Contents

1 Introduction

This book is written for those who want to learn to analyze data. This challenge arises frequently when you need to determine a previously unknown fact. For example: does this new medicine have an effect on a patient's symptoms? Or: Is there a difference between the public's rating of two politicians? Or: how will the value of the dollar change in the next week?

You might think that you can find the answer to such a question simply by looking at the numbers. Unfortunately this is often not the case. For example, after surveying 262 people exiting a polling site, it was found that 52% voted for candidate A and 48% for candidate B. Do the results of this exit poll tell you that candidate A won the election? Thinking about it, many would say "yes," and then, considering it for a moment, "Well, I don't know, maybe?" But there is a simple (from the point of view of modern computer programs) "proportion test" that tells you not only the answer (in this case, "No, the results of the exit poll do not indicate that Candidate A won the election") but also allows you to calculate how many people you would need to survey to be able to answer that question. In this case, the answer would be "about 5000 people"—see the explanation at the end of the chapter about one-dimensional data.

Not knowing how to use methods of statistical analysis can lead to mistakes and misinterpretations. Unfortunately, understanding of these methods is far from common. Many college majors require a course in probability theory and mathematical statistics, but all many of us remember from these courses is horror and/or frustration at complex mathematical formulae filled with Greek letters, some of them wearing hats. But probability theory forms the basis of most data analysis methods. On the other hand, it's not always necessary to understand the physics of radio waves to enjoy listening to the radio. For the practical purposes of analyzing data, you don't have to be fully fluent in mathematical statistics and probability theory. Textbooks with titles like "Statistics Without Tears: An Introduction for Non-Mathematicians" abound.

Some caution is required, though, on the part of both authors and readers of such books: many methods of statistical analysis have, so to speak, a false bottom. You can apply these methods without delving too

deeply into the underlying principles, get results, and discuss these results in your report. But you might find one day that a given method was totally unsuitable for the data you had, and therefore your conclusions are invalid. You must be careful and aware of the limitations of any method you try to use and determine whether they are applicable to your situation.

On examples: We have tried to use as many examples as possible, both simple and complex, from a number of fields. We also tried to reduce the amount of theoretical material because we know that many people find concrete examples to be most useful for learning. Because this book is based on a script-based computer program [?], we have made many of the scripts used here publicly available to download at [address]. This site also contains any data files that are not included in the R software, and additional useful links.

How this book is structured: The first chapter is almost entirely theoretical. If you don't feel like reading these discussions, you can skip to Chapter 2. But the first chapter contains information that will help you avoid many common pitfalls. In Chapter 2, the most important sections are those beginning with "How to download and install R," which explain how to work with R. Mastering the material in these sections is crucial to get anything out of subsequent ones. We recommend carefully reading and working through all the problems in this section. Subsequent chapters make up the core of the book, explaining the most widely used methods of data analysis. The chapter titled "Statistical Investigation," which talks about the general organization [?] of statistical analysis, concludes the book by discussing again the methods introduced in previous chapters. The appendices contain useful information about graphical interfaces in R, delineate a simple example of a project in R, describe the basics of programming in R, and excerpt the official documentation. Every appendix is a small handbook that can be used more or less independently from the rest of the book.

Of course, many statistical methods, including quite important ones, are not discussed in this book. We almost completely neglect statistical modeling, don't discuss contrasts, don't examine many standard distributions besides the normal, effect tests, survival curves, Bayesian methods, factor analysis, geostatistics, we don't talk about how to do multifactorial or block dispersion analysis, design experiments, and much else. Our goal is to teach fundamentals of statistical analysis. Having mastered the basics, more advanced methods can be grasped without much difficulty with the help of the scholarly literature, internal documentation, and online resources.

A few technical notes: many example problems in this book can and should be reproduced independently. These examples are written in `typewriter font` and begin with the `>` symbol. If an example doesn't fit on one line, a `+` sign indicates the line's continuation—do not type the `+` sign when reproducing the code. When a reference is made to loading data, it's assumed that the necessary files are stored at the subdirectory "data" of the active directory. If you are downloading data from the above site, don't forget to create this directory and copy the data there.

2 How to evaluate the overall trend

In any sample, there are two most common characteristics: *center* (central tendency) and *spread* (pp). As the center of the most commonly used *mean* and *median*, and as a scatter—*standard deviation* and *quartile*. mean the arithmetic average of the median differs primarily in that it works well mostly when the data distribution is close to normal (we 'll talk more about this later). Median not so dependent on the characteristics of the distribution is said to statistics, it is a *robust* robustness (stable). understand the difference is easiest in this sample. Consider again our hypothetical employees. Here is their salary (in thous.)

```
> salary <- c (21, 19, 27, 11, 102, 25, 21)
```

The difference in salaries is due, in particular, the fact that Sasha—forwarder and Kate—head of the firm.

```
> mean (salary); median (salary)
[1] 32.28571
[1] 21
```

It turns out that due to the high average salary Katina worse reflects typical " ", central salary than the median. Why does it happen ? The fact that the median is calculated quite differently than the average.

—The median is the value that cuts off half of the ordered sample. To better see this, go back to the two vectors on which example in the previous chapter showed how grades are assigned:

```
> a1 <- c (1,2,3,4,4,5,7,7,7,9,15,17)
> a2 <- c (1,2,3,4,5,7,7,7,9,15,17)
> median (a1)
[1] 6
> median (a2)
[1] 7
```

In the vector `a1` only twelve values, that is an even number. In this case, the median—the average between the two central figures. Vector Y `a2`, everything is easier, there eleven values, so the median is taken just mean.

Also median sample to evaluate the properties are very useful quartile, that is, those values that cut accordingly 0%, 25%, 50%, 75% and 100% of the total distribution data. If you read the previous paragraph carefully, you might have realized that the median—it's just the third quartile (50%). The first and fifth quartile—are respectively a minimum and a maximum, and the second and fourth quartiles used to compute robust dispersion (see below). Concept can quartile " " expand and introduce a special term for the value of the clipping any interest ordered distribution (not necessarily Quarters)—this is called " *quantile* ". Quantiles are used, for example, when analyzing the data for normality (see below).

To characterize the spread is often used and parametric value—*standard deviation*. Widely known " " three sigma rule, which states that if the means of two samples differ by more than triple the standard deviation, then the samples are different, that is taken from various general populations. This rule is very convenient, but, unfortunately, means that both samples must obey the normal distribution. To calculate the standard deviation in R provides the function `sd ()`.

But average and median, there is one central distributions, the so-called *mod* fashion, the most frequently occurring value in the sample. Fashion is rarely used and mostly for nominal data. Here's how to calculate it in R (we used to calculate the variable `sex` from the previous chapter):

```
> sex <- c ("male", "female", "male", "male", "female", "male",
+ "Male")
> t.sex <- table (sex)
> mode <- t.sex [which.max (t.sex)]
> mode
male
   5
```

Thus, our sample fashion—`male`.

Often the task is to count the mean the arithmetic mean (or median) for the entire data table. There are several techniques to facilitate the life. Show them the example of embedded data `trees`:

```
> attach (trees) # The first method
> mean (Girth)
[1] 13.24839
> mean (Height)
[1] 76
> mean (Volume / Height)
[1] 0.3890012
> detach (trees)
> with (trees, mean (Volume / Height)) # The second method
[1] 0.3890012
> lapply (trees, mean) # The third way
$ Girth
[1] 13.24839
$ Height
[1] 76
$ Volume
[1] 30.17097
```

The first method (using the `attach ()`) allows you to connect to the database table column list of current variables. After that variables can be referenced by name without mentioning the name of the table. It is important not to forget to do at the end of `detach ()`, because the great danger of confused that you have attached, and that—no. If variables were connected somehow modified on the table itself is not affected.

The second method, in fact, similar to the first, attachment occurs only within parentheses function `with ()`. The third method uses the fact that the data table is—lists of speakers. For strings, this technique does not work, it will be necessary run `apply ()`. (If you came up with the fourth method, then recall that the cyclic structure of type `for` in R without the need not welcome).

standard deviation standard deviation, dispersion (its square) and the so-called interquartile variation caused similar average:

```
> sd (salary); var (salary); IQR (salary)
[1] 31.15934
[1] 970.9048
[1] 6
```

The last expression, the distance between the second and fourth quartiles IQR (interquartile or scatter), robustness and better suited for example, with a salary than the standard deviation.

We apply these functions to embedded data `trees`:

```
> attach (trees)
> mean (Height)
[1] 76
> median (Height)
[1] 76
> sd (Height)
[1] 6.371813
> iQR (Height)
[1] 8
> detach (trees)
```

It is evident that these characteristics trees much closer to each other. Reasonable to assume that the distribution of tree height is close to normal. We'll check it below.

In our data on wages—only 7 digits. And how do you know whether there are any outstanding " " figures, such as Katina salaries in large data, " " thousandth the size ? For this is the graphics functions. The simplest—the so-called " box - with - mustache " or boksplot. First, add to our data a thousand hypothetical workers with wages randomly taken from the interquartile spread of source data (Fig. 1)

```
> new.1000 <- sample ((median (salary) - IQR (salary)):
+ (Median (salary) + IQR (salary)), 1000, replace = TRUE)
> salary2 <- c (salary, new.1000)
> boxplot (salary2, log = "y")
```

This is an interesting example of another because it is the first presented technique to generate random values. function `sample ()` able to select data from a random sample. In this case, we used `replace = TRUE`, because we had a lot of numbers to choose from a much smaller sample. If you write on R imitation card games (and such programs are written !), You must use `replace = FALSE`, because of the deck can not get back the same card. Incidentally, the fact that the random values, it follows that the results of subsequent calculations may differ if they play again, so your schedule may look slightly different.

But back to boxplot. As can be seen, Katina salary represented high points (so high that we even had to enter the option `log = "y"`, to the underlying point become more visible). The box itself, that is the main rectangle bounded above and below quartile quartile, so that the height of the rectangle—it IQR. The so-called " " mustache default represent points on the remote half IQR. The middle line of the rectangle—it is easy to guess, median median. Points lying outside " " whiskers are considered as outliers and therefore are drawn separately. Boksploty been specially devised renowned statistician John Tukey, in order to quickly, efficiently and consistently reflect the basic characteristics of the sample robust. R can draw several boxplots immediately (ie, the command vectorized see the result in Fig. 2)

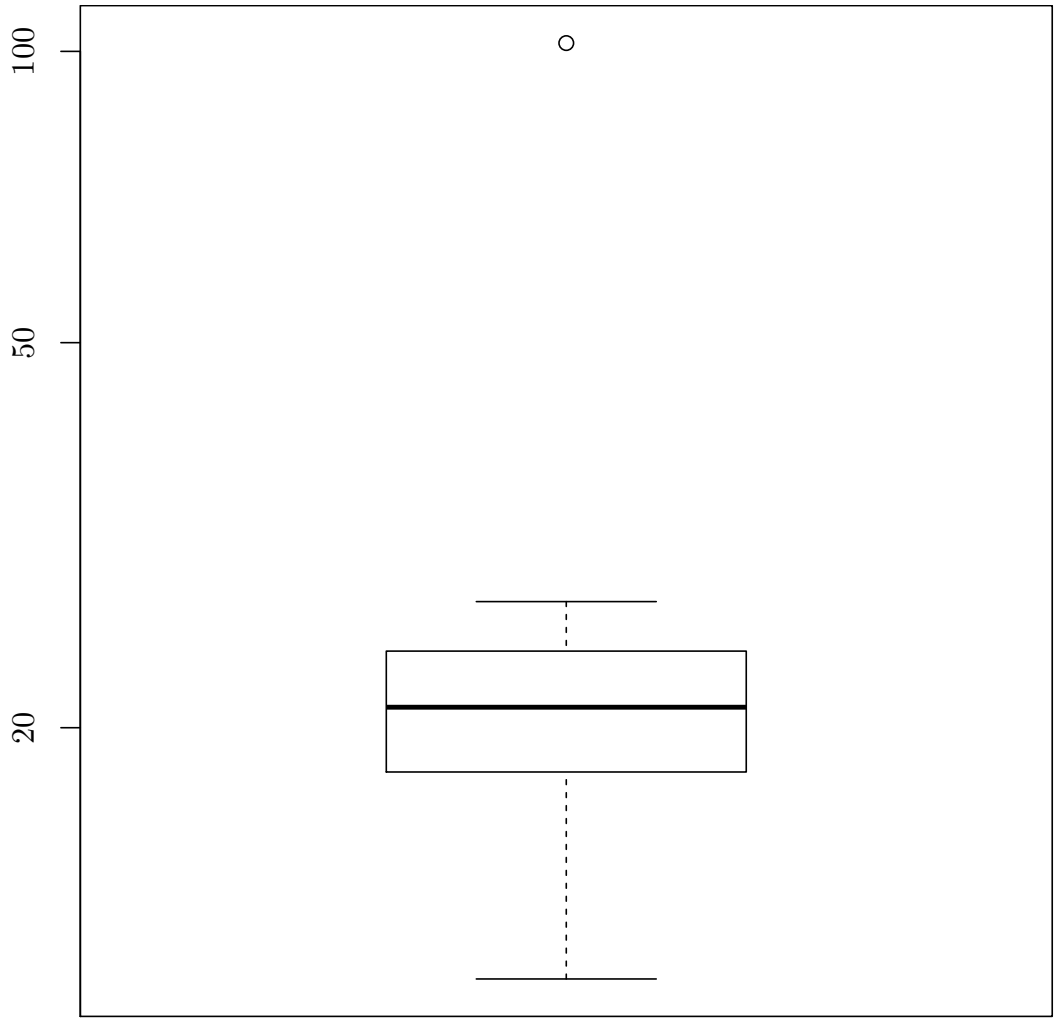


Figure 1: Boxplot

```
> boxplot (trees)
```

There are two functions that are associated with boksplotami. The function `quantile ()` by default gives all five quartile quartile, and the function `fivenum ()`—the main characteristics of the distribution by Tukey.

Another way the graphic is— histogram histogram, ie the line of columns whose height corresponds to the occurrence of data that fall within a certain range (Fig. 3)

```
> hist (salary2, breaks = 20, main = "")
```

In our case, `hist ()` default variable splits into 10 intervals, but their number can be configured manually, as in the present example. Numerical analog of the histogram is the function `cut ()`. With this function, you can find out what type of data as we have:

```
> table (cut (salary2, 20))
(10.9,15.5] (15.5,20] (20,24.6] (24.6,29.1] (29.1,33.7]
         76391295244 0
(33.7,38.3]
          0
(38.3,42.8] (42.8,47.4] (47.4,51.9] (51.9,56.5] (56.5,61.1]
          0 0 0 0 0
```

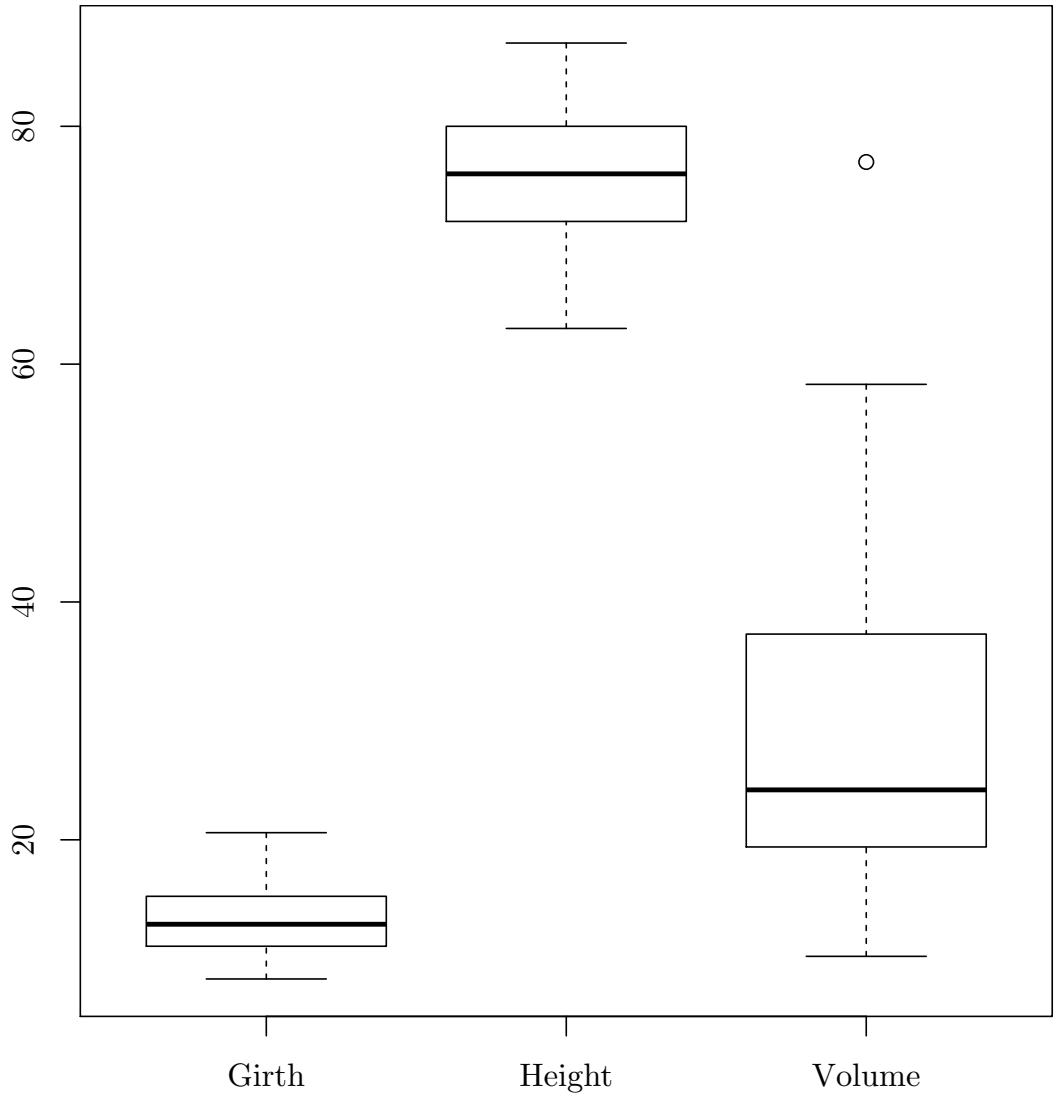


Figure 2: Three boksplota, each represents one column of the data table

```
(61.1,65.6]
          0
(65.6,70.2] (70.2,74.7] (74.7,79.3] (79.3,83.9] (83.9,88.4]
          0 0 0 0 0
  (88.4,93] (93,97.5] (97.5,102]
          0 1 0
```

There are two graphics functions close to the histogram. One is `stem()`:

```
> stem (salary, scale = 2)
  The decimal point is 1 digit (s) to the right of the |
   1 | 19
   2 | 1157
   3 |
   4 |
   5 |
   6 |
   7 |
   8 |
   9 |
  10 | 2
```

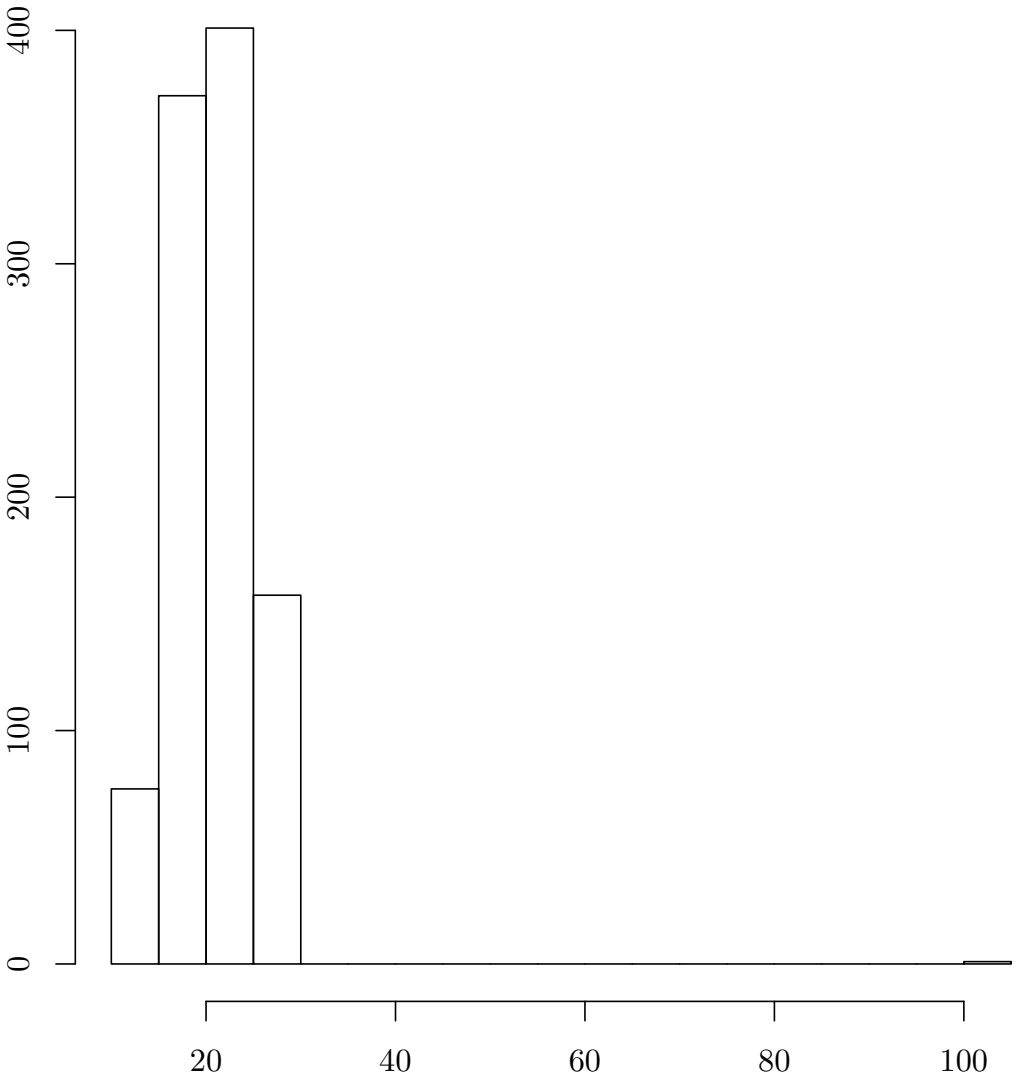


Figure 3: Histogram hypothetical staff salaries in 1007

It is very simple—data values are represented not dots, and numbers corresponding to these values themselves. Thus, it is seen that in the range of from 10 to 20, there are two pay (11 and 19) in the range of from 20 to 30 four—, and so etc.

Another feature too close to the histogram, but requires much more sophisticated calculations. This graph density distribution (Fig. 4)

```
> plot (density (salary2, adjust = 2), main = "")
> rug (salary2)
```

(We used " " adds a graphics function `rug ()`, to highlight the places with the highest density values.)

In fact, we have *histogram smoothing*—attempt to turn it into a continuous smooth function. How smooth it will be depends on the parameter `adjust` (by default, it is equal to unity). Result of smoothing is also called distribution schedule.

Also boksplotov and various family schedules " " histograms in R and many other one-dimensional graphs. Graph " " hive such reflects not only the density of a sample distribution of values, but also how these values are themselves located (dot). To plot the hive need to download (and possibly also install first) package `beeswarm`. After that we can look at myself " " hive (Fig. 5)

```
> library ("beeswarm")
> beeswarm (trees)
> boxplot (trees, add = TRUE)
```

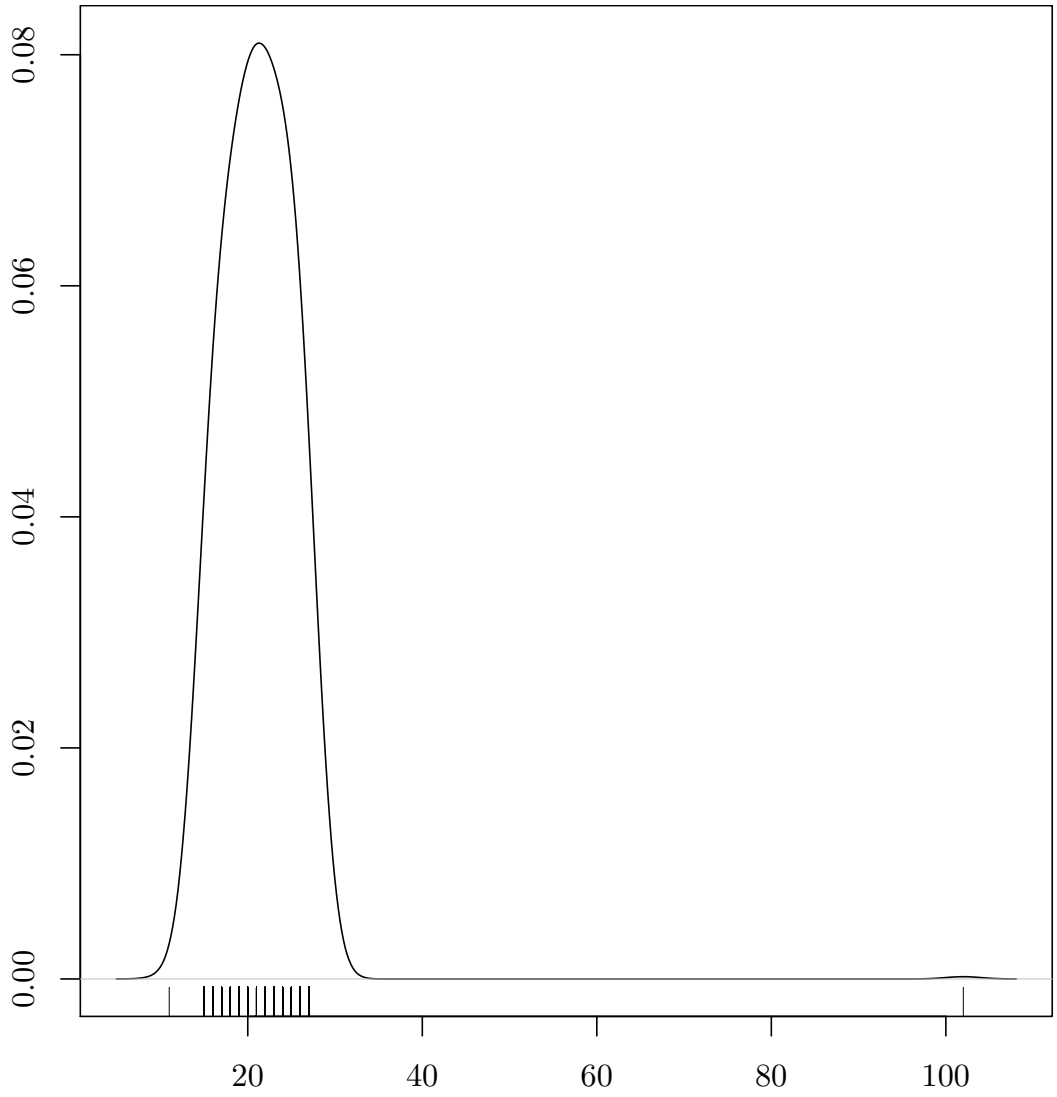


Figure 4: Density distribution of salaries of employees in 1007 hypothetical

We are not here simply plotted hive, but also added there boksplot to expose the quartile quartile and median median. For this we need an argument `add = TRUE`.

And finally, the most important function, `summary ()`:

```
> lapply (list (salary, salary2), summary)
[[ 1]]
   Min. 1st Qu. Median Mean 3rd Qu.
  11.00 20.00 21.00 32.29 26.00
   Max.
 102.00

[[2]]
   Min. 1st Qu. Median Mean 3rd Qu.
  11.00 18.00 21.00 21.09 24.00
   Max.
 102.00
```

In fact, it returns the same data as the `fivenum ()` with the addition of the mean value (`Mean`). Note, incidentally, that both wages “ ” median median identical, while the average differ greatly. This is another example of instability mean values—because with the addition of randomly combined salaries “ ” form of the distribution should not have changed significantly.

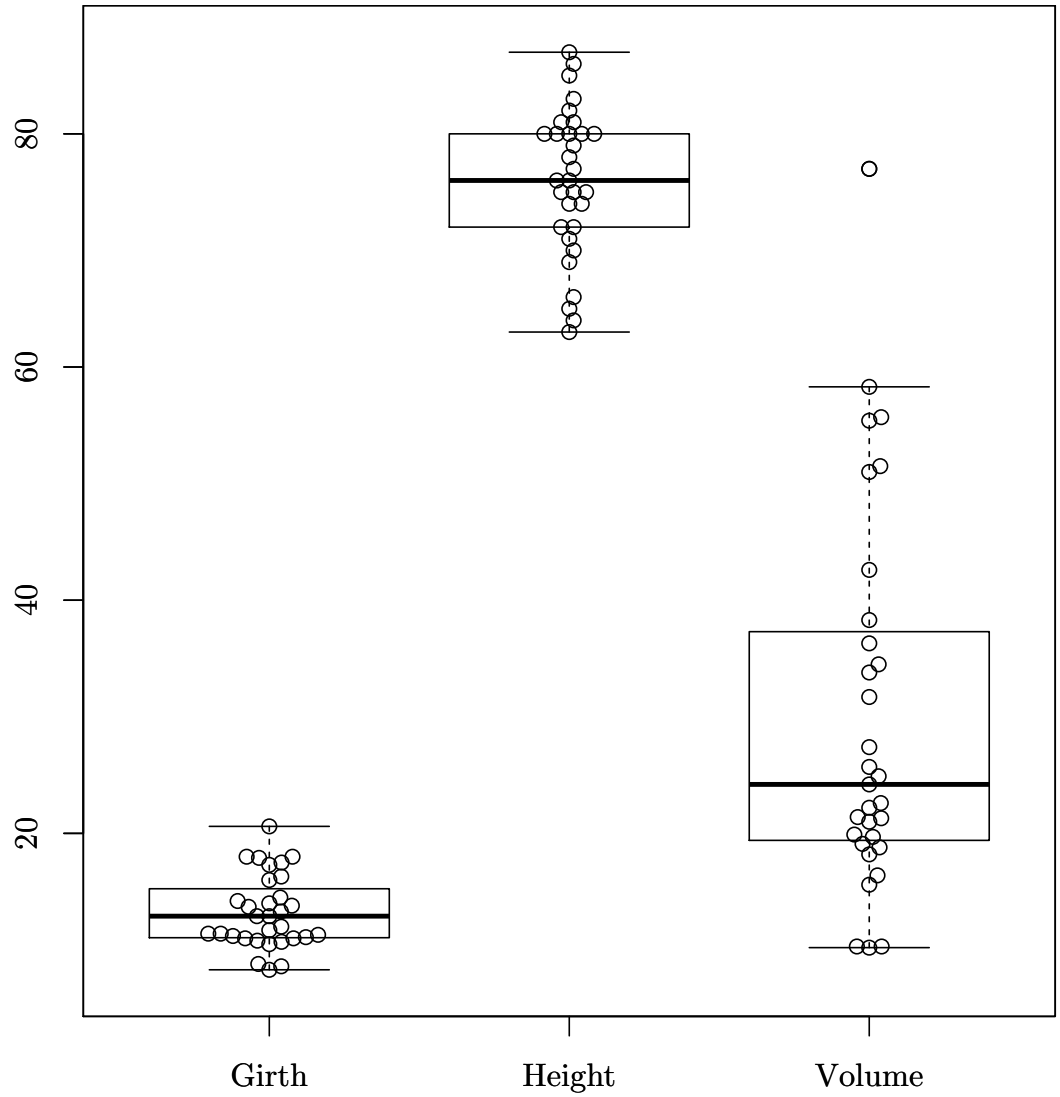


Figure 5: Graph - hive “ ” superimposed boksplotami for three characteristics of trees

function `summary ()`—general, and the laws of object # -oriented approach, it returns different values for different types of objects. You have just seen, it works for numeric vectors. For lists, it works a little differently. The output can be, for example, so (for example, embedded data `attenu` about 23 earthquakes in California):

```
> summary (attenu)

     event mag station dist
 Min.: 1.00 Min.: 5.000 117: 5 Min.: 0.50
 1st Qu.: 9.00 1st Qu.: 5.300 1028: 4 1st Qu.: 11.32
 Median: 18.00 Median: 6.100 113: 4 Median: 23.40
 Mean: 14.74 Mean: 6.084 112: 3 Mean: 45.60
 3rd Qu.: 20.00 3rd Qu.: 6.600 135 3 3rd Qu.: 47.55
 Max.: 23.00 Max.: 7.700 (Other): 147 Max. 370.00
                                  NA’s: 16
     accel
 Min.: 0.00300
 1st Qu.: 0.04425
 Median: 0.11300
 Mean: 0.15422
 3rd Qu.: 0.21925
```

```
 Max.: 0.81000
```

Variable `station` (station number of observations)— factor, and besides missing data, therefore appears otherwise.

Before you finish the story about the main characteristics of the sample, it is necessary to mention another characteristic variation. To compare the variability of traits (especially those that are measured in different units) are often used dimensionless quantity— coefficient of variation. This is simply the ratio of the standard deviation to the mean, taken as a percentage. That's how you can compare the coefficient of variation for different attributes of trees (embedded data `trees`):

```
> 100 * sapply (trees, sd) / colMeans (trees)
    Girth Height Volume
23.686948 54.482331 8.383964
```

Here we have applied for speed `sapply ()`—option `lapply ()` with simplified output and `colMeans ()`, which simply calculates the average for each column. Just note that functions like `colMeans ()`, in R few. For example, a very widely used function `colSums ()` and `rowSums ()`, which give the totals, respectively, in rows and columns (the main function of a spreadsheet !). There are, of course, more and `rowMeans ()`.

3 Bad Data

Ability to function `summary ()` to indicate missing data, the highs and lows is very good help at an early stage of data analysis—quality control. Suppose we have data typed correctly, and they are located in the directory `data` in the current directory:

```
> dir ("data")
[1] "errors.txt"...
> err <- read.table ("data / errors.txt", h = TRUE, sep = "\t")
> str (err)
'data.frame': 7 obs. of three variables:
 $ AGE: Factor w / 6 levels " 12", " 22", " 23",..: 3 4 3 5 1 6 2
 $ NAME: Factor w / 6 levels "", "John", "Kate",..: 2 3 1 4 5 6 2
 $ HEIGHT: num 172 163 161 16.1 132 155 183
> summary (err)
 AGE NAME HEIGHT
 12:1: 1 Min.: 16.1
 22:1 John: 2 1st Qu.: 143.5
 23:2 Kate: 1 Median: 161.0
 24:1 Lucy: 1 Mean: 140.3
 56:1 Penny: 1 3rd Qu.: 167.5
 a: 1 Sasha: 1 Max. 183.0
```

Processing begins with the availability of the desired file. Besides command `summary ()`, here used as a very useful command `str ()`. As can be seen, the variable `AGE` (age) somehow became a factor, and `summary ()` shows why: in one of the cells has crept letter `a`. In addition, one of the names is empty, likely because cell forgotten put `NA`. Finally, the minimum height 16.1 cm— ! This does not happen even in newborns usually, so that we can confidently assert that the typesetter just accidentally put a point.

4 Univariate statistical tests

Finished deal with descriptive statistics, we turn to a simple statistical tests (more tests in the next chapter). Let's start with the so-called one-dimensional " " that allow you to check assertions about how distributed source data.

Suppose we know that the average salary in our first example—about 32 thousand rubles. Let us now check how reliable this figure:

```
> t.test (salary, mu = mean (salary))
One Sample t-test
data: salary
t = 0, df = 6, p-value = 1
alternative hypothesis: true mean is not equal to 32.28571
95 percent confidence interval:
  61.103302 3.468127
sample estimates:
mean of x
 32.28571
```

This version of the Student test for one-dimensional data. Statistical tests (including this), the so-called attempt to calculate test statistic, in this case, the statistics Student (t- statistics). Then, based on this statistic is calculated " p-value " (p-value), reflecting the probability of *type I error*. A error of the first kind (also called false alarm " "), in turn, is a situation where we take the so-called alternative hypothesis, while the *really* holds zero (by hypothesis " default "). Finally, the calculated p-value is used for comparison with a predetermined threshold (level) significance. If the p-value is below the threshold, the null hypothesis is rejected if the above—adopted. More information about the statistical hypotheses can be found in the next chapter.

In our case, the null hypothesis is that the true mean (ie the mean of the total population) than the calculated average world (ie 32.28571).

We proceed to analyze the function output. Statistics Student with six degrees of freedom (`df = 6`, since we only 7 values) gives a single p-value, that is 100%. Whatever common threshold we have not taken (0.1% 1% or 5%), this value is still more. Therefore, we accept the null hypothesis.

As an alternative hypothesis in this case—is that now " " average (initial sample) is not equal to the calculated average, it turns out that in fact " " these numbers are not statistically different. Besides all this, and still function gives CI (confidence interval), which, in her opinion " ", may be present in between. Here he greatly—from three and a half thousand to 61 thousand rubles.

Nonparametric (ie not associated assumptions about the distribution) analogue of this test also exists. This is called the Wilcoxon rank test:

```
> wilcox.test (salary2, mu = median (salary2), conf.int = TRUE)
Wilcoxon signed rank test with continuity correction
data: salary2
V = 221949, p-value = 0.8321
alternative hypothesis: true location is not equal to 21
95 percent confidence interval:
 20.99999 21.00007
sample estimates:
(pseudo) median
      21.00004
```

This function displays almost the same as the `t.test()` above. Note, however, note that this test is not associated with a moderate, but with median median. Calculated (if you specify `conf.int = TRUE`) and confidence interval. Here it is much narrower because the median is much more stable than the average.

Understand whether the normal distribution of the data (or at least close to normal if it is), it is very, very important. For example, parametric statistical methods are all based on the assumption that the data are normally distributed. Therefore, in R implements several different techniques to answer questions about the normality of data. Firstly, this statistical tests. The easiest of them— Shapiro- Wilks (try to test it yourself):

```
> shapiro.test (salary)
> shapiro.test (salary2)
```

But what it shows ? This function displays much smaller than in previous cases. Moreover, even the built-in help does not contain an explanation of what is, for example, an alternative hypothesis. Of course, you can refer to the literature, the benefit of help provides references to publications. And you can just set up an experiment:

```
> set.seed (1638)
> shapiro.test (rnorm (100))

Shapiro-Wilk normality test

data: rnorm (100)
W = 0.9934, p-value = 0.9094
```

`rnorm ()` generates as many random numbers normally distributed, as indicated in his argument. This is analogous to the function `sample ()`. Once we got a high p-value, it indicates that the alternative hypothesis in this case: " distribution does not correspond to the normal ". In addition to the results of the secondary playing were the same as those used by the function `set.seed ()`, regulating built- in R random number generator random number generator so that the numbers in the following command were created by one and the same law " ".

Thus, the distribution of data in `salary` and `salary2` different from normal.

Another popular way to check whether the distribution is similar to normal—graphic. Here's how (Fig. 6)

```
> qqnorm (salary2, main = "")
> qqline (salary2, col = 2)
```

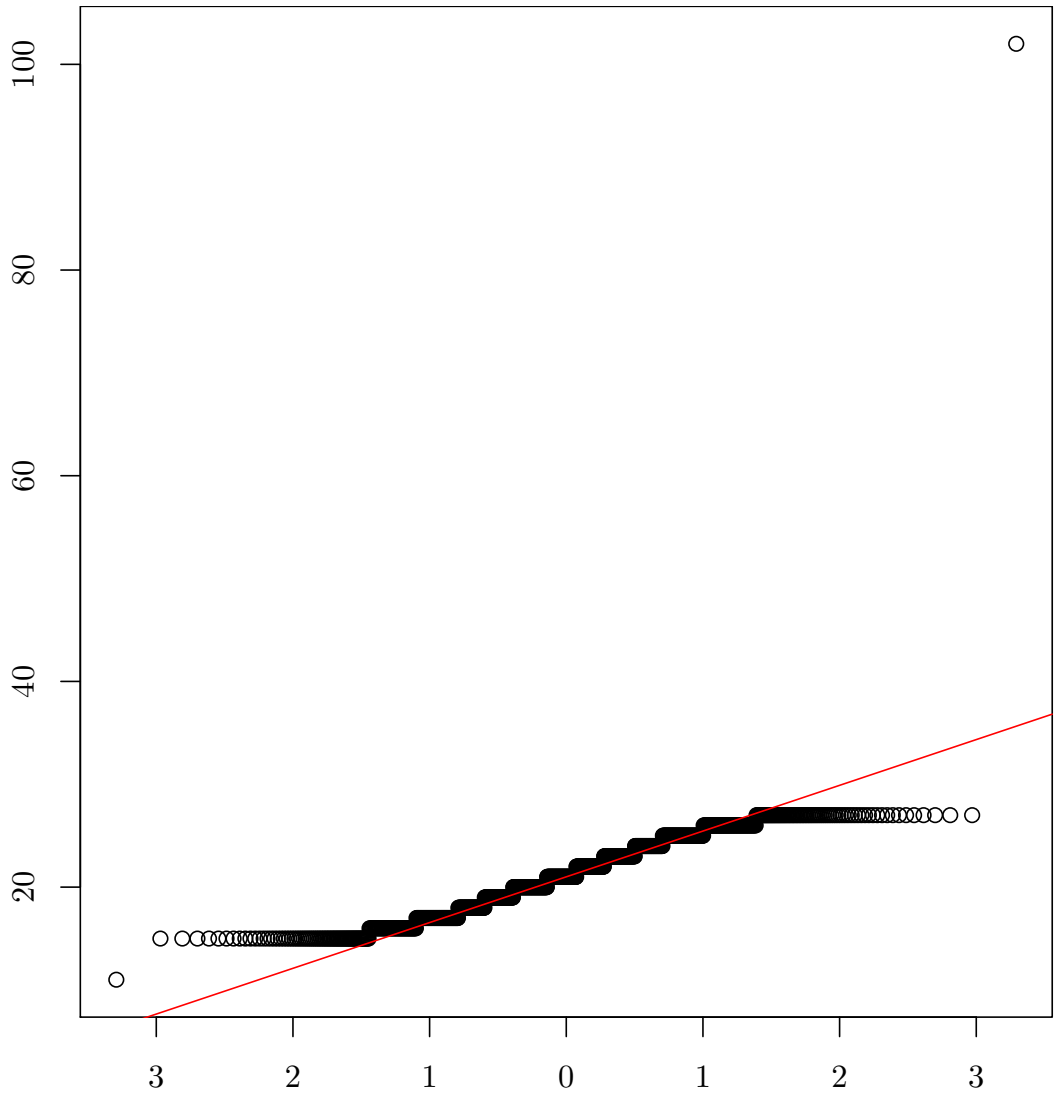


Figure 6: Graphic checking normality

For each element is calculated, what place it occupies in the sorted data (the so-called " ") and what place he would take if the normal distribution quantile is conducted through Direct quartile quartile. If the points are on the line, the distribution is normal. In our case, many points are far enough away from the red line, so it does not look like normal distributed.

To test the normality can be used more versatile Kolmogorov—Smirnov, who compares any two distributions, so for comparison with the normal distribution it is necessary to explicitly state " `pnorm` ", ie the so-called cumulative normal distribution function (it's embedded in R):

```
> ks.test (salary2, "pnorm")
One-sample Kolmogorov-Smirnov test

data: salary2
D = 1, p-value <2.2e- 16
alternative hypothesis: two-sided
```

It produces about the same as the text Shapiro- Wilks.

5 How to create your own functions

Shapiro- Wilks test all good, but not vectorized, like many other tests in R, so apply it to multiple columns of data table will not work. did it on purpose, in order to emphasize the undesirability of multiple pairwise comparisons (more about this in the chapter written about the two-dimensional data.) but in this case there is no paired comparisons, and want to save time. possible, of course, pressing the up arrow " " carefully repeat the test for each column, but the more correct approach—create a custom function. Here is an example of such a function:

```
> normality <- function (data.f)
+{
+ Result <- data.frame (var = names (data.f), p.value = rep (0,
+ Ncol (data.f)), normality = is.numeric (names (data.f)))
+ For (i in 1: ncol (data.f))
+{
+ Data.sh <- shapiro.test (data.f [, i]) $ p.value
+ Result [i, 2] <- round (data.sh, 5)
+ Result [i, 3] <- (data.sh>.05)
+}
+ Return (result)
+}
```

To function to work, you need to copy these lines to the console window or burn them to a separate file (preferably with the extension `*. R`), and then download the command `source ()`. After that, it can cause:

```
> normality (trees)
     var p.value normality
1 Girth 0.08893 TRUE
2 Height 0.40342 TRUE
3 Volume 0.00358 FALSE
```

Function not only runs the Shapiro- Wilks test several times, but still legible, and the result of the draws. Let us consider the function in more detail. The first line indicated its argument—`data.f`. Next, surrounded by curly braces, is the function body. On the third line formed by an empty table data such dimension, which we need in the end. Then begins the cycle: for each column test is performed, and then (this is important !) Extracted from the test p-value. This procedure is based on the knowledge of the structure of the test O—the list where the item `p-value` contains p-value. You can check this by looking at the certificate, but you can experimentally (how?—See the answer at the end of the chapter). All p-values

are retrieved, rounded, compared with a threshold level of significance (in this case 0.05) and recorded in the table. Then the table is given out " ". The proposed function completely optimized. It can easily be made a bit shorter and also more " " smarter, so to speak:

```
> normality2 <- function (data.f, p =.05)
+{
+ Nn <- ncol (data.f)
+ Result <- data.frame (var = names (data.f), p.value = numeric (nn),
+ Normality = logical (nn))
+ For (i in 1: nn)
+{
+ Data.sh <- shapiro.test (data.f [, i]) $ p.value
+ Result [i, 2:3] <- list (round (data.sh, 5), data.sh> p)
+}
+ Return (result)
+}

> normality2 (trees)
```

The results, of course, no different. But you can see how you can add an argument, and at once with a default value. Now you can write:

```
> normality2 (trees, 0.1)

     var p.value normality
1 Girth 0.08893 FALSE
2 Height 0.40341 TRUE
3 Volume 0.00358 FALSE
```

That is, if instead of 5% to take a ten percent threshold, it is for the first column, you can reject the normal distribution.

Has been said that the cycles in R should be avoided. Can it be done in this case ? It turns out, yes:

```
> lapply (trees, shapiro.test)
$ Girth

Shapiro-Wilk normality test

data: X [[1L]]
W = 0.9412, p-value = 0.08893

$ Height
...
```

As you can see, things are even simpler ! If we want to improve the visual effect, you can do so:

```
> lapply (trees, function (.X) ifelse (shapiro.test (.X) $ p.value>
+.05, "NORMAL", "NOT NORMAL"))

$ Girth
[1] "NORMAL"

$ Height
[1] "NORMAL"

$ Volume
[1] "NOT NORMAL"
```

Here applied the so-called anonymous function, function without a name, usually used as a last argument type `apply ()`. Also, a logical construct used `ifelse ()`.

And finally, on this basis can make the third user-defined function (**check** yourself how it works):

```
> normality3 <- function (df, p =.05)
+{
+ Lapply (df, function (.X) ifelse (shapiro.test (.X) $ p.value>
+ P, "NORMAL", "NOT NORMAL"))
+}

> normality3 (list (salary, salary2))
> normality3 (log (trees +1))
```

Examples also interesting. First, our third feature can be applied not only to the data tables, but also to present lists of unequal length elements. Secondly, the simple logarithmic transformation immediately changed normality of columns.

6 Are percentages always accurate?

A useful feature of the study is the proportion of data (share). In statistics, a proportion understand the attitude objects studied feature of the total number of observations. Since the fraction— is part of the whole, the ratio to the whole part is in the range from 0 to 1. For convenience, it is multiplied by the fraction of 100% yield—percentage number from 0% to 100%. Care should be taken to the calculation of shares and do not forget about the original data. In the 1960s, the district department of agriculture has been found that deaths of horses in one rural settlement of 50%. I had to make a competent commission to verify the reasons for this heinous crime. Upon arrival, the Commission found that this settlement was only two horses, including recently sdohshaya old horse, which was a cause of the formation and departure to the place of commission!

Consider the problem, which is often found in statistical studies. How to find out whether different percentage calculated by us from the true " " percent, ie the proportion of the objects of interest in the general population ?

Here's an example. At the hospital, a group of 476 patients, including 356 smokers. We know that on average, the percentage of smokers in the hospital was 0.7 (70%). But in our group it a little more— about 75%. In order to test the hypothesis that the proportion of smokers in the group of patients under consideration differs from the average share of the hospital, we can use the so-called Binomial binomial test:

```
> binom.test (x = 356, n = 476, p = 0.7, alternative = "two.sided")

Exact binomial test

data: 356 and 476
number of successes = 356, number of trials = 476,
p-value = 0.02429
alternative hypothesis: true probability of success is not
equal to 0.7
95 percent confidence interval:
 0.7063733 0.7863138
sample estimates:
probability of success
             0.7478992
```

Since p-value less than 0.05 and significantly alternative hypothesis is that the proportion of smokers (it is here quite mockingly called "probability of success") is not equal to 0.7, then we can reject the null hypothesis and accept the alternative, that is, to decide what our 74% different from the average in the hospital 70% no accident. As an option, we used `alternative = "two.sided"`, but could have been done differently— test the hypothesis that the proportion of smokers in the group of patients considered exceeds

the average proportion of smokers in the hospital. Then the alternative hypothesis would have to be written as `alt = "greater"`.

In the binomial, we can apply here the so-called *test* proportions. should be noted that it is used widely because it is more versatile:

```
> prop.test (x = 356, n = 476, p = 0.7, alternative = "two.sided")

1 -sample proportions test with continuity correction

data: 356 out of 476, null probability 0.7
X-squared = 4.9749, df = 1, p-value = 0.02572
alternative hypothesis: true p is not equal to 0.7
95 percent confidence interval:
 0.7059174 0.7858054
sample estimates:
        p
0.7478992
```

As you can see, the result is almost the same.

Proportions test can be carried out with two samples, it uses all the same function `prop.test ()` (two-sample text for proportions), and `mcnemar.test ()` (for Mac Nemara test, which takes place when the sample related to each other). Learn how to use them, you can read the FAQ (and especially the examples !) For both functions.

* * *

The answer to the problem of function `normality ()`. To find out where to get the values of p-value, we must first recall that R almost everything that appears on the screen, this is the result—" " print various lists using invisible command `print ()`. A list of easy to get " " desired either by name or by number (if, say, there is no name for the elements—so sometimes). First, find out what is in the list, the output of the `shapiro.test ()`:

```
> str (shapiro.test (rnorm (100)))
List of 4
 $ Statistic: Named num 0.992
 .. - Attr (*, "names") = chr "W"
 $ P.value: num 0.842
 $ Method: chr "Shapiro-Wilk normality test"
 $ Data.name: chr "rnorm (100) "
 - Attr (*, "class") = chr "htest"
```

This list of 4 elements is an element called `p.value`, which is what we needed. Verify in any case, whether it is:

```
> set.seed (1683)
> shapiro.test (rnorm (100)) $ p.value
[1] 0.8424077
```

That's what we need. It remains only to insert it in our name.

7 Analyzing relationships: Two-dimensional data

Here we will talk about how to work with two samples. If we have two sets of numbers, the first thing we might think to do is compare them. For this, we will need statistical tests.

7.1 What is a statistical test?

It is time to learn about the core of statistics: hypothesis testing. In the previous chapter, we got a brief introduction to constructing statistical hypotheses. We didn't need to delve too deeply into this topic while we were discussing single samples, but in analyses of relationships this concept is crucial.

7.1.1 Statistical hypotheses

We know that a sample needs to be representative, or adequately represent the true properties of the population it's drawn from. But how can we know whether the sample is representative unless we sample the whole population? Even when we meet the criteria of randomization and repeatability, uncertainty remains. Furthermore, when we consider that random sampling is a probabilistic process, all further considerations of our samples need to be thought of in terms of probability. We will never be able, on the basis of our sample, to infer the properties of the population with 100% certainty. We will only be able to formulate hypotheses and calculate their probability.

Philosophers of science such as Karl Popper postulated that science can never prove a theory, but only disprove it. If we collect 1000 facts that support a theory, it doesn't mean we have proved it—it's possible that the 1001st piece of evidence will disprove it. For this reason, in statistical testing, two hypotheses are put forth. The one we are trying to prove is called the alternative hypothesis (H1). The other is called the null hypothesis (H0). The null hypothesis is a proposition of absence of something (for example, difference between two samples or relationship between two variables). We can't prove the alternative hypothesis, but we can reject the null hypothesis and accept the alternative. If we cannot reject the null hypothesis, then we must accept it.

7.1.2 Statistical errors

When you're performing a statistical test on a hypothesis, there are four possible outcomes.

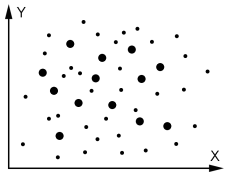
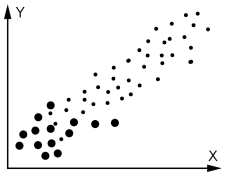
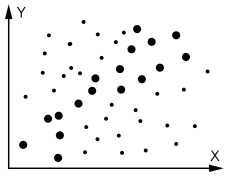
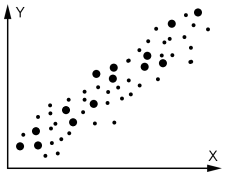
Population / Sample	Null true	Alternative true
Accept null		
Accept alternative		

Table 1: Type I and Type II errors.

If we've accepted the null hypothesis H0, and the hypothesis is indeed correct for the population, then we have reached the correct conclusion and everything is fine. Analogously with the alternative hypothesis. (Obviously, we can't really know what is true about the general population, but we are now simply considering all possible outcomes.)

If we've accepted the alternative hypothesis, when in reality it is not correct for the population, we have committed what is called a Type I statistical error—we have found a pattern that doesn't exist. The probability of committing a Type I error, called a p-value, is always reported when a statistical test is performed. If the probability of committing a Type I error is too high, we must reject the alternative hypothesis. The obvious question is: what probability is "too high"? Just as with choosing sample sizes (see first chapter) there is no definitive answer to this question. The conventional answer is that the threshold value should be 0.05—the alternative hypothesis is accepted if the probability of erroneously rejecting the

null hypothesis is less than 5%. In medicine, where the cost of mistakes can be in human lives, the thresholds are set more strictly, at .01 or .001 (that is, a pattern is considered to be real if the probability of an error is negligible).

In this way, the result of a statistical test is mainly based on the probability of a Type I error. The amount of confidence a researcher has that a conclusion based on a sample will be accurate for the sampled population is reflected by... ?

In cases where we accept the null hypothesis when in fact the alternative hypothesis is true, we have committed a type II error—that is, we fail to detect a pattern that actually exists. This parameter is characterized by what is called the power of the statistical test. The smaller the probability of a type II error, the more powerful the test.

7.2 Is there a difference? Or, comparing two samples

It is important to remember that the tests discussed here test for differences only between measures of central tendency (for example, means) and assume that the variances of the samples approximately similar. For example, the samples (example) and (example) have the same mean, but different variances, and thus would not be detected as different from each other by statistical tests.

To conduct a statistical test, two hypotheses must be put forth. The null hypothesis is "there is no difference between these two samples"—that is, they are both drawn from the same population. The alternative hypothesis is "there is a difference between these two samples."

Your data must be organized in the form of two vectors, separate or organized into a data table. For example, if you want to find out whether men and women differ in height, then one vector needs to consist of the heights of men, and the other of the heights of women, with each entry in the vector corresponding to a measurement of one person.

If the data are parametric, then a parametric t-test is required. If the variables that we want to compare were obtained on different [objects?], we will use a two-sample t-test for independent variables, which is called with the command t.test(). For example, if the two samples we want to compare are written in the first and second columns of data table data, the command t.test(data[,1], data [,2]) will carry out a two-sample t-test.

If the pairs of measurements to be compared were obtained on one [object?], that is, the variables are dependent (for example, heart rate before and after exertion as measured on the same person), then a paired t-test must be used. To do this, specify in the command t.test that the parameter paired=TRUE.

The paired test is more powerful. Imagine that we were taking heart rate measurements before exercise from one person, and after exercise from another. It would be unclear, then, what would explain the measured difference: the effect of exercise or the two people generally having different heart rates. Paired measurements allow each person to serve as their own control, and the difference between the measurements can only be explained by the exercise factor.

If we are dealing with nonparametric data, a nonparameric Wilcoxon test (also known as a Mann-Whitney test) is required, under the command wilcox.test(). Analogously to the t-test, paired data is specified by specifying the parameter paired=TRUE.

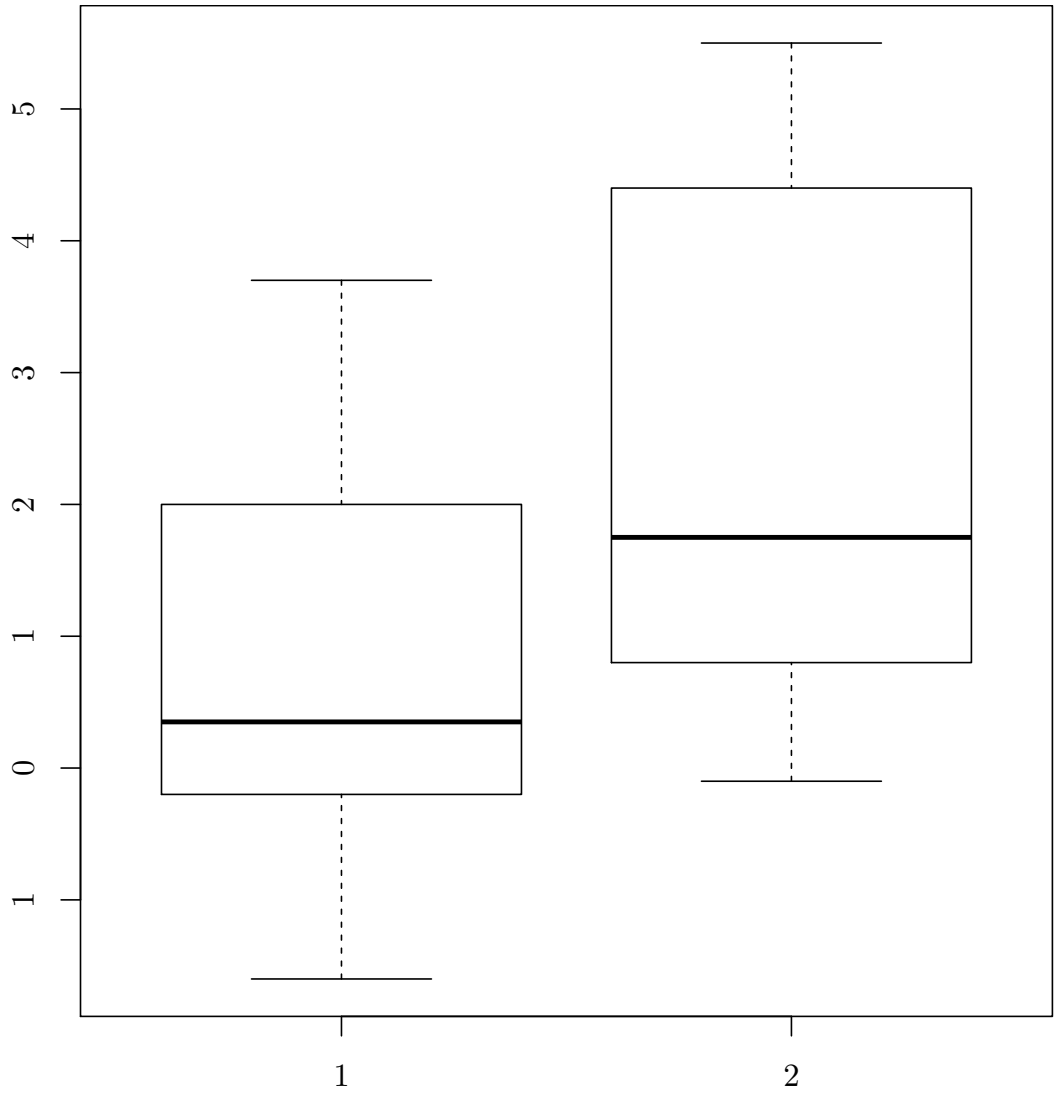
Several examples follow, using the classic data set used in the original work of Student (the pseudonym of mathematician William Sealy Gossett). This work was concerned with comparing the effects of two sleep aids on the duration of sleep. In R these data are available under the name sleep. The column extra contains the average additional duration of time asleep (compared to the control group), while the column group contains the code of each drug.

```
> plot(extra ~ group, data = sleep)
```

This uses the "model formula": in this case, verb|extra group|.

The effect of each drug on each person is individual, but the average length by which the drug prolongs sleep can be considered a reasonable representation of the "strength" of the drug. With this assumption, we will attempt to use a t-test to determine whether there is a significant difference between the means of the two samples corresponding to the two drugs.

```
> with(sleep, t.test(extra[group == 1], extra[group == 2],
+ var.equal = FALSE))

Welch Two Sample t-test
```




```
data:  extra[group == 1] and extra[group == 2]
t = -1.8608, df = 17.776, p-value = 0.0794
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y
     0.75      2.33
```

The parameter var.equal allows one to choose between the classic t-test, which assumes that the variances of the two samples are equal (var.equal=TRUE), or the t-test with the Welch correction, which is free from this assumption.

Although formally we cannot reject the null hypothesis (that the means are equal), the p-value (0.794) is still small enough to suggest trying other methods of testing the hypothesis—increasing the number of observations, checking again the normality of the distribution, and so forth. You might consider carrying out a one-sided test—these are usually more sensitive. This should not be done. Most statistical tests are designed to be carried out ad hoc, that is without knowing any additional information. Post hoc tests also exist (for example, Tukey's Honest Significant Difference test, discussed below), but there are not many of these.

For comparing two samples, nonparametric tests also exist. One of these, the sign test, is so simple that it doesn't exist in R. It is, however, simple to do on one's own. The sign test calculates the differences

between every pair of elements in two samples of equal size (that is, it is a paired test). Then, disregard any negative values and consider only the positive ones. If the samples are taken from the same distribution, then approximately half the differences should be positive, and then the familiar binomial test will not find a significant difference between 50% and the proportion of positive differences. If the samples are different, then the proportion of positive difference should be significantly more or less than half.

Exercise. Come up with R code to carry out such a test, and test the two samples that were mentioned at the beginning of the section.

We will proceed now to more complicated nonparametric tests with an example. The standard data set airquality contains information about the amount of ozone in the atmosphere around New York City from May to September 1973. The concentration of ozone is presented as a rounded mean for every day, so to analyze it conservatively we use nonparametric methods. (As an extra exercise, determine how close to normally distributed the monthly concentration measurements are.)

Let us test the hypothesis that ozone levels in May and August were the same:

```
> wilcox.test(Ozone ~ Month, data = airquality,
+ subset = Month %in% c(5, 8))

Wilcoxon rank sum test with continuity correction

data:  Ozone by Month
W = 127.5, p-value = 0.0001208
alternative hypothesis: true location shift is not equal to 0
```

Since Month is a discrete variable (the number simply represents the month), the values of Ozone will be grouped by month. We aksi used the parameter subset with the operator `%in%`, which chooses May and August, the 5th and 8th month.

The test rejects the null hypothesis, of equality between the distribution of ozone concentrations in May and August, fairly confidently. This is plausible because the ozone level in the atmosphere strongly depends on solar activity, temperature and wind.

Differences between samples are well represented by box plots,

```
> boxplot(Ozone ~ Month, data = airquality,
+ subset = Month %in% c(5, 8))
```

Notice that in the `boxplot()` command we use the same formula as the statistical model.

It's conventionally considered that if the boxes overlap by more than a third of their length, the samples are not significantly different.

The t-test and Wilcoxon test can be used on one sample, if the goal is compare it to a particular standard. These are called one sample tests. The null hypothesis in this case is formulated as equality between the sample mean and the given value of μ.

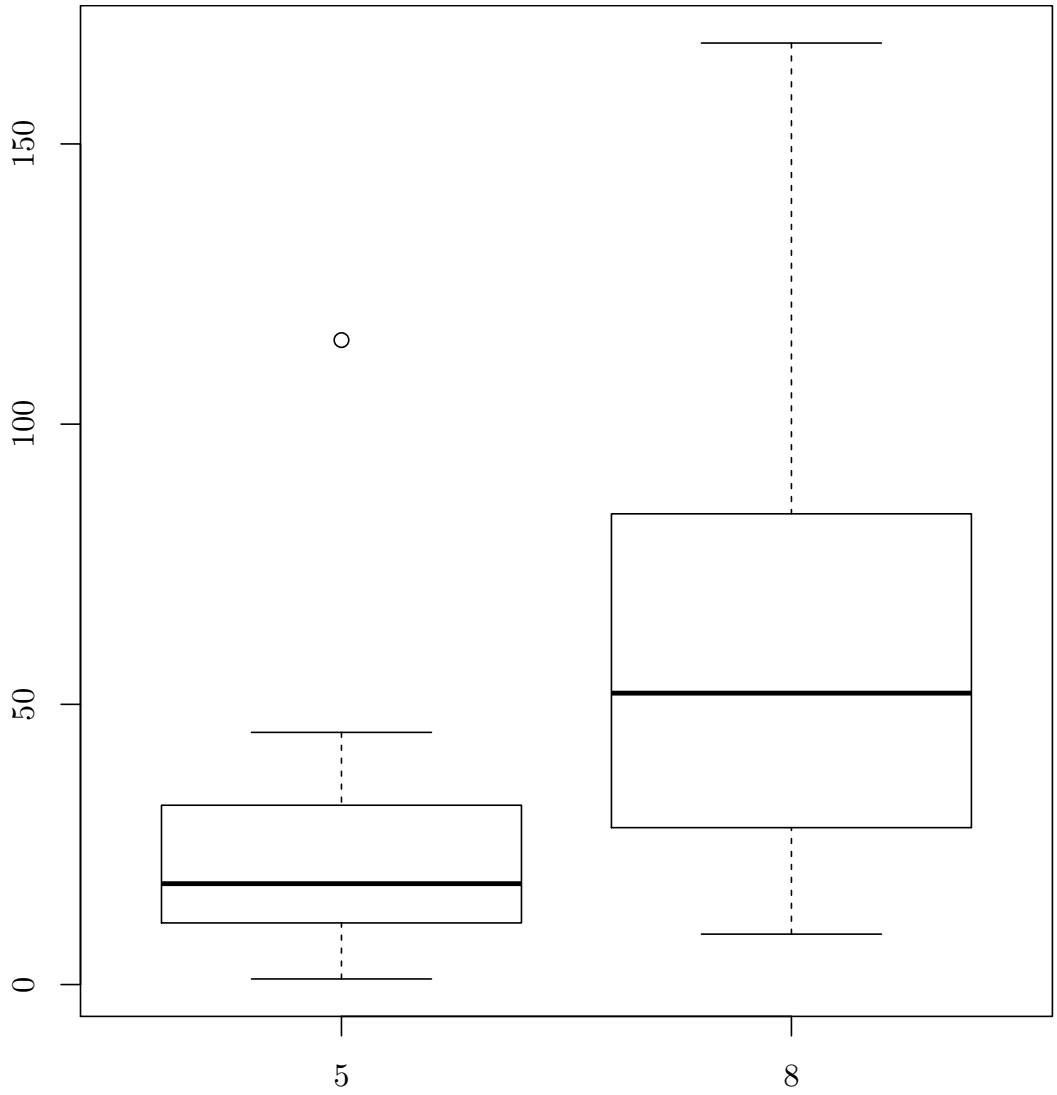
Exercise. In the data file `otsenki.txt` are the grades of a particular group of students for the first academic quarter (in the column labeled A1) and the second quarter (A2), as well as the grades of a second group of students for the first quarter (B1). Do the first group's grades for the first and second quarters differ? Which class did better in the first quarter—A or B?

Exercise. A supermarket has two cashiers. To analyze their work efficiency, the length of the line at each of their registers is recorded several times a day. The data are recorded in `kass.txt`. Which cashier processes customers more quickly?

7.3 Is there an association? Analysis of tables

How do you compare two samples of categorical data? For this, contingency tables are used. A contingency table can be built with the function `table()`:

```
> with(airquality, table(cut(Temp, quantile(Temp)), Month))

         Month
           5  6  7  8  9
```




```
  (56,72] 24  3  0  1 10
  (72,79]  5 15  2  9 10
  (79,85]  1  7 19  7  5
  (85,97]  0  5 10 14  5
```

The rows of this table are temperature intervals, while the columns are months. Each cell's value is the number of observations in each month that fall into the given temperature interval.

If there are more than two factors, R will build a multi-dimensional table and print it as a series of two-dimensional tables, which is not always convenient. A "flat" contingency table can be built if all the factors except one are combined into one multidimensional factor. To do this, use the command `ftable()`:

```
> ftable(Titanic, row.vars = 1:3)

                  Survived  No Yes
Class Sex    Age
1st   Male   Child            0   5
             Adult          118  57
      Female Child            0   1
             Adult            4 140
2nd   Male   Child            0  11
             Adult          154  14
      Female Child            0  13
             Adult           13  80
```

```
3rd   Male   Child          35  13
             Adult         387  75
      Female Child          17  14
             Adult          89  76
Crew  Male   Child           0   0
             Adult         670 192
      Female Child           0   0
             Adult           3  20
```

The parameter row.vars allows you to specify the variables in the data set that should be combined into one factor, the subcategories of which will be indexed by the rows of the contingency table. Col.vars does the same thing with columns.

The function table can be used for other purposes as well. The simplest is calculation of frequencies.

```
> d <- factor(rep(c("A","B","C"), 10), levels=c("A","B","C","D",
+ "E"))
> is.na(d) <- 3:4
> table(factor(d, exclude = NULL))

   A    B    C <NA>
   9   10    9    2
```

The function `mosaicplot()` creates a graphical representation of a contingency table.

```
> titanic <- apply(Titanic, c(1, 4), sum)
> titanic
> titanic
      Survived
Class   No Yes
  1st  122 203
  2nd  167 118
  3rd  528 178
  Crew 673 212
> mosaicplot(titanic, col = c("red", "green"), main = "",
+ cex.axis=1)
```

The function `chisq.test()` allows one to test a hypothesis about independence of two factors with a chi-squared test. For example, let's see whether hair and eye color are associated:

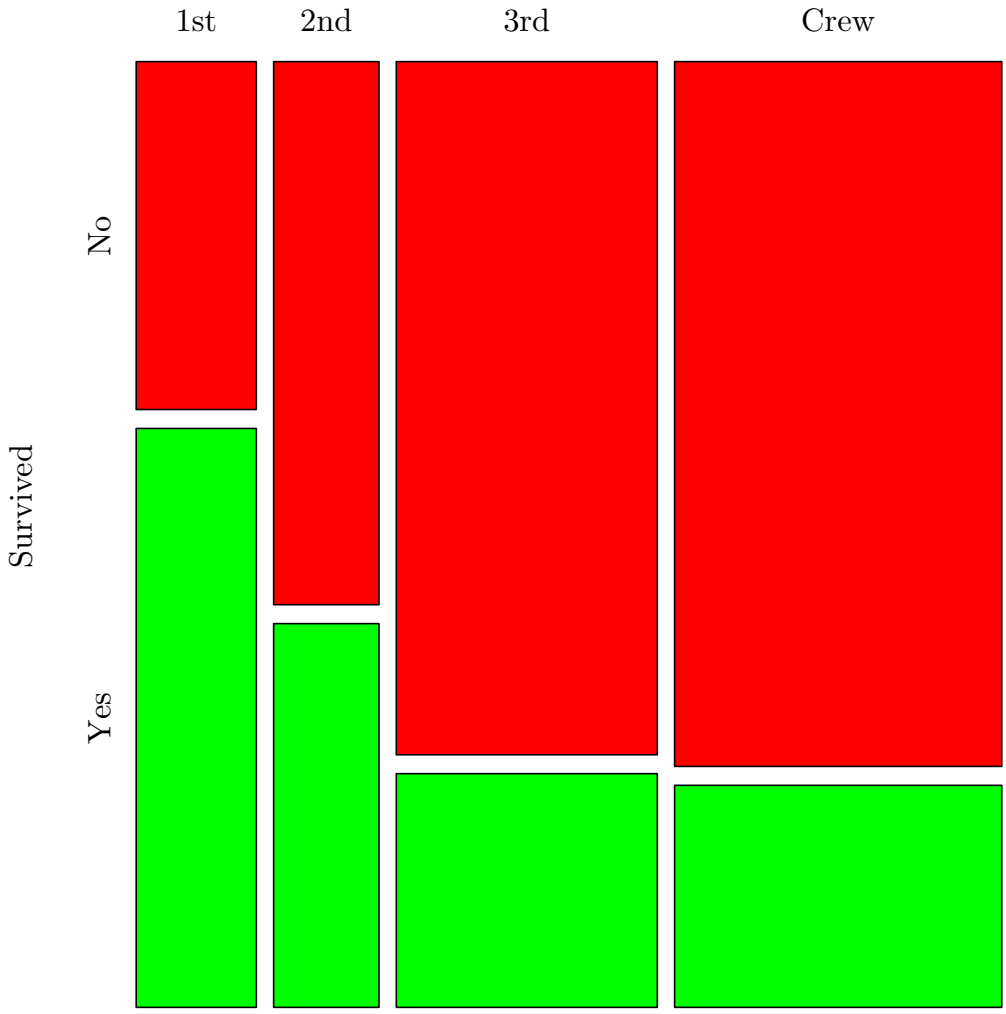
```
> x <- margin.table(HairEyeColor, c(1, 2))
> chisq.test(x)

Pearson's Chi-squared test

data:  x
X-squared = 138.2898, df = 9, p-value < 2.2e-16
```

(The same effect can be achieved by using the contingency table as the argument in the function `summary()`).

The dataset `HairEyeColor` is a multidimensional contingency table. For the summed frequencies across all dimensions except two, the function margin.table is used. The result will be a two-dimensional contingency table. The chi-squared test takes as its null hypothesis the independence of the factors, so in our example, since we reject the null hypothesis, we find that the factors are associated.

To graphically represent these correspondences, the function `assocplot()` can be used:

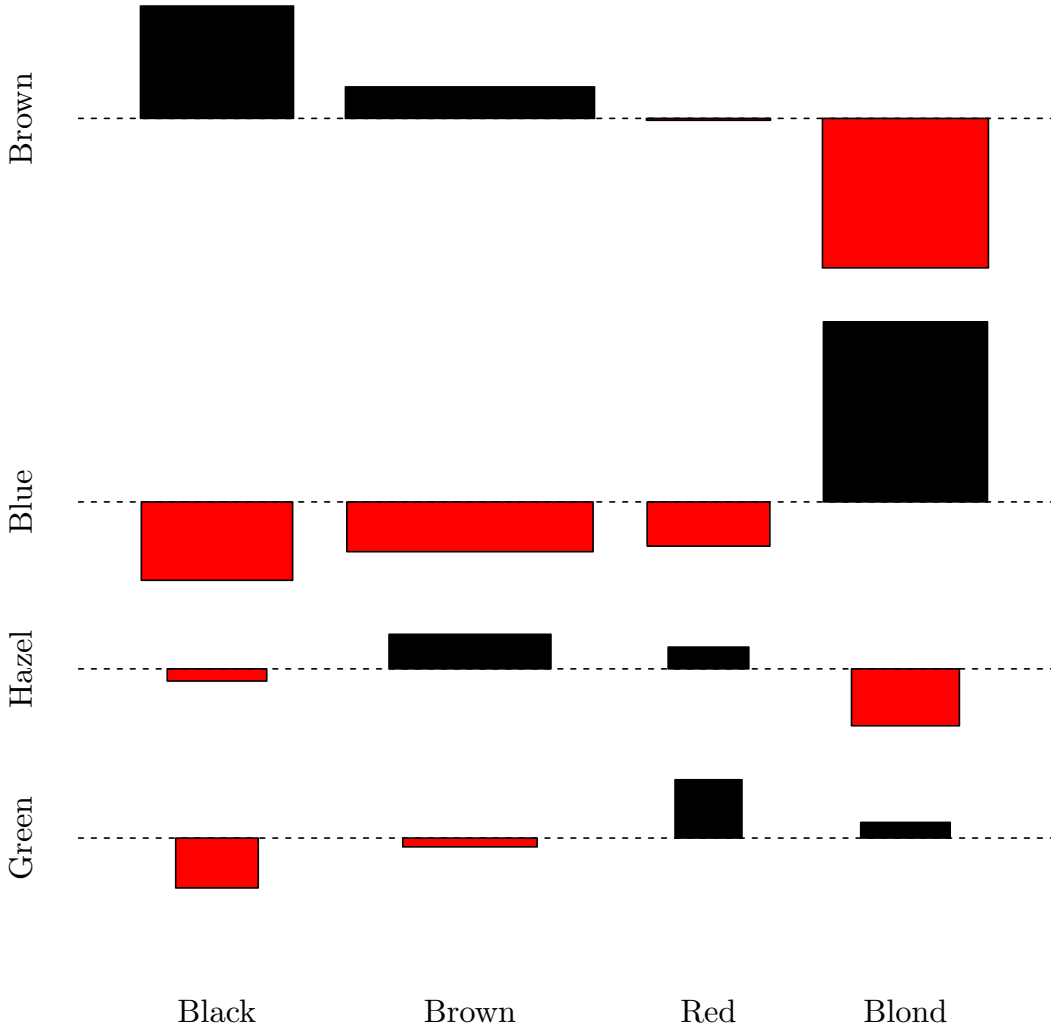
```
> x <- margin.table(HairEyeColor, c(1,2))
> assocplot(x)
```



The plot shows the differences between expected and observed values. The height of the bar shows the absolute value of this difference, while its position shows the sign of the difference. It is clear that among fair-haired people blue eyes are overrepresented and brown eyes are underrepresented.

Let us examine a further example. A large group of statistical epidemiologists gathered for a banquet. The next morning, many woke up with symptoms of food poisoning. Because they were statistical epidemiologists, they decided to remember what each of them ate at the banquet, and thus determine what was the cause of the illness. The gathered data take the following format:

```
> tox <- read.table("data/otravlenie.txt", h=TRUE)
> head(tox)
  ILL CHEESE CRABDIP CRISPS BREAD CHICKEN RICE CAESAR TOMATO
1   1      1       1      1     2       1    1      1      1
2   2      1       1      1     2       1    2      2      2
3   1      2       2      1     2       1    2      1      2
4   1      1       2      1     1       1    2      1      2
5   1      1       1      1     2       1    1      1      1
6   1      1       1      1     1       1    2      1      1
  ICECREAM CAKE JUICE WINE COFFEE
1        1    1     1    1      1
2        1    1     1    1      2
3        1    1     2    1      2
4        1    1     2    1      2
5        2    1     1    1      1
```




```
6        2    1     1    2      2
```

The first variable (`ILL`) tells whether the participant got sick or not (1 or 2 respectively); the remaining variables correspond to different foods.

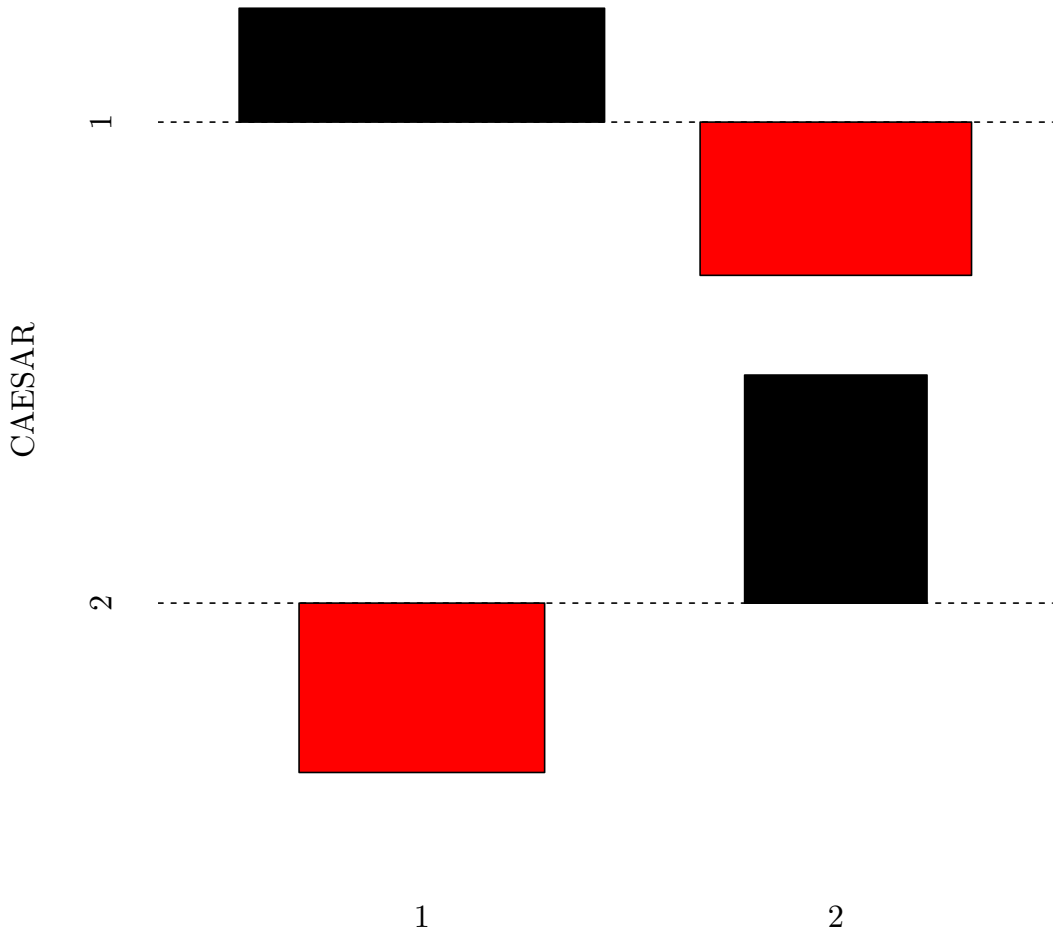
A simple glance at the data will not reveal anything, as the banquet had 45 participants and 13 different foods. Therefore, statistical methods must be used. Since the data are nominal, a contingency table can be used:

```
> for (m in 2:ncol(tox))
+ {
+ tmp <- chisq.test(tox$ILL, tox[,m])
+ print(paste(names(tox)[m], tmp$p.value))
+ }
[1] "CHEESE 0.840899679390882"
[1] "CRABDIP 0.94931385140737"
[1] "CRISPS 0.869479670886473"
[1] "BREAD 0.349817724258644"
[1] "CHICKEN 0.311482217451896"
[1] "RICE 0.546434435905853"
[1] "CAESAR 0.000203410168460333"
[1] "TOMATO 0.00591250292451728"
[1] "ICECREAM 0.597712594782716"
[1] "CAKE 0.869479670886473"
```

```
[1] "JUICE 0.933074267280188"
[1] "WINE 0.765772843686273"
[1] "COFFEE 0.726555246056369"
```

A `for()` loop allows us not to write the code for the test 13 times. The result is that two foods exhibit significant associations with illness—Caesar salad and tomatoes. We can examine these with an association table:

```
> assocplot(table(ILL=tox$ILL, CAESAR=tox$CAESAR))
```



A very similar situation is seen with tomatoes. The culprit is identified! Almost. After all, it's unlikely that both dishes were contaminated. Now we must try to determine what was the main cause of the food poisoning. We will return to this subject in our introduction to logistic regression.

* * *

Besides contingency table-based methods like the chi-squared test, nominal and ordinal data can be analyzed with various more specialized methods. For example, to compare results of expert evaluations, concordance tests are commonly used. Among these tests are the Cohen test, which calculates the Cohen's kappa, a measure of agreement ranging from 0 to 1, and also calculates the p-value for the null hypothesis that kappa=0. An example: in 2003, two groups independently and approximately at the same time investigated an island in the White Sea. The goal was to compile a comprehensive list of all the plant species present on the island. The data were binary (0 means the species is absent on the island, 1 means it is present). The results are recorded in the data table `pokorm03.dat`:

```
> pok <- read.table("data/pokorm_03.dat", h=TRUE, sep=";")
> library(concord)
> cohen.kappa(as.matrix(pok))
Kappa test for nominally classified data
2 categories - 2 methods
kappa (Cohen) = 0.718855 , Z = 8.56608 , p = 0
kappa (Siegel) = 0.67419 , Z = 7.11279 , p = 5.68656e-13
kappa (2*PA-1) = 0.761658
```

Kappa is close to 1 (0.718...), and the probability of the null hypothesis is zero. This means that we can consider the results of the investigations to be in concordance with each other.

Exercise. The file `prorostki.txt` contains results of an experiment examining germination of seeds infected with different types of fungi. In all, three fungi were tested, 20 seeds were tested for each fungus, and therefore with the controls 80 seeds were tested. Do the germination rates of the infected seeds differ from the controls?

7.4 Analysis of correlations

The measure of linear correlation between two variables is the Pearson correlation coefficient r. The absolute value of the correlation coefficient can vary from 0 to 1. A correlation coefficient of 0 means that the values of one variable are unconnected with the values of the other variable. A correlation coefficient of 1 is evidence of a linear relationship between the two variables. A positive value of r means the correlation is positive (the higher the value of one variable, the higher the value of the other), while negative values mean the correlation is negative (the higher the value of one, the lower of the other). The degree of dependence between the variables is reflected by the coefficient of determination: this is the correlation coefficient squared.

The correlation coefficient characterizes the *extent* of the linear relationship between the variables. Two variables can be closely correlated, but if the relationship is not linear but for example parabolic, the correlation coefficient will be close to 0. An example of such a relationship could be the relationship between a person's energy level and their ability to solve math problems. A person in a state of low energy (for example, falling asleep) or a state of very high energy (for example, excited by watching a football game) will both have a much lower ability to solve math problems than a person at an intermediate energy level. For this reason, before evaluating the relationship quantitatively, one must examine it graphically. The best option here is a scatterplot, created in R with the command `plot()` with two vector arguments.

It is also important to note that the discussion here deals with the *existence* and *strength* of a correlation between variables, not nature of this relationship. If we find a significant correlation between variables, this could mean that A depends on B, B depends on A, A and B depend on each other, or A and B depend on a third variable C but have no relation to each other.

A famous example is the correlation between ice cream sales and fires. It would be strange to suggest that eating ice cream causes people to be negligent and start fires, or that experiencing fires causes people to buy ice cream. In fact, both of these parameters depend on air temperature.

To calculate the correlation coefficient in R, the function `cor()` is used

```
> cor(5:15, 7:17)
[1] 1
> cor(5:15, c(7:16, 23))
[1] 0.9375093
```

In the simplest case, it is given two arguments (vectors of equal length). It can also be called with one argument if using a matrix or data table. In this case, the function `cor()` calculates a correlation matrix, composed of correlation coefficients between all pairs of data columns.

```
> cor(trees)
           Girth    Height    Volume
Girth  1.0000000 0.5192801 0.9671194
Height 0.5192801 1.0000000 0.5982497
Volume 0.9671194 0.5982497 1.0000000
```

If the numbers of observations in the columns are unequal (i.e. some columns are missing data), the parameter use in the command cor() can be used. Its default setting is all.obs, which results in an error message when any data points are missing. If use is set to complete.obs, observations with missing data are automatically excluded.

8 Choosing right method

Data			One group	Two groups: differences	Two groups: relations	Three and more groups: relations	Three and more groups: general picture
Measurement	Parametric	Independent	summary()	t.test()	cor.test(, method="pe")	oneway.test(), paiwise.t.test(), anova(), lm()	lda(), manova()
		Dependent		t.test(..., paired = TRUE)		–	–
	Non-parametric	Independent		wilcox.test()	cor.test(, method="sp")	kruskal.test()	pca(), tree(), cor(), hclust(), isoMDS(), cmdscale()
		Dependent		wilcox.test(..., paired = TRUE)		–	–
Categorocal or ranked	Non-parametric	Independent		chisq.test(), prop.test(), binom.test()	glm(..., "binomial")	–	cor(), dist(), hclust(), isoMDS(), corresp()
		Dependent		mcnemar.test()	–	–	–

9 Essential commands

`?` Help

`<-` Assign right to left

`[` Select part of object

`$` Call list element by name

`abline()` Addition to the graph: line from linear regression model

`anova()` Analysis of variation

`as.character()` Convert to text

`as.numeric()` Convert to number

`boxplot()` Boxplot

`c()` Concatenate into vector

`cbind()` Concatenate columns into matrix

`chisq.test()` Chi-squared test

`cor()` Correlation of multiple variables

`colSums()` Sum every column

`cor.test()` Correlation test

`data.frame()` Make data table

`dotchart()` Replacement for "pie" graph

`download.file()` Take file from Internet

`example()` Call example of command

`file.show()` Show file

`function()` Make new function

`head()` Show first rows of data table

`help()` Help

`hist()` Histogram

`legend()` Addition to the graph: legend

`length()` Length of variable

`lines()` Addition to the graph: lines

`lm()` Linear model

`log()` Natural logarithm

`max()` Maximal value

`mean()` Mean

`median()` Median

`min()` Minimal value

`NA` Missed value

`names()` Show names of elements

`nrow()` How many rows?

`order()` Create order of objects

`plot()` Graph

`points()` Addition to graph: points (dots)

`predict()` Predict values

`q()` Quit R

`qqnorm(); qqline()` Check for the normality: graph

`rbind()` Concatenate into matrix by rows

`read.table()` Read data file

`rep()` Make the sequence of same elements

`sample()` Random selection

`savehistory()` Save history of commands

`scale()` Make all variables comparable

`sd()` Standard deviation

`source()` Run script

`str()` Structure of object

`summary()` Main descriptional statistics

`t()` Transpose matrix (rotate on right angle)

`t.test()` Student test (t-test)

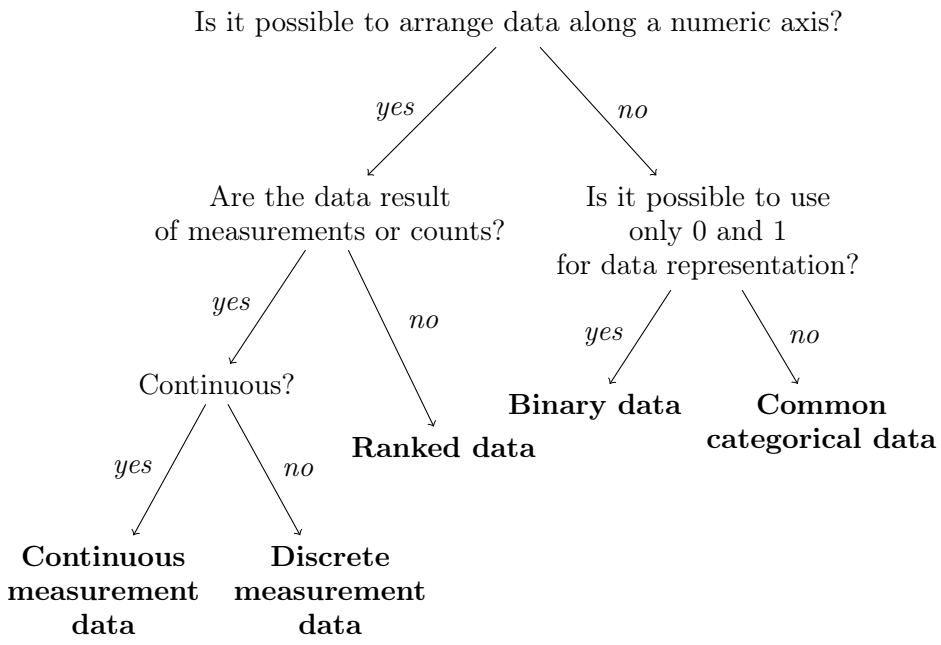
`table()` Make contingency table

`text()` Addition to graph: text

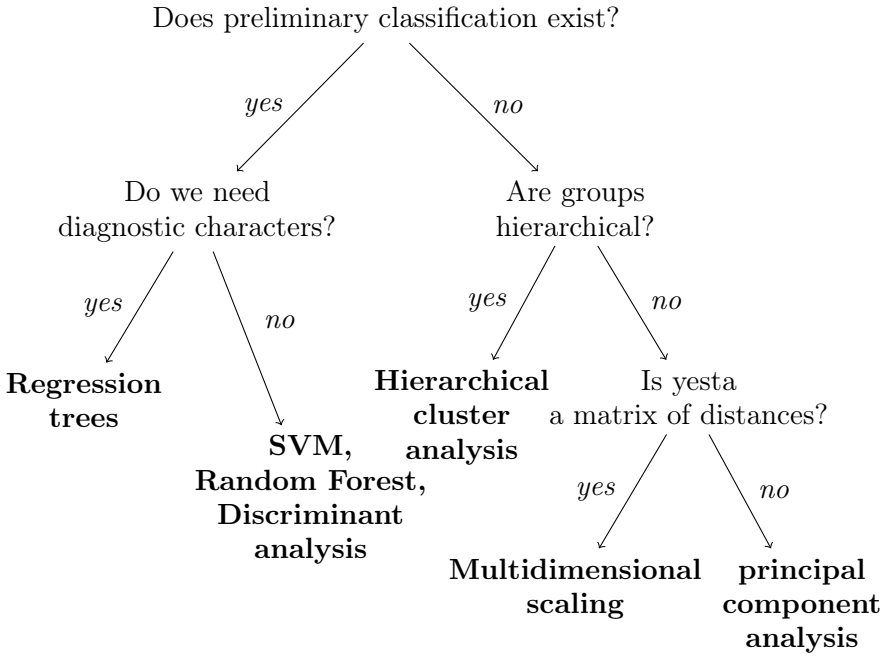
`wilcox.test()` Wilcoxon and Mann-Whitney tests

`write.table()` Write object to disk

10 Types of data



11 Multivariate methods



12 Example of R session

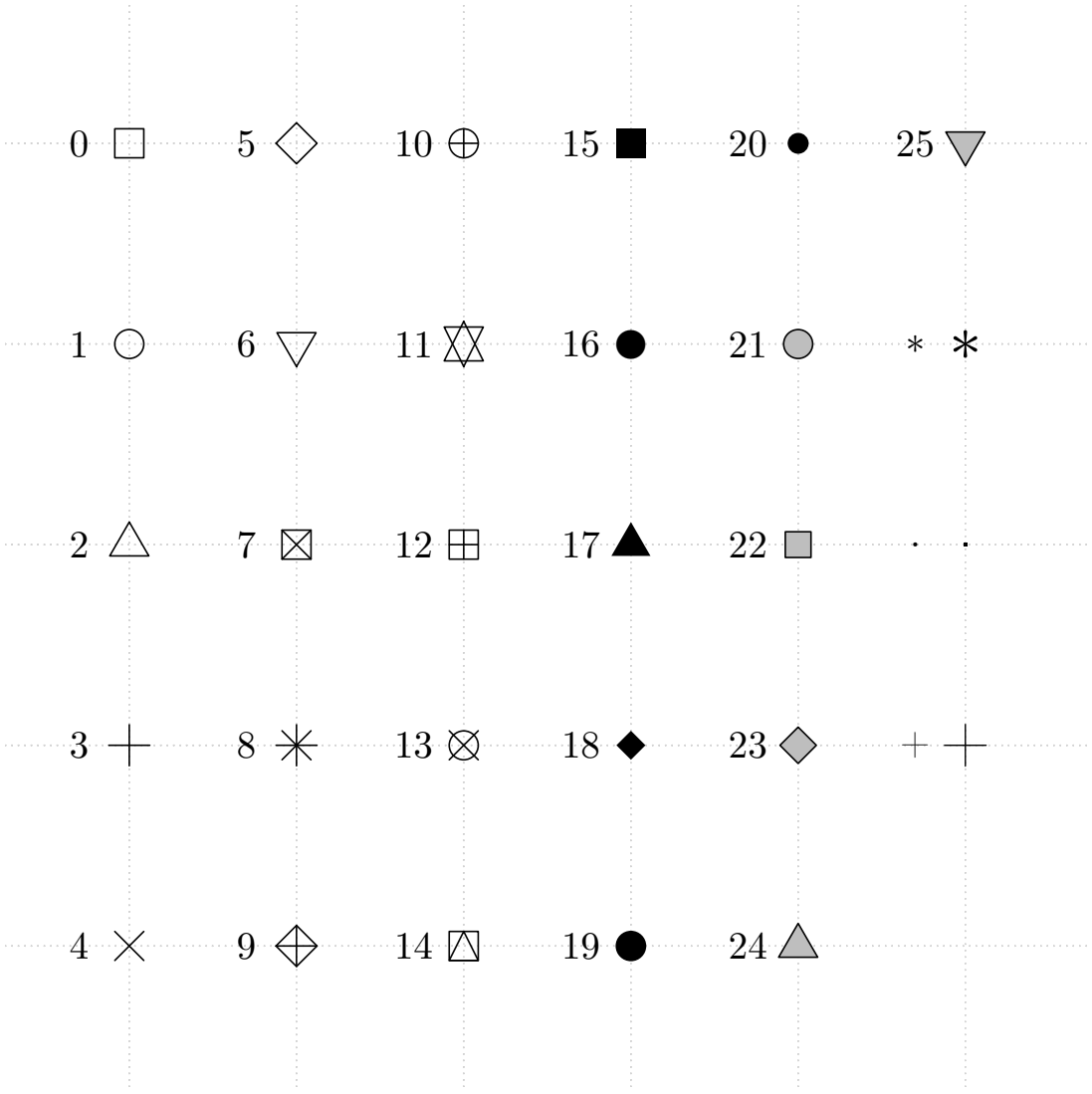


Figure 7: Numbers of symbols in standard R plots