
> summary(armaxfit)

Series: ts(train.alldat$cmort)
Regression with ARIMA(2,0,0) errors

Coefficients:
         ar1     ar2  intercept    trend     temp   temp2    part
      0.3846  0.4319    87.3664  -0.0298  -0.0244  0.0154  0.1620
s.e.  0.0440  0.0405     2.8359   0.0084   0.0499  0.0021  0.0276

sigma^2 = 26.47:  log likelihood = -1513.24
AIC=3042.49   AICc=3042.78   BIC=3076.14

Training set error measures:
                      ME     RMSE      MAE        MPE     MAPE
Training set -0.004576755 5.108579 3.962783 -0.3493508 4.442817
                    MASE        ACF1
Training set 0.7433428 -0.01110938
```

From the output, we see that the automatic model fitting selects $P = 2$ and $Q = 0$. The estimates of the AR coefficients are $\hat{\phi}_1 = 0.3846$ with a SE of 0.0440, and $\hat{\phi}_2 = 0.4319$ with a SE of 0.0405. To see whether these are significant, we can use z-tests for $H_0 : \phi_j = 0$ versus $H_1 : \phi_j \neq 0$, for $j = 1, 2$. The test statistics are

$$\begin{aligned} z_1 &= \hat{\phi}_1 / SE(\hat{\phi}_1) = 0.3846/0.0440 = 8.741 \\ z_2 &= \hat{\phi}_2 / SE(\hat{\phi}_2) = 0.4319/0.0405 = 10.664, \end{aligned}$$

showing that both ϕ_1 and ϕ_2 are significantly different than zero. Similarly, we can test whether the regression coefficients for trend, temp, etc. are significant. The output shows a few quantities that enable us to assess the fitted model. Of these, AIC is the Akaike information criterion, while RMSE is the square root of the MSE.

We forecast `cmort` for the test dataset.

```
regmat.test <- as.matrix(test.alldat[,2:5])
armaxfcst.all <- forecast(armaxfit, xreg = regmat.test)
```

The forecasts of `cmort` for $t = 497, \ldots, 508$, together with the 80% and 95% prediction intervals are shown below. Not that the 95% prediction intervals are wider than the 80% prediction intervals for each row. Also, the prediction intervals become wider from $t = 497$ to $t = 508$.

```
> armaxfcst
    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
497       80.29317 73.69957 86.88677 70.20913 90.37721
498       75.87219 68.80784 82.93654 65.06820 86.67618
499       83.48948 75.45714 91.52182 71.20507 95.77388
500       84.35400 75.92200 92.78600 71.45837 97.24963
501       82.46656 73.63170 91.30141 68.95481 95.97830
502       78.42964 69.34350 87.51578 64.53358 92.32570
503       82.57541 73.28129 91.86952 68.36129 96.78953
504       77.58630 68.14364 87.02897 63.14499 92.02762
505       78.41931 68.86044 87.97819 63.80027 93.03835
506       82.52452 72.87873 92.17032 67.77256 97.27649
507       80.55650 70.84374 90.26926 65.70211 95.41089
508       81.74664 71.98296 91.51032 66.81438 96.67890
```

We routinely use a few forecast evalation criteria when we model data observed over time. Let $\hat{e}_t$ denote the difference between Y_t and the fitted values $\hat{Y}_t$ for $t = 1, \ldots, n_t$ (number of observations in n.test). Formulas for these criteria are shown below. ME and MPE close to 0 and small values of MSE, MAE and MAPE indicate a better model. MAPE is a percentage, which is computed to be agnostic to the scale of the response, and is widely used for selecting the best model(s).

$$ME = \bar{\hat{e}} = \frac{1}{n_t}\sum_{t=1}^{n_t}\hat{e}_t; \tag{10.5}$$

$$MPE = \frac{100}{n_t}\sum_{t=1}^{n_t}\left(\hat{e}_t/Y_t\right); \tag{10.6}$$

$$MSE = \frac{1}{n_t}\sum_{t=1}^{n_t}\hat{e}_t^2; \tag{10.7}$$

$$MAE = \frac{1}{n_t}\sum_{t=1}^{n_t}|\hat{e}_t|; \tag{10.8}$$

$$MAPE = \frac{100}{n_t}\sum_{t=1}^{n_t}|\hat{e}_t/Y_t|. \tag{10.9}$$

We can evaluate the forecast validity criteria ME, MPE, MSE, MAE, and MAPE using the setup shown below. We pull out the point forecasts for the test period.

```
>(armaxfcst <- armaxfcst.all$mean)
Time Series:
Start = 497
End = 508
Frequency = 1
 [1] 80.29317 75.87219 83.48948 84.35400 82.46656 78.42964 82.57541 77.58630 78.41931 82.52452
[11] 80.55650 81.74664
```

We next compute forecast errors as the difference between the held-out test responses and the point forecasts from the fitted model.

```
> (armaxfcsterr <- test.alldat$cmort - armaxfcst)
Time Series:
Start = 497
End = 508
Frequency = 1
 [1] -2.9331728 -2.2421909 -2.3194768 -0.4440016 -0.1065568  1.3103604 -9.1154078  1.4436955
 [9] -1.8593130 -4.0045237  8.8734979  3.7433641
```

We then evaluate the five forecast evaluation criteria using their formula shown in (10.5)-(10.9). Note that the forecast bias is reasonably small and the MAPE is only 4%. This seems to be a reasonable model for `cmort`.

```
> (me.armaxfcst <- mean(armaxfcsterr))           #ME
[1] -0.6378105
> (mpe.armaxfcst <- 100*(mean(armaxfcsterr/test.alldat$cmort))) # MPE
[1] -1.043255
> (mse.armaxfcst <- sum(armaxfcsterr**2)/nrow(test.alldat))      # MSE
[1] 18.19636
> (mae.armaxfcst <- mean(abs(armaxfcsterr)))               # MAE
[1] 3.19963
> (mape.armaxfcst <- 100*mean(abs(armaxfcsterr/test.alldat$cmort))) # MAPE
[1] 4.005098
```

Bibliography

Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, New Jersey: John Wiley & Sons, 2^{nd} ed.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, New York: John Wiley & Sons.

Bloch, C. and Willig, M. (2006), "Context-dependence of long-term responses of terrestrial gastropod populations to large-scale disturbance," *Journal of Tropical Ecology*, 22, 111–122.

Breiman, L. (2001), "Random forests," *Machine Learning*, 45, 5–32.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (2017), *Classification and Regression Trees*, Routledge.

Chatterjee, S. and Hadi, A. S. (1988), *Sensitivity Analysis in Linear Regression*, New York: John Wiley & Sons.

Cleveland, W. S. (1985), *The Elements of Graphing Data*, Monterey, CA: Wadsworth.

Conover, W. J. (1999), *Practical Nonparametric Statistics*, New York: John Wiley & Sons, 3^{rd} ed.

Conover, W. J. and Iman, R. L. (1981), "Rank transformations as a bridge between parametric and nonparametric statistics," *The American Statistician*, 35, 124–129.

Cook, R. D. and Weisberg, S. (1983), "Diagnostics for heteroscedasticity in regression," *Biometrika*, 70, 1–10.

D'Agostino, R. B. and Stephens, M. A. (1986), *Goodness-of-fit Techniques*, Marcel Dekker.

Deng, X., Huang, J., Uchida, E., Rozelle, S., and Gibson, J. (2011), "Pressure cookers or pressure valves: do roads lead to deforestation in China?" *Journal of Environmental Economics and Management*, 61, 79–94.

Galton, F. (1886), "Regression towards mediocrity in hereditary stature," *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15, 246–263.

Glynn, E. (2005), "Chart of R colors," *Stowers Institute of Medical Research*.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2, Springer.

Hoaglin, D. C. and Welsch, R. E. (1978), "The hat matrix in regression and ANOVA," *The American Statistician*, 32, 17–22.

Hoerl, A. E. and Kennard, R. W. (1970), "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, 12, 55–67.

Hollander, M., Wolfe, D. A., and Chicken, E. (2013), *Nonparametric Statistical Methods*, New York: John Wiley & Sons, 3^{rd} ed.

Huber, P. J. (1981), *Robust Statistics*, New York: John Wiley & Sons.

Koch, G. and Edwards, S. (1988), *Clinical efficiency trials with categorical data. In K. E. Peace (ed.), Biopharmaceutical Statistics for Drug Development.*, Marcel Dekker, New York.

Kruskal, W. H. and Wallis, W. A. (1952), "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, 47, 583–621.

Kutner, M. H., Nachtsheim, C. J., Neter, J., and Li, W. (2005), *Applied Linear Statistical Models*, McGraw Hill/Irwin, 5^{th} ed.

Mallows, C. L. (1973), "Some comments on C_p," *Technometrics*, 15, 661–675.

Mann, H. and Whitney, D. (1947), "On a test of whether one of two random variables is stochastically larger than the other," *Annals of Mathematical Statistics*, 18, 50–60.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, London: Chapman & Hall/CRC, 2^{nd} ed.

Montgomery, D. C. (2017), *Design and Analysis of Experiments*, New York: John Wiley & Sons, 9^{th} ed.

Moro, S., Laureano, R., and Cortez, P. (2011), "Using data mining for bank direct marketing: An application of the CRISP-DM methodology," .

Muralidharan, K. and Prakash, N. (2017), "Cycling to School: Increasing Secondary School Enrollment for Girls in India," *American Economic Journal: Applied Economics*, 9, 321–50.

Pruzek, R. M. and Helmreich, J. E. (2009), "Enhancing dependent sample analyses with graphics," *Journal of Statistics Education*, 17.

Rahlf, T. (2017), *Data Visualization with R: 100 Examples*, Springer-Verlag.

Ramsey, F. L. and Schafer, D. W. (2013), *The Statistical Sleuth: A Course in Methods of Data Analysis*, Boston: Brooks/Cole, 3^{rd} ed.

Shapiro, S. S. and Wilk, M. B. (1965), "An analysis of variance test for normality," *Biometrika*, 52, 591–611.

Snedecor, G. and Cochran, W. G. (1989), *Statistical Methods*, Blackwell Publishing, 8th ed.

Welch, B. L. (1947), "The generalization of Student's problem when several different population variances are involved," *Biometrika*, 34, 28–35.

Wilcoxon, F. (1949), *Some Rapid Approximate Statistical Procedures*, Stanford.

Zou, H. and Hastie, T. (2005), "Regularization and variable selection via the elastic net," 67, 301–320.