THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

STAT1301
Advanced Analysis of Scientific Data

by the
Statistics Group of
the School of Mathematics and Physics

November 4, 2020

CONTENTS

PREFACE

These notes are intended for first-year students who would like to more fully understand the logical reasoning and computational techniques behind statistics. Our intention was to make something that would be useful as a self-learning guide for advanced first-year students, providing both a sound theoretical foundation of statistics as well as a comprehensive introduction to the statistical language R.

STAT1301 is the advanced version of STAT1201, and will explain several concepts on a deeper level than is feasible in STAT1201. Our guiding principle was that it is just as important to know the "why" as the "how". To get the most use out of these notes it is important that you carefully read the whole story from beginning to end, annotate the notes, check the results, make connections, do the exercises, try the R programs, visit the lectures, and *most importantly*, ask questions about things you do not understand. If you are frightened by the maths, it is good to remember that the mathematics is there to make life *easier*, not harder. Mathematics is the language of science, and many things can be said more precisely in one single formula, definition, or with a simple artificial example, than is possible in many pages of verbose text. Moreover, by using mathematics it becomes possible to build up statistical knowledge from very basic facts to a high level of sophistication. Of course in a first-year statistics course, however advanced, it is not possible to cover all the knowledge that has been built up over hundreds of years. We will sometimes only give a glimpse of new things to discover, but we have to leave something for your future studies! Knowing the mathematical reasoning behind statistics avoids using statistics only as a black box, with many "magic" buttons. Especially when you wish to do further research it is important to be able to develop your own statistical reasoning, separate from any statistical package.

The material in these notes was partly based on:

- Dirk P. Kroese and Joshua C.C. Chan (2014). *Statistical Modeling and Computation*, Springer, New York.

- Pierre Lafaye de Micheaux, Rémy Drouilhet, and Benoit Liquet (2014). *The R Software: Fundamentals of Programming and Statistical Analysis*, Springer, New York.

We will introduce the topics in these notes in a linear fashion, starting a brief introduction to data and evidence in Chapter 1. In Chapter 2 we describe how to summarize and visualize data. We will use the statistical package `R` to read and structure the data and make figures and tables and other summaries. Chapter 3 is about *probability*, which deals with the modeling and understanding of randomness. We will learn about concepts such as random variables, probability distributions, and expectations. Various important probability distributions in statistics, including the *binomial* and *normal* distributions, receive special attention in Chapter 4. We then continue with a few more probability topics in Chapter 5, including multiple random variables, independence, and the central limit theorem. At the end of that chapter we introduce some simple statistical models. After this chapter, we will have built up enough background to properly understand the statistical analysis of data. In particular, we discuss *estimation* in Chapter 6 and and *hypothesis testing* in Chapter 7, for basic models. The remaining chapters consider the statistical analysis of more advanced models, including *regression* (Chapter 8) and *analysis of variance* (Chapter 9), both of which are special examples of a *linear model* (Chapter 10). The final Chapter 11 touches on additional statistical techniques, such as goodness of fit tests, logistic regression, and nonparametric tests. The `R` program will be of great help here. Appendix A gives a short introduction to `R`.

Brisbane,

Statistics Group
School of Mathematics and Physics
The University of Queensland
November 4, 2020

CHAPTER 1

Data and Evidence

The aim of this chapter is to give a short introduction to the statistical reasoning that we will be developing during this course. We will discuss the typical steps taken in a statistical study, emphasize the distinction between observational and designed statistical experiments, and introduce you to the language of hypothesis testing.

Statistics is an essential part of science, providing the language and techniques necessary for understanding and dealing with chance and uncertainty in nature. It involves the design, collection, analysis, and interpretation of numerical data, with the aim of extracting patterns and other useful information.

1.1 Statistical Studies

In science the typical steps that are taken to answer a real-life research question are:

Steps for a Statistical Study

1. Design an experiment to give information about the research question.
2. Conduct this experiment and collect the data.
3. Summarize and visualize the observed data.
4. Make a statistical model for the data.
5. Analyse this model and make decisions about the model based on the observed data.
6. Translate decisions about the model to decisions and predictions about the research question.

To fully understand statistics it is important that you follow the reasoning behind the steps above. Let's look at a concrete example.

■ **Example 1.1 (Biased Coin)** Suppose we have a coin and wish to know if it is fair — that is, if the probability of Heads is 1/2. Thus the research questions here is: is the coin fair or biased? What we could do to investigate this question is to conduct an experiment where we toss the coin a number of times, say 100 times, and observe when Heads or Tails appears. The data is thus a sequence of Heads and Tails— or we could simply write a 1 for Heads and 0 for Tails. We thus have a sequence of 100 observations, such as 1 0 0 1 0 1 0 0 1 . . . 0 1 1. These are our data. We can visualize the data by drawing a bar graph such as in Figure 1.1.

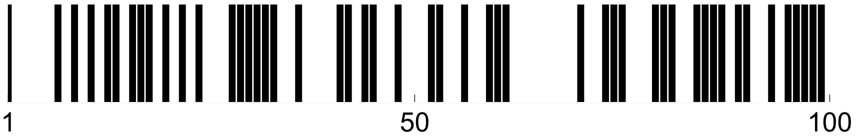


Figure 1.1: Outcome of an experiment where a fair coin is tossed 100 times. The dark bars indicate when Heads (=1) appears.

Think about the pros and cons of this plot. If we are only interested in the biasedness of the coin, then a simple chart that shows the total numbers of Heads and Tails would suffice, as knowing exactly where the Heads or Tails appeared is irrelevant. Thus, we can *summarize* the data by giving only the total number of Heads, x say. Suppose we observe $x = 60$. Thus, we find 60 Heads in 100 tosses. Does this mean that the coin is not fair, or is this outcome simply due to chance?

Note that if we would repeat the experiment with the same coin, we would likely get a different series of Heads and Tails (see Figure 1.2) and therefore a different outcome for x.

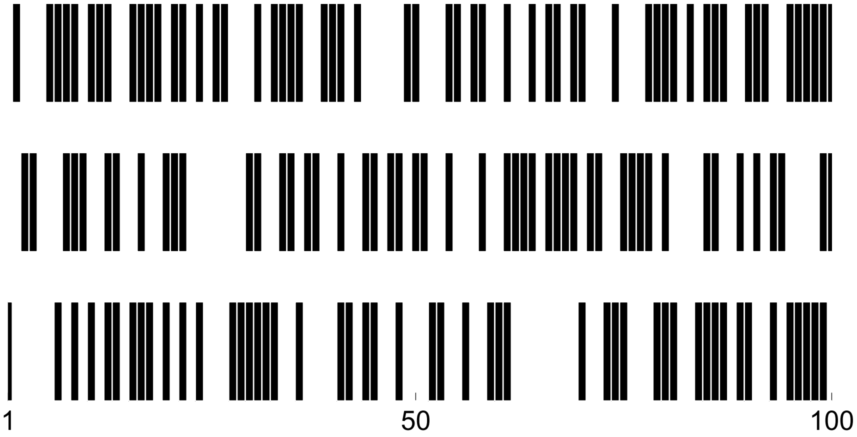


Figure 1.2: Outcomes of three different experiments where a fair coin is tossed 100 times.

We can now reason as follows (and this is crucial for the understanding of statistics): if we denote by X (capital letter) the total number of Heads (out of 100) that we will observe *tomorrow*, then we can view $x = 60$ as just one possible outcome of the *random variable* X. To answer the question whether the coin is fair, we need to say something about how likely it is that X takes a value of 60 or more for a fair coin. To calculate probabilities and other quantities of interest involving X we need an appropriate statistical *model* for X, which tells us how X behaves probabilistically. Using such a model we can calculate, in particular, the probability that X takes a value of 60 or more, which is about 0.028 for a fair coin — so quite small. However, we *did* observe this quite unlikely event, providing reasonable evidence that the coin may not be fair.

Interestingly, we don't actually need any formulas to calculate this probability. Computers have become so fast and powerful that we can quickly approximate probabilities via *simulations*. Simulating this coin flip experiment in R is equivalent to sampling 100 times (with replacement) from a "population" $\{0, 1\}$ and counting how many 1s there are. In R:

```
> coin = c(0,1)

> sample(coin, 100, replace = T)

   [1] 0 0 1 1 0 1 1 1 1 0 0 0 1 0 1 0 1 1 0 1 1 0 1 0 0 0 0 0
  [29] 1 0 0 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 0 0 1 1 0 1 0 0 0 0
  [57] 0 1 1 0 1 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 1 0 1 1 0 1 0 0
  [85] 1 0 1 0 0 0 0 0 1 1 1 1 0 0 1 1

> sum(sample(coin, 100, replace = T))

  [1] 48
```

In this case we had 48 Heads out of 100. If we repeat it two times:

```
> sum(sample(coin, 100, replace = T))

  [1] 54

> sum(sample(coin, 100, replace = T))

  [1] 38
```

Now, let's repeat this 1000 times and save the output in a variable:

```
> data.we.could.have.seen
       = replicate(1000, sum(sample(coin, 100, replace = T)))
```

```
> data.we.could.have.seen

   [1] 43 47 56 54 49 45 46 51 41 47 48 44 54 53 43 54 46 49
  [19] 48 44 47 52 53 39 44 52 53 45 52 57 49 54 48 56 42 47
  [37] 42 46 44 47 49 46 51 53 59 57 50 45 51 55 50 53 60 53
  ...
 [973] 45 49 42 53 54 51 56 46 49 48 53 46 55 37 47 49 51 54
 [991] 50 49 49 50 57 35 44 49 45 52

> data.we.could.have.seen >= 60

   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [10] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [19] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [28] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
  [46] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
  ...
 [991] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [1000] FALSE

> sum(data.we.could.have.seen >= 60)

  [1] 21

> mean(data.we.could.have.seen >= 60)

  [1] 0.021
```

So, *without any knowledge of probability*, we have found that the probability that X takes a value of 60 or more is approximately 0.021 for a fair coin. If the coin is indeed fair, then what we have witnessed was quite a rare event — entirely possible, but rather rare. We can either:

- accept that the coin is fair and that we just happened to see a rather rare occurrence; or

- do not accept that we've been so unlucky, and instead suspect that the coin is rigged.

■

You have already carried out your first scientific study and statistical hypothesis test! The rest of this course will build up your foundational knowledge in probability and statistics so that you can tackle a wider range of research questions and data types.

1.2 Data

Data comes in many shapes and forms, but is often represented as a spreadsheet in a "standard format", where columns represent features such as height, gender, and income, and rows represent individuals or units.

The data in a spreadsheet could be the result of an **observational study**, where we have no control over each feature (corresponding to a column in a spread sheet). A typical example is survey data. If we would repeat the whole study, the values in all columns would change.

Alternatively, The data in a spreadsheet could be the result of an **designed experiment**, where certain experimental conditions (features) are controlled (fixed) to reduce unwanted variability in the measurements. If we would repeat the whole experiment, the experimental feature columns would stay the same.

Regardless of whether we have data from an observational or designed experiment, some measurements, would *change* if the data were collected again. There may be various causes of variability/randomness in measurements. For example, in height data the main source of variability is the natural diversity of heights in a population. Another source of variability is the measurement variability (how accurately we can measure each height). Later on in this course we will consider statistical models that aim to explain the variability in the data. For example, we could try to explain the variability in heights not only via the natural variability in the population, but also taking into account variables such as gender, ethnicity, and shoe size. Any remaining variability that cannot be explained by the model is called the **residual variability**. A good model has a small residual error and predicts new data well.

1.3 Designed Experiment: Alice's Caffeine Data

In this section we consider a simple designed experiment, to which we will come back several times during this course, and which we will refer to as Alice's Caffeine Data experiment. Alice's research question is: does drinking caffeinated cola increase the heart rate compared to drinking decaf cola? To answer this question, she designed the following experiment:

- Measure the heart rates of 20 friends as subjects, using a pulse meter.
- Give 10 friends 250mL of *caffeinated* Diet Coke while the other 10 friends are given 250mL of *caffeine-free* (decaf) Diet Coke.
- Wait half an hour after the drink, and then measure the heart rates again.
- Record the *difference* in heart rates for each subject.

Even for such a seemingly simple study, there are a lot of design issues to think about. Let us discuss a few points.

1. Alice chose 20 subjects in her study. Is this a sufficient sample size? There are many considerations for choosing a “good” sample size. In general, the sample size is determined by (a) the size of the effect that we are trying to detect and (b) the variability in the data. If the data has little variability, it may suffice to only use a small sample size to detect a certain effect. In contrast, if the data exhibits a large amount of variability it may require a large sample size to detect any effects, especially if they are small. However, bigger sample sizes do not always make for better experiments. Running an overly large study often leads to poor quality of data, as it is difficult to enforce compliance to the study protocols at all levels of the study. A large sample size may also be impractical, potentially dangerous (e.g., for experimental medical treatments), costly, or time-consuming. Think of all the cola Alice would have to buy for a study with 1000 individuals!

2. Alice chose her friends as test subjects. Is that fair? Let us go back to the research question: to detect if caffeinated cola increases the heart rate. If the population of interest is not just Alice’s friends, but the general population, then choosing the subjects within her circle of friends may introduce a sampling bias. For example, suppose that the effect of caffeine would depend on age and gender. If most of Alice’s friends would be between 19 and 22 years old and female, the sample group would no longer be representative of the general population. Any conclusions from this study would pertain to the smaller population of people that are similar to Alice’s friends. But maybe the effect of caffeine does not depend on factors such as age, weight, gender, or the subject’s cola drinking behaviour, and then the conclusions might be applicable to a larger population.

   There is another possible source of bias in Alice’s experiment. Recall that she gives 10 friends caffeinated Diet Coke and 10 friends decaf Diet Coke. How are the two groups chosen? Perhaps she divides the groups into 10 males and 10 females. But this could lead to a bias in the results, if the difference in heart rate would depend on gender. To avoid any bias in the group selection, we can randomly select the treatment and placebo group, by using the random number generator of `R`, for example. Such a **randomization** process is an important ingredient in many designed experiments.

   Another issue is whether Alice’s friends know if they are getting caffeinated or decaf cola. This may influence the measurements (increase in heart rate) through the **placebo effect**: friends that know they have consumed caffeinated Diet Coke may increase their heart rate beyond what would have happened if they knew their cola did not have caffeine. This is especially pertinent in **comparative experiments** of new medical treatments. In such experiments it is customary that one portion (say one half) of the subjects is offered the new treatment and the other half receives a placebo. In a **blind experiment** the subjects do not know whether they receive the treatment or a placebo. Even in this case, the experimenter may inadvertently influence the outcome if they know which subjects are assigned the treatments and placebos. To remove also this bias, the gold-standard procedure is to employ a **double-blind experiment**, where the experimenters do not know how the treatments and placebos are distributed over the subjects.

3. Why did Alice wait half an hour after consuming drinks before measuring the pulse rates again? In this experiment, Alice used “subject-specific” knowledge. Namely, she did a thorough literature search on the average time it takes for humans to absorb and metabolize caffeine in drink form — most sources claimed this to be around 20–30 minutes.

4. Why did Alice choose to compare Diet Coke with decaf Diet Coke instead of the regular (non-Diet) cola? The reason is given in Figure 1.3: the only difference between caffeinated Diet Coke and decaf Diet Coke is the amount of caffeine. In contrast, caffeinated Regular Coke and decaf Regular Coke have, in addition to the caffeine content, different energy, protein, carbohydrate, and sodium contents. So a change in heart rate could be the result of the sugar content, for example, rather than caffeine content. We say that decaf Diet Coke serves as a suitable **control** for caffeinated Diet Coke, as their only difference is the caffeine content.

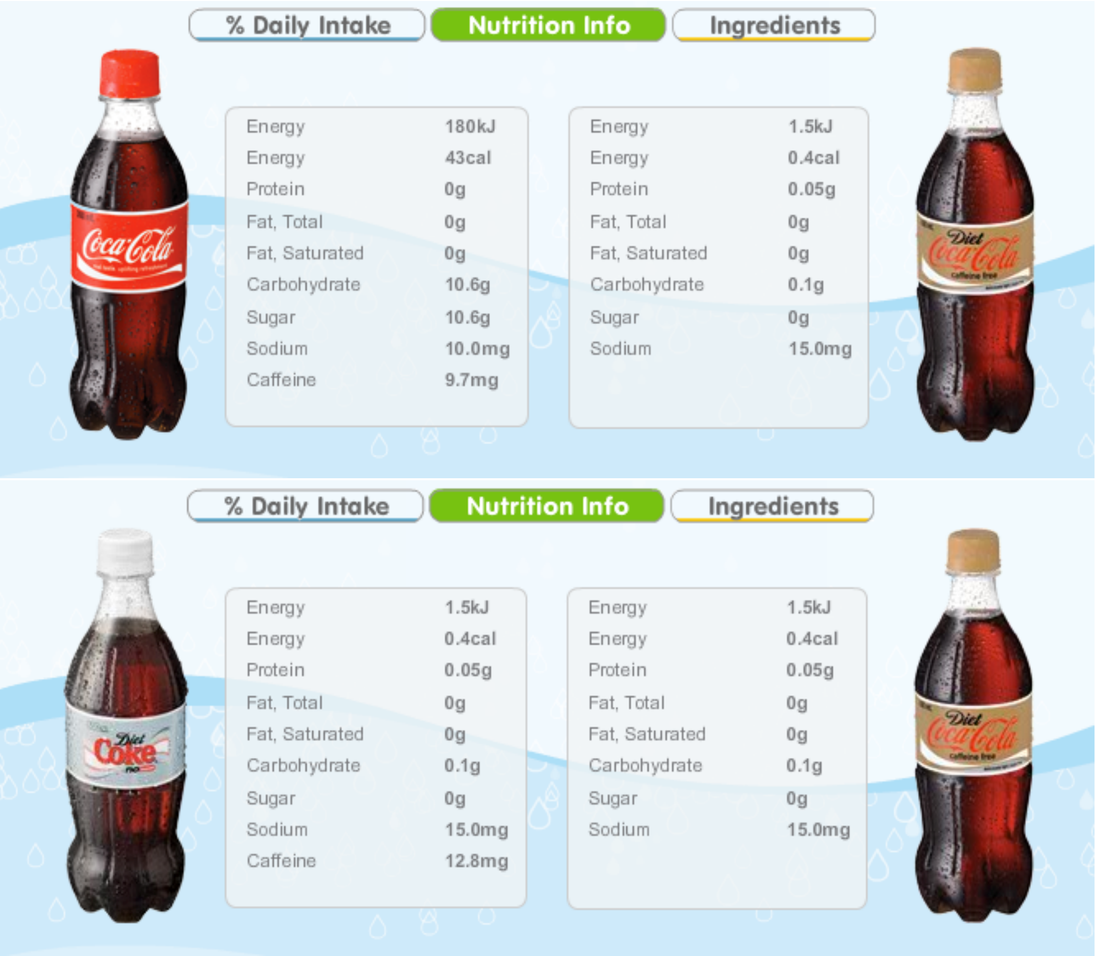
Later on in this course we will have a closer look at designed experiments and how to analyse them via Analysis of Variance (ANOVA) methods. But for now let us examine Alice’s data and introduce some experimental design terminology on the way. Alice’s experiment involves actively applying treatments to subjects and observing their responses. An experimental **treatment** is a combination of factors at different levels. The variables describing the treatments are the **explanatory variables** in the study. The **response** from an experiment is the variable/s of interest.

For Alice’s experiment, the response variable is the change in pulse rate, and the explanatory variable is the caffeine content, which is considered at two levels (yes and no). The resulting changes in pulse rate are given in Table 1.1:

Table 1.1: Changes in pulse rate for Alice’s caffeine experiment.

Caffeinated	17	22	21	16	6	−2	27	15	16	20
Decaf	4	10	7	−9	5	4	5	7	6	12

We mentioned in Section 1.2 that when using software such as R to analyse and display data it is important that the data is represented/stored in a standard format. Table 1.1 is not in a standard format. To convert it to standard format, we should store the pulse beat changes of all 20 subjects (Alice’s friends) in a single column — called `pb`, for example. And a second column, `Caffeine`, indicates whether a subject receives the caffeinated or decaf cola. Of course such a table is very tall and skinny and does not present as well on paper as Table 1.1.



Energy	180kJ
Energy	43cal
Protein	0g
Fat, Total	0g
Fat, Saturated	0g
Carbohydrate	10.6g
Sugar	10.6g
Sodium	10.0mg
Caffeine	9.7mg

Energy	1.5kJ
Energy	0.4cal
Protein	0.05g
Fat, Total	0g
Fat, Saturated	0g
Carbohydrate	0.1g
Sugar	0g
Sodium	15.0mg

Energy	1.5kJ
Energy	0.4cal
Protein	0.05g
Fat, Total	0g
Fat, Saturated	0g
Carbohydrate	0.1g
Sugar	0g
Sodium	15.0mg
Caffeine	12.8mg

Energy	1.5kJ
Energy	0.4cal
Protein	0.05g
Fat, Total	0g
Fat, Saturated	0g
Carbohydrate	0.1g
Sugar	0g
Sodium	15.0mg

http://www.coca-cola.com.au/ourdrinks/nutrition-comparison-tool.jsp

Figure 1.3: Nutritional information for caffeinated/decaf Regular Coke (top) and caffeinated/decaf Diet Coke (bottom).

Do the results in Table 1.1 provide any *evidence* that caffeine increases pulse rate? Let us now go through the same 6 steps of a statistical study as in Section 1.1. The first two steps (designing the study and collecting the data) have already been discussed above.

Step 3 is about visualizing and summarizing data. Figure 1.4 shows a possible visualization of the data in a so-called stripplot.

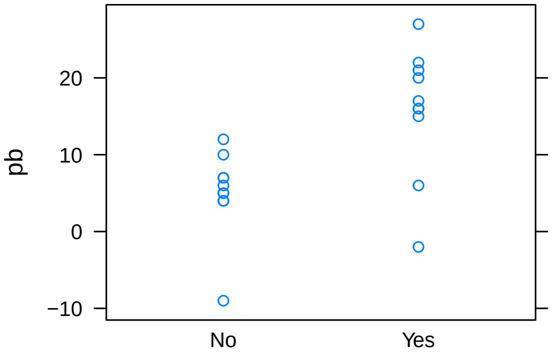


Figure 1.4: A visualization of Alice's caffeine data.

It was made using the following code:

```
> alice = read.csv("alice.csv")
> library(lattice)
> stripplot(pb ~ Caffeine, data=alice)
```

Two key summaries of the data are:

- The mean (i.e., average) increase in pulse rate for the decaf group is 5.1 bpm.
- The mean increase in pulse rate for caffeine group is 15.8 bpm.

The *group difference* is thus $15.8 - 5.1 = 10.7$ bpm. Is this evidence that caffeine increases pulse rate? In other words, if we summarize our data via the group difference, what evidence is there that the group difference 10.7 was due to the effect of the caffeine presence, rather than this happening by chance?

Compare this with the coin toss experiment in Section 1.1, where we summarized the data via the total number of Heads. There, we considered the behaviour of the total number of heads X for a fair coin, and compared it with the observed number of heads $x = 60$. In the next section we will carry out a similar comparison for the Alice data in the context of *hypothesis testing*.

1.4 Hypothesis Testing

In this section we introduce the language of hypothesis testing and how to make decisions using hypothesis tests.

Recall Alice's caffeine experiment, where we observed a group difference of 10.7 beats per minute between the caffeine and decaf groups. We have two explanations for this difference:

1. Caffeine really has *no effect on pulse rate* and the observed group difference of 10.7 was just *due to the chance variability* in pulse rates.

2. The group difference of 10.7 arose because caffeine *does increase pulse rate*.

Suppose that the first explanation is correct — i.e., caffeine really has no effect. Instead of two different groups we have really made 20 observations of the *same* process. That is, the changes in pulse rate for each subject would have been the same regardless of which treatment they were given. Only in the selection of the control/treatment groups did the observations happen to end up in the groups that they did. If we distribute the 20 measurements randomly amongst the two groups, how likely is it that we end up with a *group difference* of 10.7 or more?

Suppose we want to randomly select 10 of the 20 subjects to be in the caffeinated group (with the other 10 going to the decaf group). Let's label the first subject by "1", second subject by "2", third subject by "3", . . . , and so on. We can do this in R via:

```
> subjects = 1:20
```

We can then sample 10 numbers from this list at random, without replacement, via:

```
> sample(subjects, 10, replace = F)
```

```
 [1]  4 14 11  3 16 15  2 18  6  7
```

We have now simulated *one particular* randomization that could have been observed. How many possible randomizations are there? A total of $\binom{20}{10} = 184,756$ — ☞ 49 you will learn to count this in Section 3.4. For our original data, the 10 subjects in the caffeinated group were chosen as 1,2,. . . ,10. We need a fast way to compute the group difference for any group. To do this, we first store the original data in a variable `alice`:

```
> alice = c(17,22,21,16,6,-2,27,15,16,20,4,10,7,-9,5,4,5,7,6,12)
```

and then define a function to calculate the group differences (i.e., differences in the sample means):

```
> diff.mean = function(data){
    mean(data[1:10]) - mean(data[11:20]) }
```

Let us check for the original data:

```
> diff.mean(alice)
```

```
10.7
```

Now shuffle the observations randomly:

```
> s = sample(alice)
> print(s)


 7  7 -9 -2 17 15  4 12 21  5 10  4  6 20 16 22  5 16  6 27
```

and then get the difference in sample means:

```
> diff.mean(s)


  -5.5
```

Repeat this process many times, e.g., 200,000 times, and count how many reshuffles lead to a group difference greater than the observed 10.7.

```
> sum(replicate(200000,diff.mean(sample(alice))>=10.7))


410
```

Or as a proportion of the total number of replications:

```
> mean(replicate(200000,diff.mean(sample(alice))>=10.7))


0.00205
```

We see that this number is very small, indicating that it is very unlikely that we obtain a group difference of 10.7 or more by reshuffling only. However, we *did* observe this group difference, giving strong evidence against the hypothesis that caffeine has no effect on the pulse rate.

In the language of hypothesis testing, we conducted a specific statistical test called a **randomization test**. The first explanation we gave is called the **null hypothesis**, H_0, of the test ("nothing is really happening"). The second explanation (the statement we wish to show) is the **alternative hypothesis**, H_1. The function of the data on which we base our conclusion is called the **test statistic**; in this case the group difference. The *probability of obtaining such unusual data* (or even more unusual) under the null hypothesis is the **P-value** of the test (here, $p = 0.002$).

- A small P-value suggests the null hypothesis may be wrong, giving evidence for the alternative hypothesis.

- A large P-value suggests that the data are consistent with the null explanation, giving inconclusive evidence of an effect.

Figure 1.5 illustrates the strength of evidence, expressed in words, associated with a P-value.

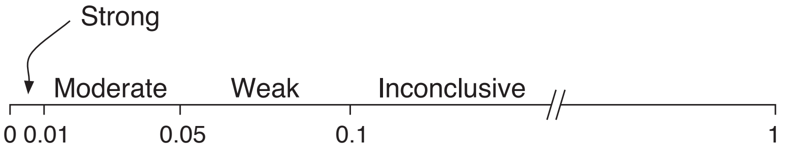


Figure 1.5: Strength of evidence for a P-value.

If a decision is required regarding H_0 or H_1, then a threshold for evidence needs to be set. For example, if we find a P-value less than 0.05 we might say that "the results were *significant* at the 5% level" and write "$p < 0.05$". This practice is overused in scientific publications where a dichotomy is usually not required. It is better to simply report actual P-value where possible (e.g., "$p = 0.002$").

CHAPTER 2

Describing Data

This chapter describes how to structure data, calculate simple numerical summaries and draw standard summary plots.

2.1 Data as a Spreadsheet

Data is often stored in a table or spreadsheet. A statistical convention is to denote variables as columns and the individual items (or units) as rows. It is useful to think of three types of columns in your spreadsheet:

1. The first column is usually an identifier column, where each unit/row is given a unique name or ID.

2. Certain columns can correspond to the design of the experiment, specifying for example to which experimental group the unit belongs, after using a randomization procedure.

3. Other columns represent the observed measurements of the experiment. Usually, these measurements exhibit *variability*; that is, they would change if the experiment were to be repeated.

In this course, we will store data in CSV (**Comma Separated Values**) format. That is, the data is given as a text file where, as the name suggests, values are separated by commas. You can open and create a CSV file/spreadsheet via Excel or, better, via `R`. It will be convenient to illustrate various data concepts by using the CSV file `nutrition_elderly.csv`, which contains nutritional measurements of thirteen variables (columns) for 226 elderly individuals (rows) living in Bordeaux, who were interviewed in the year 2000 for a nutritional study (see Table 2.1 for a description of the variables).

Table 2.1: Description of the variables in the nutritional study

Description	**Unit** or **Coding**	**Variable**
Gender	1=Male; 2=Female	`gender`
Family status	1=Single 2=Living with spouse 3=Living with family 4=Living with someone else	`situation`
Daily consumption of tea	Number of cups	`tea`
Daily consumption of coffee	Number of cups	`coffee`
Height	Cm	`height`
Weight (actually: mass)	Kg	`weight`
Age at date of interview	Years	`age`
Consumption of meat	0=Never 1=Less than once a week 2=Once a week 3=2/3 times a week 4=4/6 times a week 5=Every day	`meat`
Consumption of fish	Idem	`fish`
Consumption of raw fruits	Idem	`raw_fruit`
Consumption of cooked fruits and vegetables	Idem	`cooked_fruit_veg`
Consumption of chocolate	Idem	`chocol`
Type of fat used for cooking	1=Butter 2=Margarine 3=Peanut oil 4=Sunflower oil 5=Olive oil 6=Mix of vegetable oils (*e.g.* Isio4) 7=Colza oil 8=Duck or goose fat	`fat`

You can import the data into `R` using for example the function **`read.csv`**, as in:

```
> nutri = read.csv("nutrition_elderly.csv",header=TRUE)
```

This causes **`nutri`** to be stored as a so-called `data.frame` object in `R` — basically a list of columns. To check the type of your object you can use the `R` function **`class`**.

```
> class(nutri)
 [1] "data.frame"
```

The `R` function **`head`** gives the first few rows of the data frame, including the variable names.

```
> head(nutri)
  gender situation tea coffee height weight age meat fish
1      2         1   0      0    151     58  72    4    3
2      2         1   1      1    162     60  68    5    2
3      2         1   0      4    162     75  78    3    1
4      2         1   0      0    154     45  91    0    4
```

```
5      2         1   2      1    154     50  65    5    3
6      2         1   2      0    159     66  82    4    2
  raw_fruit cooked_fruit_veg chocol fat
1         1                4      5   6
2         5                5      1   4
3         5                2      5   4
4         4                0      3   2
5         5                5      3   2
6         5                5      1   3
```

The names of the variables can also be obtained directly via the function `names`, as in `names(nutri)`. This returns a list of all the names of the data frame. The data for each individual column (corresponding to a specific name) can be accessed by using R's *list*$*name* construction. For example, `nutri$age` gives the vector of ages of the individuals in the nutrition data set. ☞ 196

Note that all the entries in `nutri` are *numerical* (that is, they are numbers). However, the *meaning* of each number depends on the respective columns. For example, a 1 in the "gender" column means here that the person is male (and 2 for female), while a 1 in the "fish" column indicates that this person eats fish less than once a week. Note also that it does not make sense to take the average of the values in the "gender" column, but it makes perfect sense for the "weights" column. To better manipulate the data it is important to specify exactly what the structure is of each variable. We discuss this next.

2.2 Structuring Variables According to Type

We can generally classify the measurement variables into two types: *quantitative* and *qualitative* (also called categorical). For quantitative variables we can make a distinction[1] between continuous quantitative and discrete quantitative variables:

Continuous quantitative variables represent measurements that take values in a continuous range, such as the height of a person or the temperature of an environment. Continuous variables capture the idea that measurements can always be made more precisely.

Discrete quantitative variables have only a small number of possibilities, such as a count of some outcomes. For example in the data frame `nutri`, the variable `tea` (representing the daily number of cups of tea) is a discrete quantitative variable.

For qualitative variables (often called **factors**), we can distinguish between nominal and ordinal variables:

[1] As all measurements are recorded to a finite level of accuracy, one could argue that all quantitative variables are discrete. The actual issue is how we *model* the data, via discrete and continuous (random) variables. Random variables and their probability distributions will be introduced in Chapter 3.

Nominal factors represent groups of measurements without order. For example, recording the sex of subjects is essentially the same as making a group of males and a group of females.

Ordinal factors represent groups of measurement that do have an order. A common example of this is the age group someone falls into. We can put these groups in order because we can put ages in order.

■ **Example 2.1 (Variable Types)** The variable types for the data frame `nutri` are given in Table 2.2.

Table 2.2: The variable types for the data frame `nutri`

Nominal	`gender`, `situation`, `fat`
Ordinal	`meat`, `fish`, `raw_fruit`, `cooked_fruit_veg`, `chocol`
Discrete quantitative	`tea`, `coffee`
Continuous quantitative	`height`, `weight`, `age`

■

Initially, all variables in `nutri` are identified as quantitative, because they happened to be entered as numbers[2]. You can check the type (or structure) of the variables with the function `str`.

```
> str(nutri)
 'data.frame':       226 obs. of  13 variables:
$ gender          : int  2 2 2 2 2 2 2 2 2 2 ...
$ situation       : int  1 1 1 1 1 1 1 1 1 1 ...
$ tea             : int  0 1 0 0 2 2 2 0 0 0 ...
$ coffee          : int  0 1 4 0 1 0 0 2 3 2 ...
$ height          : int  151 162 162 154 154 159 160 163 154
                          160 ...
$ weight          : int  58 60 75 45 50 66 66 66 60 77 ...
$ age             : int  72 68 78 91 65 82 74 73 89 87 ...
$ meat            : int  4 5 3 0 5 4 3 4 4 2 ...
$ fish            : int  3 2 1 4 3 2 3 2 3 3 ...
$ raw_fruit       : int  1 5 5 4 5 5 5 5 5 5 ...
$ cooked_fruit_veg: int  4 5 2 0 5 5 5 5 5 4 ...
$ chocol          : int  5 1 5 3 3 1 5 1 5 0 ...
$ fat             : int  6 4 4 2 2 3 6 6 6 3 ...
```

We shall now set up an modified `R` structure for each variable that better reflect their type and meaning.

[2] If `gender` had been entered as M and F, the variable would have automatically been structured as a factor. In the same way, the entries for the other two factor variables, `situation` and `fat`, could have been entered as letters or words.

2.2.1 Structuring Nominal Factors

For qualitative variables without order (that is, factor variables), set up the structure with the function `as.factor`. It might be useful to also use the function `levels` to recode the different levels of a qualitative variable. Let us perform these operations on the factor variables from our dataset:

```
> nutri$gender = as.factor(nutri$gender)
> levels(nutri$gender) = c("Male","Female")
> nutri$situation = as.factor(nutri$situation)
> levels(nutri$situation) = c("single","couple",
+                                       "family","other")
> nutri$fat = as.factor(nutri$fat)
> levels(nutri$fat) = c("butter","margarine","peanut",
+              "sunflower","olive","Isio4","rapeseed","duck")
```

2.2.2 Structuring Ordinal Factors

For ordinal factors, the structure can be set up with the function `as.ordered`. As for nominal factors, it is possible to recode the different levels via the function `levels`. Let us perform these operations on the ordinal factors from our dataset:

```
> nutri$meat = as.ordered(nutri$meat)
> nutri$fish = as.ordered(nutri$fish)
> nutri$raw_fruit = as.ordered(nutri$raw_fruit)
> nutri$cooked_fruit_veg = as.ordered(nutri$cooked_fruit_veg)
> nutri$chocol = as.ordered(nutri$chocol)
> mylevels = c("never","< 1/week.","1/week.","2-3/week.",
+               "4-6/week.","1/day")
> levels(nutri$chocol) = levels(nutri$cooked_fruit_veg) =
+                   levels(nutri$raw_fruit)  = mylevels
> levels(nutri$fish) = levels(nutri$meat) = mylevels
```

2.2.3 Structuring Discrete Quantitative Data

For a quantitative variable that takes integer values, the structure is set up with the function `as.integer`.

```
> nutri$tea = as.integer(nutri$tea)
> nutri$coffee = as.integer(nutri$coffee)
```

Note that `nutri$tea` and `nutri$coffee` were initially classified as integer types anyway, so that the above assignments are superfluous.

2.2.4 Structuring Continuous Quantitative Variables

For a continuous variable, the structure is set up with the function `as.double`.

```
> nutri$height = as.double(nutri$height)
> nutri$weight = as.double(nutri$weight)
> nutri$age = as.double(nutri$age)
```

2.2.5 Good Practice

We can now check using the R function `str` the structure of our `data.frame` `nutri`.

```
> str(nutri)
 'data.frame':        226 obs. of  13 variables:
$ gender          : Factor w/ 2 levels "Male","Female": 2 2 2
                     2 2 2 2 2 2 2 ...
$ situation       : Factor w/ 4 levels "single","couple",..: 1
                     1 1 1 1 1 1 1 1 1 ...
$ tea             : int  0 1 0 0 2 2 2 0 0 0 ...
$ coffee          : int  0 1 4 0 1 0 0 2 3 2 ...
$ height          : num  151 162 162 154 154 159 160 163
                          154 160 ...
$ weight          : num  58 60 75 45 50 66 66 66 60 77 ...
$ age             : num  72 68 78 91 65 82 74 73 89 87 ...
$ meat            : Ord.factor w/ 6 levels "never"<"< 1/week."
                     <..: 5 6 4 1 6 5 4 5 5 3 ...
$ fish            : Ord.factor w/ 6 levels "never"<"< 1/week."
                     <..: 4 3 2 5 4 3 4 3 4 4 ...
$ raw_fruit       : Ord.factor w/ 6 levels "never"<"< 1/week."
                     <..: 2 6 6 5 6 6 6 6 6 6 ...
$ cooked_fruit_veg: Ord.factor w/ 6 levels "never"<"< 1/week."
                     <..: 5 6 3 1 6 6 6 6 6 5 ...
$ chocol          : Ord.factor w/ 6 levels "never"<"< 1/week."
                     <..: 6 2 6 4 4 2 6 2 6 1 ...
$ fat             : Factor w/ 8 levels "butter","margarine",..
                     : 6 4 4 2 2 3 6 6 6 3 ...
```

We can access the variables (columns) of a data frame via the $ construction.

```
> nutri$gender[1:3]  #first three elements of gender


[1] Female Female Female
Levels: Male Female


> class(nutri$gender)
```

```
 [1] "factor"
```

You can save[3] your data in another CSV file via the `write.csv` function, as in:

```
> write.csv(nutri,"nutri_restructured.csv")
```

In the remaining sections of this chapter we discuss various ways to extract summary information from a data frame. Which type of plots and numerical summaries can be performed depends strongly on the structure of the data frame, and on the type of the variable(s) in play.

2.3 Summary Tables

It is often interesting to represent a large table of data in a more condensed form. A table of counts or a table of frequencies makes it easier to understand the underlying distribution of a variable, especially if the data are qualitative or ordinal. Such tables are obtained with the function `table`.

As a first example, we first read in the restructured CSV file of the nutritional data, obtained in the previous section. We then run `table` on the variable (column) `fat`.

```
> rm(list=ls())   # Always a good idea to clear the workspace
> nutri = read.csv("nutri_restructured.csv")
> (fat.table = table(nutri$fat))


fat
   butter  margarine   peanut  sunflower   olive   Isio4
       15         27       48         68      40      23
   rapeseed       duck
          1          4
```

The outer brackets in

```
> (fat.table = table(nutri$fat))
```

cause the value that is stored in the object `fat.table` to be printed.

It is also possible to use `table` to **cross tabulate** between two or more variables:

```
> (cross.table = table(nutri$gender, nutri$situation))
        situation
gender   single couple family other
  Male       20     63      2     0
  Female     78     56      7     0
```

[3] Note that this does not save your level ordering!

To add summed margins to this table, use the function addmargins.

```
> (table.complete = addmargins(cross.table))
         situation
gender    single couple family other sum
  Male        20     63      2     0  85
  Female      78     56      7     0 141
  sum         98    119      9     0 226
```

2.4 Summary Statistics

In the following, $\boldsymbol{x} = (x_1, \ldots, x_n)^\mathsf{T}$ is a column vector of numbers. For our nutri data set $\boldsymbol{x}$ could for example correspond to the heights of the $n = 226$ individuals.

Numerical summaries cannot be computed when some data are missing (NA). If necessary, missing data can be omitted with the function omit.

```
> x = na.omit(nutri$height) # Useless in this case
                            #  since height has no NA.
```

The **mean** of the data of $x_1, \ldots, x_n$ is denoted by $\bar{x}$ and is simply the average of the data values:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \, .$$

We will often refer to $\bar{x}$ as the **sample mean**, rather than "the mean of the data". Using the mean function in R for our nutri data, we have, for example:

```
> mean(nutri$height)
[1] 163.9602
```

The **median** of the data $x_1, \ldots, x_n$ is the value $\widetilde{x}$ "in the middle" of the data. More precisely, if we first *order* the data so that $x_1 \leqslant x_2 \leqslant \cdots \leqslant x_n$, then

- if n is odd, then the median is the value $x_{\frac{n+1}{2}}$ — that is, the value at position $\frac{n+1}{2}$,
- if n is even, then any value between the values at positions $\frac{n}{2}$ and $\frac{n}{2} + 1$ can be used as a median of the series. In practice, the median is usually the average between these two values.

The R function to calculate the median is median. For example,

```
> median(nutri$height)
[1] 163
```

The p-**quantile** ($0 < p < 1$) of the data $x_1, \ldots, x_n$ is a value y such that a fraction p of the data is less than or equal to y and a fraction $1 - p$ of the data is greater than or equal to y. For example, the sample 0.5-quantile corresponds to the sample median. The p-quantile is also called the $100 \times p$ **percentile**. The 25, 50, and 75 sample percentiles are sometimes called the first, second, and third **quartiles**. Using R we have, for example,

```
> quantile(nutri$height,probs=c(0.1,0.9))
10% 90%
153 176
```

While the sample mean and median say something about the *location* of the data, it does not provide information about the *dispersion* (spread) of the data. The following summary statistics are useful for this purpose.

The **range** of the data $x_1, \ldots, x_n$ is given by

$$\text{range} = \max_{1 \leqslant i \leqslant n} x_i - \min_{1 \leqslant i \leqslant n} x_i \ .$$

In R, the function `range` returns the minimum and maximum of the data, so to get the actual range we have to take the difference of the two.

```
> range(nutri$height)
140 188
```

Typically, when the sample size increases, the range becomes wider, and so it is difficult to compare the spreads of two data sets via their ranges, when when the sample sizes are different. A more robust measure for the spread of the data is the **interquartile range** (IQR), which is the difference between the third and first quartile.

```
> IQR(nutri$height)

[1] 13
```

The **sample variance** of $x_1, \ldots, x_n$ is defined as

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \ , \tag{2.1}$$

where $\bar{x}$ is the sample mean. We will see in later chapters that it plays an essential role in the analysis of statistical data. The square root of the sample variance $s = \sqrt{s^2}$ is called the **sample standard deviation**. In R, we have, as an example,

```
> var(nutri$height)
81.06063
> sd(nutri$height)
9.003368
```

The function **summary** applied to a vector of quantitative data calculates the minimum, maximum, mean and the three quartiles.

2.5 Making Plots

In this section we describe various methods for visualising data. The main point we would like to make is that the way in which variables are plotted should always be adapted to the variable types; for example, qualitative data should be plotted differently from quantitative data. Such a distinction is an integral part of R's philosophy.

Check out the summary of traditional R graphics at
`http://users.monash.edu.au/~murray/AIMS-R-users/ws/ws11.html`

2.5.1 Plotting Qualitative Variables

Barplot

Suppose we wish to display graphically how many elderly are living by themselves, as a couple, with family, or other. Recall that, the data are given in the `situation` column of our **nutri** data frame. Assuming that we already *restructured the data*, as ☞ 23 in Section 2.2, we can make a **barplot** of the number of people in each category.

```
> barplot(table(nutri$situation))
```

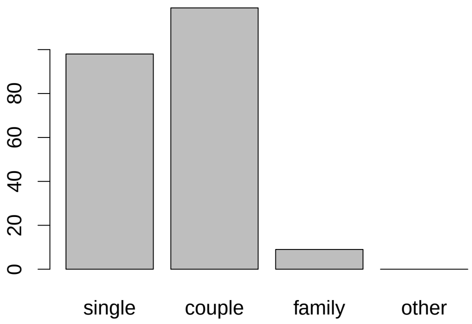


Figure 2.1: Barplot for a qualitative variable.

The function **barplot** is part of the base (i.e., default) plotting library. In addition to the base graphics package, R has many other packages for plotting. We will be using the **lattice** package frequently.

In RStudio you can save a plot as an image or PDF file via the graphical interface.

2.5.2 Plotting Quantitative Variables

We now present a few useful graphs for exploring quantitative data, again using the `nutri` data frame. We will first focus on continuous variables (e.g., age) and then add some specific graphs related to discrete variables (e.g., tea). The aim is to describe the variability present in a single variable. This typically involves a central tendency, where observations tend to gather around, with fewer observations further away. The main aspects of the distribution are the *location* (or centre) of the variability, the *spread* of the variability (how far the values extend from the centre) and the *shape* of the variability; e.g., whether or not values are spread symmetrically on either side of the centre.

It will be convenient to use the graphics package `lattice` instead of the standard `R` plotting functions, as the output will be more pleasing to the eye, and the formulation will be easier. To use a package such as `lattice`, remember to first install the package, and then to load it via `library(lattice)`. Any `lattice` function allows a syntax of the following form.

```
>  lattice.function(formula, data = my.data)
```

Here `formula` is a variable such as `situation` or an expression such as `situation ~ gender`. Formulas appear in many `R` functions, and we will encounter various examples of formulas later on in this book. Another important feature of `lattice` functions is the explicit specification of the data set via the "`data =`" construction. This avoids the use of the $ device in accessing the variables of a data frame.

The meaning of a formula is function- and package-dependent. In `lattice` some formulas are of the form ~ `variable`.

Boxplot

A **boxplot** (or, more generally, a box-and-whiskers plot) can be viewed as a graphical representation of a five-number summary of the data consisting of the minimum, maximum, and the first, second, and third quartiles. Figure 2.2 gives a boxplot of the `age` variable of the `nutri` data. It was made with the `bwplot` function from the `lattice` package.

```
> library(lattice)
> bwplot(~age, data=nutri)
```

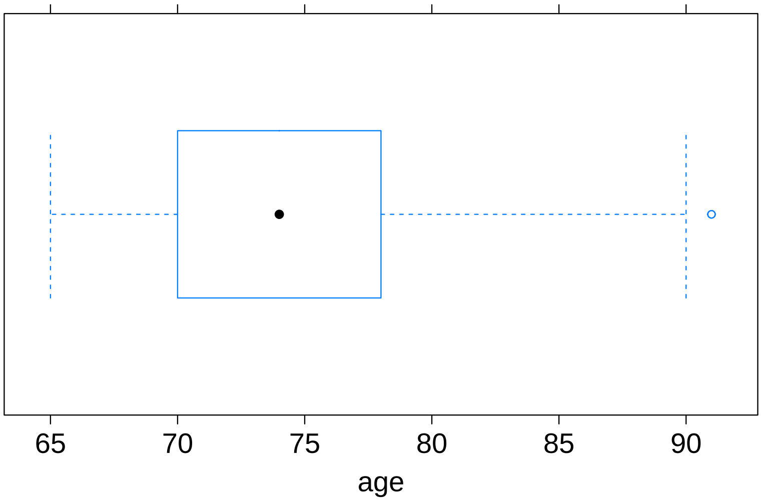


Figure 2.2: boxplot for `age` made with `bwplot`.

The box is drawn from the first quantile (Q_1) to the third quantile (Q_3). The solid dot inside the box signifies the location of the second quantile, i.e., the median. So-called “whiskers” extend to either side of the box. The size of the box is called the **interquartile range**: IQR = $Q_3 - Q_1$. The left whisker is the largest of (a) the minimum of the data and (b) $Q_1 - 1.5\,\text{IQR}$. Similarly, the right whisker is the smallest of (a) the maximum of the data and (b) $Q_3 + 1.5\,\text{IQR}$. Any data point outside the whiskers is indicated by a small open dot, indicating a suspicious or deviant point (outlier). Note a boxplot may also be used for discrete quantitative variables.

Histogram

A **histogram** is a main graphical representation of the distribution of a quantitative variable. We start by breaking the range of the values into a number of *bins* or *classes*. We tally the counts of the values falling in each bin and then make the plot by drawing rectangles whose bases are the bin intervals and whose heights are the counts. In `R` we can use the standard graphics function **`hist`** or, from the package **`lattice`**, we can use **`histogram`**. For example, Figure 2.3 shows a histogram of the 226 ages in data `nutri`.

```
> histogram(~age, data=nutri)
```

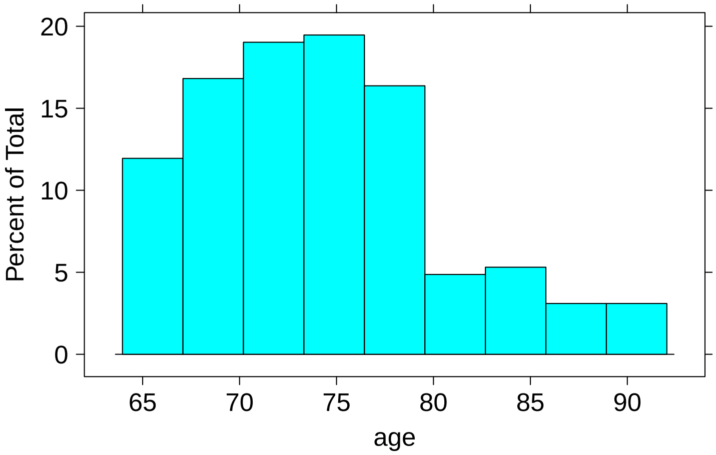


Figure 2.3: Histogram of variable age.

Here 9 bins were used. Rather than using raw counts, the vertical axis here gives the percentage in each class, defined by $\frac{count}{total}100\%$. Histograms can also be used for discrete variables, although it may be necessary to explicitly specify the bins and placement of the ticks. The number of bins can be changed via the parameter `nint`.

Density Plot

Instead of a histogram, we could use a **density plot** to visualize the distribution of a continuous variable. For example, using the function `densityplot` from `lattice`:

```
> densityplot(~weight,lwd=2,data=nutri)
```

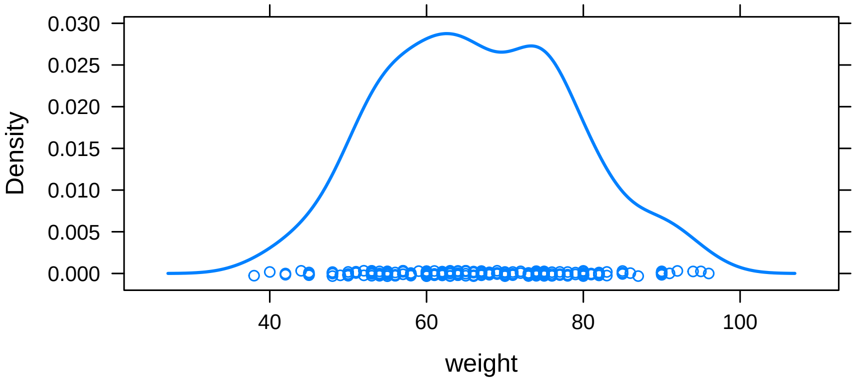


Figure 2.4: Density plot of weight.

This plot indicates that perhaps there is a bimodal distribution of the weights, caused by the two different genders. This corroborated by a density plot of the weights by gender.

```
> densityplot(~weight,groups=gender,lwd=2,data=nutri)
```

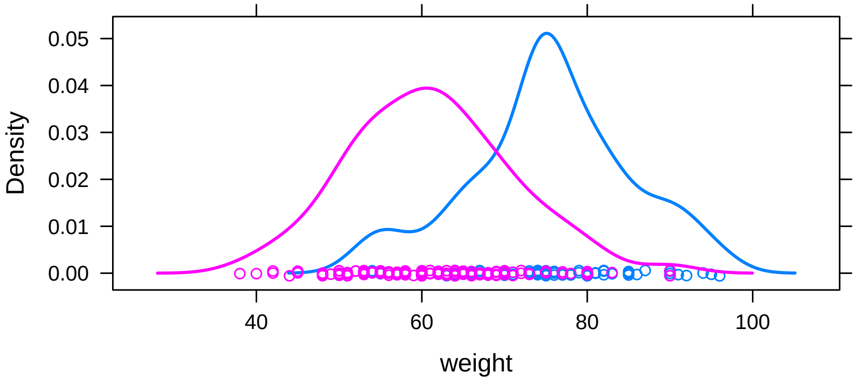


Figure 2.5: Density plot of weight by gender.

The smoothness of density plots made with `densityplot` can be tuned with the “bandwidth” parameter `bw`.

Empirical Cumulative Distribution Function

The **empirical cumulative distribution function**, denoted by $F_n(\cdot)$, is a step function which jumps an amount k/n at observation values, where k is the number of tied observations at that value. For observations $(x_1, \ldots, x_n)$, $F_n(x)$ is the fraction of observations less than or equal to x, i.e.,

$$F_n(x) = \frac{\#\{x_i \leqslant x\}}{n} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{I}_{\{x_i \leqslant x\}} ,$$

where $\mathrm{I}_{\{x_i \leqslant x\}}$ is equal to 1 when $x_i \leqslant x$ and 0 otherwise. To produce the plot of the empirical cumulative distribution function using R, we can combine the functions `plot` and `ecdf`. The result for the age data is shown in Figure 2.6. The empirical distribution function for a discrete quantitative variable is obtained in the same way.

```
> plot(ecdf(nutri$age),xlab="age")
```

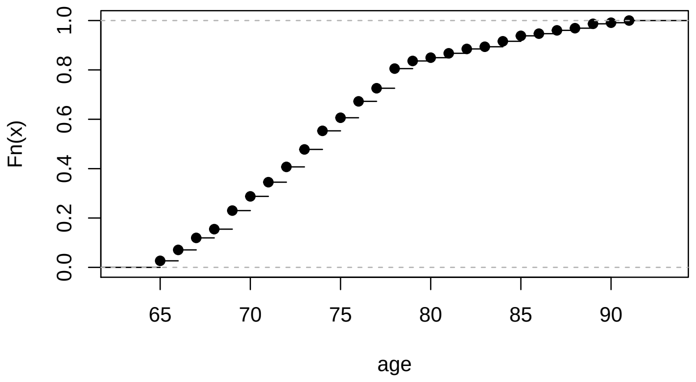


Figure 2.6: Plot of the empirical cdf for a continuous quantitative variable.

The “inverse” of the empirical cdf is obtained by swapping the x and y coordinates in the plot above. This gives a plot of the p-quantile of the data against $p \in [0, 1]$. We can view the variable p as the theoretical p-quantile of the uniform distribution on [0,1]; see also Section 4.4. Plots that compare quantiles against quantiles, whether theoretical or sample quantiles, are called **quantile-quantile plots** or **qq-plots** for short. In the following code we use the function `qqmath` from the `lattice` package. ☞ 69

```
> qqmath(~age,data=nutri, type = c("l","p"),distribution=qunif)
```

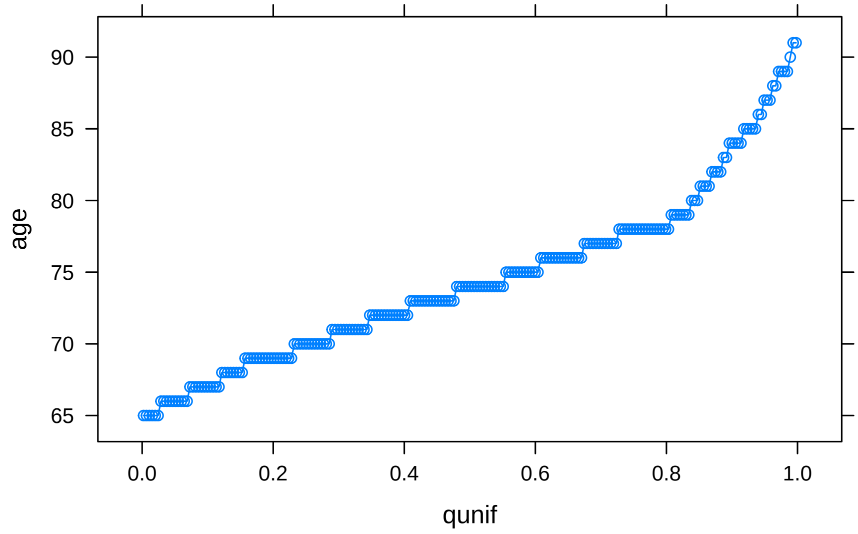


Figure 2.7: Plot of age against quantile.

In Chapters 8–10 we will be using qq-plots to assess if data could be coming from a prescribed distribution, such as the normal distribution.

2.5.3 Graphical Representations in a Bivariate Setting

In this section, we present a few useful representations to explore relationships between two variables. The graphical representation will depend on the nature of the two variables.

Two-way Plots for Two Qualitative Variables

Comparing barplots for two qualitative variables is very easy using R's basic **barplot** function. The following plots the contingency table `cross.table` of Section 2.3, which cross-tabulates the family status (situation) with the gender of the elderly people.

```
> barplot(cross.table,bes=T,leg=T)  #beside each other, with legend
```

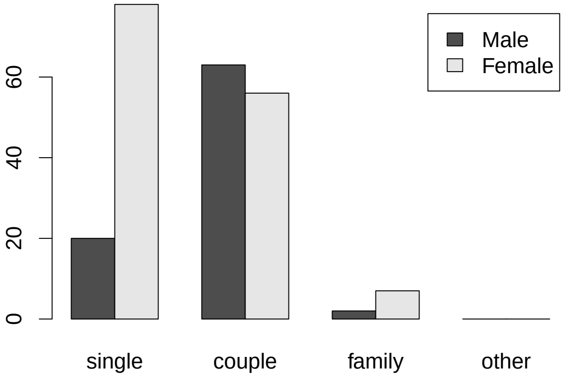


Figure 2.8: Barplot for two qualitative variables.

Two-way plots for two quantitative variables

We can visualize patterns between two quantitative variables using a **scatter plot**. These are easily drawn by making two axes, one for each variable, and then using the values of the variables as the coordinates of a point to plot each case. Recall that the basic **plot** function in R is a generic function whose output depends on the types (discrete qualitative, continuous quantitive, etc.) of the input. The following two calls lead to exactly the same scatterplot of weight against height in Figure 2.9, even though the inputs are different.

```
> plot(nutri$height, nutri$weight) #two arguments (variables)
> plot(nutri$weight~nutri$height)  #one argument (formula)
```

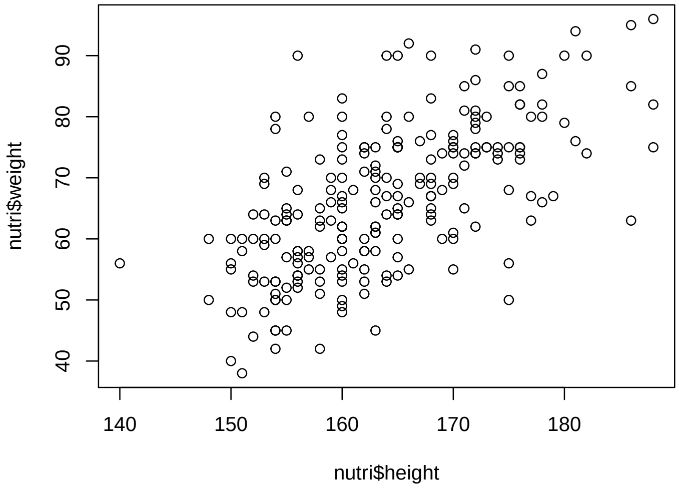


Figure 2.9: Plot of two quantitative variables.

In the second command we see the formula weight ~ height being used in a "base graphics" setting. It is often desired to add a smooth "trend" curve or a straight line through the scatterplot. This is easy to accomplish via the function **xyplot** from the **lattice** package, by specifying type parameter(s) to be plotted. Choices are "p" (points), "smooth" (smooth curve), or "r" (straight line).

```
> xyplot(weight~height,type=c("p","smooth"),
       col.line="darkorange",lwd=3, data=nutri)
```

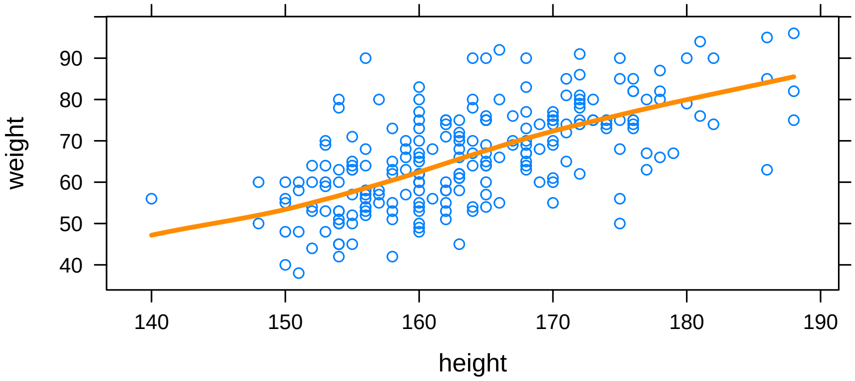


Figure 2.10: Scatterplot of weight against height, with a smoothed *loess* curve.

The following code illustrates that it is possible to produce highly sophisticated scatter plots, such as in Figure 2.11. The figure shows the birth weights (mass) of babies whose mothers smoked (blue triangles) or not (red circles). In addition, straight lines were fitted to the two groups, suggesting that birth weight decreases with age when the mother smokes, but increases when the mother does not smoke! The question is justified whether these trends are significant or due to chance. We will revisit this data set later on in the book; see Section 10.3.

☞ 165

```
library(MASS)       # load the package MASS
ls("package:MASS") # show all variables associated with this package
help(birthwt)       # find information on the data set birthwt

xyplot(bwt~age, groups = factor(smoke),type = c("p","r"),data=birthwt,
  lwd =2, cex=1,lty=c(1,2), col=c("red","blue"),      #set width,type, color
  scales=list(x=list(cex=1.1), y=list(cex=1.1)),  #change font size on axes
  xlab = list(label="age",cex=1.1),      #change label and label size x-axis
  ylab = list(label="birth weight",cex=1.1),
  par.settings = list(superpose.symbol = list(pch=c(1,2))), #new characters
  key= list(corner = c(0,1),                               #specify the legend
            text= list(c("non-smoking","smoking")),
            cex=1.1,
            lines = list(lty = c(1,2),col=c("red","blue")),
            border = T))
```

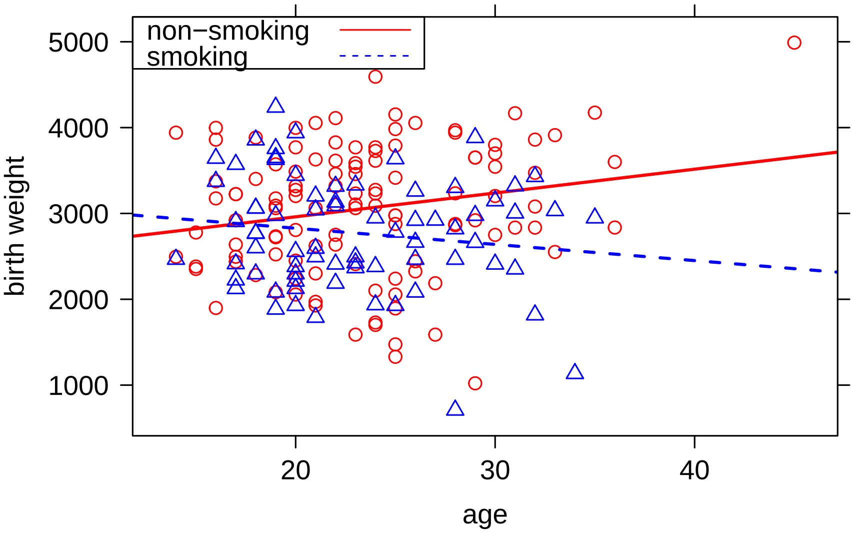


Figure 2.11: Birth weight against age for smoking and non-smoking mothers.

Two-way Plots for one Qualitative and one Quantitative Variable

In this setting, it is interesting to draw boxplots of the quantitative variable for each level of the qualitative variable. Assuming the variables are structured correctly, the function bwplot in the lattice package can be used to produce Figure 2.12 using the following command.

```
> bwplot(coffee~gender, data=nutri, fill="powderblue",
         notch=T, pch="|")
```

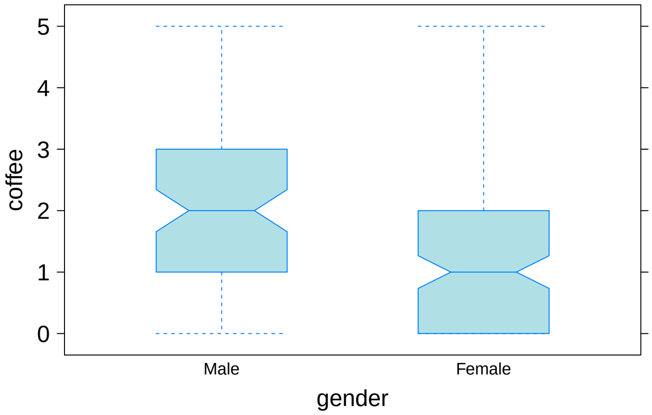


Figure 2.12: Boxplots of a quantitative variable (coffee) as a function of the levels of a qualitative variable (gender).

2.5.4 Visualising More than Two Variables

Visualising data involving more than two variables requires careful design, which is often more of an art than a science. Nevertheless, there are packages such as ggplot2 that are aimed at automating the graphical display of complicated data sets as much as possible. Lattice graphics, such as implemented in the lattice package also make life easier in this respect. As an example, consider the visualization in Figure 2.13 of tree data for R's Orange data set. This is a data frame containing the growth curves of 5 orange trees. For each tree, the age (in days) and circumference (in mm) were recorded at multiple points in time. The variable names are Tree, age, and circumference. Note that the variable Tree is an ordered factor, and that the plots are ordered (from bottom to top) in increasing order of the maximum circumference. The order is specified by the levels of this variable:

```
> levels(Orange$Tree)
```

```
[1] "3" "1" "5" "2" "4"
```

The plot was made with `xyplot` simply as follows.

```
> xyplot(circumference~age|Tree, data=Orange, type=c("p","smooth"))
```

The formula `circumference~age|Tree` specifies that the `circumference` is plotted against `age`, for each `Tree`. The `type` indicates how each graph is displayed. In this case both the individual points and a smooth fitted curve are plotted.

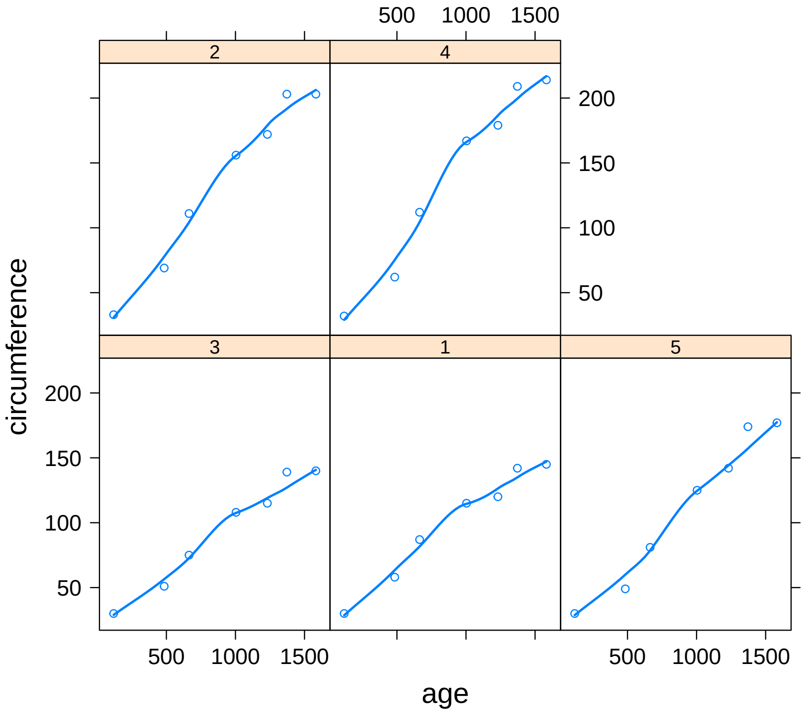


Figure 2.13: Growth curves for five orange trees from the R data set `Orange`.

Using the `lattice` package, we can go even further and display a plot with 4 variables, using the `groups` argument. Figure 2.14 show a plot for the electroconductivity of soil, for three different water contents (0%, 5%, and 15%), three levels of salinity, and three different soils types (clay, loam, and sand). It was made with the following code:

```
conduct = read.csv("Conductivity.csv", header=T)
# make Salinity an ordered factor
conduct$Salinity = as.ordered(conduct$Salinity)
# change the order of the levels
levels(Salinity) = c("Low", "Medium", "High")
library(lattice)
xyplot(Electroconductivity ~ Salinity | Water, groups = Type,
          auto.key=list(corner = c(0,0.9)), data = soildata)
```

The `auto.key` argument provides a list of further options for changing the figure. In this case we move the legend to a more suitable place.

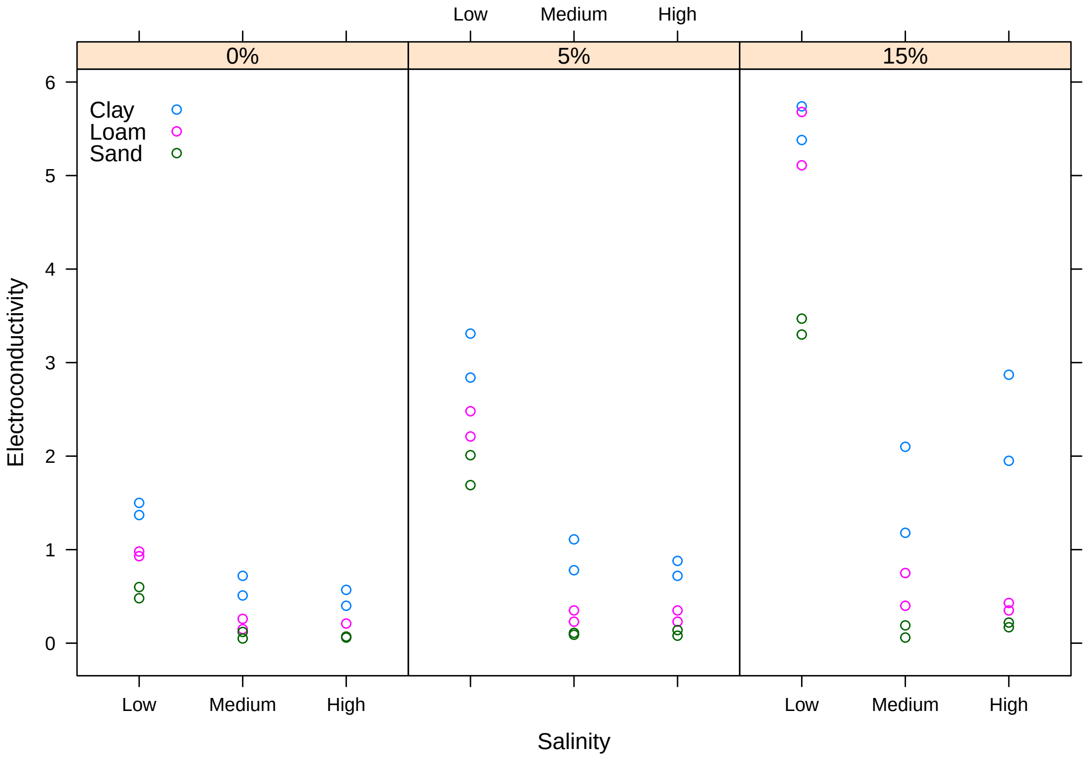


Figure 2.14: Electroconductivity as a function of salinity, water content (%) and soil type.

CHAPTER 3

UNDERSTANDING RANDOMNESS

The purpose of this chapter is to introduce you to the language of *probability*, which is an indispensable tool for the understanding of randomness. You will learn how to think about random experiments in terms of probability models and how to calculate probabilities via counting. We will discuss how to describe random measurements via random variables and their distributions — specified by the cdf, pmf, and pdf. The expectation and variance of random variables provide important summary information about the distribution of a random variable.

3.1 Introduction

Statistical data is inherently random: if we would repeat the process of collecting the data, we would most likely obtain different measurements. Various reasons why there is variability in the data were already discussed in Section 1.2. ☞ 13

To better understand the role that randomness plays in statistical analyses, we need to know a few things about the theory of *probability* first.

3.2 Random Experiments

The basic notion in probability is that of a **random experiment**: an experiment whose outcome cannot be determined in advance, but which is nevertheless subject to analysis. Examples of random experiments are:

1. tossing a die and observing its face value,

2. measuring the amount of monthly rainfall in a certain location,

3. choosing at random ten people and measuring their heights,

4. selecting at random fifty people and observing the number of left-handers,

5. conducting a survey on the nutrition of the elderly, resulting in a data frame such ☞ 21 as **nutri** discussed in Chapter 2.

The goal of *probability* is to understand the behaviour of random experiments by analysing the corresponding *mathematical models*. Given a mathematical model for a random experiment, one can calculate quantities of interest such as probabilities and expectations (defined later). Mathematical models for random experiments are also the basis of *statistics*, where the objective is to infer which of several competing models best fits the observed data. This often involves the estimation of model parameters from the data.

■ **Example 3.1 (Coin Tossing)** One of the most fundamental random experiments is the one where a coin is tossed a number of times. Indeed, much of probability theory ☞ 10 can be based on this simple experiment. In Section 1.1 we viewed this experiment from a statistical point of view (is the coin fair?). As we have already seen, to better understand how this coin toss experiment behaves, we can carry it out on a computer. The following R program simulates a sequence of 100 tosses with a fair coin (that is, Heads and Tails are equally likely), and plots the results in a bar chart.

```
> x = runif(100)<0.5    # simulate the coin tosses
> barplot(x)            # plot the results in a bar chart
```

This is what the first line of code does: the function **runif** is used to draw a vector of 100 uniform random numbers from the interval [0, 1]. By testing whether the uniform numbers are less than 0.5, we obtain a vector x of logical (TRUE or FALSE) variables, indicating Heads and Tails, say. Typical outcomes for three such experiments were ☞ 10 given in Figure 1.2.

We can also plot the average number of Heads against the number of tosses. This is accomplished by adding two lines of code:

```
> y = cumsum(x)/1:100 # calculate the cumulative  sum and divide
                      # elementwise by the vector 1:100
> plot(y,type="l")    # plot the result in a line graph
```

The result of three such experiments is depicted in Figure 3.1. Notice that the average number of Heads seems to converge to 0.5, but there is a lot of random fluctuation.

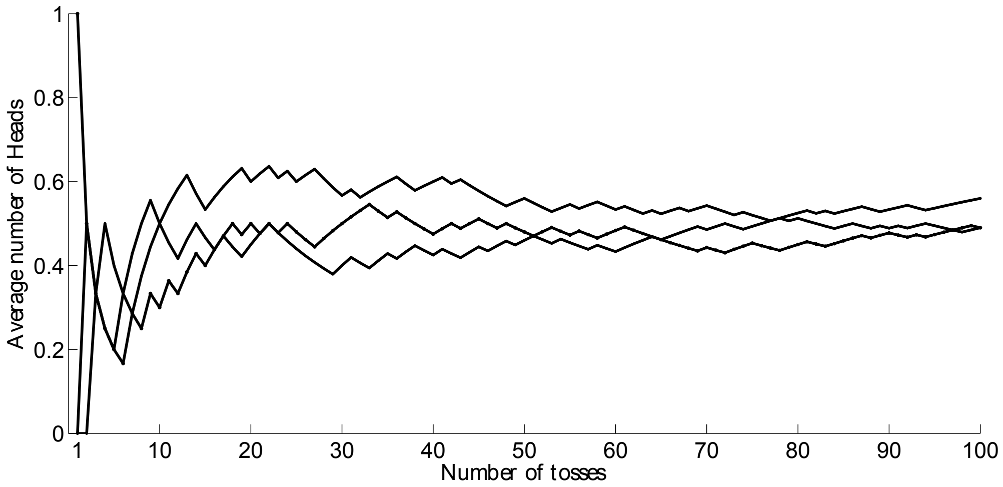


Figure 3.1: The average number of Heads in n tosses, where $n = 1, \ldots, 100$.

Similar results can be obtained for the case where the coin is *biased*, with a probability of Heads of p, say. Here are some typical *probability* questions.

- What is the probability of x Heads in 100 tosses?
- How many Heads would you expect to come up?
- What is the probability of waiting more than 4 tosses before the first Head comes up?

A statistical analysis would start from observed data of the experiment — for example, all the outcomes of 100 tosses are known. Suppose the probability of Heads p is not known. Typical *statistics* questions are:

- Is the coin fair?
- How can p be best estimated from the data?
- How accurate/reliable would such an estimate be?

■

To answer these type of questions, we need to have a closer look at the models that are used to describe random experiments.

3.3 Probability Models

Although we cannot predict the outcome of a random experiment with certainty, we usually can specify a set of possible outcomes. This gives the first ingredient in our model for a random experiment.

Definition 3.1: Sample Space

The **sample space** Ω of a random experiment is the set (collection) of all possible outcomes of the experiment.

Examples of random experiments with their sample spaces are:

1. Cast two dice consecutively and observe their face values. A typical outcome could be written as a tuple (first die, second die). It follows that Ω is the set containing the outcomes $(1, 1), (1, 2), \ldots, (1, 6), (2, 1), \ldots, (6, 6)$. There are thus $6 \times 6 = 36$ possible outcomes.

2. Measure the lifespan of a person in years. A possible outcome is for example 87.231 or 39.795. Any real number between 0 and, say, 140 would be possible. So, we could take Ω equal to the interval $[0, 140]$.

3. Measure the heights in metres of 10 people. We could write an outcome as a vector $(x_1, \ldots, x_{10})$, where the height of the first selected person is x_1, the height of the second person is x_2, and so on. We could take Ω to be the set of all positive vectors of length 10.

For modeling purposes it is often easier to take the sample space larger (but not smaller) than is strictly necessary. For example, in the second example we could have taken the set of real numbers as our sample space.

Often we are not interested in a single outcome but in whether or not one of a *group* of outcomes occurs.

Definition 3.2: Event

An **event** is a subset of the sample space Ω to which a probability can be assigned.

Events will be denoted by capital letters $A, B, C, \ldots$. We say that event A *occurs* if the outcome of the experiment is one of the elements in A.

Examples of events for the three random experiments mentioned above are:

1. The event that the sum of two dice is 10 or more:

$$A = \{(4, 6), (5, 5), (5, 6), (6, 4), (6, 5), (6, 6)\} .$$

2. The event that a person lives to become an octogenarian:

$$A = [80, 140)\,.$$

3. The event that the third selected person in the group of 10 is taller than 2 metres:

$$A = \{(x_1, \dots, x_{10}) \text{ such that } x_3 > 2\}\,.$$

> Note that a list of numbers can be *ordered* or *unordered*. It is customary to write unordered lists (that is, sets) with curly brackets, and ordered lists (that is vectors) with round brackets. Hence, $\{1, 2, 3\}$ is the same as $\{3, 2, 1\}$, but the vector $(1, 2, 3)$ is not equal to $(3, 2, 1)$.

Since events are sets, we can apply the usual set operations to them, as illustrated in the *Venn diagrams* in Figure 3.2.

1. The set $A \cap B$ (A **intersection** B) is the event that A *and* B both occur.

2. The set $A \cup B$ (A **union** B) is the event that A *or* B *or* both occur.

3. The event A^c (A **complement**) is the event that A does *not* occur.

4. If $B \subset A$ (B is a **subset** of A) then event B is said to *imply* event A.

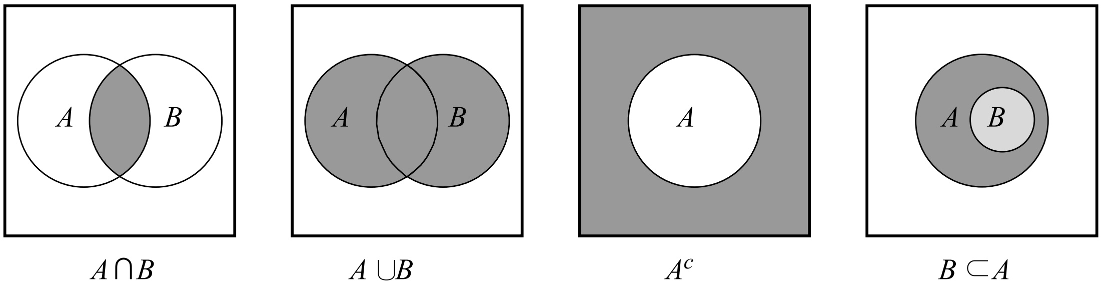


Figure 3.2: Venn diagrams of set operations. Each square shows the sample space Ω.

Two events A and B which have no outcomes in common, that is, $A \cap B = \emptyset$ (empty set), are called **disjoint** events.

■ **Example 3.2 (Casting Two Dice)** Suppose we cast two dice consecutively. The sample space is $\Omega = \{(1, 1), (1, 2), \dots, (1, 6), (2, 1), \dots, (6, 6)\}$. Let $A = \{(6, 1), \dots, (6, 6)\}$ be the event that the first die is 6, and let $B = \{(1, 6), \dots, (6, 6)\}$ be the event that the second die is 6. Then $A \cap B = \{(6, 1), \dots, (6, 6)\} \cap \{(1, 6), \dots, (6, 6)\} = \{(6, 6)\}$ is the event that both dice are 6. ■

The third ingredient in the model for a random experiment is the specification of the probability of the events. It tells us how *likely* it is that a particular event will occur. We denote the probability of an event A by $\mathbb{P}(A)$ — note the special "black board bold" font. No matter how we define $\mathbb{P}(A)$ for different events A, the probability must always satisfy three conditions, given in the following definition.

Definition 3.3: Probability Measure

A **probability measure** $\mathbb{P}$ is a function which assigns a number between 0 and 1 to each event, and which satisfies the following rules:

1. $0 \leqslant \mathbb{P}(A) \leqslant 1$.
2. $\mathbb{P}(\Omega) = 1$.
3. For any sequence $A_1, A_2, \ldots$ of *disjoint* events we have

Sum Rule: $$\mathbb{P}(A_1 \cup A_2 \cup \cdots) = \mathbb{P}(A_1) + \mathbb{P}(A_2) + \cdots \ . \tag{3.1}$$

The crucial property (3.1) is called the **sum rule** of probability. It simply states that if an event can happen in several distinct ways, then the probability that at least one of these events happens (that is, the probability of the union) is equal to the sum of the probabilities of the individual events. We see a similar property in an *area* measure: the total area of the union of nonoverlapping regions is simply the sum of the areas of the individual regions.

The following theorem lists some important consequences of the definition above. Make sure you understand the meaning of each of them, and try to prove them yourself, using *only* the three rules above.

Theorem 3.1: Properties of a Probability Measure

Let A and B be events and $\mathbb{P}$ a probability. Then,

1. $\mathbb{P}(\emptyset) = 0$,
2. if $A \subset B$, then $\mathbb{P}(A) \leqslant \mathbb{P}(B)$,
3. $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$,
4. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$.

We have now completed our general model for a random experiment. Of course for any *specific* model we must carefully specify the sample space Ω and probability $\mathbb{P}$ that best describe the random experiment.

An important case where $\mathbb{P}$ is easily specified is where the sample space has a *finite* number of outcomes that are all *equally likely*. The probability of an event $A \subset \Omega$ is in this case simply

$$\mathbb{P}(A) = \frac{|A|}{|\Omega|} = \frac{\text{Number of elements in } A}{\text{Number of elements in } \Omega} \,. \tag{3.2}$$

The calculation of such probabilities thus reduces to *counting*.

3.4 Counting

Counting is not always easy. Let us first look at some examples:

1. A multiple choice form has 20 questions; each question has 3 choices. In how many possible ways can the exam be completed?

2. Consider a horse race with 8 horses. How many ways are there to gamble on the placings (1st, 2nd, 3rd).

3. Jessica has a collection of 20 CDs, she wants to take 3 of them to work. How many possibilities does she have?

To be able to comfortably solve a multitude of counting problems requires a lot of experience and *practice*, and even then, some counting problems remain exceedingly hard. Fortunately, many counting problems can be cast into the simple framework of drawing balls from an urn, see Figure 3.3.

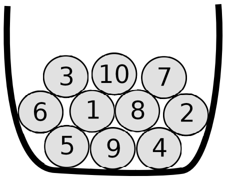


Figure 3.3: An urn with n balls.

Consider an urn with n different balls, numbered $1, \ldots, n$ from which k balls are drawn. This can be done in a number of different ways. First, the balls can be drawn one-by-one, or one could draw all the k balls at the same time. In the first case the *order* in which the balls are drawn can be noted, in the second case that is not possible. In the latter case we can (and will) still assume the balls are drawn one-by-one, but that the order is not noted. Second, once a ball is drawn, it can either be put back into the urn (after the number is recorded), or left out. This is called, respectively, drawing with and without *replacement*. All in all there are 4 possible experiments: (ordered, with replacement), (ordered, without replacement), (unordered, without replacement) and (ordered, with replacement). The art is to recognize a seemingly unrelated counting problem as one of these four urn problems. For the three examples above we have the following

1. Example 1 above can be viewed as drawing 20 balls from an urn containing 3 balls, noting the order, and with replacement.

2. Example 2 is equivalent to drawing 3 balls from an urn containing 8 balls, noting the order, and without replacement.

3. In Example 3 we take 3 balls from an urn containing 20 balls, not noting the order, and without replacement.

We have left out the less important (and more complicated) unordered with replacement case. An example is counting how many different throws there are with 3 dice.

We now consider for each of the three cases how to count the number of arrangements. For simplicity we consider for each case how the counting works for $n = 4$ and $k = 3$, and then state the general situation. Recall the notation that we introduced in Remark 3.3: ordered arrangements are enclosed by round brackets and unordered ones by curly brackets.

☞ 47

Drawing with Replacement, Ordered

Here, after we draw each ball, note the number on the ball, and put the ball back. For our specific case $n = 4$ and $k = 3$ some possible outcomes are: $(1, 1, 1), (4, 1, 2)$, $(2, 3, 2)$, $(4, 2, 1), \ldots$ To count how many such arrangements there are, we can reason as follows: we have three positions $(\cdot, \cdot, \cdot)$ to fill. Each position can have the numbers 1, 2, 3, or 4, so the total number of possibilities is $4 \times 4 \times 4 = 4^3 = 64$. This is illustrated via the tree diagram in Figure 3.4.

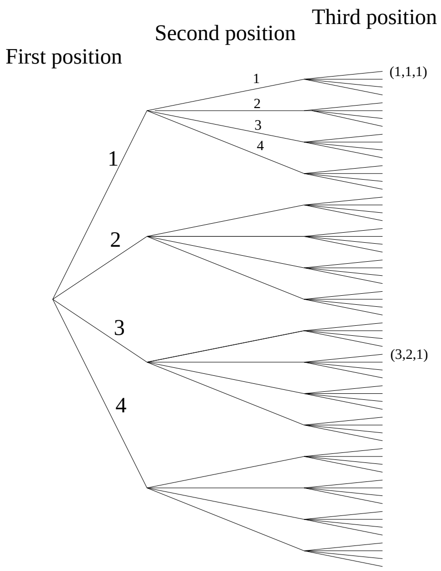


Figure 3.4: Enumerating the number of ways in which three ordered positions can be filled with 4 possible numbers, where repetition is allowed.

For general n and k we can reason analogously to find:

Theorem 3.2: Arrangements with Order and Replacment

The number of ordered arrangements of k numbers chosen from $\{1, \ldots, n\}$, with replacement (repetition) is n^k.

Drawing Without Replacement, Ordered

Here we draw again k numbers (balls) from the set $\{1, 2, \ldots, n\}$, and note the order, but now do not replace them. Let $n = 4$ and $k = 3$. Again there are 3 positions to fill $(\cdot, \cdot, \cdot)$, but now the numbers cannot be the same, e.g., (1,4,2),(3,2,1), etc. Such an ordered arrangements called a **permutation** of size k from set $\{1, \ldots, n\}$. (A permutation of $\{1, \ldots, n\}$ of size n is simply called a permutation of $\{1, \ldots, n\}$ (leaving out "of size n"). For the 1st position we have 4 possibilities. Once the first position has been chosen, we have only 3 possibilities left for the second position. And after the first two positions have been chosen there are 2 possibilities left. So the number of arrangements is $4 \times 3 \times 2 = 24$ as illustrated in Figure 3.5, which is the same tree as in Figure 3.4, but with all "duplicate" branches removed.

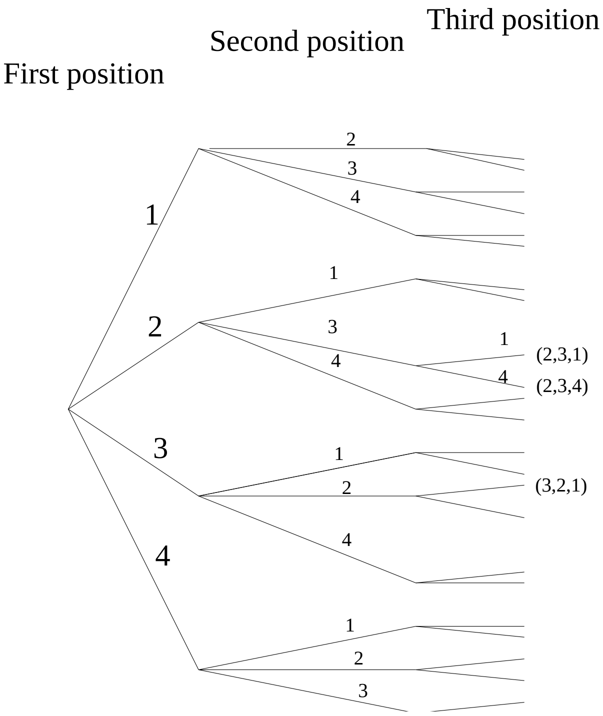


Figure 3.5: Enumerating the number of ways in which three ordered positions can be filled with 4 possible numbers, where repetition is NOT allowed.

For general n and k we have:

Theorem 3.3: Arrangements with Order and without Replacement

The number of permutations of size k from $\{1, \ldots, n\}$ is ${}^nP_k = n(n-1)\cdots(n-k+1)$.

In particular, when $k = n$, we have that the number of ordered arrangements of n items is $n! = n(n-1)(n-2)\cdots 1$, where $n!$ is called ***n*-factorial**. Note that

$$^nP_k = \frac{n!}{(n-k)!}.$$

Drawing Without Replacement, Unordered

This time we draw k numbers from $\{1, \ldots, n\}$ but do not replace them (no replication), and do not note the order (so we could draw them in one grab). Taking again $n = 4$ and $k = 3$, a possible outcome is $\{1, 2, 4\}$, $\{1, 2, 3\}$, etc. If we noted the order, there would be ${}^nP_k$ outcomes, among which would be (1,2,4), (1,4,2), (2,1,4), (2,4,1), (4,1,2), and (4,2,1). Notice that these 6 permutations correspond to the single unordered arrangement $\{1, 2, 4\}$. Such unordered arrangements without replications are called **combinations** of size k from the set $\{1, \ldots, n\}$.

To determine the number of combinations of size k we simply need to divide ${}^nP_k$ by the number of permutations of k items, which is $k!$. Thus, in our example ($n = 4, k = 3$) there are $24/6 = 4$ possible combinations of size 3. In general we have:

Theorem 3.4: Arrangements without Order and without Replacement

The number of combinations of size k from the set $\{1, \ldots n\}$ is

$$^nC_k = \binom{n}{k} = \frac{^nP_k}{k!} = \frac{n!}{(n-k)!\,k!}\ .$$

Note the two different notations for this number. Summarising, we have the following table:

Table 3.1: Number of ways k balls can be drawn from an urn containing n balls.

	Replacement	
Order	Yes	No
Yes	n^k	${}^nP_k$
No	—	${}^nC_k$

Returning to our original three problems, we can now solve them easily:

1. The total number of ways the exam can be completed is $3^{20} = 3,486,784,401$.
2. The number of placings is ${}^{8}P_3 = 336$.
3. The number of possible combinations of CDs is $\binom{20}{3} = 1140$.

Once we know how to count, we can apply the equilikely principle to calculate probabilities:

1. What is the probability that out of a group of 40 people all have different birthdays?

   Answer: Choosing the birthdays is like choosing 40 balls with replacement from an urn containing the balls 1,. . . ,365. Thus, our sample space Ω consists of vectors of length 40, whose components are chosen from $\{1, \ldots, 365\}$. There are $|\Omega| = 365^{40}$ such vectors possible, and all are *equally likely*. Let A be the event that all 40 people have different birthdays. Then, $|A| = {}^{365}P_{40} = 365!/325!$ It follows that $\mathbb{P}(A) = |A|/|\Omega| \approx 0.109$, so not very big!

2. What is the probability that in 10 tosses with a fair coin we get exactly 5 Heads and 5 Tails?

   Answer: Here Ω consists of vectors of length 10 consisting of 1s (Heads) and 0s (Tails), so there are 2^{10} of them, and all are *equally likely*. Let A be the event of exactly 5 heads. We must count how many binary vectors there are with exactly 5 1s. This is equivalent to determining in how many ways the positions of the 5 1s can be chosen out of 10 positions, that is, $\binom{10}{5}$. Consequently, $\mathbb{P}(A) = \binom{10}{5}/2^{10} = 252/1024 \approx 0.25$.

3. We draw at random 13 cards from a full deck of cards. What is the probability that we draw 4 Hearts and 3 Diamonds?

   Answer: Give the cards a number from 1 to 52. Suppose 1–13 is Hearts, 14–26 is Diamonds, etc. Ω consists of unordered sets of size 13, without repetition, e.g., $\{1, 2, \ldots, 13\}$. There are $|\Omega| = \binom{52}{13}$ of these sets, and they are all equally likely. Let A be the event of 4 Hearts and 3 Diamonds. To form A we have to choose 4 Hearts out of 13, and 3 Diamonds out of 13, followed by 6 cards out of 26 Spade and Clubs. Thus, $|A| = \binom{13}{4} \times \binom{13}{3} \times \binom{26}{6}$. So that $\mathbb{P}(A) = |A|/|\Omega| \approx 0.074$.

3.5 Conditional Probabilities

How do probabilities change when we know that some event B has occurred? Thus, we know that the outcome lies in B. Then A will occur if and only if $A \cap B$ occurs, and the relative chance of A occurring is therefore $\mathbb{P}(A \cap B)/\mathbb{P}(B)$, which is called the *conditional probability* of A given B. The situation is illustrated in Figure 3.6.

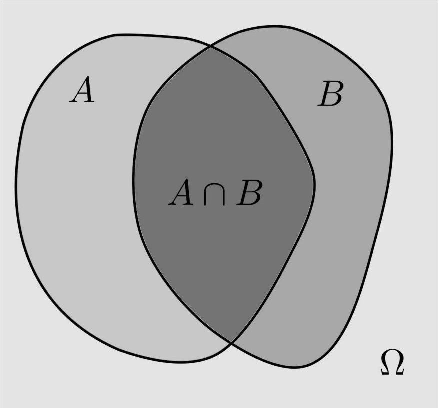


Figure 3.6: What is the probability that A occurs (that is, the outcome lies in A) given that the outcome is known to lie in B?

Definition 3.4: Conditional Probability

The **conditional probability** of A given B (with $\mathbb{P}(B) \neq 0$) is defined as:

$$\mathbb{P}(A \,|\, B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} \,. \tag{3.3}$$

■ **Example 3.3 (Casting Two Dice)** We cast two fair dice consecutively. Given that the sum of the dice is 10, what is the probability that one 6 is cast? Let B be the event that the sum is 10:

$$B = \{(4,6),(5,5),(6,4)\} \,.$$

Let A be the event that one 6 is cast:

$$A = \{(1,6),\ldots,(5,6),(6,1),\ldots,(6,5)\} \,.$$

Then, $A \cap B = \{(4,6),(6,4)\}$. And, since for this experiment all elementary events are equally likely, we have

$$\mathbb{P}(A \,|\, B) = \frac{2/36}{3/36} = \frac{2}{3} \,.$$

■

Independent Events

When the occurrence of B does not give extra information about A, that is $\mathbb{P}(A \,|\, B) = \mathbb{P}(A)$, the events A and B are said to be *independent*. A slightly more general definition (which includes the case $\mathbb{P}(B) = 0$) is:

Definition 3.5: Independent Events

Events A and B are said to be **independent** if

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\,\mathbb{P}(B)\ . \tag{3.4}$$

■ **Example 3.4 (Casting Two Dice (Continued))** We cast two fair dice consecutively. Suppose A is the event that the first toss is 6 and B is the event that the second one is a 6, then naturally A and B are independent events, knowing that the first die is a 6 does not give any information about what the result of the second die will be. Let's check this formally. We have $A = \{(6,1),(6,2)\ldots,(6,6)\}$ and $B = \{(1,6),(2,6),\ldots,(6,6)\}$, so that $A \cap B = \{(6,6)\}$, and

$$\mathbb{P}(A \,|\, B) = \frac{1/36}{6/36} = \frac{1}{6} = \mathbb{P}(A)\ .$$

■

Product Rule

By the definition of conditional probability (3.3) we have

$$\mathbb{P}(A \cap B) = \mathbb{P}(A)\,\mathbb{P}(B \,|\, A)\ .$$

It is not difficult to generalize this to n intersections $A_1 \cap A_2 \cap \cdots \cap A_n$, which we abbreviate as $A_1A_2\cdots A_n$. This gives the second major rule in probability: the **product rule**. We leave the proof as an exercise.

Theorem 3.5: Product Rule

Let $A_1,\ldots,A_n$ be a sequence of events with $\mathbb{P}(A_1\cdots A_{n-1}) > 0$. Then,

$$\mathbb{P}(A_1\cdots A_n) = \mathbb{P}(A_1)\,\mathbb{P}(A_2 \,|\, A_1)\,\mathbb{P}(A_3 \,|\, A_1A_2)\cdots\mathbb{P}(A_n \,|\, A_1\cdots A_{n-1})\ . \tag{3.5}$$

■ **Example 3.5 (Urn Problem)** We draw consecutively 3 balls from an urn with 5 white and 5 black balls, without putting them back. What is the probability that all drawn balls will be black?

Let A_i be the event that the i-th ball is black. We wish to find the probability of $A_1A_2A_3$, which by the product rule (3.5) is

$$\mathbb{P}(A_1)\,\mathbb{P}(A_2 \,|\, A_1)\,\mathbb{P}(A_3 \,|\, A_1A_2) = \frac{5}{10}\,\frac{4}{9}\,\frac{3}{8} \approx 0.083\ .$$

■

3.6 Random Variables and their Distributions

Specifying a model for a random experiment via a complete description of the sample space Ω and probability measure $\mathbb{P}$ may not always be necessary or convenient. In practice we are only interested in certain *numerical measurements* pertaining to the experiment. Such random measurements can be included into the model via the notion of a **random variable**. A random variable can be viewed as an observation of a random experiment that has not yet taken place. In other words, a random variable can be considered as a measurement that becomes available *tomorrow*, while all the thinking about the measurement can be carried out *today*. For example, we can specify today exactly the probabilities pertaining to the random variables.

We often denote random variables with *capital* letters from the last part of the alphabet, e.g., $X, X_1, X_2, \ldots, Y, Z$. Random variables allow us to use natural and intuitive notations for certain events, such as $\{X = 10\}$, $\{X > 1000\}$, $\{\max(X, Y) \leqslant Z\}$, etc.

> Mathematically, a random variable is a *function* which assigns a numerical value (measurement) to each outcome. An event such as $\{X > 1000\}$ is to be interpreted as the set of outcomes for which the corresponding measurement is greater than 1000.

We give some more examples of random variables without specifying the sample space:

1. The number of defective transistors out of 100 inspected ones.
2. The number of bugs in a computer program.
3. The amount of rain in a certain location in June.
4. The amount of time needed for an operation.

☞ 23

Similar to our discussion of the data types in Chapter 2, we distinguish between discrete and continuous random variables:

- **Discrete** random variables can only take *countably many* values.
- **Continuous** random variables can take a continuous range of values; for example, any value on the positive real line $\mathbb{R}_+$.

Let X be a random variable. We would like to designate the probabilities of events such as $\{X = x\}$ and $\{a \leqslant X \leqslant b\}$. If we can specify all probabilities involving X, we say that we have determined the **probability distribution** of X. One way to specify the probability distribution is to give the probabilities of all events of the form $\{X \leqslant x\}$. This leads to the following definition.

Definition 3.6: Cumulative Distribution Function

The **cumulative distribution function (cdf)** of a random variable X is the function F defined by

$$F(x) = \mathbb{P}(X \leqslant x), \quad x \in \mathbb{R} .$$

We have used $\mathbb{P}(X \leqslant x)$ as a shorthand notation for $\mathbb{P}(\{X \leqslant x\})$. From now on we will use this type of abbreviation throughout the notes. In Figure 3.7 the graph of a general cdf is depicted. Note that any cdf is increasing (if $x \leqslant y$ then $F(x) \leqslant F(y)$) and lies between 0 and 1. We can use any function F with these properties to specify the distribution of a random variable X.

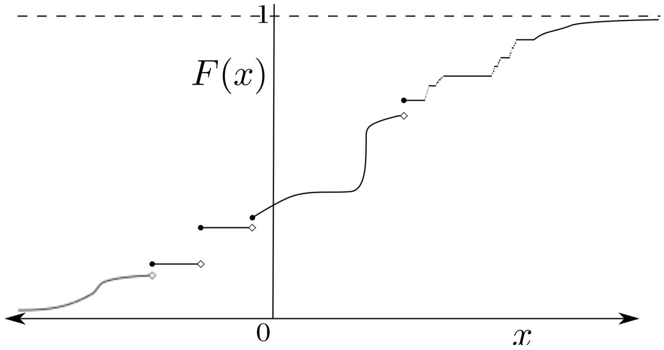


Figure 3.7: A cumulative distribution function (cdf).

If X has cdf F, then the probability that X takes a value in the interval $(a, b]$ (excluding a, including b) is given by

$$\mathbb{P}(a < X \leqslant b) = F(b) - F(a) .$$

To see this, note that $\mathbb{P}(X \leqslant b) = \mathbb{P}(\{X \leqslant a\} \cup \{a < X \leqslant b\})$, where the events $\{X \leqslant a\}$ and $\{a < X \leqslant b\}$ are disjoint. Thus, by the sum rule: $F(b) = F(a) + \mathbb{P}(a < X \leqslant b)$, which leads to the result above.

Definition 3.7: Probability Mass Function

A random variable X is said to have a **discrete distribution** if $\mathbb{P}(X = x_i) > 0$, $i = 1, 2, \ldots$ for some finite or countable set of values $x_1, x_2, \ldots$, such that $\sum_i \mathbb{P}(X = x_i) = 1$. The **probability mass function (pmf)** of X is the function f defined by $f(x) = \mathbb{P}(X = x)$.

We sometimes write f_X instead of f to stress that the pmf refers to the discrete random variable X. The easiest way to specify the distribution of a discrete random variable is to specify its pmf. Indeed, by the sum rule, if we know $f(x)$ for all x, then ☞ 48 we can calculate all possible probabilities involving X. In particular, the probability that X lies in some set B (say an interval (a,b)) is

$$\mathbb{P}(X \in B) = \sum_{x \in B} f(x) \,, \tag{3.6}$$

as illustrated in Figure 3.8. Note that $\{X \in B\}$ should be read as "X is an element of region B".

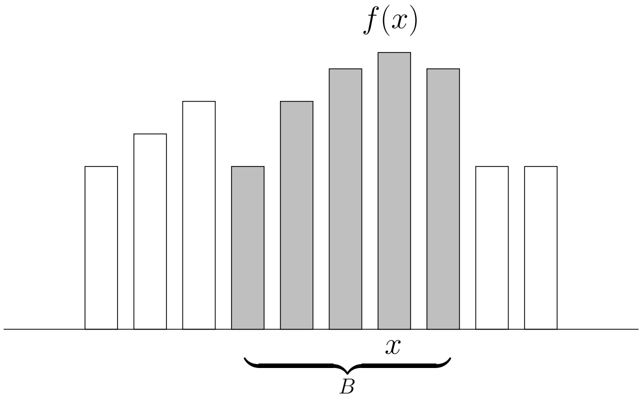


Figure 3.8: Probability mass function (pmf).

■ **Example 3.6 (Sum of Two Dice)** Toss two fair dice and let X be the sum of their face values. The pmf is given in Table 3.2.

Table 3.2: Pmf of the sum of two fair dice.

x	2	3	4	5	6	7	8	9	10	11	12
$f(x)$	$\frac{1}{36}$	$\frac{2}{36}$	$\frac{3}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{5}{36}$	$\frac{4}{36}$	$\frac{3}{36}$	$\frac{2}{36}$	$\frac{1}{36}$

■

For a continuous random variable, it makes no sense to consider probabilities of the form $\mathbb{P}(X = x)$, as every such probability is zero! Instead of a probability mass function, we have to use a probability density function, which is defined as follows.

Definition 3.8: Probability Density Function

A random variable X with cdf F is said to have a **continuous distribution** if there exists a positive function f with *total integral 1* such that for all $a < b$,

$$\mathbb{P}(a < X \leqslant b) = F(b) - F(a) = \int_a^b f(u)\,\mathrm{d}u\ . \tag{3.7}$$

Function f is called the **probability density function (pdf)** of X.

⚠ Note that we use the *same* notation f for both the pmf and pdf, to stress the similarities between the discrete and continuous case. Henceforth we will use the notation $X \sim f$ and $X \sim F$ to indicate that X is distributed according to pdf f or cdf F.

In analogy to the discrete case (3.6), once we know the pdf, we can calculate any probability that X lies in some set B by means of integration:

$$\mathbb{P}(X \in B) = \int_B f(x)\,\mathrm{d}x\ , \tag{3.8}$$

as illustrated in Figure 3.9.

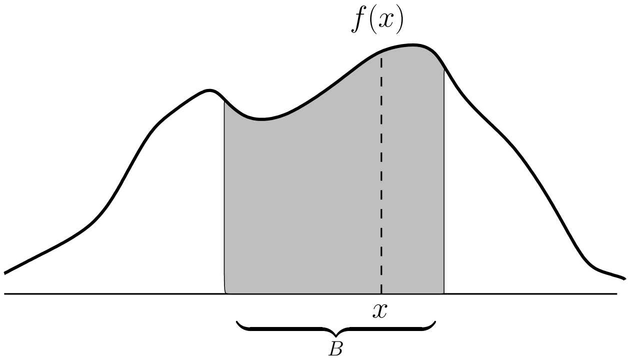


Figure 3.9: Probability density function (pdf).

Suppose that f and F are the pdf and cdf of a continuous random variable X, as in Definition 3.6. Then F is simply a *primitive* (also called anti-derivative) of f:

$$F(x) = \mathbb{P}(X \leqslant x) = \int_{-\infty}^{x} f(u)\,\mathrm{d}u\ .$$

Conversely, f is the *derivative* of the cdf F:

$$f(x) = \frac{\mathrm{d}}{\mathrm{d}x}F(x) = F'(x)\ .$$

It is important to understand that in the continuous case $f(x)$ is not equal to the probability $\mathbb{P}(X = x)$, because the latter is 0 for all x. Instead, we interpret $f(x)$ as the *density* of the probability distribution at x, in the sense that for any small h,

$$\mathbb{P}(x \leqslant X \leqslant x + h) = \int_x^{x+h} f(u)\,\mathrm{d}u \approx h\,f(x)\,. \tag{3.9}$$

Note that $\mathbb{P}(x \leqslant X \leqslant x + h)$ is equal to $\mathbb{P}(x < X \leqslant x + h)$ in this case.

■ **Example 3.7 (Random Point in an Interval)** Draw a random number X from the interval of real numbers $[0, 2]$, where each number is equally likely to be drawn. What are the pdf f and cdf F of X? We have

$$\mathbb{P}(X \leqslant x) = F(x) = \begin{cases} 0 & \text{if } x < 0, \\ x/2 & \text{if } 0 \leqslant x \leqslant 2, \\ 1 & \text{if } x > 2. \end{cases}$$

By differentiating F we find

$$f(x) = \begin{cases} 1/2 & \text{if } 0 \leqslant x \leqslant 2, \\ 0 & \text{otherwise.} \end{cases}$$

Note that this density is *constant* on the interval $[0, 2]$ (and zero elsewhere), reflecting the fact that each point in $[0, 2]$ is equally likely to be drawn. ■

3.7 Expectation

Although all probability information about a random variable is contained in its cdf or pmf/pdf, it is often useful to consider various numerical characteristics of a random variable. One such number is the *expectation* of a random variable, which is a “weighted average” of the values that X can take. Here is a more precise definition.

Definition 3.9: Expectation (Discrete)

Let X be a *discrete* random variable with pmf f. The **expectation** (or expected value) of X, denoted as $\mathbb{E}(X)$, is defined as

$$\mathbb{E}(X) = \sum_x x\,\mathbb{P}(X = x) = \sum_x x\,f(x)\,. \tag{3.10}$$

The expectation of X is sometimes written as μ_X. It is assumed that the sum in (3.10) is well-defined — possibly infinity (∞) or minus infinity ($-\infty$). One way to interpret the expectation is as a *long-run average payout*, as illustrated in the following example.

■ **Example 3.8 (Expected Payout)** Suppose in a game of dice the payout X (dollars) is the largest of the face values of two dice. To play the game a fee of d dollars must be paid. What would be a fair amount for d? If the game is played many times, the long-run fraction of tosses in which the maximum face value takes the value 1, 2,. . . , 6, is $\mathbb{P}(X = 1), \mathbb{P}(X = 2), \ldots, \mathbb{P}(X = 6)$, respectively. Hence, the long-run average payout of the game is the weighted sum of $1, 2, \ldots, 6$, where the weights are the long-run fractions (probabilities). So, the long-run payout is

$$\begin{aligned}\mathbb{E}X =& 1 \times \mathbb{P}(X = 1) + 2 \times \mathbb{P}(X = 2) + \cdots + 6 \times \mathbb{P}(X = 6) \\ &= 1 \times \frac{1}{36} + 2 \times \frac{3}{36} + 3 \times \frac{5}{36} + 4 \times \frac{7}{36} + 5 \times \frac{9}{36} + 6 \times \frac{11}{36} = \frac{161}{36} \approx 4.47 \ .\end{aligned}$$

The game is "fair" if the long-run average profit $\mathbb{E}(X) - d$ is zero, so you should maximally wish to pay $d = \mathbb{E}(X)$ dollars. ■

For a *symmetric* pmf/pdf the expectation (if finite) is equal to the symmetry point.

For continuous random variables we can define the expectation in a similar way, replacing the sum with an integral.

Definition 3.10: Expectation (Continuous)

Let X be a *continuous* random variable with pdf f. The **expectation** (or expected value) of X, denoted as $\mathbb{E}(X)$, is defined as

$$\mathbb{E}(X) = \int_{-\infty}^{\infty} x\, f(x)\, \mathrm{d}x \ . \tag{3.11}$$

If X is a random variable, then a function of X, such as X^2 or $\sin(X)$, is also a random variable. The following theorem simply states that the expected value of a function of X is the weighted average of the values that this function can take.

Theorem 3.6: Expectation of a Function of a Random Variable

If X is discrete with pmf f, then for any real-valued function g

$$\mathbb{E}(g(X)) = \sum_x g(x)\, f(x) \ .$$

Replace the sum with an integral for the continuous case.

■ **Example 3.9 (Die Experiment and Expectation)** Find $\mathbb{E}(X^2)$ if X is the outcome of the toss of a fair die. We have

$$\mathbb{E}(X^2) = 1^2 \times \frac{1}{6} + 2^2 \times \frac{1}{6} + 3^2 \times \frac{1}{6} + \cdots + 6^2 \times \frac{1}{6} = \frac{91}{6} .$$

■

An important consequence of Theorem 3.6 is that the expectation is "linear".

Theorem 3.7: Properties of the Expectation

For any real numbers a and b, and functions g and h,

1. $\mathbb{E}(a\,X + b) = a\,\mathbb{E}(X) + b$,
2. $\mathbb{E}(g(X) + h(X)) = \mathbb{E}(g(X)) + \mathbb{E}(h(X))$.

Proof: We show it for the discrete case. The continuous case is proven analogously, simply by replacing sums with integrals. Suppose X has pmf f. The first statement follows from

$$\mathbb{E}(aX + b) = \sum_x (ax + b) f(x) = a \sum_x x\, f(x) + b \sum_x f(x) = a\,\mathbb{E}(X) + b .$$

Similarly, the second statement follows from

$$\begin{aligned}\mathbb{E}(g(X) + h(X)) &= \sum_x (g(x) + h(x)) f(x) = \sum_x g(x) f(x) + \sum_x h(x) f(x) \\ &= \mathbb{E}(g(X)) + \mathbb{E}(h(X)) .\end{aligned}$$

□

Another useful numerical characteristic of the distribution of X is the *variance* of X. This number, sometimes written as σ_X^2, measures the *spread* or dispersion of the distribution of X.

Definition 3.11: Variance

The **variance** of a random variable X, denoted as $\mathrm{Var}(X)$, is defined as

$$\mathrm{Var}(X) = \mathbb{E}(X - \mu)^2 , \tag{3.12}$$

where $\mu = \mathbb{E}(X)$. The square root of the variance is called the **standard deviation**. The number $\mathbb{E}X^r$ is called the r-th **moment** of X.

Theorem 3.8

(Properties of the Variance). For any random variable X the following properties hold for the variance.

1. $\mathrm{Var}(X) = \mathbb{E}X^2 - (\mathbb{E}X)^2$.

2. $\mathrm{Var}(a + bX) = b^2\,\mathrm{Var}(X)$.

Proof: To see this, write $\mathbb{E}(X) = \mu$, so that $\mathrm{Var}(X) = \mathbb{E}(X - \mu)^2 = \mathbb{E}(X^2 - 2\mu X + \mu^2)$. By the linearity of the expectation, the last expectation is equal to the sum $\mathbb{E}(X^2) - 2\,\mu\,\mathbb{E}(X) + \mu^2 = \mathbb{E}(X^2) - \mu^2$, which proves the first statement. To prove the second statement, note that the expectation of $a + bX$ is equal to $a + b\mu$. Consequently,

$$\mathrm{Var}(a + bX) = \mathbb{E}\left((a + bX - (a + b\mu))^2\right) = \mathbb{E}(b^2(X - \mu)^2) = b^2\mathrm{Var}(X)\ .$$

□

CHAPTER 4

Common Probability Distributions

This chapter presents four probability distributions that are the most frequently used in the study of statistics: the Bernoulli, Binomial, Uniform, and Normal distributions. We give various properties of these distributions and show how to compute probabilities of interest for them. You will also learn how to simulate random data from these distributions.

4.1 Introduction

In the previous chapter, we have seen that a random variable that takes values in a continuous set (such as an interval) is said to be *continuous* and a random variable that can have only a finite or countable number of different values is said to be discrete; see Section 3.6. Recall that the distribution of a continuous variable is specified by its ☞ 56 *probability density function* (pdf), and the distribution of a discrete random variable by its *probability mass function* (pmf).

In the following, we first present two distributions for discrete variables: they are the Bernoulli and Binomial distributions. Then, we describe two key distributions for continuous variables: the Uniform and Normal distributions. All of these distributions are actually *families* of distributions, which depend on a few (one or two in this case) *parameters* — fixed values that determine the shape of the distribution. Although in statistics we only employ a relatively small collection of distribution families (binomial, normal, etc.), we can make an infinite amount of distributions through parameter selection.

4.2 Bernoulli Distribution

A **Bernoulli trial** is a random experiment that has only two possible outcomes, usually labeled “success” (or 1) and “failure” (or 0). The corresponding random variable X is called a **Bernoulli variable**. For example, a Bernoulli variable could model a single coin toss experiment by attributing the value 1 for Heads and 0 for Tails. Another example is selecting at random a person from some population and asking him if he/she approves of the prime minister or not.

Definition 4.1: Bernoulli Distribution

A random variable X is said to have a **Bernoulli** distribution with success probability p if X can only assume the values 0 and 1, with probabilities

$$\mathbb{P}(X = 1) = p \quad \text{and} \quad \mathbb{P}(X = 0) = 1 - p \, .$$

We write $X \sim \mathsf{Ber}(p)$.

Figure 4.1 gives the pmf of a Bernoulli random variable.

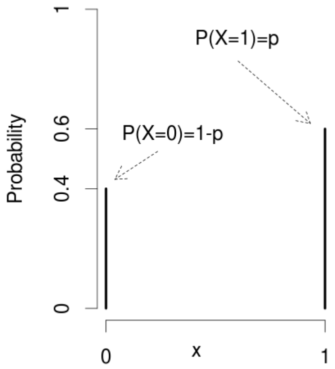


Figure 4.1: Probability mass function for the Bernoulli distribution, with parameter p (the case $p = 0.6$ is shown)

The expectation and variance of $X \sim \mathsf{Ber}(p)$ are easy to determine. We leave the proof as an exercise, as it is instructive do do it yourself, using the definitions of the expectation and variance; see (3.10) and (3.12). ☞ 60

Theorem 4.1: Expectation and Variance of the Bernoulli Distribution

Let $X \sim \mathsf{Ber}(p)$. Then,

1. $\mathbb{E}(X) = p$
2. $\mathrm{Var}(X) = p(1 - p)$

4.3 Binomial Distribution

Let us go back the coin flip experiment of Example 1.1. In particular, we flip a coin ☞ 10
100 times and count the number of success (Heads), say X. Suppose that the coin is fair. What is the distribution of the total number of successes X? Obviously X can take any of the values 0,1,...,100. So let us calculate the probability of x successes: $\mathbb{P}(X = x)$ for $x = 0, 1, \ldots, 100$. In other words we wish to derive the pmf of X. In this case we can use a counting argument, as in Section 3.4. ☞ 49
Namely, if we note the sequence of 100 tosses, there are 2^{100} possible outcomes of the experiment, and they are all equally likely (with a fair coin). To calculate the probability of having exactly x successes (1s) we need to see how many of the possible outcomes have exactly x 1s and $100 - x$ 0s. There are $\binom{100}{x}$ of these, because we have to choose exactly x positions for the 1s out of 100 possible positions. In summary, we have derived

$$\mathbb{P}(X = x) = \frac{\binom{100}{x}}{2^{100}}, \quad x = 0, 1, 2, \ldots, 100 \,.$$

This is an example of a *Binomial distribution*. We can now calculate probabilities of interest such as $\mathbb{P}(X \geqslant 60)$, which we said in Example 1.1 was approximately equal to 0.028. Let us check this, using `R` as a calculator. We need to evaluate

$$\mathbb{P}(X \geqslant 60) = \sum_{x=60}^{100} \frac{\binom{100}{x}}{2^{100}} = \frac{\sum_{x=60}^{100} \binom{100}{x}}{2^{100}} \,.$$

We can do this in `R` in one line:

```
> sum(choose(100,60:100))/2^(100)
```

```
[1] 0.02844397
```

More generally, when we toss a coin n times and the probability of Heads is p (not necessarily 1/2), the outcomes are no longer equally likely (for example, when p is close to 1 the sequence coin flips $1, 1, \ldots, 1$ is more likely to occur than $0, 0, \ldots, 0$). We can use the product rule (3.5) to find that the probability of having a particular sequence ☞ 55 with x heads and $n - x$ tails is $p^x(1-p)^{n-x}$. Since there are $\binom{n}{x}$ of these sequences, we see that X has a $\mathsf{Bin}(n, p)$ distribution, as given in the following definition.

Definition 4.2: Binomial Distribution

A random variable X is said to have a **Binomial** distribution with parameters n and p if X can only assume the integer values $x = 0, 1, \ldots, n$, with probabilities

$$\mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, \ldots, n \,. \tag{4.1}$$

We write $X \sim \mathsf{Bin}(n, p)$.

Figure 4.2 shows the pmf of the $\mathsf{Bin}(10, 0.7)$ distribution.

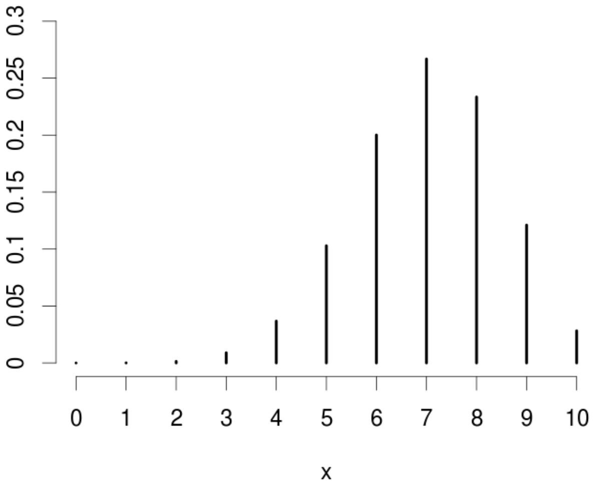


Figure 4.2: Probability mass function of the $\mathsf{Bin}(10, 0.7)$ distribution.

The following theorem lists the expectation and variance for the $\mathsf{Bin}(n, p)$ distribution. A simple proof will be given in the next chapter; see Example 5.4. In any case, the expression for the expectation should come as no surprise, as we would expect np successes in a sequence of Bernoulli experiments (coin flips) with success probability p. Note that both the expectation and variance are n times the expectation and variance of a $\mathsf{Ber}(p)$ random variable. This is no coincidence, as a Binomial random variable can be seen as the sum of n independent Bernoulli random variables.

☞ 87

Theorem 4.2: Expectation and Variance of the Binomial Distribution

Let $X \sim \mathsf{Bin}(n, p)$. Then,

1. $\mathbb{E}(X) = np$
2. $\mathrm{Var}(X) = np(1 - p)$

The number of successes in a series of n independent Bernoulli trials with success probability p has a $\mathsf{Bin}(n, p)$ distribution.

Counting the number of successes in a series of coin flip experiments might seem a bit artificial, but it is important to realize that many practical statistical situations can be treated exactly as a sequence of coin flips. For example, suppose we wish to conduct a survey of a large population to see what the proportion p is of males, where p is unknown. We can only know p if we survey *everyone* in the population, but suppose we do not have the resources or time to do this. Instead we select at random n people from the population and note their gender. We assume that each person is chosen with equal probability. This is very much like a coin flipping experiment. In fact, if we allow the same person to be selected more than once, then the two situations are *exactly* the same. Consequently, if X is the total number of males in the group of n selected persons, then $X \sim \mathsf{Bin}(n, p)$. You might, rightly, argue that in practice we would not select the same person twice. But for a large population this would rarely happen, so the Binomial model is still a good model. For a small population, however, we should use a (more complicated) urn model to describe the experiment, where we draw balls (select people) without replacement and without noting the order. Counting for such experiments was discussed in Section 3.4. ☞ 49

4.4 Uniform Distribution

The simplest continuous distribution is the uniform distribution.

Definition 4.3: Uniform Distribution

A random variable X is said to have a **uniform** distribution on the interval $[a, b]$ if its pdf is given by

$$f(x) = \frac{1}{b - a}, \quad a \leqslant x \leqslant b \quad \text{(and } f(x) = 0 \text{ otherwise).} \tag{4.2}$$

We write $X \sim \mathcal{U}[a, b]$.

A random variable $X \sim \mathcal{U}[a, b]$ can model a randomly chosen point from the interval $[a, b]$, where each choice is equally likely. A graph of the density function is given in Figure 4.3. Note that the total area under the pdf is 1.

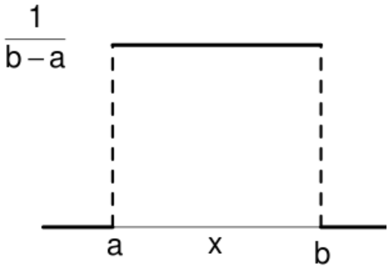


Figure 4.3: Probability density function for a uniform distribution on $[a, b]$

Theorem 4.3: Properties of the Uniform Distribution

Let $X \sim \mathcal{U}[a, b]$. Then,

1. $\mathbb{E}(X) = (a + b)/2$
2. $\mathrm{Var}(X) = (b - a)^2/12$

Proof: The expectation is finite (since it must lie between a and b) and the pdf is symmetric. It follows that the expectation is equal to the symmetry point $(a + b)/2$. To find the variance, it is useful to write $X = a + (b - a)U$ where $U \sim \mathcal{U}[0, 1]$. In words: randomly choosing a point between a and b is equivalent to first randomly choosing a point in $[0, 1]$, multiplying this by $(b - a)$, and adding a. We can now write $\mathrm{Var}(X) = \mathrm{Var}(a + (b - a)U)$, which is the same as $(b - a)^2\mathrm{Var}(U)$, using the second property for the variance in Theorem 3.7. So, it suffices to show that $\mathrm{Var}(U) = 1/12$. Writing $\mathrm{Var}(U) = \mathbb{E}(U^2) - (\mathbb{E}(U))^2 = \mathbb{E}(U^2) - 1/4$, it remains to show that $\mathbb{E}(U^2) = 1/3$. This follows by direct integration:

☞ 63

$$\mathbb{E}(U^2) = \int_0^1 u^2 1 \mathrm{d}u = \frac{1}{3}u^3\Big|_0^1 = \frac{1}{3}\,.$$

□

4.5 Normal Distribution

We now introduce the most important distribution in the study of statistics: the normal (or Gaussian) distribution.

Definition 4.4: Normal (or Gaussian) Distribution

A random variable X is said to have a **normal** or **Gaussian** distribution with parameters μ (expectation) and σ^2 (variance) if its pdf is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\,\mathrm{e}^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2},\ x \in \mathbb{R} \tag{4.3}$$

We write $X \sim \mathcal{N}(\mu, \sigma^2)$.

The parameters μ and σ^2 turn out to be the expectation and variance of the distribution, respectively. If $\mu = 0$ and $\sigma = 1$ then the distribution is known as the **standard normal** distribution. Its pdf is often denoted by φ (phi), so

$$\varphi(x) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-x^2/2}, \quad x \in \mathbb{R}.$$

The corresponding cdf is denoted by Φ (capital phi). In Figure 4.4 the density function of the $\mathcal{N}(\mu, \sigma^2)$ distribution for various μ and σ^2 is plotted.

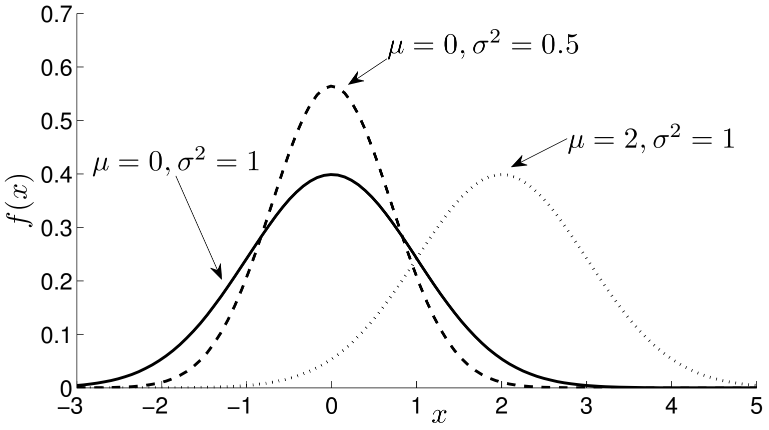


Figure 4.4: Probability density functions for various Normal distributions

You may verify yourself, by applying the definitions of expectation and variance, that indeed the following theorem holds:

Theorem 4.4: Properties of the Normal Distribution

Let $X \sim \mathcal{N}(\mu, \sigma^2)$. Then,

1. $\mathbb{E}(X) = \mu$
2. $\mathrm{Var}(X) = \sigma^2$

The normal distribution is symmetric about the expectation μ and the dispersion is controlled by the variance parameter σ^2, or the standard deviation σ (see Figure 4.4). An important property of the normal distribution is that any normal random variable can be thought of as a simple transformation of a standard normal random variable.

Theorem 4.5: Standardization

If Z has standard normal distribution, then $X = \mu + \sigma Z$ has a $\mathcal{N}(\mu, \sigma^2)$ distribution. Consequently, if $X \sim \mathcal{N}(\mu, \sigma^2)$ then the **standardized** random variable

$$Z = \frac{X - \mu}{\sigma} \tag{4.4}$$

has a standard normal distribution.

Proof: Suppose Z is standard normal. So, $\mathbb{P}(Z \leqslant z) = \Phi(z)$ for all z. Let $X = \mu + \sigma Z$. We wish to derive the pdf f of X and show that it is of the form (4.3). We first derive the cdf F:

$$F(x) = \mathbb{P}(X \leqslant x) = \mathbb{P}(\mu + \sigma Z \leqslant x) = \mathbb{P}(Z \leqslant (x - \mu)/\sigma) = \Phi((x - \mu)/\sigma) \ .$$

By taking the derivative $f(x) = F'(x)$ we find (apply the chain rule of differentiation):

$$f(x) = F'(x) = \Phi'((x - \mu)/\sigma)\frac{1}{\sigma} = \varphi((x - \mu)/\sigma)/\sigma \ ,$$

which is the pdf of a $\mathcal{N}(\mu, \sigma^2)$-distributed random variable (replace x with $(x - \mu)/\sigma$ in the formula for φ and divide by σ. This gives precisely (4.3)). □

By using the standardization (4.4) we can simplify calculations involving arbitrary normal random variables to calculations involving only standard normal random variables.

■ **Example 4.1 (Standardization)** Standardization can be viewed as a way to make comparisons between normal populations on the same scale. Suppose female heights are Normally distributed with mean 168 cm and variance 36 cm^2 and male heights are Normally distributed with mean 179 cm and variance 49 cm^2. Who is the more unusually tall for her/his gender, a female who is taller than 180 cm or a male who is taller than 200 cm? Let us denote by X and Y the heights of a randomly selected woman and man, respectively. The probability that the female is taller than 180 cm is equal to

$$\begin{aligned}\mathbb{P}(X \geqslant 180) &= \mathbb{P}(X - 168 > 180 - 168) \\ &= \mathbb{P}\left(\frac{X - 168}{6} > \frac{180 - 168}{6}\right) \\ &= \mathbb{P}(Z \geqslant 2) = 1 - \mathbb{P}(Z \leqslant 2) = 1 - \Phi(2) \ .\end{aligned}$$

For the male we have, similarly,

$$\begin{aligned}\mathbb{P}(Y \geqslant 200) &= \mathbb{P}\left(\frac{Y-179}{7} > \frac{200-179}{7}\right)\\ &= \mathbb{P}(Z > 3) = 1 - \Phi(3)\ .\end{aligned}$$

Since $\Phi(3)$ is larger than $\Phi(2)$, finding a male to be taller than 2m is more unusual than finding a female taller than 180cm.

In the days before the computer it was customary to provide tables of $\Phi(x)$ for $0 \leqslant x \leqslant 4$, say. Nowadays we can simply use statistical software. For example, the cdf Φ is encoded in R as the function `pnorm`. So to find $1 - \Phi(2)$ and $1 - \Phi(3)$ we can type:

```
> 1 - pnorm(2)

[1] 0.02275013

> 1 - pnorm(3)

[1] [1] 0.001349898
```

Unfortunately there is no simple formula for working out areas under the Normal density curve. However, as a rough rule for $X \sim \mathcal{N}(\mu, \sigma^2)$:

- the area within $c = 1$ standard deviation of the mean is 68%
- the area within $c = 2$ standard deviations of the mean is 95%
- the area within $c = 3$ standard deviations of the mean is 99.7%

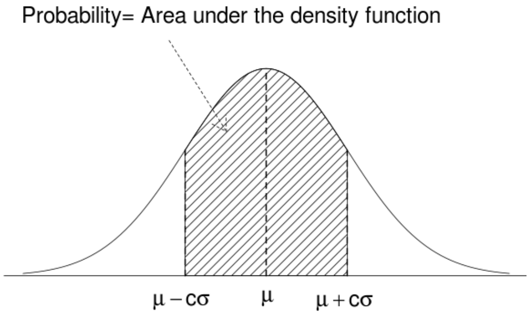


Figure 4.5: The area of the shaded region under the pdf is the probability $\mathbb{P}(|X - \mu| \leqslant c)$ that X lies less than c standard deviations (σ) away from its expectation (μ)

The function `pnorm` can also be used to evaluate the cdf of general normal distribution. For example, let $X \sim \mathcal{N}(1, 4)$. Suppose we wish to find $\mathbb{P}(X \leqslant 3)$. In R we can enter:

```
> pnorm(3,mean=1,sd=2)

[1] 0.8413447
```

Note that R uses the standard deviation as an argument, not the variance!

We can also go the other way around: let $X \sim \mathcal{N}(1, 4)$. For what value z does it hold that $\mathbb{P}(X \leqslant z) = 0.9$. Such a value z is called a **quantile** of the distribution — in this case the 0.9-quantile. The concept is closely related to the *sample quantile* discussed ☞ 28 in Section 2.4, but the two are not the same. Figure 4.6 gives an illustration. For the normal distribution the quantiles can be obtained via the R function **qnorm**.

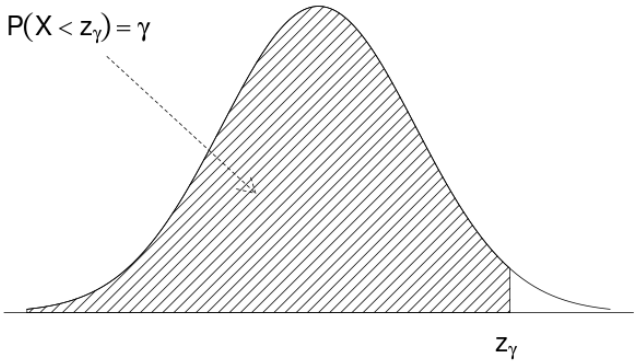


Figure 4.6: z_γ is the γ quantile of a normal distribution.

Here are some examples.

```
> qnorm(0.975)

[1] 1.959964


> qnorm(0.90,mean=1,sd=2)

[1] 3.563103


> qnorm(0.5,mean=2,sd=1)

[1] 2
```

4.6 Simulating Random Variables

This section shows how to generate (simulate) random variables on a computer. We first introduce R functions to generate observations from main distributions and then present some graphical tools to investigate the distribution of the simulated data.

Many computer programs have an inbuilt **random number generator**. This is a program that produces a stream of numbers between 0 and 1 that for all intent and purposes behave like independent draws from a uniform distribution on the interval [0,1]. Such numbers can be produced by the function `runif`. For example

```
> runif(1)

[1] 0.6453129
```

Repeating gives a different number

```
> runif(1)

[1] 0.8124339
```

Or we could produce 5 such numbers in one go.

```
> runif(5)

[1] 0.1813849 0.9126095 0.2082720 0.1540227 0.9572725
```

We can use a uniform random number to simulate a toss with a fair coin by returning TRUE if $x < 0.5$ and FALSE if $x \geqslant 0.5$.

```
> runif(1) < 0.5

[1]  TRUE
```

We can turn the logical numbers into 0s and 1s by by using the function `as.integer`

```
>  as.integer(runif(20)<0.5)

 [1] 1 1 0 1 0 0 1 0 1 0 0 1 1 1 1 1 0 0 0 0
```

We can, in principle, draw from *any* probability distribution including the normal distribution, using *only* uniform random numbers. However, to draw from a normal distribution we will use R's inbuilt `rnorm` function. For example, the following generates 5 outcomes from the standard normal distribution:

```
> rnorm(5)

-1.1871560 -0.9576287 -1.2217339 -0.0412956  0.4981450
```

In R, every function for generating **r**andom variables starts with an “r” (e.g., **runif**, **rnorm**). This is also holds for discrete random variables:

```
>  rbinom(1,size=10,p=0.5)
[1] 5
```

corresponds to the realization of a random variable $X \sim \mathsf{Bin}(10, 0.5)$ and the instruction

```
>  rbinom(1,size=1,p=0.5)
[1] 1
```

corresponds to the realization of a random variable $X \sim \mathsf{Ber}(0.5)$.

Generating artificial data can be a very useful way to understand probability distributions. For example, if we generate many realizations from a certain distribution, then ☞ 30 the histogram and empirical cdf of the data (see Section 2.5) will resemble closely the true pdf/pmf and cdf of the distribution. Moreover the summary statistics (see Sec- ☞ 28 tion 2.4) of the simulated data such as the sample mean and sample quantiles will resemble the true distributional properties such as the expected value and the quantiles. Let us illustrate this by drawing one 10,000 samples from the $\mathcal{N}(2, 1)$ distribution.

```
> x = rnorm(10e4,mean=2,sd=1)
> summary(x)


  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.573   1.328   1.997   1.997   2.670   5.865
```

The true first and third quartiles are 1.32551 and 2.67449, respectively, which are quite close to the sample quartiles. Similarly the true expectation and median are 2, which is again close to the sample mean and sample median.

The following R script (program) was used to produce Figure 4.7. We see a very close correspondence between the true pdf (on the left, in red) and a histogram of the 10,000 data points. The true cdf (on the right, in red) is virtually indistinguishable from the empirical cdf.

```
# simnorm.R
par(mfrow=c(1,2),cex=1.5)     # two plot windows, use larger font
x = rnorm(10e4,mean=2,sd=1)   # generate data
hist(x,prob=TRUE,breaks=100)  # make histogram
curve(dnorm(x,mean=2,sd=1),col="red",ylab="",lwd=2,add=T)  #true pdf
plot(ecdf(x)) # draw the empirical cdf
curve(pnorm(x,mean=2,sd=1),col="red",lwd=1,add=TRUE)       #true cdf
```

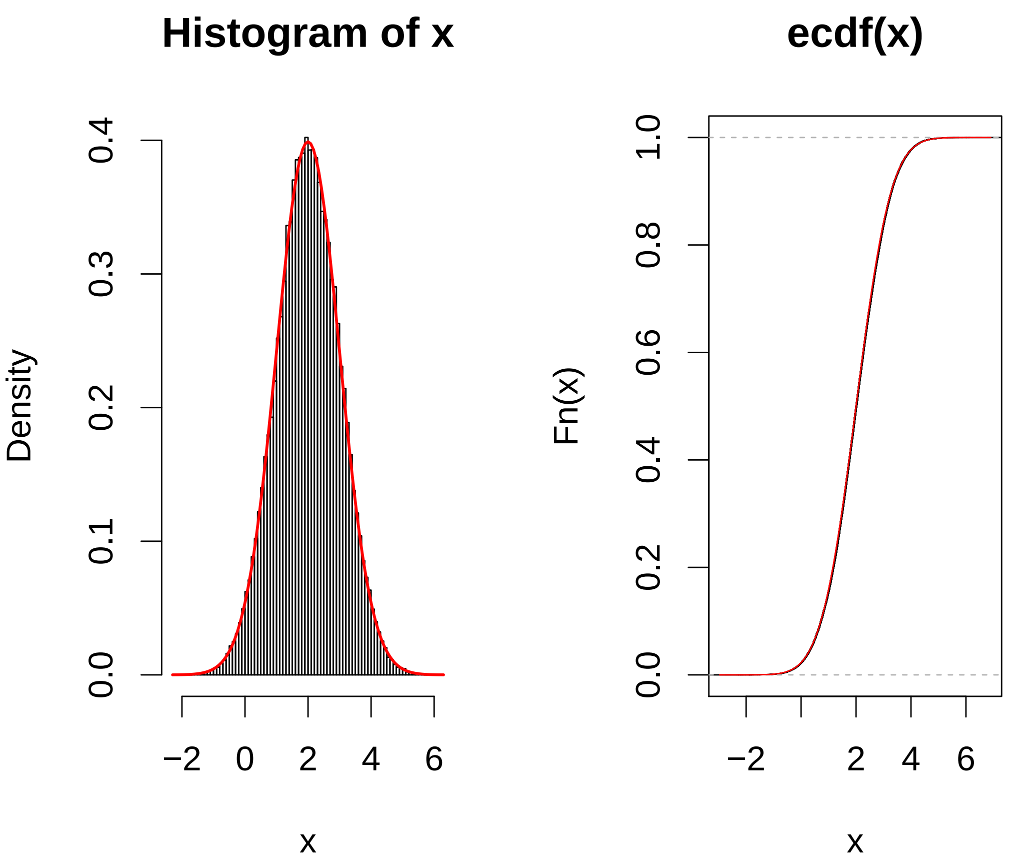


Figure 4.7: Left: pdf of the $\mathcal{N}(2, 1)$ distribution (red) and histogram of the generated data. Right: cdf of the $\mathcal{N}(2, 1)$ distribution (red) empirical cdf of the generated data.

Density functions (pmf or pdf) always start in R with “d” (e.g., **dnorm**, **dunif**). The cummulative distribution functions (cdf), which give a **p**robability, always start in R with “p” (e.g., **pnorm**, **punif**). **Q**uantiles start with “q” (e.g., **qnorm**,**qunif**).

To summarize, we present in table 4.1 the main R functions for the evaluation of densities, cumulative distribution functions, quantiles, and the generation of random variables for the distributions described in this chapter. Later on we will encounter more distributions such as the Student’s t distribution, the F distribution, and the chi-squared distribution. You can use the “d”, “p”, “q” and “r” construction to evaluate pmfs, cdfs, quantiles, and random numbers in exactly the same way!

Table 4.1: Standard discrete and continuous distributions. `R` functions for the mass or density function (`d-`), cumulative distribution function (`p-`) and quantile function (`q-`). Instruction to generate (`r-`) pseudo-random numbers from these distributions.

Distr.	**R functions**	**Distr.**	**R functions**
Ber(p)	`dbinom(x,size=1,prob=`p`)`	$\mathcal{N}(\mu, \sigma^2)$	`dnorm(x,mean=`μ`,sd=`σ`)`
	`pbinom(x,size=1,prob=`p`)`		`pnorm(x,mean=`μ`,sd=`σ`)`
	`qbinom(`γ`,size=1,prob=`p`)`		`qnorm(`γ`,mean=`μ`,sd=`σ`)`
	`rbinom(n,size=1,prob=`p`)`		`rnorm(n,mean=`μ`,sd=`σ`)`
Bin(n, p)	`dbinom(x,size=`n`,prob=`p`)`	$\mathcal{U}[a, b]$	`dunif(x,min=`a`,max=`b`)`
	`pbinom(x,size=`n`,prob=`p`)`		`punif(x,min=`a`,max=`b`)`
	`qbinom(`γ`,size=`n`,prob=`p`)`		`qunif(`γ`,min=`a`,max=`b`)`
	`rbinom(n,size=`n`,prob=`p`)`		`runif(n,min=`a`,max=`b`)`

CHAPTER 5

Multiple Random Variables

In this chapter you will learn how random experiments that involve more than one random variable can be described via their joint cdf and joint pmf/pdf. When the random variables are *independent* of each other, the joint density has a simple product form. We will discuss the most basic statistical model for data — independent and identically distributed (iid) draws from a common distribution. We will show that the expectation and variance of sums of random variables obey simple rules. We will also illustrate the *central limit theorem*, explaining the central role that the normal distribution has in statistics. The chapter concludes with the conceptual framework for statistical modeling and gives various examples of simple models.

5.1 Introduction

In the previous chapters we considered random experiments that involved only a single random variable, such as the number of heads in 100 tosses, the number of left-handers in 50 people, or the amount of rain on the 2nd of January 2021 in Brisbane. This is obviously a simplification: in practice most random experiments involve multiple random variables. Here are some examples of experiments that we could do "tomorrow".

1. We randomly select $n = 10$ people and observe their heights. Let $X_1, \ldots, X_n$ be the individual heights.

2. We toss a coin repeatedly. Let $X_i = 1$ if the ith toss is Heads and $X_i = 0$ otherwise. The experiment is thus described by the sequence $X_1, X_2, \ldots$ of Bernoulli random variables.

3. We randomly select a person from a large population and measure his/her mass X and height Y.

4. We simulate 10,000 realizations from the standard normal distribution using the `rnorm` function. Let $X_1, \ldots, X_{10{,}000}$ be the corresponding random variables.

How can we specify the behavior of the random variables above? We should not just specify the pdf of the individual random variables, but also say something about the interaction (or lack thereof) between the random variables. For example, in the third experiment above if the height Y is large, then most likely the mass X is large as well. In contrast, in the first two experiments it is reasonable to assume that the random variables are "independent" in some way; that is, information about one of the random variables does not give extra information about the others. What we need to specify is the **joint distribution** of the random variables. The theory below for multiple random variables follows a similar path to that of a single random variable described in Section 3.6. ☞ 56

Let $X_1, \ldots, X_n$ be random variables describing some random experiment. Recall that the distribution of a *single* random variable X is completely specified by its cumulative distribution function. For *multiple* random variables we have the following generalization.

Definition 5.1: Joint Cumulative Distribution Function

The **joint cdf** of $X_1, \ldots, X_n$ is the function F defined by

$$F(x_1, \ldots, x_n) = \mathbb{P}(X_1 \leqslant x_1, \ldots, X_n \leqslant x_n)\ .$$

Notice that we have used the abbreviation $\mathbb{P}(\{X_1 \leqslant x_1\} \cap \cdots \cap \{X_n \leqslant x_n\}) = \mathbb{P}(X_1 \leqslant x_1, \ldots, X_n \leqslant x_n)$ to denote the probability of the intersection of events. We will use this abbreviation from now on.

As in the univariate (that is, single-variable) case we distinguish between *discrete* and *continuous* distributions.

5.2 Joint Distributions

■ **Example 5.1 (Dice Experiment)** In a box there are three dice. Die 1 is an ordinary die; die 2 has no 6 face, but instead two 5 faces; die 3 has no 5 face, but instead two 6 faces. The experiment consists of selecting a die at random followed by a toss with that die. Let X be the die number that is selected and let Y be the face value of that die. The probabilities $\mathbb{P}(X = x, Y = y)$ in Table 5.1 specify the joint distribution of X and Y. Note that it is more convenient to specify the joint probabilities $\mathbb{P}(X = x, Y = y)$ than the joint cumulative probabilities $\mathbb{P}(X \leqslant x, Y \leqslant y)$. The latter can be found, however, from the former by applying the sum rule. For example, $\mathbb{P}(X \leqslant 2, Y \leqslant 3) = \mathbb{P}(X = 1, Y = 1) + \cdots + \mathbb{P}(X = 2, Y = 3) = 6/18 = 1/3$. Moreover, by that same sum rule, the distribution of X is found by summing the $\mathbb{P}(X = x, Y = y)$ over all values of y

— giving the last column of Table 5.1. Similarly, the distribution of Y is given by the column totals in the last row of the table.

Table 5.1: The joint distribution of X (die number) and Y (face value).

		y						
		1	2	3	4	5	6	$\sum$
	1	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{3}$
x	2	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{9}$	0	$\frac{1}{3}$
	3	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$	0	$\frac{1}{9}$	$\frac{1}{3}$
	$\sum$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	1

■

In general, for discrete random variables $X_1, \dots, X_n$ the joint distribution is easiest to specify via the joint pmf.

Definition 5.2: Joint Probability Mass Function

The **joint pmf** f of discrete random variables $X_1, \dots, X_n$ is given by

$$f(x_1, \dots, x_n) = \mathbb{P}(X_1 = x_1, \dots, X_n = x_n) \ .$$

We sometimes write $f_{X_1,\dots,X_n}$ instead of f to show that this is the pmf of the random variables $X_1, \dots, X_n$. To save on notation, we can refer to the sequence $X_1, \dots, X_n$ simply as a random "vector" $\mathbf{X} = (X_1, \dots, X_n)$. If the joint pmf f is known, we can calculate the probability of any event via summation as

$$\mathbb{P}(\mathbf{X} \in B) = \sum_{\mathbf{x} \in B} f(\mathbf{x}) \ . \tag{5.1}$$

That is, to find the probability that the random vector lies in some set B (of dimension n), all we have to do is sum up all the probabilities $f(\mathbf{x})$ over all $\mathbf{x}$ in the set B. This is simply a consequence of the sum rule and a generalization of (3.6). In particular, as ☞ 58 illustrated in Example 5.1, we can find the pmf of X_i — often referred to as a **marginal** pmf, to distinguish it from the joint pmf — by summing the joint pmf over all possible values of the other variables. For example,

$$f_X(x) = \mathbb{P}(X = x) = \sum_y \mathbb{P}(X = x, Y = y) = \sum_y f_{X,Y}(x, y) \ . \tag{5.2}$$

The converse is not true: from the marginal distributions one cannot in general reconstruct the joint distribution. For example, in Example 5.1 we cannot reconstruct the inside of the two-dimensional table if only given the column and row totals.

For the continuous case we need to replace the joint pmf with the joint pdf.

Definition 5.3: Joint Probability Density Function

The **joint pdf** f of continuous random variables $X_1, \ldots, X_n$ (summarized as $\mathbf{X}$) is the positive function with total integral 1 such that

$$\mathbb{P}(\mathbf{X} \in B) = \int_{\mathbf{x} \in B} f(\mathbf{x})\,\mathrm{d}\mathbf{x} \quad \text{for all sets } B\,. \tag{5.3}$$

The integral in (5.3) is now a multiple integral — instead of evaluating the area under f, we now need to evaluate the (n-dimensional) volume. Figure 5.1 illustrates the concept for the 2-dimensional case.

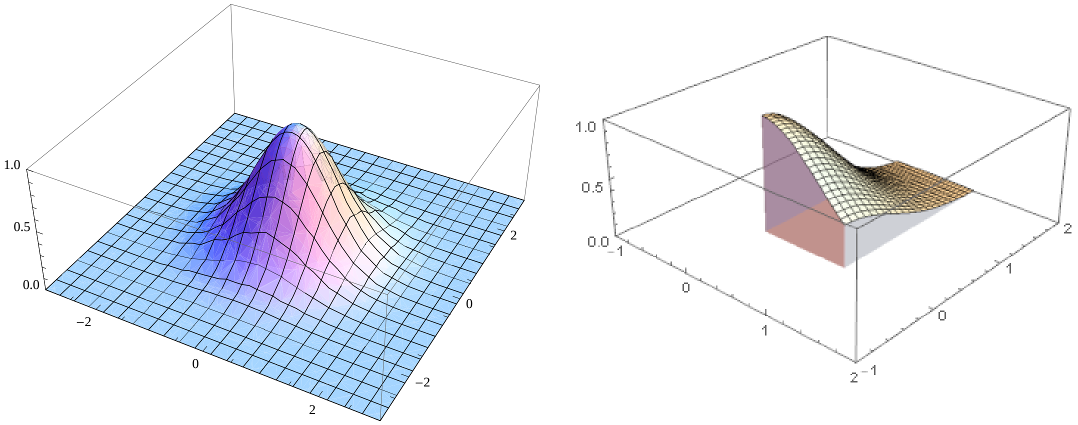


Figure 5.1: Left: a two-dimensional joint pdf of random variables X and Y. Right: the area under the pdf corresponds to $\mathbb{P}(0 \leqslant X \leqslant 1, Y \geqslant 0)$.

5.3 Independence of Random Variables

We have seen that in order to describe the behaviour of multiple random variables it is necessary to specify the joint distribution, not just the individual (that is, marginal) ones. However, there is one important exception, namely when the random variables are *independent*. We have so far only defined what independence is for *events* — see
☞ 55 (3.4). In the discrete case we define two random variables X and Y to be independent if the events $\{X = x\}$ and $\{Y = y\}$ are independent for every choice of x and y; that is,

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x)\,\mathbb{P}(Y = y)\,.$$

This means that any information about what the outcome of X is does not provide any extra information about Y. For the pmfs this means that the joint pmf $f(x, y)$ is equal to the product of the marginal ones $f_X(x) f_Y(y)$. We can take this as the definition for independence, also for the continuous case, and when more than two random variables are involved.

Definition 5.4: Independent Random Variables

Random variables $X_1, \ldots, X_n$ with joint pmf or pdf f are said to be **independent** if

$$f(x_1, \ldots, x_n) = f_{X_1}(x_1) \cdots f_{X_n}(x_n) \tag{5.4}$$

for all $x_1, \ldots, x_n$, where $\{f_{X_i}\}$ are the marginal pdfs.

■ **Example 5.2 (Dice Experiment Continued)** We repeat the experiment in Example 5.1 with three ordinary fair dice. Since the events $\{X = x\}$ and $\{Y = y\}$ are now independent, each entry in the pdf table is $\frac{1}{3} \times \frac{1}{6}$. Clearly in the first experiment not *all* events $\{X = x\}$ and $\{Y = y\}$ are independent. ■

Many statistical models involve random variables $X_1, X_2, \ldots$ that are **independent and identically distributed**, abbreviated as **iid**. We will use this abbreviation throughout this book and write the corresponding model as

$$X_1, X_2, \ldots \overset{\text{iid}}{\sim} \mathsf{Dist} \text{ (or } f \text{ or } F) \, ,$$

where Dist is the common distribution with pdf f and cdf F.

■ **Example 5.3 (Bivariate Standard Normal Distribution)** Suppose X and Y are independent and both have a standard normal distribution. We say that (X, Y) has a bivariate standard normal distribution. What is the joint pdf? We have

$$f(x, y) = f_X(x) f_Y(y) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}x^2} \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}y^2} = \frac{1}{2\pi} \mathrm{e}^{-\frac{1}{2}(x^2+y^2)} \, .$$

The graph of this joint pdf is the hat-shaped surface given in the left pane of Figure 5.1. We can also simulate independent copies $X_1, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(0, 1)$ and $Y_1, \ldots, Y_n \sim_{\text{iid}} \mathcal{N}(0, 1)$ and plot the pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ to gain insight into the joint distribution. The following lines of R code produce the scatter plot of simulated data in Figure 5.2.

```
> x = rnorm(2000)
> y = rnorm(2000)
> plot(y~x,xlim = c(-3,3), ylim= c(-3,3))
```

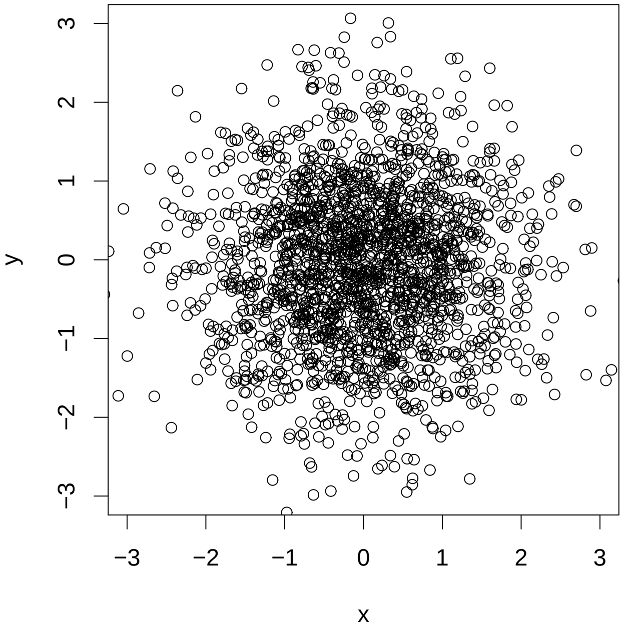


Figure 5.2: Scatter plot of 2000 points from the bivariate standard normal distribution.

We see a "spherical" pattern in the data. This is corroborated by the fact that the joint pdf has contour lines that are circles. ■

5.4 Expectations for Joint Distributions

☞ 61 Similar to the univariate case in Theorem 3.6, the expected value of a real-valued function h of $(X_1, \ldots, X_n) \sim f$ is a weighted average of all values that $h(X_1, \ldots, X_n)$ can take. Specifically, in the discrete case,

$$\mathbb{E}[h(X_1, \ldots, X_n)] = \sum_{x_1,\ldots,x_n} h(x_1, \ldots, x_n)\, f(x_1, \ldots, x_n), \tag{5.5}$$

where the sum is taken over all possible values of $(x_1, \ldots, x_n)$. In the continuous case replace the sum above with a (multiple) integral.

Two important special cases are the expectation of the *sum* (or more generally any linear transformation plus a constant) of random variables and the *product* of random variables.

Theorem 5.1: Properties of the Expectation

Let $X_1, \ldots, X_n$ be random variables with expectations $\mu_1, \ldots, \mu_n$. Then,

$$\mathbb{E}[a + b_1 X_1 + b_2 X_2 + \cdots + b_n X_n] = a + b_1 \mu_1 + \cdots + b_n \mu_n \tag{5.6}$$

for all constants $a, b_1, \ldots, b_n$. Also, for *independent* random variables,

$$\mathbb{E}[X_1 X_2 \cdots X_n] = \mu_1\, \mu_2 \cdots \mu_n\ . \tag{5.7}$$

Proof: We show it for the discrete case with two variables only. The general case follows by analogy and, for the continuous case, by replacing sums with integrals. Let X_1 and X_2 be discrete random variables with joint pmf f. Then, by (5.5),

$$\begin{aligned}\mathbb{E}[a + b_1 X_1 + b_2 X_2] &= \sum_{x_1, x_2} (a + b_1 x_1 + b_2 x_2)\, f(x_1, x_2) \\ &= a + b_1 \sum_{x_1} \sum_{x_2} x_1 f(x_1, x_2) + b_2 \sum_{x_1} \sum_{x_2} x_2 f(x_1, x_2) \\ &= a + b_1 \sum_{x_1} x_1 \left(\sum_{x_2} f(x_1, x_2) \right) + b_2 \sum_{x_2} x_2 \left(\sum_{x_1} f(x_1, x_2) \right) \\ &= a + b_1 \sum_{x_1} x_1 f_{X_1}(x_1) + b_2 \sum_{x_2} x_2 f_{X_2}(x_2) = a + b_1 \mu_1 + b_2 \mu_2\ .\end{aligned}$$

Next, assume that X_1 and X_2 are independent, so that $f(x_1, x_2) = f_{X_1}(x_1)\, f_{X_2}(x_2)$. Then,

$$\begin{aligned}\mathbb{E}[X_1\, X_2] &= \sum_{x_1, x_2} x_1\, x_2\, f_{X_1}(x_1) f_{X_2}(x_2) \\ &= \sum_{x_1} x_1 f_{X_1}(x_1)\ \times \sum_{x_2} x_2 f_{X_2}(x_2) = \mu_1\, \mu_2\ .\end{aligned}$$

□

Definition 5.5: Covariance

The **covariance** of two random variables X and Y with expectations $\mathbb{E}X = \mu_X$ and $\mathbb{E}Y = \mu_Y$ is defined as

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]\ .$$

The covariance is a measure of the amount of linear dependency between two random variables. A scaled version of the covariance is given by the **correlation coefficient**:

$$\varrho(X, Y) = \frac{\mathrm{Cov}(X, Y)}{\sigma_X\, \sigma_Y}\ , \tag{5.8}$$

where $\sigma_X^2 = \mathrm{Var}(X)$ and $\sigma_Y^2 = \mathrm{Var}(Y)$. For easy reference, Theorem 5.2 lists some important properties of the variance and covariance.

Theorem 5.2: Properties of the Variance and Covariance

For random variables X, Y and Z, and constants a and b, we have

1. $\mathrm{Var}(X) = \mathbb{E}(X^2) - (\mathbb{E}(X))^2$.
2. $\mathrm{Var}(a + bX) = b^2\mathrm{Var}(X)$.
3. $\mathrm{Cov}(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X)\,\mathbb{E}(Y)$.
4. $\mathrm{Cov}(X, Y) = \mathrm{Cov}(Y, X)$.
5. $\mathrm{Cov}(aX + bY, Z) = a\,\mathrm{Cov}(X, Z) + b\,\mathrm{Cov}(Y, Z)$.
6. $\mathrm{Cov}(X, X) = \mathrm{Var}(X)$.
7. $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X, Y)$.
8. If X and Y are independent, then $\mathrm{Cov}(X, Y) = 0$.

Proof: For simplicity of notation we write $\mathbb{E}Z = \mu_Z$ for a generic random variable Z.
☞ 63 Properties 1 and 2 were already shown in Theorem 3.7.

3. $\mathrm{Cov}(X, Y) = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)] = \mathbb{E}[X\,Y - X\,\mu_Y - Y\,\mu_X + \mu_X\,\mu_Y] = \mathbb{E}[X\,Y] - \mu_X\,\mu_Y$.
4. $\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[(Y - \mu_Y)(X - \mu_X)] = \mathrm{Cov}(Y, X)$.
5. $\mathrm{Cov}(aX + bY, Z) = \mathbb{E}[(aX + bY)Z] - \mathbb{E}[aX + bY]\,\mathbb{E}(Z) = a\,\mathbb{E}[XZ] - a\,\mathbb{E}(X)\mathbb{E}(Z) + b\,\mathbb{E}[YZ] - b\,\mathbb{E}(Y)\mathbb{E}(Z) = a\,\mathrm{Cov}(X, Z) + b\,\mathrm{Cov}(Y, Z)$.
6. $\mathrm{Cov}(X, X) = \mathbb{E}[(X - \mu_X)(X - \mu_X)] = \mathbb{E}[(X - \mu_X)^2] = \mathrm{Var}(X)$.
7. By Property 6, $\mathrm{Var}(X+Y) = \mathrm{Cov}(X+Y, X+Y)$. By Property 5, $\mathrm{Cov}(X+Y, X+Y) = \mathrm{Cov}(X, X) + \mathrm{Cov}(Y, Y) + \mathrm{Cov}(X, Y) + \mathrm{Cov}(Y, X) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\,\mathrm{Cov}(X, Y)$, where in the last equation Properties 4 and 6 are used.
8. If X and Y are independent, then $\mathbb{E}[X\,Y] = \mu_X\,\mu_Y$. Therefore, $\mathrm{Cov}(X, Y) = 0$ follows immediately from Property 3.

□

In particular, combining Properties (7) and (8) we see that if X and Y are independent, then the variance of their sum is equal to the sum of their variances. It is not difficult to deduce from this the following more general result.

Theorem 5.3: Variance for Linear Combinations of Random Variables

Let $X_1, \dots, X_n$ be independent random variables with expectations $\mu_1, \dots, \mu_n$ and variances $\sigma_1^2, \dots, \sigma_n^2$. Then,

$$\mathrm{Var}(a + b_1 X_1 + b_2 X_2 + \cdots + b_n X_n) = b_1^2 \sigma_1^2 + \cdots + b_n^2 \sigma_n^2 \tag{5.9}$$

for all constants $a, b_1, \dots, b_n$.

☞ 69

■ **Example 5.4 (Expectation and Variance for the Binomial Distribution)** We now show a simple way to prove Theorem 4.2; that is, to prove that the expectation and variance for the $\mathsf{Bin}(n, p)$ distribution are np and $np(1-p)$, respectively. Let $X \sim \mathsf{Bin}(n, p)$. Hence, we can view X as the total number of successes in n Bernoulli trials (coin flips) with success probability p. Let us introduce Bernoulli random variables $X_1, \dots, X_n$, where $X_i = 1$ is the ith trial is a success (and $X_i = 0$ otherwise). We thus have that $X_1, \dots, X_n \sim_{\text{iid}} \mathsf{Ber}(p)$. The key to the proof is to observe that X is simply the sum of the $X_i' s$; that is

$$X = X_1 + \cdots + X_n \,.$$

Since we have seen that each Bernoulli variable has expectation p and variance $p(1-p)$, we have by Theorem 5.1 that

$$\mathbb{E}(X) = \mathbb{E}(X_1) + \cdots + \mathbb{E}(X_n) = np$$

and by Theorem 5.3 that

$$\mathrm{Var}(X) = \mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_n) = np(1-p) \,,$$

as had to be shown. ■

5.5 Limit Theorems

Two main results in probability are the *law of large numbers* and *the central limit theorem*. Both are limit theorems involving sums of independent random variables. In particular, consider a sequence $X_1, X_2, \dots$ of iid random variables with finite expectation μ and finite variance σ^2. For each n define the sum $S_n = X_1 + \cdots + X_n$. What can we say about the (random) sequence of sums $S_1, S_2, \dots$ or averages $S_1, S_2/2, S_3/3, \dots$? By (5.6) and (5.9) we have $\mathbb{E}(S_n/n) = \mu$ and $\mathrm{Var}(S_n/n) = \sigma^2/n$. Hence, as n increases the variance of the (random) average S_n/n goes to 0. Informally, it means the following.

Theorem 5.4: Law of Large Numbers

The average of a large number of iid random variables tends to their expectation as the sample size goes to infinity.

This is a nice property: if we wish to say something about the expectation of a random variable, we can simulate many independent copies and then take the average of these, to get a good approximation to the (perhaps unknown) expectation. The approximation will get better and better when the sample size gets larger.

■ **Example 5.5 (Square Root of a Uniform)** Let $U \sim \mathcal{U}(0,1)$. What is the expectation of $\sqrt{U}$? We know that the expectation of U is $1/2$. Would the expectation of $\sqrt{U}$ be $\sqrt{1/2}$? We can determine in this case the expectation exactly, but let us use simulation and the law of large numbers instead. All we have to do is simulate a large number of uniform numbers, take their square roots, and average over all values:

```
> u = runif(10e6)
> x = sqrt(u)
> mean(x)
```

```
[1] 0.6665185
```

Repeating the simulation gives consistently 0.666 in the first three digits behind the decimal point. You can check that the true expectation is 2/3, which is smaller than $\sqrt{1/2} \approx 0.7071$. ■

The central limit theorem describes the approximate distribution of S_n (or S_n/n), and it applies to both continuous and discrete random variables. Informally, it states the following.

Theorem 5.5: Central Limit Theorem

The sum of a large number of iid random variables approximately has a normal distribution.

Specifically, the random variable S_n has a distribution that is approximately normal, with expectation $n\mu$ and variance $n\sigma^2$. This is a truly remarkable result and is one of the great milestones in mathematics. We will not have enough background to prove it, but we can demonstrate it very nicely using simulation.

☞ 69 Let X_1 be a $\mathcal{U}[0,1]$ random variable. Its pdf (see Section 4.4) is constant on the interval [0,1] and 0 elsewhere. If we simulate many independent copies of X_1 and take a histogram, the result will resemble the shape of the pdf (this, by the way, is a consequence of the law of large numbers). What about the pdf of $S_2 = X_1 + X_2$? We can generate many copies of both X_1 and X_2, add them up, and then make a histogram. Here is how you could do it in R and the result is given in Figure 5.3.

```
> x1 = runif(10e6)
> x2 = runif(10e6)
> hist(x1 +x2,breaks=100,prob=T)
```

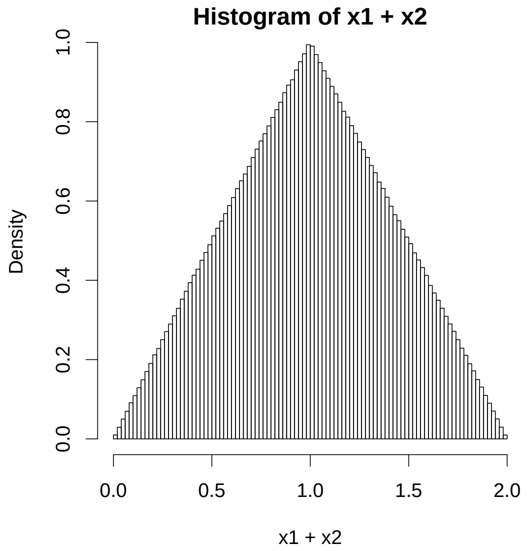


Figure 5.3: Histogram for the sum of 2 independent uniform random variables.

The pdf seems to be triangle shaped and, indeed, this is not so difficult to show. Now let us do the same thing for sums of 3 and 4 uniform numbers. Figure 5.4 shows that the pdfs have assumed a bellshaped form reminiscent of the normal distribution. Indeed, if we superimpose the normal distribution with the same mean and variance as the sums, the agreement is excellent.

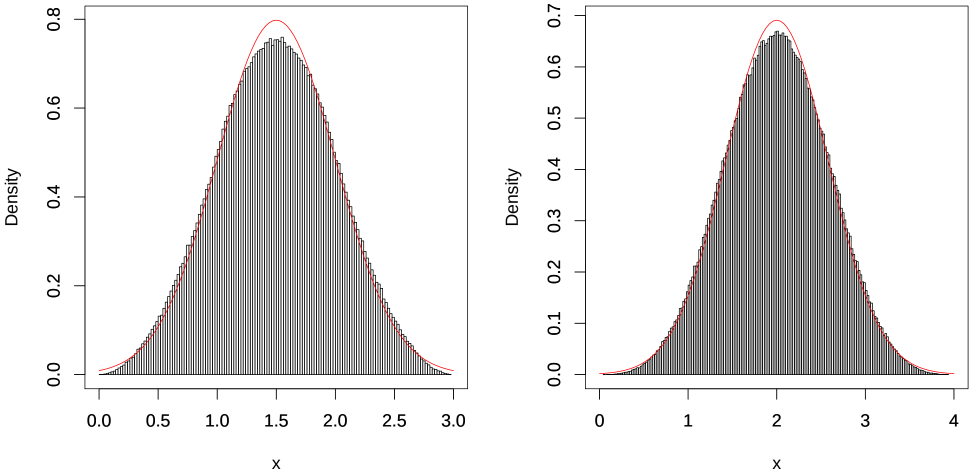


Figure 5.4: The histograms for the sums of 3 (left) and 4 (right) uniforms are in close agreement with normal pdfs.

The central limit theorem does not only hold if we add up continuous random variables, such as uniform ones, but it also holds for the discrete case. In particular, recall that a binomial random variable $X \sim \mathsf{Bin}(n, p)$ can be viewed as the sum of n iid

$\mathsf{Ber}(p)$ random variables: $X = X_1 + \cdots + X_n$. As a direct consequence of the central limit theorem it follows that for large n, $\mathbb{P}(X \leqslant k) \approx \mathbb{P}(Y \leqslant k)$, where $Y \sim \mathcal{N}(np, np(1-p))$. As a rule of thumb, this normal approximation to the binomial distribution is accurate if both np and $n(1-p)$ are larger than 5.

Finally, when we add up independent *normal* random variables, then the resulting random variable has again a normal distribution. In fact any linear combination of independent normal random variables, such as $b_1X_1 + b_2X_2 + \cdots + b_nX_n$ can be shown to have again a normal distribution. This is quite an exceptional property, which makes the standard normal distribution stand out from most other distributions. The proof is outside the scope of a first-year course, but the central limit result should give you some confidence that it is true. And you can verify particular cases yourself via simulation. Thus, the following theorem is one of the main reasons why the normal distribution is used so often in statistics.

Theorem 5.6: Linear Combinations of Normals are Again Normal

Let $X_1, X_2, \ldots, X_n$ be independent normal random variables with expectations $\mu_1, \ldots, \mu_n$ and variances $\sigma_1^2, \ldots, \sigma_n^2$. Then, for any numbers $a, b_1, \ldots, b_n$ the random variable

$$Y = a + b_1X_1 + b_2X_2 + \cdots + b_nX_n$$

has a normal distribution with expectation $a + \sum_{i=1}^n b_i\,\mu_i$ and variance $\sum_{i=1}^n b_i^2\,\sigma_i^2$.

Note that the expectation and variance of Y are a direct consequence of Theorems 5.1 and 5.3.

5.6 Statistical Modeling

Let us now return right to the beginning of these notes, to the steps for a statistical study
☞ 9 in Section 1.1. Figure 5.5 gives a sketch of the conceptual framework for statistical modeling and analysis. *Statistical modeling* refers to finding a plausible probabilistic model for the data. This model contains what we know about the reality and how the data were obtained. Once we have formulated the model, we can carry out our calculations and analysis and make conclusions.

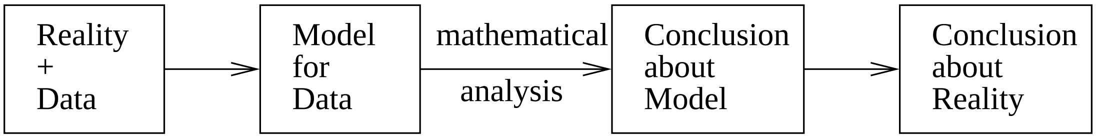


Figure 5.5: Statistical modeling and analysis.

The simplest class of statistical models is the one where the data $X_1, \ldots, X_n$ are as-
☞ 83 sumed to be independent and identically distributed (iid), as we already mentioned. In

many cases it is assumed that the sampling distribution is normal. Here is an example.

■ **Example 5.6 (One-sample Normal Model)** From a large population we select 300 men between 40 and 50 years of age and measure their heights. Let X_i be the height of the i-th selected person, $i = 1, \dots, 300$. As a model take,

$$X_1, \dots, X_{300} \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$$

for some unknown parameters μ and σ^2. We could interpret these as the population mean and variance. ■

A simple generalization of a single sample of iid data is the model where there are two independent samples of iid data, as in the examples below.

■ **Example 5.7 (Two-sample Binomial Model)** To assess whether there is a difference between boys and girls in their preference for two brands of cola, say *Sweet* and *Ultra* cola, we select at random 100 boys and 100 girls and ask whether they prefer *Sweet* or *Ultra*. We could model this via two independent Bernoulli samples. That is, for each $i = 1, \dots, 100$ let $X_i = 1$ if the i-th boy prefers *Sweet* and let $X_i = 0$ otherwise. Similarly, let $Y_i = 1$ if the i-th girl prefers *Sweet* over *Ultra*. We thus have the model

$$\begin{aligned} X_1, \dots, X_{100} &\overset{\text{iid}}{\sim} \mathsf{Ber}(p_1)\,, \\ Y_1, \dots, Y_{100} &\overset{\text{iid}}{\sim} \mathsf{Ber}(p_2)\,, \\ X_1, \dots, X_{100}, Y_1, \dots, Y_{100} &\;\text{ independent, with } p_1 \text{ and } p_2 \text{ unknown.} \end{aligned}$$

The objective is to assess the difference $p_1 - p_2$ on the basis of the observed values for $X_1, \dots, X_{100}, Y_1, \dots, Y_{100}$. Note that it suffices to only record the total number of boys or girls who prefer *Sweet* cola in each group; that is, $X = \sum_{i=1}^{100} X_i$ and $Y = \sum_{i=1}^{100} Y_i$.

This gives the **two-sample binomial model**:

$$\begin{aligned} X &\sim \mathsf{Bin}(100, p_1)\,, \\ Y &\sim \mathsf{Bin}(100, p_2)\,, \\ X, Y &\;\text{ independent, with } p_1 \text{ and } p_2 \text{ unknown.} \end{aligned}$$

■

■ **Example 5.8 (Two-sample Normal Model)** From a large population we select 200 men between 25 and 30 years of age and measure their heights. For each person we also record whether the mother smoked during pregnancy or not. Suppose that 60 mothers smoked during pregnancy.

Let $X_1, \dots, X_{60}$ be the heights of the men whose mothers smoked, and let $Y_1, \dots, Y_{140}$ be the heights of the men whose mothers did not smoke. Then, a possible model is the

two-sample normal model:

$$\begin{aligned} X_1,\dots,X_{60} &\overset{\text{iid}}{\sim} \mathcal{N}(\mu_1,\sigma_1^2)\ , \\ Y_1,\dots,Y_{140} &\overset{\text{iid}}{\sim} \mathcal{N}(\mu_2,\sigma_2^2)\ , \\ X_1,\dots,X_{60},Y_1,\dots,Y_{140} &\quad \text{independent,} \end{aligned}$$

where the model parameters μ_1, μ_2, σ_1^2, and σ_2^2 are unknown. One would typically like to assess the difference $\mu_1 - \mu_2$. That is, does smoking during pregnancy affect the (expected) height of the sons? A typical simulation outcome of the model is given in Figure 5.6, using parameters $\mu_1 = 170, \mu_2 = 175, \sigma_1^2 = 200$, and $\sigma_2^2 = 100$.

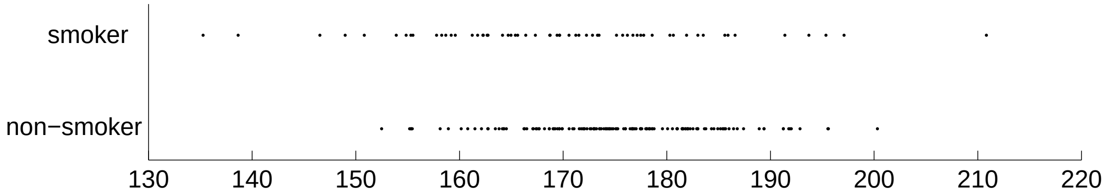


Figure 5.6: Simulated height data from a two-sample normal model.

■

Remark 5.1 (About Statistical Modeling) At this point it is good to emphasize a few points about statistical modeling.

- *Any* model for data is likely to be *wrong*. For example, in Example 5.8 the height would normally be recorded on a discrete scale, say 1000 – 2200 (mm). However, samples from a $\mathcal{N}(\mu, \sigma^2)$ can take any real value, including negative values! Nevertheless, the normal distribution could be a reasonable approximation to the real sampling distribution. An important advantage of using a normal distribution is that it has many nice mathematical properties as we have seen.

- Most statistical models depend on a number of *unknown* parameters. One of the main objectives of *statistical inference* — to be discussed in subsequent chapters — is to gain knowledge of the unknown parameters on the basis of the observed data.

- Any model for data needs to be checked for suitability. An important criterion is that data simulated from the model should resemble the observed data — at least for a certain choice of model parameters.

CHAPTER 6

Estimation

In this chapter you will learn how to estimate parameters of simple statistical models from the observed data. The difference between estimate and estimator will be explained. Confidence intervals will be introduced to assess the accuracy of an estimate. We will derive confidence intervals for a variety one- and two-sample models. Various probability distributions, such as the Student's t and the χ^2 distribution will make their first appearance.

6.1 Introduction

☞ 90

Recall the framework of statistical modeling in Figure 5.5. We are given some data (measurements) for which we construct a *model* that depends on one or more parameters. Based on the observed data we try to say something about the model parameters. For example, we wish to *estimate* the parameters. Here are some concrete examples.

■ **Example 6.1 (Biased Coin)** We throw a coin 1000 times and observe 570 Heads. Using this information, what can we say about the "fairness" of the coin? The data here (or better, *datum*, as there is only one observation) is the number $x = 570$. Suppose we view x as the outcome of a random variable X which describes the number of Heads in 1000 tosses. Our statistical model is then:

$$X \sim \mathsf{Bin}(1000, p) ,$$

where $p \in [0, 1]$ is unknown. Any statement about the fairness of the coin is expressed in terms of p and is assessed via this model. It is important to understand that p will *never be known*. The best we can do is to provide an *estimate* of p. A common sense estimate of p is simply the proportion of Heads $x/1000 = 0.570$. But how accurate is this estimate? Is it possible that the unknown p could in fact be 0.5? One can make sense of these questions through detailed analysis of the statistical model. ■

■ **Example 6.2 (Iid Sample from a Normal Distribution)** Consider the standard model for data

$$X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2) ,$$

where μ and σ^2 are unknown. The random measurements $\{X_i\}$ could represent the masses of randomly selected teenagers, the heights of the dorsal fin of sharks, the dioxin concentrations in hamburgers, and so on. Suppose, for example that, with $n = 10$, the observed measurements $x_1, \dots, x_n$ are:

77.01, 71.37, 77.15, 79.89, 76.46, 78.10, 77.18, 74.08, 75.88, 72.63.

A common-sense *estimate* (a number) for μ is the **sample mean**

$$\bar{x} = \frac{x_1 + \cdots + x_n}{n} = 75.975 , \tag{6.1}$$

and σ^2 can be estimated via the **sample variance**

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 . \tag{6.2}$$

Note that the estimates $\bar{x}$ and s^2 are functions of the data $\mathbf{x} = (x_1, \dots, x_n)$ only. We
☞ 28 encountered these summary statistics already in Section 2.4.

Why are these numbers good estimates (guesses) for our unknown parameters μ and σ^2. How accurate are these numbers? That is, how far away are they from the true parameters? To answer these questions we need to investigate the statistical properties of the sample mean and sample variance. ■

It is customary in statistics to denote the estimate of a parameter $\boldsymbol{\theta}$ by $\widehat{\boldsymbol{\theta}}$; for example, $\widehat{\mu} = \bar{x}$ in the example above.

6.2 Estimates and Estimators

If we have some data coming from some statistical model, how do we estimate the parameters? There are various systematic ways to construct sensible estimates for parameters of various models. Suppose we have n independent copies $X_1, \dots, X_n$ of a random variable X whose distribution depends on p parameters (for example, $X \sim \mathcal{N}(\mu, \sigma^2)$, with $p = 2$ parameters). A useful general approach to estimate the parameters is the **method of moments**. Recall that the ***k*-th moment** of a random variable X is defined
☞ 62 as $\mathbb{E}(X^k)$; see Definition 3.11. For example, the expectation is the first moment. In the method of moments the estimated parameters are chosen such that the first p true moments $\mathbb{E}(X^k)$ are matched to their sample averages $\sum_{i=1}^{n} x_i^k / n$.

■ **Example 6.3 (Method of Moments)** Let $X_1, \dots, X_n$ be iid copies of $X \sim \mathcal{N}(\mu, \sigma^2)$. The first moment of each X is $\mathbb{E}(X) = \mu$, and the second moment of X is $\mathbb{E}(X^2) = \mathrm{Var}(X) + [\mathbb{E}(X)]^2 = \sigma^2 + \mu^2$. To find the method of moments estimates for μ and σ^2, let us call them $\widehat{\mu}$ and $\widehat{\sigma^2}$, we need to match the first two moments to their sample averages. That is, we need to solve

$$\begin{aligned} \widehat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \widehat{\mu}^2 + \widehat{\sigma^2} &= \frac{1}{n}\sum_{i=1}^{n} x_i^2 \, . \end{aligned}$$

The first equation gives the sample mean $\widehat{\mu} = \bar{x}$ as our estimate for μ. Substituting $\widehat{\mu} = \bar{x}$ in the second equation, we find that the second equation gives

$$\widehat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^{\,2} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^{\,2}\right) \qquad (6.3)$$

as an estimate for σ^2. This estimate seems quite different from the sample variance s^2 in (6.2). But the two estimates are actually very similar. To see this, expand the quadratic term in (6.2), to get

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i^2 - 2x_i\bar{x} + \bar{x}^{\,2}) \, .$$

Now break up the sum:

$$\begin{aligned} s^2 &= \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} 2\bar{x}x_i + \sum_{i=1}^{n} \bar{x}^{\,2}\right) \\ &= \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - 2\bar{x}\sum_{i=1}^{n} x_i + \bar{x}^{\,2}\sum_{i=1}^{n} 1\right) \end{aligned}$$

and simplify

$$\begin{aligned} s^2 &= \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - 2\bar{x}n\bar{x} + n\bar{x}^{\,2}\right) \\ &= \frac{1}{n-1}\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^{\,2}\right). \end{aligned}$$

Comparing this with (6.3), we see that $s^2 = n/(n-1)\,\widehat{\sigma^2}$, so they differ only in a factor $n/(n-1)$. For large n they are practically the same. ■

To find out how *good* an estimate is, we need to investigate the properties of the corresponding **estimator**. The estimator is obtained by replacing the fixed observations x_i with the random variables X_i in the expression for the estimate. For example, the estimator corresponding to the sample mean $\bar{x}$ is

$$\bar{X} = \frac{X_1 + \cdots + X_n}{n} \,.$$

The interpretation is that $X_1, \ldots, X_n$ are the data that we will obtain if we carry out the experiment *tomorrow*, and $\bar{X}$ is the (random) sample mean of these data, which again will be obtained tomorrow.

Let us go back to the basic model were $X_1, \ldots, X_n$ are independent and identically distributed with some unknown expectation μ and variance σ^2. We do not require that the $\{X_i\}$ are normally distributed — we are only interested in estimating the expectation and variance. To justify why $\bar{x}$ is a good estimate of μ, think about what we can say (today) about the properties of the estimator $\bar{X}$. The expectation and variance of $\bar{X}$ follow easily from the rules for expectation and variance in Chapter 5. In particular,
☞ 85 by (5.6) we have

$$\begin{aligned}\mathbb{E}(\bar{X}) &= \mathbb{E}\left(\frac{1}{n}(X_1 + \cdots + X_n)\right) = \frac{1}{n}\mathbb{E}\,(X_1 + \cdots + X_n) = \frac{1}{n}(\mathbb{E}(X_1) + \cdots + \mathbb{E}(X_n)) \\ &= \frac{1}{n}(\mu + \cdots + \mu) = \mu\end{aligned}$$

☞ 87 and from (5.9) we have

$$\begin{aligned}\mathrm{Var}(\bar{X}) &= \mathrm{Var}\left(\frac{1}{n}(X_1 + \cdots + X_n)\right) = \frac{1}{n^2}\mathrm{Var}(X_1 + \cdots + X_n) = \frac{1}{n^2}\,(\mathrm{Var}(X_1) + \cdots + \mathrm{Var}(X_n)) \\ &= \frac{1}{n^2}(\sigma^2 + \cdots + \sigma^2) = \frac{\sigma^2}{n} \,.\end{aligned}$$

The first result says that the estimator $\bar{X}$ is "on average" equal to the unknown quantity that we wish to estimate (μ). We call an estimator whose expectation is equal to the quantity that we wish to estimate **unbiased**. The second result shows that the larger we take n, the closer the variance of $\bar{X}$ is to zero, indicating that $\bar{X}$ goes to the constant
☞ 87 μ for large n. This is in essence the law of large numbers; see Section 5.5.

To assess how close $\bar{X}$ is to μ, one needs to look at a confidence interval for μ.

6.3 Confidence Intervals

An essential part in any estimation procedure is to provide an assessment of the *accuracy* of the estimate. Indeed, without information on its accuracy the estimate itself would be meaningless. Confidence intervals (sometimes called **interval estimates**) provide a precise way of describing the uncertainty in the estimate.

Definition 6.1: Confidence Interval

Let $X_1, \dots, X_n$ be random variables with a joint distribution depending on a parameter θ. Let $T_1 < T_2$ be functions of the data $X_1, \dots, X_n$ but not of θ. A random interval (T_1, T_2) is called a **stochastic confidence interval** for θ with confidence $1 - \alpha$ if

$$\mathbb{P}(T_1 < \theta < T_2) \geqslant 1 - \alpha \quad \text{for all } \theta . \tag{6.4}$$

If t_1 and t_2 are the observed values of T_1 and T_2, then the interval (t_1, t_2) is called the **numerical confidence interval** for θ with confidence $1 - \alpha$. If (6.4) only holds approximately, the interval is called an **approximate confidence interval**.

The actual *meaning* of a confidence interval is quite tricky. Suppose we find a 90% numerical confidence interval (9.5,10.5) for θ. Does this mean that $\mathbb{P}(9.5 < \theta < 10.5) = 0.9$? No! Since θ is a fixed number the probability $\mathbb{P}(9.5 < \theta < 10.5)$ is either 0 or 1, and we don't know which one, because we don't know θ. To find the meaning we have to go back to the definition of a confidence interval. There we see that the interval (9.5,10.5) is an *outcome* of a *stochastic* (i.e., random) confidence interval (T_1, T_2), such that $\mathbb{P}(T_1 < \theta < T_2) = 0.9$. Note that θ is constant, but the interval bounds T_1 and T_2 are random. If we would repeat this experiment many times, then we would get many numerical confidence intervals, as illustrated in Figure 6.1

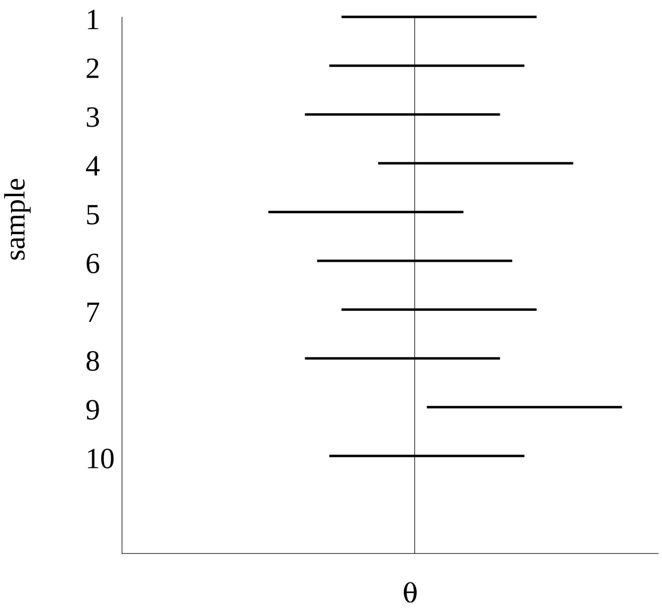


Figure 6.1: Possible outcomes of stochastic confidence intervals.

Only in (on average) 9 out of 10 cases would these intervals contain our unknown θ. To put it in another way: Consider an urn with 90 white and 10 black balls. We pick at random a ball from the urn *but we do not open our hand to see what colour ball we have*. Then we are pretty confident that the ball we have in our hand is white. This is how confident you should be that the unknown θ lies in the interval $(9.5, 10.5)$.

> Reducing α widens the confidence interval. A very large confidence interval is not very useful. Common choices for α are $0.01, 0.05$, and 0.1.

6.3.1 Approximate Confidence Interval for the Mean

Let $X_1, X_2, \ldots, X_n$ be an iid sample from a distribution with mean μ and variance $\sigma^2 < \infty$ (both assumed to be unknown). We assume that the sample size n is large. ☞ 88 By the central limit theorem, we know then that $X_1 + \cdots + X_n$ has approximately a normal distribution, so $\bar{X}$ also has approximately a normal distribution. We found the corresponding expectation and variance above, so

$$\bar{X} \overset{\text{approx.}}{\sim} \mathcal{N}(\mu, \sigma^2/n) \ .$$

Standardising $\bar{X}$ gives

$$\frac{\bar{X} - \mu}{\sqrt{\sigma^2/n}} \overset{\text{approx.}}{\sim} \mathcal{N}(0, 1) \ .$$

In order to construct a confidence interval for μ, we would like to create a so-called **pivot** variable that (1) depends on all the data and on the parameter to be estimated and (2) has a distribution that does not depend on any unknown parameters. The above standardized form of $\bar{X}$ is not a pivot yet because it depends on σ^2. However, we can fix this by replacing σ^2 with its unbiased estimator S^2. By the law of large numbers S^2 looks more and more like the constant σ^2 as n grows larger. So, we have for large n

$$T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \overset{\text{approx.}}{\sim} \mathcal{N}(0, 1) \ , \tag{6.5}$$

where $S = \sqrt{S^2}$ is the sample standard deviation. Because T is approximately standard normal, we have, for example,

$$\mathbb{P}(T \leqslant 1.645) \approx 0.95 \quad \text{and} \quad \mathbb{P}(T \leqslant 1.96) \approx 0.975$$

because 1.645 is the 0.95 quantile of the normal distribution and 1.96 the 0.975 quantile, ☞ 70 both of which are good to remember; see also Section 4.5. Because the standard normal distribution is symmetrical around 0, we also have, for example,

$$\mathbb{P}(-1.96 < T < 1.96) \approx 0.95 \ .$$

Now, let us have a closer look at this, and plug back in the expression for the pivot T, so

$$\mathbb{P}\left(-1.96 < \frac{\bar{X} - \mu}{S/\sqrt{n}} < 1.96\right) \approx 0.95 \ .$$

We can rearrange the event

$$A = \left\{ -1.96 < \frac{\bar{X} - \mu}{S/\sqrt{n}} < 1.96) \right\}$$

as follows. Multiplying the left, middle, and right parts of the inequalities by $S/\sqrt{n}$ still gives the same event, so

$$A = \left\{ -1.96\frac{S}{\sqrt{n}} < \bar{X} - \mu < 1.96\frac{S}{\sqrt{n}} \right\}.$$

Subtracting $\bar{X}$ from left, middle, and right parts still does not change anything about the event, so

$$A = \left\{ -\bar{X} - 1.96\frac{S}{\sqrt{n}} < -\mu < -\bar{X} + 1.96\frac{S}{\sqrt{n}} \right\}.$$

Finally we multiply the left, middle, and right parts with -1. This will flip the $<$ signs to $>$. For example, $-3 < -2$ is the same as $3 > 2$. So, we get:

$$A = \left\{ \bar{X} + 1.96\frac{S}{\sqrt{n}} > \mu > \bar{X} - 1.96\frac{S}{\sqrt{n}} \right\},$$

which is the same as

$$A = \left\{ \bar{X} - 1.96\frac{S}{\sqrt{n}} < \mu < \bar{X} + 1.96\frac{S}{\sqrt{n}} \right\} .$$

If we write this as $A = \{T_1 < \mu < T_2\}$, with $\mathbb{P}(A) \approx 0.95$, then we see that (T_1, T_2) is an approximate 95% confidence interval for μ. We can repeat this procedure with any quantile of the normal distribution. This leads to the following result.

Theorem 6.1: Approximate Confidence Interval for μ

Let $X_1, X_2, \ldots, X_n$ be an iid sample from a distribution with mean μ and variance $\sigma^2 < \infty$. Let q be the $1 - \alpha/2$ quantile of the standard normal distribution. An approximate stochastic confidence interval for μ is

$$\left(\bar{X} - q\frac{S}{\sqrt{n}},\ \bar{X} + q\frac{S}{\sqrt{n}} \right), \text{ abbreviated as } \bar{X} \pm q\frac{S}{\sqrt{n}} . \tag{6.6}$$

The quantity $qS/\sqrt{n}$ is called the **margin of error** for the confidence interval.

Since (6.6) is an asymptotic result only, care should be taken when applying it to cases where the sample size is small or moderate and the sampling distribution is heavily skewed.

■ **Example 6.4 (Oil Company)** An oil company wishes to investigate how much on average each household in Melbourne spends on petrol and heating oil per year. The company randomly selects 51 households from Melbourne, and finds that these spent on average \$1136 on petrol and heating oil, with a sample standard deviation of \$178. We wish to construct a 95% confidence interval for the expected amount of money per year that the households in Melbourne spend on petrol and heating oil. Call this parameter μ.

We assume that the outcomes of the survey, $x_1, \ldots, x_{51}$, are realizations of an iid sample with expectation μ. Although we do not know the outcomes themselves, we know their sample mean $\bar{x} = 1136$ and standard deviation $s = 178$. An approximate numerical 95% confidence interval is thus

$$1136 \pm 1.96 \frac{178}{\sqrt{51}} = (1087, 1185) \, .$$

■

6.3.2 Normal Data, One Sample

For an iid sample from the normal distribution, $X_1, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu, \sigma^2)$, it is possible to construct *exact* confidence intervals for μ and σ^2, rather than only approximate ones.

Confidence Interval for μ

For iid $\mathcal{N}(\mu, \sigma^2)$ data, the pivot variable T in (6.5) can be shown to have a **Student's *t*-distribution**. This distribution is named after its discoverer W.S. Gosset, who published under the pseudonym "Student". The t-distribution is actually a family of distributions, depending on a single parameter called the (number of) **degrees of freedom**. We write $Z \sim \mathrm{t}_{\mathrm{df}}$ to indicate that a random variable Z has a student distribution with df degrees of freedom. Here is the exact version of the approximate result (6.5).

Theorem 6.2: Standardized Mean and Student *t*-distribution

Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$. Then

$$\frac{\bar{X} - \mu}{S / \sqrt{n}} \sim \mathrm{t}_{n-1} \, . \tag{6.7}$$

Figure 6.2 gives graphs of the probability densities functions for the t_1, t_2, t_5, and t_{50}. Notice a similar bell-shaped curve as for the normal distribution, but the tails of the distribution are "fatter" than for the normal distribution. As n grows larger the pdf of the t_n gets closer and closer to the pdf of the $\mathcal{N}(0, 1)$ distribution.

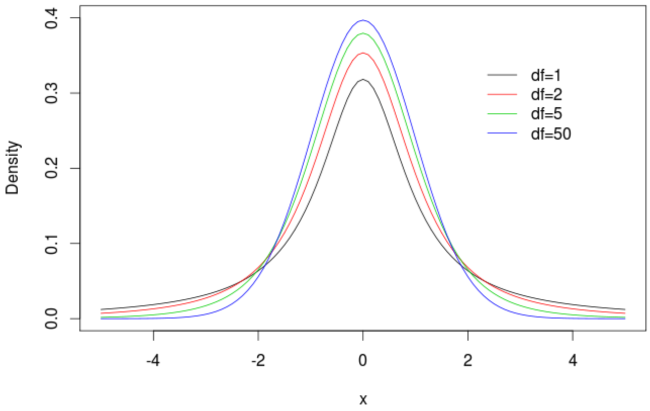


Figure 6.2: The pdfs of Student t distributions with various degrees of freedom (df).

We can use `R` to calculate the pdf, cdf, and quantiles for this distribution. For example, the following `R` script produces Figure 6.2.

```
curve(dt(x,df=1),ylim=c(0,0.4),xlim=c(-5,5),col=1,ylab="Density")
curve(dt(x,df=2),col=2,add=TRUE)
curve(dt(x,df=5),col=3,add=TRUE)
curve(dt(x,df=50),col=4,add=TRUE)
legend(2.1,0.35,lty=1,bty="n",
                  legend=c("df=1","df=2","df=5","df=50"),col=1:4)
```

To obtain the 0.975 quantile of the t_{df} distribution for df $=$ $1, 2, 5, 50$, and 100, enter the following commands.

```
> qt(0.975,df=c(1,2,5,50,100))

[1] 12.706205  4.302653  2.570582  2.008559  1.983972
```

As a comparison, the 0.975 quantile for the standard normal distribution is given by `qnorm(0.975)` = 1.959964 ($\approx$ 1.96).

Returning to the pivot T in (6.5), it has a t_{n-1} distribution. By repeating the rearrangement steps from Section 6.3.1, we find the following exact confidence interval ☞ 98 for μ in terms of the quantiles of the t_{n-1} distribution.

Theorem 6.3: Exact Confidence Interval for μ

Let $X_1, X_2, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu, \sigma^2)$ and let q be the $1-\alpha/2$ quantile of the Student's t_{n-1} distribution. An exact stochastic confidence interval for μ is

$$\bar{X} \pm q\frac{S}{\sqrt{n}}\,. \tag{6.8}$$

■ **Example 6.5 (Volume of a Drop of Water)** A buret is a glass tube with scales that can be used to add a specified volume of a fluid to a receiving vessel. We wish to determine a 95% confidence interval for the average volume of *one* drop of water that leaves the buret, based on the data in Table 6.1.

Table 6.1: An experiment with a buret

Volume in buret (ml)	
initial	25.36
after 50 drops	22.84
after 100 drops	20.36

Our model for the data is as follows: let X_1 be the volume of the first 50 drops, and X_2 the volume of the second 50 drops. We assume that X_1, X_2 are iid and $\mathcal{N}(\mu, \sigma^2)$ distributed, with unknown μ and σ^2. Note that μ is the expected volume of 50 drops, and therefore $\mu/50$ is the expected volume of one drop.

With $n = 2$ and $\alpha = 0.05$, we have that the 0.975 quantile of the t_1 distribution is $q = 12.71$. The outcomes of X_1 and X_2 are respectively $x_1 = 2.52$ and $x_2 = 2.48$. Hence,

$$s = \sqrt{(2.52 - 2.50)^2 + (2.48 - 2.50)^2} = 0.02\sqrt{2}\ .$$

Hence, a numerical 95% CI for μ is

$$2.50 \pm 12.71 \times 0.02 = (2.25, 2.75)\ .$$

However, we want a 95% CI for $\mu/50$! We leave it as an exercise to show that we can simply divide the 95% CI for μ by 50 to obtain a 95% CI for $\mu/50$. Thus, a 95% (numerical) confidence interval for the average volume of one drop of water is

$$(0.045, 0.055) \quad (\mathrm{ml})\ .$$

■

Confidence Interval for σ^2

Next, we construct a confidence interval for σ^2. As before, let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$. Consider the sample variance

$$S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - \bar{X})^2\ .$$

It turns out that $(n-1)S^2/\sigma^2 = \sum_{i=1}^{n}(X_i - \bar{X})^2/\sigma^2$ has a known distribution, called the χ^2 **distribution**, where χ is the Greek letter *chi*. Hence, the distribution is also written (and pronounced) as the chi-squared distribution. Like the t distribution, the χ^2 distribution is actually a family of distributions, depending on a parameter that is again called the **degrees of freedom**. We write $Z \sim \chi^2_{\text{df}}$ to denote that Z has a chi-square distribution with df degrees of freedom. Figure 6.3 shows the pdf of the $\chi^2_1, \chi^2_2, \chi^2_5$, and χ^2_{10} distributions. Note that the pdf is not symmetric and starts at $x = 0$. The χ^2_1 has a density that is infinite at 0, but that is no problem — as long as the total integral under the curve is 1.

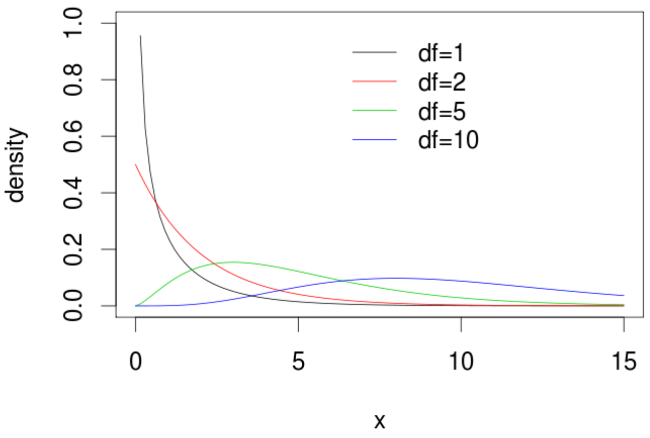


Figure 6.3: The pdfs of chi-square distributions with various degrees of freedom (df).

Figure 6.3 was made in a very similar way to Figure 6.2, mostly by replacing `dt` with `dchisq` in the `R` code. Here is the beginning of the script — you can work out the rest.

```
> curve(dchisq(x,df=1),xlim=c(0,15),ylim=c(0,1),ylab="density")
```

To obtain the 0.025 and 0.975 quantiles of the χ^2_{24} distribution, for example, we can issue the command:

```
> qchisq(p=c(0.025,0.975),24)

[1] 12.40115 39.36408
```

Because $(n-1)S^2/\sigma^2$ has a χ^2_{n-1} distribution, if we denote the $\alpha/2$ and $1-\alpha/2$ quantiles of this distribution by q_1 and q_2, then

$$\mathbb{P}\left(q_1 < \frac{(n-1)}{\sigma^2}S^2 < q_2\right) = 1 - \alpha\ .$$

Rearranging, this shows

$$\mathbb{P}\left(\frac{(n-1)S^2}{q_2} < \sigma^2 < \frac{(n-1)S^2}{q_1}\right) = 1 - \alpha\ .$$

This gives the following exact confidence interval for σ^2 in terms of the quantiles of the χ^2_{n-1} distribution.

Theorem 6.4

Let $X_1, X_2, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu, \sigma^2)$ and let q_1 and q_2 be the $\alpha/2$ and $1-\alpha/2$ quantiles of the χ^2_{n-1} distribution. An exact stochastic confidence interval for σ^2 is

$$\left(\frac{(n-1)S^2}{q_2}, \frac{(n-1)S^2}{q_1} \right) . \tag{6.9}$$

■ **Example 6.6 (Aspirin)** On the label of a certain packet of aspirin it is written that the standard deviation of the tablet weight (actually mass) is 1.0 mg. To investigate if this is true we take a sample of 25 tablets and discover that the sample standard deviation is 1.3mg. A 95% numerical confidence interval for σ^2 is

$$\left(\frac{24 \times 1.3^2}{39.4}, \frac{24 \times 1.3^2}{12.4} \right) = (1.04, 3.27) ,$$

where we have used (in rounded numbers) $q_1 = 12.4$ and $q_2 = 39.4$ calculated before with the `qchisq()` function. A 95% numerical confidence interval for σ is found by taking square roots (why?):

$$(1.02, 1.81) .$$

Note that this CI does not contain the asserted weight of 1.0 mg. We therefore have some doubt whether the "true" standard deviation is indeed equal to 1.0 mg. ■

6.3.3 Normal Data, Two Samples

Consider now *two* independent samples $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_n$ from respectively a $\mathcal{N}(\mu_X, \sigma_X^2)$ and $\mathcal{N}(\mu_Y, \sigma_Y^2)$ distribution. We wish to make a confidence interval (approximate or exact) for $\mu_X - \mu_Y$.

Approximate Confidence Interval for $\mu_X - \mu_Y$

To make an approximate confidence interval for $\mu_X - \mu_Y$, we can reason in similar way as in Section 6.3.1. By the central limit theorem, we have

$$\bar{X} - \bar{Y} \overset{\text{approx.}}{\sim} \mathcal{N}(\mu_X - \mu_Y, \sigma_X^2 + \sigma_Y^2).$$

So, if we standardize and replace σ_X^2 and σ_Y^2 with their sample variances, we have

$$\frac{\bar{X} - \bar{Y}}{\sqrt{\frac{S_X^2}{m} + \frac{S_Y^2}{n}}} \overset{\text{approx.}}{\sim} \mathcal{N}(0, 1).$$

For small m and n the standard normal approximation may not be very accurate. Fortunately, it is possible to obtain a much better approximation using a Student distribution where df is given by the so-called **effective degrees of freedom**:

$$\mathrm{df} = \frac{\left(\frac{S_X^2}{m} + \frac{S_Y^2}{n}\right)^2}{\frac{1}{m-1}\left(\frac{S_X^2}{m}\right)^2 + \frac{1}{n-1}\left(\frac{S_Y^2}{n}\right)^2}. \tag{6.10}$$

We thus have the following approximate confidence interval for $\mu_x - \mu_Y$, using the above *Satterthwaite approximation*.

Theorem 6.5: Confidence Interval for $\mu_X - \mu_Y$

Let $X_1, X_2, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu, \sigma^2)$ and $Y_1, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu_Y, \sigma_Y^2)$ be independent, and let q be the $1 - \alpha/2$ quantile of the Student's t_{df} distribution, with df as in (6.10). An approximate $1 - \alpha$ stochastic confidence interval for μ is

$$\bar{X} - \bar{Y} \pm q\sqrt{\frac{S_X^2}{m} + \frac{S_Y^2}{n}}\,. \tag{6.11}$$

■ **Example 6.7 (Human Movement Study)** A human movement student has a theory that the expected mass of 3rd year students differs from that of 1st years. To investigate this theory, random samples are taken from each of the two groups. A sample of 15 1st years has a mean of 62.0kg and a standard deviation of 15kg, while a sample of 10 3rd years has a mean of 71.5kg and a standard deviation of 12kg. Does this show that the expected masses are indeed different?

Here we have $m = 15$ and $n = 10$. The outcomes $\bar{X} - \bar{Y}$ is $\bar{x} - \bar{y} = 62 - 71.5 = -9.5$. Using (6.10), the effective degrees of freedom is df $= 22.09993$. You may verify also that

$$\sqrt{\frac{s_X^2}{m} + \frac{s_Y^2}{n}} = 5.422177.$$

To construct a 95% numerical confidence interval for $\mu_X - \mu_Y$, we need to also evaluate the 0.975 quantile of the t_{df} distribution, using the R command `qt(0.975,22.09993)`. This gives $q = 2.073329$. So that the 95% numerical confidence interval for $\mu_X - \mu_Y$ is given by

$$-9.5 \pm 2.073329 \times 5.422177 = (-20.74,\ 1.74)\,.$$

This contains the value 0, so there is not enough evidence to conclude that the two expectations are different. ■

6.3.4 Binomial Data, One Sample

How do we construct an approximate confidence interval for binomial data? Let us look at a concrete example first.

■ **Example 6.8 (Opinion Poll)** In an opinion poll of 1000 registered voters, 227 voters say they will vote for the Greens. How can we construct a 95% confidence interval for the proportion p of Green voters of the total population? A systematic way to proceed is to view the datum, 227, as the outcome of a random variable X (the number of Green voters under 1000 registered voters) with a $\mathsf{Bin}(1000, p)$ distribution. In other words, we view X as the total number of "Heads" (= votes Green) in a coin flip experiment with some unknown probability p of getting Heads. Note that this is only a *model* for the data. In practice it is not always possible to truly select 1000 people at random from the population and find their true party preference. For example, a randomly selected person may not wish to participate or could deliberately give the "wrong answer". ■

Now, let us proceed to make a confidence interval for p, in the general situation that we have an outcome of some random variable X with a $\mathsf{Bin}(n, p)$ distribution. It is not so easy to find an exact confidence interval for p that satisfies (6.4) in Definition 6.1. ☞ 87 Instead, for large n we rely on the central limit theorem (see Section 5.5) to construct an *approximate* confidence interval. The reasoning is as follows:

For large n, X has approximately a $\mathcal{N}(np, np(1-p))$ distribution. Let $\widehat{P} = X/n$ denote the estimator of p. We use capital letter $\widehat{P}$ to stress that the estimator is a random variable. The outcome of $\widehat{P}$ is denoted $\widehat{p}$, which is an estimate of the parameter p. Then $\widehat{P}$ has approximately a $\mathcal{N}(p, p(1-p)/n)$ distribution. For some small α (e.g., $\alpha = 0.05$) let q be the $1 - \alpha/2$ quantile of the standard normal distribution. Thus, with Φ the cdf of the standard normal distribution, we have

$$\Phi(q) = 1 - \alpha/2 \, .$$

Then, using the pivot variable

$$\frac{\widehat{P} - p}{\sqrt{p(1-p)/n}},$$

which is approximately standard normal, we have

$$\mathbb{P}\left(-q < \frac{\widehat{P} - p}{\sqrt{p(1-p)/n}} < q\right) \approx 1 - \alpha \, .$$

Rearranging gives:

$$\mathbb{P}\left(\widehat{P} - q\sqrt{\frac{p(1-p)}{n}} < p < \widehat{P} + q\sqrt{\frac{p(1-p)}{n}}\right) \approx 1 - \alpha \, .$$

This would suggest that we take $\widehat{p} \pm q\sqrt{\frac{p(1-p)}{n}}$ as an numerical (approximate) $(1-\alpha)$ confidence interval for p, were it not for the fact that the bounds still contain the unknown p! However, for large n the estimator $\widehat{P}$ is close to the real p, so that we have

$$\mathbb{P}\left(\widehat{P} - q\sqrt{\frac{\widehat{P}(1-\widehat{P})}{n}} < p < \widehat{P} + q\sqrt{\frac{\widehat{P}(1-\widehat{P})}{n}}\right) \approx 1-\alpha\ .$$

Hence, an numerical *approximate* $(1-\alpha)$-confidence interval for p is

$$\widehat{p} \pm q\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}}\ . \tag{6.12}$$

■ **Example 6.9 (Opinion Poll (Continued))** In Example 6.8, we have $\widehat{p} = 227/1000 = 0.227$, and $q = 1.960$, so that an approximate 95% numerical CI for p is given by

$$(0.227 - 1.960 \times 0.0132, 0.227 + 1.960 \times 0.0132) = (0.20, 0.25)\ .$$

■

6.3.5 Binomial Data, Two Samples

We next wish to construct an approximate confidence interval for the difference of two proportions. We again start with an example.

■ **Example 6.10 (Nightmares)** Two groups of men and women are asked whether they experience nightmares "often" (at least once a month) or "seldom" (less than once a month). The results are given in Table 6.2.

Table 6.2: Counts of people experiencing nightmares.

	Men	Women	Total
Often	55	60	115
Seldom	105	132	237
Total	160	192	

The observed proportions of frequent nightmares by men and women are 34.4% and 31.3%. Is this difference statistically significant, or due to chance? To assess this we could make a confidence interval for the difference of the true proportions p_X and p_Y. ■

The general model is as follows.

- Let X be the number of "successes" in Group 1; $X \sim \mathsf{Bin}(m, p_X)$, where p_X is unknown.
- Let Y be the number of "successes" in Group 2; $Y \sim \mathsf{Bin}(n, p_Y)$, where p_Y is unknown.
- Assume X and Y are independent.

We wish to compare the two proportions via an approximate $(1-\alpha)$-confidence interval for $p_X - p_Y$. The easiest way is to again rely on the central limit theorem. We assume from now on that m and n are sufficiently large (mp_X and $m(1-p_X) > 5$, np_Y and $n(1-p_Y) > 5$), so that the normal approximation the binomial distribution can be applied.

Let $\widehat{P}_X = X/m$ and $\widehat{P}_Y = Y/n$. By the central limit theorem,

$$\frac{\widehat{P}_X - \widehat{P}_Y - (p_X - p_y)}{\sqrt{\frac{p_X(1-p_X)}{m} + \frac{p_Y(1-p_Y)}{n}}}$$

has approximately a $\mathcal{N}(0,1)$ distribution. Hence, with q the $(1-\alpha/2)$-quantile of the $\mathcal{N}(0,1)$ distribution (as in Section 6.3.4), we have

$$\mathbb{P}\left(-q \leqslant \frac{\widehat{P}_X - \widehat{P}_Y - (p_X - p_Y)}{\sqrt{\frac{p_X(1-p_X)}{m} + \frac{p_Y(1-p_Y)}{n}}} \leqslant q\right) \approx 1 - \alpha\ .$$

Rewriting, this gives

$$\begin{aligned}\mathbb{P}\Bigg(\widehat{P}_X - \widehat{P}_Y - q\sqrt{\frac{p_X(1-p_X)}{m} + \frac{p_Y(1-p_Y)}{n}} &\leqslant p_X - p_Y \\ &\leqslant \widehat{P}_X - \widehat{P}_Y + q\sqrt{\frac{p_X(1-p_X)}{m} + \frac{p_Y(1-p_Y)}{n}}\Bigg) \\ &\approx 1 - \alpha.\end{aligned}$$

As in the 1-sample case of Section 6.3.4, the same is *approximately* true, if we replace p_X and p_Y in the square root terms above by $\widehat{P}_X$ and $\widehat{P}_Y$ (law of large numbers). We now have stochastic bounds which only depend on the data.

Hence, an numerical *approximate* $100(1-\alpha)\%$ stochastic confidence interval for $p_X - p_Y$ is

$$\widehat{p}_X - \widehat{p}_Y \pm q\sqrt{\frac{\widehat{p}_X(1-\widehat{p}_X)}{m} + \frac{\widehat{p}_Y(1-\widehat{p}_Y)}{n}}\ , \tag{6.13}$$

where q is the $1-\alpha/2$ quantile of the standard normal distribution.

■ **Example 6.11 (Nightmares (Continued))** We continue Example 6.10. We have $\widehat{p}_X = 55/160$, $\widehat{p}_Y = 60/192$ and $q = 1.96$, so that an approximate 95% numerical CI for $p_X - p_Y$ is given by

$$(0.031 - 0.099, 0.031 + 0.099) = (-0.07, 0.13) .$$

This interval contains 0, so there is no evidence that men and women are different in their experience of nightmares. ■

CHAPTER 7

Hypothesis Testing

Hypothesis testing involves making *decisions* about certain hypotheses on the basis of the observed data. In many cases we have to decide whether the observations are due to “chance” or due to an “effect”. We will guide you through the steps that need to be taken to carry out a statistical test. Standard tests for various one and twosample problems involving Normal and Binomial random variables are provided.

7.1 Introduction

We had a first look at hypothesis testing in Chapter 1. Namely, in Section 1.1 we ☞ 9
investigated a coin flip experiment (is the coin fair?) and in Section 1.4 we studied ☞ 17
Alice’s cola experiment (does drinking caffeinated Diet cola increase the heart rate?). In this chapter we will revisit both these experiments and describe their analysis in a framework that is more generally applicable.

In particular, suppose that we have a general model for data $\mathbf{X}$ that is described by a family of probability distributions that depend on a parameter $\boldsymbol{\theta}$. For example, in the onesample normal model, we have $\mathbf{X} = (X_1, \ldots, X_n)$, where $X_1, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu, \sigma^2)$. In this case $\boldsymbol{\theta}$ is the vector (μ, σ^2).

The aim of *hypothesis testing* is to decide, on the basis of the observed data $\mathbf{x}$, which of two competing hypotheses on the parameters is true. For example, one hypothesis could be that $\mu = 0$ and the other that $\mu \neq 0$. Traditionally, the two hypotheses do not play equivalent roles. One of the hypothesis contains the “status quo” statement. This is the **null hypothesis**, often denoted by H_0. The **alternative hypothesis**, denoted H_1, contains the statement that we wish to show. A good analogy is found in a court of law. Here, H_0 (present state of affairs) could be the statement that a suspect is innocent, while H_1 is the statement that the suspect is guilty (what needs to be demonstrated). The legal terms such as “innocent until proven guilty”, and “without reasonable doubt” show clearly the asymmetry between the hypotheses. We should only be prepared to reject H_0 if the observed data, that is the evidence, is very unlikely to have happened under H_0.

The decision whether to reject H_0 or not is dependent on the outcome of a **test statistic** T, which is a function of the data $\mathbf{X}$ only. The **P-value** is the probability that under H_0 the (random) test statistic takes a value as extreme as or more extreme than the one observed. Let t be the observed outcome of the test statistic T. We consider three types of tests:

- **Left one-sided test**. Here H_0 is rejected for small values of t, and the P-value is defined as $p = \mathbb{P}_{H_0}(T \leqslant t)$.
- **Right one-sided test**: Here H_0 is rejected for large values of t, and the P-value is defined as $p = \mathbb{P}_{H_0}(T \geqslant t)$,
- **Two-sided test**: In this test H_0 is rejected for small or large values of t, and the P-value is defined as $p = \min\{2\mathbb{P}_{H_0}(T \leqslant t),\ 2\mathbb{P}_{H_0}(T \geqslant t)\}$.

The smaller the P-value, the greater the strength of the evidence against H_0 provided by the data. As a rule of thumb (see also Figure 1.5):

☞ 20

$p < 0.10$	weak evidence,
$p < 0.05$	moderate evidence,
$p < 0.01$	strong evidence.

The following decision rule is generally used to decide between H_0 and H_1:

Decision rule : *Reject H_0 if the* P-value *is smaller than some* **significance level** α.

In general, a statistical test involves the following steps.

Steps for a Statistical Test

1. Formulate a statistical model for the data.
2. Give the null and alternative hypotheses (H_0 and H_1).
3. Choose an appropriate test statistic.
4. Determine the distribution of the test statistic under H_0.
5. Evaluate the outcome of the test statistic.
6. Calculate the P-value.
7. Accept or reject H_0 based on the P-value.

Choosing an appropriate test statistic is akin to selecting a good estimator for the unknown parameter $\boldsymbol{\theta}$. The test statistic should summarize the information about $\boldsymbol{\theta}$ and make it possible to distinguish between the two hypotheses.

■ **Example 7.1 (Blood Pressure)** Suppose the systolic blood pressure for white males aged 35–44 is known to be normally distributed with expectation 127 and standard deviation 7. A paper in a public health journal considers a sample of 101 diabetic males and reports a sample mean of 130. Is this good evidence that diabetics have on average a higher blood pressure than the general population?

To assess this, we could ask the question how likely it would be, *if diabetics were similar to the general population*, that a sample of 101 diabetics would have a mean blood pressure this far from 127.

Let us perform the seven steps of a statistical test. A reasonable model for the data is $X_1, \ldots, X_{101} \sim_{\text{iid}} \mathcal{N}(\mu, 49)$. Alternatively, the model could simply be $\bar{X} \sim \mathcal{N}(\mu, 49/101)$, since we only have an outcome of the sample mean of the blood pressures. The null hypothesis (the status quo) is $H_0 : \mu = 127$; the alternative hypothesis is $H_1 : \mu > 127$. We take $\bar{X}$ as the test statistic. Note that we have a right one-sided test here, because we would reject H_0 for high values of $\bar{X}$. Under H_0 we have $\bar{X} \sim \mathcal{N}(127, 49/101)$. The outcome of $\bar{X}$ is 130, so that the P-value is given by

$$\mathbb{P}(\bar{X} \geqslant 130) = \mathbb{P}\left(\frac{\bar{X} - 127}{\sqrt{49/101}} \geqslant \frac{130 - 127}{\sqrt{49/101}}\right) = \underbrace{\mathbb{P}(Z \geqslant 4.31)}_{\texttt{1-pnorm(4.31)}} \approx 8.16 \cdot 10^{-6} ,$$

where $Z \sim \mathcal{N}(0, 1)$. So it is extremely unlikely that the event $\{\bar{X} \geqslant 130\}$ occurs if the two groups are the same with regard to blood pressure. However, the event *has* occurred. Therefore, there is *strong* evidence that the blood pressure of diabetics differs from the general public. ■

■ **Example 7.2 (Biased Coin (Revisited))** We revisit Example 1.1, where we observed 60 out of 100 Heads for a coin that we suspect to be biased towards Heads. Is there enough evidence to justify our suspicion? ☞ 10

What are the 7 hypothesis steps in this case? A good model (step 1) for the data X (the total number of Heads in 100 tosses) is: $X \sim \mathsf{Bin}(100, p)$, with the probability of Heads, p, is unknown. We would like to show (step 2) the hypothesis $H_1 : p > 1/2$; otherwise, we do not reject (accept) the null hypothesis $H_0 : p = 1/2$. Our test statistic (step 3) could simply be X. Under H_0, $X \sim \mathsf{Bin}(100, 1/2)$ (step 4). The outcome of X (step 5) is $x = 60$, so the P-value for this right one-sided test is

$$\mathbb{P}(X \geqslant 60) = \underbrace{\sum_{k=60}^{100} \binom{100}{k} \left(\frac{1}{2}\right)^{100}}_{\texttt{1-pbinom(59,100,1/2)}} \approx 0.02844397 .$$

This is quite small. Hence, we have *reasonable* evidence that the die is loaded. ■

In the rest of this chapter we are going to look at a selection of basic tests, involving one or two iid samples from either a Normal or Bernoulli distribution.

7.2 One-sample t-test

☞ 17 Let us return to Alice's cola experiment in Section 1.4, and consider only the changes in pulse rate for the Decaf (control) group; see Table 7.1. Is there evidence that the expected change in pulse rate is greater than 0 for this group? If we found such evidence for the control group, this would put doubt on any conclusion that an increase in pulse rate for the treatment group is only due to caffeine — there could be other factors involved.

Table 7.1: Changes in pulse rate for the Decaf group in Alice's cola experiment.

4	10	7	−9	5	4	5	7	6	12

To answer this question, we again consider an appropriate model for this situation. We represent the observations by $X_1, \ldots, X_{10}$, and assume that they form an iid sample from a $\mathcal{N}(\mu, \sigma^2)$ distribution, where both μ and σ^2 are *unknown*; note that is different to Example 7.1, where the variance is known. The hypotheses can now be formulated as: $H_0 : \mu = 0$ against $H_1 : \mu > 0$.

Which test statistic should we choose? Since we wish to make a statement about μ, the test statistic should reflect this. We could take $\bar{X}$ as our test statistic and reject H_0 for large values of $\bar{X}$. However, this leads to a complication. It looks like our null hypothesis only contains one parameter value, but in fact it contains *many*, because we should have written

$$H_0 : \mu = 0, \quad 0 < \sigma^2 < \infty \,.$$

It is the unknown variance σ^2 that leads to the complication in choosing $\bar{X}$ as our test statistic. To see this, consider the following two cases. First consider the case where the standard deviation σ is small, say 1. In that case, $\bar{X}$ is under H_0 very much concentrated around 0, and therefore any deviation from 0, such as 7 would be most unlikely under H_0. We would therefore reject H_0. On the other hand, if σ is large, say 10, then a value of 7 could very well be possible under H_0, so we would not reject it.

This shows that $\bar{X}$ is not a good test statistic, but that we should "scale" it with the standard deviation. That is, we should measure our deviation from 0 in units of σ rather than in units of 1. However, we do not know σ. But this is easily fixed by replacing σ with an appropriate estimator. This leads to the test statistic

$$T = \frac{\bar{X}}{S/\sqrt{10}} \,.$$

☞ 100 The factor $\sqrt{10}$ is a "standardising" constant which enables us to utilize Theorem 6.7. Namely, under H_0 the random variable T has a $\mathrm{t}_{n-1} = \mathrm{t}_9$ distribution. Note that this is true for *any* value of σ^2. The observed outcome of T is

$$\frac{5.1}{5.59/\sqrt{10}} \approx 2.89\ .$$

Using R,

```
> 1 - pt(2.89,df=9)
```

```
[1]  0.008942135
```

we find the P-value

$$\mathbb{P}_{H_0}(T \geqslant 2.89) \approx 0.0089\ .$$

Since this is rather small, we reject H_0. Therefore, there is strong evidence that the expected difference in pulse rate is greater than 0.

The above test is often called a **onesample *t*-test**. In general, let $X_1, \ldots, X_n \sim_{\text{iid}} \mathcal{N}(\mu, \sigma^2)$. Let μ_0 be a given number. We wish to test the hypothesis $H_0 : \mu = \mu_0$ against left-, right-, and two-sided alternatives by using the test statistic

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}\ , \tag{7.1}$$

with $\bar{X} = \frac{1}{n}\sum_{i-1}^{n} X_i$ and $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$. Under H_0 we have $T \sim \mathrm{t}_{n-1}$. Reject/accept H_0 based on the magnitude of the P-value. Note that the P-value depends on whether the test is left one-sided, right one-sided or two-sided. In the example above, the test is right-one sided (we reject H_0 for large value of T).

Using R

Note that in to order to carry out a onesample t-test, we only need the summary statistics $\bar{x}$ and s of the data. When the individual measurements are available, it is convenient to carry out the t-test using the R function `t.test`. As an example, we enter the data in Table 7.1 into R and print out the sample mean and standard deviation using the function `sprintf`, which can be used to format output neatly:

```
> x = c(4, 10, 7, -9, 5, 4, 5, 7, 6, 12)
> sprintf(fmt="mean=%s   sd=%3.3f", mean(x), sd(x))
```

```
"mean=5.1   sd=5.587"
```

Applying the `t.test` function, we get for the above data:

```
> t.test(x,alternative="greater")
```

```
  data:  x
t = 2.8868, df = 9, p-value = 0.008989
alternative hypothesis: true mean is greater than 0
95 percent confidence interval:
 1.8615    Inf
sample estimates:
mean of x
      5.1
```

The main output of the function `t.test` are: the outcome of the T statistic (t = 2.8868), the P-value = 0.008989, the alternative hypothesis (*true mean is greater than 0*) and the sample mean $\bar{x} = 5.1$. To output just the P-value, we can use:

```
> t.test(x,alternative="greater")$p.value


[1] 0.008988979
```

By default, the `t.test` function takes a two-sided alternative. The option `alternative = "greater"` forces a right-onesided alternative. Note that in this case `t.test` returns a onesided confidence interval. To obtain a 99% two-sided confidence interval for μ we can use:

```
> t.test(x,conf.level=0.99)$conf.int


[1] -0.6413761 10.8413761
attr(,"conf.level")
[1] 0.99
```

To find the variable names that are returned by a function, use `names()`, as in

```
h = t.test(x)
names(h)


[1] "statistic" "parameter" "p.value" "conf.int" "estimate"
[6] "null.value" "alternative" "method" "data.name"
```

7.3 Type-I Error, Type-II Error, and Power

In any hypothesis test we can make two types of mistakes, illustrated in Table 7.2.

Table 7.2: Type-I and type-II errors

	True state of nature	
Decision	**H_0 is true**	**H_1 is true**
Accept H_0	Correct	Type II Error
Reject H_0	Type I Error	Correct

Whether we make a right or wrong decision is the result of a random process. Thus, for any statistical test where we make a decision in the end, there is a *probability* of a Type I error, Type II error, or correct decision. Ideally, we would like to construct tests which make the probabilities of Type-I and Type-II errors, (let's call them e_I and e_{II}) as small as possible. Unfortunately, this is not possible, because the two errors "compete" with each other: if we make e_I smaller, e_{II} will increase, and vice versa.

Because, as mentioned, the null and alternative hypothesis do not play equivalent roles, a standard approach is to keep the probability e_I of a Type I error at (or below) a certain threshold: the significance level, say 0.05. The decision rule: *reject* H_0 *if the* P-value *is smaller than some significance level* α ensures that $e_I \leqslant \alpha$.

Next, given that e_I remains at (or below) level α, we should try to make e_{II} as small as possible. The probability $1 - e_{II}$ is called the **power** of the test. It is the probability of making the right decision (reject H_0) under some alternative in H_1. So, minimizing the probability of a Type II error is the same as maximizing the power. Note that the power heavily depends on what alternative is used.

■ **Example 7.3 (Simulating the Power)** Suppose we have a one-sample t-test, where we want to test $H_0 : \mu = 0$ versus $H_1 : \mu > 0$. Our test statistic is $\bar{X}/(S/\sqrt{n})$ and under H_0, this test statistic has a t_{n-1} distribution. Suppose we have a significance level of $\alpha = 0.05$. What is the power of the test when the real parameters are $\mu = 1$ and $\sigma = 2$, for example?

Imagine what would happen if we conducted the test tomorrow, with the data $X_1, \ldots, X_n$ coming from $\mathcal{N}(1, 4)$. We would form the test statistic $T = \bar{X}/(S/\sqrt{n})$ and then calculate the corresponding P-value for this right-onesided test. In `R` we would do it via: `pt(T,df=n-1)`. Finally, we would reject the null hypothesis if the P-value is less than 0.05. So let's do this many times on a computer and see how many times we correctly reject the null hypothesis. In the program below we use a sample size of $n = 5$.

```r
R = 1e5   # number of repeats
n = 5     # sample size
mu = 1    # the actual mu
sigma = 2 # the actual standard deviation
pval = vector(mode="numeric", length=R) # initialize P-value vector
t = vector(mode="numeric", length=R)    # initialize test statistic vector

for (i in 1:R){
  x = rnorm(n,mean=mu,sd=sigma) # simulate the data
  t[i] = mean(x)/(sd(x)/sqrt(n)) # test statistic
  pval[i] = 1-pt(t[i],df=n-1)  # P-value
}

pow = print(length(which(pval < 0.05))/R) # estimate of the power

# Or we can use the power.t.test function:
power.t.test(n=n, delta=mu, sig.level=0.05, alternative="one.sided", type="one.sample", sd=sigma)
```

In this way, we calculate a power of 0.24 for the alternative $\mu = 1, \sigma = 2$. This is not so high! With this standard deviation and small sample size it will be very difficult to detect that $\mu = 1$. Let's repeat it with $n = 50$. We now get a power of 0.97, so close to 1. Hence a sample size of 50 is enough to detect a difference of 1 unit, if the standard deviation is 2.

■

The above example illustrates that the power depends on various factors: the sample size, the significance level, as well as μ and σ. In fact (you can verify it yourself) the power in the above code only depends on μ/σ, which is sometimes called the "signal to strength ratio". In R, we can make power calculations via the **power.t.test** function. Here is the output of the last lines in the code above:

```
One-sample t test power calculation

              n = 5
          delta = 1
             sd = 2
      sig.level = 0.05
          power = 0.2389952
    alternative = one.sided
```

A power analysis, as carried out above, allows us to choose a sample size large enough to determine some minimal effect, as long as we have an idea of the standard deviation. The latter can be estimated with a trial run, for example.

7.4 One-sample Test for Proportions

The statistical test in Example 7.2 is an example of a one-sample test for proportions. In this section we explore such tests in more detail, using the following example.

■ **Example 7.4 (Market Research)** In a certain market research study we wish to investigate whether people would prefer a new type of sweetener in a certain brand of yoghurt. Ten people were given two packets of yoghurt, one with the old sweetener and one with the new sweetener. Eight of the ten people preferred the yoghurt with the new sweetener and two preferred the old yoghurt. Is there enough evidence that the new style of yoghurt is preferred?

First we formulate the model. Let $X_1, \ldots, X_{10}$ be such that

$$X_i = \begin{cases} 1 & \text{if person } i \text{ prefers the new yoghurt,} \\ 0 & \text{if person } i \text{ prefers the old yoghurt,} \end{cases}$$

$i = 1, \ldots, 10$. We assume that $X_1, \ldots, X_{10}$ are independent and that for all i, $\mathbb{P}(X_i = 1) = p = 1 - \mathbb{P}(X_i = 0)$, for some unknown p (between 0 and 1). We wish to test

$$H_0 : p = 0.5 \quad \text{against} \quad H_1 : p > 0.5 \ .$$

As test statistic we could use the total number of people preferring the new yoghurt, $X = \sum_{i=1}^{10} X_i$, and we would reject H_0 for large values of X. Under H_0 the test statistic has a $\mathsf{Bin}(10, 1/2)$ distribution. The P-value is thus, similar to Example 7.2,

$$\mathbb{P}_{H_0}(X \geqslant 8) = \sum_{k=8}^{10} \binom{8}{k} (1/2)^{10} \approx 0.0546875 \ .$$

Note that we can evaluate the probability above in `R` using `1 - pbinom(7,10,0.5)`. Since the P-value is reasonably small (0.055), there is some doubt about H_0. ■

Remark 7.1 Our model above is in a sense over-specific. We assume that we observe the preference X_i for each individual. But in fact, we only observe the total number of preferences $X = X_1 + \cdots + X_n$ for the new yoghurt. An alternative and simpler model would suffice here, namely: let X be the total number of preferences for the new type of yoghurt, we assume $X \sim \mathsf{Bin}(n, p)$, for some unknown p. The test now proceeds in exactly the same way as before.

We now describe the general situation for the **one-sample binomial test**. Suppose that $X_1, \ldots, X_n$ are the results of n independent Bernoulli trials with success parameter p. That is the X_i's are independent and

$$\mathbb{P}(X_i = 1) = p = 1 - \mathbb{P}(X_i = 0) \ .$$

Then, $X := X_1 + \cdots + X_n \sim \mathsf{Bin}(n, p)$. We wish to test $H_0 : p = p_0$ against left-, right-, and two-sided alternatives.

As test statistic we can use X, which under H_0 has a $\mathsf{Bin}(n, p_0)$ distribution. We accept/reject H_0 based on the P-value of the test.

■ **Example 7.5 (Market Research (Continued))** For one-sample binomial test, we can use the R function `binom.test`. The parameters and output are very similar to those used with `t.test` function for one-sample t-test.

```
> binom.test(x=8,n=10,p=0.5,alternative="greater")

        Exact binomial test

data:  8 and 10
number of successes = 8, number of trials = 10, p-value = 0.05469
alternative hypothesis: true probability of success is greater than 0.5
95 percent confidence interval:
 0.4930987 1.0000000
sample estimates:
probability of success
                   0.8
```

■

Using the Normal Approximation

For large n, analogously to Sections 6.3.4, X has approximately a $\mathcal{N}(np, np(1-p))$ distribution and then the estimator $\widehat{P} = X/n$ has approximately a $\mathcal{N}(p, p(1-p)/n)$ distribution. It follows that

$$\frac{\widehat{P}-p}{\sqrt{p(1-p)/n}},$$

has approximately a $\mathcal{N}(0,1)$ distribution. Now, under $H_0: \ p = p_0$, our test statistic

$$Z = \frac{\widehat{P}-p_0}{\sqrt{p_0(1-p_0)/n}},$$

has approximately a $\mathcal{N}(0,1)$ distribution.

■ **Example 7.6 (Market Research (Continued))** Returning to Example 7.4, from our data we have the estimate $\widehat{p} = \frac{8}{10}$. Thus, the outcome of the test statistic is

$$z = \frac{0.8-0.5}{\sqrt{0.5(1-0.5)/10}} = 1.897367.$$

This gives a P-value of $\mathbb{P}_{H_0}(Z \geqslant 1.897367) \approx 0.02889$ (in R type `1 - pnorm(1.897367)`). This approximate P-value is quite different from the one for the exact test (0.05469), as our sample size is not enough large to use the central limit theorem. The R function `prop.test` uses this normal approximation, as in: `prop.test`:

```
> prop.test(x=8,n=10,p=0.5,alternative="greater",correct=FALSE)$p.value

[1] 0.02888979
```

Hence, for small sample sizes it is recommended to use `binom.test`. ■

7.5 Two-sample t-test

We next look at two-sample data, again using Alice's cola experiment as a guiding example. Below we repeat the table and stripplot from Section 1.3. Do the results in Table 7.3 provide any *evidence* that caffeine increases pulse rate?

Table 7.3: Changes in pulse rate for Alice's caffeine experiment.

Caffeinated	17	22	21	16	6	−2	27	15	16	20
Decaf	4	10	7	−9	5	4	5	7	6	12

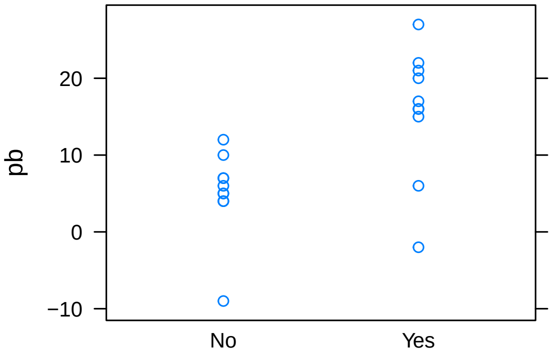


Figure 7.1: A visualization of Alice's caffeine data.

Let us go through the 7 steps of a hypothesis test. First, we could model the data as coming from different normal distributions. Let $X_1, \ldots, X_m$ (with $m = 10$) be the change in heartbeat for the caffeinated Diet cola (treatment) group and let $Y_1, \ldots, Y_n$ (with $n = 10$) be the change in heartbeat for the decaf Diet cola (control) group. We assume that

- $X_1, \ldots, X_m \overset{\text{iid}}{\sim} \mathcal{N}(\mu_X, \sigma_X^2)$.
- $Y_1, \ldots, Y_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu_Y, \sigma_Y^2)$.
- $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ are *independent*,

where μ_X, μ_Y, σ_X^2, and σ_X^2 are unknown parameters. We wish to test $H_0 : \mu_X = \mu_Y$ versus $H_1 : \mu_X > \mu_Y$. Following the reasoning in Section 6.3.3, we use as our test ☞ 104 statistic:

$$T = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{S_X^2}{m} + \frac{S_Y^2}{n}}},$$

which under H_0 has approximately a Student t_{df} distribution where df is given by

$$\mathrm{df} \;=\; \frac{\left(\frac{S_X^2}{m} + \frac{S_Y^2}{n}\right)^2}{\frac{1}{m-1}\left(\frac{S_X^2}{m}\right)^2 + \frac{1}{n-1}\left(\frac{S_Y^2}{n}\right)^2}, \tag{7.2}$$

which we already encountered in (6.10). Even for small m and n this approximation is very accurate. This two-sample t-test is attributed to *Bernard Welch*. This completes steps 1–4. Let us finish the remaining steps of the test by using R as a calculator.

```
x = c(17,22,21,16,6,-2,27,15,16,20)
y = c(4,10,7,-9,5,4,5,7,6,12)
mx = mean(x)
my = mean(y)
sx = sd(x)
sy = sd(y)
a = sx^2/10
b = sy^2/10
t = (mx - my)/sqrt(a + b)
df = (a + b)^2/(a^2/9 + b^2/9)
pval = 1 - pt(t,df=df)
cat("t = ", t, ", df =", df,", pva l=", pval)  #print the values
```

This gives the output (using the `cat` (for concatenate) function):

```
t = 3.37521 , df = 15.74042 , pval = 0.001965818
```

We conclude that there is strong evidence that the caffeine has an effect on the change in pulse beat.

Having defined `x` and `y` as in the above code, we can obtain the same results by using the `t.test` function:

```
> t.test(x,y, alternative="greater")

          Welch Two Sample t-test

data:  x and y
t = 3.3752, df = 15.74, p-value = 0.001966
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 5.159642      Inf
sample estimates:
mean of x mean of y
     15.8       5.1
```

Equal Variance Assumption

In the above analysis, we did not assume that the variances for both groups were equal. If we *do* make such an assumption, it is possible to obtain a test statistic with an *exact* (not just approximate) Student distribution. The reasoning is as follows. To estimate the common variance of the groups (σ^2, say), we should “pool” the squared deviations from the means, giving the **pooled sample variance**

$$S_p^2 = \frac{(m-1)S_X^2 + (n-1)S_Y^2}{m+n-2} = \frac{\sum_{i=1}^m (X_i - \bar{X})^2 + \sum_{j=1}^n (Y_j - \bar{Y})^2}{m+n-2},$$

where $\bar{X} = m^{-1} \sum_{i=1}^m X_i$ and $\bar{Y} = n^{-1} \sum_{j=1}^n Y_j$. Since, under $H_0 : \mu_X = \mu_Y$ the random variable $\bar{X} - \bar{Y}$ has a $\mathcal{N}(0, \sigma^2 (1/m + 1/n))$ distribution, a natural test statistic in this case is

$$T = \frac{\bar{X} - \bar{Y}}{S_p \sqrt{1/m + 1/n}}\,, \tag{7.3}$$

It turns out that, under H_0 this test statistic has exactly a t_{m+n-2} distribution. We accept/reject H_0 depending on the P-value associated with the alternative (left-, right-, or two-sided).

For the Alice example, we can perform this test using:

```
t.test(x,y,alternative = "greater", var.equal = T)
```

```

        Two Sample t-test

data:  x and y
t = 3.3752, df = 18, p-value = 0.001686
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 5.202718      Inf
sample estimates:
mean of x mean of y
     15.8       5.1
```

We see that the P-value is slightly smaller under the assumption of equal variances, but that in essence we come to the same conclusions.

Paired Data

When conducting a twosample *t*-test, it is important to ascertain that the random variables are not *paired*. Such data often arises in “before–after” experiments or on replicated experiments involving the same subjects, as in the following example.

■ **Example 7.7 (Paired Lab Data)** We wish to compare the results from two labs for a specific examination. Both labs made the necessary measurement on *the same* fifteen patients.

```
>  lab1 = c(22,18,28,26,13,8,21,26,27,29,25,24,22,28,15)
>  lab2 = c(25,21,31,27,11,10,25,26,29,28,26,23,22,25,17)
```

In this case the measurements between the goups are not independent, as the measurements are conducted on the same patient. For example, both labs report high measurements (29 and 28) for patient 10, and both labs reported low measurements (8 and 10) for patient 6. ■

In general, suppose we wish to compare the difference in the expectations of two *dependent* random variables X and Y, based on paired samples $\{X_i\}$ and $\{Y_i\}$. To this end, we use the difference random variable $D = X - Y$, and we compare the expected difference $\delta = \mu_X - \mu_Y$ of D with the reference value 0. We are thus back to the case of a **one-sample *t*-test** if we assume a normal model for the difference $D_i = X_i - Y_i \sim \mathcal{N}(\mu_X - \mu_Y, \sigma^2)$. The hypotheses of the test are $H_0 : \mu_X - \mu_Y = 0$ and $H_1 : \mu_X - \mu_Y \neq 0$. Under H_0, the test statistic is:

$$T = \frac{\bar{D}}{S/\sqrt{n}} \sim \mathsf{t}_{n-1},$$

with $\bar{D} = \frac{1}{n}\sum_{i=1}^{n} D_i$, and $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(D_i - \bar{D})^2$.

■ **Example 7.8 (Paired Lab Data (Continued))** To use `t.test` on the pair lab data, we need to set the parameter `paired=TRUE`:

```
> t.test(lab1,lab2,paired=TRUE)


Paired t-test

data:  lab1 and lab2
t = -1.7618, df = 14, p-value = 0.09991
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.0695338  0.2028671
sample estimates:
mean of the differences
             -0.9333333
```

Since the P-value is rather high (0.1) there is not enough evidence to conclude that the two labs give different results on average. ■

7.6 Two-sample Test for Proportions

In this section we consider two-sample binomial data and construct a test to compare the two proportions.

■ **Example 7.9 (Are ABC Viewers More Left-wing?)** A politician believes that audience members of the ABC news are in general more left wing than audience members of a commercial news broadcast. A poll of two-party preferences is taken. Of seventy ABC viewers, 40 claim left wing allegiance, while of 100 commercial station viewers, 50 claim left wing allegiance. Is there any evidence to support the politician's claim?

Our model is as follows. Let X be the number of left-wing ABC viewers out of $m = 70$, and let Y be the number of left-wing "commercial" viewers out of $n = 100$. We assume that X and Y are independent, with $X \sim \mathsf{Bin}(m, p_X)$ and $Y \sim \mathsf{Bin}(n, p_Y)$, for some unknown p_X and p_Y. We wish to test $H_0 : p_X = p_Y$ against $H_1 : p_X > p_Y$.

Since m and n are fairly large here, we proceed by using the central limit theorem (CLT), analogously to Sections 6.3.4 and 6.3.5. Let $\widehat{P}_X := X/m$ and $\widehat{P}_Y := Y/n$ be the empirical proportions. By the CLT $\widehat{p}_X$ has approximately a $\mathcal{N}(p_X, p_X(1 - p_X)/m)$ distribution, and $\widehat{p}_Y$ has approximately a $\mathcal{N}(p_Y, p_Y(1 - p_Y)/n)$ distribution. It follows that ☞ 107

$$\frac{\widehat{P}_X - \widehat{P}_Y}{\sqrt{\frac{p_X(1-p_X)}{m} + \frac{p_Y(1-p_Y)}{n}}}$$

has approximately a $\mathcal{N}(0, 1)$ distribution. Now, under H_0, $p_X = p_Y = p$, say, and hence under H_0

$$\frac{\widehat{P}_X - \widehat{P}_Y}{\sqrt{\frac{p(1-p)}{m} + \frac{p(1-p)}{n}}}$$

has approximately a $\mathcal{N}(0, 1)$ distribution. As we don't know what p is, we need to estimate it. If H_0 is true, then $X + Y \sim \mathsf{Bin}(m + n, p)$, and thus p can be estimated by the *pooled* success proportion

$$\widehat{P} := \frac{X + Y}{n + m} . \tag{7.4}$$

Concluding, we take as our test statistic:

$$Z = \frac{\widehat{P}_X - \widehat{P}_Y}{\sqrt{\widehat{P}(1 - \widehat{P})\left(\frac{1}{m} + \frac{1}{n}\right)}}, \tag{7.5}$$

which under H_0 has approximately a $\mathcal{N}(0, 1)$ distribution. ■

Our general formulation for the **two-sample binomial test** (also called the **test for proportions**) is as follows. First, the model is:

- $X \sim \mathsf{Bin}(m, p_X)$, where p_X is unknown.
- $Y \sim \mathsf{Bin}(n, p_Y)$, where p_Y is unknown.
- X and Y independent.

We wish to test $H_0 : p_X = p_Y$ against various alternatives (left one-sided, right one-sided, and two-sided). As test statistic we use Z given in (7.5). We accept/reject H_0 on the basis of the P-value.

■ **Example 7.10 (Are ABC Viewers More Left-wing? (Continued))** Returning to Example 7.9, from our data we have the estimates $\widehat{p_X} = \frac{40}{70}$, $\widehat{p_Y} = \frac{50}{100}$, and

$$\widehat{p} = \frac{40 + 50}{70 + 100} = \frac{90}{170} .$$

Thus, the outcome of the test statistic is

$$\frac{\frac{40}{70} - \frac{50}{100}}{\sqrt{\frac{90}{170} \times \frac{80}{170} \left(\frac{1}{70} + \frac{1}{100}\right)}} = 0.9183 .$$

This gives a P-value of $\mathbb{P}_{H_0}(Z \geqslant 0.9183) \approx 0.1792$ (in R type `1 - pnorm(0.9183)`), so there is no evidence to support the politician's claim.

As for one-sample test for proportions, we can also use the R function **prop.test** to compare two proportions:

```
> prop.test(x=c(40,50),n=c(70,100),alternative="greater",correct=F)


        2-sample test for equality of proportions without continuity
        correction

data:  c(40, 50) out of c(70, 100)
X-squared = 0.8433, df = 1, p-value = 0.1792
alternative hypothesis: greater
95 percent confidence interval:
 -0.05596576  1.00000000
sample estimates:
   prop 1    prop 2
0.5714286 0.5000000
```

Note that, as expected, we obtain the same P-value. However, observed test statistic here is the square of the one we used before ($0.9183^2 = 0.8433$). The function provides also the sample proportion $\widehat{p}_X$ and $\widehat{p}_Y$. ■

CHAPTER 8

REGRESSION

This chapter gives an introduction to simple and multiple linear regression. We present how to get confidence and prediction intervals for new observations. We discuss model validation with a study of residuals.

8.1 Introduction

Francis Galton observed in an article in 1889 that the heights of adult offspring are, on the whole, more "average" than the heights of their parents. Galton interpreted this as a degenerative phenomenon, using the term *regression* to indicate this "return to mediocrity". Karl Pearson continued Galton's original work and conducted comprehensive studies comparing various relationships between members of the same family. Figure 8.1 depicts the measurements of the heights of 1078 fathers and their adult sons (one son per father).

The average height of the fathers was 67 inches, and of the sons 68 inches. Because sons are on average 1 inch taller than the fathers we could try to "explain" the height of the son by taking the height of his father and adding 1 inch. However, the line $y = x + 1$ (dashed) does not seem to predict the height of the sons as accurately as the solid line in Figure 8.1. This line has a slope less than 1, and demonstrates Galton's "regression" effect. For example, if a father is 5% taller than average, then his son will be on the whole *less* than 5% taller than average.

Regression analysis is about finding relationships between a *response* variable which we would like to "explain" via one or more *explanatory* variables. In regression, the response variable is usually a *quantitative* (numerical) variable.

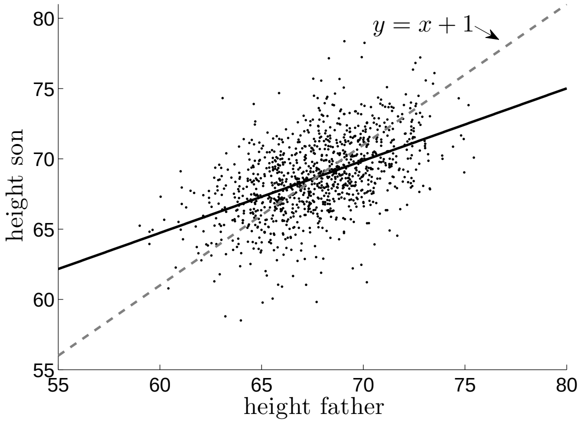


Figure 8.1: A scatter plot of heights from Pearson's data.

8.2 Simple Linear Regression

The most basic regression model involves a linear relationship between the response and a single explanatory variable. As in Pearson's height data, we have measurements $(x_1, y_1), \ldots, (x_n, y_n)$ that lie approximately on a straight line. It is assumed that these measurements are outcomes of vectors $(x_1, Y_1), \ldots, (x_n, Y_n)$, where, for each *deterministic* explanatory variable x_i, the response variable Y_i is a *random* variable with

$$\mathbb{E}(Y_i) = \beta_0 + \beta_1 x_i, \quad i = 1, \ldots, n \tag{8.1}$$

for certain *unknown* parameters β_0 and β_1. The (unknown) line

$$y = \beta_0 + \beta_1 x \tag{8.2}$$

is called the **regression line**. To completely specify the model, we need to designate the joint distribution of $Y_1, \ldots, Y_n$. The most common linear regression model is given next. The adjective "simple" refers to the fact that a *single* explanatory variable is used to explain the response.

Definition 8.1: Simple Linear Regression

The response data $Y_1, \ldots, Y_n$ depend on explanatory variables $x_1, \ldots, x_n$ via the linear relationship

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i\ , \quad i = 1, \ldots, n\ , \tag{8.3}$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.

This formulation makes it even more obvious that we view the responses as random variables which would lie exactly on the regression line, were it not for some "disturbance" or "error" term (represented by the $\{\varepsilon_i\}$).

To make things more concrete let us consider the student survey dataset stored in the dataset `studentsurvey.csv`, which can be found on Blackboard. Suppose we wish to investigate the relation between the shoe size (explanatory variable) and the height (response variable) of a person.

First we load the data:

```
> rm(list=ls())   # good practice to clear the workspace
> survey = read.csv("studentsurvey.csv")
> names(survey) # check the names

  [1] "sex"         "laptop"      "height"      "weight"
 [5] "pulserate"   "forearm"     "shoe"        "sleep"
 [9] "eyes"        "piercings"   "attractive" "country"
[13] "pizza"       "residence"   "work"        "grade"
[17] "life"        "superpower" "kiss"        "handed"
```

In the notation of Definition 8.1, x_i denotes the i-th shoe size in cm (stored in `shoe`) and y_i denotes the corresponding height in cm (stored in `height`). For the pairs $(x_1, Y_1), \ldots, (x_n, Y_n)$, we assume model (8.3). Note that the model has three unknown parameters: β_0, β_1, and σ^2. What can we say about the model parameters on the basis of the observed data $(x_1, y_1), \ldots, (x_n, y_n)$?

A first step in the analysis is to draw a scatterplot of the points (height versus shoe size). Here we use the `xyplot` function from the `lattice` library:

```
> library(lattice) # load the lattice library
> xyplot(height~shoe, data=survey)
```

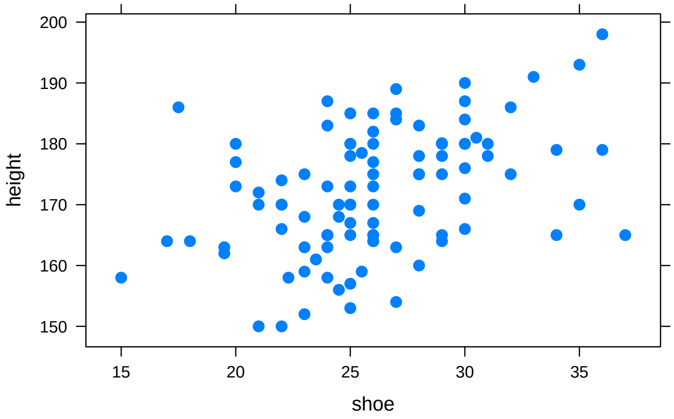


Figure 8.2: Scatter plot of height (in cm) against shoe size (in cm).

We observe a slight increase in the height as the shoe size increases, although this relationship is not very clear.

8.2.1 Estimation for Linear Regression

Obviously, we do not know the true regression line $y = \beta_0 + \beta_1 x$, but we can try to find a line $y = \widehat{\beta}_0 + \widehat{\beta}_1 x$ that best "fits" the data. Here $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are estimates for the unknown intercept β_0 and slope β_1. Note that by substituting x_i for x, we find that the corresponding y-value is $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$. For each i, the difference $e_i = y_i - \widehat{y}_i$ is called a **residual error**, or simply **residual**. There are various measures for "best fit", but a very convenient one is minimize the Sum of the Squared residual Errors, SSE $= \sum_{i=1}^n e_i^2$. This gives the following *least-squares* criterion:

$$\text{minimize SSE} . \tag{8.4}$$

The solution is given in the next theorem.

Theorem 8.1: Least-squares Estimates

The values for $\widehat{\beta}_1$ and $\widehat{\beta}_0$ that minimize the least-squares criterion (8.4) are:

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2} \tag{8.5}$$

$$\widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1 \overline{x} . \tag{8.6}$$

Proof: We seek to minimize the function $g(a,b) = \text{SSE} = \sum_{i=1}^n (y_i - a - bx_i)^2$ with respect to a and b. To find the optimal a and b, we take the derivative of SSE with respect to a, b and set it equal to 0. This leads to two linear equations:

$$\frac{\partial \sum_{i=1}^n (y_i - a - bx_i)^2}{\partial a} = -2 \sum_{i=1}^n (y_i - a - bx_i) = 0$$

and

$$\frac{\partial \sum_{i=1}^n (y_i - a - bx_i)^2}{\partial b} = -2 \sum_{i=1}^n x_i (y_i - a - bx_i) = 0 .$$

From the first equation, we find $\overline{y} - a - b\overline{x} = 0$ and then $a = \overline{y} - b\overline{x}$. We put this expression for a in the second equation and get (omitting the factor -2):

$$\begin{aligned}\sum_{i=1}^n x_i (y_i - a - bx_i) &= \sum_{i=1}^n x_i \left(y_i - \overline{y} + b\overline{x} - bx_i\right) \\ &= \sum_{i=1}^n x_i y_i - \overline{y} \sum_{i=1}^n x_i + b\left(n\overline{x}^2 - \sum_{i=1}^n x_i^2\right) .\end{aligned}$$

Since this expression has to be 0, we can solve for b to obtain

$$b = \frac{\sum_{i=1}^n x_i y_i - n\overline{xy}}{\sum_{i=1}^n x_i^2 - n\overline{x}^2} = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^n (x_i - \overline{x})^2} \; .$$

Replacing a with $\widehat{\beta}_0$ and b with $\widehat{\beta}_1$, we have completed the proof. □

If we replace in (8.5) and (8.6) the values y_i and $\overline{y}$ with the *random variables* Y_i and $\overline{Y}$, then we obtain the *estimators* of β_1 and β_0. Think of these as the parameters for the line of best fit that we would obtain if we would carry out the experiment *tomorrow*.

⚠ When dealing with parameters from the Greek alphabet, such as β, it is customary in the statistics literature —and you might better get used to it— to use the *same* notation (Greek letter) for the estimate and the corresponding estimator, both indicated by the "hat" notation: $\widehat{\beta}$. Whether $\widehat{\beta}$ is to be interpreted as random (estimator) or fixed (estimate) should be clear from the context.

Since both estimators $\widehat{\beta}_0$ and $\widehat{\beta}_1$ are linear combinations of normal random variables, their distributions are again normal (Theorem 5.6). Moreover, it is not too difficult to calculate the corresponding expectations and variances. These are summarized in the next theorem. We leave the proofs for second and third year courses. ☞ 90

Theorem 8.2: Properties of the Estimators $\widehat{\beta}_0$ and $\widehat{\beta}_1$

Both $\widehat{\beta}_0$ and $\widehat{\beta}_1$ have a normal distribution. Their expected values are

$$\mathbb{E}(\widehat{\beta}_0) = \beta_0 \quad \text{and} \quad \mathbb{E}(\widehat{\beta}_1) = \beta_1 \; , \tag{8.7}$$

so both are *unbiased* estimators. Their variances are

$$\mathrm{Var}(\widehat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^n (x_i - \overline{x})^2} \right) \tag{8.8}$$

and

$$\mathrm{Var}(\widehat{\beta}_1) = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \overline{x})^2} \; . \tag{8.9}$$

To estimate the unknown σ^2, we can reason as follows: For each x_i, σ^2 is the variance of the true error $\varepsilon_i = Y_i - (\beta_0 + \beta_1 x_i)$, where the $\{\varepsilon_i\}$ are iid with a $\mathcal{N}(0, \sigma^2)$ distribution. So, if we knew the true errors $\{\varepsilon_i\}$, we could estimate σ^2 via their sample variance, which is $\sum_{i=1}^n \varepsilon_i^2/(n-1)$. Unfortunately, we do not know the true errors, because the parameters β_0 and β_1 are unknown. However, we could replace the true error ε_i with the residual error $e_i = y_i - (\widehat{\beta}_0 + \widehat{\beta}_1 x_i)$. Our estimator for σ^2 is then $\frac{1}{n-1} \sum e_i^2$.

It turns out we need to slightly scale this to $\frac{1}{n-2}\sum e_i^2$ to get an *unbiased* estimator for σ^2. This is sometimes called the **mean squared error** (MSE) or also **residual squared error** (RSE).

8.2.2 Hypothesis Testing for Linear Regression

It is of interest to test whether the slope β_1 is 0. If this is the case, then there is no association between the response and the explanatory variable. There are two approaches that we could use to construct a good test statistic.

One approach is to utilize the fact that, by Theorem 8.2, the estimator $\widehat{\beta}_1$ has a normal distribution with expectation 0 and variance $\sigma^2 / \sum_{i=1}^n (x_i - \overline{x})^2$, under H_0. Hence, similar to the construction of the test statistic for the one-sample normal model, we could use the test statistic

$$T = \frac{\widehat{\beta}_1 \sqrt{\sum (x_i - \overline{x})^2}}{\sqrt{\text{MSE}}} \,. \tag{8.10}$$

It can be shown that under H_0, T has a Student's t distribution with $n - 2$ degrees of freedom. A similar test statistic can be used to test whether β_0 is 0, but this is less relevant.

8.2.3 Using the Computer

The relevant R function to do linear regression is `lm` (abbreviation of *linear model*). The main parameter of this function is the usual R formula — in this case `height~shoe`.

```
> model1 = lm(height ~ shoe, data = survey)
> model1


Call:
lm(formula = height ~ shoe, data = survey)

Coefficients:
(Intercept)         shoe
    145.778        1.005
```

The above R output gives the least squares estimates of β_0 and β_1. For the above example, we get $\widehat{\beta}_0 = 145.778$ and $\widehat{\beta}_1 = 1.005$. We can now draw the regression line on the scatter plot, using:

```
> xyplot(height~shoe,data=survey,type = c("p","r"),
        cex=1.2,pch=16, col.line="red", lwd =3)
```

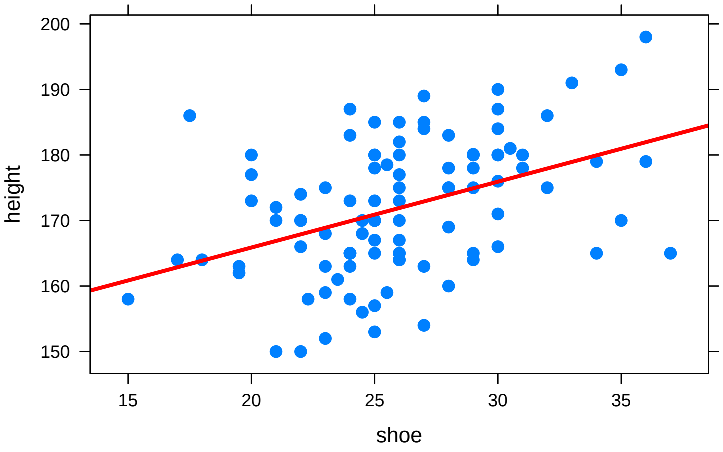


Figure 8.3: Scatter plot of height (in cm) against shoe size (in cm), with the fitted line.

The function `lm` performs a complete analysis of the linear model. The function `summary` provides a summary of the calculations:

```
> sumr1 = summary(model1)
> sumr1


Call:
lm(formula = height ~ shoe, data = survey)

Residuals:
     Min       1Q   Median       3Q      Max 
-18.9073  -6.1465   0.1096   6.3626  22.6384 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 145.7776     5.7629  25.296  < 2e-16 ***
shoe          1.0048     0.2178   4.613 1.2e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 9.299 on 98 degrees of freedom
Multiple R-squared: 0.1784,        Adjusted R-squared:  0.17 
F-statistic: 21.28 on 1 and 98 DF,  p-value: 1.199e-05
```

Here is a description of the information in this output.

- `Call:` formula used in the model.

- `Residuals:` summary information for the residuals $e_i = y_i - \widehat{y}_i$.
- `Coefficients:` this table has four columns:
    - `Estimate` gives the estimates of the parameters of the regression line;
    - `Std. Error` gives the estimate of the standard deviation of the estimators of the regression line. These are the square roots of the variances in (8.8) and (8.9);
    - `t value` gives the realization of Student's test statistic associated with the hypotheses $H_0 : \beta_i = 0$ and $H_1 : \beta_i \neq 0$, $i = 0, 1$. In particular, the t-value for the slope corresponds to the outcome of T in (8.10);
    - `Pr(>|t|)` gives the P-value of Student's test (two-sided test).
- `Signif. codes:` codes for significance levels.
- `Residual standard error:` the estimate $\sqrt{\text{MSE}}$ of σ, and the associated degree of freedom $n - 2$.

We will explain the `R-squared` and `F-statistic` values in the next chapter on analysis of variance. For now, it suffices to know that the `R-squared` value indicates how well the linear model fits the data, in the sense that it gives the fraction of variance that is explained by the regression model, compared with the "default" model where the height data follow a normal distribution with some μ and σ^2. The closer this value is to 1, the better the fit.

In this case, the fraction of variance explained by the regression (R^2) is 0.1784. Only 17.8 % of the variability of the height is explained by shoe size. We therefore need to add to the model other explanatory variables (multiple linear regression), to increase the model's predictive power.

The estimate of the slope indicates that the difference between the average height of students whose shoe size is different by one cm is 1.0048 cm.

You can access all the numerical values from the summary object directly. First check which names are available

```
> names(sumr1)

[1] "call"          "terms"        "residuals"     "coefficients"
[5] "aliased"       "sigma"        "df"            "r.squared"
[9] "adj.r.squared" "fstatistic"   "cov.unscaled"
```

Then access the values via via the dollar ($) construction. For example, the following code extracts the P-value for the slope:

```
> sumr1$coefficients[2,4]

[1] 1.1994e-05
```

8.2.4 Confidence and Prediction Intervals for a New Value

Linear regression is most useful when we wish to *predict* how a new response variable will behave, on the basis of a new explanatory variable x. For example, it may be difficult to measure the response variable, but by knowing the estimated regression line and the value for x, we will have a reasonably good idea what Y or the expected value of Y is going to be.

Thus, consider a new x and assume $Y \sim \mathcal{N}(\beta_0 + \beta_1 x, \sigma^2)$. First we're going to look at the *expected* value of Y; that is, $y = \mathbb{E}(Y) = \beta_0 + \beta_1 x$. Since we do not know β_0 and β_1, we do not know (and will never know) the expected response y. However, we can *estimate* y via

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1\, x.$$

It is also possible to give a (numerical) confidence interval for y:

$$\widehat{y} \pm c\,\sqrt{\text{MSE}}\,\sqrt{\frac{1}{n} + \frac{(x - \overline{x})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}\;,$$

where c is a constant that depends on n and the confidence level α (it is the $1 - \alpha/2$ quantile of the t_{n-2} distribution). Recall that MSE estimates the variance σ^2 of the model error.

If we wish to predict the value of Y (not just its expectation) for a given value of x, then we have *two* sources of variation:

1. Y itself is a random variable, which is normally distributed with variance σ^2,

2. we don't know the expectation $\beta_0 + \beta_1 x$ of Y. Estimating this number on the basis of previous observations $Y_1, \ldots, Y_n$ brings another source of variation.

Thus, instead of a confidence interval for $\beta_0 + \beta_1 x$ we need a *prediction interval* for a new response Y. A random prediction interval is an interval (U, V) where U and V depend only on the (random) data and are chosen such that $\mathbb{P}(U \leqslant Y \leqslant V) = 1 - \alpha$. A corresponding outcome (u, v) is a numerical prediction interval. Using Theorem 8.2, we can find the following numerical prediction interval:

$$\widehat{y} \pm c\,\sqrt{\text{MSE}}\,\sqrt{1 + \frac{1}{n} + \frac{(x - \overline{x})^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}}\;,$$

where c is the same constant as for the confidence interval above.

The `R` function to find the prediction interval and confidence interval for a new value x is **`predict`**. For example, for our student survey data, suppose we wish to find a confidence interval for the expected height for a shoe size $x = 30$. This is found as follows:

```
> predict(model1,data.frame(shoe=30),interval="confidence")
```

```
       fit      lwr      upr
1 175.9217 173.4261 178.4172
```

We can also predict the weight of a person whose shoe size is 30 to lie in the following interval, with probability 0.95.

```
> predict(model1,data.frame(shoe=30),interval="prediction")


        fit      lwr      upr
1 175.9217 157.2999 194.5434
```

Note that the prediction interval is much wider.

8.2.5 Validation of Assumptions

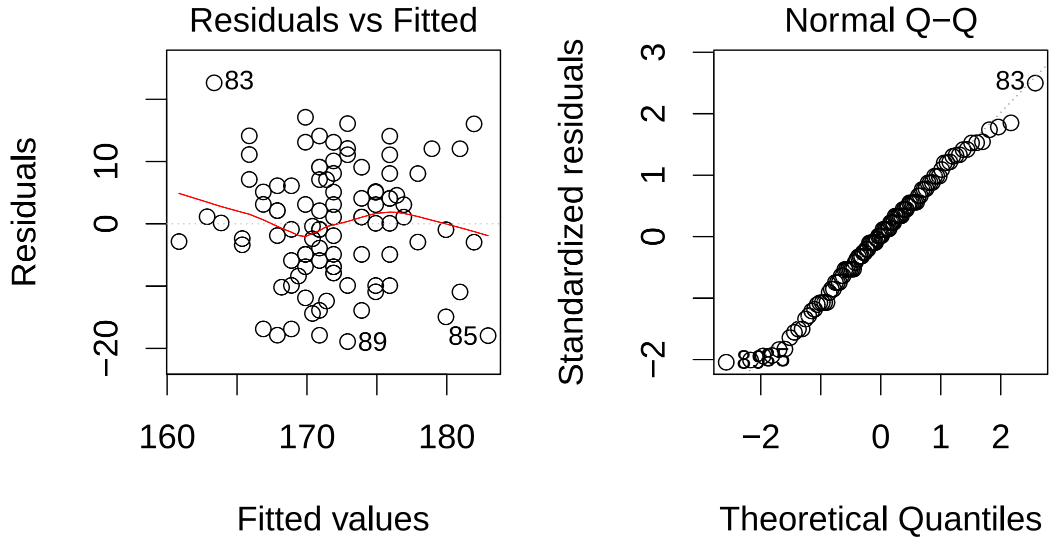
Recall the assumptions on the error terms $\{\varepsilon_i\}$ of a linear regression model:

1. The error terms are *independent* of each other.
2. The error terms are *normally distributed*.
3. The error terms have a *constant variance* and expectation 0.

Although we do not know each error term $\varepsilon_i = Y_i - (\beta_0 + \beta_1 x_i)$, the observed residual $e_i = y_i - (\widehat{\beta_0} + \widehat{\beta_1} x_i)$ will be an estimate of it. Hence, we can do an analysis of residuals to examine whether the underlying assumptions of the linear regression model are verified. The independence assumption is the most difficult to test. If data are collected over time, then one could plot the residuals against time to see if there is any correlation. Various plots for verifying the model assumptions are:

- *Plotting a histogram or density plot of the residuals to check for normality.*
- *Producing a quantile–quantile plot (qq-plot) of the standardized residuals.* Here, the sample quantiles of the standardized residuals are plotted against the theoretical quantiles of the standard normal distribution. Under the normality assumption the points should lie approximately on the straight line with slope 1.
- *Plotting the residuals e_i as a function of the predicted values $\widehat{y_i}$.* When all the model assumptions are verified, residuals and predicted values are uncorrelated. This plot should have no particular structure. This plot also gives indications on the validity on the linearity assumption, and on the constant variance of the errors. The plot of e_i against $\widehat{y_i}$ should show a uniform spread of the residuals around the zero horizontal line.

```
> par(mfrow=c(1,2))
> plot(model1,1:2)
```



Examining the residuals as a function of predicted values, we see that the residuals are correctly spread, symmetrical about the x axis: the conditions of the model seem valid.

Note that the instruction `plot(model1)` can draw four plots; some of these are for outlier detection.

8.3 Multiple Linear Regression

A linear regression model that contains more than one explanatory variable is called a *multiple linear regression model*.

Definition 8.2: Multiple Linear Regression Model

In a **multiple linear regression model** the response data $Y_1, \ldots, Y_n$ depend on d-dimensional explanatory variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, with $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^\top$, via the linear relationship

$$Y_i = \beta_0 + \beta_1 \, x_{i1} + \cdots + \beta_d \, x_{id} + \varepsilon_i \ , \qquad i = 1, \ldots, n \ , \tag{8.11}$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.

To put across the idea, let us go back to the student survey data set `survey`. Instead of "explaining" the student height via their shoe size, we could include other quantitative explanatory variables, such as the weight (stored in `weight`). The corresponding `R` formula for this model would be

$$\texttt{height} \sim \texttt{shoe} + \texttt{weight}$$

meaning that each random height `Height` satisfies

$$\texttt{Height} = \beta_0 + \beta_1\texttt{shoe} + \beta_2\texttt{weight} + \varepsilon,$$

where ε is a normally distributed error term with mean 0 and variance σ^2. The model has thus 4 parameters.

Before analysing the model we present a scatter plot of all pairs of variables, using the `R` function `pairs`.

```
> pairs(height ~ shoe + weight, data = survey)
```

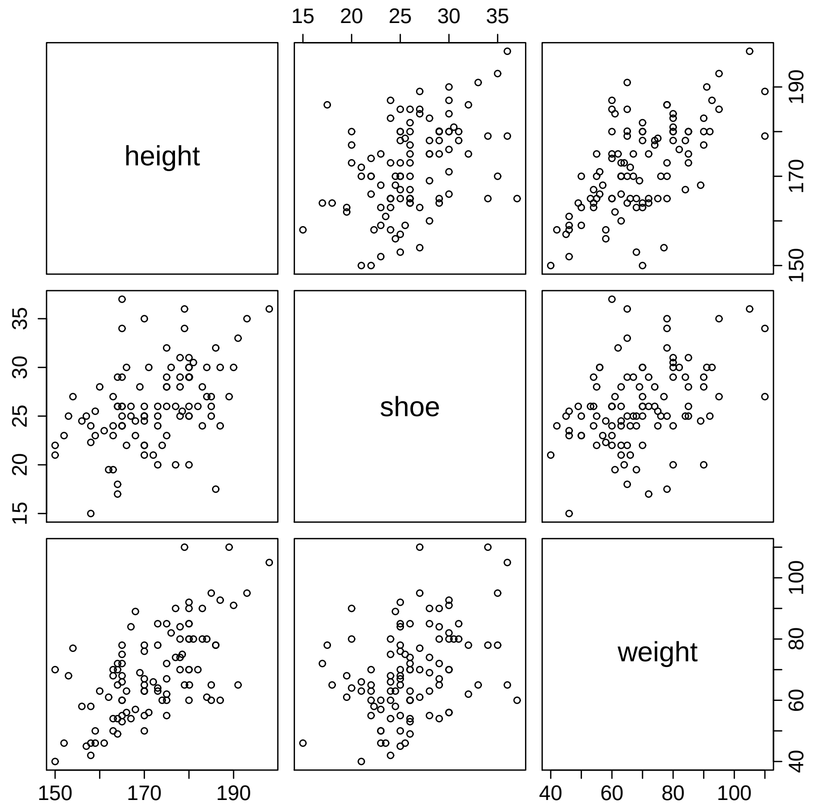


Figure 8.4: Scatter plots for all pairs of variables.

8.3.1 Analysis of the Model

As for simple linear regression, the model can be analysed using the function `lm`:

```
> model2 = lm(height~ shoe + weight)
> summary(model2)
```

```
Call:
lm(formula = height ~ shoe + weight)

Residuals:
     Min       1Q   Median       3Q      Max 
-21.4193  -4.0596   0.1891   4.8364  19.5371 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 132.2677     5.2473  25.207  < 2e-16 ***
shoe          0.5304     0.1962   2.703   0.0081 ** 
weight        0.3744     0.0572   6.546 2.82e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.785 on 97 degrees of freedom
Multiple R-squared: 0.4301,       Adjusted R-squared: 0.4184 
F-statistic: 36.61 on 2 and 97 DF,  p-value: 1.429e-12
```

The results returned by **summary** are presented in the same fashion as for simple linear regression. The individual Student tests indicate that:

- shoe size is linearly associated with student height, after adjusting for weight (P-value = 0.0081). At the same weight, an increase of one cm in shoe size corresponds to an increase of 0.53 cm of average student height;

- weight is linearly associated with student height, after adjusting for shoe size (P-value = 2.82×10^{-09}). At the same shoe size, an increase of one kg of the weight corresponds to an increase of 0.3744 cm of average student height.

Confidence intervals for regression parameters can be found with **confint**:

```
> confint(model2)
```

```
                  2.5 %      97.5 %
(Intercept) 121.8533072 142.6821199
shoe          0.1410087   0.9198251
weight        0.2608887   0.4879514
```

Confidence and prediction intervals can be obtained via the **predict** function. Suppose we wish to predict the height of a person with shoe size 30 and weight 75 kg. A confidence interval for the expected height is obtained as follows (notice that we can abbreviate "confidence" to "conf").

```
> predict(model2,data.frame(shoe=30,weight=75),interval="conf")
```

```
       fit      lwr      upr
1 176.2617 174.1698 178.3536
```

Similarly, the corresponding prediction interval is found as follows.

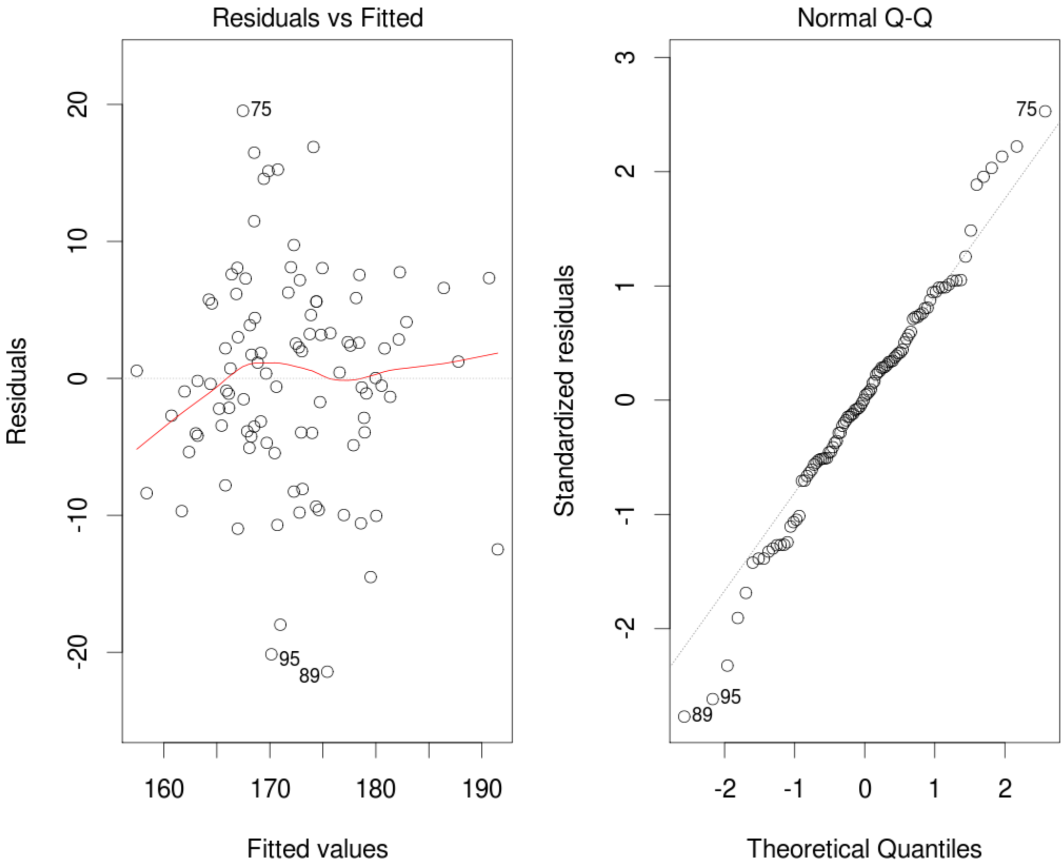
```
> predict(model2,data.frame(shoe=30,weight=75),interval="pred")

       fit      lwr      upr
1 176.2617 160.6706 191.8528
```

8.3.2 Validation of Assumptions

We check the assumptions of this multivariate model by investigating the residuals plots.

```
> par(mfrow=c(1,2))
> plot(model2,1:2)
```



The residuals are correctly spread, symmetrical about the x axis: the conditions of the model seem valid. Moreover, the QQ-plot indicates no extreme departure from the normality.

CHAPTER 9

Analysis of Variance

We present an introduction to the analysis of grouped data via an analysis of variance (ANOVA). We discuss ANOVA models with one factor and two factors, with or without interaction. You will learn how to estimate parameters of the models and how to carry out hypothesis tests using R.

9.1 Introduction

Analysis of variance (ANOVA) is used to study the relationship between a *quantitative* variable of interest and one or several *categorical* variables. As in regression, the variable of interest is called the **response variable** (in some fields confusingly called "dependent variable") and the other variables are called **explanatory variables** (or "independent variables"). Recall (see Section 2.2) that categorical variables take values in a *finite* number of categories, such as yes/no, green/blue/brown, and male/female. In R, such variables are called **factors**. They often arise in designed experiments: controlled statistical experiments in which the aim is to assess how a response variable is affected by one or more factors tested at several **levels**. A typical example is an agricultural experiment where one wishes to investigate how the yield of a food crop depends on two factors: (1) *pesticide*, at two levels (yes and no), and (2) *fertilizer*, at three levels (low, medium, and high). Treatment pairs were assigned to plots via randomization. Table 9.1 gives an example of data that is produced in such an experiment. Here three responses (crop yield) are collected from each of the six different combinations of levels.

☞ 23

Table 9.1: Crop yield data

Crop Yield	Pesticide	Fertilizer
3.23	No	Low
3.20	No	Low
3.16	No	Low
2.99	No	Medium
2.85	No	Medium
2.77	No	Medium
5.72	No	High
5.77	No	High
5.62	No	High
6.78	Yes	Low
6.73	Yes	Low
6.79	Yes	Low
9.07	Yes	Medium
9.09	Yes	Medium
8.86	Yes	Medium
8.12	Yes	High
8.04	Yes	High
8.31	Yes	High

Note that the pesticide factor only has two levels. To investigate whether using pesticide is effective (produces increased crop yield) we could simple carry out a two-sample t-test; see Section 7.5. Let us carry out the usual steps for a statistical test here: ☞ 121

1. The model is a two-sample normal model. Let $X_1, \ldots, X_9 \sim_{\text{iid}} \mathcal{N}(\mu_1, \sigma^2)$ be the crop yields without pesticide and $Y_1, \ldots, Y_9 \sim_{\text{iid}} \mathcal{N}(\mu_2, \sigma^2)$ be the crop yields with pesticide; all variables are assumed to be independent of each other. Note that we assumed here equal variances for both groups; you may verify graphically that this assumption is reasonable.

2. H_0 is the hypothesis that there is no difference between the groups; that is, $\mu_1 = \mu_2$. The alternative hypothesis is that there is a difference: $\mu_1 \neq \mu_2$.

3. As a test statistic we use the T statistic given in (7.3). ☞ 123

4. We find the outcome $t = -7.2993$ (e.g., using `t.test`)

5. The P-value is $1.783 \cdot 10^{-6}$, which is very small.

6. We therefore fail to accept the null-hypothesis. There is very strong evidence that using pesticide makes a difference.

Note that the above t-test does not tell us whether the pesticide was *successful* (that is, gives a higher average yield). Think how you would assess this.

What if we consider instead whether fertilizer “explains” crop yield. For this factor we have three levels: low, medium, and high. So a two-sample t-test does no longer work. Nevertheless, we would like to make a similar analysis as above. Steps 1 and 2 are easily adapted:

1. The model is a three-sample normal model. Let $Y_1, Y_2, Y_3, Y_{10}, Y_{11}, Y_{12} \sim_{\text{iid}} \mathcal{N}(\mu_1, \sigma^2)$ be the crop yields with low fertilizer, $Y_4, Y_5, Y_6, Y_{13}, Y_{14}, Y_{15} \sim_{\text{iid}} \mathcal{N}(\mu_2, \sigma^2)$ be the crop yields with medium fertilizer, and $Y_7, Y_8, Y_9, Y_{16}, Y_{17}, Y_{18} \sim_{\text{iid}} \mathcal{N}(\mu_3, \sigma^2)$ be the crop yield with high fertilizer. We assume equal variances for all three groups, and that all variables are independent of each other.

2. H_0 is the hypothesis that there is no difference between the groups; that is, $\mu_1 = \mu_2 = \mu_3$. The alternative hypothesis is that there is a difference.

The question is now how to formulate a test statistic (a function of the data) that makes it easy to distinguish between the null and alternative hypothesis. This is where ANOVA comes. It will allow us to compare the means of any number of levels within a factor. Moreover, we will be able to explain the response variable using multiple factors at the same time. For example, how does the crop yield depend on both pesticide and fertilizer.

The following code reads the data and produces Figure 9.1. We have used a few tricks in this code that you might find useful to know. Firstly, we plotted the levels in the order from "Low" to "High". This is done in Line 5. Without this line, the levels would be taken in alphabetical order, starting with "High". Secondly, we indicated what the Pesticide level was for the data in each Fertilizer group. This is done by specifying the plotting characters (numbers) in Line 6, for each data point, and in Line 7, we use these characters via the " pch = " option. Note that the **rep** function replicates numbers or strings; in this case nine 4s (producing crosses) and nine 1s (producing circles).

```
1 crop = read.csv("cropyield.csv")
2 library(lattice)
3 # reorder the levels from low to high
4 crop$Fertilizer = factor(crop$Fertilizer,
5                           levels = c("Low", "Medium", "High"))
6 chs = c(rep(4,9),rep(1,9))  # define two groups of plotting characters
7 stripplot(Yield~Fertilizer,pch=chs,cex=1.5,data=crop,xlab="Fertilizer")
8 #stripplot(Yield~Fertilizer,groups=Pesticide,cex=1.5,data=crop,
9 #          xlab="Fertilizer")
```

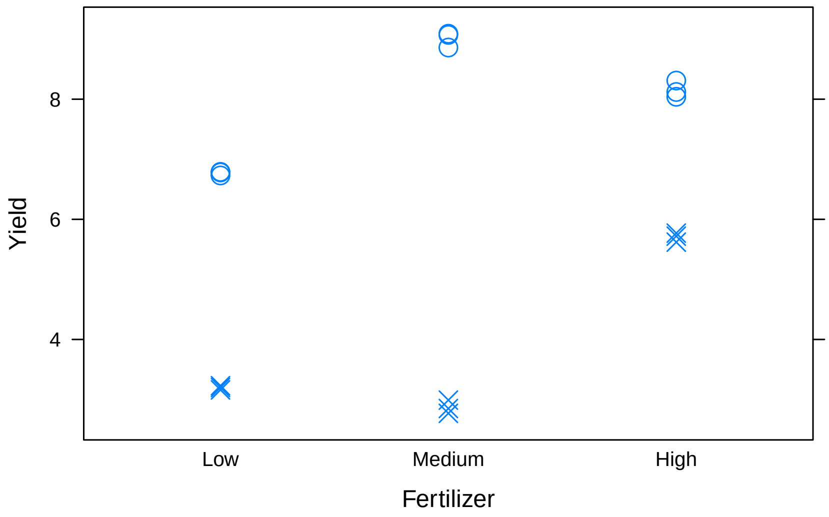


Figure 9.1: Strip plot of crop yield against fertilizer level. Whether pesticide was applied is also indicated (circle for Yes, cross for No).

9.2 Single-Factor ANOVA

We start with single-factor experiments. Think of the crop yield example where we only consider the fertilizer factor, which is applied at 3 levels (low, medium, high).

9.2.1 Model

Consider a response variable which depends on a single factor with d levels, denoted $1, \ldots, d$. Let us use the letter ℓ to indicate a level. So, $\ell \in \{1, \ldots, d\}$. Within each level ℓ there are n_ℓ independent measurements of the response variable. The total number of measurements is thus $n = n_1 + \cdots + n_d$. As in the crop yield example, think of the response data as a single column (of size n), and the factor (explanatory variable) as another column (with entries in $\{1, \ldots, d\}$). An obvious model for the data is that the $\{Y_i\}$ are assumed to be independent and normally distributed with a mean and variance which depend only on the level. Such a model is simply a d-sample generalization of ☞ 91 the two-sample normal model in Example 5.8. To be able to analyse the model via ANOVA one needs, however, the additional model assumption that the *variances are all equal*; that is, they are the same for each level. Using the *indicator* notation

$$\mathrm{I}(x = \ell) = \begin{cases} 1 & \text{if } x = \ell, \\ 0 & \text{if } x \neq \ell, \end{cases}$$

☞ 128 we can write this model in a very similar way to the regression model in Definition 8.1:

Definition 9.1: Single-Factor ANOVA Model

The response data $Y_1, \dots, Y_n$ depend on explanatory variables $x_1, \dots, x_n$ via the linear relationship

$$Y_i = \mu_1 \, \mathrm{I}(x_i = 1) + \cdots + \mu_d \, \mathrm{I}(x_i = d) + \varepsilon_i \,, \quad i = 1, \dots, n \,, \tag{9.1}$$

where $\varepsilon_1, \dots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.

So, for a single response Y with explanatory factor x, we have

$$Y = \mu_1 \, \mathrm{I}(x = 1) + \cdots + \mu_d \, \mathrm{I}(x = d) + \varepsilon \,. \tag{9.2}$$

Instead of (9.2), one often sees the "factor effects" formulation

$$Y = \mu_1 + \alpha_2 \, \mathrm{I}(x = 2) + \cdots + \alpha_d \, \mathrm{I}(x = d) + \varepsilon, \tag{9.3}$$

where μ_1 is the mean effect of the *reference* level (level 1 in this case) and $\alpha_\ell = \mu_\ell - \mu_1$ is the *incremental effect* of level ℓ, relative to the reference level. The latter approach is used in R.

9.2.2 Estimation

The model (9.2) has $d + 1$ unknown parameters: $\mu_1, \dots, \mu_d$, and σ^2. Each μ_ℓ can be estimated exactly as for the 1-sample normal model, by only taking into account the data in level i. In particular, the estimator of μ_ℓ is the sample mean within the ℓ-th level:

$$\widehat{\mu}_\ell = \overline{Y}_\ell = \frac{1}{n_\ell} \sum_{i=1}^{n} Y_i \, \mathrm{I}(x_i = \ell), \quad i = 1, \dots, d \,.$$

To estimate σ^2, we should utilize the fact that all $\{Y_i\}$ are assumed to have the same variance σ^2. So, as in the two-sample normal model case, we should *pool* our data and not just calculate, say, the sample variance of the first level only. The model (9.1) assumes that the errors $\{\varepsilon_i\}$ are independent and normally distributed, with a constant variance σ^2. If we knew each $\varepsilon_i = Y_i - \sum_{\ell=1}^{d} \mu_\ell \, \mathrm{I}(x_i = \ell)$, we could just take the sample variance $\sum_i \varepsilon_i^2/(n-1)$ to estimate σ^2 unbiasedly. Unfortunately, we do not know the $\{\mu_\ell\}$. However, we can estimate each μ_ℓ with $\overline{Y}_\ell$. This suggests that, similar to the regression case, we replace the unknown true errors ε_i with the **residual errors** (or simply residuals), which are here given by

$$e_i = Y_i - \sum_{\ell=1}^{d} \overline{Y}_\ell \, \mathrm{I}(x_i = \ell).$$

A sensible estimator for σ^2 is therefore: $\frac{\sum_{i=1}^{n} e_i^2}{n-1}$. However, this turns out not to be unbiased. An unbiased estimator is obtained by dividing the sum of the squared residual errors $\sum_i e_i^2$ (abbreviated as SSE) by $n-d$ instead of $n-1$. We thus obtain the unbiased estimator

$$\widehat{\sigma^2} = \frac{\text{SSE}}{n-d}. \tag{9.4}$$

The latter is also called the **mean squared residual error** (MSE).

9.2.3 Hypothesis Testing

The typical aim is to test whether the d levels have the same means; that is, to test the hypothesis

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_d$$

versus the alternative that this is not the case (at least two different means).

Note that we compare here two models: Under H_0 we simply have the standard 1-sample normal model for data, and under H_1 we have the single-factor ANOVA model. To assess which model is more appropriate, we could compare the variability of the data in the simpler model to the variability of the data in the second, more complex, model. More precisely, we would like to compare the variances σ^2 of the error terms for both models. Let's call them σ_1^2 and σ_2^2 to distinguish between them. Because the first model is a special case of the second, $\sigma_1^2 > \sigma_2^2$ if H_1 is true, and $\sigma_1^2 = \sigma_2^2$ if H_0 is true. It therefore makes sense to base the test statistic on estimators of σ_1^2 and σ_2^2. We already saw that σ_2^2 is estimated via $\text{SSE}/(n-d)$. And we can estimate σ_1^2 simply via the sample variance

$$\frac{\sum_{i=1}^n (Y_i - \overline{Y})^2}{n-1} = \frac{\text{SST}}{n-1}, \tag{9.5}$$

where $\overline{Y}$ denotes the sample mean of all $\{Y_i\}$, and $\text{SST} = \sum_{i=1}^n (Y_i - \overline{Y})^2$ is called the **total sum of squares**.

So, a sensible test statistic could be based on a simple function of SST and SSE whose distribution under H_0 can be computed. The actual test statistic that is used in this situation is

$$F = \frac{(\text{SST} - \text{SSE})/(d-1)}{\text{SSE}/(n-d)}, \tag{9.6}$$

where the difference SST − SSE is again a "sum of squares":

$$\text{SST} - \text{SSE} = \sum_{\ell=1}^{d} \sum_{k=1}^{n_i} (\overline{Y}_\ell - \overline{Y})^2 \ . \tag{9.7}$$

Let us denote this by SSF (Sum of Squares due to the Factor). It measures the variability *between* the different levels of the factor. If we further abbreviate $\text{SSF}/(d-1)$ to MSF (mean square factor) and $\text{SSE}/(n-d)$ to MSE (mean square error), then we can write our test statistic as

$$F = \frac{\text{MSF}}{\text{MSE}} \ .$$

The test statistic F thus compares the variability *between* levels with the variability *within* the levels. We reject H_0 for large values of F (right-onesided test). To actually carry out the test we need to know the distribution of F under H_0, which is given in the following theorem, the proof of which is beyond a 1-st year course.

Theorem 9.1

Under H_0, $F = \text{MSF}/\text{MSE}$ has an $\mathsf{F}(d-1, n-d)$ distribution.

This **F-distribution** is named after R.A. Fisher — one of the founders of modern statistics. So, in addition to the Student's t distribution and the χ^2 distribution this is the third important distribution that appears in the study of statistics. Again, this is a *family* of distributions, this time depending on two parameters (called, as usual, *degrees of freedom*). We write $\mathsf{F}(\text{df}_1, \text{df}_2)$ for an F distribution with degrees of freedom df_1 and df_2. Figure 9.2 gives a plot of various pdfs of this family. We used a similar script as for the plotting of Figure 6.2. Here is the beginning of the script — you can work out the rest. ☞ 101

```
> curve(df(x,df1=1,df2=3),xlim=c(0,8),ylim=c(0,1.5),ylab="density")
```

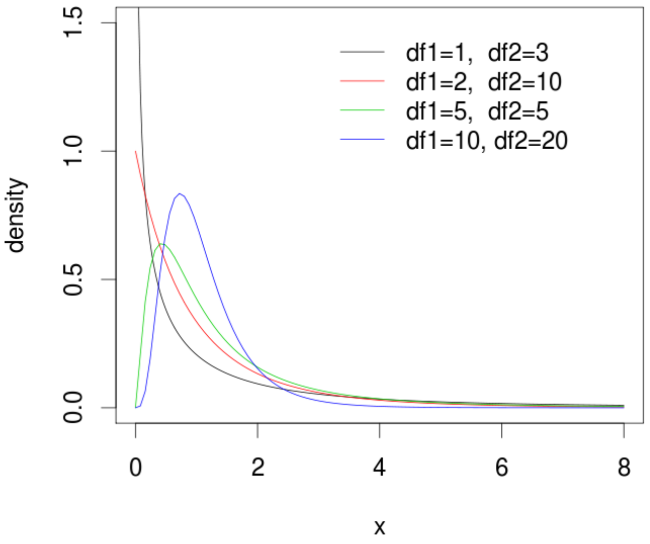


Figure 9.2: The pdfs of F distributions with various degrees of freedom (df).

It is out of the scope of this 1-st year course to discuss all the properties of the F distribution (or indeed the t and the χ^2), but the thing to remember is that it is just a probability distribution, like the normal and uniform one, and we can calculate pdfs, cdfs, and quantiles exactly as for the normal distribution, using the "d, p, q, r" construction, as in Table 4.1. ☞ 78

Fortunately, software can do all the calculations for us and summarize the results in an **ANOVA table**. For the one-factor case, it is of the form given in Table 9.2.

Table 9.2: One-factor ANOVA table. f is the outcome of the F statistic.

Source of Variation	DF	SS	Mean Squares	F	$\mathbb{P}[F > f]$
Treatment	$d - 1$	SSF	MSF	$\frac{\text{MSF}}{\text{MSE}}$	P-value
Error	$n - d$	SSE	MSE		
Total	$n - 1$	SST			

9.2.4 Worked Example

Five treatments ($T_1, \ldots, T_5$) against cold sore, including one placebo, were randomly assigned to thirty patients (six patients per treatment group). For each patient, the time (in days) for the cold sore to completely heal was measured. The results are given in Table 9.3.

Table 9.3: Cold sore healing times for 5 different treatments. T_1 is a placebo treatment.

T_1	T_2	T_3	T_4	T_5
5	4	6	7	9
8	6	4	4	3
7	6	4	6	5
7	3	5	6	7
10	5	4	3	7
8	6	3	5	6

The aim here is to compare the mean healing times. The times in the placebo column seem a little higher. But is this due to chance or is there a real difference. To answer this question, let us first load the data into R.

```
> x = data.frame(Placebo=c(5,8,7,7,10,8),T2=c(4,6,6,3,5,6),
+ T3=c(6,4,4,5,4,3),T4=c(7,4,6,6,3,5),T5=c(9,3,5,7,7,6))
```

The first important point to note is that while Table 9.3 (and the data frame `x`) is a perfectly normal table (and data frame) it is *in the wrong format* for an ANOVA study. Remember (see Chapter 2) that the measurements (the healing times) must be in a single column. In this case we should have a table with only two columns (apart from the index column): one for the response variable (healing time) and one for the factor (treatment). The factor has here 5 levels ($T_1, \ldots, T_5$). An example of a correctly formated table is Table 9.1.

We need to first “stack” the data using the `stack` function. This creates a new data frame with only two columns: one for the healing times and the other for the factor (at levels $T_1, \dots, T_5$). The default names for these columns are `values` and `ind`. We rename them to `times` and `treatment`.

```
> coldsore = stack(x)
> names(coldsore)  = c("times", "treatment")
```

The second important point is that both columns in the reformated data frame `coldsore` now have the correct type (check with `str(coldsore)`): the response is a quantitative variable (numerical) and the treatment is a categorical variable (factor) at five levels.

We can do a brief descriptive analysis, giving a data summary for the healing times within each of the factor levels. In `R` this can be done conveniently via the function `tapply`, which applies a function to a table.

```
> tapply(coldsore$times,coldsore$treatment,summary)
```

This applies the `summary` function to the vector `times`, grouped into `treatment` levels. The output is as follows.

```
$Placebo
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    5.0     7.0     7.5     7.5     8.0    10.0
$T2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    4.25    5.50    5.00    6.00    6.00
$T3
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   4.000   4.000   4.333   4.750   6.000
$T4
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   4.250   5.500   5.167   6.000   7.000
$T5
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   5.250   6.500   6.167   7.000   9.000
```

In particular, the level means (the $\bar{y}_\ell, \ell = 1, \dots, 5$) are given in the fourth column.

A boxplot of `times` versus `treatment` gives more information:

```
> library(lattice)
> bwplot(times~treatment, data = coldsore, xlab="treatment")
```

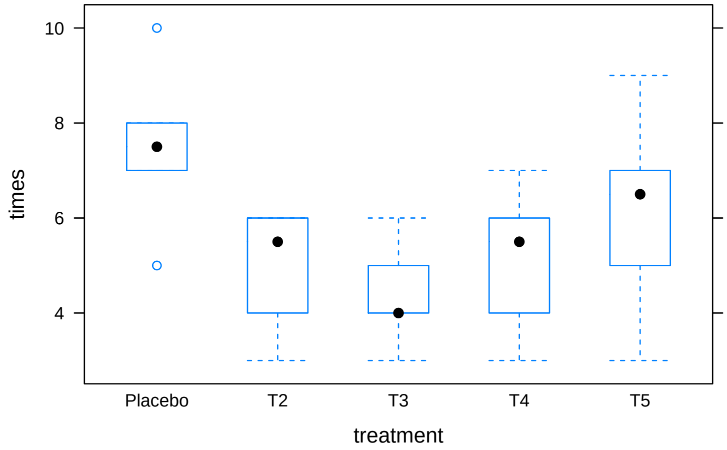


Figure 9.3: Box plot of healing times for each treatment.

Using a 1-factor ANOVA model, we wish to test the hypothesis H_0 that all treatment levels have the same means versus the alternative that this is not the case. Our test statistic is $F = \text{MSF}/\text{MSE}$, which, if H_0 is true, we know has an F distribution; see Theorem 9.2.3. In this case $d = 5$ and $n = 30$, so F has an $\mathsf{F}(4, 25)$ distribution under H_0. The next step is to evaluate the outcome f of F based on the observed data, and then to calculate the P-value. Since we have a right-onesided test (we reject H_0 for large values of F), the P-value is $\mathbb{P}(F > f)$, where $F \sim \mathsf{F}(4, 25)$. Fortunately, R can do all these calculations for us, using for instance the function **aov**. All we need to do is specify the R formula.

```
> my.aov = aov(times~treatment, data = coldsore)
> summary(my.aov)

              Df Sum Sq Mean Sq F value   Pr(>F)
treatment      4 36.467  9.1167   3.896 0.01359 *
Residuals     25 58.500  2.3400
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The values listed are the parameters (degrees of freedom, Df) for the F distribution (4 and 25), the sum of squares of the treatment SSF = 36.467 and the residuals SSE = 58.500, the corresponding mean squares MSF = 9.1167 and MSE = 2.3400 and, finally, the outcome of the test statistic $f = 3.896$, with corresponding P-value 0.01359, which is quite small. There is thus fairly strong evidence to believe that the treatments have an effect.

Validation of the Assumptions

The ANOVA model (9.1) assumes that the errors $\{\varepsilon_i\}$ are independent and normally distributed with a constant variance. As in the linear regression model, independence is difficult to check. In the context of an experiment, independence needs to be ensured by using an appropriate experimental design (e.g., via randomization). We can verify the other model assumptions by investigating the residuals. If the model is correct, the residuals should behave as independent random variables from a $\mathcal{N}(0, \sigma^2)$ distribution, for some σ^2.

The assumptions of the model can be inspected graphically using the following commands.

```
> plot(my.aov)
```

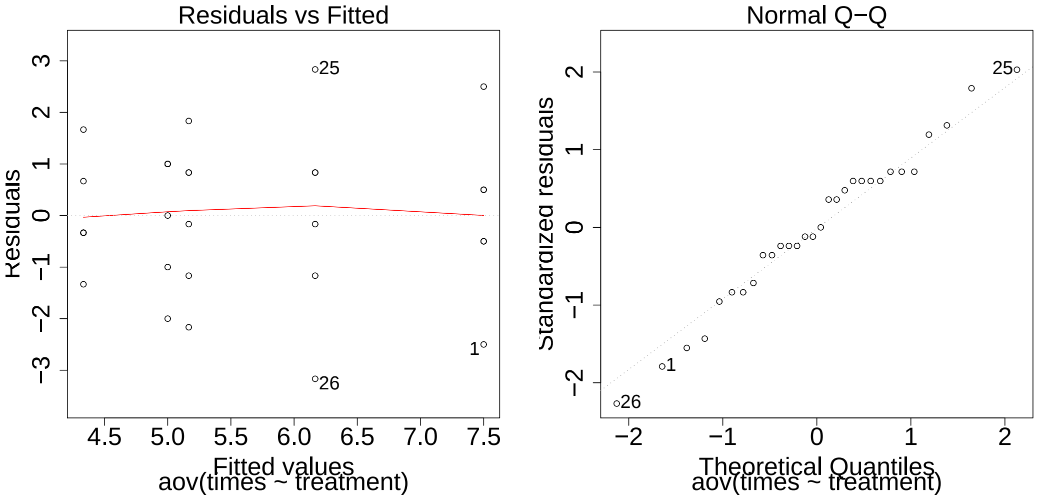


Figure 9.4: Analysing the residuals in single-factor ANOVA

R actually returns four diagnostic plots, but we have listed only two in Figure 9.4. Examining the residuals as a function of predicted values, the residuals are correctly spread, symmetrical about the x-axis: the conditions of the model (i.e., zero mean and constant variance) seem valid. The normality of the residuals is indicated by the observed straight line in the qq-plot.

9.3 Two-factor ANOVA

Many designed experiments deal with responses that depend on more than one factor. Think of the crop-yield data in Table 9.1. Here we have two factors (fertilizer and pesticide). We wish to investigate if either (or both) of them have any effect on the crop yield.

9.3.1 Model

Consider a response variable with depends on two factors. Suppose Factor 1 has d_1 levels and Factor 2 has d_2 levels. Within each pair of levels (ℓ_1, ℓ_2) there are n_{ij} replications, so that the total number of observations is $n = \sum_{\ell_1=1}^{d_1} \sum_{\ell_2=1}^{d_2} n_{ij}$. A direct generalization of (9.2) gives the following model.

Definition 9.2: Two-Factor ANOVA Model

The response data $Y_1, \ldots, Y_n$ depend on the explanatory pairs $(x_{1,1}, x_{1,2})$, $\ldots, (x_{n,1}, x_{n,2})$ via the linear relationship

$$Y_i = \sum_{\ell_1=1}^{d_1} \sum_{\ell_2=2}^{d_2} \mu_{\ell_1,\ell_2}\, \mathrm{I}(x_{i,1} = \ell_1,\ x_{i,2} = \ell_2) + \varepsilon_i\ , \quad i = 1, \ldots, n\ , \tag{9.8}$$

where $\varepsilon_1, \ldots, \varepsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.

This is just saying that the response variables are independent of each other and that for each pair of explanatory variables $(x_1, x_2) = (\ell_1, \ell_2)$, the corresponding response Y has a $\mathcal{N}(\mu_{\ell_1,\ell_2}, \sigma^2)$ distribution. The model thus has $d_1 d_2 + 1$ parameters.

Note that the variances of the responses are all assumed to be the same (equal to σ^2). To obtain a "factor effects" representation, we can reparameterize the model for a single response Y with explanatory pair (x_1, x_2) as follows:

$$\begin{aligned} Y = \mu_{1,1} &+ \sum_{\ell_1=2}^{d_1} \alpha_{\ell_1}\, \mathrm{I}(x_1 = \ell_1) + \sum_{\ell_2=2}^{d_2} \beta_{\ell_2}\, \mathrm{I}(x_2 = \ell_2) \\ &+ \sum_{\ell_1=2}^{d_1} \sum_{\ell_2=2}^{d_2} \gamma_{\ell_1,\ell_2}\, \mathrm{I}(x_1 = \ell_1, x_2 = \ell_2) + \varepsilon\ . \end{aligned} \tag{9.9}$$

The parameter $\mu_{1,1}$ is the "reference" mean response, with both factors at level 1. For any explanatory pair (ℓ_1, ℓ_2) that is not the reference pair (1,1) we *add* to this reference mean response:

- an incremental effect α_{ℓ_1} due to Factor 1,
- an incremental effect β_{ℓ_2} due to Factor 2,
- an interaction effect γ_{ℓ_1,ℓ_2} due to both factors.

Notice that there are again $d_1 d_2 + 1$ parameters. The advantage of the formulation (9.9) is that we can consider "nested" models by setting some parameters to zero. For example, if no interaction terms are included, we get the model

$$Y = \mu_{1,1} + \sum_{\ell_1=2}^{d_1} \alpha_{\ell_1}\, \mathrm{I}(x_1 = \ell_1) + \sum_{\ell_2=2}^{d_2} \beta_{\ell_2}\, \mathrm{I}(x_2 = \ell_2) + \varepsilon\ . \tag{9.10}$$

The assumption that there is no interaction and Factor 2 has no effect leads to the model:

$$Y = \mu_{1,1} + \sum_{\ell_1=2}^{d_1} \alpha_{\ell_1} \, \mathrm{I}(x_1 = \ell_1) + \varepsilon \,, \tag{9.11}$$

which is a 1-factor ANOVA model. The simplest model is the default normal model, where neither of the factors has an effect:

$$Y = \mu_{1,1} + \varepsilon \,. \tag{9.12}$$

Which of these models is most appropriate can be investigated via statistical tests.

9.3.2 Estimation

For the model (9.8), a natural estimator of μ_{ℓ_1,ℓ_2} is the sample mean of all the responses at level ℓ_1 of Factor 1 and level ℓ_2 of Factor 2; that is,

$$\widehat{\mu}_{\ell_1,\ell_2} = \overline{Y}_{\ell_1,\ell_2} = \frac{1}{n_{ij}} \sum_{\ell_1=1}^{d_1} \sum_{\ell_2=2}^{d_2} Y_i \, \mathrm{I}(x_{i,1} = \ell_1, \; x_{i,2} = \ell_2).$$

For the factor effects representation (9.9) the parameters can be estimated in a similar way. The reference mean is estimated via $\overline{Y}_{1,1}$, as given above. The incremental effect α_{ℓ_1} can be estimated via $\overline{Y}_{\ell_1,\bullet} - \overline{Y}_{1,1}$, where $\overline{Y}_{\ell_1,\bullet}$ is the average of all the $\{Y_i\}$ within level ℓ_1 of Factor 1. Similarly, β_{ℓ_2} can be estimated via $\overline{Y}_{\bullet,\ell_2} - \overline{Y}_{1,1}$, where $\overline{Y}_{\bullet,\ell_2}$ is the average of all the $\{Y_i\}$ within level j of Factor 2. Finally, γ_{ℓ_1,ℓ_2} is estimated by taking the average of all responses at the level pair (ℓ_1, ℓ_2) and subtracting from this the estimates for $\mu_{1,1}$, α_{ℓ_1} and β_{ℓ_2}.

To estimate σ^2 we can reason similarly to the 1-factor case and consider the residuals $e_i = Y_i - \widehat{Y_i}$ as our best guess of the true model errors, where $\widehat{Y_i}$ is the fitted value to the i-th response. Specifically, if the i-th expanatory pair is (ℓ_1, ℓ_2), then $\widehat{Y_i} = \overline{Y}_{\ell_1,\ell_2}$. Similar to (9.5) we have the unbiased estimator ☞ 146

$$\widehat{\sigma^2} = \mathrm{MSE} = \frac{\mathrm{SSE}}{n - d_1 \, d_2} = \frac{\sum_{i=1}^{n} e_i^2}{n - d_1 \, d_2} \,.$$

9.3.3 Hypothesis Testing

The aim here is to detect

- whether Factor 1 has an effect on the response variable;
- whether Factor 2 has an effect on the response variable;
- and whether there is an interaction effect between Factors 1 and 2 on the response variable.

Following the usual steps for hypothesis testing, we need to formulate the questions above in terms of hypotheses on the model parameters. Let us take the model formulation (9.3). Remember that the null hypothesis should contain the "conservative" statement and the alternative hypothesis contains the statement that we wish to demonstrate. So, whether Factor 1 has an effect can be assessed by testing

$$H_0 : \alpha_{\ell_1} = 0 \quad \text{for all } \ell_1,$$

versus H_1: at least one α_{ℓ_1} is not zero.

Similarly, we can assess the effectiveness of Factor 2 by testing

$$H_0 : \beta_{\ell_2} = 0 \quad \text{for all } \ell_2,$$

versus H_1: at least one β_{ℓ_2} is not zero.

We can test for interaction by considering the hypothesis

$$H_0 : \gamma_{\ell_1,\ell_2} = 0 \quad \text{for all } \ell_1, \ell_2,$$

versus H_1: at least one of the γ_{ℓ_1,ℓ_2} is not zero.

Similar to the 1-factor ANOVA case we can again decompose the total sum of squares $\text{SST} = \sum_i^b (Y_i - \overline{Y})^2$ into the sum

$$\text{SST} = \text{SSF1} + \text{SSF2} + \text{SSF12} + \text{SSE},$$

where SSF1 measures the variability between the levels of Factor 1, SSF2 measures the variability between the levels of Factor 2, SSF12 measures the variability due to interaction between the factors, and SSE measures the residual variability (i.e., within the levels).

As in the 1-factor ANOVA case, the test statistics for the above hypotheses are quotients of the corresponding mean square errors, and have an F distribution with a certain number of degrees of freedom. The various quantities of interest in an ANOVA table are summarized in Table 9.4.

Table 9.4: Two-factor ANOVA table. f is the outcome of the F statistic.

Source of Variation	DF	SS	Mean Squares	F	$\mathbb{P}[F > f]$
Factor 1	$d_1 - 1$	SSF1	MSF1	$\frac{\text{MSF1}}{\text{MSE}}$	P-value
Factor 2	$d_2 - 1$	SSF2	MSF2	$\frac{\text{MSF1}}{\text{MSE}}$	P-value
Interaction	$(d_1 - 1)(d_2 - 1)$	SSF12	MSF12	$\frac{\text{MSF12}}{\text{MSE}}$	P-value
Error	$n - d_1 d_2$	SSE	MSE		
Total	$n - 1$	SST			

9.3.4 Worked Example

Consider the data in Table 9.5, representing the crop yield using four different crop treatments (e.g., strengths of fertilizer) on four different regions.

Table 9.5: Crop yield.

	Treatment		
Region	1	2	3
1	9.18, 8.26, 8.57	9.69, 8.25, 9.83	7.87, 8.91, 7.78
2	10.05, 8.92, 9.39	9.80, 10.90, 10.75	8.33, 8.18, 9.78
3	11.23, 11.11, 9.72	12.13, 12.01, 9.67	9.38, 10.10, 10.90
4	11.60, 9.83, 11.07	12.09, 10.15, 12.04	11.73, 8.86, 11.23

These data can be entered into R using the following script. The code shows also a few "tricks of the trade". The **`attach`** function makes the variables `region, fertilizer, yield` available without having to use the $ construction, such as in `yield$region`. The function **`paste`** concatenates (joins) strings, after converting numbers into strings. So, we can get the string "Region 1", for example. The function **`gl`** generates factors by specifying the pattern of their levels.

Alternatively, you could of course enter the data in a CSV file with appropriate headers, and read the file into a data frame with **`read.csv`**.

```r
yield = c(9.18, 8.26, 8.57, 10.05, 8.92, 9.39, 11.23, 11.11,
    9.72, 11.60, 9.83, 11.07, 9.69, 8.25, 9.83, 9.80, 10.90,
    10.75, 12.13, 12.01, 9.67, 12.09, 10.15, 12.04, 7.87,
    8.91, 7.78, 8.33, 8.18, 9.78, 9.38, 10.10, 10.90, 11.73,
    8.86, 11.23)
fertilizer = gl(3,12,36,labels=paste("Fertilizer",1:3))
region = gl(4,3,36,labels=paste("Region",1:4))
wheat = data.frame(yield,fertilizer,region)
```

We wish to study the effect of the type of fertilizer on the yield of the crop and whether there is a significantly different yield between the four regions. There could also be an interaction effect; for example, if a certain treatment works better in a specific region.

```
> interaction.plot(region,fertilizer,yield)     # use this
> interaction.plot(fertilizer,region,yield)     # or this
```

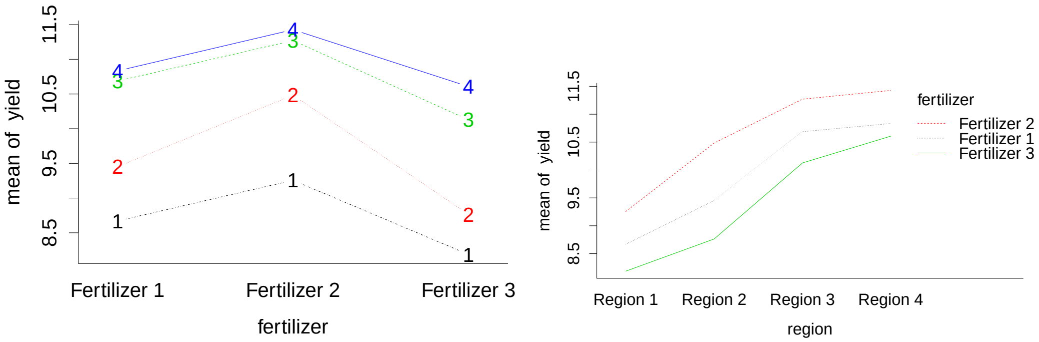


Figure 9.5: Exploration of interaction in two-way ANOVA.

These plots contain a lot of information. For example, the left figure makes it easier to investigate the Fertilizer effect. We can observe that the mean yield is always better with Fertilizer 2, whatever the region. A graph with horizontal lines would indicate no effect of the Fertilizer factor. The figure on the right may indicate an effect of the Region factor, as we can observe an increase of the mean yield from Region 1 to Region 4, whatever the Fertilizer used.

If there is no interaction between the two factors, the effect of one factor on the response variable is the same irrespective of the level of the second factor. This corresponds to observing parallel curves on both plots in figure (9.5). Indeed, the differences of the black dotted curve (Region 1) and the red dotted curve (Region 2) in the left plot represent the differential effects of the Region 2 versus Region 1 for each Fertilizer. If there is no interaction, these differences should be the same (i.e., parallel curves). Both plots in Figure 9.5 might indicate an absence of interaction as we can observe parallel curves. We will confirm it by testing the interaction effect in the next sub-section.

We plotted the two interaction plots in two different ways. To find out about the possible plotting parameters, type: `?interaction.plot` and `?par`.

ANOVA Table

Similar to the 1-factor ANOVA case, the R function **aov** provides the ANOVA table:

```
> summr = summary(aov(yield~region*fertilizer))
> summr

          fertilizer region yield

                      Df   Sum Sq Mean Sq F value     Pr(>F)
region                 3 29.3402   9.7801 11.2388 8.451e-05 ***
fertilizer             2   8.5596  4.2798   4.9182   0.01622 *
region:fertilizer      6   0.6954  0.1159   0.1332   0.99067
```

```
Residuals          24 20.8849  0.8702
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

> The formula `region*fertilizer`, used in `aov`, corresponds in fact to the formula `region+fertilizer+region:fertilizer`; i.e., the factor `region`, the factor `fertilizer`, and the interaction between these two factors.

The P-value associated with the test of interaction is not significant (P-value= 0.99). This implies that the effect of fertilizer of yield is the same whatever the region. In this case, we perform an ANOVA without an interaction term which makes it easier to interpret the principal effect. The corresponding additive model is given in (9.10): ☞ 152

$$Y = \mu_{1,1} + \sum_{\ell_1=2}^{d_1} \alpha_{\ell_1}\, \mathrm{I}(x_1 = \ell_1) + \sum_{\ell_2=2}^{d_2} \beta_{\ell_2}\, \mathrm{I}(x_2 = \ell_2) + \varepsilon\ .$$

In R, we specify this by the model `yield ~ region+fertilizer`, as in:

```
> summr2 = summary(aov(yield~region+fertilizer))
> summr2


            Df  Sum Sq Mean Sq F value    Pr(>F)
region       3 29.3402  9.7801 13.5959 8.883e-06 ***
fertilizer   2  8.5596  4.2798  5.9496  0.006664 **
Residuals   30 21.5802  0.7193
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both P-values are significant which indicate a significant effect of region and fertilizer on crop yield.

> When you have only one observation per combination of levels of the factors A and B (i.e., $n_{ij} = 1$ for all i, j), you can only estimate two-way ANOVA without interaction: `aov(yield~region+fertilizer)`.

Note that when there is interaction, we do not interpret the principal effects in the ANOVA table output. Suppose we found in our example an significant effect of the interaction term. This implies that the effect of fertilizer of yield can be different depending on the region. For example, we wish to know whether there is a fertilizer effect in Region 1. To this end, we use the function `subset`, which only uses data from a given region.

```
> fertilizer.region1 = summary(aov(yield~fertilizer
+                          ,subset=region=="Region 1"))
> fertilizer.region1
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
fertilizer   2  1.723  0.8613   1.875  0.233
Residuals    6  2.757  0.4595
```

The test in this ANOVA table corresponds to ANOVA with one factor (fertilizer) of the yield of wheat in Region 1. It does not take into account any information from data in the other regions, which would allow for a better estimation of the residual variance.

Validation of Assumptions

As in one-way ANOVA, we validate the model with a study of the residuals of the underlying linear model.

```
> plot(my.aov)
```

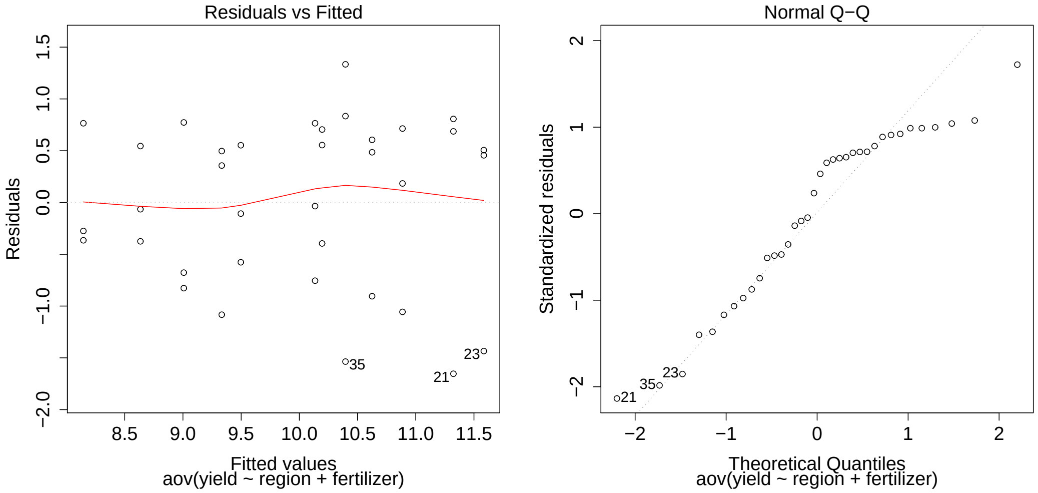


Figure 9.6: Residual analysis in two-way ANOVA.

However, if the data size is large enough for each pair of factor levels, it is better to check for normality in each subpopulation and for homoscedasticity.

9.4 Randomization and Blocking

Recall that, in the Alice experiment (Section 1.3), Alice divided her 20 friends in two groups of 10 using *randomization*. One group was given the treatment of caffeinated diet cola, and the other was the control group who received decaf diet cola. Designed experiments such as Alice's often involve some form of randomization to alleviate the effect of **nuisance** factors. These are factors that are not of primary interest to the researcher, but may influence the response. An example of a nuisance factor for crop data could be the *plant location* of the crop. ☞ 13

As an example, consider crop yield data such as in Table 9.5. Suppose we wish to test the effectiveness of 3 treatments on the crop yield. We could plant the crop in a test plot with subplots arranged in rows and columns; for example, 4 rows and 12 columns in the figures below. The question is now how to assign treatments to the test plots. It seems reasonable to allocate each treatment 16 times. In a **completely randomized design**, we allocate the treatments in such a way that the each 3 × 16 allocation is equally likely. This can be done in ℝ as follows. We first simulate a random permutation of the numbers 1,. . . , 48:

```
> x = sample(c(1:48),48)
> x


[1] 14 38 19 40 42  2 23 37 44 18 41 17 25 21  4 30  8 43
[19] 10 28 36 46 47 29 16 26 12 13  6  3 39 24 22 45  1  7
[37] 33 27 34 11 31  9 35 32 48 15 20  5
```

Then we assign treatment 1 to the first 16 subplots, treatment 2 to the next 16, and treatment 3 to the last 16. If we colour the subplots red, green, and yellow, this gives the left panel in Figure 9.7

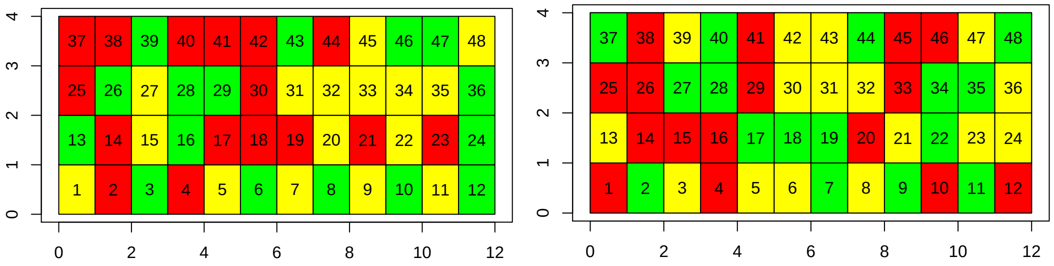


Figure 9.7: Left: completely randomized design. Right: randomized block design, with 4 treatments per block (row).

Now suppose that the soil conditions vary a lot within each column; for example the bottom row could lie at the bottom of a hill and the top row on the top of a hill. Then the soil condition of the row in which the crop was plotted could be an important factor (but a nuisance factor) in explaining the crop yield. Complete randomization as described above would alleviate the bias caused by the row soil conditions. However, note that in the left panel of Figure 9.7 rows 1 and 3 only have two treatments of type 1 (red). If the rows indeed are a factor, it would be better (less variability in the data) if we chose our design to *block* the treatments such that each block (i.e., row) has the same number of treatments. Of course we still should randomize within each block. The right panel of Figure 9.7 shows such a **randomized block design**. We made the design and figure with the following code.

```
colv = c(rep("white",48))  # a vector of colours
for (i in 1:4) {           # for each row
  x = sample(c(1:12), 12)  # random permutation 1,...,12
  p1 = x[1:4]              # column indices for treatment 1
  p2 = x[5:8]
  p3 = x[9:12]
  colv[(i-1)*12 + p1] = "red"   # assign colours to indices in row i
  colv[(i-1)*12 + p2]= "green"
  colv[(i-1)*12 + p3] = "yellow"
}
# plot the coloured rectangles
for (j in 0:3){
  for (i in 0:11 ){
    col = colv[1 + j*12 + i]   # colour of the rectangle
    rect(xleft = i, ybottom = j, xright = i+1, ytop = j+1, col = col)
    text(i+.5, j+.5, labels = 1+j*12 +i)  # numbers of the rectangles
  }
}
```

Non-random designs may be useful as well. An example is given in Figure 9.8. Note that each row has the same number of treatments, but that there also is an “even” distribution over the columns.

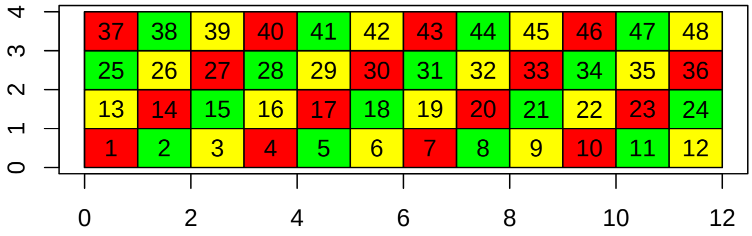


Figure 9.8: Non-random design where treatments are evenly distributed over both rows and columns.

CHAPTER 10

Linear Model

Much of modeling in applied statistics is done via the versatile class of linear models. We will give a brief introduction to such models, which requires some knowledge of linear algebra (mostly vector/matrix notation). We will learn that both linear regression and ANOVA models are special cases of linear models, so that these can be analysed in a similar way (i.e., using the `lm` and `aov` functions). In addition to estimation and hypothesis testing, we consider model selection to determine which of many competing linear models is the most descriptive of the data.

10.1 Introduction

The linear regression and ANOVA models in Chapters 9 and 8 are both special cases of a (normal) **linear model**. Let $\mathbf{Y}$ be the column vector of response data $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$.

Definition 10.1: Normal Linear Model

In a **normal linear model** the response data vector $\mathbf{Y}$ depends on a matrix $\boldsymbol{\mathcal{X}}$ of explanatory variables (called the **model matrix** or **design matrix**) via the linear relationship

$$\mathbf{Y} = \boldsymbol{\mathcal{X}}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta}$ is a vector of parameters and $\boldsymbol{\varepsilon}$ a vector of independent error terms, each $\mathcal{N}(0, \sigma^2)$ distributed.

■ **Example 10.1 (Simple Linear Regression)** For the simple linear regression model

☞ 128 (see Definition 8.1) we have

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} .$$

■

The situation for linear models in which the explanatory variables are *factors* is a little more complicated, requiring the introduction of indicator variables. We explain it with an example.

■ **Example 10.2 (One-factor ANOVA)** Consider a one-factor ANOVA model (see

☞ 144 Section 9.2) with 3 levels and 2 replications per level. Denote the responses by

$$\underbrace{Y_1, Y_2}_{\text{level 1}}, \underbrace{Y_3, Y_4}_{\text{level 2}}, \underbrace{Y_5, Y_6}_{\text{level 3}} .$$

Let μ_1 be the mean (i.e., expected) response at level — the reference level — and let α_2 and α_3 be the incremental effects of the other two levels. We can write the vector **Y** as

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \mu_1 \\ \mu_1 + \alpha_2 \\ \mu_1 + \alpha_2 \\ \mu_1 + \alpha_3 \\ \mu_1 + \alpha_3 \end{pmatrix} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix}}_{\boldsymbol{\varepsilon}} = \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}}_{\boldsymbol{X}} \underbrace{\begin{pmatrix} \mu_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}}_{\boldsymbol{\beta}} + \boldsymbol{\varepsilon} .$$

If we denote for each response Y the level by x, then we can write

$$Y = \mu_1 + \alpha_2\, \mathrm{I}(x = 2) + \alpha_3\, \mathrm{I}(x = 3) + \varepsilon, \tag{10.1}$$

where $\mathrm{I}(x = k)$ is, as in Chapter 9, an **indicator** variable that is 1 if $x = k$ and 0 otherwise. ■

In R, all data from a general linear model is assumed to be of the form

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \ldots, n , \tag{10.2}$$

where x_{ij} is the j-th explanatory variable for individual i and the errors ε_i are independent random variables such that $\mathbb{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$. In matrix form, $\mathbf{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, with

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad \boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix} \text{ and } \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Thus, the first column can always be interpreted as an “intercept” parameter. The corresponding R formula for this model would be

$$\texttt{y} \sim \texttt{x1} + \texttt{x2} + \cdots + \texttt{xp}\ .$$

Examples 10.1 and 10.2 show that it is important to treat quantitative (numbers) and qualitative (factors) explanatory variables differently. Fortunately, R automatically introduces indicator variables when the explanatory variable is a factor. We illustrate this with a few examples in which we print the model matrix, obtained via the function **model.matrix**.

In the first model variables x_1 and x_2 are both considered (by R) to be quantitative.

```
> my.dat = data.frame(y = c(10,9,4,2,4,9),
+    x1=c(7.4,1.2,3.1,4.8,2.8,6.5),x2=c(1,1,2,2,3,3))
> mod1 = lm(y~x1+x2,data = my.dat)
> print(model.matrix(mod1))
```

```
  (Intercept)  x1 x2
1           1 7.4  1
2           1 1.2  1
3           1 3.1  2
4           1 4.8  2
5           1 2.8  3
6           1 6.5  3
```

Suppose we want the second variable to be factorial instead. We can change the type as follows, using the function **factor**. Observe how this changes the model matrix.

```
> my.dat$x2 = factor(my.dat$x2)
> mod2 = lm(y~x1+x2,data=my.dat)
> print(model.matrix(mod2))
```

```
  (Intercept)  x1 x22 x23
1           1 7.4   0   0
2           1 1.2   0   0
3           1 3.1   1   0
4           1 4.8   1   0
5           1 2.8   0   1
6           1 6.5   0   1
```

In this example, the variable $x2$ is a categorical variable:

```
> my.dat$x2
```

```
[1] 1 1 2 2 3 3
Levels: 1 2 3
```

The model `mod2` is an extension of the model presented in equation (10.1):

$$Y = \mu + \beta_1 x_1 + \alpha_2 \,\mathrm{I}(x_2 = 2) + \alpha_3 \,\mathrm{I}(x_2 = 3) + \varepsilon, \tag{10.3}$$

In this model, μ is interpreted as the expected response in level 1 in a model adjusted with the x_1 variable. The parameter α_2 should be interpreted as the expected difference between the response in level 2 and the response in level 1. A similar interpretation holds for the parameter α_3.

> By default, `R` sets the incremental effect α_i of the first-named level (in alphabetical order) to zero. To impose the model constraint $\sum_i \alpha_i = 0$ for a factor `x`, use `C(x,sum)` in the `R` formula, instead of `x`.

10.2 Estimation and Hypothesis Testing

Suppose we observe a data vector $\mathbf{y}$ from a linear model $\mathbf{Y} = \boldsymbol{\mathcal{X}}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\mathcal{X}}$ is a known model matrix, and $\boldsymbol{\varepsilon}$ is a vector of iid $\mathcal{N}(0, \sigma^2)$ errors. We wish to estimate the parameter vector $\boldsymbol{\beta}$ and the model variance σ^2. We can again use a least-squares approach to estimate $\boldsymbol{\beta}$: Find $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_0, \ldots, \widehat{\beta}_p)^\top$ such that

$$\sum_{i=1}^{n} (y_i - \{\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2} + \cdots + \widehat{\beta}_p x_{ip}\})^2 \quad \text{is minimal.}$$

It can be shown that this gives the least squares estimate $\widehat{\boldsymbol{\beta}} = (\boldsymbol{\mathcal{X}}^\top \boldsymbol{\mathcal{X}})^{-1} \boldsymbol{\mathcal{X}}^\top \mathbf{y}$, where $(\boldsymbol{\mathcal{X}}^\top \boldsymbol{\mathcal{X}})^{-1}$ is the inverse of the matrix $\boldsymbol{\mathcal{X}}^\top \boldsymbol{\mathcal{X}}$. The quantity

$$e_i = y_i - \{\widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \widehat{\beta}_2 x_{i2} + \cdots + \widehat{\beta}_p x_{ip}\}$$

is the i-th residual error. Hence, the least squares criterion minimizes the sum of the squares of the residual errors, denoted SSE. To estimate σ^2 we can, as in Chapters 9 and 8, take the mean square error

$$\widehat{\sigma^2} = \mathrm{MSE} = \frac{\mathrm{SSE}}{n - (p+1)},$$

where $p + 1$ is the number of components in the vector $\boldsymbol{\beta}$.

For hypothesis testing, we can test whether certain parameters in $\boldsymbol{\beta}$ are zero or not. This can be investigated with an analysis of variance, where the residual variance of the full model is compared with the residual variance of the reduced model. The corresponding test statistics have an F distribution under the null hypothesis. The exact details are beyond a first introduction to statistics, but fortunately `R` provides all the information necessary to carry out a statistical analysis of quite complicated linear models.

If we are interested in a single parameter β_i, we also can use the same approach as the Student's test used to test if a single parameter is equal to zero or not; see (8.10). ☞ 132 In a multivariate model, the individual test statistic used in `R` is following a Student's t distribution with $n - (p + 1)$ degrees of freedom (p being the number of covariates in the model).

10.3 Using the Computer

To make things more concrete, we return to the dataset `birthwt` which we used at the end of Section 7.5. We wish to explain the child's weight at birth using various char- ☞ 121 acteristics of the mother, her family history, and her behaviour during pregnancy. The explained variable is weight at birth (quantitative variable `btw`, expressed in grammes); the explanatory variables are given below.

First we load the data:

```
> library(MASS)       # load the package MASS
> ls("package:MASS") # show all variables associated with this package
> help(birthwt)       # find information on the data set birthwt
```

Here is some information from `help(birthwt)` on the explanatory variables that we will investigate.

```
age:   mother`s age in years
lwt:   mother`s weight in lbs
race:  mother`s race (1 = white, 2 = black, 3 = other)
smoke: smoking status during pregnancy (0 = no, 1 = yes)
ptl:   no. of previous premature labors
ht:    history of hypertension (0 = no, 1 = yes)
ui:    presence of uterine irritability (0 = no, 1 = yes)
ftv:   no. of physician visits during first trimester
bwt:   birth weight in grams
```

We can see the structure of the variables via `str(birthwt)`. Check yourself that all variables are defined as *quantitative* (`int`). However, the variables `race`, `smoke`, `ht`, and `ui` should really be interpreted as *qualitative* (factors). To fix this, we could redefine them with the function `as.factor`, similar to what we did in Chapter 2. Alternatively, we could use the function `factor` in the `R` formula to let the program know that certain variables are factors. We will use the latter approach.

For *binary* response variables (that is, variables taking the values 0 or 1) it does not matter whether the variables are interpreted as factorial or numerical, as `R` will return identical summary tables for both cases.

We can now investigate all kinds of models. For example, let us see if the mother's weight, her age, her race, and whether she smokes explain the baby's birthweight.

```
> model1 = lm(bwt~lwt+age+factor(race)+smoke, data = birthwt)
> sumr1 = summary(model1)
> sumr1
```

```
Call:
lm(formula = bwt ~ lwt + age + factor(race) + smoke, data = birthwt)

Residuals:
    Min      1Q  Median      3Q     Max
-2281.9  -449.1    24.3   474.1  1746.2

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    2839.433    321.435   8.834  8.2e-16 ***
lwt               4.000      1.738   2.301  0.02249 *
age              -1.948      9.820  -0.198  0.84299
factor(race)2  -510.501    157.077  -3.250  0.00137 **
factor(race)3  -398.644    119.579  -3.334  0.00104 **
smoke          -401.720    109.241  -3.677  0.00031 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 682.1 on 183 degrees of freedom
Multiple R-squared: 0.1483,        Adjusted R-squared: 0.125
F-statistic: 6.373 on 5 and 183 DF,  p-value: 1.758e-05
```

The results returned by `summary` are presented in the same fashion as for simple linear regression. Parameter estimates are given in the column `Estimate`.

The realizations of Student's test statistics associated with the hypotheses $H_0 : \beta_i = 0$ and $H_1 : \beta_i \neq 0$ are given in column `t value`; the associated P-values are in column `Pr(>|t|)`. `Residual standard error` gives the estimate of σ and the number of associated degrees of freedom $n-p-1$. The coefficient of determination R^2 (`Multiple R-squared`) and an adjusted version (`Adjusted R-squared`) are given, as are the realization of Fisher's global test statistic (`F-statistic`) and the associated P-value.

Fisher's global F test is used to test the global joint contribution of all explanatory variables in the model for "explaining" the variability in Y. The null hypothesis is $H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$ (under the linear model, the p explanatory variables give no useful information to predict Y). The assertion of interest is H_1 : at least one of the coefficients β_j $(j = 1, 2, \ldots, p)$ is significantly different from zero (at least one of the explanatory variables is associated with Y after adjusting for the other explanatory variables).

Given the result of Fisher's global test (P-value = 1.758×10^{-5}), we can conclude that at least one of the explanatory variables is associated with child weight at birth, after adjusting for the other variables. The individual Student tests indicate that:

- mother weight is linearly associated with child weight, after adjusting for age, race and smoking status, with risk of error less than 5% (P-value = 0.022). At the same age, race status and smoking status, an increase of one pound in the mother's weight corresponds to an increase of 4 g of average child weight at birth;
- the age of the mother is not significantly linearly associated with child weight at birth when mother weight, race and smoking status are already taken into account (P-value = 0.843);
- weight at birth is significantly lower for a child born to a mother who smokes, compared to children born to non-smoker mothers of same age, race and weight, with a risk of error less than 5 % (P-value=0.00031). At the same age, race and mother weight, the child weight at birth is 401.720 g less for a smoking mother than for a non-smoking mother;
- regarding the interpretation of the variable race, we recall that the model performed used as reference the group race=1 (white). Then, the estimation of −510.501 g represents the difference of child birth weight between black mothers (`race=2`) and white mothers (reference group), and this result is significantly different from zero (P-value=0.001) in a model adjusted for mother weight, mother age and smoking status. Similarly, the difference in average weight at birth between group `race = 3` and the reference group is −398.644 g and is significantly different from zero (P-value=0.00104), adjusting for mother weight, mother age and smoking status.

Interaction

We can also include interaction terms in the model. Let us see whether there is any interaction effects between `smoke` and `age` via the model

$$\texttt{Bwt} = \beta_0 + \beta_1\texttt{age} + \beta_2\texttt{smoke} + \beta_3\texttt{age} \times \texttt{smoke} + \varepsilon\ .$$

In R this is done as follows:

```
> model2 = lm(bwt~age*smoke, data=birthwt)
> summary(model2)

Call:
lm(formula = bwt ~ age * smoke, data = birthwt)

Residuals:
     Min       1Q   Median       3Q      Max 
-2189.27  -458.46    51.46   527.26  1521.39 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2406.06     292.19   8.235 3.18e-14 ***
age            27.73      12.15   2.283   0.0236 *  
smoke         798.17     484.34   1.648   0.1011    
age:smoke     -46.57      20.45  -2.278   0.0239 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 709.3 on 185 degrees of freedom
Multiple R-squared: 0.06909,       Adjusted R-squared: 0.054 
F-statistic: 4.577 on 3 and 185 DF,  p-value: 0.004068
```

We observe that the estimate for β_3 (-46.57) is significantly different from zero (P-value = 0.024). We therefore conclude that the effect of mother age on child weight is not the same depending on the smoking status of the mother. The results on association between mother age and child weight must therefore be presented separately for the smoker and the non-smoker group. For non-smoking mothers (`smoke = 0`), the mean child weight at birth increases on average by 27.73 grams for each year of the mother's age. A confidence interval can be found as follows.

```
> confint(model2)[2,]

   2.5 %   97.5 % 
 3.76278 51.69998 
```

For smoking mothers, there seems to be a decrease in birthweight, $\widehat{\beta_1}+\widehat{\beta_3} = 27.73138-46.57191 = -18.84054$. To see if this is significant, we can again make a confidence interval and see if 0 is contained in it or not. A clever way of doing this is to create a new variable `nonsmoke = 1-smoke`, which reverses the encoding for the smokers and nonsmokers. Then, the parameter $\beta_1 + \beta_3$ in the original model is the same as the parameter β_1 in the following model

$$\texttt{Bwt} = \beta_0 + \beta_1\texttt{age} + \beta_2\texttt{nonsmoke} + \beta_3\texttt{age} \times \texttt{nonsmoke} + \varepsilon\ .$$

Hence the confidence interval can be found as follows.

```
> nonsmoke = 1 - birthwt$smoke
> confint(lm(bwt~age*nonsmoke, data=birthwt))[2,]
```

```
    2.5 %    97.5 %
-51.28712  13.60605
```

Since 0 lies in this confidence interval, the effect of age on bwt is not significant for smoking mothers.

10.4 Variable Selection

Among the large number of possible explanatory variables, we wish to select those which explain the observed responses the best. This way, we can decrease the number of predictors (giving a parsimonious model) and get good predictive power by eliminating redundant variables.

In this section, we briefly present two methods for variable selection available in R. They are illustrated on a few variables from data set birthwt. In particular, we consider the explanatory variables lwt, age, ui, smoke, ht and two recoded variables ftv1 and ptl1. We define ftv1 = 1 if there was at least one visit to a physician, and ftv1= 0 otherwise. Similarly, we define ptl1 = 1 if there is at least one preterm birth in the family history, and ptl1 = 0 otherwise.

```
> ftv1 = as.integer(birthwt$ftv>=1)
> ptl1 = as.integer(birthwt$ptl>=1)
```

10.4.1 Forward Selection

The forward selection method is an iterative method. In the first iteration we consider which feature f1 is the most significant in terms of its P-value for the F-test in the models bwt ~ f1, with f1 ∈ { wt, age,...,}. This feature is then selected into the model. In the second iteration, the feature f2 that has the smallest P-value in the models bwt ~ f1+f2 is selected, where f2 ≠ f1, and so on. Usually only features are selected that have a P-value of at most 0.05. To carry out a forward selection procedure, we could run an lm or aov analysis for each feature. For example, ANOVA analyses on the lwt and age variables yields:

```
> summary(aov(bwt~lwt,data=birthwt))
```

```
             Df   Sum Sq Mean Sq F value Pr(>F)
lwt           1  3448639 3448639   6.681 0.0105 *
Residuals   187 96521017  516155
```

and

```
> summary(aov(bwt~age,data=birthwt))


             Df   Sum Sq Mean Sq F value Pr(>F)
age           1   815483  815483   1.538  0.216
Residuals   187 99154173  530236
```

The numbers of interest are here the P-values 0.0105 and 0.216. Repeating this for all 7 variables is tedious, and fortunately we can automate this in R using the **add1** function. Watch this method in action:

```
> form1 = formula(bwt~lwt+age+ui+smoke+ht+ftv1+ptl1) #formula
> add1(lm(bwt~1),form1,test="F", data=birthwt)


Single term additions

Model:
bwt ~ 1
       Df Sum of Sq      RSS    AIC F value     Pr(F)
<none>                99969656 2492.8
lwt     1   3448639 96521017 2488.1  6.6814  0.010504 *
age     1    815483 99154173 2493.2  1.5380  0.216475
ui      1   8059031 91910625 2478.9 16.3968 7.518e-05 ***
smoke   1   3625946 96343710 2487.8  7.0378  0.008667 **
ht      1   2130425 97839231 2490.7  4.0719  0.045032 *
ftv1    1   1340387 98629269 2492.2  2.5414  0.112588
ptl1    1   4755731 95213925 2485.6  9.3402  0.002570 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We conclude that `ui` is the most significant variable to be included into the model. Next, we investigate which variable could be further added.

```
add1(lm(bwt~ui),form1,test="F", data=birthwt)


Single term additions

Model:
bwt ~ ui
       Df Sum of Sq      RSS    AIC F value   Pr(F)
<none>                91910625 2478.9
lwt     1   2074421 89836203 2476.6  4.2950 0.03960 *
age     1    478369 91432256 2479.9  0.9731 0.32518
```

```
smoke   1   2996636 88913988 2474.6  6.2687 0.01315 *
ht      1   3162595 88748030 2474.3  6.6282 0.01082 *
ftv1    1    950090 90960534 2478.9  1.9428 0.16503
ptl1    1   2832244 89078381 2475.0  5.9139 0.01597 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, `ht` is the most significant variable to be added to the model. We now look for a third possible variable:

```
> add1(lm(bwt~ui+ht),form1,test="F", data=birthwt)

Single term additions

Model:
bwt ~ ui + ht
       Df Sum of Sq      RSS    AIC F value    Pr(F)
<none>              88748030 2474.3
lwt     1   3556661 85191369 2468.5  7.7236 0.006013 **
age     1    420915 88327114 2475.4  0.8816 0.348988
smoke   1   2874044 85873986 2470.0  6.1916 0.013720 *
ftv1    1    698945 88049085 2474.8  1.4686 0.227120
ptl1    1   2678123 86069907 2470.5  5.7564 0.017422 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Thus, `lwt` is the most significant variable to be added.

```
add1(lm(bwt~ui+ht+lwt),form1,test="F", data=birthwt)

Single term additions

Model:
bwt ~ ui + ht + lwt
       Df Sum of Sq      RSS    AIC F value   Pr(F)
<none>              85191369 2468.5
age     1     97556 85093813 2470.3  0.2109 0.64657
smoke   1   2623742 82567628 2464.6  5.8469 0.01658 *
ftv1    1    510128 84681241 2469.4  1.1084 0.29380
ptl1    1   2123998 83067371 2465.8  4.7048 0.03136 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now, `smoke` is the most significant variable.

```
> add1(lm(bwt~ui+ht+lwt+smoke),form1,test="F",data=birthwt)
```

```
Single term additions

Model:
bwt ~ ui + ht + lwt + smoke
       Df Sum of Sq      RSS    AIC F value   Pr(F)
<none>              82567628 2464.6
age     1     67449 82500178 2466.5  0.1496 0.69935
ftv1    1    274353 82293275 2466.0  0.6101 0.43576
ptl1    1   1425291 81142337 2463.3  3.2145 0.07464 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No further variable is significant. The method thus stops at the model with variables: ui, ht, lwt, and smoke.

10.4.2 Backward Elimination

This time, we start with the complete model and at each step, we delete the variable with lowest value of Student's test statistic (largest P-value) in absolute value, as long as it is not significant (at a specified level α).

Watch this method in action with function **drop1** for level $\alpha = 0.05$.

```
> drop1(lm(form1),test="F",data=birthwt)
```

```
Single term deletions

Model:
bwt ~ lwt + age + ui + smoke + ht + ftv1 + ptl1
       Df Sum of Sq      RSS    AIC F value     Pr(F)
<none>              80682074 2466.2
lwt     1   2469731 83151806 2469.9  5.5405 0.0196536 *
age     1     90142 80772217 2464.5  0.2022 0.6534705
ui      1   5454284 86136359 2476.6 12.2360 0.0005899 ***
smoke   1   1658409 82340484 2468.1  3.7204 0.0553149 .
ht      1   3883249 84565324 2473.1  8.7116 0.0035808 **
ftv1    1    270077 80952151 2464.9  0.6059 0.4373584
ptl1    1   1592757 82274831 2467.9  3.5731 0.0603190 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We delete variable age. Let us see which variable should be dropped next, if any:

```
> drop1(lm(bwt~lwt+ui+smoke+ht+ftv1+ptl1),test="F",data=birthwt)
```

```
Single term deletions

Model:
bwt ~ lwt + ui + smoke + ht + ftv1 + ptl1
       Df Sum of Sq      RSS    AIC F value     Pr(F)
<none>              80772217 2464.5
lwt     1   2737552 83509769 2468.8  6.1684 0.0139097 *
ui      1   5561240 86333456 2475.0 12.5309 0.0005082 ***
smoke   1   1680651 82452868 2466.3  3.7869 0.0531944 .
ht      1   3953082 84725299 2471.5  8.9073 0.0032306 **
ftv1    1    370120 81142337 2463.3  0.8340 0.3623343
ptl1    1   1521058 82293275 2466.0  3.4273 0.0657462 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We thus delete variable `ftv1`. Continuing, we enter:

```
> drop1(lm(bwt~lwt+ui+smoke+ht+ptl1),test="F",data=birthwt)


Single term deletions

Model:
bwt ~ lwt + ui + smoke + ht + ptl1
       Df Sum of Sq      RSS    AIC F value    Pr(F)
<none>              81142337 2463.3
lwt     1   2887694 84030031 2467.9  6.5126 0.011528 *
ui      1   5787979 86930316 2474.3 13.0536 0.000391 ***
smoke   1   1925034 83067371 2465.8  4.3415 0.038583 *
ht      1   4215957 85358294 2470.9  9.5082 0.002362 **
ptl1    1   1425291 82567628 2464.6  3.2145 0.074642 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We delete variable `plt1`. The method stops at the model with variables: `ui`, `ht`, `lwt` and `smoke`.

It should be noted that different methods of automatic selection may not lead to the same choice of variables in the final model. They have the advantage of being easy to use, and of treating the question of variable selection in a systematic manner. The main drawback is that variables are included or deleted based on purely statistical criteria, without taking into account the aim of the study. This usually leads to a model which may be satisfactory from a statistical point of view, but in which the variables are not the most relevant when it comes to understanding and interpreting the data in the study.

10.5 Analysis of Residuals

We present here a few elements on analysis of residuals. Suppose for the `birthwt` data set we ended up with the model represented by the following `R` formula.

$$\texttt{bwt} \sim \texttt{smoke + age + lwt + factor(race) + ui + ht + smoke:age}\,.$$

☞ 162 It is good to review what the actual model looks like, in terms of (10.2).

$$\texttt{Bwt} = \beta_0+\beta_1\texttt{smoke}+\beta_2\texttt{age}+\beta_3\texttt{lwt}+\beta_4\texttt{race2}+\beta_5\texttt{race3}+\beta_6\texttt{ui}+\beta_7\texttt{ht}+\beta_8\texttt{smoke}\times\texttt{age}+\varepsilon.$$

The following `R` code checks various model assumptions.

```
> finalmodel=lm(bwt~smoke+age+lwt+factor(race)+ui+ht+smoke:age,data=birthwt)
> par(mfrow=c(1:2))
> plot(finalmodel,1:2,col.smooth="red")
```

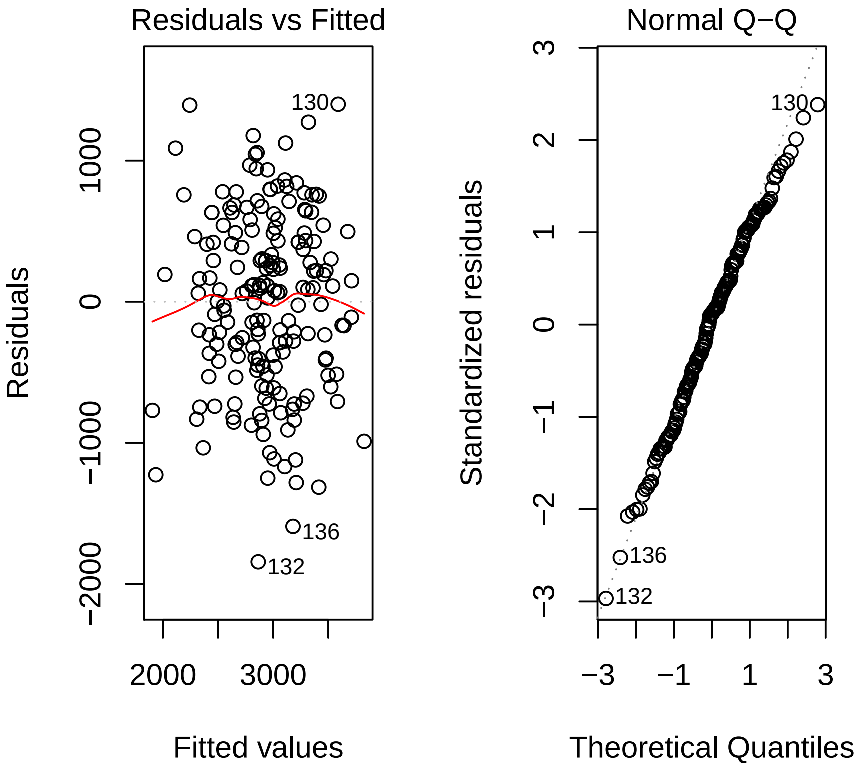


Figure 10.1: Checking the assumptions of homoscedasticity (left) and normality (right).

It can also be useful to plot the residuals as a function of each explanatory variable, as shown in Figure 10.2. This plot is useful to check whether there is a relationship between the error term and the explanatory variables. This plot is also useful to detect outliers.

```
> res = residuals(finalmodel)
> par(mfrow=c(2,3))
> plot(res~smoke);plot(res~age);plot(res~lwt)
> plot(res~race);plot(res~ui);plot(res~ht)
```

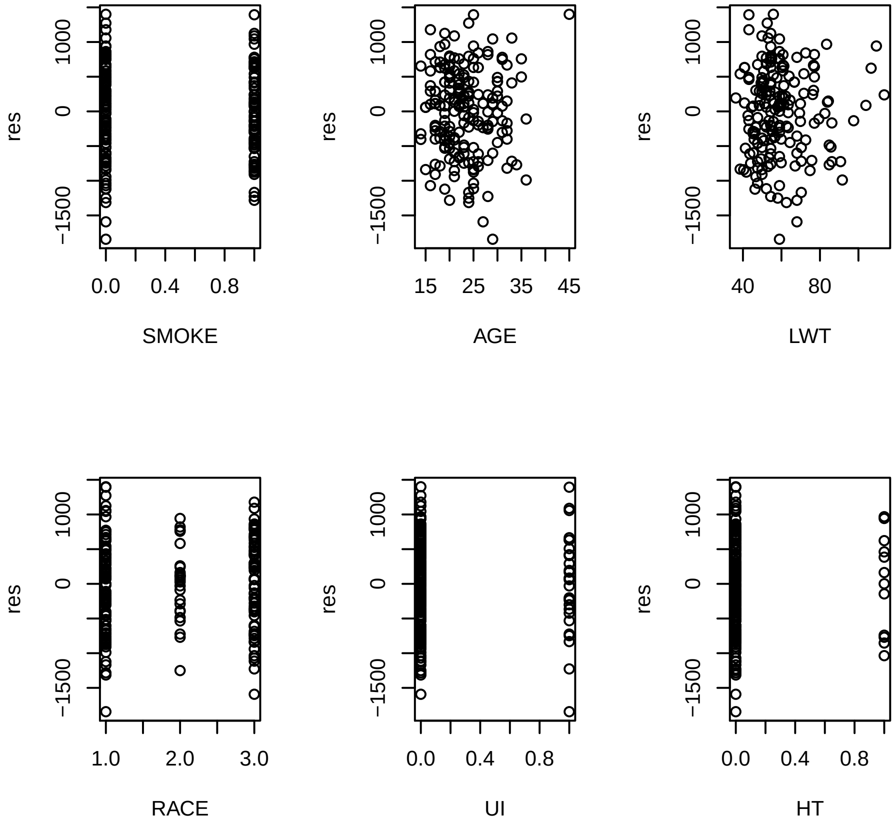


Figure 10.2: Residuals as a function of explanatory variables.

CHAPTER 11

OTHER STATISTICAL TECHNIQUES

The purpose of this chapter is to give you a taste of other useful techniques for data analysis. *Goodness of fit tests* can be used for verifying that the data comes from a described distribution; by applying *logistic regression*, one can perform regression on binary responses; and *nonparametric test* are useful when standard model assumptions, such as normality, are not valid.

You will see that the ideas behind goodness of fit tests, logistic regression and nonparametric tests naturally extend the concepts that you have already learned in previous chapters. Emphasis will be more on the practical side (how can we apply the techniques, for example in R) rather than on full mathematical proofs, which would be out of scope for a first-year course.

11.1 Multinomial Distribution

We return to the very beginning of the course where, in Example 1.1, we discussed how to simulate 100 coin tosses with a fair coin in order to estimate the probability of obtaining 60 or more Heads. Later on, we found that the number of Heads out of 100 tosses with a fair coin followed a $\mathsf{Bin}(100, 1/2)$ distribution. More generally, we found that the number X of Heads in n tosses, with a coin that has probability p of Heads, has a $\mathsf{Bin}(n, p)$ distribution. An equivalent way of simulating X is by throwing n balls in two boxes (0 and 1) with probability $1 - p$ and p, respectively, and counting how many balls are in box 1. The number in box 0 is then $n - X$. ☞ 10

We want to generalize this to throwing n balls into k boxes, numbered $1, \ldots, k$ with probabilities $p_1, \ldots, p_k$. The resulting counts $X_1, \ldots, X_k$ turn out to have a *multinomial* distribution.

Definition 11.1: Multinomial Distribution

Random variables $X_1, \ldots, X_k$ are said to have a **multinomial** distribution, with parameters $n, p_1, \ldots, p_k$ if

$$\mathbb{P}(X_1 = x_1, \ldots, X_k = x_k) = \frac{n!}{x_1!\, x_2! \cdots x_k!}\, p_1^{x_1} p_2^{x_2} \cdots p_k^{x_k}\,,$$

for all $x_1, \ldots, x_k \in \{0, 1, \ldots, n\}$ for which $x_1 + x_2 + \cdots + x_k = n$. We write $(X_1, \ldots, X_k) \sim \mathsf{Mnom}(n, p_1, \ldots, p_k)$.

■ **Example 11.1 (Army Recruits)** Suppose the IQ of army recruits is $\mathcal{N}(100, 16^2)$ distributed. Army recruits are classified as

$$\begin{aligned} \text{Class 1} &: \ \text{IQ} \leqslant 90 \\ \text{Class 2} &: \ 90 < \text{IQ} \leqslant 110 \\ \text{Class 3} &: \ \text{IQ} > 110 \end{aligned}$$

The *proportion* p_1, p_2 and p_3 of army recruits in the three categories are given by $\mathbb{P}(Y \leqslant 90) = p_1$, $\mathbb{P}(90 < Y \leqslant 110) = p_2$ and $\mathbb{P}(Y > 110) = p_3$, where $Y \sim \mathcal{N}(100, 16^2)$. It follows that we have the following proportions:

$$\begin{aligned} \text{Class 1} &: \ p_1 = 0.266 \\ \text{Class 2} &: \ p_2 = 0.468 \\ \text{Class 3} &: \ p_3 = 0.266 \end{aligned}$$

Now suppose we have 7 new recruits. What is the probability that of these 7 new recruits, two are Class 1; four are Class 2 and one is Class 3?

To answer this, let X_i be the number in class $i, i = 1, 2, 3$. Then, $(X_1, X_2, X_3) \sim \mathsf{Mnom}(7, p_1, p_2, p_3)$. Thus, it follows immediately that

$$\mathbb{P}(X_1 = 2, X_2 = 4, X_3 = 1) = \frac{7!}{2!\,4!\,1!}\, p_1^2\, p_2^4\, p_3^1 \approx 0.0957.$$

■

For the goodness-of-fit tests that we will discuss next, the following theorem is of utmost importance. The proof relies on the Central Limit Theorem and the fact that the square of a standard normal random variable has a χ_1^2 distribution.

Theorem 11.1: Multinomial Data, Known Parameters

Let $(X_1, \ldots, X_k) \sim \mathsf{Mnom}(n, p_1, \ldots, p_k)$, then the random variable

$$\sum_{i=1}^{k} \frac{(X_i - n\, p_i)^2}{n\, p_i}$$

has approximately a χ_{k-1}^2 distribution, for large n.

Remark 11.1 As a rule of thumb, we can use the approximation above provided that

$$np_i \geqslant 5, \qquad \text{for all } i.$$

Example 11.2 (Simulation Experiment) It is relatively easy to verify Theorem 11.1 for specific cases through simulation — in the same way that we verified the central limit theorem in Figure 5.4. In particular, suppose we throw $n = 100$ balls into $k = 10$ boxes, numbered 1,...,10, with equal probability. Let $X_1, \ldots, X_{10}$ be the counts. A typical count outcome is $14, 6, 13, 12, 11, 12, 5, 11, 7, 9$. The corresponding outcome of the random variable in Theorem 11.1 is 8.6 (check yourself). Figure 11.1 shows a histogram for $R = 1000$ such outcomes. We see that it matches well the density of the χ^2_9 distribution. The following code was used.

☞ 89

```
x = 1:10
n = 100
R = 1000
t = vector()   # initialize a vector t
for (i in 1:R){
  s = sample(x, size=n, replace=TRUE)
  h = hist(s, breaks=0:10) # create a histogram object
  ob  = h$counts   # contains the observed counts
  ex = 10
  t[i] = sum((ob - ex)^2/ex)
}

hist(t,breaks=30,freq = FALSE,main="")
curve(dchisq(x,df=9),xlim=c(0,30), col=2,lw=2,add = TRUE)
```

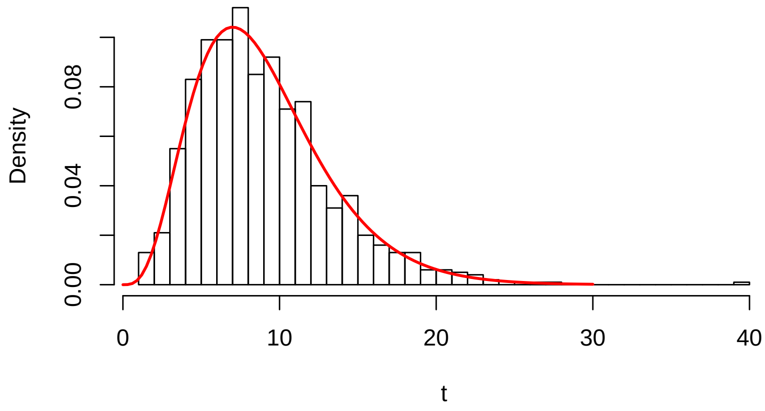


Figure 11.1: The histogram of the test statistic values closely matches the pdf of the χ^2_9 distribution (red curve).

11.2 Goodness of Fit with Known Parameters

We can use Theorem 11.1 to formulate a **goodness of fit test** for count data. Specifically, suppose we have a multinomial data

$$(X_1, \ldots, X_k) \sim \mathsf{Mnom}(n, p_1, \ldots, p_k).$$

We can test $H_0 : p_1 = \pi_1, \ldots, p_k = \pi_k$ against the alternative hypothesis that H_0 is not true by using the test statistic

$$T = \sum_{i=1}^{k} \frac{(X_i - n\,\pi_i)^2}{n\,\pi_i},$$

which, under H_0, has a χ^2_{k-1} distribution, by Theorem 11.1. We reject H_0 at the α level of significance if

$$T \geqslant q,$$

where q is the $(1-\alpha)$-quantile of the χ^2_{k-1} distribution.

Remark 11.2 We can symbolically write the test statistic as

$$T = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i},$$

where O_i is the *observed* number of observations in class i and E_i is the *expected* number of observations in class i. This form for the test statistic is found in any goodness of fit test.

■ **Example 11.3 (Frizzled Chickens)** The phenomenon of *complete dominance* predicts that progeny whose genetic component is (F, F) will be extremely frizzled, progeny with (F, f) slightly frizzled and (f, f) will be normal. According to the genetics theory, the proportions FF : Ff : ff should be 1 : 2 : 1. Out of 93 randomly selected chickens the observed frequencies phenotypes are 23 (extremely frizzled), 50 (slightly frizzled) and 20 (normal). Is this in accordance with the theory?

We can test this with a χ^2-goodness-of-fit test. Let us go through the usual steps of a statistical test (see Section 7.1). ☞ 112

1. Let X_1, X_2, X_3 be the total number of FF, Ff and ff chickens out of 93. Our model is: $(X_1, X_2, X_3) \sim \mathsf{Mnom}(93, p_1, p_2, p_3)$, for unknown p_1, p_2, p_3.

2. We want to test: $H_0 : p_1 = \frac{1}{4}, p_2 = \frac{1}{2}, p_3 = \frac{1}{4}$ against the alternative hypothesis that H_0 is not true.

3. As test statistic we use

$$T := \frac{(X_1 - 93/4)^2}{93/4} + \frac{(X_2 - 93/2)^2}{93/2} + \frac{(X_3 - 93/4)^2}{93/4}\ .$$

4. Under H_0 this has approximately a χ^2_2 distribution, see Theorem 11.1.
5. The outcome of T is
$$t = \frac{(23 - 93/4)^2}{93/4} + \frac{(50 - 93/2)^2}{93/2} + \frac{(20 - 93/4)^2}{93/4} = 0.72\ .$$
6. The P-value for this right-onesided test is $\mathbb{P}_{H_0}(T \geqslant 0.72) = 0.70$.
7. Because the value is high (0.70), we accept H_0; that is, we find no evidence to reject the theory.

■

11.3 Testing Independence

In Theorem 11.1, it is assumed that the probabilities $\{p_i\}$ are *known*. In many cases, however, these probabilities need to be *estimated* from the data, giving rise to the following modification of Theorem 11.1.

Theorem 11.2: Multinomial Data, Unknown Parameters

Let $(X_1, \ldots, X_k) \sim \mathsf{Mnom}(n, p_1, \ldots, p_k)$, where the $p_i = p_i(\theta)$ depend on an unknown r-dimensional parameter vector θ. Denoting $\widehat{p_i} = p_i(\widehat{\theta})$ the (maximum likelihood) estimate of $p_i(\theta)$, the random variable
$$\sum_{i=1}^{k} \frac{(X_i - n\,\widehat{p_i})^2}{n\,\widehat{p_i}}$$
has approximately a χ^2_{k-1-r} distribution, for large n.

Comparing this with Theorem 11.1 we see that we apparently "lose" r degrees of freedom if we have to estimate r parameters.

An important application of Theorem 11.2 occurs in a *two-way table* of counts (also called a **contingency table**), where we wish to test for an association (i.e., dependence) between the two variables. We explain the idea via a specific example first.

■ **Example 11.4 (ESP Belief)** We wish to examine whether artists differ from non-artists in Extra-Sensory Perception (ESP) belief. Table 11.1 lists the amount of belief in ESP for a group of 114 Artists and a group of 344 Non-artists. We wish to investigate whether being an artist or not is "independent" of the ESP belief (strong, moderate or not).

Table 11.1: ESP belief

	ESP belief			
	Strong	Moderate	Not	total
Artists	67	41	6	114
Non-artists	129	183	32	344
	196	224	38	458

To see that this is a type of goodness of fit situation, we need to properly formulate a model for the data and express the null and alternative hypotheses in terms of the parameters in the model.

If we ignore the row and column totals, we have a table with $r = 2$ rows and $c = 3$ columns. We can imagine the table to be filled in the following way: We randomly select 458 people and ask whether they are an artist or not and what their ESP belief is. Let (U_k, V_k) denote the response for the kth selected person, where $U_k \in \{1, 2\}$, where $(1 =$ artist, 2=non-artist), and $V_k \in \{1, 2, 3\}$, where, $(1 =$ strong belief , $2 =$ medium belief, 3= no belief). We assume that $(U_1, V_1), \dots, (U_n, V_n)$ are independent and distributed as a random vector (U, V) that can take values $(1, 1), (1, 2), (1, 3), (2, 1), (2, 2)$ and $(2, 3)$ with probabilities $p_{11}, p_{12}, \dots, p_{23}$.

Now, instead of recording all (U_k, V_k), we could instead *count* how many people are artist with a strong ESP belief, artist with a Moderate ESP belief, etc. Let $X_{i,j}$ be the *count* in row i and column j. That is, the total number of observations out of $n = 458$ that fall in "cell" (i, j). For example, the outcome of $X_{2,2}$ is 183. From the model above we have

$$(X_{11}, \dots, X_{23}) \sim \mathsf{Mnom}(n, p_{11}, \dots, p_{23}) .$$

We wish to test whether null hypothesis that the random variables U and V are *independent*. In terms of the parameters of the model, the null hypothesis can be written as

$$H_0 : \quad p_{ij} = p_i\, q_j, \quad \text{for all } i, j,$$

where p_1, p_2, q_1, q_2 and q_3 are unknown probabilities. Using Theorem 11.2, we can test the null hypothesis against the alternative hypothesis that $p_{ij} \neq p_i\, q_j$ for some i and j, by using the test statistic

$$T = \sum_{i=1}^{2} \sum_{j=1}^{3} \frac{(X_{ij} - E_{ij})^2}{E_{ij}},$$

where E_{ij} is an estimator of np_{ij}, the expected number of observations in cell (i, j). Under H_0, this is $np_i\, q_j$. The natural estimators for p_i and q_j are

$$\widehat{p_i} = \frac{\sum_{j=1}^{3} X_{ij}}{458} \quad \text{and} \quad \widehat{q_j} = \frac{\sum_{i=1}^{2} X_{ij}}{458};$$

and hence we estimate the expected count as $E_{ij} = n\widehat{p_i}\widehat{q}_j$. In other words,

$$E_{ij} = \frac{\text{total count in row } i \times \text{total count in column } j}{\text{total count}}. \tag{11.1}$$

By Theorem 11.2 the test statistic T has under H_0 approximately a χ^2_2 distribution, because the total number of parameters to be estimated is $r = 1 + 2 = 3$. We have to subtract r from the total number of classes $r \times c$ minus 1, i.e., $6 - 1 = 5$ to get number of degrees of freedom: is $5 - 3 = 2$. We reject the null hypothesis for large values of T. The various estimated counts (in brackets) are given in the table below.

Table 11.2: Observed and estimated counts for the ESP belief.

	ESP belief			
	Strong	Moderate	Not	total
Artists	67 (48.8)	41 (55.8)	6 (9.46)	114
Non-artists	129 (147)	183 (168)	32 (28.5)	344
	196	224	38	458

For example, $E_{11} = 114 \times 196/458 \approx 48.8$. It follows that the outcome of T is $t = 6.79 + 3.93 + 1.27 + 2.20 + 1.34 + 0.43 = 15.96$. The p-value is 0.00034. Hence, we strongly reject H_0. Artists indeed seem to differ from non-artists in ESP belief. ■

For the *general* contingency table, we have count data in r rows and c columns. Again, we wish to test for association between the variables. Let X_{ij} be the total number of observations (out of n) that fall in *cell* (i, j) (i.e., in the ith row and jth column). We have

$$(X_{11}, \ldots, X_{rc}) \sim \mathsf{Mnom}(n, p_{11}, \ldots, p_{rc}) .$$

If there is no association between the two variables (null hypothesis), then

$$p_{ij} = p_i\, q_j, \quad \forall i, j,$$

for some (unknown) $p_1, \ldots, p_r$ and $q_1, \ldots, q_c$. We can test this by using the test statistic

$$T = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(X_{ij} - E_{ij})^2}{E_{ij}},$$

where E_{ij} is as in (11.1). Under the null hypothesis of no association, the test statistics has approximately a χ^2_{df} distribution with the degrees of freedom parameter equal to

$$\text{df} = rc - 1 - (r - 1) - (c - 1) = (r - 1)(c - 1) .$$

And we reject H_0 for large values of T.

11.4 Logistic Regression

In the linear models in Chapters 8 – 10 the response was always modeled as a *normal* random variable. To allow different response distributions, the concept of *generalized linear models* (GLMs) can be used. In this section, we will look at a special GLM called the **logistic regression** or **logit** model. In particular, we consider the logistic regression model with a single explanatory variable. As in the simple linear regression case, it is assumed that the response variables $Y_1, \ldots, Y_n$ are independent, but now each response Y (random) depends on the explanatory variables x (fixed) according to $Y \sim \mathsf{Ber}(h(\beta_0 + \beta_1 x))$, where h here is defined as the cdf of the **logistic distribution**:

$$h(x) = \frac{1}{1 + \mathrm{e}^{-x}}.$$

Thus, for an explanatory variable x, the response Y is equal to 1 with probability

$$p = \frac{1}{1 + \mathrm{e}^{-\beta_0 - \beta_1 x}} = \frac{\mathrm{e}^{\beta_0 + \beta_1 x}}{1 + \mathrm{e}^{\beta_0 + \beta_1 x}},$$

and Y is 0 otherwise. Large values of $\beta_0 + \beta_1 x$ lead to a high probability that $Y = 1$, and small (negative) values of $\beta_0 + \beta_1 x$ cause Y to be 0 with high probability. Note that there is a linear relationship between the logarithm of the “odds” $p/(1 - p)$ and x:

$$\ln\left(\frac{p}{1 - p}\right) = \beta_0 + \beta_1 x.$$

The parameters β_0 and β_1 can be estimated from the observed data $(x_1, y_1), \ldots, (x_n, y_n)$ by maximizing the likelihood of the observed $\{y_i\}$; that is, $\mathbb{P}(Y_1 = y_1, \ldots, Y_n = y_n)$ seen as a function of β_0 and β_1. For example, if we have the sample $y_1 = 1, y_2 = 0, y_3 = 1$ and denote $p_i = (1 + \exp(-\beta_0 - \beta_1 x_i))^{-1}, i = 1, 2, 3$, then

$$\mathbb{P}(Y_1 = 1, Y_2 = 0, Y_3 = 1) = p_1(1 - p_2)p_3.$$

We can use the **glm** function in R to do this estimation/optimization for us, as explained in the following example.

■ **Example 11.5 (Logistic Regression)** The code below first simulates 100 explanatory variables, $\{x_i\}$, chosen uniformly between -1 and 1. Then, the binary response variable $\{y_i\}$ are obtained from the logistic model with parameters $\beta_0 = -3$ and $\beta_1 = 10$. The data are stored in a data frame `mydata`, consisting of two columns (one for the $\{x_i\}$ and one for the $\{y_i\}$.

```
set.seed(123)  # for reproducibility
n = 100
x = sort(2*runif(n)-1)
```

```
b0 = -3
b1 = 10
p = exp(b0+b1*x)/(1 + exp(b0+b1*x))
y = rbinom(n,size=1,p)
mydata = data.frame(x,y)
```

Figure 11.2 shows the (x_i, y_i) pairs as black circles. The true logistic curve is also shown (dashed line). Let us now try to "recover" the parameters and the curve from the observed data only, using the `glm` function:

```
> mod = glm(y~x,data=mydata,family= binomial)
> summary(mod)


Call:
glm(formula = y ~ x, family = binomial, data = mydata)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.25039  -0.05859  -0.00072   0.03185   1.95385

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   -4.520      1.439  -3.141  0.00168 **
x             14.701      4.587   3.205  0.00135 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 130.684  on 99  degrees of freedom
Residual deviance:  27.824  on 98  degrees of freedom
AIC: 31.824
```

We find estimates $\widehat{\beta}_0 = -4.520$ and $\widehat{\beta}_1 = 14.701$, which are not that far from the true parameter values. The output also allows us to construct confidence intervals. For example, an approximate 95% confidence interval for β_1 is $14.701 \pm 1.96 \times 4.587$. We can test for association between the response and explanatory variable by testing whether β_1 is 0 or not. The P-values in the output summary indicate that both β_0 and β_1 are not zero (as indeed they are not). The predicted probability $\widehat{p}$ for a new explanatory variable $x = 0.3$ is

$$\widehat{p} = \frac{1}{1 + e^{4.520 - 14.701 \times 0.3}} = 0.4726.$$

The same probability (up to rounding errors) can also be obtained via the `predict` function:

```
> ypred = predict(mod, newdata = data.frame(Height=x), type="response")
> ypred


       1
0.472675
```

In fact, we can estimate and plot the entire logistic curve via:

```
> plot(y~x,data=mydata, xlab="x") #observed responses
> lines(x,p,lw=2,lty=2)           #true logistic curve
> ypred = predict(mod, newdata = data.frame(x=x), type="response")
> lines(x, ypred,col='red',lw=2)  #predicted logistic curve
```

The estimated curve is given by the red curve in Figure 11.2. For a new explanatory variable x we can classify/predict the response as 0 or 1, if the estimate for p is respectively $< 1/2$ or $> 1/2$.

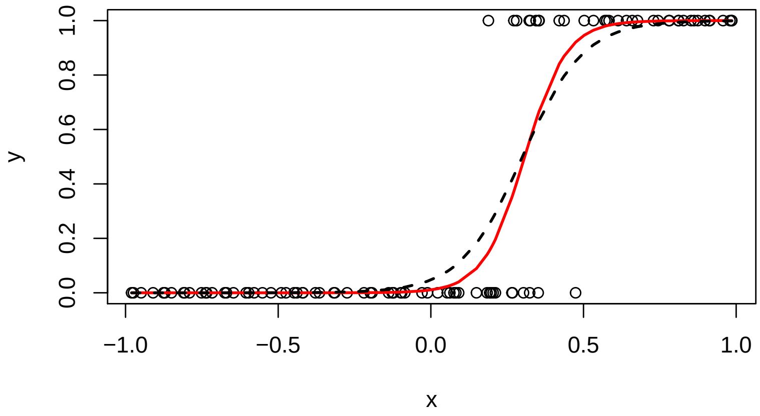


Figure 11.2: Logistic regression data (black circles), fitted curve (red), and true curve (black dashed).

■

11.5 Nonparametric Tests

For our final topic, we return one last time to Alice's caffeine experiment in Section 1.3. ☞ 13 To analyse the data we assessed how likely it would be that a random reshuffling of the caffeine and decaf groups would give a response (difference in increased pulse rate between groups) as extreme as observed (10.7 beats per minute), using a *randomization test*.

Later on, in Section 7.5, we make extra assumptions on the distribution of the data. ☞ 121 In particular, we assumed that the data in both groups came from two possibly different normal distributions, characterized by 4 unknown parameters (two expectations and two variances). This then led to the 2-sample t-test. The advantage of such a *parametric* approach is that (1) we can specify our hypotheses in terms of the parameters of the model, and (2) the distribution of our test statistic can be readily obtained; e.g., a t-distribution.

However, making assumptions about the distribution of the data is fraught with risks (e.g., incorrect calculation of P-values may lead to the wrong conclusions), especially if the assumptions are not true. The randomization test is an example of a **nonparametric test**, where we still may make assumptions about the data (e.g., independence), but we do not model the data via a specific parametric class of distributions. Nonparametric test tend to be more "robust" to outliers in the data. The downside is that they are less "powerful" than parametric test, in the sense that it is more difficult to reject the null hypothesis when it indeed should be rejected. ☞ 117

Fortunately, there are nonparametric versions of the standard tests (e.g., 1- and 2-sample t-tests) available and we will discuss a number of these. The simplest one is the **sign test**, which is a robust alternative to the 1-sample t-test. This is simply the 1-sample binomial test in disguise, as we will see in the following example.

■ **Example 11.6 (Sign Test)** In Section 7.2 we had the following data for the Decaf group in Alice's cola experiment:

Table 11.3: Changes in pulse rate for the Decaf group in Alice's cola experiment.

4	10	7	−9	5	4	5	7	6	12
+	+	+	−	+	+	+	+	+	+

To apply the 1-sample t-test, it was assumed that these data came from some normal distribution. What if we do not assume anything about the underlying distribution of the data, other than that the 10 independent observations come from the same probability distribution? Can we still conduct a test? For the null hypothesis, we could test if the median of the unknown distribution is 0 or greater than 0. And as a summary of the data we could indicate the *signs* of change in pulse rate; positive or negative (or 0, but this does not apply here). In fact, as our test statistics, we could simply take the number of positive changes — in this case 9 out of 10. The situation is now equivalent to a 1-sample binomial test for the probability p of a positive change. The null and alternative hypothesis are $H_0 : p = 1/2$ vs. $H_1 : p > 1/2$. The P-value of the test is $\mathbb{P}(X \geqslant 9)$, where $X \sim \mathsf{Bin}(10, 1/2)$. Using R:

```
> 1 - pbinom(8,size=10,prob=0.5)
```

```
[1] 0.01074219
```

There is thus strong evidence that the change in pulse rate is positive for the Decaf (control) group.

Recall that the data in Table 11.3 was the result of *paired* data (before, after), and we recorded only the changes in pulse rate (after – before). The sign test is particularly useful for such paired data, as a nonparametric alternative to the paired t-test (see Page 123). Another nonparametric alternative for paired data is the **Wilcoxon signed rank test**. The procedure is as follows:

1. First rank the absolute differences, where ties are given equal fractional ranks.
2. The test statistic S is the sum of the ranks of the positive differences.

Under the null hypothesis and for large sample size n, the test statistic has approximately a normal distribution with

$$\mathbb{E}(S) = \frac{n(n+1)}{4} \quad \text{and} \quad \text{sd}(S) = \sqrt{\frac{n(n+1)(2n+1)}{24}}.$$

Example 11.7 (Wilcoxon Signed Rank Test) Consider again the data Table 11.3, which lists the differences (changes) in pulse rate for the Decaf group. The absolute differences are exactly the same, apart from −9, which is changed to 9. The ranks for the absolute differences are given in Table 11.4. Note that there are several ties, which both get the same average rank.

Table 11.4: Ranks of absolute differences (changes) in pulse rate for the Decaf group in Alice's cola experiment.

|change|	4	10	7	9	5	4	5	7	6	12
rank	1.5	9	6.5	8	3.5	1.5	3.5	6.5	5	10

The outcome of the test statistic is $s = 1.5 + 9 + 6.5 + 3.5 + 1.5 + 3.5 + 6.5 + 5 + 10 = 47$. Under H_0, S has approximately a normal distribution with expectation 27.5 and standard deviation 9.810708, so that the P-value for this right-onesided test (we reject for large values of S) can be computed as follows:

```
n = 10
s = 1.5 + 9 + 6.5 + 3.5 + 1.5 + 3.5 + 6.5 + 5 + 10
es = n*(n+1)/4
sds = sqrt(n*(n+1)*(2*n+1)/24)
```

```
Pval = 1 - pnorm((s - es)/sds)
print(Pval)
```

```
[1] 0.02342664
```

Alternatively, using the function **wilcox.test**, we can enter:

```
> x = c(4,10,7,-9,5,4,5,7,6,12)
> wilcox.test(x,alternative = "greater")


        Wilcoxon signed rank test with continuity correction

data:  x
V = 47, p-value = 0.02616
alternative hypothesis: true location is greater than 0
```

Note that the answers differ slightly because R applies a continuity correction to the central limit theorem. R also gives a warning message (not shown here) that it cannot calculate the P-value exactly due to the ties in the data. ■

We can also replace the 2-sample t-test with a nonparametric alternative called **Wilcoxon's rank sum test**. As in the 2-sample t-test, the assumption is that we have two independent groups of data, and we wish to test if the distributions of the two groups are the same or not. The test statistic is the sum of the ranks of the first group, adjusted for ties if necessary.

Under the null hypothesis and for large sample size n, the test statistic has approximately a normal distribution with

$$\mathbb{E}S = \frac{n_1(n_1+n_2+1)}{2} \quad \text{and} \quad \mathrm{sd}(S) = \sqrt{\frac{n_1 n_2(n_1+n_2+1)}{12}}.$$

To compute $\mathbb{E}S$ above, note that under the null hypothesis all ranks are equally likely, so the rank R of one observation is a discrete uniform random variable taking values in $1,\ldots,n_1+n_2$. Its expectation is thus $\mathbb{E}R = (n_1+n_2+1)/2$, and since there are n_1 observations in the first group, we have an expected rank total of $\mathbb{E}S = n_1\mathbb{E}R = n_1(n_1+n_2+1)/2$. The variance of R and S is a bit more difficult to derive, and we leave this as an exercise. In R, the same function **wilcox.test** can be used to perform a rank sum test. However, the test statistic is here the rank sum of the first group *minus* $n_1 \times (n_1+1)/2$. This equivalent test statistic is known as the *Mann–Whitney* test statistic.

■ **Example 11.8 (Rank Sum Test)** To see how the rank sum test works, consider Alice's caffeine study one last time. Table 11.5 shows the original observations and along with their ranks.

Table 11.5: Changes in pulse rate for Alice's caffeine experiment. Ranks are given below the observations.

Caffeinated	17	22	21	16	6	−2	27	15	16	20
Rank	16	19	18	14.5	7.5	2	20	13	14.5	17
Decaf	4	10	7	−9	5	4	5	7	6	12
Rank	3.5	11	9.5	1	5.5	3.5	5.5	9.5	7.5	12

The smallest increase was −9 bpm, so it gets rank 1 while the second smallest increase was −2 bpm. The next smallest increase, 4 bpm, occurs twice and so we give both of them the average of the ranks that they would have had if they were not tied, 3 and 4.

The null hypothesis is that there is no difference between the distributions of the Caffeinated and Decaf groups. The outcome of the test statistic is:

$$s = 16 + 19 + 18 + 14.5 + 7.5 + 2 + 20 + 13 + 14.5 + 17 = 141.5.$$

If the subjects with caffeine tended to have higher increases in pulse rate then they would tend to have higher ranks and so S would tend to be bigger. The P-value is the probability of getting a value as extreme or more extreme, so here we want $\mathbb{P}(S \geqslant 141.5)$. Using the normal approximation, with $\mathbb{E}S = 105$ and $\mathrm{sd}(S) = 13.22876$, we obtain:

```
> Pval = 1 - pnorm((s - es)/sds)
> Pval
```

```
[1] 0.002897679
```

To check, we use `wilcox.test`:

```
x = c(17, 22, 21, 16, 6, -2, 27, 15, 16, 20)  # caffeinated
y = c(4, 10, 7 ,-9, 5, 4, 5, 7, 6, 12)        # decaf
wilcox.test(x,y,alternative="greater")
```

```
         Wilcoxon rank sum test with continuity correction

data:  x and y
W = 86.5, p-value = 0.003201
alternative hypothesis: true location shift is greater than 0
```

We see a similar P-value (obtained via a more accurate computation than we did by hand) and observe that the outcome of the Mann–Whitney test statistic is indeed $141.5 - 10 \times 11/2 = 86.5$. ■

APPENDIX A

R Primer

A.1 Installing R and RStudio

R is both a programming language and a work environment specifically developed for data analysis. The creators, Ross Ihaka & Robert Gentleman, wished to make available a free and open-source version of the statistical package S+, developed by AT&T Bell Laboratories in 1988. This piece of software is used to manipulate data, draw plots and perform statistical analyses of data. R works across multiple platforms (Windows, Mac, Linux) and is constantly evolving due to the contributions of a large and growing community of volunteers. Some advantages of using R:

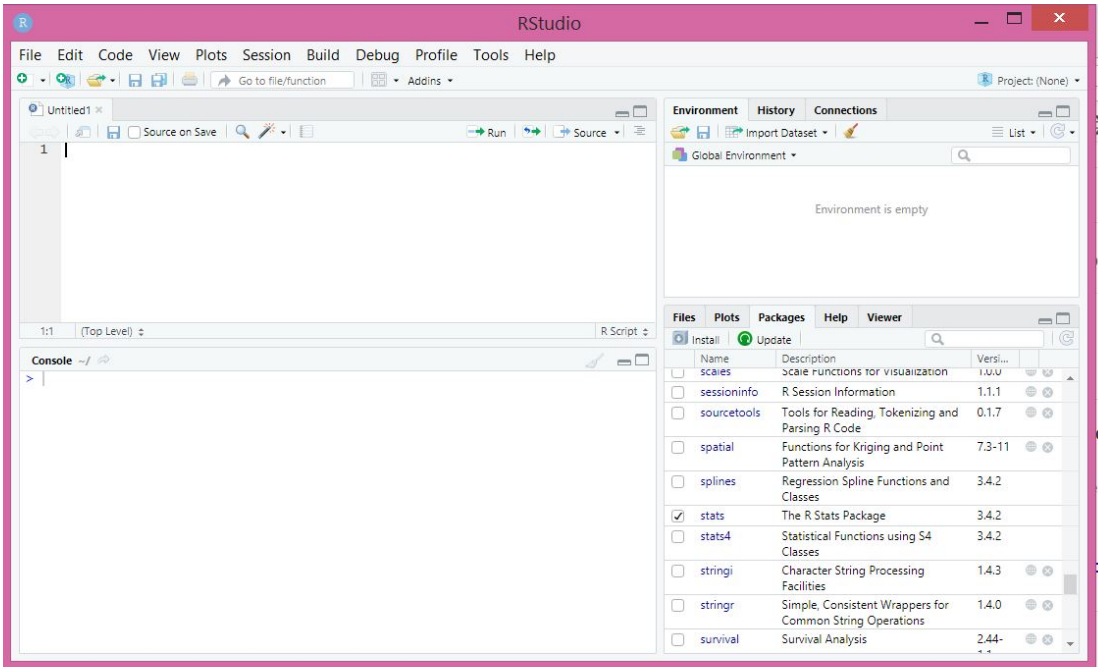
- R is *free* and *easy to install and maintain*, especially when using integrated development environments (IDEs) such as RStudio.
- R has many external *packages*: collections of functions and data tailored to certain tasks. These packages can be easily installed, e.g., via RStudio.
- R has many efficient inbuilt procedures for *statistics*, data management and visualization.
- R has an integrated and accessible *documentation* system.

Install R from the *Comprehensive R Archive Network* (CRAN): `http://cran.r-project.org`.

R's base system comes with a rudimentary Graphical User Interface. We recommend instead the use of RStudio's integrated development environment (IDE), depicted in Figure A.1.

Install RStudio from `https://www.rstudio.com/`.

This IDE comprises (customizable) windows for R programs (top-left), the R console (bottom-left), environment variables and history (top-right), and plotting and packages information (bottom-right).



	Name	Description	Versi...
□	scales	Scale Functions for Visualization	1.0.0
□	sessioninfo	R Session Information	1.1.1
□	sourcetools	Tools for Reading, Tokenizing and Parsing R Code	0.1.7
□	spatial	Functions for Kriging and Point Pattern Analysis	7.3-11
□	splines	Regression Spline Functions and Classes	3.4.2
☑	stats	The R Stats Package	3.4.2
□	stats4	Statistical Functions using S4 Classes	3.4.2
□	stringi	Character String Processing Facilities	1.4.3
□	stringr	Simple, Consistent Wrappers for Common String Operations	1.4.0
□	survival	Survival Analysis	2.44-

Figure A.1: A typical `RStudio` layout.

A.2 Learning `R`

There are many resources available to help you learn `R`. In RStudio, for example, the *Help>R Help* menu gives access to the comprehensive tutorial “An Introduction to R”, as well as to the precise “R Language Definition”. In this section we will merely give an overview of `R`. If at any point you need help, the first thing to do is consult R’s **`help`** function. For example `help("sin")` will show information about the **`sin`** function and other trigonometric functions. An Internet search is nowadays also a good alternative, which often will bring you to a *Stack Exchange* question and answer website.

The URL `https://www.stat.berkeley.edu/~spector/Rcourse.pdf` by Phil Spector gives a comprehensive 103-slide introduction to the `R` language.

A.2.1 R as a Calculator

The simplest thing you can do with R is to use it as a basic calculator, as in

```
> 1*2*3*4

[1] 24
```

and

```
> sin(1)

[1] 0.841471
```

Here `sin` is the built-in trigonometric function. As on your calculator, numbers can be stored in memory. This is done via the **assignment operator** =, as in:

```
> xx = 10
```

To see the contents of `xx`, type its name:

```
> xx

[1] 10
```

10 is clearly the contents of `xx`, [1] is the row number of the object that 10 is on. You can create objects with words and other characters is the same way.

```
> my.text = "I like R"
```

An object's type is important to keep in mind as it determines what we can do it. For example, you cannot take the mean of a character object like the `my.text` objects:

```
> mean(my.text)

[1] NA
Warning message:
In mean.default(my.text) : argument is not numeric or logical:
returning NA
```

Trying to find the mean of your `my.text` object gives us a warning message and return `NA`: not applicable. To find out an object's type use the `class` function:

```
> class(my.text)

[1] "character"
```

Names of objects are case-sensitive, and must begin with a letter and not contain spaces. Names may include fullstops, such as `my.name`.

A.2.2 Vector and Data Frame Objects

A **vector** is simply a group of numbers, character strings, and so on. Let's create a simple numeric vector containing the numbers 50, 38.5, 37.5. To do this we will use the `c` (concatenate) function:

```
> age = c(50,38.5,37.5)
> age


[1] 50.0 38.5 37.5
```

Vectors of character strings are created in a similar way.

```
> Author = c("Dirk","Benoit","Michael")
> Author


[1] "Dirk"    "Benoit"  "Michael"
```

Vectors consisting of a sequence of numbers can be created via the "colon" operator, e.g., `1:5` is the same as `c(1,2,3,4,5)`, or via the `seq` function, as in:

```
> my.sequence = seq(from=1, to=20, by=2)
> my.sequence


 [1]  1  3  5  7  9 11 13 15 17 19
```

A vector can be rearranged into a matrix via the `matrix` function:

```
> matrix(my.sequence,ncol=5,nrow=2)


     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    3    7   11   15   19
```

If the number of elements in the vector is smaller than the number of elements in the matrix, the vector elements will be "cycled":

```
> matrix(1:5, ncol=5, nrow=2)
```

```
[1,]    1    3    5    2    4
[2,]    2    4    1    3    5
```

Let's now combine the two vectors age and Author into a new object with the cbind (column bind) function.

```
> AgeAuthorObject = cbind(age,Author)
> AgeAuthorObject
```

```
 age     Author
[1,] "50"   "Dirk"
[2,] "38.5" "Benoit"
[3,] "37.5"   "Michael"
```

We have created again a matrix object. Since, matrix objects must have the same type of objects, R has coerced (cast) the numerical age vector into a vector of strings. You can see that the numbers in the age column are between quotation marks. In R, a matrix object is seen as vector with extra attributes, in particular the dimension of the matrix, and possibly the row and column names. The attributes of an object can be obtained and set via the attributes function. The functions colnames and rownames make it possible to retrieve or set column or row names of a matrix-like object.

If you want to have an object with rows and columns and allow the columns to contain data with *different* types, you need to use **data frame** objects, which can be constructed via the data.frame function.

```
> AgeAuthorObject = data.frame(age,Author)
> AgeAuthorObject
```

```
   age  Author
1 50.0    Dirk
2 38.5  Benoit
3 37.5 Michael
```

You can use the names command to see the data frame's name. The command names is not specific to the data.frame object but can be applied to other R objects as well, such as the list object, which is defined later.

```
> names(AgeAuthorObject)
```

```
 [1] "age"    "Author"
```

Notice that the first column of the data set has no name and is a series of numbers. This is the row.names attribute of the data frame. We can use the rownames command to set the row names from a vector.

```
> rownames(AgeAuthorObject) = c("First","Second","Third")
> AgeAuthorObject
```

```
        age  Author
First    50    Dirk
Second 38.5  Benoit
Third  37.5 Michael
```

A.2.3 Component Selection

The dollar sign ($) is called the **component selector**. It enables to extract any column of a matrix-type object via its name.

```
> AgeAuthorObject$age
```

```
 [1] 50.0 38.5 37.5
```

In this example, it extracted the `age` column from the `AgeAuthorObject`. You can then compute for example the mean of the age by using

```
> mean(AgeAuthorObject$age)
```

```
 [1] 39.66667
```

Using the component selector can create long repetitive code if you want to select many components. You can streamline your code by using the `attach` command. This command attaches a database to R's search path (you can see what is in your current search path with the `search` command; just type `search()` into your R console). R will then search the database for variables you specify. You don't need to use the component selector to tell R again to look in a particular data frame after you have attached it. For example, let's attach the `cars` data that comes with the default packages of R. It has two variables, `speed` and `dist` (type `?cars` for more information on this dataset)

```
> attach(cars)
> head(speed)   # Display the first values of speed
```

```
[1] 4 4 7 7 8 9
```

```
> mean(speed)
```

```
[1] 15.4
```

It is a good idea to **detach** a data frame after you are done using it, to avoid confusing R.

```
> detach(cars)
```

Another way to select parts of an object is to use subscripts. They are denoted with squares brackets `[]`. We can use subscripts to select not only columns from data frames but also rows and individuals values. Let's see it in action with the data frame `cars`

```
> head(cars)


 speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10


> cars[3:7,]  # select information from the third through
                  # seventh row


  speed dist
3     7    4
4     7   22
5     8   16
6     9   10
7    10   18


> cars[4,2]  # select the fourth row of dist


[1] 22
```

An equivalent way is:

```
> cars[4,"dist"]


[1] 22
```

Also note the functions **which**, **which.min** and **which.max**, which are often very useful to extract information.

```
> mask = c(TRUE,FALSE,TRUE,NA,FALSE,FALSE,TRUE)
> which(mask) # Outputs the indices corresponding TRUE.
```

```
[1] 1 3 7
```

```
> x = c(0:4,0:5,11)
> which.min(x)  # Outputs the index of the smallest value.
```

```
[1] 1
```

```
>  which.max(x)  # Outputs the index of the largest value.
```

```
[1] 12
```

We can also select the cars with a speed less than 9 mph by using

```
> cars[which(cars$speed<9),]
```

```
 speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
```

An another way is to use the function `subset`:

```
> subset(cars,speed<9)
```

```
 speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
```

A.2.4 List Objects

The most flexible and richest data structure in R is the **list**. Lists can group together data of different types, without altering them. Generally speaking, each element of a list can thus be a vector, a matrix or even a list. Here is a first example:

```
> A = list(TRUE,-1:3,matrix(1:4,nrow=2),"A character string")
> A

[[1]]
[1] TRUE

[[2]]
[1] -1  0  1  2  3

[[3]]
     [,1] [,2]
[1,]    1    3
[2,]    2    4

[[4]]
[1] "A character string"
```

In such a structure, with heterogeneous data types, element ordering is often completely arbitrary. Elements can therefore be explicitly named, which makes the output more user-friendly. Here is an example:

```
> B = list(my.matrix=matrix(1:4,nrow=2),my.numbers=-1:3)
> B

$my.matrix
     [,1] [,2]
[1,]    1    3
[2,]    2    4

$my.numbers
[1] -1  0  1  2  3
```

Naming elements will make it easier to extract elements from a list:

```
> B$my.matrix

      [,1] [,2]
[1,]    1    3
[2,]    2    4
```

A.2.5 Linear Algebra

We can do the usual linear algebra operations in R. Here are some examples.

```
> (A = matrix(1:6,nrow = 2)) # define matrix A and show output

     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Notice that we have used brackets around the assignments of A to force the output to be shown in the console. You can also achieve this by typing the name of the variable in the console or by using `print(A)` in the source code.

A vector in R is always interpreted as a column vector, even though it is printed as a row vector.

```
> (x = 3:1)  # define vector x and show output

[1] 3 2 1
```

Multiplying A with x results in a matrix object with 2 rows and 1 column.

```
> A %*% x  # multiply  matrix A with (column) vector x

     [,1]
[1,]   14
[2,]   20
```

We can coerce this matrix back into a vector (if needed) with the `as.vector` function. The transpose of a matrix is found via the `t` function.

```
> t(A)  # transpose of A

     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

Functions of a matrix are foremost treated in an *elementwise* way, as in

```
> 1/A

     [,1]      [,2]      [,3]
[1,]  1.0 0.3333333 0.2000000
[2,]  0.5 0.2500000 0.1666667
```

and

```
  A * A

     [,1] [,2] [,3]
[1,]    1    9   25
[2,]    4   16   36
```

A common mistake is to use * instead of %*% for matrix multiplication. This need not give an error message, as the elementwise operation may be perfectly legitimate.

Let us introduce another matrix, B, which is a square matrix.

```
> B = matrix(1:4,nrow = 2)
> print(B)
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

The inverse of an invertible square matrix B can be found by solving the linear equation $BX = I$ for matrix X. In R we use the `solve` function:

```
> (solve(B))    # compute and print the inverse of B
```

```
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
```

Next, we solve the linear equation $B\mathbf{x} = (1, 2)^\top$:

```
> x  = solve(B,c(1,2))
> x
```

```
[1] -0.5  0.5
```

The function **apply** can be used to apply a function to the rows or columns of a matrix. For example, with matrix A as above, the column means are:

```
> apply(A,MARGIN=2,FUN=mean)  #column means
```

```
[1] 1.5 3.5 5.5
```

and the row means are

```
> apply(A,1,mean) #row means
```

```
[1] 3 4
```

A.2.6 Flow Control

R has the usual flow control statements for programming, including “if”, “for” and “while” statements:

- if (*condition*) { *expression* } else { *expression* }
- for (*var* in *seq*) { *expression* }
- while (*condition*) { *expression* }

The R code below gives some examples. We can execute the code in RStudio via the “source” button. Typing `source("filename")` in the console, where `filename` is to be specified by you, will execute the code as well. The code illustrates also the use of the **`cat`** and **`print`** functions to output results. See the help files for their different uses. The function **`scan`** can be used to input data, from file, URL, or keyboard. To output a new line, use the special character “\n”. Note that $x == y$ (that is, double equal sign) is used to compare x with y. In the code below, two strings are compared.

```r
cat("Input name");
name = scan(,what="char",nmax=1)  # read the name from keyboard input
if (name == "Dirk"){ print("Welcome back Dirk")
  } else {cat("Hello",name,"\n")}  #important to have "} else"

for (i in 1:10) cat(i^2," ")   # output numbers in a row
cat("\n")                      # put newline in output

# this does the same but prints the results as a column
i = 1
while (i <= 10){
  print(i^2)
  i = i+1
}
```

A common mistake in “if else” statements is to start the “else” as the first word of a new line. This confuses R, as it deals with the previous statement as an “if” statement without the “else” part.

A.2.7 Functions

Functions are simply a set of statements that transform an “input” objects into an “output” object. We have already seen several examples of functions, such as the function **`mean`**. The input to this function is a vector of numbers, and the output is the mean (i.e., average) of these numbers.

> The standard way to use a function is to assign the result of the function f to an object y, as in `y = f(x)`. However, some functions in R can change the attribute $f(x)$ of x to z via an assignment `f(x) = z`. A common example is the **names** function, which not only shows the names of an object, but can be used to change the names of that object as well. Another example is the **levels** function.

Some functions can be called with and “empty” argument. For example, **getwd** gives the current working directory:

```
> getwd()

"c:/Users/JohnSmith/DataProject"
```

Arguments are the input into a function, and use the `ARGUMENTLABEL=VALUE` syntax. To find all of arguments that a command can accept look at `Arguments` section of the command’s help file. Argument labels may be put *in any order* and also can be abbreviated provided there is no ambiguity. It is advised, though, to keep the labels and the order exactly as in the specified argument list.

```
> ?rnorm  #open help file for rnorm
> x = rnorm(n=10,mean=3,sd=2)  #generate normal random variables
> (q = quantile(x, probs = c(0.25,0.75))) #output two quantiles

     25%      75%
1.021414 4.237529
```

Basic Functions

Here are some important data manipulation functions. See the help files for extra arguments.

- **length**: returns the length of a vector.

  ```
  > length(c(1,3,6,2,7,4,8,1,0))

  [1] 9
  ```

- **sort**: sorts the elements of a vector, in increasing order.

  ```
  > sort(c(1,3,6,2,7,4,8,1,0))

  [1] 0 1 1 2 3 4 6 7 8
  ```

- **order**, **rank**: the first function returns the vector of ranking indices of the elements. In case of a tie, the ordering is always from left to right. The second function returns the vector of ranks of the elements. In case of a tie, the ranks are shared and can be non-integer.

  ```
  > vec = c(1, 3, 6, 2, 7, 4, 8, 1, 0)
  > names(vec) = 1:9
  > vec
  ```

  ```
  1 2 3 4 5 6 7 8 9
  1 3 6 2 7 4 8 1 0
  ```

  ```
  > sort(vec)
  ```

  ```
  9 1 8 4 2 6 3 5 7
  0 1 1 2 3 4 6 7 8
  ```

  ```
  > order(vec)
  ```

  ```
  [1] 9 1 8 4 2 6 3 5 7
  ```

  ```
  > rank(vec)
  ```

  ```
   1   2   3   4   5   6   7   8   9
  2.5 5.0 7.0 4.0 8.0 6.0 9.0 2.5 1.0
  ```

- **unique**: this function removes the duplicates of a vector.

  ```
  > unique(c(1,3,6,2,7,4,8,1,0))
  ```

  ```
  [1] 1 3 6 2 7 4 8 0
  ```

- **which**: gives the indices of a boolean vector that are TRUE.

  ```
  > x = c(1,3,6,2,7,4,8,1,0)
  > which(x > 2)
  ```

  ```
  [1] 2 3 5 6 7
  ```

  ```
  > x[ind]
  ```

```
[1] 3 6 7 4 8
```

- **rep**: replicates the values of a vector or object.

```
> rep(1:3,4)

 [1] 1 2 3 1 2 3 1 2 3 1 2 3
```

Create your own functions

We have just seen some brief notions on executing functions in R. The R language can also be used to create your own functions. We give only a brief overview here. You should scrutinize the code below to ensure that you understand it well. To illustrate simply the function creation process, we shall focus on the computation of the Body Mass Index (BMI), from the weight (actually mass!) (in kg) and the height (in m), using the well-known formula

$$\text{BMI} = \frac{\text{Weight}}{\text{Height}^2}.$$

The function BMI defined below returns a list of three named elements (`Weight`, `Height` and `BMI`).

```
BMI = function(weight,height){
            bmi = weight/height^2
            res = list(Weight=weight,Height=height,BMI=bmi)
            return(res)}
```

We can now execute the function `BMI` we just created:

```
>  BMI(weight=70, height=1.82)


 $Weight
[1] 70

$Height
[1] 1.82

$BMI
[1] 21.13271
```

A.2.8 Graphics

R comes with a "base" graphics package. To see the available functions and variables, type:

```
> ls(package:graphics)
```

Some of these are *high-level* functions, which produce complete plots with a single or just a few commands. Examples inlcude **plot**, **boxplot**, **contour**, **barplot**, and **hist**. Other plotting functions are *low-level*, plotting only parts of plots, such as **abline**, **points**, **curve**, **frame**, **axis**, **text**, and **legend**.

The function **plot** is a *generic* function for plotting R objects. Each object invokes its own plot function, called a *method*. Type methods(plot) to see all the methods that are associated with the **plot** function.

Various parameters can be used to change the appearance of a plot. Although in general R does a good job at selecting the right layout, when the plots need to be incorporated in a pdf for example, it may be important to change the size of the fonts. This is generally done via the function **par**. Type ?par to find the many plotting parameters.

Plotting parameters of particular use are:

- cex : changes the size of the characters (especially useful when exporting graphs to be included in a LaTeX document).
- mar: a vector of form c(bottom, left, top, right) giving the margins of the plot.
- pch : the point type: either specified by a character or an integer. See ?points for a list.
- lty: the line type, specified by an integer.
- lwd: the line width.
- col: the color of a line or character, specified as a character string or a number. Type **colors** for available colors.

The following code gives the graph in Figure A.2.

```
f = function(x) sin(x)
g = function(x) sin(x)*exp(-x)
windows(width=8,height=5)  # draws an external window in MS Windows
par(cex=1.5,mar = c(4,2,0.2,0.2))
curve(f,0,pi, lwd = 3,col="blue",xlab = "x",ylab="")
curve(g,0,pi, lwd = 3,lty="dashed",col="darkorange",xlab = "x",add=T)
```

```
legend(-0.05,1.02,c("f(x)","g(x)"),col=c("blue","darkorange"),lwd=2,
       lty=c("solid","dashed"),bty="n")
```

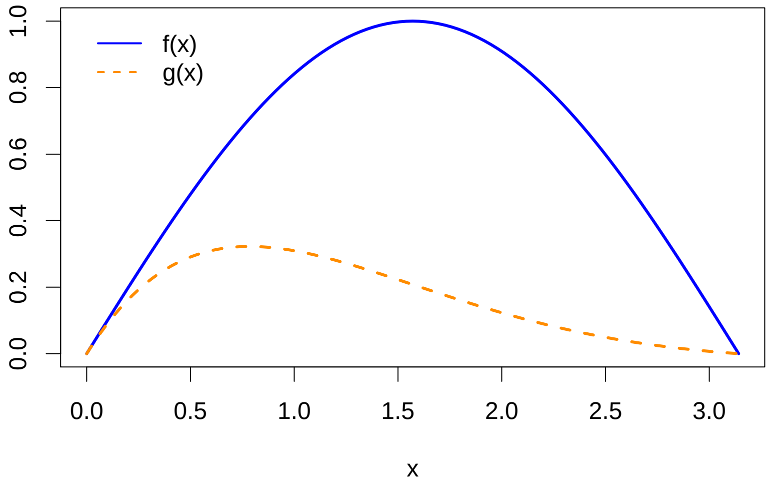


Figure A.2: A simple plot.

Using the commands below we can quickly plot a fitted line to the `cars` regression data. The result is depicted in Figure A.3.

```
> plot(cars)
> abline(lm(dist~speed,data=cars),col="blue")
> points(cars[30,],col="red",pch=20)
```

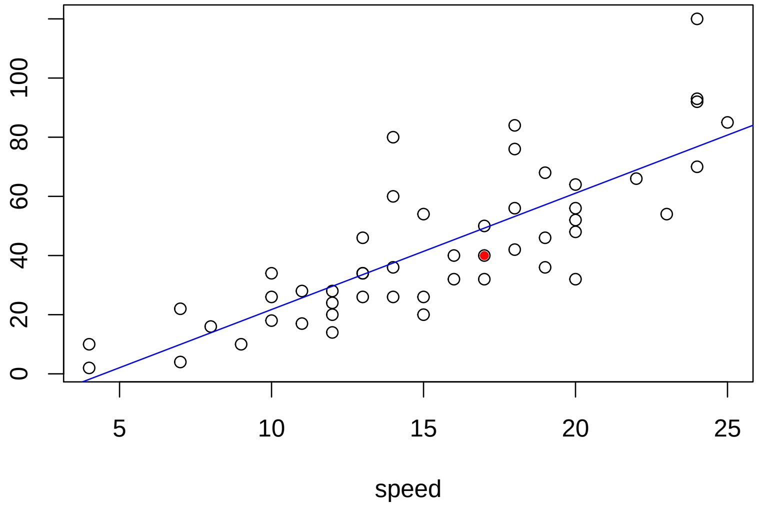


Figure A.3: Plotting a fitted line to regression data, and highlighting one point.

Instead of writing to a window, R can also write to other devices, such as a pdf or postscript file. For example:

```
> pdf("cars.pdf")
> plot(cars)
> dev.off()
```

Plots the cars data into a pdf file called cars.pdf. The file is not written until the command `dev.off()` is issued.

A.2.9 Reading and Writing Data

The following `R` instruction will read the data present in a file (to be chosen in a dialog window) and import them into `R` as a data.frame which we have chosen to call `my.data`.

```
> my.data = read.table(file=file.choose(),header=T,sep="\t",
+                                  dec=".",row.names=1)
```

The function **`read.table`** accepts many arguments; see the helpfile. For CSV (Comma Separated Values) file, you can use instead **`read.csv`**.

When using the function **`read.table`**, you will need to specify the value of the argument `file` which must contain, in a character string, the name of the file and its complete path. You might have noticed that we used the function **`file.choose`**, which opens up a dialog window to select a file and returns the required character string. This is an easy method to get the path to a file, but the path can also be specified explicitly:

```
> my.data = read.table(file="C:/MyFolder/data.txt")
```

> ⚠ Note that file paths are specified using slashes (/). This notation comes from the UNIX environment. In R, you cannot use backslashes (\), as you would in Microsoft Windows, unless you double all the backslashes (\\).

Another option is using the function **`setwd`** to change the work directory. The argument `file` will then accept the file name alone, without its path.

```
> setwd("C:/MyFolder")
> my.file = "mydata.txt"
> data = read.table(file=my.file)
```

Your data are now available in the `R` console: they are stored in the object which you have chosen to call `data`. You can visualize them by typing `data`; you can also type `head(data)` or `tail(data)` to display only the beginning or the end of the dataset. You can also use `str(data)` to see the nature of each column of your data.

For writing data, the relevant function is `write.table`. Suppose you have a data.frame called `mydata`, containing data that you wish to save in a text file. You would then use the instruction:

```
> write.table(mydata, file = "myfile.txt", sep = "\t")
```

A.2.10 Workspace, Batch Files, Package Installation

All of the objects you create become part of your workspace. Use the **ls** function to list all of the objects in your current workspace.

```
> ls()   # the brackets () are essential!
```

```
> [1] "age"  "AgeAuthorObject" "Author"  "my.text"
```

You can remove specific objects by using the **rm** function:

```
> rm(my.text)  #remove my.text object
```

If you want to remove all objects in the workspace use `rm(list=ls())`.

When you enter a command into R it becomes part of your history. To see the most recent commands in your history use the **history** command or use the `History` pane in RStudio. You can also use the up and down arrows on your keyboard when your cursor is in the R console to scroll through your history.

Finally, we mention that R can be called in a shell. In RStudio, a shell can be created under the Tools menu. In the shell, you can type

```
> R CMD --help
```

to get a list of possible commands. In particular,

```
> R CMD BATCH infile.R outfile.txt
```

executes the statements from `infile.R` and writes the results to `outfile.txt`, or to standard output when the output file is not provided. You can also use `Rscript` instead of `R CMD BATCH`. To try things out, run the following batch file, e.g, named `batch.R`, in a command shell:

```
x = seq(0,2*pi,by=0.1)
print(x)      # write to standard output
windows()     # open a window
plot(sin(x))
Sys.sleep(5)  # wait 5 seconds before exiting
```

Before you can load a package with `library()`, you will need to *install* it first. In R studio the “Packages” tab in the lower-right IDE pane shows the packages that have already been installed. Clicking on the “Install” tab opens a window to search for new packages, which then will be automatically installed. Packages can also be manually installed via the **install.package** function.

INDEX