Statistical methods for archaeological data analysis

Martin Hinz

Contents

Chapter 1

Preface

Hallo Welt!

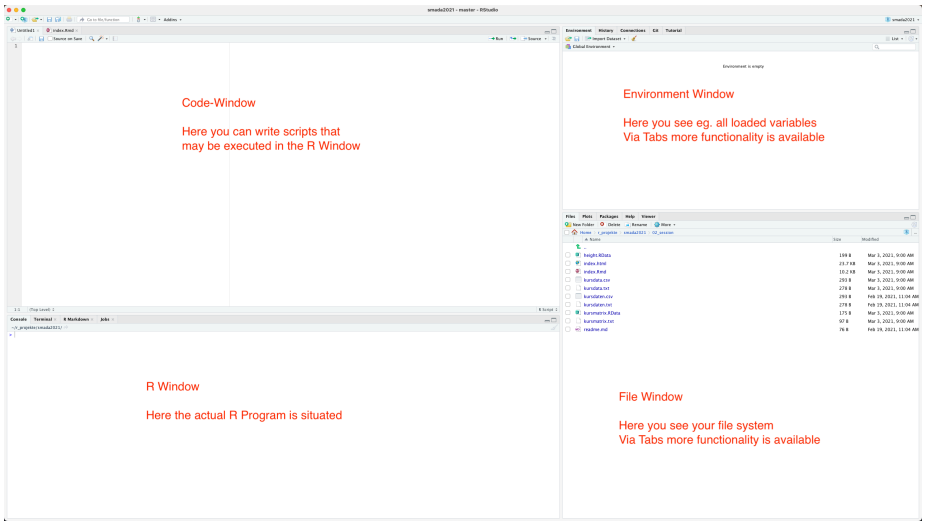
Chapter 2

Introduction into R

2.1 Start R-Studio

When we first start R Studio, we see a screen divided into several windows. On the left-hand side, directly after the start, we are greeted by the large R window, the Console. This is where the actual R programme is located. On the right, there are windows that provide further helpful functions. In the upper area we have the window in which we can see the working environment. On the one hand, there is the actual environment, marked by the tab 'Environment'. Next to this, perhaps of interest to us at the moment, is the 'History' tab, in which we can see the sequence of commands entered so far. The file manager is located in the lower right-hand corner. Other tabs contain information about diagrams (plots), packages and a window in which we can use the R help system.

One important window is still missing: the code or script window. This only appears when we open a new R file. To do this, either click on the plus symbol at the top left or select 'File -> New File' from the menu. This opens another window which is placed in the top left by default and in which you enter your programme code for the analyses. This window functions as a normal text editor window, i.e. if you press Enter here, the text is not directly executed, but a new line is created. To actually execute a command, you can either click on the Run symbol in the upper area or use the keyboard shortcut Control Enter.



2.2 Using R

2.2.1 Start of the system:

After R is started, you end on the prompt.

```
>
```

This prompt expects your commands. It can be used to directly enter commands or conduct calculations like with a normal calculator. We mainly will not use R in this way.Most of the real work is done using the script window. But we can start trying out our directly using the console window.

2.2.2 Simplest way to use: R as calculator

As R is an statistical program, of course it can do calculations. We can try that out by entering some basic calculations using the well-known mathematical operators.

```
2+2
```

```
## [1] 4
```

```
2^2
```

```
## [1] 4
```

2.2.3 Multiple commands are separated by ;

If we want to enter multiple commands in one line, Either in the console or in the script window, we can separate them by using a semicolon. Each part divided by a; is treated like an individual command and is executed before the next in turn is then executed.

```
(1 - 2) * 3; 1 - 2 * 3
```

```
## [1] -3
```

```
## [1] -5
```

2.2.4 Using functions:

Beside the basic calculations R also offer us the possibility to do more complex calculations. Here we start using functions in R for the first time. Functions are commands that produce a certain output, most of the time requiring a certain input.The input usually is given by writing it in between the rounds brackets that distinguish a function call from a variable which we will see later. Functions can sometimes take more than one parameter these are then divided by, within the round brackets.

In the following example in the first line of "we calculate the square root, In the second example the natural logarithm of 10. If we would like to calculate the living room to the base of 10, we have to specify that using a second parameter.

```
sqrt(2) #square root
```

```
## [1] 1.4142
```

```
log(10) #logarith base e
```

```
## [1] 2.3026
```

```
log(10, 10) #logarith base 10, like log(10, base=10)
```

```
## [1] 1
```

2.3 Getting help

There is a specific function for getting help. Not surprisingly this function is called help. It takes as a parameter the name of the function for which you would like to get some information.

Call of the help function:

```
help(sqrt)
```

Like it even simpler? You can also use the '?' For getting help instead of writing the function name 'help'. The name of the function for which you would like to have help it's written after that '?' .

```
? sqrt
```

You can also search within the help files of R. Research capabilities are a limited only a fulltext search is conducted and you will not get any semantic relevant results. This means that if you would like to search for a specific topic, you probably already should know basically what you are searching for. More complicated searches probably better take place in the Internet. There are plenty of sites where you could get help or explanation how certain analyses are conducted.

Searching the help:

```
help.search('logarithm')
```

2.4 Assignment of data to variables

A very essential concept in R is the concept of a variable. Variable can be seen as a kind of labelled drawer or replacement for an actual value that can be changed. It can become quite handy if for example you are writing a script or analyses, In which certain values might be changed in individual runs. Here you can define a replacement for the actual value that is the variable and specify the content of the variable for example in the beginning of the analyses. Here it can easily be changed if necessary.

Setting the value of a variable is also called assignment. If we assign a value to a variable are is not reporting any message back. If we want to see the content of the variable we have to enter this variable itself without any other additions.

There are some data shipped with our. We will talk about datasets later. Some inbuilt constants are the letters of the alphabet, the names of the month and also the value of pi.

```
x <- 2 # no message will be given back

x
```

```
## [1] 2
```

There are some data shipped with our. We will talk about datasets later. Some inbuilt constants are the letters of the alphabet, the names of the month and also the value of pi.

```
pi # build in variable
```

```
## [1] 3.1416
```

When selecting variable names you're quite free to choose. It is necessary, that the name of the variable starts with the letter. You should avoid using mathematical signs, because they could be interpreted as actual calculation. This means, you should not use the minus sign, but you're perfectly free to use the underscore "_" or the dot ".".

2.4.1 Arrow or equal sign?

There are different options for the assignment sign in our. The traditional one is the arrow composed of a 'smaller than' sign and minus sign. Most other programming languages and now also our takes the = as an assignment. What you would like to use as a matter of taste. Personally I'd like the Aero more because it is more speaking and more clear.

Classic assignment symbol in R is the arrow. Also possible:

```
x=2
```

Both are possible.

2.5 Working with variables

And this is helpful to get an overview about which variables we have already Defined. For this in our studio in the right hand area there is the environment window. If we want to get an overview about the assigned variables in our itself, we can use the command `ls()`. Currently there is only one variable in our environment. That is the variable X that we just assigned.

Display of already uses variables:

```
ls()
```

```
## [1] "x"
```

Sometimes it might be helpful to get rid of one of the variables. To do this you can use the `rm()` command. This stands for remove. The name of the variable that has to be deleted is given within the round brackets ending the function call. If we after the removal of a variable get a listing of the variable environment again the variable should have gone.

Delete a variable:

```
rm(x) # no message will be given back
ls()
```

```
## character(0)
```

2.6 Using variables

Already have been said a variable can be used instead of an actual value. To do this we simply replace the use of the value with the name of the variable. For example if we want to use a variable when we calculate 2×2 we can at first assign 2 to one Variable and use it instead of actually writing to in our calculation. An important concept is also that the result of the calculation can also be assigned to a variable. With this we can chain analyses together and use the output of one of the functions as the input of the next function. In our example we assign to to the variable `x`, then we double its value and assign the results to the variable `y`. The result of this calculation is then used to calculate the square roots using the function `sqrt()`.

Calculations with variables:

```
x <- 2
y <- 2 * x
z <- sqrt(y) # no message will be given back
```

No using the function `ls()`, We can't get an overview over our current environment. We should see now the 3 variable that we have created. Additionally if we inspect the individual variables, we shall see that `y` contains the value of four while `z` contains the value of two.

```
ls()
```

```
## [1] "x" "y" "z"
```

```
y
```

```
## [1] 4
```

```
z
```

```
## [1] 2
```

Exercise 2.1 (Calculation of a circle)**.**

Exercise 2.2. Given is a circle with the radius r=5. Calculate the diameter d (2 * r), the circumference u (2 * * r) and the area a (* r^2).

Add area a and circumference u, assign the result to the variable v and delete u and a.

Solution

```
r <- 5
d <- 2 * r
u <- 2 * pi * r
a <- pi * r^2
v <- a + u
rm(u)
rm(a)
```

2.7 Data types in R Variables

There are four main data types in R: Scalars, vectors, matrices, data frames.

2.7.1 Scalar

Scalar are individual values. This can be numbers text strings or true/false values. The essential characteristic is that it is only one value that is represented by a scalar.

Examples of Scalar are all those variables that we used until now.

```
pi
```

```
## [1] 3.1416
```

All these variables stored only one value at the time.

2.7.2 Vector

A vector is a variable that holds multiple values at the time in a one-dimensional data structure. You can't imagine it as a kind of list where every item off the list again is a scalar.

We have already seen an example of a vector: the result of the listing of the variables, resulting from the command `ls()` represents a vector, where every position in this vector holds a scalar information, that is the name of the variable.

```
ls()
```

```
## [1] "d" "r" "v" "x" "y" "z"
```

2.7.3 Matrix:

A vector is a one-dimensional data structure. If we add more dimensions to this idea, we end up with a Matrix. In the simplest implementation you can imagine a matrix as a table with rows and columns. That we have rows and columns represents the two-dimensionality of this data structure. Matrices with more dimensions are easily implementable, although our imagination probably will stop with three dimensions. Most of the time we will use two-dimensional matrices.

As with vectors, each element in a matrix represents a scalar value. One of the specific features of the data type matrix in R is, that all values have to be of the same kind. That means with in one and the same metrics, there can only be numbers, characters, or true and false values at once. We can't mix these types

of information in a matrix, which is the difference from the next data structure that we will learn.

There are also inbuilt matrices in R, for example are matrix holding the transfer rates between different European currencies. Of course these are restoring values and not updated online all the time

```
euro.cross
```

```
##            ATS       BEF       DEM        ESP       FIM       FRF        IEP        ITL        L
## ATS  1.0000000  2.931615 0.1421357  12.091742 0.4320931 0.4767025 0.05723451  140.7142  2.9316
## BEF  0.3411089  1.000000 0.0484838   4.124601 0.1473908 0.1626075 0.01952320   47.9989  1.0000
## DEM  7.0355297 20.625463 1.0000000  85.071811 3.0400035 3.3538549 0.40267508  989.9991 20.6254
## ESP  0.0827011  0.242448 0.0117548   1.000000 0.0357346 0.0394238 0.00473335   11.6372  0.2424
## FIM  2.3143163  6.784684 0.3289470  27.984116 1.0000000 1.1032405 0.13245876  325.6572  6.7846
## FRF  2.0977442  6.149778 0.2981644  25.365382 0.9064207 1.0000000 0.12006336  295.1825  6.1497
## IEP 17.4719769 51.221107 2.4833918 211.266640 7.5495198 8.3289358 1.00000000 2458.5557 51.2211
## ITL  0.0071066  0.020834 0.0010101   0.085931 0.0030707 0.0033877 0.00040674    1.0000  0.0208
## LUF  0.3411089  1.000000 0.0484838   4.124601 0.1473908 0.1626075 0.01952320   47.9989  1.0000
## NLG  6.2441519 18.305449 0.8875170  75.502675 2.6980546 2.9766031 0.35738096  878.6410 18.3054
## PTE  0.0686361  0.201215 0.0097556   0.829930 0.0296572 0.0327190 0.00392835    9.6581  0.2012
```

You can see that we have rows and columns here and both the rows and columns have names. In this example row and column names are the same because we have a special kind of matrix. But in general the names of the rows and columns can differ from each other. Matrices are specific data types with which you can conduct matrix algebra, which is a specific branch of mathematics that is also used in statistics. We will not deal with this very much. That's why we most of the time will probably work more with the next data type.

2.7.4 Data frame:

The fourth of our data types is the data type `data.frame`. Similar to the matrix, this datatype represents a more than one dimensional data storage unit. Different from the matrix, in data frames values of different kinds can be stored. More specifically the different columns of the data frame can differ in respect of the contains data type. That means we can combine columns that have character values with columns that hold numeric values.

Tables in data frames are usually structured in a specific way: the rules usually hold the item off on investigation or the observations, while the columns usually holds the different features or variables of interest.

One example of such a data frame that is inbuilt in our is the data frame `mtcars`. This data frame contains the technical details of different cars. Also this is more a historical dataset.

```
mtcars
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4          21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag      21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710         22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive     21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant            18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360         14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D          24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230           22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280           19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C          17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE         16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL         17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC        15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial  14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128           32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic        30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla     33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona      21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger   15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin        15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28         13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9          27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2      26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa       30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino       19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora      15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E         21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

You can see, that the real names contains the names of the different cars, identifying them. The column names contains different measurements or information is, there are specific for the individual cars. The individual values identified by row and column then holds to specific values that are unique for this individual item or car.

Data frames are the data type that we will use most of the time, especially if we import data from other sources. How we can do that, will be shown subsequently. But at first we have to make sure, that we get our data from the

right location on our computer. For that we need to concept of the working directory.

2.7.5 The working directory

Historically, R is a software that has always been run within the console. Therefore it expects all its inputs from a specific folder on your computer, the working directory. Also, if any output is written to the disk on your computer, this also will take place in the specified working directory.

Of course this working directory is not fixed, but you can specify that according to your specific workflow. At first we can use the command `getwd()` to see where on the computer I will working direct with currently is located.

```
getwd()
```

Then we can use the command `setwd("your/working/directory")` to set this working directory to a specific folder of your computer.

```
setwd("U:\R") # or something else
```

How specific folder has to be addressed, depends on the operating system. While Linux and macOS computers treat directory name is more or less the same, in Windows computers the path is prepared by the volume letter. With RStudio, there are different other options how are you can use the graphical user interface to specify the working directory. This might be more convenient than typing the path, especially if you are not used to it. You will find options for this in the files window of our studio, under the icon 'More', or in the main menu under the item 'session'. Change the path according to your needs. Also you can make it a habit to check in the beginning of every R session, what do you work in directory is and if it is correctly specified.

2.7.6 Download data for further tasks into your working directory

In the reminder of the chapter we will need some files that can be downloaded using the following links:

- height.RData
- kursmatrix.txt

- kursdata.txt
- kursdata.csv

Please save these files to the directory that you have defined as your working directory. In the following the used example will assume that the files are accessible directly, as they should be if they are placed in the working directory.

Remember:

```
getwd()
setwd("my/location/of/my/working/directory")
```

2.8 Data import through reading of files

Data can be imported into R from different formats and sources. The most straightforward version is to directly scan a text file and read it into an R variable. For directly reading in a file we can use the function `scan()`. The file `kursmatrix.txt` is a simple text file in which ages and bodies sizes of individuals are listed consecutively. Scan reads in each item and translate it to a position in a vector.

```
scan("kursmatrix.txt")
```

```
##  [1]  39  34  23  38  23  21  23  31  25  31  24  23  23  39  21 181 170 185 163 175
```

If we, for example, want to turn this factor into a two-dimensional structure, like a matrix, we can use the command matrix to define such a structure and then use as an input the scanned content of the file. For the command `matrix()`, One of its parameters is the content that should be turned into a matrix, the second parameter is the number of columns that this matrix should have in end.

```
kursmatrix <- matrix(scan("kursmatrix.txt"),ncol=2)
```

The result is a two-dimensional structure, with two columns, in which body height and age are listed in different columns.

```
kursmatrix
```

```
##       [,1] [,2]
##  [1,]   39  181
##  [2,]   34  170
##  [3,]   23  185
##  [4,]   38  163
##  [5,]   23  175
##  [6,]   21  163
##  [7,]   23  162
##  [8,]   31  172
##  [9,]   25  172
## [10,]   31  180
## [11,]   24  187
## [12,]   23  158
## [13,]   23  184
## [14,]   39  156
## [15,]   21  168
```

The file `kursdata.txt` contains a more complicated data structure. Here we have information is of different kinds, for example strings, but also numeric values. This kind of data can be imported into an data frame. The most general function to import table data is the function `read.table()`.

```
kursdata <- read.table("kursdata.txt")
```

One of the most widely used text file for exchange of numerical and other data are those in the CSV format. This format comes into flavours, Differentiated by the character that separates the columns. The original CSV format has a column separator "," and a decimal separator using ".". In European and other countries the "," it's often used as decimal separator. Therefore also a CSV2 format exists. Here the column separator is a ";", while the decimal separator is, ". In Switzerland most of the time we will probably use the CSV2 format. In this format we have the same data available like we have in the `kursdata.txt`, the file is now called `kursdata.csv`.

```
kursdata <- read.csv2("kursdata.csv")
kursdata
```

```
##           X age height sex
## 1  Matthias  39    181   m
## 2   Jannick  34    170   m
## 3   Nicolas  23    185   m
## 4    Silvia  38    163   f
## 5      Till  23    175   m
## 6      Anna  21    163   f
## 7    Ilaria  23    162   f
```

```
## 8     Sarah  31    172   f
## 9     Clara  25    172   f
## 10    Alain  31    180   m
## 11   Adrian  24    187   m
## 12   Marlen  23    158   f
## 13  Michael  23    184   m
## 14   Helena  39    156   f
## 15  Nephele  21    168   f
```

If we read in the data like this, you will realise, that there is a numeric naming, that is automatically given by R. If the dataset already consists of a unique identifier, that is a value, that is not repeated within the whole dataset, and that uniquely identify every individual item of the dataset, this can be used instead of the numeric identifier. This unifier of individual items is called row names in R. So if we specify in the read.CSV2 command, that we want to use for example the first column as row names, we can do it like this.

```
kursdaten <- read.csv2("kursdata.csv",row.names = 1)
kursdaten
```

```
##          age height sex
## Matthias  39    181   m
## Jannick   34    170   m
## Nicolas   23    185   m
## Silvia    38    163   f
## Till      23    175   m
## Anna      21    163   f
## Ilaria    23    162   f
## Sarah     31    172   f
## Clara     25    172   f
## Alain     31    180   m
## Adrian    24    187   m
## Marlen    23    158   f
## Michael   23    184   m
## Helena    39    156   f
## Nephele   21    168   f
```

2.9 Using c() for data entry

Now we know how we can assign more complicated data sets two variables by loading them from the file system. Sometimes, it might also be necessary, to

directly assign more than one value to a variable. Let's start with the example of a vector. A vector is created in R using the command `c()`. This 'c' stands for combine, and enables us to combine multiple values to be assigned to a variable, but also for different purposes.

Let's assume that we would like to make a vector of different Bronze Age sites. We assign the result to a variable called places.

```
places <- c("Leubingen", "Melz", "Bruszczewo")
```

As in every other situation, in R actual value can be replaced with a variable. Also when we combine values we can not only combine actual values, in this case strings, but we also could use variables and combined them with other variables. To demonstrate that let's make another vector of side categories that we call categories.

```
categories <- c("burial", "depot", "settlement")
categories
```

```
## [1] "burial"     "depot"      "settlement"
```

Now we can combine these two factors into one.

```
c(places, categories)
```

```
## [1] "Leubingen"  "Melz"       "Bruszczewo" "burial"     "depot"      "settlement"
```

2.10 Named vectors

We already learnt to concept of row names and column names. Also places in a vector can have a specific identifier, the name. Since vectors do not have rows and columns, this feature is called only called 'name'. We can use another vector to assign names, or we could directly enter names for the individual positions. In this case we use our category vector as base vector and the sites in the places vector as identifiers.

```
names(categories) <- places
categories
```

```
##    Leubingen         Melz   Bruszczewo
##     "burial"      "depot" "settlement"
```

The result is a vector, in which every position has the name of the site is unique identifier, and where the values are the site categories for this specific archaeological sites.

2.11 Applying functions to more complex variables

Also variables with more complex content can, of course, be used in calculations and other functions. Due to their nature, and the fact that they contain more than one value, this of course changes the range of functions that can be applied to them. I will demonstrate that with a reduced version of our data. We will use only a vector of the body height of the individuals.

For this we explore a way of loading data into R. This time we use the need to data storage option of R. This format is called 'RData', and different from other loading or saving, we do not have to specify a variable name. In this case the variable is stored with its content, and if we load this dataset again, the variable is restored with the same name.

```
load("height.RData")
height
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara
##      181      170      185      163      175      163      162      172      172
```

Now we can use this vector that is assigned to the variable name height, to demonstrate some functions that make calculations over all the values that are stored in this vector. The first step probably comes to mind, is to sum up all the values. This can be done in R own using the function `sum()`.

```
# Sum:
sum(height)
```

```
## [1] 2576
```

We can also count the number of values in the vector. The command for this is `length()`.

```
# Count:
length(height)
```

```
## [1] 15
```

If we have the number of cases, and to some of their individual values, we easily can calculate the arithmetic mean.

```
# Mean:
sum(height)/length(height)
```

```
## [1] 171.73
```

Since this is a very essential statistical value or parameter, of course there exists a specific command for this in R. There is no big surprise that this function is called `mean()`.

```
# Or more convenient:
mean(height)
```

```
## [1] 171.73
```

Other possible functions might for example be related to the order and the extremes of the values within our dataset. We can sort the dataset according to the values, using the function `sort()`. In case of numerical values, the items will be sorted according to the numerical order. In case of characters, the items will be sorted according to the character. Our height data on numerical, therefore we will get them sorted from the smallest to the largest person.

```
# sort:
sort(height)
```

```
##   Helena   Marlen   Ilaria   Silvia     Anna  Nephele  Jannick    Sarah    Clara     Till    A
##      156      158      162      163      163      168      170      172      172      175
```

Immediately we can identify the smallest and the largest person. But we can also explicitly get the values using the function `min()` for minimum, and `max()` for maximum. The function `range()` gives both values at the same time.

```
# minimum:
min(height)
```

```
## [1] 156
```

```
# maximum:
max(height)
```

```
## [1] 187
```

```
# Or both at the same time:
range(height)
```

```
## [1] 156 187
```

2.12 Calculations with vectors

Not only can we use functions on more complex variables like vectors, we also can do calculations. If, for example, we combine a scalar value With a mathematical expression with a vector, the calculation is done at every position of this vector. For example, if we want our height vector in metre, we have to divided by 100. We can directly apply this calculation to the whole variable, and the results will change every individual position in that vector. That means, we divide the variable by 100, and all the items in the variable are then divided by 100, causing every value to be in meter instead of centimeter.

```
height.in.m <- height/100
height.in.m
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara
##     1.81     1.70     1.85     1.63     1.75     1.63     1.62     1.72     1.72
```

The case is different if we combine to vectors with a mathematical expression. In this case, the first value of the first vector is combined with the first value of the second vector.The second value of the first vector is then combined with the second value of the second vector, and so forth.

```
test<-c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
height.in.m + test
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara
##     2.81     3.70     4.85     5.63     6.75     7.63     8.62     9.72    10.72
```

In case, that we have different number of positions in the individual factors vectors, the short one is "recycled". That means, it starts again from the beginning. You can try that out yourself, if you take the example above and remove some items from the test vector.

Exercise 2.3 (Data collection lithics)**.**

Exercise 2.4. An excavation produced the following numbers of flint artefacts:

flakes	blades	cores	debris
506	104	30	267

Assign the values to a named vector, calculate the proportion of the artefacts and sort the vector according to their percentage

During the data collection on box with artefacts was missing, the following numbers has to be added to the vector:

flakes	blades	cores	debris
52	24	15	83

Moreover were 10 items each artefact type missing. Make a vector for the box, add it and the 10 missing to the original data and repeat the calculations.

Solution

```
artefacts <- c(506, 104, 30, 267)
names(artefacts) <- c("flakes", "blades", "cores", "debris")

prop <- artefacts/sum(artefacts)
sort(prop)
```

```
##    cores   blades   debris   flakes
## 0.033076 0.114664 0.294377 0.557883
```

```
missing_box <- c(52,24,15,83)
all_artefacts <- artefacts + missing_box + 10

prop <- all_artefacts/sum(all_artefacts)
sort(prop)
```

```
##    cores   blades   debris   flakes
## 0.049063 0.123104 0.321142 0.506690
```

Variant:

We also could have over written the content of the artefact variable with the new values including the missing box and the 10 additional items. In that case the court would look like this:

```
artefacts <- artefacts + missing_box + 10

prop <- artefacts/sum(artefacts)
sort(prop)
```

```
##    cores   blades   debris   flakes
## 0.049063 0.123104 0.321142 0.506690
```

You see, that artefact is twice present in the first line. This is possible, because the right-hand side of the assignment is evaluated first, and then the result is assigned to the actual variable.

This technique can also be used in actual scripts if you don't need the intermediate values of the variable. It can become quite handy, to reduce the amount of variables and doing names. But you always will have to take care: you lose the intermediate values! So if you have to repeat any step in between, or later you would need some of the intermediate values you will not have them.

2.13 Sequences and repeated data

Now we have seen, how we can produce vectors ourselves, and how we can use them in calculations. There are some specific vectors, either consisting of the repetition of an individual value, or sequences of values. There are some inbuilt functions in R that can help you producing these kinds of vectors fast.

Let's start with a simple sequence. Let's assume, that we need the values from 1 to 10. We can produce such a simple sequence rather easily like this:

```
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

But also more complicated sequences are possible. For this we need an explicit function call for the function seq(). This command takes several parametres, the first one is the starting value, the second one the end value. You can also define the increment using the parameter by, or the desired length of the resulting vector, using the power meter length.

```
seq(1,10,by=2)
```

```
## [1] 1 3 5 7 9
```

```
seq(1,20,length=5)
```

```
## [1]  1.00  5.75 10.50 15.25 20.00
```

You can check out other options and use cases indeed help documentation for this command.

The other mentioned option, the repetition, works for letters as well as for numeric values. The command here is `rep()`. Here, the first parameter is the value that should be repeated. This value can also be a vector. The second para meter is the number of times, that this value should be repeated. Also hear further options can be found in the documentation of the command.

```
rep(1,10)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1
```

```
rep(1:3,3)
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

```
rep(c("Anton","Berta","Claudius"),3)
```

```
## [1] "Anton"    "Berta"    "Claudius" "Anton"    "Berta"    "Claudius" "Anton"    "Berta"    "C
```

2.14 Data access

2.14.1 by index/position

And important possibility is to access data with in such a complex structure like for example a vector. By convention, for accessing data in R, square brackets are used. Indicates of a one-dimensional data structure, within the brackets you can give the position of the item that you would like to access. This can be an individual number, a vector of numbers, Or, by using the minus sign, you can also exclude eighter individual value or a range of values. Here, sequences can become very handy.

```
height[1]
```

```
## Matthias
##      181
```

```
height[5]
```

```
## Till
##  175
```

```
height[1:3]
```

```
## Matthias  Jannick  Nicolas
##      181      170      185
```

```
height[-(1:3)]
```

```
##  Silvia    Till    Anna  Ilaria   Sarah   Clara   Alain  Adrian  Marlen Michael  Hel
##     163     175     163     162     172     172     180     187     158     184
```

If we have a named vector, like for example with our heigth data, these positions have also and unique identifier. In that case, we can also use the unique identifier, to access a specific position in our data storage vector.

```
height["Clara"]
```

```
## Clara
##   172
```

This data access is two ways: not only can we get the values at a specific position, but we can also change the values, given that we indicate a specific position in the vector. In the following example at first the content of the vector height is shown, then we change the entry in the first value, and you can inspect the effect.

```
height
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara
##      181      170      185      163      175      163      162      172      172
```

```
height[1] <- 168
height
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara    Alain   Ad
##      168      170      185      163      175      163      162      172      172      180
```

Of course the same is true for the access by name.

```
height["Till"] <- 181
height
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara    Alain   Ad
##      168      170      185      163      181      163      162      172      172      180
```

2.15 Logical values

Until now we had only vectors or other variables that stored either numeric values or strings. No we learn another category of data type: the logical values. These are also called binary, boolean, or true/false values. These values can result from inequations or checks:

```
pi>4
```

```
## [1] FALSE
```

```
height > 175
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara    Alain   Ad
##    FALSE    FALSE     TRUE    FALSE     TRUE    FALSE    FALSE    FALSE    FALSE     TRUE
```

but you can also enter them yourself. Logical values are entered as 'TRUE' or 'FALSE'. But there is also a shortcut, 'T' or 'F' would be enough.

```
logic_test <- c(T,F)
logic_test == T
```

```
## [1]  TRUE FALSE
```

```
logic_test == F
```

```
## [1] FALSE  TRUE
```

Above you can also see another specific way of how an equation sign is used in our in a comparison. In this situation, two '=' are used to distinguish it from the assignment situation.

Comparisons, and the resulting logical values, can become very helpful when selecting specific values in a dataset. For example, if you want to select all the individuals that are larger than 1 m 75, you can do that by including a comparison in the square brackets used for accessing data. You can also use the command `which()` to identify in which cases a certain comparison would be true. Lastly, logical values are internally sorted as 0 and 1, and can therefore also be used in calculations or counts. For example, if we want to identify, how many percent of our individuals are larger than 1 m 75, we can sum the results from this comparison. In case that this comparison would return true, it would also return one. By summing up the ones, we get a count. Dividing the count by the number of cases, we get the percentage.

```
height[height>175]
```

```
## Nicolas    Till   Alain  Adrian Michael
##     185     181     180     187     184
```

```
which(height>175)
```

```
## Nicolas    Till   Alain  Adrian Michael
##       3       5      10      11      13
```

```
sum(height>175)/length(height)
```

```
## [1] 0.33333
```

2.16 Factors

The last type of information are factors. A factor is a codified textual information that is within a very specific range of values. An example for a factor might be the sex of an individual. From the biological determination, this can result in male, female, or undetermined. This means we have only three values. The difference between a factor variable and character variable is, that internally the values are stored as numbers. The table translates then the number to the actual textual representation.

```
sex <- factor(c("m", "m", "m", "f", "m", "f", "f",
                "f", "f", "m", "m", "f", "m", "f", "f"))
sex
```

```
##  [1] m m m f m f f f f m m f m f f
## Levels: f m
```

Another specific feature of factor variables is that they can also represent ordered values. We might see this later.

2.17 missing (NA) values

Missing values are annoying in every kind of investigation. They have to be treated in a specific way, distinguishing them from the situation where the value is zero. If we have a value that is zero, this means we have information that the value is actually zero.

In our example you can see the effect. If we set the height of an individual person to 0, and then calculate the mean, we get the wrong result.

```
height["Marlen"] <- 0

mean(height)
```

```
## [1] 160.73
```

```
sum(height)/14
```

```
## [1] 172.21
```

So this can cause problems, if we would use the 0 as an encoding for missing information. For this purpose there is a specific value called 'not available' or NA. If we set the value of an individual item to not available NA, and then calculate the mean, the result is NA. This is a warning sign, that in the dataset there are missing cases. We can use the parameter `na.rm=T`, read NA remove it's true, to ignore all the NAs and to conduct the calculation of the mean value. This is true for a lot of other functions.

```
height["Marlen"] <- NA

mean(height)
```

```
## [1] NA
```

```
mean(height, na.rm=T)
```

```
## [1] 172.21
```

2.18 Matrices

We initially have already talked about the matrices, Two or more dimentional data storage, which also can be used in mathematical procedures. This of course is only true, if the matrix contains numerical values only. And, as we have already seen, do matrices also have names. Since we talk about more dimensional objects, we have to be specific, about which names we talk. That is because in the case of matrices, but also in the case of data frames, we talk about row names and column names.

We already have loaded the information about people in the form of the kursmatrix.

```
kursmatrix
```

```
##       [,1] [,2]
##  [1,]   39  181
##  [2,]   34  170
##  [3,]   23  185
##  [4,]   38  163
##  [5,]   23  175
##  [6,]   21  163
##  [7,]   23  162
##  [8,]   31  172
##  [9,]   25  172
## [10,]   31  180
## [11,]   24  187
## [12,]   23  158
## [13,]   23  184
## [14,]   39  156
## [15,]   21  168
```

This is already in the conventional representation: the rows contain information about a specific item, the columns contain each specific variable. To make this more clear, we should assign row and column names. Also here, like with the names for vectors, we can use either variables or actual values.

```
rownames(kursmatrix) <- names(height)
colnames(kursmatrix)<-c("age", "height")
kursmatrix
```

```
##          age height
## Matthias  39    181
## Jannick   34    170
## Nicolas   23    185
## Silvia    38    163
## Till      23    175
## Anna      21    163
## Ilaria    23    162
## Sarah     31    172
## Clara     25    172
## Alain     31    180
## Adrian    24    187
## Marlen    23    158
## Michael   23    184
## Helena    39    156
## Nephele   21    168
```

Like with vectors, mathematical operations are possible with matrices. Actually that is a their prime purpose. For example, we can divide a metrics by 100 or any other scalar value. The result will be a matrix, in which every individual value is divided by this scalar, in the specific case 100.

```
kursmatrix / 100
```

```
##           age height
## Matthias 0.39   1.81
## Jannick  0.34   1.70
## Nicolas  0.23   1.85
## Silvia   0.38   1.63
## Till     0.23   1.75
## Anna     0.21   1.63
## Ilaria   0.23   1.62
## Sarah    0.31   1.72
## Clara    0.25   1.72
## Alain    0.31   1.80
## Adrian   0.24   1.87
```

```
## Marlen   0.23   1.58
## Michael  0.23   1.84
## Helena   0.39   1.56
## Nephele  0.21   1.68
```

We can also access individual values within a matrix. This is done in the same way like with vectors. So either, using the position in the form of a number, or by name. Since now we have a more dimensional data object, we also have more dimensions to specify, if we would like to access a specific value. In the case of a two-dimensional matrix, for example, we have to give two positions to identify a specific value. These positions are separated by a comma. General, rows are the first dimension, while columns are the second dimension in our. So rows first, Callums second is a rule, that is applicable for a lot of other situations.

If we specify only one of the positions, we refer to either the whole column, or the whole row. The result is then again a vector. Also on this selection, like on every other vector, we can apply mathematical operations.

```
kursmatrix[, 1] / 100
```

```
## Matthias  Jannick  Nicolas   Silvia     Till     Anna   Ilaria    Sarah    Clara
##     0.39     0.34     0.23     0.38     0.23     0.21     0.23     0.31     0.25
```

Also in this case, if we combine a matrix with a vector, the same logic is to like if we combine to vectors. So if we combine a mattress and a vector, Every value of the vector is combined with every value of the Matrixx starting with the first vector within the matrix. If we combine a matrix and the matrix, then the first value in the first column of the first matrix is combined with the first value of the first column in the second matrix, and so on, equivalent to the way in which vectors are combined.

```
kursmatrix / c(1:15, rep(2, 15))
```

```
##              age height
## Matthias 39.0000   90.5
## Jannick  17.0000   85.0
## Nicolas   7.6667   92.5
## Silvia    9.5000   81.5
## Till      4.6000   87.5
## Anna      3.5000   81.5
## Ilaria    3.2857   81.0
## Sarah     3.8750   86.0
## Clara     2.7778   86.0
## Alain     3.1000   90.0
## Adrian    2.1818   93.5
```

```
## Marlen    1.9167   79.0
## Michael   1.7692   92.0
## Helena    2.7857   78.0
## Nephele   1.4000   84.0
```

To get a feeling for these rules, it is best that you try out different combinations, and observe the results.

2.19 Data frames

The last of the major data types, that we have already seen, is the data frame. A data frame results either from the import of a CSV file, or it can be created on the spot in R by combining different vectors in a more dimensional table. These factors also can come from a matrix. For this we used to command `data.frame()`, which constructs a data frame. The columns are their names are given in this construction, and their values are assigned with an = after this. You aware, that we do not assign actually in this example a variable age with the values of the matrix, but only a column within the data frame. That is one of the reasons, why the syntax using the assignment arrow is more clear, because it differentiate from this construction of the data frame.

```
kursdata <-
  data.frame(age = kursmatrix[,1],
             height = kursmatrix[,2],
             sex=sex)
kursdata
```

```
##          age height sex
## Matthias  39    181   m
## Jannick   34    170   m
## Nicolas   23    185   m
## Silvia    38    163   f
## Till      23    175   m
## Anna      21    163   f
## Ilaria    23    162   f
## Sarah     31    172   f
## Clara     25    172   f
## Alain     31    180   m
## Adrian    24    187   m
## Marlen    23    158   f
## Michael   23    184   m
## Helena    39    156   f
## Nephele   21    168   f
```

Also in the case of a data frame, very similar to the situation with the matrix, we can access individuals rows or columns by either index or name. In case of the data frame there is specific notation for accessing the content of a specific column: for this we can use the $.

```
kursdata[,"age"]
```

```
##  [1] 39 34 23 38 23 21 23 31 25 31 24 23 23 39 21
```

```
kursdata$age
```

```
##  [1] 39 34 23 38 23 21 23 31 25 31 24 23 23 39 21
```

Like with matrices, we can use data frames in calculations. Since in a DataFrame also non-numerical values can be stored, this is not always make sense. But we can use the notation above to specify individual columns and assign calculations to them.

Additionally a very useful command can be the command `summary()`. This gives you a summary of the individual columns of data frame, but can also be used with other objects in R. The way in which the summary is conducted might depend on the specific object.

```
kursdata$height / 100
```

```
##  [1] 1.81 1.70 1.85 1.63 1.75 1.63 1.62 1.72 1.72 1.80 1.87 1.58 1.84 1.56 1.68
```

```
summary(kursdata)
```

```
##       age           height    sex
##  Min.   :21.0   Min.   :156   f:8
##  1st Qu.:23.0   1st Qu.:163   m:7
##  Median :24.0   Median :172
##  Mean   :27.9   Mean   :172
##  3rd Qu.:32.5   3rd Qu.:180
##  Max.   :39.0   Max.   :187
```

```
tapply(kursdata$height, kursdata$sex, mean, na.rm=T)
```

```
##      f      m
## 164.25 180.29
```

The last line in this piece of code above is an example, how are you can use this $notation very handy in function calls. This example applies to the vector height in the dataset the calculation of the mean, differentiated by the sex, and also ignores potentially NA values. This kind of notation is very close to what you probably will use later on a lot in your actual analyses.

There are several datasets inbuilt in R that can be used for experimentation or testing out certain functionalities. You can get a list of this using the command `data()`. The resulting list might be very long, its length depends on the number of packages that you have installed. The list below only serves as an example. You have to try it out yourself, if you want to have the full list.

```
data()
```

```
Data sets in package 'datasets':

AirPassengers           Monthly Airline Passenger Numbers 1949-1960
BJsales                 Sales Data with Leading Indicator
BJsales.lead (BJsales)
                        Sales Data with Leading Indicator
BOD                     Biochemical Oxygen Demand
CO2                     Carbon Dioxide Uptake in Grass Plants
ChickWeight             Weight versus age of chicks on different diets
DNase                   Elisa assay of DNase
EuStockMarkets          Daily Closing Prices of Major European Stock
                        Indices, 1991-1998
Formaldehyde            Determination of Formaldehyde
HairEyeColor            Hair and Eye Color of Statistics Students
Harman23.cor            Harman Example 2.3
```

2.20 Data export through save

Finally, if we finished our analyses, most of the time we also need to export some data. This is an analogy to the options to read data into our. The most basic option would be to directly write a simple text file. With this option you lose a lot of the internal structure of the dataset, and it is not for certain, that it will be imported in the right way.

```
write(kursmatrix,"kursmatrix.txt")
```

Specially, if you have a data frame, it makes more sense to write it directly as a table. With the command `wright.table()` you can specify a lot of options how the dataset will be stored. If you'd like to know, please consult do you documentation.

```
write.table(kursdata,"kursdata.txt")
```

But most of the time, you will probably not need all the flexibility, that ride table gives you. Most of the time, you will like to write a CSV file, because this is the standard exchange file between R and a lot of other software, including spreadsheet software like Microsoft Excel.

```
write.csv2(kursdata,"kursdata.csv")
```

As we have said earlier, please pay attention to the language setting off your computer. Most of the time, at least in Switzerland, you will have a European continental setting, where the decimal separator is a comma. In that case, it is likely that you would like to use the CSV2 format. To try out the differences, you can run the cockpit below, and open the resulting file in your spreadsheet software. You can also inspect it with a text editor.

```
kursdata$height <- kursdata$height/100
write.csv(kursdata,"kursdata.csv")
```

You very likely will have problems with Microsoft Excel. Over spreadsheet software might be smarter. Nevertheless, if you are on the continent, you would like to save your data in the way like below.

```
write.csv2(kursdata,"kursdata.csv")
```

Of course, R also offer us packages, that directly can save files in .XLSX format. The downside of these packages are, that most of the time they require additional dependencies or programming languages, for example perl or python. And actually using the CSV format this is not necessary at all. So it is best that you develop the habit to use CSV as you exchange format between different software.

With this, now I hope, that we have everything at hand, so that we can start using R as actual software. In the next chapter we will start producing that part of statistics, that most people think of when statistics are mentioned: graphs, diagrams, and tables.

Chapter 3

Explorative statistics & graphical display

(The videos for this chapter are here, Number 10-16)

Like I already described in the introduction, statistics can be divided into different subfields. On the one hand there is the statistical inference, dealing with the testing of hypothesis on data. On the other hand there are fields like explorative statistics, in which the detection of pattern in the data is in the foreground. Decides that there also exists the field of descriptive statistics, in which parameters or distributions of data are the main object of investigation.

In this chapter we will deal with explorative statistics and descriptive statistics kind of at the same time, because we will learn how to visualise data with the help of R. Especially graphical visualisation is in between descriptive and explorative statistics.

3.1 Dataset used for this chapter

The dataset, that we are using for this chapter, comes from the burial-ground of Münsingen/Rain. Maybe in the later version of this book I will give more details to the state of said. For the time being every Swiss archaeologist should have a slight idea what is burial-ground consist of. You can download the data using the following link: muensingen_fib.csv.

The data is such represents different fibulae found on the side. Please download the file and save it into a directory of your choice. For this chapter, like with all of the chapters, you might like to specify a specific folder for this chapter. Save the data there, and, as we have learnt in the last chapter, select this folder as your current working directory.

If you have downloaded the data to this folder and correctly selected it as you're working directory, you should be able to reach the Münsingen data and inspect its structure. Since it is a dataset in the "Continental" CSV file, you can load the data into are using to `read.csv2()` format. Also you will realise, that the dataset already contains row numbers. You might like to specify 'row.names = 1'.

```
muensingen <- read.csv2("muensingen_fib.csv", row.names = 1)
head(muensingen) # For getting a glimpse to the data
```

```
##    Grave Mno FL BH BFA FA CD BRA ED FEL  C   BW  BT FEW Coils Length fibula_scheme
## 1    121 348 28 17   1 10 10   2  8   6 20  2.5 2.6 2.2     4     53             B
## 2    130 545 29 15   3  8  6   3  6  10 17 11.7 3.9 6.4     6     47             B
## 3    130 549 22 15   3  8  7   3 13   1 17  5.0 4.6 2.5    10     47             B
## 8    157  85 23 13   3  8  6   2 10   7 15  5.2 2.7 5.4    12     41             B
## 11   181 212 94 15   7 10 12   5 11  31 50  4.3 4.3  NA     6    128             C
## 12   193 611 68 18   7  9  9   7  3  50 18  9.3 6.5  NA     4    110             C
```

The dataset originally comes from the R package `archdata`. It is a data frame consisting of 30 observations with some variables describing the characteristics of the fibulae. If you want to full description of what the state of me, I suggest that you consult the documentation of the ice data package. There this dataset is called `Fibulae`. While we will graphically display different variables, I will explain what is variables mean.

3.2 Cross tables (contingency tables)

The first category of visual representation of data is actually not a diagram. It is the table. Tables are today very widespread for the representation of information, so I don't think that I will need to explain to you what the table looks and our table works. A bit formalised, how we use tables here, is that in most of the cases the rules will hold the items of investigation, while the columns will contain different variables.

But this is not true all of the time. For a very specific type of table, that we will learn to know now, this is not true. The kind of table, that I'm talking about, is the contingency table. This kind of table is also known as cross tabulation, or crosstab. This kind of representation is used to show the interrelation between two variables. That means, contrary to what I have stated in the paragraph above, did hear both the rows and columns represent variables.

Let's have a look to one of these cross tabs, so it becomes clear, what is meant by that. For this, we will tabulate the scheme of the fibula against the grave in which it was found. Fibula Scheme is a standardised way of how different

types of fibulae are are produced, they represent archaeological types. In one grave there may be more than one fibula. Therefore, the great number does not represent a unique identifier of the object fibula, the item of the investigation. It is just one variable among others.

In R, you can use to come on table to display the number of fibula in a specific fibula scheme per grave.

```
my_table <- table(muensingen$fibula_scheme, muensingen$Grave)
my_table
```

```
##
##     6 23 31 44 48 49 61 68 80 91 121 130 157 181 193
##   A 1  1  1  1  0  0  0  0  0  0   0   0   0   0   0
##   B 0  0  0  0  1  1  2  1  1  1   1   2   1   0   0
##   C 0  0  0  0  0  0  0  0  0  0   0   0   0   1   1
```

As you can see, the variable `muensingen$fibula_scheme` is given us the first parameter of the function, while the other variable `muensingen$Grave` represents the second parameter. The result is the table, in which the first parameter is mapped in the rows, while the second parameter, the grave number, is mapped to the columns. Each cell now represents the number of items, in this case of specific fibula types, in each burial. More abstract speaking, crosstab is the representation in which the current occurrence of two variable values are mapped.

If we also want to have an idea, how many items there are per row and how many per column, we might like to add the margins to the table. Table margins gives us to sum of the values for each role, each column, and in total. An R, the command for that is `addmargins()`.

```
addmargins(my_table)
```

```
##
##       6 23 31 44 48 49 61 68 80 91 121 130 157 181 193 Sum
##   A   1  1  1  1  0  0  0  0  0  0   0   0   0   0   0   4
##   B   0  0  0  0  1  1  2  1  1  1   1   2   1   0   0  11
##   C   0  0  0  0  0  0  0  0  0  0   0   0   0   1   1   2
##   Sum 1  1  1  1  1  1  2  1  1  1   1   2   1   1   1  17
```

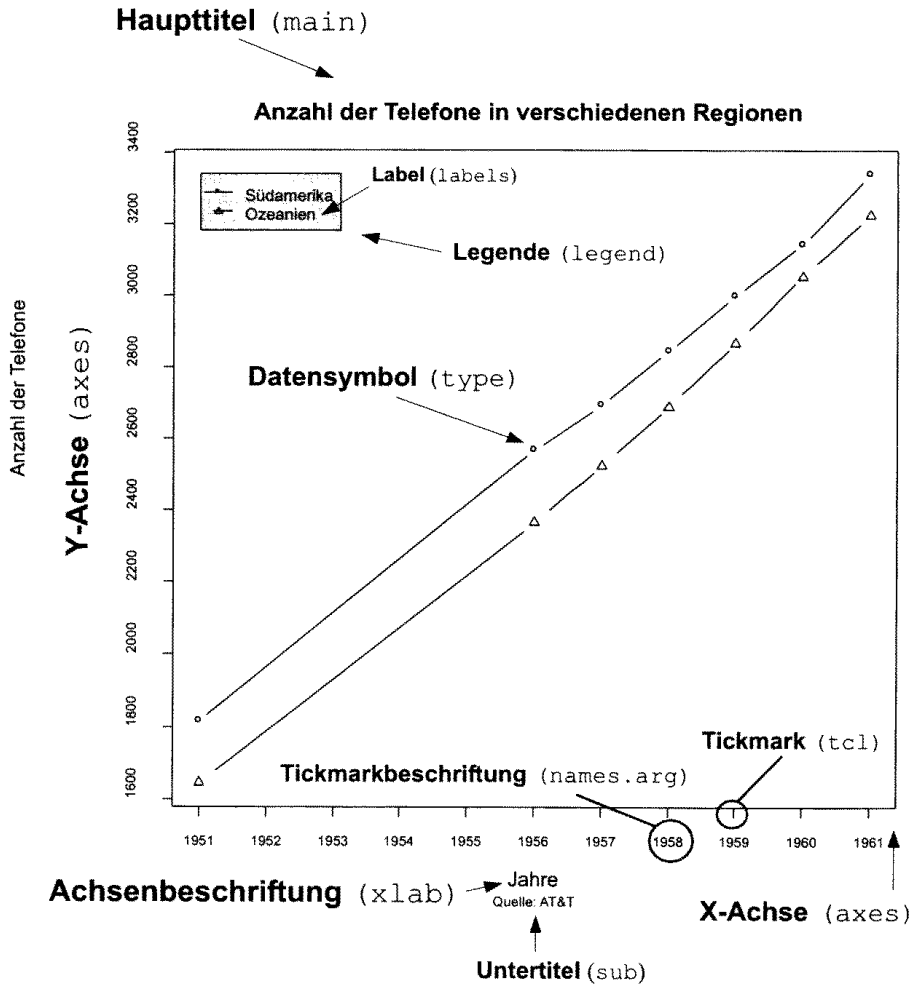
You can see, that we have a new row and a new column. The row contains the sum per column, while the column contains the sum per row. In the lower right most sell we have to total sum of all items.

In data sets of low dimensionality, this represents a rather straightforward and convenient way to investigate the relationship between two, probably more,

variables. They also represent a starting point for different other statistical approaches and techniques, for example the Chi-square test, that we will learn about later. In the context of spreadsheet software, crosstabs are often also called pivot table.

3.3 Basics about charts

Besides the tables, which also contains some graphical elements, like lines, the visualisation, which comes to the minds of most people in respect of statistics, are charts. Most charts or diagrams contain a certain set of elements, that can repeatedly be seen even with different types of graphical display. Most of the time we have some axis, which represents the structure of the variable underlying to representation. Quite often, this access has some marks, most of the time regularly spaced, and often also with some annotation. These marks are called tick marks. For the representation of one variable, one axis might be enough. But most of the time, we have a two-dimensional visualisation. This is already necessary, if we want to represent the category of some items and the count of items in that specific category. Very often, we have charts that represents the relationship of two variables. In both cases, we have two axis. In the area, that is defined by the axis, we have to representation of the actual data. This takes place, using different symbols, lines, or other graphical elements. Here, different types of charts different. Very often, we have also label for the axis, labels for the whole plot, and sometimes some subheadings describing more in detail, what the plot is about.



There are certain rules or guidelines, how shots should be designed, to be most efficient. Edward Tufte, a professor emeritus of statistics and computer science at University, is well known for his publications in respect to data visualisation. In one of his publications, he defined the principles for a good graphical representation of information.

> Graphical excellence is that which gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space. E. Tufte 1983

Just rule of thumb it also known as the data-ink-ratio, the 'proportion of a graphic's ink devoted to the non-redundant display of data-information'. As these are only guidelines, it is clear, that whole details are charged will be very

much depends on its use case. But it's also clear, that one should aim for a reduced use of graphical elements, so that the information, that needs to be transmitted, is in the foreground. Also, certain bells and whistles might enhance this information transmission quite a bit. I trust your sense of style to choose the right amount of ink for the right purpose.

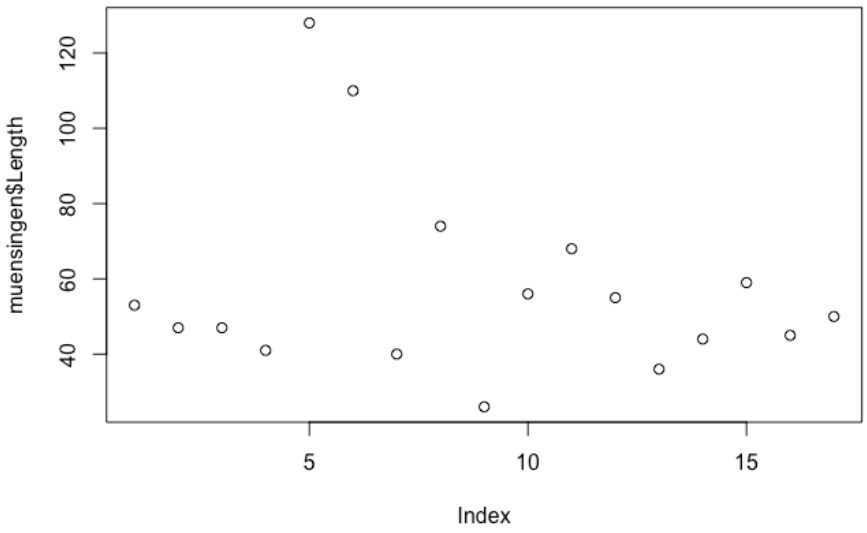
3.4 The command `plot()`

The command `plot()` is the basic commands in R to produce a graphical visualisation. What happens, when you do use plot with different data sets very much depends on, what kind of dataset you have. This command is a kind of chameleon, changing its appearance according to the necessities. This philosophy is also true for different other commands in R. There are some standardisations, that comparable effects should result from comparable names. So, whenever you use the command `plot()`, a plot will be the result. How this plot looks like, depends on the package, from which the data structure is coming, that you are plotting. The same is true for example for the command `summary()`. This command gives a summary, least surprisingly, good structured according to the data underline the summary.

3.4.1 Basic plotting

Lets come back to our command plot(). It's a basic manifestation can be seen, when we plot one variable. For this purpose, let's take the length of the fibulae from our dataset.

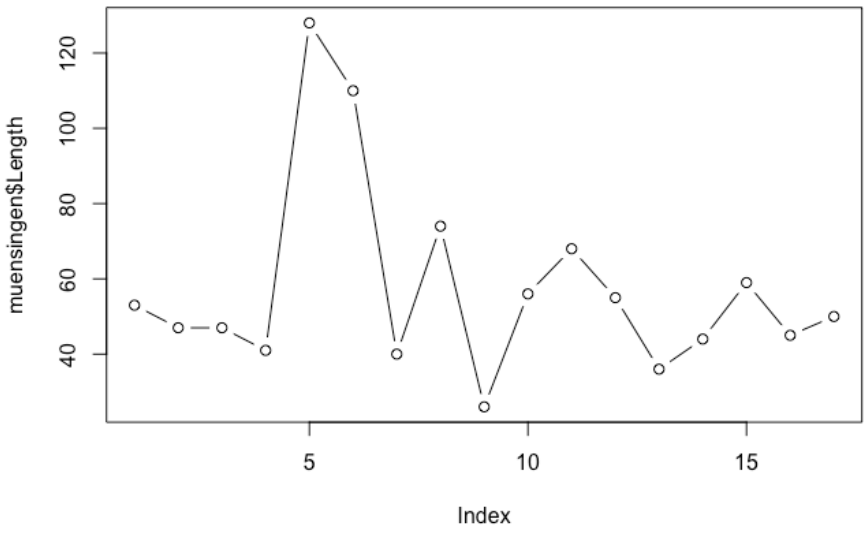
```
plot(muensingen$Length)
```



You can see, that like with most other commands in R, within the brackets the parameters are written. Here it is a reference to the variable, that should be plotted. You also should be able to see elements from the general outline of the plot, that we just have introduced. In this basic implementation, the values to be plotted are visualised using the Y axis, while the X axis represents the order of the values in the vector. The values in the dataset themselves are represented as points, positions according to their actual value on the Y axis, and to their order in the dataset on the X axis. That is why the label of the X axis is index.

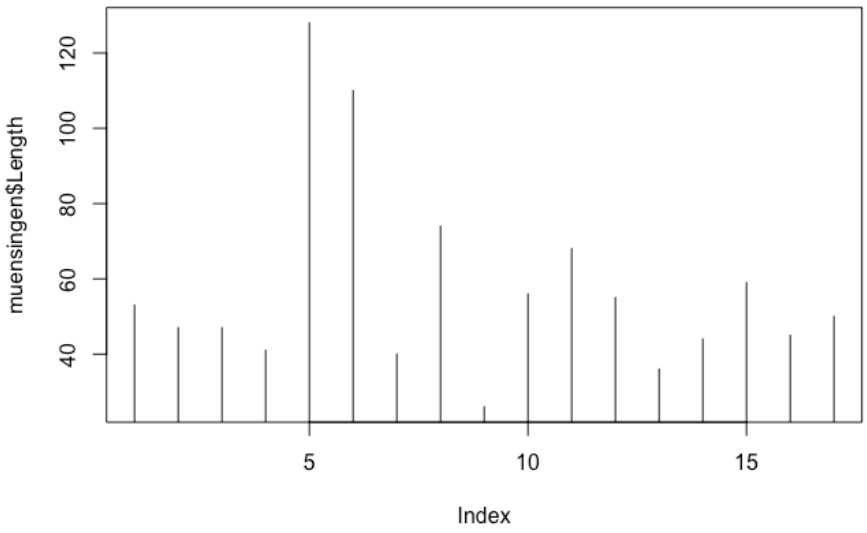
Some standard layouts in respect to specific data can be selected using the parameter 'type'. For example, `type="p"` is the default setting and results in the plots that we just saw. Specifying `type="b"` results in a plot of the points connected by lines, as you can see below.

```
plot(muensingen$Length, type = "b")
```



But this kind of visualisation is not correct here. Line implies that there is a continuous process going on, which is not the case between the individual values of our unconnected similar. Better representation of the nature of our data might be, if we are using the parameter `type="h"`, That gives us vertical lines from the origin of our coordinate system to the actual value.

```
plot(muensingen$Length, type = "h")
```



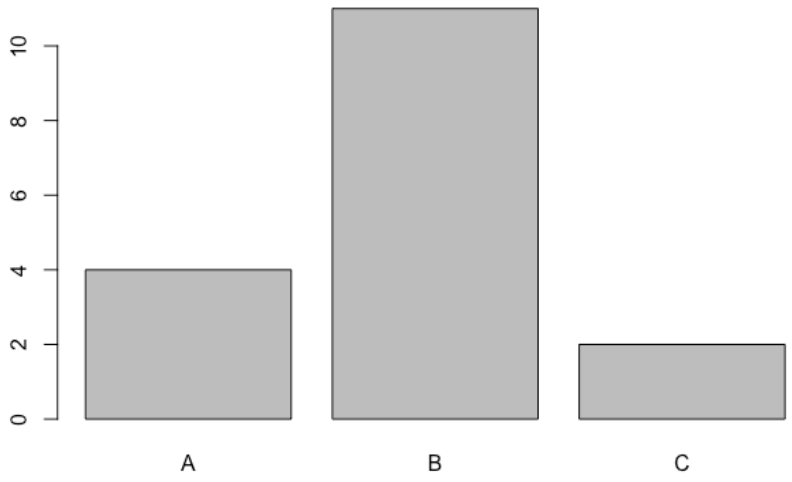
Below you can find a list of possible options:

- p – points (default)
- l – solid line
- b – line with points for the values
- c – line with gaps for the values
- o – solid line with points for the values
- h – vertical lines up to the values
- s – stepped line from value to value
- n – empty coordinate system

The option `"n"`, although seemingly quite useless, will become one of the most interesting options to be selected here. This option can be used, to draw a coordinate system, without filling it in the first place. In this way, we can ask R to draw to coordinate system, that we can then fill with our own symbols or other graphical elements.

As I said, the command `plot()` is a chameleon that changes its appearance according to the dataset. For example, if we used to command on a dataset containing factor variables, the resulting visualisation will be a bar chart, counting the number of items per category, instead of the kind of visualisation that we have seen with the length data.

```
plot(as.factor(muensingen$fibula_scheme))
```



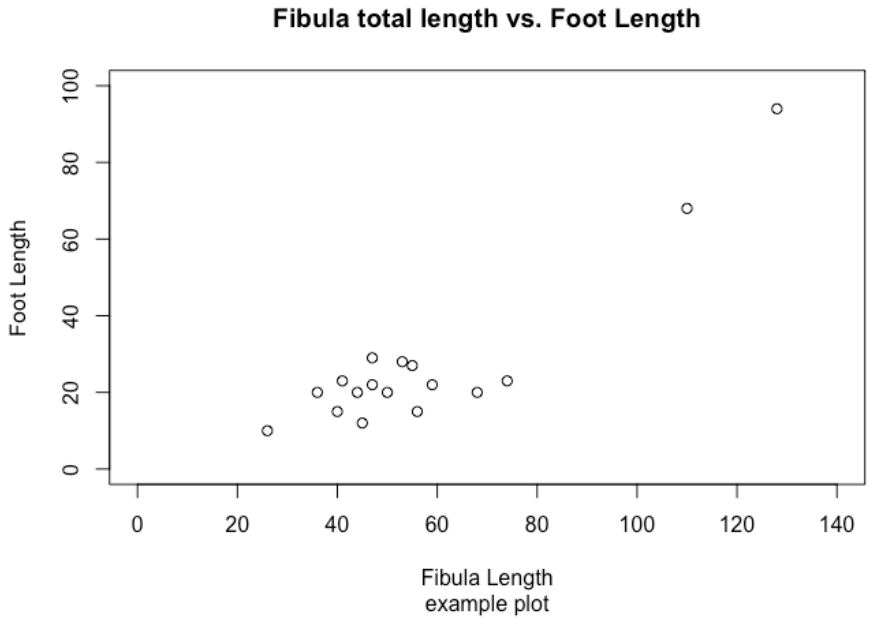
3.4.2 Enhancing the plot with optional components & Text

Of course, we can influence all the different elements of shops in such a flexible software like R. For example, we can specify the size of our X and Y axis, we can change the labels of both axis and also the heading and the subtitle of the chart. This is done using different parameters. By adding up so many parameters, the commands can become quite intimidating. But it is essentially just adding up parameter by parameter. So from a structural point of view there is no complex logic behind that. Also, when writing R commands, You can always us new lines behind elements that indicate, that our command is not finished yet. Such elements might be a mathematical symbols or for example commas like you can see below.

.tiny[

```
plot(muensingen$Length, muensingen$FL,
     xlim=c(0, 140), # limits of the x axis
     ylim = c(0, 100), # limits of the y axis
     xlab = "Fibula Length", # label of the y axis
     ylab = "Foot Length", # label of the x axis
```

```
    main = "Fibula total length vs. Foot Length", # main title
    sub="example plot" # subtitle
    )
```



]

So you can see the effects of the different parameters. The extent of the axis is defined by `xlim` and `ylim`. The labels of the axis is given by `xlab` and `ylab`. The explanatory headings are defined by `main` and `sub`. Also, this plot represented by various plots, in which we met do you already know length of the fibula against the length of the foot of the fibula. It is quite obvious, that the longer the fibula is, the longer also its foot by be. So using this kind of so-called scatterplot, we can visualise this relationship between the two variables.

Plot is doing a lot for you:

- Opens a window for display
- Determines the optimal size of the frame of reference
- Draws the coordinate system
- Draws the values

In the background, also the last plot is remembered, and this plot is still changeable. You can use specific commands, to add elements to the already existing plot. These elements can be:

- lines – additional lines to an existing plot
- points – additional points to an existing plot
- abline – additional special lines to an existing plot
- text – additional text on choosen position to an existing plot

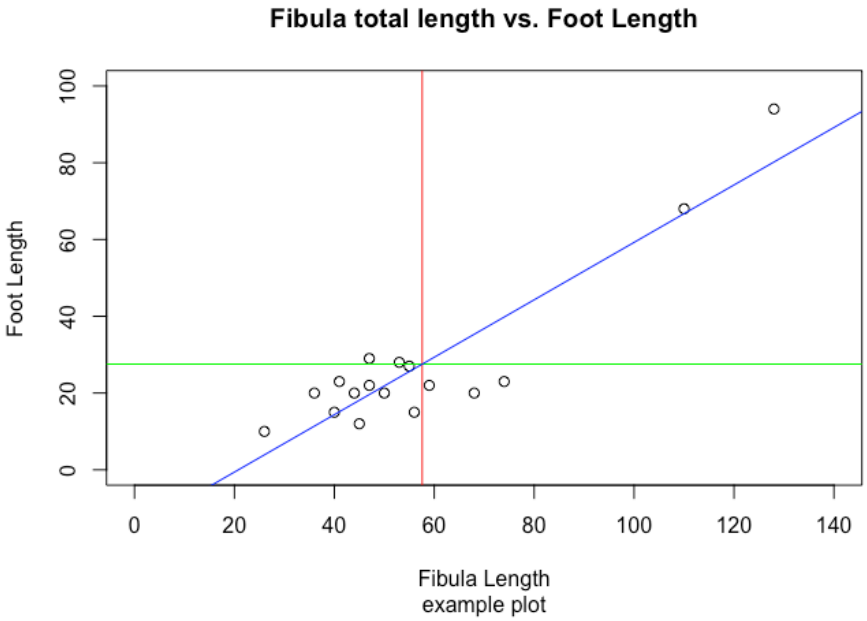
This is the reason, why it sometimes might be reasonable, to plot an empty plot using the option "n". You can use this version, to add first create a coordinate system, and then fill it up with lines or points.

There are some more possibilities to change the layout and style of the plots. For example, the command `par()` will give you the tools to change a lot of the look of and feel. I suggest that you look up the help page for this command to see which possibilities exists to change the layout of the plot.

```
? par
```

As an example, how you can add elements to an existing plot, we will draw some straight lines into the plot of the total length versus the foot length of the fibula. At first, we will draw a line in red at the mean value of the length of the fibula. Since the length is on the X axis, the line with the mean of the length must be a vertical line going up. Accordingly, the mean of the foot length, that is represented by the Y axis, must be a horizontal line. For both, we are using to command `abline()`. The difference is, that for vertical line specify the parameter 'v', for a horizontal line we specified a perimeter 'h'. In both cases the parameter 'col' specifies the colour of the line. The third line is somehow special: it represents the relationship between the foot length and the total length in the data. Therefore, it is a diagonal line. It is defined by linear model of these two parameters. What this means we will explain later, for now you can just keep in mind, that in this way we can represent the trends in the data, drawing a trend line.

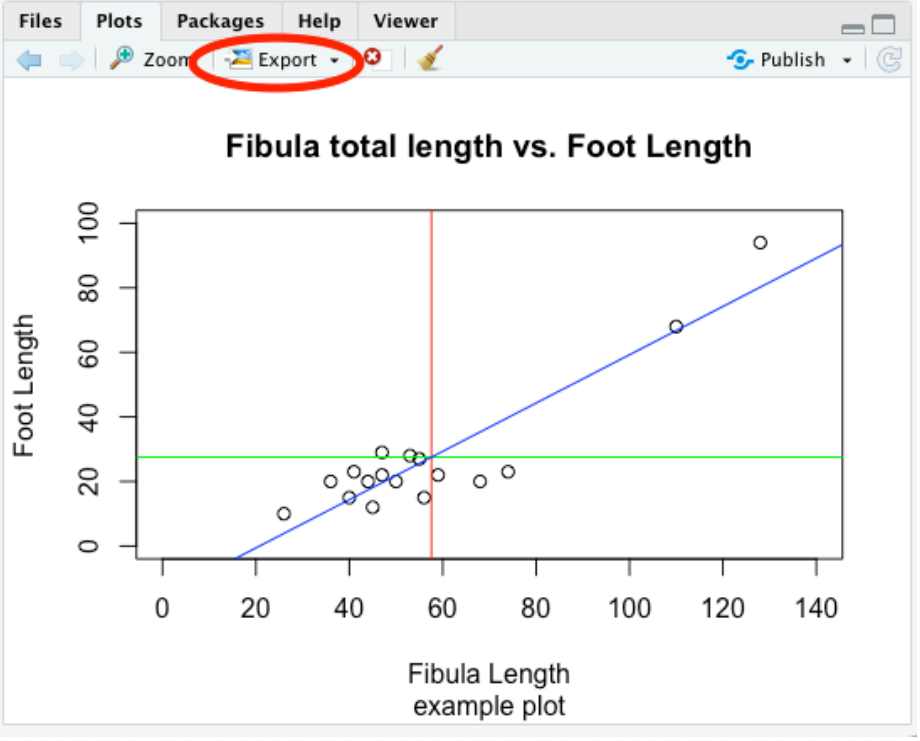
```
abline(v = mean(muensingen$Length), col = "red")          # draw a red vertical line
abline(h = mean(muensingen$FL), col = "green")            # draw a green vertical line
abline(lm(FL~Length, data = muensingen), col = "blue")    # draw a blue diagonal line
```



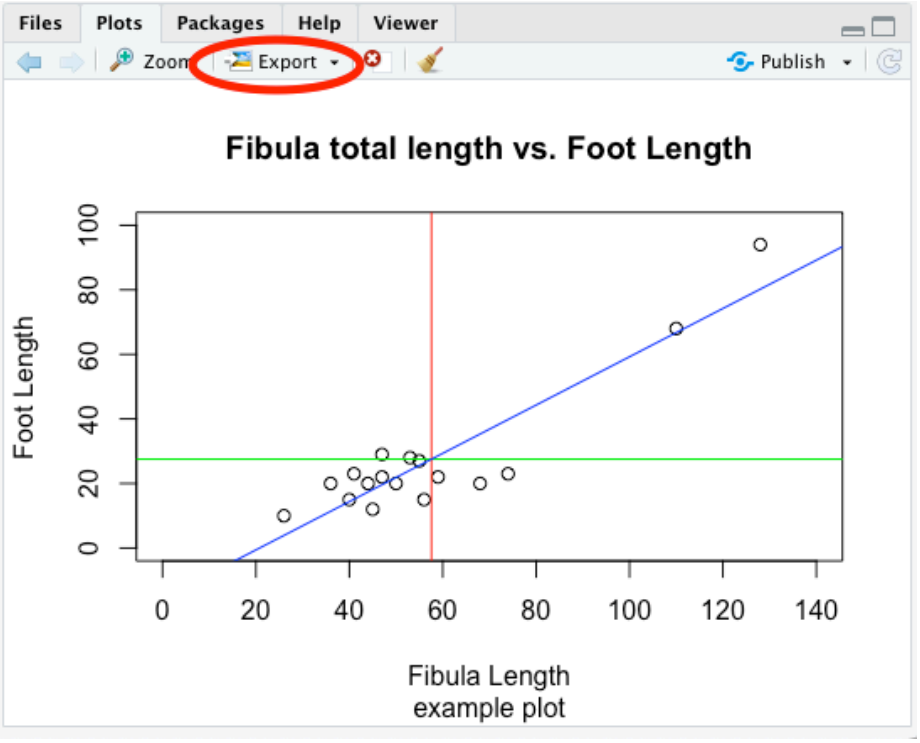
3.5 Export the graphics

Of course, if we have created a decent plot, we probably don't want it to remain in R. Most of the time, we would probably like to use it in other contexts, be at homework, an article or a presentation. To do that, we need to wait to explore our plots.

Using RStudio, probably the easiest way is to be used to graphical user interface, especially when it comes to exporting only individual plots. For this, in the plots window, there is a button called export.



Here, you can select if you want to export your plot as an image (roster image), or as a PDF (vector file). The export windows should be more or less self-explanatory, you can specify the size of the resulting image and also the location, where it should be saved.



Before RStudio, saving plots took place most of the time using commands from the command line. This still is very useful, if you use scripts and want to create multiple or even very many plots at once, changing the inputs data. Two very common formats when it comes to vector format or 'PDF' or probably more common 'eps' specifically for images. You can copy your current plot window to a vector file using the command `dev.copy2...()` and then the file format in which the result should be saved. So let's save our current plots to both formats using to come on spill.

```
dev.copy2eps(file="test.eps")
dev.copy2pdf(file="test.pdf")
```

There are even more raster then vector file formats. By default, R is capable of exporting to PNG, JPEG, TIFF and BMP. You can use the command `savePlot()` for this.

```
savePlot(filename="test.tif", type="tiff")
```

If you really plan to use the graphical visualisation in R in that way, it is worthwhile to dive deeper into the export format and options then we can present here. For most of the basic use cases, exporting the files via the graphical

user interface is the most convenient and most controllable way of storing your valuable plots on your file system.

3.6 Pie chart

Let's not start discussing different plot types. We will begin with one of the most widespread used type of plots: the pie chart. You can see pie charts all over the media, newspapers, and also books, might it be scientific or popular books. Pie charts are used to display proportions of the total. For this reason, they are most suitable (if at all) for nominal data. Or, of course, percentage data, if this is the original data type. For example, results of the election that often represented in pie chart. Here, we see the percentage of voters voting for a specific party. Basically, this represents normal data, the choices of the individuals for one or the other party.

I would like to opportunity to throw in an unnecessary formula here:

$$a_i = \frac{n_i}{N} * 360°$$

The proportions of the categories i in relation to the total number, represented by the number of items of that category n_i divided by the total number N is multiplied by 360°. The resulting angle is the angle, which can be used for a circular visualisation of this amount.

Pie charts have different disadvantages, some they share with other graphical representation of data, but some are unique to this kind of display. Examples to colour selection very influential when it comes to perception. Red as an aggressive colour is perceived larger then for example grey. Very specific to a pie chart is the fact that we as humans are more trained to see differences in length then in area. And in the pie chart, the differences are visualised using the area of the different pieces of the pie. This results in effect, that small differences are not so easily visible in a pie chart, then this would be the case for example with a bar chart, a visualisation technique that we will learn about below.

Like in many other cases, 3-D representation does not work so well on printed paper. Since the human eye and brain must compromise between both the perception that we are looking at a 3-D image but in reality it's a 2-D object, differences are distorted. Let's look at the following example that I took from the literature:

.caption[source: http://www.lrz-muenchen.de/~wlm]

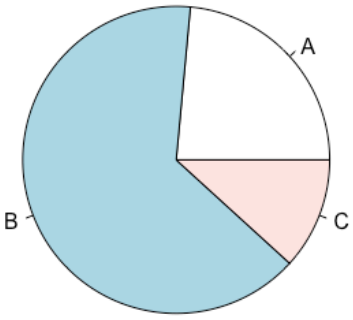
The pieces »viel zu wenig«, »etwas zu wenig« und »gerade richtig« have exactly the same size, the piece »viel zu viel« is a bit smaller. Perception wise, the different shares seem to be quite different because of the reasons mentioned above. So in any case, 3-D or not, pie charts are inferior to a lot of other visualisation techniques. Nevertheless, because it is a very widespread used technique, still I would like to demonstrate how are you can create them using R.

The actual commands to draw a pie chart in R is `pie()`. This command expects the number of counts, and all the normalisation to percentages and then the visualisation will take place automatically. This means, that it might be necessary to recode data. In this case, we will use the fibula schemes and visualise there ratio. This variable comes in the form of a character vector indicating to different schemes. Here, we can use the command table, to transform the nominal presence into a table of counts.

```
table(muensingen$fibula_scheme)
```

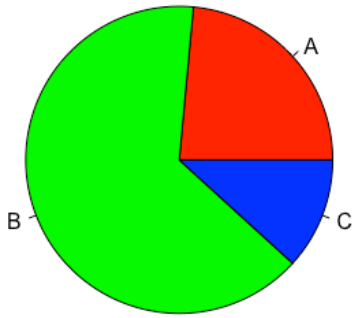
```
##
##  A  B  C
##  4 11  2
```

```
pie(table(muensingen$fibula_scheme))
```



Do you original colour scheme of the `pie()` command is rather pastel. Of course, you can change the colours, using the col parameter. Here, in the order of their appearance, you can specify different colours, that will be used to represent this category.

```
pie(table(muensingen$fibula_scheme),
    col=c("red","green","blue"))
```



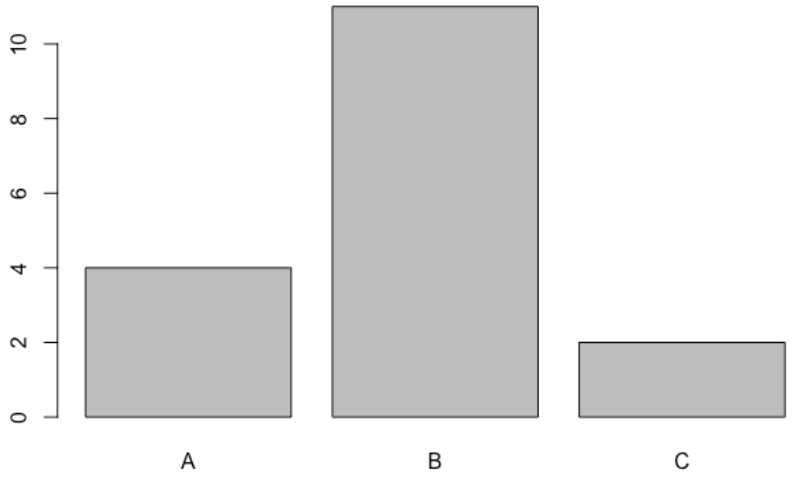
This should be enough to be set for the pie chart. It had already had the honour to be the first mentioned. That's enough. We will from now on turn to more scientifically useful visualisations.

3.7 Bar plot

A worthy replacement for any kind of pie chart is a bar chart or a plot. Most of the time it is the better alternative. Since here, the differences are represented by the length of the bars, humans can more easily perceive the differences between different categories. Also, this kind of visualisation is more flexible, because you can also represent absolute data and measurements with a bar chart, not only percentages.

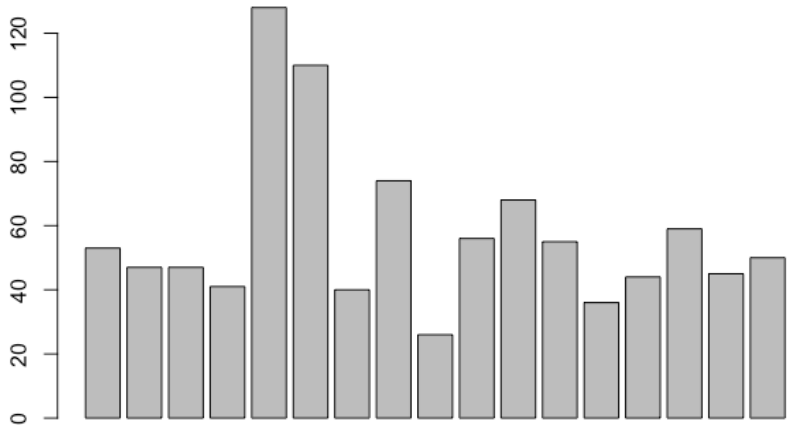
Also, the command `barplot()` requires a vector containing the data. This can be the number of fibulae in the different styles, like so:

```
barplot(table(muensingen$fibula_scheme))
```



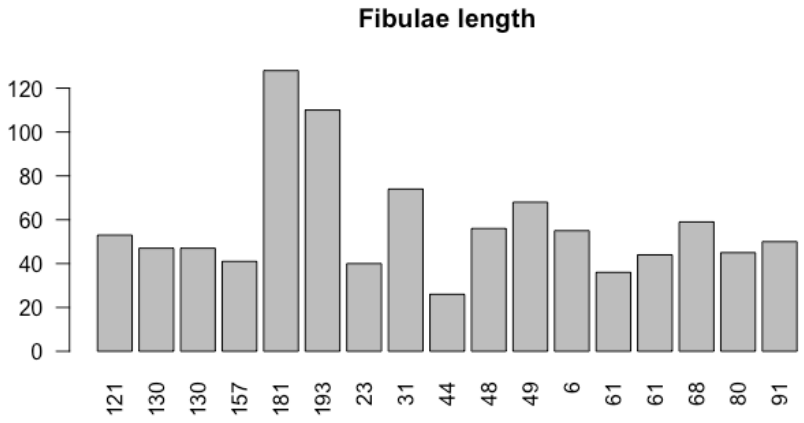
Or it can be length of the different fibulae.

```
barplot(muensingen$Length)
```



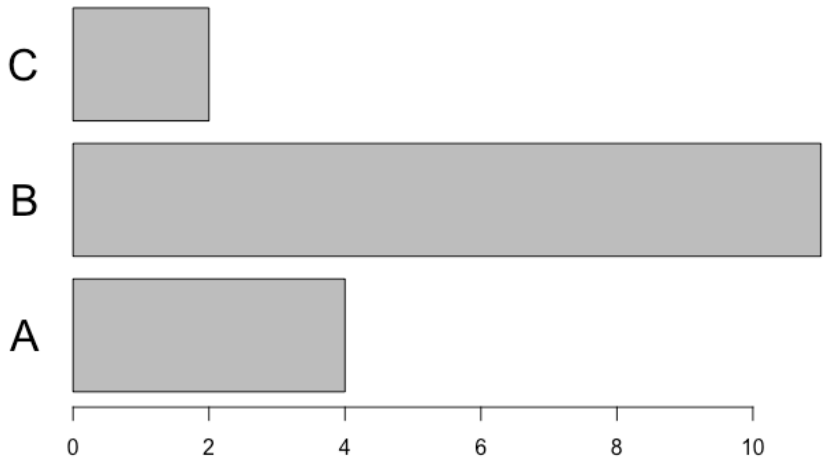
Both versions are meaningful and can help visualising the data. Although especially the last plot is difficult to understand, because it it lacking some essential information. For example, what is represented here by the bars, in total, but also, what does the individual bars represent. Since the vector resulting from the table commands automatically had names, these names were used in the case of the fibula scheme. In the case of the length, we have to specify these names on our own. For this, there is the parameter "names.arg". Also, you might like to turn the names so that they become more readable and do not flow into each other. For this, you can use the parameter "las=2". Lastly, I'll give you a variant of how you can put the main title to a plot using to command `title()`.

```
par(las=2)                          # turn labels 90°
barplot(muensingen$Length,          # plot fibulae length
        names.arg=muensingen$Grave) # with names of the graves
title("Fibulae length")             # add title
```



Of course, you can also turn the bar chart around, making it horizontal. In that case, you probably would like to turn the labels again. Also, you can influence the size of text using a parameter 'cex'. You can also specify what should be changed in size, in this case the names.

```
par(las=1)                          # turn labels back again
barplot(table(muensingen$fibula_scheme), # Plot counts fibulae scheme
        horiz=T,                    # horizontal
        cex.names=2)                # make the labels bigger
```



Bar charts are also much more flexible compared to pie charts in so far, as you can easily display more than two variables. In this way, your plot can become 3-D or more in a meaningful way.

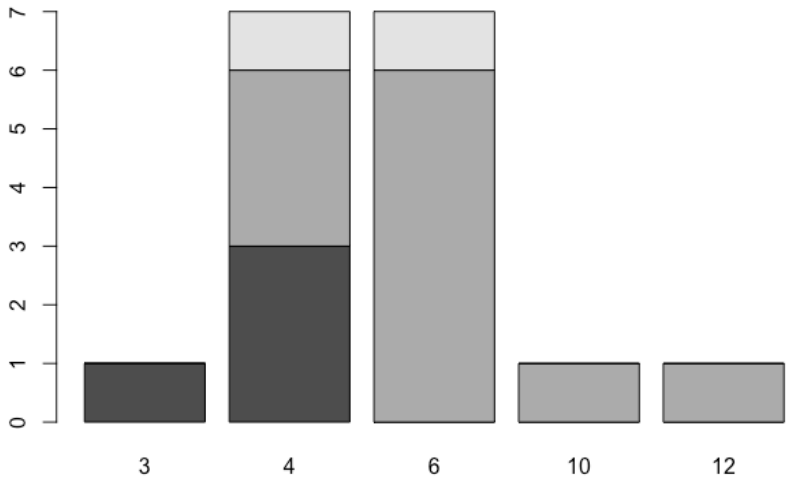
Let's assume, we want to visualise the number of coils of the fibula in relationship to the style. We can use the table command to produce a table accordingly.

```
my_new_table <- table(muensingen$fibula_scheme,
                      muensingen$Coils)
my_new_table
```

```
##
##     3 4 6 10 12
##   A 1 3 0  0  0
##   B 0 3 6  1  1
##   C 0 1 1  0  0
```

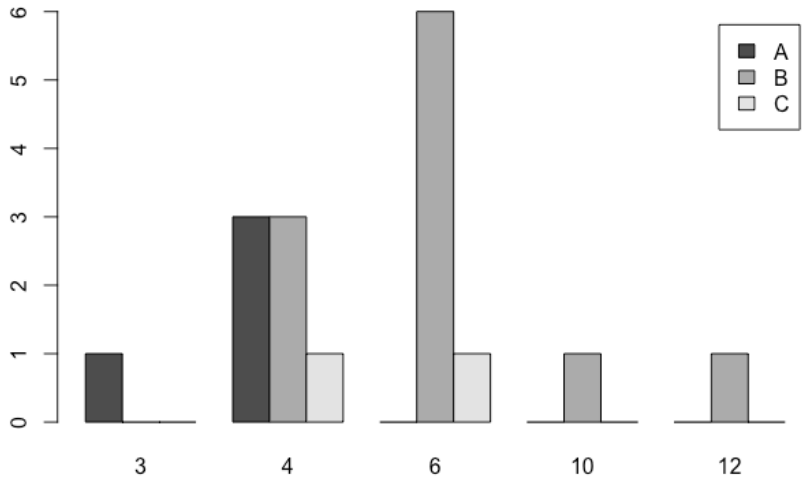
No, we can directly put this table into the `barplot()` command. Let's see the output:

```
barplot(my_new_table)
```



You can see, that the different styles (fibula schemes) get different colours. With this, we not only seeing the number of items in respect to the different number of coils, but also at the same time, in which style to fibula is produced. This way of representing subcategory it's called 'stacked'. If you don't like this way of representation probably you like more the version where the different categories are put side-by-side like below.

```
barplot(my_new_table, beside=T, legend.text=T)
```



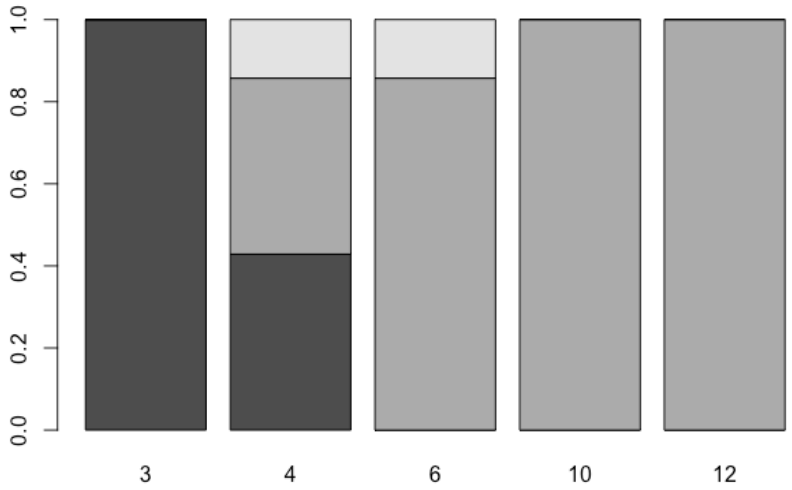
Until now, we have only seen bars of different height. The beauty of the pie command was, that it automatically transformed absolute values into percentages. With a slight alteration of the table commands, we can achieve the same with a bar chart, and even better. For that, we are using the `prop.table()` command. This stands for proportional table. In current versions of R, you can use also the command `proportions()`. Its first parameter is the dataset for which the proportion should be calculated, in the format of the table. That means, it takes the result of the table command, and then transform it into a proportional table. The second parameter defines, what should sum up to 100, or in other words, what is the full total to which the proportion should be calculated. As always in R rows come first, so they have the number 1, while columns come second, so they have the number 2. That means, the following command will calculate the percentages in respect to the columns, which each will sum up to 1.

```
table.prop<-prop.table(my_new_table,2)
table.prop
```

```
##
##           3       4       6      10      12
##   A 1.00000 0.42857 0.00000 0.00000 0.00000
##   B 0.00000 0.42857 0.85714 1.00000 1.00000
##   C 0.00000 0.14286 0.14286 0.00000 0.00000
```

If we put the results into the `barplot()` command, we will get a result comparable or even better to the pie chart.

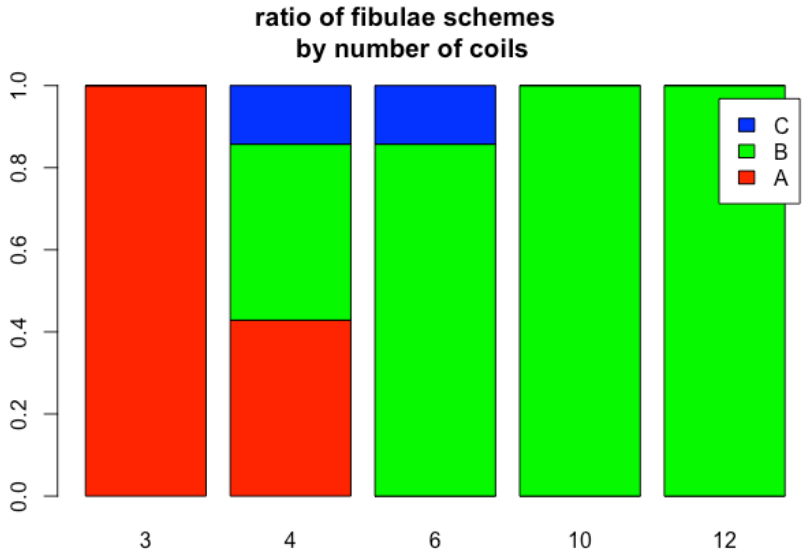
```
barplot(table.prop)
```



Of course, you can also change elements of the bar plot. For example, you can get fancy with the colours. Here, are used to command rainbow, specified with the number of colours I would like to get, to create a colour spectrum from the rainbow. Also, I would like to have a legend. Additionally, I want to have a title. This title is quite long, so I divided it in two rows, using the special "\n" sign. Since it is such a long title, I also have to use space outside of the actual plot area. You can get a feeling for the effects of this different para metres by playing around a bit with them.

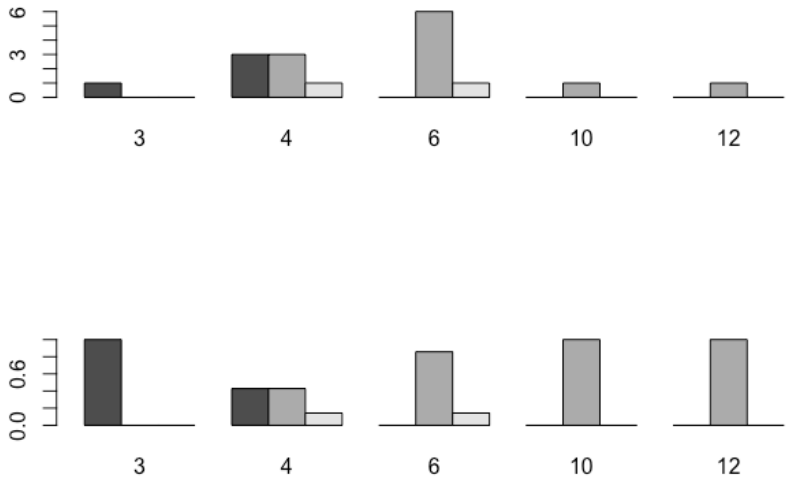
```
barplot(table.prop,
             legend.text=T,  # add a legend
             col=rainbow(3)  # make it more colorful
             )

# add a title
title("ratio of fibulae schemes \n by number of coils",
      outer=TRUE,            # outside the plot area
      line=- 3)              # on line -3 above
```



But also bar plots do not solve every problem that we have with graphical representation. For example, there is often the question, what is better: percentage or absolute numbers. If we would like to compare different situations, with different total numbers, and we are most interested in the ratios, then the percentages are a good choice. But at the same time, due to this characteristic, they can hide differences in the underlying total numbers. This can become especially problematic, if you divide your bars also in subcategories.

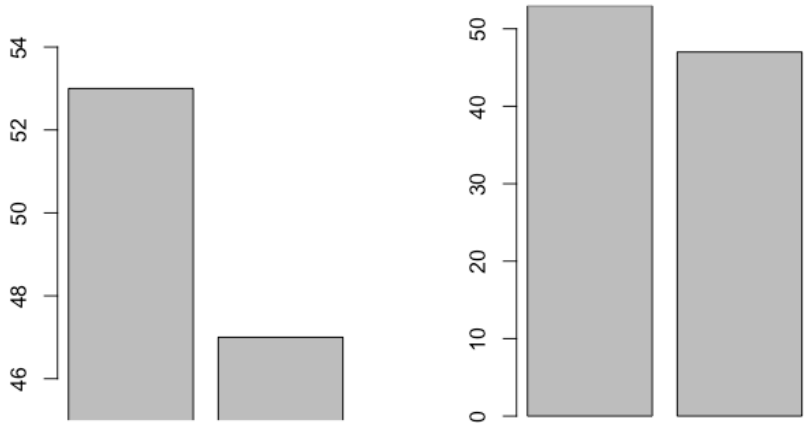
```
par(mfrow=c(2,1))
barplot(my_new_table,beside=T)
barplot(table.prop,beside=T)
```



Just from the visualisation of the percentages here it seems, as if there are more fibulae of scheme A with three coils then with four. So it is absolutely necessary to always provide the absolute numbers, if you present your data as percentage of the total. It can take place in the caption, or directly in the plot. And this problem or better this consideration must be taken into account not only with bar charts, but also with any other representation of percentages.

Another source of visual confusion can be the scales already ranges of the axis. For example, if we do not draw an access from 0 to the maximum value, but let it start at an arbitrary value, small differences can visually become very big. In the example below, I visualise the first and the second fibula respectively their length.

```
par(mfrow=c(1,2))
barplot(muensingen$Length[1:2],xpd=F,ylim=c(45,55))
barplot(muensingen$Length[1:2],xpd=F)
```




```
par(mfrow=c(1,1))
```

Although it is obvious, if you actually look at the axis, there is a difference it's not that big (it's only six), only from visual inspection it seem to be enormous compared to the visual representation of the same difference in the diagram to the right. Most of the time, it is better to have your axis ranging from 0 to the actual values, except for situations, where the comparison between different bars or other elements is hindered by the fact that the relative differences are so little.

You might have realised, that in the last two examples are used to command `par(mfrow=c(..))`. With this comment, you can find that plots are placed side-by-side or one on top of the other. Also here do usual rule of R present: Rows are the first number, columns are the second number. So both I said that I want to have two rows of plots with one column, below I said I want to have one row with two columns. You can use this to lay out your plots in a smarter way.

3.8 Box-plot (Box-and-Whiskers-Plot)

The next type of plot is not totally dissimilar to the bar plot. Also in this case, in the case of the box plot, we have a rectangular element representing our data. But the logic behind the box plot is very different.

A box plot, which is also called box and whisker plot, is used to describe the distribution of values in a range of data. Let's try to explain that using the numbers from 1 to 9 as our values. If we sort these values, than we get the root of numbers from 1 to 9.

No, we divide the values by their position. That value, that is placed in the centre, is of specific importance because it is the most centred value of this dataset. Here we draw a line. If we take half the number between the first and the centremost, we get the first quarter of our data. When we do the same towards the end, we get the last quarter of the data. Where the first quarter ends, and we are the last quarter starts, we also mark this dataset. The values at these positions are 3 for the start of the first quarter, 5 for the most central value, and 7 for the start of the second quarter.

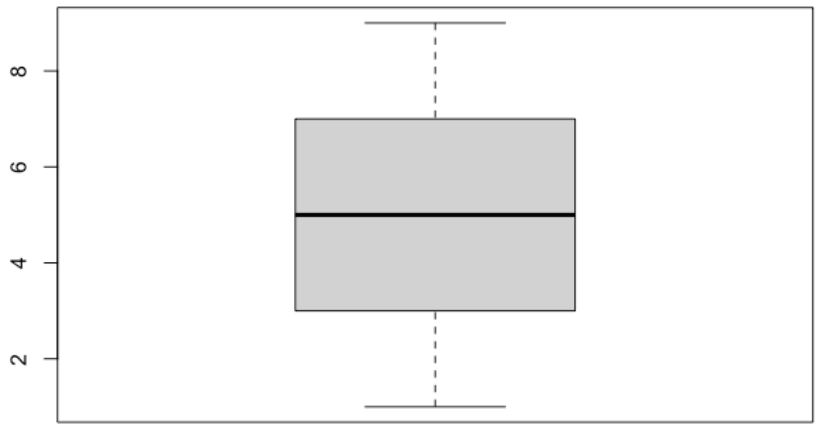
Compare this with the following distribution of data.

Here, two marks the beginning of the second quarter, seven customers sent a value, and 20 marks the beginning of the last quarter of the data. Notice, that this does not depend on their actual values, but only on the position within the ordered dataset. If we now have a Y axis, on which we have continuous scale of the actual values, and we draw a box according to the parameters we just defined (position of the first the second and the third quarter of the data), than we get a feeling for how the values of the data within our dataset are distributed.

In the visualisation of the box plot, the box marks to inner half of the data so the second and the third quarter of the data. The border between the second of the fourth quarter, the most central value, is marked by a line (This value is also "median", we will learn about it soon). Beside the box itself at the thick line dividing it, there are also thin lines sticking out from the box on top and bottom. This lines are called whiskers. The end line of a whisker is drawn at that value, that is less than 1.5 times the distance of the inner half of the data away. Every other point, that is more far away, is considered to be an outlier, and is visualised by a point.

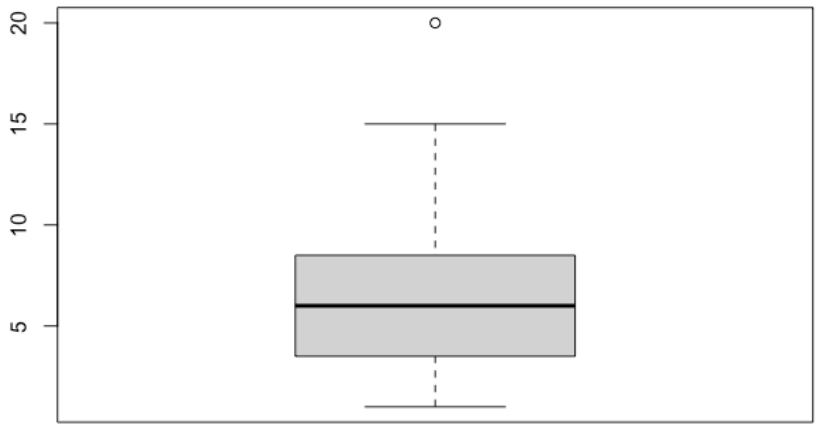
Let's see the actual box plot of all numbers 1 to 9.

```
boxplot(1:9)
```



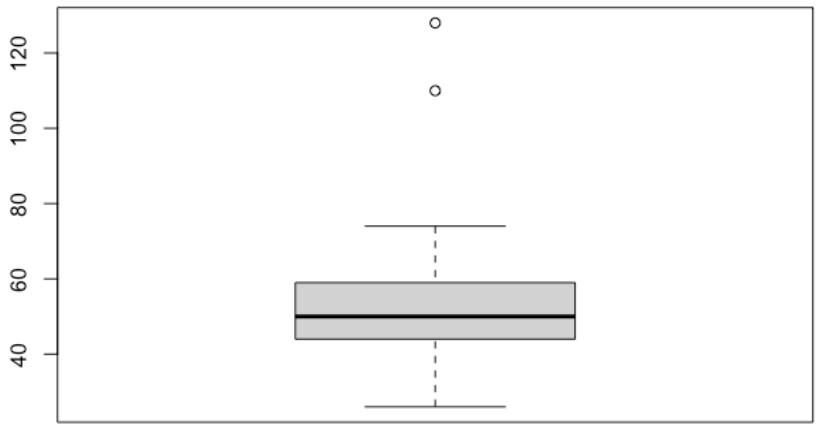
Here you can see all elements I've just described except for the outliers. To produce an outlier, we will add a very high value to our dataset.

```
boxplot(c(1:9,15, 20))
```



You can see, did I edit the value of 15, which is now marked by the whisker. I also edit the value 20, which is now displayed as an outlier. If we applied to our actual data, the length of the fibula at Münsingen, you will see kind of the same.

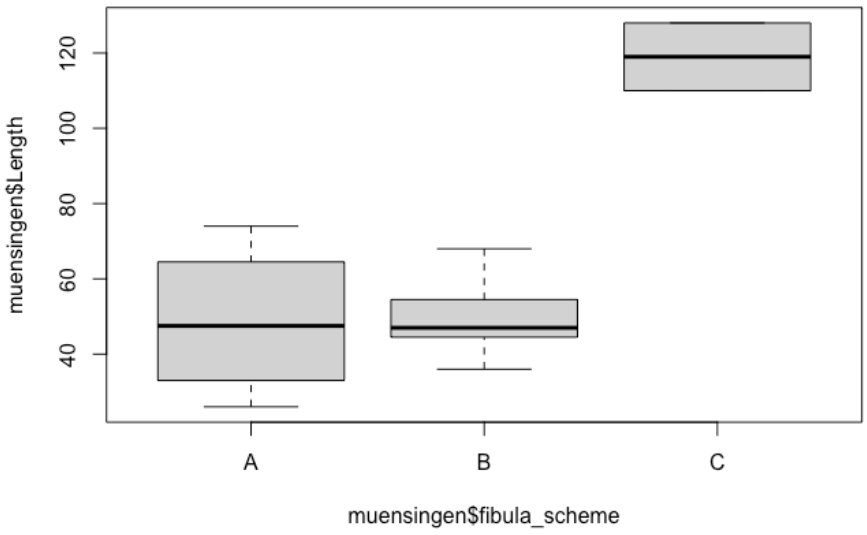
```
boxplot(muensingen$Length)
```



The interpretation would be like following, most of the data are evenly distributed between 30 and 80 mm. The thick line is a bit lower in the box, this means, that there are slightly more low values than high values. Besides the usual values, we have two outliers at approximately 120 mm plus or minus. By inspecting this plot, I already learnt a lot about distribution of the data in our dataset.

Box plots are especially useful, if you want to compare the distribution of data between different categories of data. For example, you might like to compare the distribution of the length of the fibula in respect to their style. For this, I will introduce to you in new syntax in how are you can formulate circumstances in R. This notation is called formula notation. It is centred around the tilde '~'. This sign means "in relation to". So, if I like to draw a box plot of the length of the fibula in respect to its style, I can express it like a below:

```
boxplot(muensingen$Length ~
          muensingen$fibula_scheme)
```



We will work with this kind of formula notation more in more advanced topics here. This notation becomes especially helpful when it comes to modelling. Because there, you model things in relation to other things.

Back to our box plot. Of course, we can also use parameters here, to add elements to our plot. This elements might be a title, the colour, or the labels of the axis. You might like to play around a bit with the example below to get a feeling of the effects.

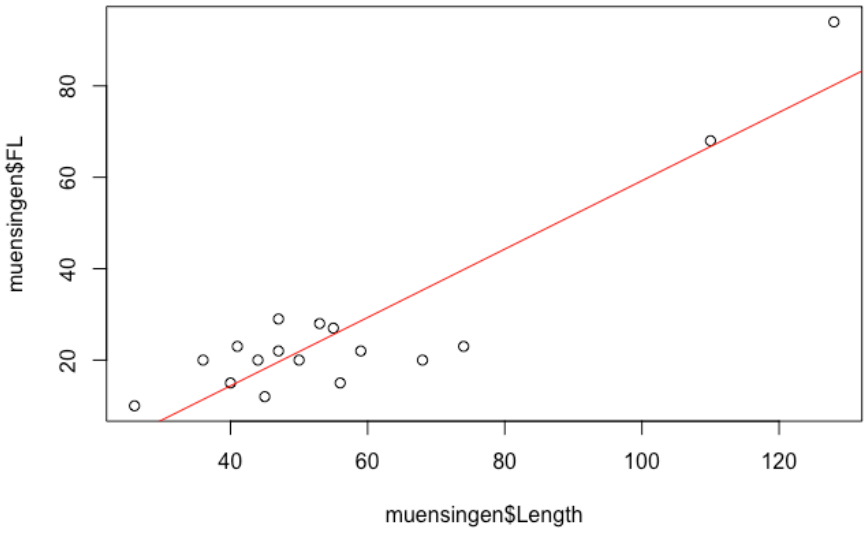
```
par(las=1)
boxplot(Length ~ fibula_scheme,
        data = muensingen,
        main = "Length by type",
        col="green",
        xlab="fibulae scheme",
        ylab= "length"
        )
```

All in all, a box plot is a very helpful tool when it comes to condensed have a look on the distribution of data. As I have explained, this is specifically helpful if you want to compare different distributions with each other. If you are more the girl that likes to watch the matrix uncoded, the scatterplot is probably more your type. We will learn about that below.

3.9 Scatterplot

Just get a proper is probably that kind of plot of people (beside a pie chart) imagine most of the time when they think about statistical visualisation. It is also one of the most basic plot types. That's why it is also the standard configuration of the command `plot()`. Basically it is used to display one variable in relation to another one. This other variable can also be the order of the values, as we have seen in the example above. In general, scatterplot is suited for all data types and scales of variables, although most of the time for nominal and ordinal data other chart types might be preferable.

Since we discussed basic elements of the scatterplot already above, without mentioning the name, I would like to use the space here to show you some alternative ways how are you can produce a scatterplot in R. For this, we start with the basic plot of the similar length against the foot length, with the Trent line edit in red, like we have done before.

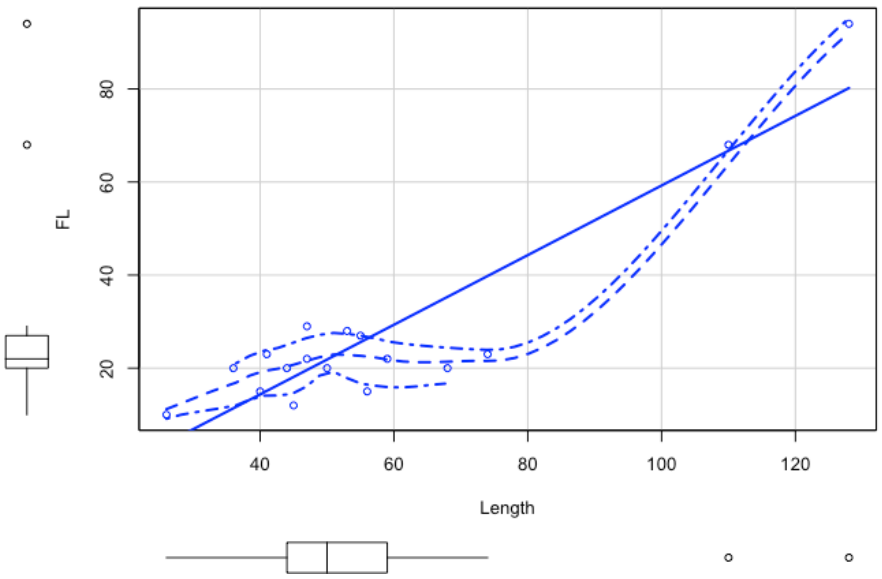


Although libraries offer a different ways of displaying scatterplots. One option here is the library car, that is specifically used for regression analyses (analyses of the relationship of two variables). To get access to the functionality of another library in R, at first we have to load this library. For this we used to come on `library()`. As the parameter you use the name of the library that need to be loaded. Pay attention: you don't need to use quotation mark " to load known and installed libraries.

The command from 'car' to produce a scatterplot is `scatterplot()`. In here

you also use the formula notation that we have seen already with the box plot, but also with the linear model for the trendline in the example above. Here you specify the names of the columns in the dataset as variables, and as data you give the name of the variable which holds the whole dataset, the data frame `muensingen`.

```
library(car) # library for regression analysis
scatterplot(FL ~ Length, data = muensingen)
```

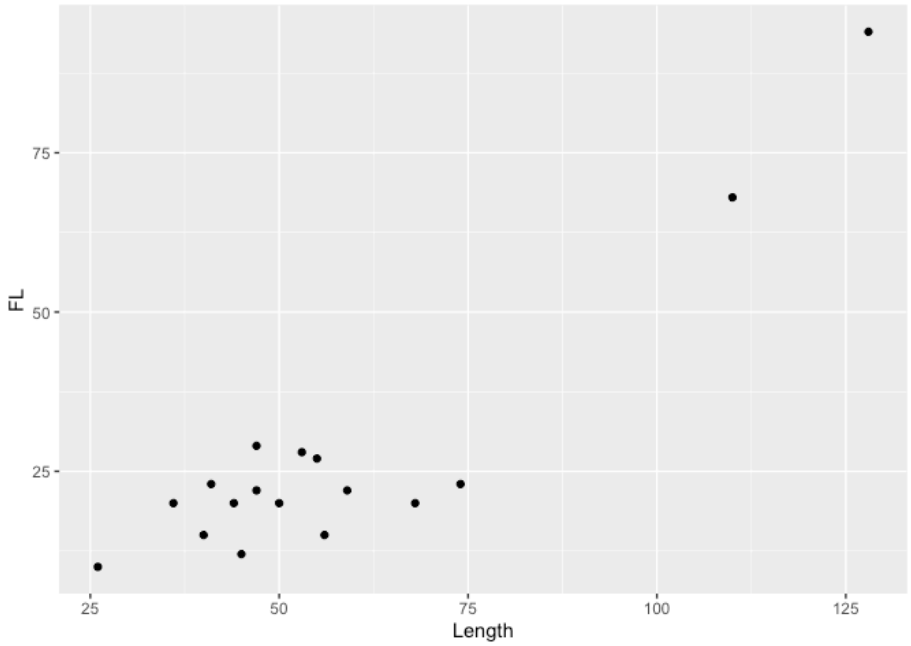


At this resulting scatterplot, you can see different elements, that we have seen before: for example, the box plots. Also, we see a trend line. But also, we see more lines. If you want to know, what is lines mean, you should consult do you help for the function `scatterplot()`. Luckily, you know how this can be done.

Another suggestion is to use the library ggplot2. This library is very powerful and it is used a lot in professional visualisation. In publications and in conference presentations you will see ggplot style visualisations very often. The reason, why we are not using it here, is, that it comes with its own syntax philosophy. To learn this, would overwhelm potentially students that already have to cope with the basic understanding of R. But once you have mastered R in general, I strongly suggest that you have a closer look to this plot library.

```
library(ggplot2) # advanced plots library
b<- ggplot(muensingen,aes(x=Length,y=FL))
```

```
graph<-b + geom_point()
show(graph)
```

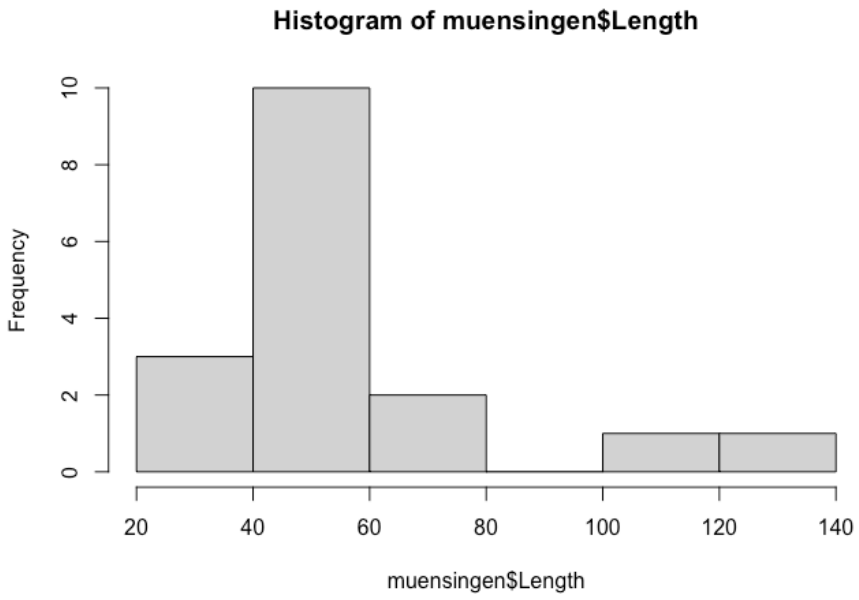


3.10 Histogramm

The next type of plants that we would like to have a look to is the histogram. Here, we take a different perspective compare to the scatterplot, and more similar to the box plot. Also here, we are looking at the distribution of the data. We visualise, in which part of the value range of our data most of our data are located. So what we will see, is if most of the data rather low or rather high values, for example.

At first I would like to give you an example of an histogram, so that we can understand its character and its use cases. Therefore let's again use the length of the fibula in our dataset and plot the histogram accordingly.

```
hist(muensingen$Length)
```

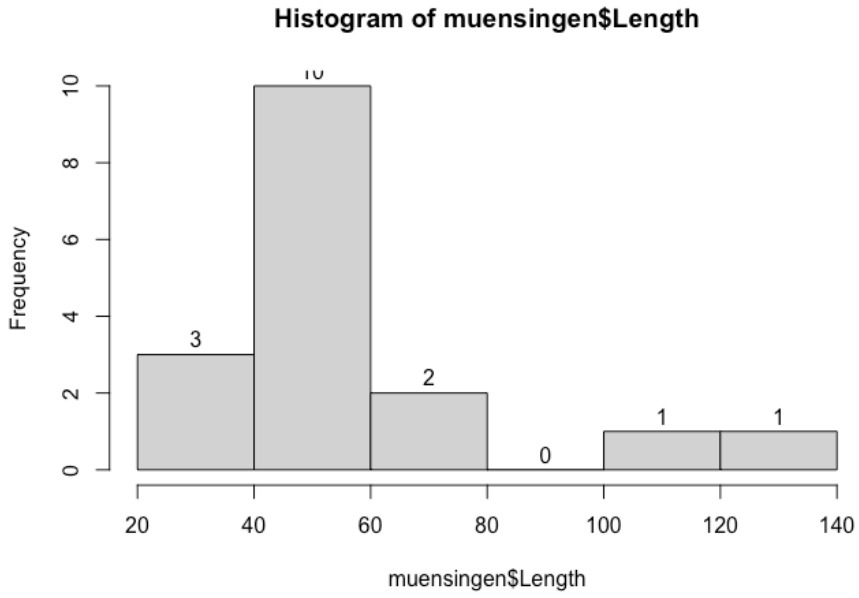


You can see that on the X axis of the plot we can see the actual values. The Y axis has to label frequency. The data are represented like with a bar plot in between values of length. In the first bar, between 20 and 40, we have to representation of all fibulae with a length between 20 and 40. If we look to the frequency, we can see that there are three. The next class of fibulae is between 40 and 60. Here, we can see that we have 10. This goes on.

So, in histogram, we don't see any longer the individual values, but we see, how many items have values within a certain range. With this visualisation and the perspective on the distribution we are reducing the complexity of the data. You don't any longer see the individual item, but we get a better understanding about the distribution of the values of the individual items within the whole dataset. Quite often in visualisation, but also statistics in general, we have to compromise between the consideration of the new visual case and the extraction of the general pattern.

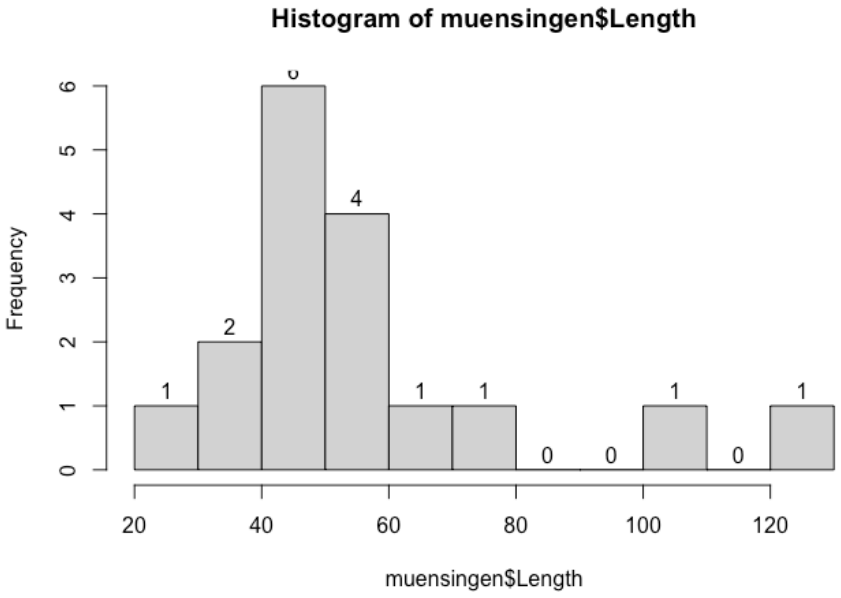
In the standard display of histogram, we can only guess the individual values or the total numbers of cases per class. If we add labels, we can also see the actual numbers represented in the plot.

```
hist(muensingen$Length, labels = T)
```



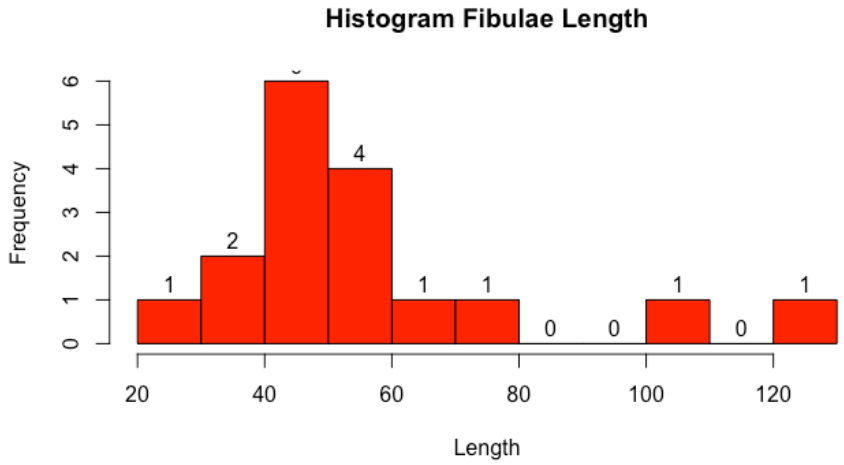
Of course, we are not forced to only use the classes between 20 and 40, 40 and 60 and so on. We can also define our own classes. For example, if we want to have a finer resolution, we could decide to display classes off with of 10. We do that, using the parameter 'breaks'.

```
hist(muensingen$Length,
     labels = T,
     breaks = 10)
```



Please note the differences: before, we had some blocks, that now are divided into finer structures. The choice of the class with can be decisive for the interpretation. Although our dataset is rather small, if you look to the highest values, just from visual inspection in the first case it seems that we have a rather constant data distribution between 100 and 140. With the smaller class with, the holes in this distribution become obvious. So also here, we have to make a compromise between visualising the individual case and total pattern.

Again, of course, we can use the usual suspects to change the look of our histogram.



The disadvantages of the Instagram are, that makes the data reduction necessary and therefore we lose some information in the visualisation. Also, as we have seen, is the actual display show me dependent on the choice of the class with. There are different techniques to overcome especially this problem. The first, the stem and leaf chart, comes from an age, where computers we are not able to produce plots. For reasons of completeness, but also because it's mentioned in the Stephen Shennan's book, we included here. The other and currently much more popular version is the kernel density estimation or kernel smoothing. We will learn about it afterwards.

3.11 stem-and-leaf chart

The stem and leave chart is a clever idea to at the same time represent the general pattern, but also the individual cases. On the one hand, it is also a kind of histogram. But at the same time, it shows the values of the individual cases. Nevertheless, it has become a bit out of fashion lately.

Let's demonstrate it with our usual example. The command is stem.

```
stem(muensingen$Length)
```

```
##
##   The decimal point is 2 digit(s) to the right of the |
##
##   0 | 34444
##   0 | 5555566677
##   1 | 13
```

Our dataset consists of several similar length below 100, and two above 100. The latter you can see as the last line of the result of the stem and leave plot. The line above represents all the fibulae between 50 and 100. The top line represents all those between zero and 50. You can see, that the individual cases are represented by figures. This figure indicates the next value. So for example, in the top row the first number is zero. This means, that the first number is below 100. It starts with a 3, so it's round about 30. After that, we have four 4s. This means, we have four more fibula with length of around 40. By now, you should've understand the pattern.

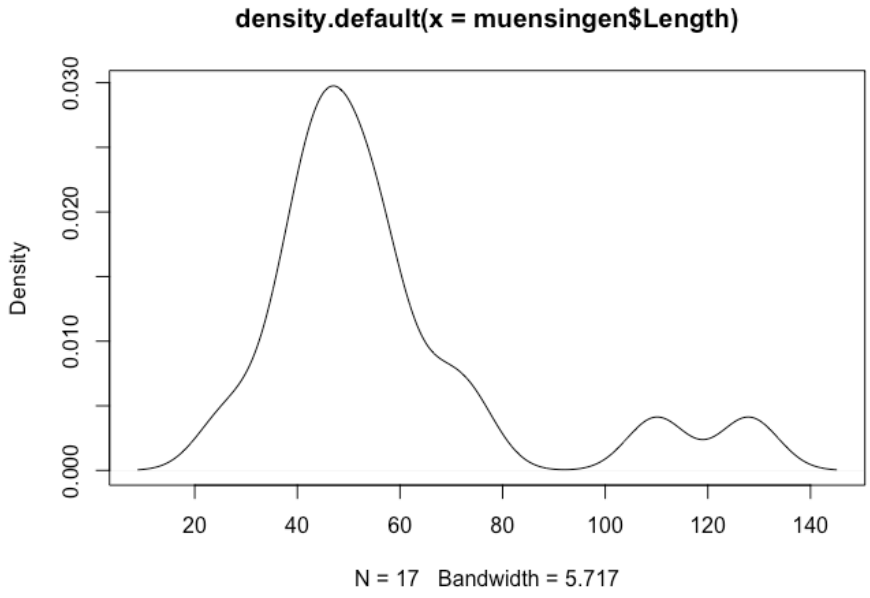
So as promised, the steam and leaf plot is as such a clever idea to represent data. It's only drawback is, that it doesn't look very visual, and reminds a lot on the age of computers without decent graphical displays. And has also announced, there is some more modern alternative to that that we will have a look to in the next part: the kernel smoothing.

3.12 kernel smoothing (kernel density estimation)

The last plot version that I would like to introduce to you is the kernel density estimation, quite often abbreviated as KDE. Also this visualisation is very similar to the histogram. Let's have a look at it and then discuss its features.

This time, we actually have to specify two commands. The first, `density()` is doing the actual calculation. The second is the usual `plot()` command. The density command is encapsuled into the brackets of the plot command. In this way, the output of the density serves as an input for the plot.

```
plot(density(muensingen$Length))
```



You can see, that's the essential elements are quite comparable to the histogram. We have an X axis visualising the values within the dataset. And we have a Y axis, this time not with frequency, but with density. This concept of density is the most difficult part to understand here. Therefore, let's postpone it for a second.

Let's first concentrate on this moving part. For this, let's assume that we are not looking at the actual value, But to a more blurred representation of the value. Like, if you are looking with half closed eyes or in not fitting glasses to a point. It will be blurred. The most intensity will still be in the centre, but there will be a halo of lesser and lesser intensity the more far you will get from the centre.

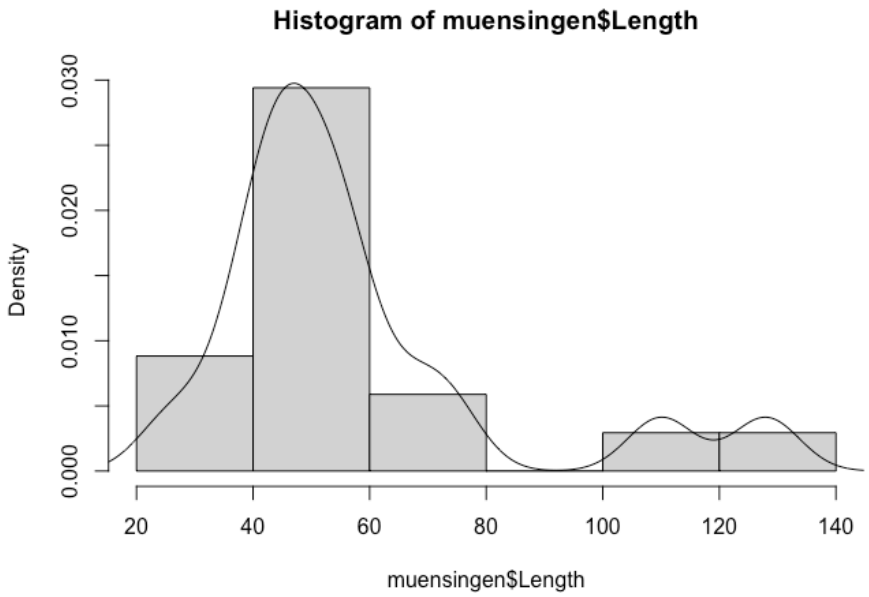
Or probably a better way to understand this is to think about the actual values as hole in a vessel filled with sand. At the hole, which is at the position of the actual value, the sand will fall down and will form a heap. This heat will be the highest in the centre so at the actual value, but it will form a small hill, starting from left and right of the actual value. If two values are nearby, the hills will merge and join in a bigger hill. This is how you can interpret the picture above. Around 40 to 50, there are the most holes in our sandbox, so here most of the sand will fall down and for the big heap. On the other hand, at 110 and 130, there are only two holes each, so they're only small hills will form.

With this kind of visualisation, we avoid a problem to artificially draw boundaries between classes. Still, we can see a representation of the total distribution of the values within our dataset. And this is more precise than the box plot,

because we get more information about the internal structuring of the data.

We can also combine the KDE with histogram. For this, we have to bring both to the same scale. The scale of the KDE is such, that the total area under the curve of the will sum up to 1. We can also be scaled histogram to be in the same scale. For this, we used to parameter 'prob=T'.

```
hist(muensingen$Length, prob=T)
lines(density(muensingen$Length))
```



We later will learn more about the concept of area under the curve, for no visionalisation is in the foreground, so we will stick with that.

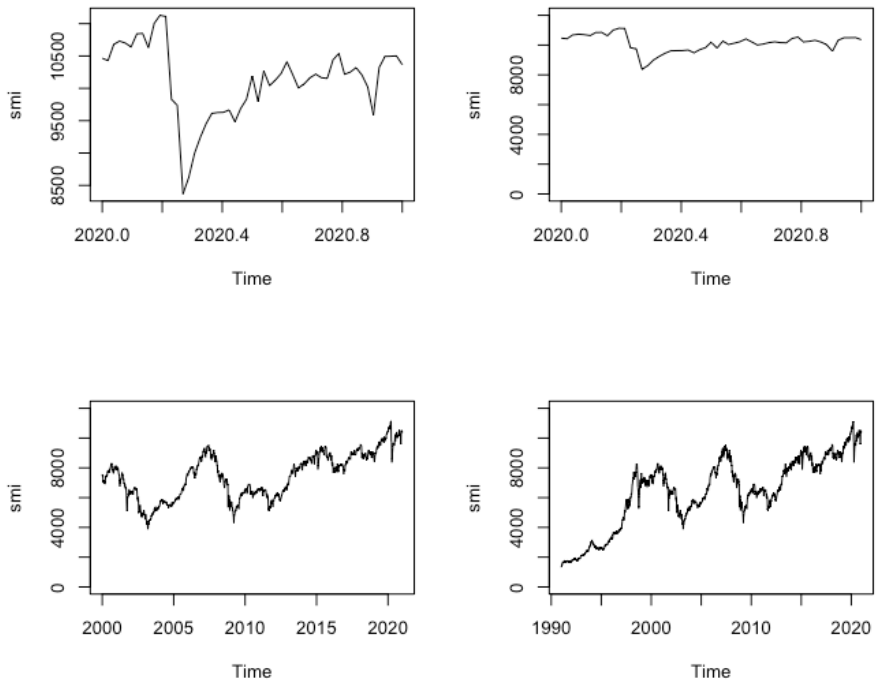
3.13 Guidelines

3.13.1 Stay honest!

Some final suggestions for guidelines, that you might like to consider, when you're plotting data. The first and probably most relevant is that you should stay honest in your data representation. It is easy to cheat with different techniques. Although for example everyone probably can understand the scales of your axis, nevertheless the presentation can produce very different perceptions.

The choice of the way how are you display your data has a strong influence on the statement and how it will be received.

Choice of display has a strong influence on the statement. Let's use the example of the Swiss stock market index to see, how different scales can influence the visualisation.



The upper left panel shows the development of the Swiss stock market within the last year. Usually you can detect the crash that took place in the course of the Corona epidemy. This is the kind of display that is quite often shown in respect to such developments. The upper limit represents the upmost value, the lower limit the lower most value. Of course, this is visible at the Y axis. In this visualisation, the development looks very dramatic. If we only change the Y axis starting from zero, this crash looks less dramatic immediately.

If we additionally enlarge our investigation window (or at least the window of data that we are displaying), it becomes obvious that much stronger deteriorations took place in the past. The lower right panel shows Shows the development starting from 1990. This shows, how low this value was in the beginning.

However you might like to interpret this developments in respect of today's severity, it is clear, that the different scales put this development in very different frames of references.

So, you could sum up the suggestions for graphical representation like so:

- Stay honest!
  - Choice of display has a strong influence on the statement.
- Clear layout!
  - Minimise Ratio of ink per shown information!
- Use the suitable chart for the data!
  - Consider nominal-ordinal-interval-ratio scale

For the last point, I have compiled a small table that can give you an advice which kind of visualisation you should choose in which kind of situation.

What to display	suitable	not suitable
Parts of a whole: few	Pie chart, stacked bar plot	
Parts of a whole: few	Stacked bar plot	
Multiple answers (ties)	Horizontal bar plot	Pie chart, stacke
Comparison of different values of different variables	Grouped bar plot	
Comparison of parts of a whole	Stacked bar plot	
Comparison of developments	Line chart	
Frequency distribution	Histogram, kernel density plot	
Correlation of two variables	scatterplot	

Sometimes, it is illustrative to look at bad examples. Such bad representation of data is also called chartjunk. I will link here to the respective chapter of the book of Edward Tufte to you, so if you would like to go deeper into the subject, can I have a look here. But with his keywords, entered into a search engine of your choice, you will be delighted with a lot of examples of how not to do it. Enjoy.

Chapter 4

Descriptive Statistics

The field of descriptive statistics serves as a mean to summarise and make accessible more complicated data. The main difference to the field of hypothesis testing, That's also called inferential statistics, is, that with descriptive statistics we are describing the sample, and do not make any inferences about the underlying population.

What specifically belongs to descriptive statistics, is not perfectly fixed. In some way, also contingency tables or graphical representation of data, like charts, can also be understand as descriptive statistics. In its more narrow sense, determines the calculation of certain parameters, this describes features of the sample, or more precisely, its distribution. Most widely used are two aspects: the central tendency and dispersion. We will learn about those aspects more in the next section.

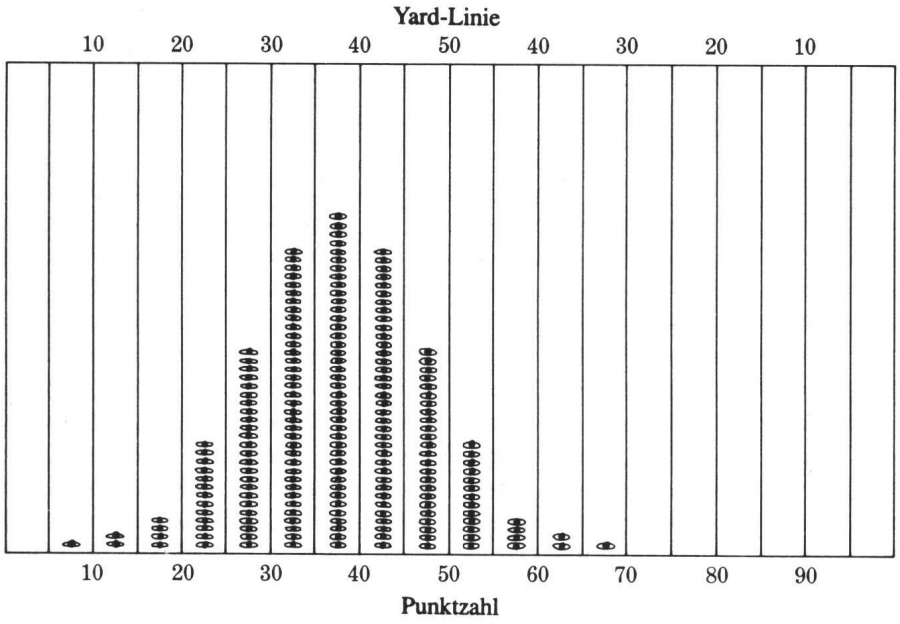
The videos for this chapter can be found here, relevant are number 17, 18 and 19.

4.1 Introduction

A distribution is the collection of all the outcomes in the sample. More precisely, you can imagine it as a way to describe the frequency of certain outcomes. Let's imagine the following example:

After their final exam, the students of a mathematical class go to the football field, and line them self up according to their scores in the exam. Starting from the 0 yard line, they move so many yards as they have scored points in the test. Now, if we count the number of students in certain areas of the football field, we get their frequency distribution among the values. So, with a distribution, we are not talking about the individual value, but about their general distribution.

From this it is clear, that it only makes sense to talk about distributions if we have more than one value (although technically also individual value is a distribution).



Studenten, die sich nach ihren Testergebnissen in Reihen auf einem Footballfeld aufgestellt haben – eine Häufigkeitsverteilung.

Figure 4.1: source: Phillips 1997

But not only students can form distributions, also other data sets can. You will see, that depending on the data type, different manifestations of distributions are possible. Nevertheless, all these versions are data distributions. The characteristics of the different distribution can be described using certain parameters.

4.1.1 Parameters of distributions

4.1.1.1 Central tendency

The central tendency can be understand as a description of those values, that are central or "typical" for a certain distribution. This also can be called centre or location of that distribution. There are multiple specific values, that can be used, to describe this specific centre, depending on the scientific question and the data. For example, certain data quality can be described only with

certain central tendency parameters. The most commonly used central tendency parameters are the *mean*, *median*, and the *mode*.

4.1.1.2 Dispersion

While the central tendency describes the most centres or most typical value, the dispersion describes how much the values in the whole sample vary around this central value. It is therefore also a measure of how much variety is in the sample. The most intuitive measure of dispersion is the *range.* We will see, that it's not a perfect measure for the whole dataset. Instead, in statistics often the *variance* or the *standard deviation* is used. We will also learn about the *coefficient of variation* as a means to compare the dispersion of different distributions.

4.1.1.3 Shape

Also not so often used explicitly in quantified descriptive statistics, of course the shape of the distribution also is very important for its interpretation. For example, the question if distribution is similar shaped on both sides of the central value, might give us quite an insight into this processes leading to this distribution. There are two other parameters that can be used to describe the shape of distribution in a quantitative way: *skewness* and *curtosis*. We will see, how we can calculate these values, and what their meaning is.

4.1.2 Loading data for the following steps

Also in this example, I would like to use the Münsingen data to demonstrate some concepts. Please download the data and then load this data into your R environment:

4.1.2.1 download data

- muensingen_fib.csv

4.1.2.2 Read the Data on Muensingen Fibulae

```
muensingen <- read.csv2("muensingen_fib.csv")
head(muensingen)
```

```
##    X Grave Mno FL BH BFA FA CD BRA ED FEL  C   BW  BT FEW Coils Length fibula_scheme
## 1  1   121 348 28 17   1 10 10   2  8   6 20  2.5 2.6 2.2     4     53             B
```

```
## 2  2   130 545 29 15   3  8  6   3  6  10 17 11.7 3.9 6.4     6     47
## 3  3   130 549 22 15   3  8  7   3 13   1 17  5.0 4.6 2.5    10     47
## 4  8   157  85 23 13   3  8  6   2 10   7 15  5.2 2.7 5.4    12     41
## 5 11   181 212 94 15   7 10 12   5 11  31 50  4.3 4.3  NA     6    128
## 6 12   193 611 68 18   7  9  9   7  3  50 18  9.3 6.5  NA     4    110
```

4.2 Central tendency

Let's start with those parameters to describe the location of the distribution. This means, with these parameters we can describe where the whole distribution is centred in the range of possible values of that variable. Some of those parameters you will probably know already very well, some might be not so familiar to you.

4.2.1 Mean

The arithmetic mean is one of the most widespread used parameter to describe the central tendency of a dataset, also in everyday life. But that doesn't mean, but it's the best value for describing this feature of distribution in every individual case. It is only suitable for metric data (interval or ratio). Why this is the case, becomes apparent if you look to the formula how to calculate it.

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

In this formula, we at first sum all the n values up, starting from the first (index 1) and ending with the last (index n). After this, we divide by the number of values. As we have seen already, in R we can recreate this formula, using some of all the values and divided by the length. But of course, since it's such an important descriptive parameter, there is a specific function for calculating the `mean()`:

```
sum(muensingen$Length) / length(muensingen$Length)
```

```
## [1] 57.588
```

```
mean(muensingen$Length)
```

```
## [1] 57.588
```

Dividing all the values by their number means, that we compare the differences of these values. This can only meaningful be done in situations, where the

distances between values have a defined metric. For this reason, we cannot use the arithmetic mean to analyse the central tendency ordinal or nominal data. Also, since with the sum every individual value is included in the way in the calculation of the mean, it describes the whole distribution and not certain values within.

Although the arithmetic mean is the most commonly used central tendency value, it has certain features that are not optimal for describing the whole dataset. Especially, it is very sensitive to outliers, as we will see below. Alternative is the median, which is not used so often, but whose philosophy is also quite easy to be understood.

4.2.2 Median

Different from the arithmetic mean, the median is a parameter that can be calculated for metric, but also for ordinal variables. And this calculation is rather trivial: if we have an uneven number of values, to get the median result all the values and then selecting the middle value. That means, literally the value that is in the middle of the sorted vector.

In our example, we have the values from 1 to 7. If we sort them and determine the middle value, it will be 4. So 4 is the median for this distribution.

```
1 2 3 4 5 6 7
      |
```

In R, the calculation of the median is not more complicated than the calculation of the mean. Also, we don't have to sort the vector ourselves, R is doing that for us. The command is `median()`.

```
median(c(3,2,1,7,5,4,6))
```

```
## [1] 4
```

A bit different is the situation, when we have an even number of values. Here, there is no middle value as such. In that case, the median is calculated by taking the mean between the two middlemost values. Assume, we have to numbers from 1 to 8. The median will be the mean of the values four and five.

```
1 2 3 4 5 6 7 8
       |
```

Also in this case, the calculation of the median is not more complicated an R.

```
median(c(3,2,8,1,7,5,4,6))
```

```
## [1] 4.5
```

4.2.3 Mode

The last of the parameters that I want to introduce here to you for central tendency is the mode. This is simply the most frequent value of a vector. So it fulfils our definition from above, that it is the most typical value. Also, since it relies only on counting, the mode can be determined for every data quality: nominal, ordinal and metric data.

Let's assume, that we have the animal bones from an excavation and we can determine the minimal number of individuals for goat, sheep, cattle and pig. In that case, goat will be the most frequent value, and by that the mode of that distribution.

goat sheep goat cattle cattle goat pig goat

mode: goat

In R, since for nominal values the mode is trivial, and for other data types it can be rather complicated, there is no specific function to calculate the mode. Instead, in the example of nominal values, we can calculate the frequency using the function `table()`, and then determine which of the frequencies is the biggest, using the function `which.max()`.

```
which.max(
  table(
    c("goat", "sheep", "goat", "cattle","cattle", "goat", "pig", "goat")
    )
  )
```

```
## goat
##    2
```

4.2.4 Comparing Central tendency parameters

As I've said already, for certain data types or levels of measurement only certain central tendency parameters are available. You can see a visualisation of this in the table below:

nominal	ordinal	intervall+
mode	mode	mode
-	median	median
-	-	mean

But every ability for different data quality is not the only reason, why are you should consider having a look to other parameters for central tendency beside the mean. What other reason is, that the mean is strongly affected by outliers. This is not so much to keep the median, and the mood is hardly affected by any outliers at all. To demonstrate that in a practical matter, let's look at the following distribution. Here we have most of the time the value is between one and nine but only one time the value 120. This value does not very well represent any typical value of this distribution. So far a good value for central tendency it should not have so much of an influence.

```
test<-c(1,2,2,3,3,3,4,4,5,5,6,7,8,8,8,9,120)
```

If we calculate the mean of this dataset, it will be above any other value then the one outlier. So in that case it's not really the central tendency of the whole dataset.

```
mean(test)
```

```
## [1] 11.647
```

On the other hand, the median is virtually unchanged by this extreme value. Because it's only one very high value, it also counts in the calculation of the median only as 1.

```
median(test)
```

```
## [1] 5
```

And the same is true for the mode: because this outlier is only one time present, it will also not affect the mode.

```
which.max(table(test))
```

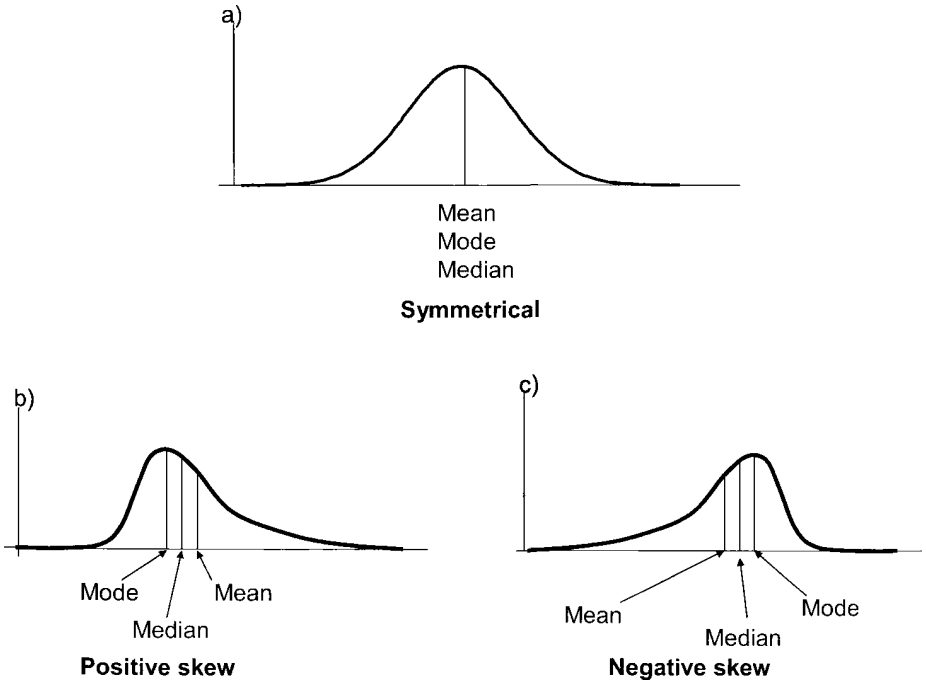
```
## 3
## 3
```

Although the mode is very insensitive to outliers, from a practical perspective in most of the cases it also represents very well essential value of the distribution, when it comes to metric or ordinal data. Only when a more or less symmetric distribution is present, the mode might be helpful, like in the example below.

```
which.max(table(c(1,2,2,3,3,3,4,4,4,4,5,5,5,6,6,7)))
```

```
## 4
## 4
```

We haven't talked about symmetry so much, but also when it comes to distributions, this concept is quite intuitive. When we have a small amount of items with small values, a major amount of items with intermediate values, and again a small amount of items with small values, then we have something that can be considered as a symmetrical distribution. When we have more small values then intermediate or higher values, or more high values then intermediate in small values, then we have a skewed distribution, which is not symmetrical. You can see examples of this in the image below. Here, you can also see, what happens to the different parameters of central tendency in respect to each other, when we have a positive or negative skewed distribution.



In case of a positive skew, the median will be smaller than the mean, since the meeting is strongly influenced by outliers. So a small number of high values will 'drag' mean towards the higher values. Conversely, if we have more high values

than low values, then the main will be smaller than the median, for the same, but inverted reasons. This means, just by comparing these two parameters of central tendency, we can already make a statement about the shape of the distribution, without even having to plot it.

Exercise 4.1. Analyse the measurements of the width of cups (in cm) from the burial ground Walternienburg (Müller 2001, 534; selection):

- tassen.csv

```
tassen<-read.csv2("tassen.csv",row.names=1)
tassen$x
```

```
##  [1] 12.0 19.5 18.6 12.9 13.2  9.9 19.5  8.4 21.0 18.9  7.5 18.9  8.1  9.0  7.8  9.9 10.2  8.1
```

Identify the mode, median and mean and determine if the distribution is symmetric, positive or negative skewed.

Solution

```
mean(tassen$x)
```

```
## [1] 13.677
```

```
median(tassen$x)
```

```
## [1] 12
```

```
which.max(table(tassen$x))
```

```
## 8.1
##   3
```

```
# The median is smaller than the mean. The distribution has a positive skew with more smaller tha
```

4.3 Dispersion

In the previous chapter, we have learnt different ways how to express to central tendency of the dataset. But with specifying the location, we have not sufficiently described our dataset yet. One of the more relevant characteristics of

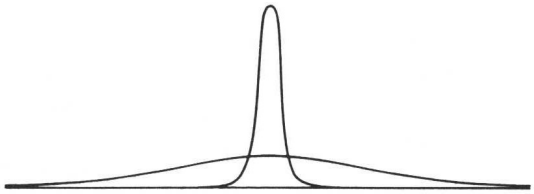


Abb. 4.1 Zwei Verteilungen mit denselben Ns, aber unterschiedlicher Streuung.

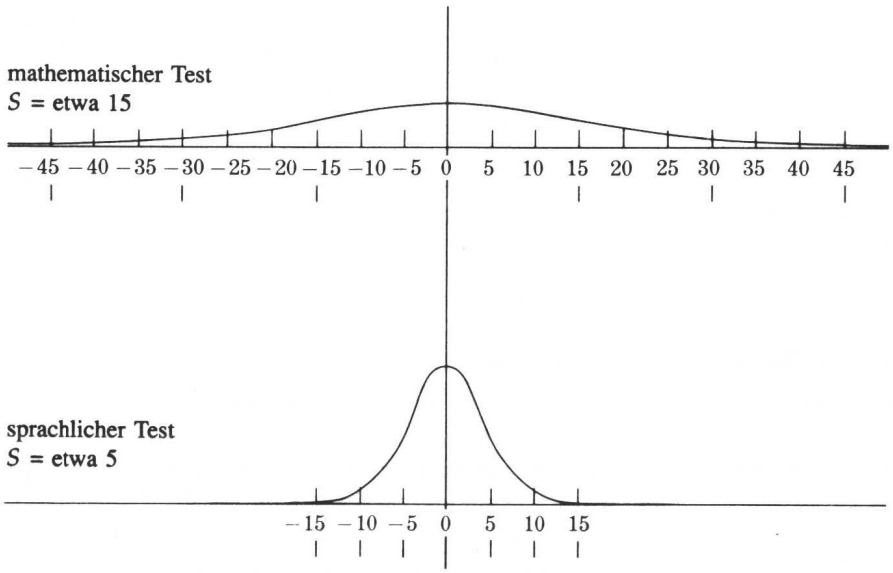


Figure 4.2: source: Phillips 1997

the dataset is, if the data are rather close together, or if they are distributed over a larger range of the values. Consider for example the distributions below:

The upper image shows two distributions with the same centre, but with different spreads. The figures below show the same situation, but this time with actual values. You can see, that both distributions are centred around the value zero, but the upper distribution ranges from -45 to 45, while the distribution below is much more centred.

To describe the differences, we have parameters of dispersion, that quantify, how wide the data are distributed over the range of values.

4.3.1 Range

The simplest of this parameters is the range. The range gives the range from minimum to the maximum value of the dataset. Practically, usually the range is constituted of two values, the minimum and maximum. In R, the function to get the vector of the range of the dataset is `range()`.

```
range(muensingen$Length)
```

```
## [1]  26 128
```

```
range(tassen$x)
```

```
## [1]  7.5 26.1
```

With the range, by definition we get the highest and the lowest value. This means on the one hand, that by constructing the range only two values are considered, just these two. All the other values in the dataset do not play any role in the calculation of the range. Therefore, the range does not describe the whole dataset very well. Also, by definition we consider the most extreme values of the dataset. This means, that the range is very sensitive for outliers.

A better measurement for the dispersion of the data should have the following characteristics:

- Less sensitive to outliers
- Broader representation of all the data in the dataset

4.3.2 Towards a better parameter for dispersion

We have already seen, that the mean, although also sensitive to outliers, at least consider all the values in the dataset as constitutional elements for the

parameter. A good description for the dispersion might be the mean distance of all values from the mean. The following plot shows us the values of the Münsingen dataset from a different perspective. Every line represents one fibula. We have substracted the mean of the length of the different fibulae from the individual values. Again, we can see our two unusual long fibulae as high values. The bars each represents how much smaller or bigger the fibulae are in respect to the mean.

Now, to get a value for the dispersion of the dataset to consider all individual items, we have to calculate the mean distance from the main. But we cannot simply substract the mean from each value and calculate the mean from it because this would result in zero. The reason for this is the very definition of the mean.

$$\sum_{i=1}^{n}(x_i - \bar{x}) = 0$$

In our calculation, we must ignore if the difference is negative or positive. We have to consider the absolute difference, not the relative one. To get rid of the sign, easiest way is to square the result.

$$\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Now we get to some of the squared differences of each value to the mean. To get to mean after differences, we have to divided by the number of cases.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

With this, we have tools to calculate the mean differences from the mean. The formula above shows the calculation of the so-called variance for the population. It's notation is using the Greek letter sigma to the square. But because we are dealing here with a sample, and not the full population, the resulting estimation of this mean difference might be biased. For that reason, this bias must be corrected. For reasons, we cannot explain fully here, the easiest way for correcting against a bias that case is to substract one from the number of cases. This gives us our first parameter for dispersion: the empirical variance.

4.3.3 (empirical) variance

As explained and introduced above, the empirical variance is a measure for the variability in the data that is more insensitive against outlier then the range. It is equal to the sum of the squares distances from the main divided by the number of observations -1. Formula notation is s^2.

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

Let's use this formula above, transform it into R code, and compare it to the results of the actual function `var()` from R.

```
sum((tassen$x-mean(tassen$x))^2)/(length(tassen$x)-1)
```

```
## [1] 31.111
```

```
var(tassen$x)
```

```
## [1] 31.111
```

In this case, we have compared the diameter of the cups. The diameter themselves are measured in centimetres. Since in the formula above we square the measurement, the unit is square centimetre. Also the resulting unit of the variance is square centimetre. The values themselves are squared centimetres. Of course, it is much more convenient, to have a measurement of the mean distance from the mean that is in the units of the actual measurement. To get rid of this square in the units as well as in the values, of course we can just take the square root. If we are doing that, we end up with what is called standard deviation.

4.3.4 (empirical) standard deviation

The formula for the standard deviation is essentially the same as for the variance. The only differences, like we have explained above, is that we take the square root.

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})}{n-1}}$$

Again, if we compare the results from the recorded formula to the actual functioning are, the function `sd()`, the results should be the same. Keep in mind, that we are talking about the empirical version of these calculations, suited for samples.

```
sqrt(sum((tassen$x-mean(tassen$x))^2)/(length(tassen$x)-1))
```

```
## [1] 5.5778
```

```
sd(tassen$x)
```

```
## [1] 5.5778
```

With this, finally we have the mean differences from the mean for our dataset. This parameter of dispersion is one of the most widespread used for describing distributions. It has some specific features that make it very useful. Some of them we will learn about later in the course.

With what we just achieved, the measurement of the standard deviation is in the exact same units as the measurements themselves. Most of the time, this is desirable. Especially, if we would like to understand better what this mean distance mean fall our individual values. But sometimes, you might like to compare spreads of distributions, that have different central tendencies, or to put it in other words, different locations. For this case, it is more helpful, to have a measurement of dispersion that is unitless. This is the last parameter of the dispersion from this branch that we will learn about.

4.3.5 Coefficient of variation

Let's assume, we wonder whether the length of the fibulae in our Münsingen case it's more variable then the foot length or the other way round. We cannot simply compare the standard deviations, because the feet of the fibulae are shorter than the whole fibula by definition. Consequently, also the variability will have smaller values for the foot than for the total length.

In such situations, we need a unitless measurements. We would like to make the value independent from the location of the distribution. To make something independent, most of the time you have to divide by it. So if we divide the standard deviation (measured in the unit of the measurement) by the mean (measured in the unit of the measurement), we get a parameter that is independent from the location (and also, since the units negate each other, has no unit). This coefficient of variation can be used to compare distributions that do not share the same location in the value range. We can compare apples with oranges.

In R, there is no specific come out for that. But it's easy to calculate it anyway. All we have to do, is to divide the standard deviation `sd()` by the mean `mean()`.

```
sd(muensingen$Length)/mean(muensingen$Length)
```

```
## [1] 0.4509
```

```
sd(muensingen$FL)/mean(muensingen$FL)
```

```
## [1] 0.77325
```

The result is, that the variation coefficient for the length is smaller than the valuation coefficient from the foot length. It seems, that the feet of the fibulae differ more relatively then the total length.

4.3.6 Quantile

The concept of the standard deviation and related measurements is very similar to the concept of the mean, that we have learnt as one of the parametres for location of the dataset. Another way to describe the dispersion of the dataset is the concept of quantiles, it is more related to the concept of the median that we also learnt about above. We already have introduce this concept, when we have talked about the box plot.

Quantile in general are arbitrary points in the range of a dataset where this can be divided into parts. Specific quantiles are the quartiles. Quartiles divide the data into four equal parts, that means, parts of equal number of items. So let's assume, we have 16 items. The first quartile consists of the first four of this sorted items and so on.

Let's assume, we have 13 values from 1 to 13. If we sort them and marked the beginning of each quarter of the items, We would end up dividing our dataset by one (the beginning of our dataset), 4 (beginning of the second quarter of a dataset), 7 (the median of our dataset), 10 (the beginning of the fourth quarter of our dataset), and finally 13 (the highest value at the end of our sorted the dataset). When we have an even number of items, the same rule applies like in the case of the median: we take to mean of the two centremost values.

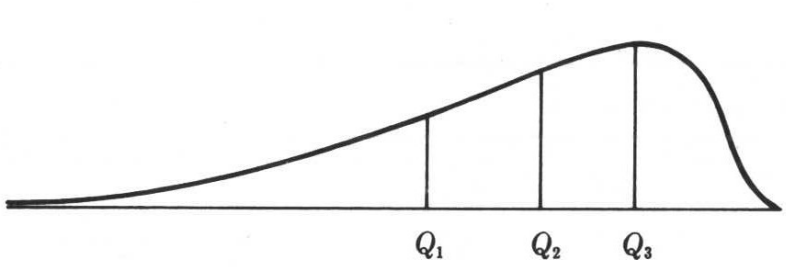
If we now plot our data on an X axis with the actual values, we can mark the positions of the quantile stare. Such a plot could look like this:

In R, the command to calculate the quantiles is quantile. To make the confusion total, the default setting of the quantiles is to calculate the quartiles.

```
quantile(tassen$x)
```

```
##   0%  25%  50%  75% 100%
##  7.5  9.0 12.0 18.9 26.1
```

But we also can parameterised it differently. For example, we can calculate the so-called percentiles. And example below, we calculated the position of the cut points in that way, that we get the 10% intervals of the data.



Linksschiefe Verteilung mit einer in Viertel geteilten Fläche.

Figure 4.3: Phillips 1997

```
quantile(tassen$x, probs=seq(0,1,0.1))
```

```
##    0%   10%   20%   30%   40%   50%   60%   70%   80%   90%  100%
##  7.50  8.10  8.52  9.27 10.02 12.00 13.08 18.81 19.38 20.31 26.10
```

The position of the quantiles (or the quartiles) depends on the spread of the data. In case of the quintiles, the first and the last value that is calculated here represents the minimum of the maximum value. So it is clear, that their position is strongly related to the spread of the data. But also the position of the beginning of the second and the third quartile is dependent on how strong the dispersion is. If it's higher, it's distance will be bigger. That is the reason why the distance between the beginning of the second and beginning of the fourth quartile is a measurement of the spread. Its relationship to the standard deviation is the same like the relationship between the mean and median in respect to the location of the dataset. This measurement is also called the Interquartile Range or `IQR()`.

```
IQR(tassen$x)
```

```
## [1] 9.9
```

Like the median is more insensitive to outlast in the main, the Interquartile Range is less sensitive to all clients that the standard deviation. But since we only consider the centre at half of the data for the calculation of this measure, we lose some information about the so-called tails of the distribution, that is (in this case) the lowest and highest quarter.

Exercise 4.2. Analyse the sizes of areas visible from different megalithic graves of the Altmark (Demnick 2009):

- altmark_denis2.csv

```
altmark<-read.csv2("altmark_denis2.csv",row.names=1)
head(altmark)
```

```
##         sichtflaeche region
## La01            2.72  Mitte
## Lg1            26.78  Mitte
## Li02           26.96  Mitte
## Sa01           27.05  Mitte
## Li06           32.93  Mitte
## K\xf601        34.76  Mitte
```

Evaluate in which region the visible area is more equal (less disperse).

Solution

```
# There are 3 regions in the dataset:
table(altmark$region)
```

```
##
## Mitte   Ost  West
##    14    23    93
```

```
# lets use the region as separator and
# calculate the coefficient of variation
# (because each region might have different terrain):

va_east <- altmark$sichtflaeche[altmark$region == "Ost"]
va_west <- altmark$sichtflaeche[altmark$region == "West"]
va_mid <- altmark$sichtflaeche[altmark$region == "Mitte"]

sd(va_east)/mean(va_east)
```

```
## [1] 0.91189
```

```
sd(va_west)/mean(va_west)
```

```
## [1] 0.94471
```

```
sd(va_mid)/mean(va_mid)
```

```
## [1] 1.0057
```

```
# it seems, that the visible areas differ
# the most in the center of the working area
```

4.4 Shape of the distribution

As interesting as it might be, as difficult it is to describe the shape of a distribution in a single quantitative way. There are multiple ways how distributions can differ from each other: might it be related to the skewness, to dispersion of the data, to the symmetry or to the number of peaks.

Below, there is an image from Bortz 2006 dead visualise some of these possible parameters.

As already announced, we will discuss here to shape parameters that can be calculated: First, the skewness, of which we have already seen that it can be positive or negative. But there is also a way to calculate the value of the positivity or negativity. The second parameter is called Curtosis (curvature). This parameter indicates if the distribution is a letter or a steeper, compared to the so-called standard normal distribution.

In both cases, it might make sense to plot data and then calculate the values to get a feeling what they do mean. On the other hand, these values are rarely used in practise, so we will use this opportunity to introduce a new concept in our: how to make your own functions.

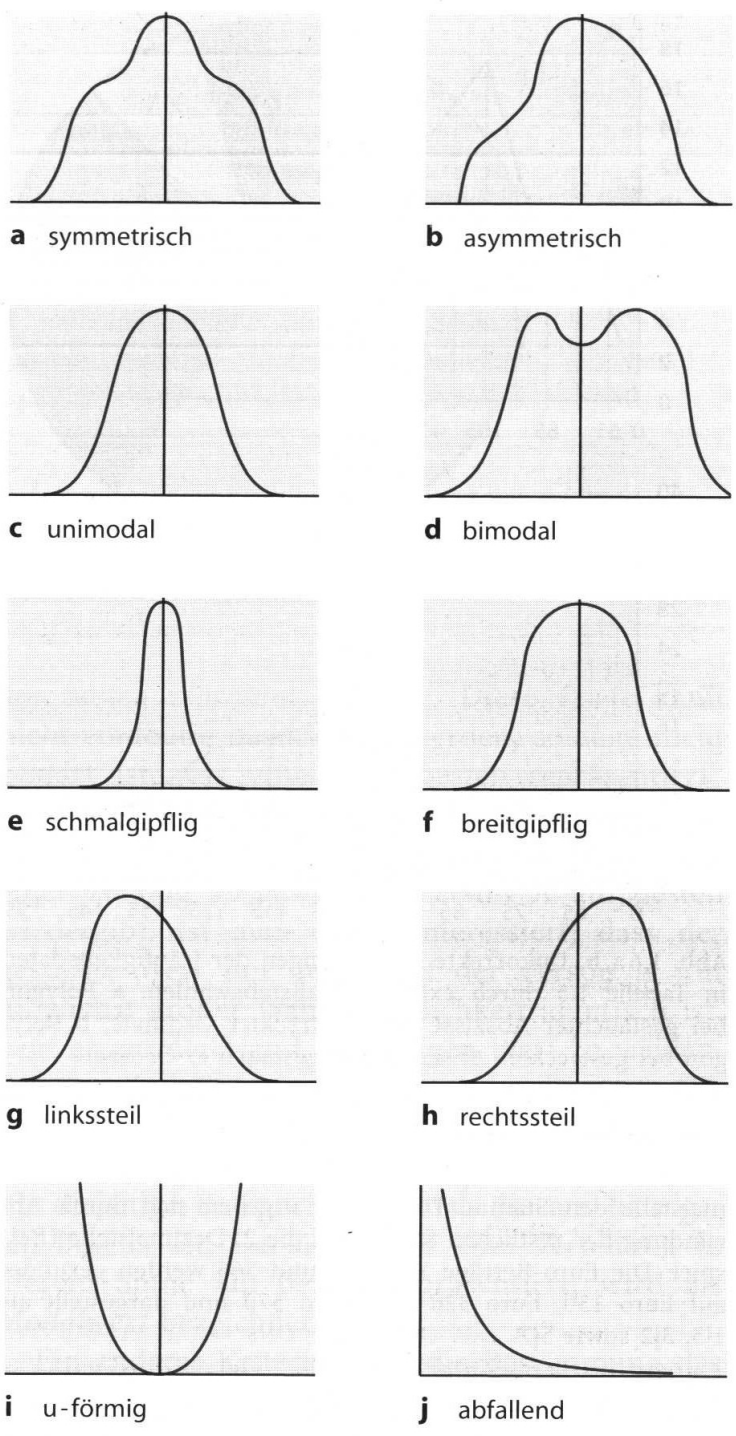
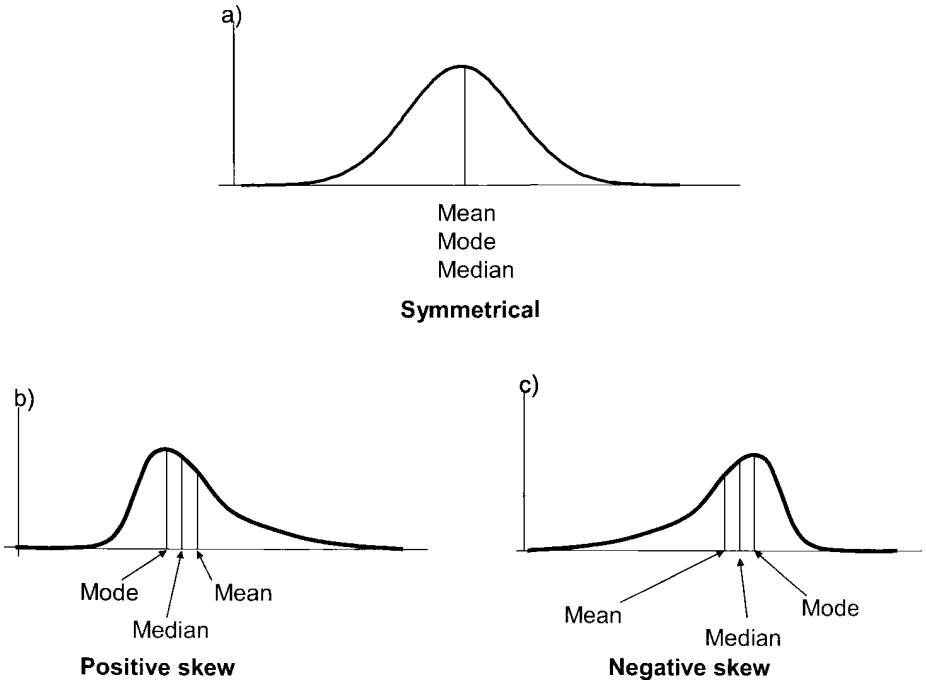


Figure 4.4: Shape of distributions (after Bortz 2006)

4.4.1 Skewness



It's already introduced, skewness is a measure of how skewed your dataset is. On the one hand, it can be distinguished between positive and negative skew. This can be estimated by comparing the mean and median of the dataset. Also, you can try to describe how intense the skewing of the dataset is. For this, probably the easiest option is to look at the plot and describe it. But there is also a way to calculate that, given by the formula below:

$$\hat{S} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{n * s^3}$$

Here we will not take a part of this formula, but use it as a cooking recipe to produce our own function. For the interpretation, it should be said, that positive values of skewness indicates a positive skew, while negative values indicate a negative skew. The reason, why it's reasonable to create your own function here, is the fact, that there do not exist a ready-made function for this purpose in the base package of R. So lets build our own:

As I have already indicated, everything in R is a variable. This is also true for functions. There are specific function to feel a variable with a function. You will not be surprised that the name of the function is `function()`. Whatever is inside of the (this time currently) brackets off the function call will be evaluated every time the function is called. In the round bracket of the function `function()` there are the variables that might be evaluated inside of the function.

So in our case, the skewness is calculated by dividing the sum of the differences of the individual values to the power of three by the number of the values times the standard deviation, also to the power of three. This is what takes place inside of the function definition. Let's dissect that from bottom to top. The last line `skew` is the value that is returned after the function has been called. This variable `skew` has to be filled. This takes place in the line before that. The value m3 is divided by the denominator of the formula. Of course, this value M3 is filled with the numerator of the same formula. Also, you can see inside of this function body the variable `x`. This variable comes from the outside of the function and represents the actual dataset. That this function expects this variable `x` is indicated by the fact, that in the rounds brackets of the function call `function()` this `x` is mentioned.

```
skewness <- function(x) {
  m3 <- sum((x-mean(x))^3) #numerator
  skew <- m3 / ((sd(x)^3)*length(x)) #denominator
  skew
  }
```

If now we have to find our function `skewness()`, we can use it as any other regular function in R.

```
test<-c(1,1,1,1,1,1,1,1,1,1,2,3,4,5)
skewness(test)
```

```
## [1] 1.4068
```

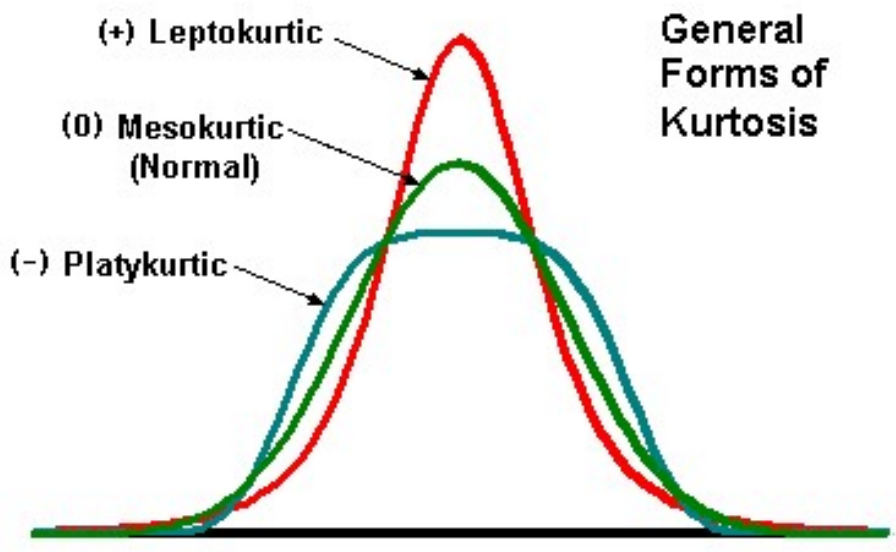
```
test<-c(3,3,3,3,3,3,3,3,3,3,3,3,2,1)
skewness(test)
```

```
## [1] -2.2312
```

With this basic concept, you can structure and reuse parts of your code all over in your analyses. Also, the actual result of our function seems to be fitting. What we expect to be a positive skewed distribution results in a positive number and vice versa.

4.4.2 Kurtosis

The second parameter that we may talk about in this context is to kurtosis. It describes the curvature of the distribution in relationship to the standard normal distribution.



Positive values of kurtosis means, that the distribution is steeper than a normal distribution, while negative values mean, that it is flatter than a standard normal distribution. Of course, you can read the curvature from a plot, but also there is a way to calculate this. If you look to the formula, it will be quite familiar to you, because it resembles very much the formula for skewness. The only difference is, that instead of to the power of three no we calculate to the power of four.

$$K = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{n * s^4}$$

So, of course, we also can write a function for this. It will be very similar to the function as we've just programmed: You only have to replace two numbers. Oh, and by the way probably also the name of one of the variables.

We write a function for that, too:

```
kurtosis <- function (x) {
  m3 <- sum((x-mean(x))^4)
  k <- m3 / ((sd(x)^4)*length(x))-3
  k
  }
```

Let's test this out with one very steep and one very flat distribution.

```
test<-c(1,2,3,4,4,5,6,7)
kurtosis(test)
```

```
## [1] -1.4688
```

```
test<-c(1,2,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,6,7)
kurtosis(test)
```

```
## [1] 2.0114
```

4.5 Take Home

There are many other possibilities to describe data, values, and distribution. The most relevant and most widely applicable concepts are those of central tendency and here especially median and mean, and those of dispersion, here especially variance and standard deviation. If you have understood this computations and they are meeting, you are very well prepared for the weather, more elaborated analyses.

Chapter 5

Nonparametric Tests

5.1 Inductive statistics or statistical inference

Until now, we have primarily described and presented the data of our sample using descriptive statistics and graphical visualisation. We also used graphical visualisations to explore our dataset in the sense of explorative statistics. In all of these techniques, we stayed on the page of the sample. We did not make any conclusions about the underlying population.

This is no different, when we come to the field of inductive statistics, or statistical inference. Here, we explicitly try to estimate characteristics of our population using our sample. Since in nearly all cases, our sample will be smaller (in archaeology actually very much smaller) then our population, we have only access to a small part of the information describing the full population. So necessarily, when we base our estimation only on a fraction of the population, our estimation will be wrong. But it doesn't need to be true, it just needs to be sufficiently correct. Nevertheless, it is essential to keep that in mind: whatever comes out of the statistical analyses, only is to with a certain probability. The knowledge gained is always statistical meaning, there is a certain chance that the world is like we assume it. We have to have a link from the sample that we have at hand, to the population. This link is provided to us by probability theory. We will explain that a bit more detailed, when it comes to parametric tests.

5.2 Population and sample

5.2.1 Repetition:

A small repetition of what we have learnt in the beginning: We call **population** the collection of all items that are relevant for our investigation. But in most cases, we cannot investigate the full population. Therefore we investigate only a fraction of it. This fraction, that we ideally select on certain criteria (represent activity), is called our **sample**. We always should keep in mind, that the dataset that we are analysing is only a random selection of the whole collection of items that we are really interested in. In so far, it is like our probe that we send out as spaceship Enterprise into the nebula of data that lie in front of us. This is especially true in archaeology, where neither the population nor the conditions under which our sample is created is under our control. We never will be able to access the population, and we never will be able to verify our interpretation with it.

5.2.2 Parameter

Even if we have no way of precisely knowing them, our population always has certain values or distributions of values, there are fixed. For example, there is a specific value for the mean foot length of all the La Tène B fibulae ever produced (given, that we can agree on what a La Tène B fibula is). This value is there. It is an actual number. But we never will know it, and it's lost in the depth of time.

Such variables, that we can't be sure exists and have a certain value we call **parametres**. This parameters can only be estimated by us using the sample. So, they are not accessible for us, nevertheless that they really exist and really have a specific value.

Also, samples have parammeters. But these parameters are accessible: we can measure them. But we have to distinguish these parameters from the actual parameters of the population. Quite often, this is already visible in the notation that is used in the formula, which differentiate between sample and population. For example:

Population	Sample
μ : mean of the population	$\bar{x}$: mean of the sample
σ: standard deviation of the population	s: standard deviation of the sample

Since we can only measure the parameters of the sample, we use them, and some knowledge about the general distribution of values in random processes, to estimate those values of the population. This is done in statistical tests, where it is estimated, if and how likely certain values of a sample, under certain

conditions of a random process, result from the population with a certain parameter. The quality of the statement of a test therefore depends on the choice of the sample (representativity)! For this, we make a hypothesis about the value of the population, and then we test this hypothesis using statistical hypothesis testing.

5.3 Statistical Hypothesis testing

5.3.1 Validation of an assumption about the population

As I have just explained, does statistical hypothesis testing takes place before the background of the sample. We make a hypothesis about the value of the parameter of the population, and then we estimate, how likely the sample that we have at hand could have resulted from such a population. If it is very likely, then it is an indication that our hypothesis is not too bad. If it is very unlikely, that might be indication that there is something wrong with our hypothesis.

Before we start to talk about actual and individual values, most of the time hypothesis actually unfold them self more as general questions like the one below:

How probable is it that two or more samples descend from the different/the same population?

(eg. Is the custom of grave goods for man and women so different that two different social groups are visible?)

How probable is it that a given sample descend from a population with certain parameters?

(Is the amount of grave goods random or is a pattern visible?)

In the first place, when we have two samples and try to estimate if they originate from the same or from different populations, we have a test for independence. That means, we test, based on the samples, if the population of both values are independent from each other or not. In the second case, which is also called goodness of fit test, we test how good our data fits to a specific assumption, like I have explained above. But it is very difficult to prove something about the population, given that we have a sample. Maybe, we just have to wrong sample? Maybe, while our sample just doesn't fit to our assumption, the sample next to it would probably do. It is much easier to falsifier statement using a sample: only one black swan falsifies do you hypothesis, that all swans are white. Therefore, in statistical hypothesis testing, we use a detour over falsification to make our hypothesis more plausible. This detour is called null-hypothesis.

5.3.2 Null hypothesis

In statistical tests most of the times not the statement is tested which one expects to be true, but one tries to disprove the statement which one expects to be wrong: the null hypothesis. This hypothesis states mostly, that a association do not exists or that there is no differences between the samples and the distribution of the observations is by chance. So, most of the time, it is the rather boring standard situation, that exciting scientific investigations try to disprove.

An example: Is the composition of grave goods different between male and female deceased?

This is the scientific question. Now, there are two possible answers to that, resulting into possible hypothesis:

H_0: The composition is the same

H_1: The composition is different

So instead of proving our hypothesis one, we try to disprove or falsify its opposite. When we have shown, that the assumption, that the composition is the same (or only different due to random chance), then we have shown at the same time, that the composition is different. We did not say anything about how it is different, or why it is different, we only state, that is different. As we have shown above, this has two reasons:

1. It is (logical) easier to prove, that a statement is wrong (falsify) then to prove that a statement is true (verify).
2. Most of the times it is easier to formulate a null hypothesis (How exactly is the composition different?). It doesn't make a assumption about how the character of a association/difference exactly is.

So the "workflow" of the statistical test is the following: At first, usually from our scientific question, we have our alternative hypothesis. At the moment, it is probably not with an alternative, because it is the only hypothesis that we have. Usually, we hope or expect to find something interesting in the data. For example in relation to gender differences in burial items:

Construction of a alternative hypothesis:

The composition of the grave goods is different between male and female deceased.

From this assumption, we build our no hypothesis in such a way that we are stating the opposite of what we might expect:

Construction of the null hypothesis:

The composition of the grave goods is the same in male and female burials.

[To be honest, this is a bit of cheating. If you get further into the discussion about the validity of statistical analyses, you will learn decides to discussion on the P value as such, that one should actually only compare real scientific hypothesis with each other, and not such a strawman like another hypothesis. Nevertheless, this is (quite successful) practice in frequentist statistical analyses since decades, and in most of the time it actually also works quite well. Only to let you know, that this is not the only way of how you can interpret this workflow.]

Now, that we have our normal hypothesis, we can test with our data from our sample how likely it is. How do you specifically is done, depends very much on the statistical test that we are using, and the data quality, that we are investigating. But the general logic is always the same: the next step is the

Test of the null hypothesis

From this, we have two possible results. Either:

If the result of the test is significant:

Then we have evidence, that our Null Hypothesis might not be true. And, depending on the level of security that we would like to have, this might be rather overwhelming evidence. In that case, we reject the null hypothesis. This also means, that we can choose the alternative hypothesis. In our example:

The composition of the grave goods is different between male and female deceased.

If the result of the test is not significant, This means, that we do not have enough evidence to securely falsify the null hypothesis. Or in other words:

The null hypothesis could not be rejected.

This now does not mean, get the null hypothesis is actually true. You see that quite often, that When statistical test is not significant, that interpretation is that there is no difference for example between two populations. But this is not correct because the null hypothesis can be kept Ivor, because it's true (or better not wrong), or, because there are not enough data. Statistical significance doesn't and cannot differentiate between these two possibilities.

5.3.3 One-tailed/Two-tailed hypothesis

Sometimes, you see one-tailed or two-tailed as qualifier resulting from statistical tests. This comes from the fact that technically in most of the hypothesis testing we compare our value to a range of theoretical values. In such situations, our value can be bigger or smaller than this range. Now, it depends on the question asked and therefore on the hypothesis, in which case we can reject our null hypothesis. Probably can be best explained using example:

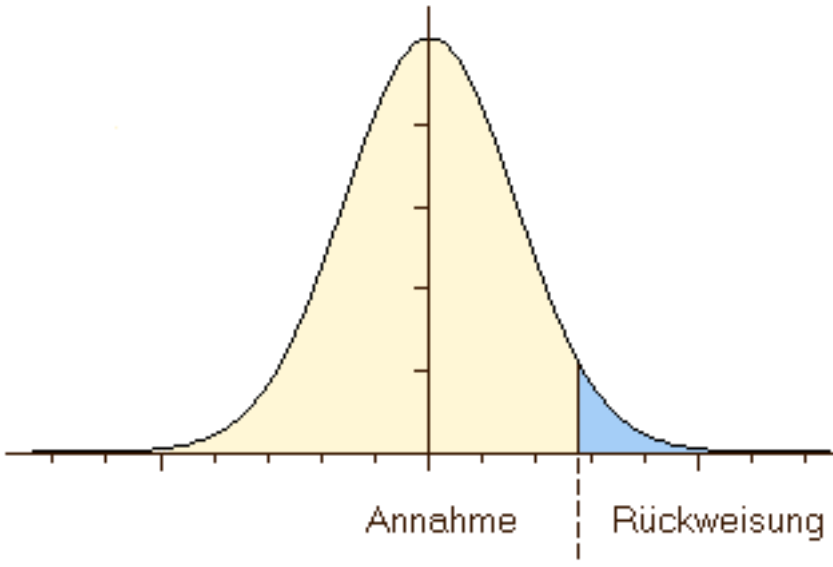
Is the number of grave goods in female burials higher than in male?

With this type of question, there is only one way in which our hypothesis (or it's null hypothesis) can be falsified. Only in the case, when the number of greatcoats in May burials are higher, then we have a falsification of our hypothesis. There is only one way, one tail of the values, on which it can be falsified.

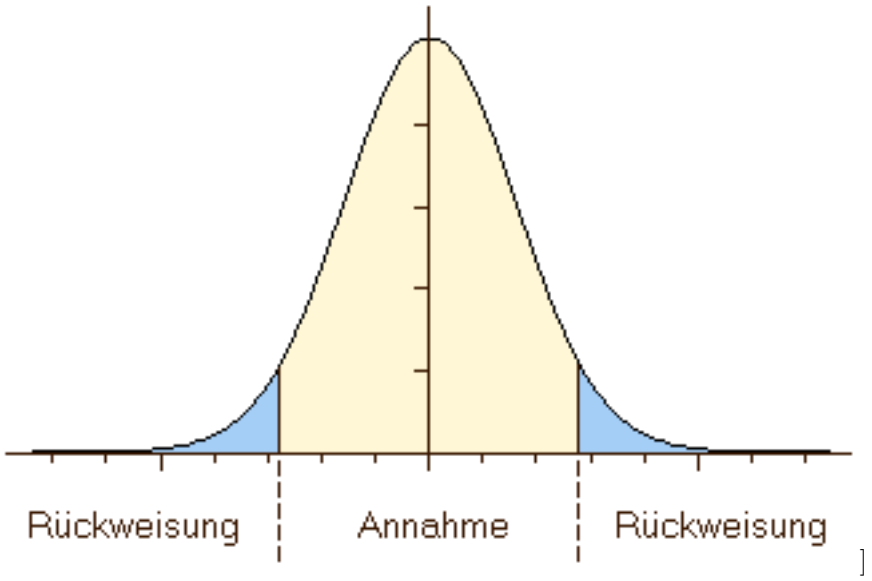
Is the number of grave goods in female burials different from male?

Here, the number of grave goods in female burials can differ into directions: There can be more or less burial items. So on both tails of our value range (smaller, bigger) there is falsification of the null hypothesis possible. Therefore, this is a two-tailed hypothesis.

Here, we have more chance for a random process to result in the pattern that we are observing. That's why in statistical tests the result is often two significances



(onetailed, two-tailed). .center[



5.3.4 Stat. Significance

5.3.5 How true is true?

Now, we quite often have used the term significant. Of course, in common language this means important. In the realm of statistical analyses, significance means a measure of how certain decision for one or the other hypothesis is. Or, to put it the other way round, it is a measure of uncertainty.

We have already introduced the null hypothesis, and there we have explained that statistical significance decides whether or not to accept the null hypothesis. More specifically, significance is exactly a measure of how probable an error is when we reject the null hypothesis. That is not directly the same as how likely the null hypothesis itself is, although quite often it will be mixed up with this meaning. One important factor in this difference is, that also sample size and with that certainty of evidence against the null hypothesis is factored in here. But from this perspective, it also makes quiet sense that based on this criterion the null hypothesis is rejected or not. This measure of course is a continuous variable, so there are no specific values per se that would define a threshold. Nevertheless, there are some arbitrary classical thresholds:

- 0.05: significant, with 95% probability the decision is right.
- 0.01: very significant, with 99% probability the decision is right.
- 0.001: highly significant, with 99,9% probability the decision is right.

In archaeology, where no life depends on our decision, usually the 0.05 level is perfectly fine for decision-making. In medical trial or situations, where you have a lot of experimental data, you probably would like to have stricter significance levels than that. But technically, no one can stop you to introduce your own significance level if you like.

This measure of significance is also often called p-value in the literature in the English literature, or alpha (α) in other languages. If you look up these names, you probably will find that there is quite a discussion going on about the value of the p-value in scientific analyses. Since this year it's a beginners course, we will not go into this debate. For us the p-value is significant enough, and we will stick to this arbitrary 0.05 measure. This measure also means, that in 1 out of 20 cases the null hypothesis will be rejected even though it should not be rejected. When interpreting statistical results, one should always keep this in mind: even with seemingly objective statistical methods, there is quite a big margin of error possible. Actually, there are two possible arrows here: we can except the null hypothesis, even if it is not true. Or, we can reject the null hypothesis even if it is true. We will learn about these two versions in the next section.

5.4 - und -error

As I have described above, we can go wrong in two ways with the rejection of a null hypothesis in statistical investigations:

The null hypothesis is rejected although it is true -> *Type I error, false positive, α-error*

Such an error would be, for example, when the result of a pregnancy test shows a pregnancy although there is none.

The null hypothesis is not rejected although it is wrong -> *Type II error, false negative, β-error*

In this situation, the example would be the result of a pregnancy test if it shows no pregnancy although there is one.

So in general, there are two situations where we can decide correct: If we accept the null hypothesis, and it is the correct choice, or we reject the null hypothesis, and that is the correct choice. If we reject the null hypothesis, but there is in reality no difference between our samples, then we make this type one error. In that case, we would think that we gained some new knowledge, so we probably will build further interpretations on top of that. That makes type one errors more severe. Because, with a Type II error, when we keep the null hypothesis although there is a difference in our data justifying the alternative hypothesis, if we correctly interpret the statistical results we just miss an opportunity to find an interesting pattern. When another hypothesis cannot be rejected, we should

not learn anything about the population.So, not so much damage should result from that.

	True condition: H0 (There is no difference)	True condition: H1 (There is a difference)
By the use of a statistical test the decision was made for: H0	Correct decision	Type II error
By the use of a statistical test the decision was made for: H1	Type I error	Correct decision

Nevertheless, both situations are annoying, so that is why statistical tests usually try to avoid both types of errors. But they can do so only in so far as they adapt their strictness. If they are too strict, they will not reject the null hypothesis to easily, leading to type 2 errors. If they are not strict enough, do you wanna risk type 1 errors. So it is always a balancing act between both error possibilities. The **power** of the statistical test is its capability to avoid type two errors without risking type one error. This capability depends on the amount of information that can be put into the test itself. That means, if we have not very much information not very much evidence, then type two errors are more likely. In general, this also not very desirable, because more powerful tests enable us to differentiate more clearly between random effects and actual patterns in the data.

5.5 Parametric vs. Nonparametric

The question, what nonparametric tests mean, and what distinguish them from parametric tests, is strongly connected to what we have described just above. If we have very little information, because from the data themselves or from general knowledge they can't be put so much into the equation, then we probably will have tested and not very powerful. On the other hand, this tests might be applicable in situations, we are more powerful tests are not valid, and cannot be used. On the other hand, if we can introduce more information into our test, we have more grounds to differentiate between random effects and actual patterns. This precisely is the difference between parametric and nonparametric tests.

Parametric tests, we make certain assumptions about the distributions of the values in the population. Quite often, we assume that the values follow the so-called normal distribution. But there are also other possibilities: for example we could have a independence test that require the variance of the data to be

equal. We will learn about one of these tests later, and but alternatives we do have in this situation. Quite often, parametric tests also, resulting from their prerequisites, can applied only to metric variables.

Nonparametric tests on the other hand do not make any of these assumptions. They do not care, what distribution the data might have in the population. Also, most tests for nominal or even ordinal variables are nonparametric. Another benefit is that quite often they are applicable for rather a small sample sizes. But for this flexibility, of course, your pay a price: this tests are not as powerful as their parametric counterparts. Nevertheless, for the dirty and sparse data that archaeology usually has to deal with, we can declared that this kind of tests exist.

5.6 χ^2 test

The first statistical test, that we will cover here, is probably one of the most nonparametric tests there is. The Chi-Square test requires only two or more sets of nominal variables, and will try to test whether or not the court occurrence of values is related to each other or not.

This probably doesn't sound so universal, but in archaeological reality you can cover already quite wide range of possible questions with this. And also, you can base your interpretations much more secure, if you have the ability to differentiate between actual patterns and random results from sampling processes. Such questions might be:

Do settlements tend to be situated on rather good soil or is the distribution random?

If we can differentiate between situations, where to settlement behaviour is just independent from disorder quality, and such situations, where decide quality or a specific side quality makes a difference for the settlement behaviour, we can learn a lot about settlement behaviour in general and it economical dependencies specifically.

Do older individuals have more shoe-last celt as grave goods than younger?

If we assume, and archaeology did so, that shoe-last celt (Schuhleistenkeil) are a sign of social rank in neolithic Linear Band Ceramic groups, then their distribution among the individuals of different age classes can you give us an indication about the construction and reproduction of status and role in society. If, for example these items are evenly distributed among older and younger individuals, can we have an indication that it is not activities during lifetime that define the social rank of a person, but this rank can be inherited. Or the other way round, if these objects are concentrated in order age classes, we have an indication that only with a long life you get the allowance to have one of these objects as burial

item. [One problem here: Of course, this is a vicious circle, because the fact that these objects occur in older age classes primarily make them suspicious for being a social marker of rank in the first place.]

All these questions and the related categories are probably not the first thing, and that comes to your mind, when you think about statistical tests. Here, we talk about nominal variables. And this is especially the value of the Chi-Squared test for archaeology: Most of our variables and of our interesting questions are nominal scaled. Let's have a look how we can perform this test.

5.6.1 Facts sheet

5.6.1.1 χ^2 Test for independence of two distributions

Requirements: at least 1 nominal scaled variable (one sample case) and 1 nominal scaled grouping variable (two sample case)

Procedure with one sample: observed values are compared with expected values given a certain distribution, no expected value should be < 5; n should be > 50

Procedure with two samples: observed values of both distributions are compared with expected values if the samples would be even distributed, no expected value should be < 5; n should be > 50

If sample size is too small: Fishers Exact Test

Test statistics: χ^2

Significance depend on degree of freedom (df)

5.6.1.2 What do these facts mean

Most of this will explain itself when we come to the practical examples. The test itself requires one nominal scaled variable, that means the counts of objects within at least more than one category. So actually not the count number, but the categorisation is the original variable. The nominal scaled grouping variable describes simply the fact that we can differentiate to samples, if we would like to compare to samples. We have to be able to distinguish between apples and oranges. In both cases, we compare the observed values of our dataset with some expected values, that will result from certain assumptions or expectations about how the distribution of the values should be. Also this will be more clearly understandable with the practical example. But in the one sample as well as in the two sample version of the test, there is a requirement of having no expected value (the number of objects of a certain category that you would expect given your assumptions) should be smaller than five, and the total number of cases should be at least 50. Please note, that if this pre-requisites

cannot be fulfilled, you can use the so-called Fishers Exact Test instead. Its interpretation is very similar to the one for the Chi-Square test statistic. The test itself is called after the Chi-Square distribution that it uses as test statistic. In statistical tests, often certain theoretical distributions are used as shortcuts to approximates other, more complicated distributions to decide, whether or not the values might come from random processes. The last thing we have to explain to the concept of degrees of freedom.

5.6.2 Excursus degree of freedom

Very general, degree of freedom in statistics means a number of values in a calculation or equation that can vary freely. It is the amount of choices that you have before everything is determined. Let's assume the following numbers of burials, divided into different classes:

	male	female	total
cremation			201
inhumation			197
total	216	182	398

We have the total sums in the margins, we have in total 216 male burials, and 182 female burials. Also, we know already that we have 201 cremation burials and 197 inhumations. But the distribution within the table is not yet determined. We don't know, how does different values are distributed among the classes. But if we have information about just one cell:

	male	female	total
cremation	123		201
inhumation			197
total	216	182	398

Now everything is determined. We can calculate the number of male inhumations, we can also calculate the number of female creation cremations, And then also the number of female inhumations. With just one value set (and it doesn't matter which value it is actually) the whole table is determined.

	male	female	total
cremation	123	78	201
inhumation	93	104	197
total	216	182	398

This means, that this table structure has a degree of freedom of one.

df=1: if one value is chosen all other can be calculated with the help of the margins

In general, there is a very simple formula to calculate the degrees of freedom of such tables:

(number of columns – 1)*(number of rows – 1)

So we have to substract from the number of column 1, and multiplied it by the number of rows -1. You can calculate the example above with this formula, or check the sample below with the same formula and have already an estimation about how many degrees of freedom this will have.

	male	female	uncertain	total
cremation				201
inhumation				197
total	196	179	23	398

No we have three columns, and still two rows. According to our formula, we should have a degree of freedom of two. Let's try that out. If we introduce one value, there is still possibility to vary other values freely. For example, if we fixed a number of female cremations, we still don't know or can't calculate all the other values.

	male	female	uncertain	total
cremation		78		201
inhumation				197
total	196	179	23	398

Only if we add another value, for example the number of male cremations, we can calculate all the other values.

	male	female	uncertain	total
cremation	113	78		201
inhumation				197
total	196	179	23	398

	male	female	uncertain	total
cremation	113	78	10	201
inhumation	83	101	13	197

	male	female	uncertain	total
total	196	179	23	398

As expected from our formula, we have a degree of freedom of two.

df=2: if two values are chosen all other can be calculated with the help of the margins

(number of columns – 1)*(number of rows – 1)

Using again the formula, I'll leave the last example for you to figure out yourself.

	male	female	uncertain	total
cremation				201
inhumation				197
other				30
total	201	187	40	428

5.6.3 χ^2 Test for one sample (example after Shennan)

For demonstration using example from the Stephen Shennans book on quantifying archaeology: the number of Neolithic settlements on different soil types in eastern France

Soil type	Number of settlements
Rendzina	26
Alluvial	9
Brown earth	18
total	53

The question that should be answered here is whether or not there is a certain preference for specific soil types. From the data themselves it seems quite obvious, that Rendzina is very much preferred for settlements. So we will not need to statistical methods to check out, which of the sort of type is this preferred, but rather, if this high number of settlements could also result from just random effects. Or to be more concrete, how likely such a random configuration might be, and if it is likely to falsely exclude the possibility of random effects with more than our standard 5% probability for statistical significance.

This is actually a one sample test, because we have only one sample: the amount of settlements on the different soil types. It is probably easier to understand, that this is one variable, if you imagine it is a list of settlements, where we have the values "Rendzina", "Rendzina", "Brown earth", "Alluvial", and so on.

With this it becomes clear that we have a variable that is nominal scaled.

We will calculate two versions of this test here: at first a naive version, in which we assume, that the soil types are evenly distributed over the landscape, with the same area ratio. In the second version, we will add information about the actual distribution of the soil types in the landscape to our equation, and with that we will alter the values that we will expect for its distribution.

5.6.3.1 Version 1: even distributed

In this version, we assume that every soil type has the same proportion area in the landscape. If we now randomly distribute our settlements over the landscape, we would expect that each soil type has the same same probability of settlement. Consequently, we would expect that the number of settlements for each side type is the same. In total, we have 53 settlements. If we would evenly distribute these settlements over our three soil types, we would end up with the expection of 17.6667 settlements per soil type.

Soil type	Number of settlements	Proportion of soil type	expected number of settlements
Rendzina	26	1/3	17.6667
Alluvial	9	1/3	17.6667
Brown earth	18	1/3	17.6667
total	53	1	53

Now we can formulate a hypothesis: We are interested, if the settlements are not evenly distributed. Therefore, we need to disprove or make unlikely an even distribution. Soour hypothesis look like that:

H_0: The settlements are evenly distributed on all soil types.

H_1: The settlements are **not** evenly distributed on all soil types.

Of course, from the data we already see, that they are not distributed like our expectation. But of course, there is random chance that a certain soil type gets a higher number of settlements then another. This would also be true if we would use a dice to distribute the number of settlements. We are only interested in the situation, when this rolling a dice scenario is very unlikely, and a pattern behind this distribution cannot be ignored. For that, we have to measure how far the actual data are from our expectation (you could call that measure of surprise), and how likely such a distance from our expectation would be given the possible random nature of the distribution. For the later part, we will use pre-calculated tables, or the statistical program R. But the first part we can calculate ourselves.

Please don't be afraid: here come to formula! But we will gently introduce and

explain this formula to you.

Formula for χ^2:

$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$

O_i: number of **observed** cases

E_i: number of **expected** cases

χ^2: symbol for the test statistic chi-squared

The number of observed cases the present our actual data. This is the data that we observed. Consequently, the number of expected cases is represented by our column "expected number of settlements". In this formula, from our exact cases our expectations are substracted to calculate the difference. Because we don't care if I were observed cases are bigger or smaller than our expectations, because we are only interested in the difference, we square them to get rid of the sign. Then we have to take care, get our measure of surprise is independent from the number of cases. This is because the difference of 10 will probably surprise us, if the total number of cases is 20, but in the situation, where the total number is 5000, this difference of 10 probably is totally irrelevant. That is why we divide by the number of expected cases again, to normalise for the expected magnitude of our data. That must be repeated for every cell of the table. Once this is done, we can sum up all over individual measures of surprise to get the total surprise for the table, or more formally does Chi-Squared value for the table.

Soil type	Number of observed cases	Number of expected cases	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i - E_i)^2}{E_i}$
Rendzina	26	17.6667	8.3333	69,4444	3.9308
Alluvial	9	17.6667	-8.6667	75,1117	4.2516
Brown earth	18	17.6667	0.3333	0.1111	0.0063
total	53	53			**8.18868**

In this specific case, and starting from our expectations, we are rather surprised about the value of Rendzina, Because it is much higher than our expectations and even more surprised about the value of Alluvial, but this time, because it's much lower than our expectation. We seem not to be surprised at all by the value of brown earth, because the difference to our expectation is below one. The total measure of surprise (or Chi-Square value) is 8.18868. Currently, we do not know how to interpret that, we have no framework for it. What approach could be now to calculate several random settings and see, how because the price can be given random distribution. This easily can be done today using a computer. Before a computer was available to everyone, this was very hard, and so a lot of

precalculated tables were available. One of those food precalculated tables can be found in the book of Stephen Shennan. These tables are structured according to the level of significance (in our case 0.05) and the degrees of freedom (in our case 2, because we have one column for the observed and one column for the expected cases). With degree of freedom and level of significance we will find the threshold value. If our calculated threshold value is above this threshold, we have a significant result. This means, that there is a chance below 5% to go wrong to reject the null hypothesis, based on our data. In our case, the threshold is 5.99145. The value we have calculated is much higher, therefore we have a significant result. There's quite enough evidence to assume, that there is a preference for specific soil type visible in this settlement behaviour.

5.6.3.2 Version 2: even distributed with consideration of the proportion of the soil types on the total area

In the second version, we know how much area the difference or types take up in the landscape. This does not alter our data, but our expectations. If we know, how much percentage of the landscape is constituted off the individual soils, we also would exceed proportional amount of settlements to be situated on these soils. Now in this example, the distribution it's not so far away from the even distribution, but slightly.

Soil type	Number of settlements	Proportion of soil type	expected number of settlements
Rendzina	26	32%	16.69
Alluvial	9	25%	13.25
Brown earth	18	34%	22.79
total	53	1	53

For example, Rendzina makes 32% of the size of the landscape, resulting in an expectation of 16.69 settlements on this soil (multiplying the total number of 53 settlements by this percentage). From here on the onward, everything is the same like in the example before. We have to compare our expectation with our data, measure our surprise, and compare that to the possible surprise from random effects.

$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$

Soil type	Number of observed cases	Number of expected cases	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i - E_i)^2}{E_i}$
Rendzina	26	16.69	9.04	81.7216	4.8185
Alluvial	9	13.25	-4.25	18.0625	1.1363

Soil type	Number of observed cases	Number of expected cases	$O_i - E_i$	$(O_i - E_i)^2$	$\frac{(O_i-E_i)^2}{E_i}$
Brown earth	18	22.79	-4.79	22.9441	1.007
total	53	53			**7.1885**

The resulting number is slightly lower, but still beyond the threshold of 5.99145. So even considering the different proportions of the soils in the landscape, the result is still significant, and we have to interpret the preference pattern for specific soils

5.6.3.3 χ^2 test in R

Let's recreate the example in our. At first, we have to create a vector with our data.

```
settlements <- c(26,9,18)
names(settlements) <- c("Rendzina","Alluvial","brown earth")
settlements
```

```
##    Rendzina    Alluvial brown earth
##          26           9          18
```

If we don't have any specific expectations about the distribution, we directly can input the vector into the commands. It will default to an even distribution, and will compare it to this.

```
chisq.test(settlements)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  settlements
## X-squared = 8.19, df = 2, p-value = 0.017
```

Given our knowledge and our own endeavours, we can easily interpret the output of the command now. The `X-squared` is that Chi-Squared value that we also calculated. Because it's a one sample test, R also calculate the degrees of freedom correctly. The last return value is the p-value. Instead of just giving us significant or not significant, R calculates the error probability for rejecting

the null hypothesis. Since this p-value is below 0.05, we can reject the null hypothesis and collet a significant result.

We can introduce our expected proportions using the parameter `p`. Here, we give the percentages of the sort of types on the landscape as fractions, so that they sum up to 1

```
chisq.test(settlements,p=c(0.32,0.25,0.43))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  settlements
## X-squared = 7.19, df = 2, p-value = 0.027
```

You can read the result in the same way like the one above. You will realise that the Chi-Squared value is lower, and that the p-value is higher, meaning there is a higher chance of making an error rejected the null hypothesis.

Now that you have seen, how are you can calculate to one sample version by hand, and using R, we will turn our attention to the two sample version of the Chi squared test.

5.6.4 Two sample case (Test for independence)

In the examples before, we have compared one sample to a theoretical distribution, also enriched with some external knowledge. There, we have analysed how well our sample fits to some assumptions about the population. In the two sample case, we compare if the distribution of the values is independent between two populations. The data from this task may also come from the same data collection. Using a grouping variable, we divide this sample into two possible independent populations and test, if this independence is actually true.

As an example, I will use the distribution of amber in the sites of the Unetice culture. Here, we compare graves and settlements in relation to the presence of amber. We would like to find out, if the deposition conditions, and the causing traditions and behaviour is so different that we have to assume that different Processes must have been involved at the different site categories.

For this, we construct 2 x 2 table. Quite often, you will encounter situations, where this 2 x 2 set up might be applicable to your data. For example, every time you may like to find out if the presence of what artefact is connected to the presence of another artefact, this restructuring of the data might be useful.

(example after Hinz, beautified)

Type of site	amber		total
	+	-	
settlement	6	18	24
grave	132	44	176
total	138	62	200

So, our scientific question is if amber is primary a grave good in this society, or if it is distributed evenly across the different site categories. Like in most other situations, also here we aim for a certainty (level of significance) of 0.05. So, we will reject or not hypothesis only if we are 95% sure that it is wrong. Having 2 x 2 table, we have two rolls and two columns, which results in a degree of freedom of 1 (remember the formula to calculate degrees of freedom from above).

Our first task is to establish our expectation. We have to calculate the expected number of occurrence in the different side categories, given the assumption, that the distribution is independent from the site categories. In such a case, the number of sites with amber in settlements and in burials would only depend on The total number of sites with ember, and the ratio between sides of the type of settlement and sides of the type grave. Conversely, this is also true for the number of sites without amber.

So in total, we have 24 sites of the category settlement. If it would not matter, if it is a settlement or a burial, then the ratio between settlements with and without amber would be the same as the total ratio of sites with and without amber. The total ratio of amber present versus amber not present is 138 to 62. So we have more sites with amber in our sample then without. Of over 200 sites in total, 24 or settlements. That means, that 24÷200, or 12% of the sites are settlements. Also, 138÷200, or 69% of the sites have amber. That means, that of our 12% settlement sites (that is 24), 69% (that is 16.56) should be settlement sites with amber, given that amber is distributed independently from site categories.

This means, that we can use the margins of our table to calculate the expectation values. Using Cross-multiplication, we can multiply the margins and divide the result by the total number to get the number of sites that we would expect given independence. We need to repeat this procedure for every cell in our table, to get the expectation values for every individual possibility.

Type of site	amber		total
	+	-	
settlement	24/200*138 = 16.56	24/200*62=7.44	24
grave	138/200*176=121.44	62/200*176=54,56	176
total	138	62	200

From here onwards, the procedure is essentially the same like in the one sample case. For every cell, Using the Chi-Square formula, we calculate our measure of surprise, substracting the expectation from the observation, square result and divide by expectatio. Then we have to sum this up for the total table.

Type of site	amber		total
	+	-	
settlement	O=6 vs. E=16.56	O=18 vs. E=7.44	24
grave	O=132 vs. E=121.44	O=44 vs. E=54.56	176
total	138	62	200

$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$

Type of site	amber		total
	+	-	
settlement	(6-16.56)^2/16.56=6.73	(18-7.44)^2/7.44=14.99	24
grave	(132-121.44)^2/121.44=0.92	(44-54.56)^2/54.56=2.04	176
total	138	62	200

The total χ^2 is 24.68. Looking up in the table, we will find for a significant level of 0.05 and a degree of freedom of Df=1 the threshold of 3.84146. This means, we are much more surprised by the distribution of our data then our threshold value for possible random processes indicates. Therefore, the result is significant, and we can reject the null hypothesis and state, given our sample, that amber seem to be primarily a grave goods in the Unetice culture.

5.6.4.1 χ^2 test for indipendence in R

To calculate the same table in R, at first we have to enter all our data. Since now we have a 2 x 2 table, we have to produce a matrix representing all data. The calculation of course only needs the numbers, but for convenience we name our matrix with row and column names. Please note that we are using the `matrix()` command. In this command, as first parameter we give our values at first for the first column, then for the second column and so on in a vector. Then, we specify that we want a matrix with two columns. With that, our values are distributed between these two columns, and the result is a table representing the data.

```
amber<-matrix(c(6,132,18,44),ncol=2)
colnames(amber)<-c("with amber","without amber")
rownames(amber)<-c("settlement","grave")
amber
```

```
##            with amber without amber
## settlement          6             18
## grave             132             44
```

Now we can run our Chi-Squared test light before, entering the variable into the command for the test itself.

```
chisq.test(amber)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  amber
## X-squared = 22.4, df = 1, p-value = 2.2e-06
```

If you have a closer look to the result, you will be able to see that R is given us the correct degree of freedom, and the P-value indicating a significant result, because it is below 0.05. But you also will see that the total Chi-Square value is only 22.402, and not 24.684, as we have calculated 'by hand'! Did we, or R, made a mistake here?

No, actually not. On an even closer inspection, you will see, that the headline of the test result states that it is 'Pearson's Chi-squared test with Yates' continuity correction'. By default, our is introducing a correction for small sample sizes, that is known as 'Yates' continuity correction'. Instead of directly squaring the observed minus the expectation values, it's abstracts 0.5 from the absolute difference of both. The formula for this is $(|O - E| - 0,5)^2/E$. Substracting the small number does make a difference in case of a small sample size, because in this situation also the results from substracting the expectation from the observed will result in a small number. But if the result is large, because the sample size is also a large, 0.5 will not change so much in the end. Of course, we can also calculate the raw Chi-Square result. In this case we have to specify that we do not want the correction to take place.

```
chisq.test(amber,correct=F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  amber
## X-squared = 24.7, df = 1, p-value = 6.8e-07
```

Now the result is exactly like we have calculated it on our own. In case of small sample sizes, the correction already helps a bit to make the result robust. But

in such a situation, especially when the number of cases is very small, it might be a good idea to use the Fishers Exact test instead of the Chi-Squared test. If you're interested, you may find a description of this test procedure here, and its implementation R here.

5.6.4.2 χ^2 excercise

You can try out your skills in Chi-Square testing yourself using the following exercise:

Exercise 5.1. Given all the animal bones from the Neolithic side of welcome Wolkenwehe (Mischka et al. 2005). We have here to Neolithic layers, one from the middle and one from the late Neolithic. The minimum number of individuals of different animals are divided into wild and domestic animals. Please test, if the ratio between wild and domestic animals is independent from the layer, and with that independent from the period of use of the site.

layer	Domestic animal	Wild animal
202 (late neolithic)	159	32
203 (middle neolithic)	84	54

Solution

At first, we have to construct our matrix for the data we have.

```
ww<-matrix(c(159,84,32,54),ncol=2)
colnames(ww)<-c("domestic","wild")
rownames(ww)<-c("late_neo","mid_neo")
ww
```

```
##          domestic wild
## late_neo      159   32
## mid_neo        84   54
```

Then, we performed a Chi-Squared test.

```
chisq.test(ww)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  ww
## X-squared = 19.6, df = 1, p-value = 9.4e-06
```

```
chisq.test(ww, correct = F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  ww
## X-squared = 20.8, df = 1, p-value = 5.2e-06
```

You can see, that now, since we have more cases, the yates correction does not change the results in the same intensity like another example. But in both situations, we have a significant result: the differences between the different layers are statistical significant.

5.7 Kolmogorov–Smirnov test

Our second nonparametric test is the Kolmogorov–Smirnov test, or short KS test. This test can be used in situations, where are you having ordinal scales or better variable. Such a variable has more informational value, and therefore the test can also perform with a higher power. That means, that this test is more capable of avoiding type 2 errors. The downside is that it is only applicable if we have this kind of data present. Also, Kolmogorov–Smirnov test can be used as an goodness of fit test in the one sample case, and as a test for independence in the two sample case.

5.7.1 Facts sheet KS-Test

requirements: at least one ordinal scaled Variable (one sample case) and 1 nominal scaled grouping variable (two sample case)

Procedure one sample case: the cumulative procentual frequency of the sample is compared with a standard distribution (often normal distribution)

Procedure two sample case: the cumulative procentual frequencies of the samples is compared

5.7.1.1 What do these facts mean

We test against a at least ordinal scaled variable. Please note, that the KS-Test is also very useful in situations where you have a metric data. When we have to distinguish between two samples, we need again a at least nominal scaled grouping variable.

In the test procedure, we order our values according to the ordinal scale variable, and then we calculate the procentual frequency for every cell in our table. Finally, we sum these frequencies up starting from the top to the bottom. This gives us the cumulative procentual frequency. Now we calculate again the difference to our expectation, either against a theoretical distribution, or against the other sample. Then we are looking forward to the maximum difference, and compare this to a threshold, that this time we calculate ourselves given the number of cases that we have in the different samples. If we are more surprised by the difference then a random situation could produce within 95% of the cases, we assume statistical significance. Please follow example to understand precisely what this means.

5.7.2 Example (after Shennan)

Here, again we are using an example from the Stephen Shennan book. We are analysing female Bronze Age burials from a graveyard. The individuals buried our listed according to the different age classes. This age classes represent our ordinal variable.

	rich	poor
infans I	6	23
infans II	8	21
juvenilus	11	25
adultus	29	36
maturus	19	27
senilis	3	4
Sum	76	136

Looking at the data, you will realise that there are more individuals in the poorer category that died in younger ages. But also in total, there are more individuals in the category poor. You can have the hypothesis, get people that were equipped more rich when buried also represents richer individuals during lifetime. And you can also have the assumption, that rich individuals had more access to resources, and probably also had the chance to get older. So here, the question might arise, it needs to be answered in a statistical way, if there is a significant difference between the mortality pattern in both categories.

From this data and our assumption, we build up our test configuration. Our hypothesis is that there is the difference between the two mortality patterns. Therefore, another hypothesis is that there is no difference. So we assume, that the distribution of age at death is independent from the number of burial items.

H_0: There is no difference between rich and poor graves according to age of death.

H_1: There is a difference between rich and poor graves according to age of death.

Since the difference can be that poor died earlier, or later, we have a two-tailed test here. As always, our level of significance is 0.05. We use the age classes as ordinal scaled test variable, and the wealth classes as at least nominal scaled grouping variable. Wealth proceed can be understood as an ordinal scale. But we can always scale down the variable, so we treated just as if it is an nominal scaled one. In the table, the levels are already sorted according to our ordinal scaled variable. If not, of course you can easily do it yourself.

Level of significance: 0.05

variables:

1. ordinal scaled age classes
2. (at least) nominale (ordinale) scaled wealth classes

At first, we will calculate the procentual frequency. That is the ratio of the individual classes in respect to the total number. Simply, we divide every cell by the total sum. If you would multiply that by 100, you would get the percentage.

	rich	rich_ratio	poor	poor_ratio
infans I	6	0.07895	23	0.16912
infans II	8	0.10526	21	0.15441
juvenilus	11	0.14474	25	0.18382
adultus	29	0.38158	36	0.26471
maturus	19	0.25000	27	0.19853
senilis	3	0.03947	4	0.02941
Sum	76	1.00000	136	1.00000

You can see, that while the actual numbers sum up to the total number for each category, the ratios some up to 1. Is the next step, we have to calculate the cumulative frequency. That means, but with sum up the values from top to bottom for each row, including all the values of the categories that are smaller than the actual category. So in the first row, infans I, the cumulative frequency is actually the same value as the original frequency. In the second row, infans II, the cumulative frequency is the value of infans I plus the value of infans II. And so on.

	rich	rich_ratio	rich_cumsum	poor	poor_ratio	poor_cumsum
infans I	6	0.07895	0.07895	23	0.16912	0.16912
infans II	8	0.10526	0.18421	21	0.15441	0.32353
juvenilus	11	0.14474	0.32895	25	0.18382	0.50735
adultus	29	0.38158	0.71053	36	0.26471	0.77206
maturus	19	0.25000	0.96053	27	0.19853	0.97059
senilis	3	0.03947	1.00000	4	0.02941	1.00000
Sum	76	1.00000	3.26316	136	1.00000	3.74265

Now you can see, that while the sum of the procentual frequency is 1, but after cumulative frequency is the number different from 1. At the same time the last

real value in this column is 1 ('senilis'). And the other values are constantly rising going down our age classes. That means, for example in the role of 'adultus' the valley represents all the individuals that died in that age class or younger.

```
## Warning in latex_new_row_builder(target_row, table_info, bold, italic, monospace, : Setting fu
## colors are not really easily configable with this package. Please consider turn off full_width
```

	rich_cumsum	poor_cumsum	difference
infans I	0.07895	0.16912	0.09017
infans II	0.18421	0.32353	0.13932
juvenilus	0.32895	0.50735	0.17841
adultus	0.71053	0.77206	0.06153
maturus	0.96053	0.97059	0.01006
senilis	1.00000	1.00000	0.00000

Now we have to identify the situation, we are both distributions different the most. For this, we have to substract one cumulative frequency from the other, and make the values absolute removing to sign. Having done this, we can identify those at the age class 'juvenilus' the difference is the highest with a value of 0.178. This is our test statistic that we will compare to a calculated threshold.

By comparing the percentages, our degree of surprise within our data is already independent of the number of cases. Thus, we now need to find a threshold value that indicates how large the deviation could be within a sample that is only dependent on the random distribution. At the same time, however, this surprise also depends on the number of cases: if we have fewer cases, larger differences can also arise purely by chance. The law of large numbers applies here, which we will get to know in more detail later: The more cases we consider, the more purely random effects will cancel each other out. Therefore, we need more surprise for a smaller number of cases in order to be convinced that this could not have been caused by a random distribution. At the same time, the required degree of surprise also depends on the significance level that we want to achieve: the more certain we want to be, i.e. the greater our significance level must be, the greater the surprise must be in order to be considered significant.

In the following formula for calculating threshold, both aspects are considered.

$$threshold = f * \sqrt{\frac{n_1+n_2}{n_1*n_2}}$$

In the formula, you can see n_1 and n_2. These numbers represent the number of cases in the first and the second sample. In the numerator of the fraction, you can see that both sample sizes are added, while in the denominator they are multiplied. This means, that the resulting number will be higher, when the total sample size is smaller, and vice versa. The other factor is f, which is the constant that depends on the level of significance that we would like to achieve.

Factor f:

- Level of significance 0.05: 1.36
- Level of significance 0.01: 1.63
- Level of significance 0.001: 1.95

So all we have to do is to fill in the numbers from our samples and the value for the level of significance that we would like to achieve. In our case, these are the following numbers:

Total number rich: $76 = n_1$

Total number poor: $136 = n_2$

So the calculation for a threshold is unfolding like this:

$threshold = 1.36 * \sqrt{\frac{76+136}{76*136}} = 0.19477$

No, we can compare that to the value that we have calculated from all sample:

Difference max (D_max): 0.178

You can see, that $0.19477 > 0.178$. This means, that the difference can not be considered to be significant.

But this does not mean, that the distribution is actually equal! Since we have made is dependent on the sample size, it is clear that there are two reasons why a significant result could not have been reached: Either, the resulting difference is really not big enough, so that it could result from a random process. Or I will sample size is just too small, so that we can really differentiate between random and patterned processes.

Here, with archaeological thinking, we can already spot a possible error in our assumption: quite big differences exist in the lowest category, that is 'infans I'. From our science we know, that burials of children are often not be equipped with very many of your items, independent from the circumstances in life. So here is a possible source of bias, that our calculation cannot reveal, but that we have to identify using our domain knowledge. But also in this situation, statistical tests can help to identify these divergence from our expectation, and make us think about possible reasons for this.

5.7.3 KS-Test in R

Let's do the same test using R. You probably like to download the data for your own experimentation.

- graeberbrz.csv

First, let's look at the data and inspect them:

```
graeberbrz <- read.csv2("graeberbrz.csv",
                        row.names = 1)
head(graeberbrz)
```

```
##   alter reichtum
## 1     1    reich
## 2     1    reich
## 3     1    reich
## 4     1    reich
## 5     1    reich
## 6     1    reich
```

You can see, that in our dataset the age classes are coded using numbers starting from one for 'infans I' to 6 for 'senilus'. Quite often it makes sense to recode ordinal variables in such a numerical way to make it accessible for calculations in R. With this recording, we achieved that the ordinal character preserved. We can use the table comand to display the full dataset with the frequency of burials in the different classes:

```
table(graeberbrz)
```

```
##       reichtum
## alter arm reich
##     1   6    23
##     2   8    21
##     3  11    25
##     4  29    36
##     5  19    27
##     6   3     4
```

For the `ks.test()`, which is the R command for conducting this test, it is convenient to take the data set apart into two vectors. The first vector will include the list of ages, the second vector will store the association with a specific wealth class.

```
age <- graeberbrz$alter
head(age)
```

```
## [1] 1 1 1 1 1 1
```

```
wealth <- graeberbrz$reichtum
head(wealth)
```

```
## [1] "reich" "reich" "reich" "reich" "reich" "reich"
```

Using the wealth vector as access criterion, we will feed into the `ks.test()` command age classes of the rich and the poor graves separatly as two samples

```
ks.test(age[wealth=="arm"],
        age[wealth=="reich"]
        )
```

```
## Warning in ks.test(age[wealth == "arm"], age[wealth == "reich"]): p-value will be a
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  age[wealth == "arm"] and age[wealth == "reich"]
## D = 0.178, p-value = 0.09
## alternative hypothesis: two-sided
```

You will see, that in the output the value D represents the maximum difference between the two samples that we also have calculated by hand. Additionally, you will see the P value that here is 0.09. So given the classical 0.05 significance level, we just have not reached enough certainty that we can speak about a patterned distribution with 95% security.

You also will see a warning, that there are ties. This means, that there are equal values in both samples present. In that case, exact P values cannot be calculated. This is only possible, if we have a metric variable instead of an ordinal ones, or a very large sample size (beyond 10000). For us, the approximated p-value is more than enough.

Lastly, you will see that the alternative hypothesis is two sided. This means, that by default the test assumes a two-tailed hypothesis. So in total, the test using R confirms our own calculation of the result: it is not significant.

5.7.3.1 KS-test excercise

You can try out what you have learned using this exercise:

Exercise 5.2. This dataset represents cups from 'relative closed' finds from late neolithic inventories (Müller 2001).

File: mueller2001.csv

If you inspect the dataset, you will see that it contains the height of cups and a classification whether there profile is subdivided by a break or not. Please analyse with the Kolmogorov-Smirnov-Test if the heigths of cups with and without breaks differ significant on a 0.05-level.

Solution

At first, we have to load our data and construct our two sample vectors. This time, we divide the data set in a slightly different way, to give you an alternative. But it would also work in the same manner like we divided the example of the burials. Note also, that this situation we have metric data, and not ordinal ones.

```
cups <- read.csv2("mueller2001.csv")

cups.with_breaks <- cups$hoehe[cups$tassentyp == "zweigliedrig"]
cups.without_breaks <- cups$hoehe[cups$tassentyp == "eingliedrig"]
```

Then, we performed a KS test.

```
ks.test(cups.with_breaks, cups.without_breaks)
```

```
## Warning in ks.test(cups.with_breaks, cups.without_breaks): cannot compute exact p-value with t
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  cups.with_breaks and cups.without_breaks
## D = 0.252, p-value = 0.1
## alternative hypothesis: two-sided
```

Also here, there is no significant difference between the two types of cups in respect to the height.

5.8 Interpretation of significance tests

5.8.1 Pay attention also when the statistic seem to be clear

As with every version of statistical tests, the quality of the results strongly depends on:

- Having the right data

- Asking the right question

So even a significant result probably doesn't mean anything, if you have asked the wrong question. Here, science and domain knowledge is much more important then statistical knowledge!

After the test as well as before the test: The interpretation determines the result!

Also, what do you do with the significant results strongly depends on your interpretation. For example, if you analyse a dataset, you probably will find a lot of associations, especially if it is more complex. But not every dependence, not every association it's meaningful! Remember that by definition you will get a significant result in every 20th test even if all underlying data are just random. This means:

Statistically significant archaeologically significant!

You have to make sure, that the results that you can obtain really something that could also have relevance in the life of prehistoric people to become relevant.

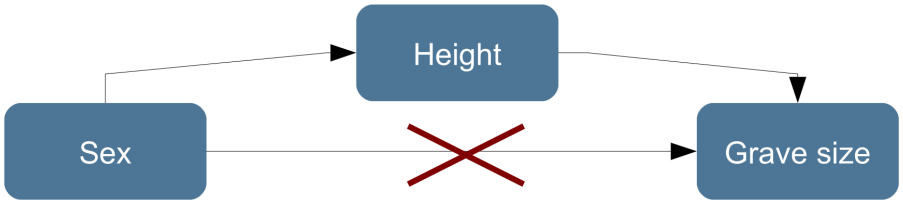
All in all:

Statistical results stay statistical: significance is always probability that the choice of a hypothesis is correct, but there is also a probability that it is by chance...

5.8.2 Statistical association not mean causal association!

The classic saying in statistics is that association does not mean causation. Even if there is really a pattern in the data, this could result from observed variables, all variables that you did not take into account when constructing your hypothesis.

One example according to the Stephen Shennan book is the association between grave size and sex of the buried individuals Although there might be a statistically significant association between grave size and sex, this could be caused by a third factor. Maybe mail individuals have just bigger body height, making it necessary to dig bigger burial pits so that they can fit in. A conclusion which says that grave size are causally determined by sex would be at least not correct.



Always, doing statistical analyses does not save you from making your own thoughs and trying to contact good science!

Chapter 6

Basic Probability Theory

6.1 Repetition

We talked about it before, but to be sure, that we are on the same page, here are some repetitions of some of the concepts we have already introduced. We will meet them in the progress of this chapter.

6.1.1 Population and sample

We already learnt the differentiation between the population and the sample. The **population** is the amount of all things that might be relevant for our analyses, or in other words, those objects in the real world about which we would like to make a statement with our statistical analyses. Quite often, the population cannot be analysed directly. From a practical point of you, this might be, because such an analyses would be too expensive. In archaeology, we do not have access to the full population of e.g. objects, because most of them have disintegrated over time.

In such a situation, instead of investigating the full population, we take a **sample** of the population. Under ideal circumstances, we select our sample according to some controlled criteria, most often represent activity (in archaeology, quite often we take everything that we can get access to).

The difference between sample and population is important. Realistically, we can only make statements about the sample. So we never know, if in the population there might be one or more individuals, they are totally different from those that we have sampled. But given some credible assumptions about the selection process of our sample, we can guess (or better measure) the probability that our sample represents the population well. In the end of the chapter you will know, how do use can be done.

To make quantifiable statements about our population (or our sample), most of the time we will talk about **parameter**. These parameters are numerical representation of features of objects, or collections of objects. In case of the sample, we can measure these parameters directly. In case of the population, we cannot do this. Nevertheless, also the population has a fixed value for each individual parameter that we could measure on our sample. Here, we have to believe that there is a reality independent from us (or any other observer). Sold the parameters of the population represents real, fixed numbers, although we do not know them, and even if it is impossible to know them at all.

In many statistical tests, we measure a parameter of sample, and then estimate our uncertainty, if the value measured would fall within a certain range of possible parameter values of the population. So essentially, we are building a bridge between our sample and the population. Our building material consists of probability theory. But from what has been said, it is clear, that the quality of our statement about the population (and its parameters) show me depends on the selection of the sample (and it's representativity).

6.1.2 Null hypothesis

We also learnt the concept of the **null hypothesis**. Here, instead of proving our statement about the population, we are testing with the sample how likely it is, That's the contrary of our original assumption (this contrary is the **alternative hypothesis**) is wrong. For example, for a specific parameter in the sample (e.g. the mean), we estimate how likely it is, that the sample with the specific parameter (the meaning of the sample) would come from the population with totally different mean value. Or, in the case of the Chi-Square test, how likely it is that the distribution of our values within the sample could come from a population, where the categories are independent from each other, just by random chance.

If you are still uncertain, if you fully understand this concept, I would suggest that you have another look at the previous chapter.

6.2 The concept of probability

The concept of probability has to be differentiated between the "common sense" version and the scientific concept. In everyday life, we constantly use probabilities to express our assumptions about the world. If it rains, before you started reading this chapter, probably it is still raining. And you are very likely aware, did you learning result will be probably not as good as it could be, if you don't do your homework. You will have a feeling about the likelihood of these probabilities, but this is subjective, and you will probably not be able to express that in numbers.

But most of the time, these probabilities are not very specific. To make probabilities useful in a scientific environment, we have to be able to quantify them. For this reason, the statistical concept of probability is based on mathematical probability laws. These laws come from thought experiments or analyses of random trials. Such random trials depends on the idea, that they are (in principle) eternally repeatable operations, and that their result is (in principle) not predictable in the individual case. In that sense, you could say, that the mathematical principles of statistics resemble gambling. And a lot of the basic theory was indeed developed in the context of gambling. It might be very beneficial, if you have the ability to estimate how likely you will win a game of cards choosing one strategy or the other.

Of course, if you roll a dice, or you draw a card from a mixed set of cards, the result is already determined: The laws of physics or the order of the cards in the deck determine, what number of the dice will show, or which card you will draw. But the actual result depends on so many independent influences, that it is nearly impossible (and with that impossible enough) to predict the individual case.

The same is true for parameters of a population of objects, or the values within the sample: of course, these values are fixed and existent in the real world independent from the observer. But since the actual manifestation depends on so many influences and effects, that for us the values seem to be random. That is why we can use the same logic to estimate the best gambling strategy and to estimate the mean rim diameter of Neolithic pottery from a specific site. In both situations, you try to predict an unknown value or quantity using known values within a certain framework of parameters, and try to estimate your certainty (or probability of error). Kolmogorov, we already know from the Kolmogorov-Smirnov test, has developed a definition which is the foundation of today's probability theory. Before we come to that, at first we have to understand some principles of set theory (Mengenlehre), related to probability theory, and some notations from that. For this, we use one of the classical examples: rolling the dice multiple times.

6.2.1 Some Notations and Definitions

If you roll a dice (for pen and paper role players, we are talking about a W6), the result of an individual might be for example 5. This is called an *elementary event* (or also *atomic event*, although no nuclear fission is involved, or *sample point*). The name refers both to the process and its result. In german it would be called "Ereignis". This is the result of one actual trial, it is the realised result out of a range of possible results.

The possible results in the case of the dice (again, dear old players, we are still talking about the W6) are one, two, three, four, five, six. These possible results are called *event space*, or *sample space*. Quite often, you find the greek letter

Omega (Ω) as the symbol representing this sample space. In our case, you could also write that as follows:

$\Omega = \{1, 2, 3, 4, 5, 6\}$

The notation above describes, the Omega consists of the set ("Menge") of the numbers from 1 to 6. Also, if we consider the results of different dice rolls in one go, they also can be understood as a set of values. Here, we are free to assign a symbol of our choice. E.g. like this:

$A = 2, 4, 1, 3, 5$

There are some more symbols in set theory and the respective notation that might be of relevance here. In the table, I'll give you some examples, and I hope, with this practical representation you will be able to understand, what is going on here:

Set A={1,2,3,4}; Set B={4,5,6};	
Event space Ω={1,2,3,4,5,6}	
1 is an element of set A	$1 \in A$
C is the union of A and B {1,2,3,4,4,5,6}	$C = A \cup B$
D is the intersection of A and B {4}	$D = A \cap B$
E is A minus B {1,2,3}	$E = A - B$
Not A (=event space - A)	$\bar{A} = \Omega - A = 5, 6$
The intersection of D {4} and E {1,2,3} is the empty set	$D \cap E = \emptyset$

The symbol $\in$ describes membership. The union $\cup$ combines all elements from both involved sets. The intersection $\cap$ takes only those objects, that are represented in both sets. If you substract one set from the other, you remove all the elements from the first set that are present in the second set. The specific definition of a set is the qualifier "Not", most of the time expressed as a bar above the symbol. This refers to everything else in the event space except for those objects that are in the negated set. The last important set is the empty set $\emptyset$, that consists of zero elements.

With this notation, now we have everything at hand so that we can talk very precise (science!) about possible results of the dice roll, add about all other possible random variables. We will use this knowledge now to calculate probabilities.

6.2.2 Classical Probability Definition by Laplace

The very basic and classical definition of probability comes from Pierre-Simon Laplace (1749–1827).It states that the probability for a specific set of events (positive events) is calculated by dividing the number of possible outcomes with these positive events by the total number of all possible events. Or expressed in formulas:

$p(A) = \frac{Number\ of\ positive\ results}{Number\ of\ possible\ results}$

The resulting number is also the relative frequency of this positive results. Let's explain that with our dice example.

To win the game, we have to throw a six. How likely is that?

$A = 6, Event\ space = \{1, 2, 3, 4, 5, 6\}$

The number of possible results in our event space is 6. With a dice there's only one way to roll the number 6, the positive result in our example.

$p(6) = \frac{1}{6} = 0.1667 = 16.67\%$

All other results do not represent our positive result. Given our knowledge about the notation from above, we can also express it like that:

$p(\bar{6}) = p(\Omega) - p(6) = 1 - \frac{1}{6} = \frac{5}{6} = 0.8333 = 83.33\%$

So there is a probability of 16.67% to win the game, and the probability of 83.33% to not win the game. With this, we have already used intuitively most of the Kolmogorov axioms. Let's dissect that a bit, and then come to the formal definition.

In the random experiment, but also in the real world, if we have a result, all the time something must have happened. So the probability, that anything happened, irrespectable of the specific result, it's always 100%. Or, a bit more formula, the probability of the event space is always 1.

$p(\Omega) = 1$

This also can be called the *safe event*: it is safe to assume, that this will happen. The safe event is the event with 100% probability.

$p(A) = 1$

$p(this\ is\ a\ statistics\ course) = 1$

If it is impossible, that the specific event might occur, then it has zero (0%) probability:

$p(A) = 0$

$p(here\ you\ can\ learn\ something\ about\ knitting) = 0$

And event itself, and its negation represents *complimentary events*:

Without physical tricks a dice roll always has a number as result. No matter, what it actually is, with normal dice its probability is 1/6

$p(6) = \frac{1}{6} \rightarrow p(A) = \frac{1}{6}$

Rolling not this specific result is the complimentary event of rolling exactly this result:

$p(1...5) = \frac{5}{6} \rightarrow p(\bar{A}) = \frac{5}{6}$

With a normal dice, you always will roll a number as a result. The likelihood of this number will be always 1/6, because there is only one way this result (out of six possible results) could be the outcome. Not getting a specific number as a result, in the case of the dies, as always the probability of 5/6. There are five ways to get a different result from this specific one. So the result and its complimentary result describe the whole event space, and its combined probability is therefore 1.

$p(A) + p(\bar{A}) = 1$

Since the probability of an event and its opposite is always 1, you can calculate one from the other.

Let's switch the example to test this out:

A card game has 4 colors (diamonds, hearts, spades, clubs).

The probability to draw a heart card is 1 out of 4: 0.25

The probability of not drawing a Heart card is 3 out of 4: 0.75, or 1 - (1 out of 4): 1-0.25 = 0.75

6.2.3 Kolmogorovs probability axioms

Now let's have a look to the formal representation of the Kolmogorov probability axioms:

6.2.3.1 1. axiom

Each event from the event space is assigned a number p(A), which describes the probability of the event. This is between 0 and 1.

$0 \leq p(A) \leq 1$

This means, for every possible event we express its probability with numbers ranging from 0 to 1.

6.2.3.2 2. axiom

The safe event has the value one.

$p(E) = 1$

This means, that if we know something must happen in this way, then its probability is 100% or 1.

6.2.3.3 3. axiom

For pairwise disjunctive events, i.e. those that do not have an intersection (e.g. {1,2} and {3,4}), the probability for their union is the sum of their individual probabilities.

$p(A_1 \cup A_2 ... \cup A_n) = \sum_{i=1}^{n} p(A_i)$

This means, that four sets to do not include the same members, the unified probability will be the sum of the individual probabilities. Not only in the case of the complimentary event, but in every situation where we talk about disjunctive events.

So, in the case of our dies, one probable set of outcomes A (one way to win) might be 1 & 2, the second B (another way to win) with 3 & 4.

e.g. $\Omega = \{1, 2, 3, 4, 5, 6\}, A = \{1, 2\}, B = \{3, 4\}$

Given that, the total probability of winning the game in one way or the other can be calculated as the sum of the individual probabilities of one way of winning and the other.

$p(A) = \frac{2}{6}, P(B) = \frac{2}{6}, p(A \cup B) = p(A) + p(B) = \frac{2}{6} + \frac{2}{6} = \frac{4}{6} = 66,67\%$

All of this seems to be quite complicated ways to express quite common sense concepts. But you will see, as soon as we leave this very basic ground, that is very defined expression makes sense. This is especially true, when it comes to conditional probability and composite, repeated events. Here, quite often the commonsense understanding of probability has its limits.

6.2.4 Conditional and independent events

In estimating total probabilities, it is very important to distinguish between situations, where the results are independent from each other, from those, where the result of the first changes the probabilities of the second result. For independent results, we can stick to our example with the dice. Rolling a dice once does not change the probabilities of individual results from the second dice roll. In case of individual independent results, the total probability of a set of results is calculated by multiplying the individual probabilities.

Therefore the probability is to roll first a 5 and then a 6 is calculated like so:

$p(A \cap B) = p(A) * p(B) = \frac{1}{6} * \frac{1}{6} = \frac{1}{36}$

You can also understand that in a way, that after we have drawn the 5 out of our bag of possible numbers in the first random trial, we have put it back into the bag of random numbers. But what if, when we have an event, this is not longer available for further trials?

Here is an experiment (after Dolić): A (non-transparent) bag with a chocolate biscuit, a sugar biscuit and an eco biscuit. How likely is it to get out the chocolate biscuit first and then the eco biscuit?

It is wrong to assume, that we can use the same strategy like above: $p(choco\ than\ eco) = p(choco) * p(eco) = \frac{1}{3} * \frac{1}{3} = \frac{1}{9}$

because after the chocolate biscuit's out, there's only two biscuits left $p(eco\ if\ choco) = \frac{1}{2}$

That's why: $p(choco\ than\ eco) = p(choco) * p(eco\ if\ choco) = \frac{1}{3} * \frac{1}{2} = \frac{1}{6}$

First, it is necessary to calculate how likely it is to get out the chocolate biscuit. Here, we are still on safe ground: This first probability acts like in the case of the dice. But for the second trial, we have to calculate the probability of getting the eco biscuit, if we have already taken out the chocolate biscuit. Since now our event space consists only of two possibilities, it is 1 out of 2. Having figured this out, now we can multiply the probabilities.

A more general definition of this is the **axiom of (conditional) probability**, as also defined by Kolmogorov:

$p(A \cap B) = p(A) * p(B|A)$

This means, that we can calculate the total probability of A and B by multiplying the probability of A and the probability of B given A. Or, if we like to reformulated:

$p(B|A) = \frac{p(A \cap B)}{p(A)}$

The (conditional) probability of B given A is the total probability of A and B, divided by the probability of A. You might find this definition in some textbooks.

6.2.5 Addition law of probability

This axiom of (conditional) probability helps us to make the third axiom more general. It gives us the tools to calculate probabilities in situations, where we don't have mutual exclusive events.

Let's introduce this using a set of cards. The set contains 32 cards of for colors [ace (Ass), king (König), queen (Dame), jack (Bube), 10, 9, 8 and 7 in all four suits (clubs , spades , hearts and diamonds)].

First let's calculate how likely it is to draw hearts card using our knowledge:

32 cards, 1/4 is hearts (8)

$p(A) = \frac{8}{32}$

$p(A) = \frac{1}{4}$

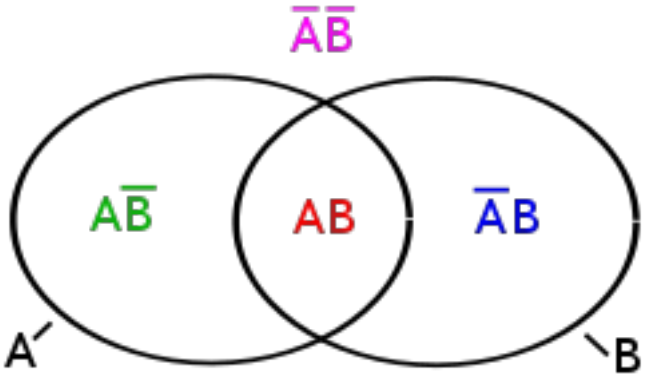
Or similarly, how likely it is to draw a queen:

32 cards, 4 queen

$p(A) = \frac{4}{32}$

$p(A) = \frac{1}{8}$

That's easy isn't it? But now let's assume, that Queen and hearts are trump cards. How likely is it to draw a trump card? We cannot simply multiply nor add up the probabilities, because they are not mutually exclusive.



Imagine come out that in this image A represents all the hearts, and B represents all the Queens. The Queen of hearts is present in both sets. When calculating the probability of this combination, we have to consider this fact.

So, drawing cards, how likely is it to draw a trump card (queens and hearts are trump)? 32 cards, 4 queen, 8 heart, one queen of hearts

First, we have to calculate the probability of drawing hearts, then we have to calculate the probability of drawing Queens. Since the Queen of hearts is present in both sets, we have to remove one of occurences or better, its probability.

$p(A) = \frac{1}{4}$; $p(B) = \frac{1}{8}$; $p(A \cup B) = p(A) + p(B) - p(A \cap B)$

$p(A \cup B) = \frac{1}{4} + \frac{1}{8} - \frac{1}{32} = \frac{11}{32} = 0.34375$

This, we have developed the general addition law of probabilities:

For all possible combination of events:

$p(A \cup B) = p(A) + p(B) - p(A \cap B)$

With both, the conditional probability and the addition law of probability, it becomes clear that to calculate probabilities we have to find a way to count possible outcomes. This is the field that is called combinatorics. Luckily, there are only a few possibilities. Actually, there are only two factors that can change the set up of such random experiments: If the order is relevant, or, if we have independent trials

6.3 Combinatorics

To calculate how many possible outcomes an experiment has is rather easy for the individual trial. But if the total probability consists of multiple events, then it is of importance holders events are combined.

In essence, it boils down to the question, how many possibilities are there to select k elements from n elements? The following table gives you an overview of the possibilities and also shows, how the number of possibilities are calculated in this situations:

	Variation (with respect to order)	Combination (without respecting the order)
with 'putting back'; with replacement	n^k	$\frac{(n+k-1)!}{k!*(n-1)!}$
without 'putting back'; without replacement	$\frac{n!}{(n-k)!}$	$\frac{n!}{k!*(n-k)!} = \binom{n}{k}$

Let's illustrate these possibilities with a few examples:

6.3.1 How many possibilities are there to combine 2 dice results?

$\Omega = \{1, 2, 3, 4, 5, 6\}; n(\Omega) = 6; number\ dices\ k = 2$

We have already figured out, that with the dice the probabilities do not change: We are dealing with a situation with replacement. Now we have to decide, if the order matters to us. This might be especially true, if we talk about a board game.

$B_n^{k=2} = \{(x_1, x_2) | x_i \in \Omega\}$

So, we have the situation with putting back and with respect to order (x_1, x_2). This situation, we calculate the number of possible outcomes by putting the number of possible events to the power of the number of trials.

$n(B) = n^k = n(x_1) * n(x_2) = n(\Omega) * n(\Omega) = 6 * 6 = 6^2 = 36$

Therefore, the Probability for a 6 in the first roll and a 5 in the second is: $p(x_1 = 6, x_2 = 5) = \frac{1}{6^2} = \frac{1}{36} = 0,0278 = 2,78\%$

6.3.2 How many possible unique Lotto tickets (6 of 42) are there?

$\Omega = \{1, 2, 3..., 42\}, n(\Omega) = 42, number\ draws\ k = 6$

We have to numbers from 1 to 42, so the size of our event space is 42. There are six numbers drawn, it does not matter in which order. But the numbers are not being put back, so we deal with no replacement.

$B_n^{k=6} = \{(x_1, x_2, ..., x_6)||x_i \in \Omega|(x_1, ...x_{i-1})\}$

This situation, we have to use the formula in the lower right cell of our table. You can see a lot of !'s in this formula. This exclamation mark means "factorial". In mathematics, the factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n:

$n! = 1 \cdot 2 \cdot 3 \cdots (n-2) \cdot (n-1) \cdot n$

So, in the case of factorial 6, this translates to:

$6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6$

The whole expression has a specific name: it is the binomial coefficient. To calculate the binomial for our situation, we just have to fill in our values.

$n(B) = \frac{n!}{k! * (n-k)!} = \frac{42!}{6! * (42-6)!} = 5245786$

It's a rather high number. It's a bit more than half of the population of Switzerland. That means, if everybody in Switzerland would play lotto with a different set of numbers, only 1 to 2 persons would get it right.

Therefore: Probability for 6 correct numbers:

$p(6_{right}) = \frac{1}{5245786} = 0 = 0.00002\%$

This now is a rather small number. It means, you as an individual have a rather low chance to win a jackpot. Probably it's better to continue learning statistics to get a job. Even an archaeology the chances are better to make a living from it.

6.4 Law of large numbers

No we have all the tools to calculate probabilities of different settings. It probably will not help us to win the jackpot. Even with the best statistical tools that, it is impossible to predict an individual outcome from a random experiment.

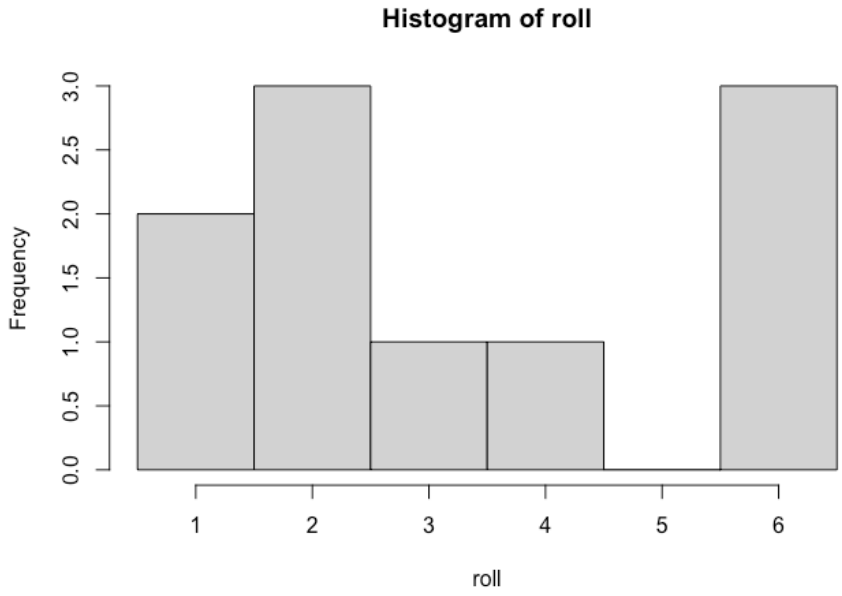
Hence it is a random experiment. Of course, with your first game of lotto you might win a jackpot against the odds.

Also, if we would like to predict the mean result of the dice roll, the actual result of the individual trial might be very different from our expectation. At least for the first one. Probably also, if we had to dices. But the more dices we have, the closer will be our theoretical mean of the dice rolls to the actual one. Here, we have our bridge from the sample to the population. In this case, the population is the mean of all dices ever thrown. Like with an archaeological population, this is theoretical value that actually could have been observed by an omnipotent observer (omniscience). But we as ordinary mortals have no way of knowing the real value. But it will likely be around 3.5. Our sample all those dices that we actually throw. And we can start estimating the value of the population from our sample. In this case, the larger our sample, the better our estimation. Or more general:

The larger the sample, the more similar the distribution of sample and population

Let's add a practical experiment to our thought experiment. Although we will not be able to throw very many dices, we have a computer that can do this for us. We will use at first 10 dices.

```
n_dice <- 10
# the sample command is our random number generator, our dice
roll <- sample(1:6,n_dice, replace=T)
hist(roll, breaks=0.5:6.5)
```

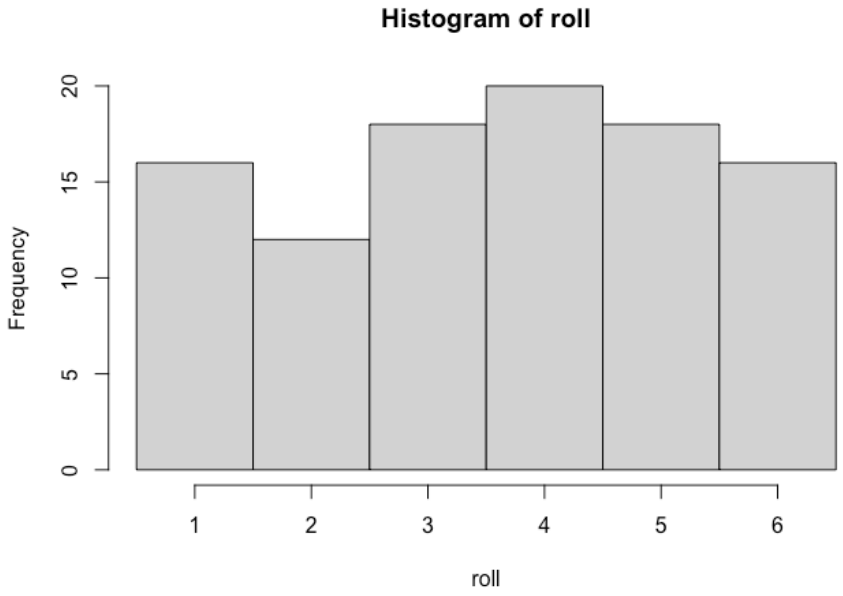



```
result <- mean(roll)
result
```

```
## [1] 3.3
```

Since this is a random experiment, at the time of writing I do not know what the result will be. But using the magic of R, I can't implement a little script snippet here so that I can now state the result is 3.3. But still I do not know whether this number is close or far away from all theoretical value. What I know is that if we increase the number of dices, we will get closer. Let's try this out:

```
n_dice <- 100
# the sample command is our random number generator, our dice
roll <- sample(1:6,n_dice, replace=T)
hist(roll, breaks=0.5:6.5)
```

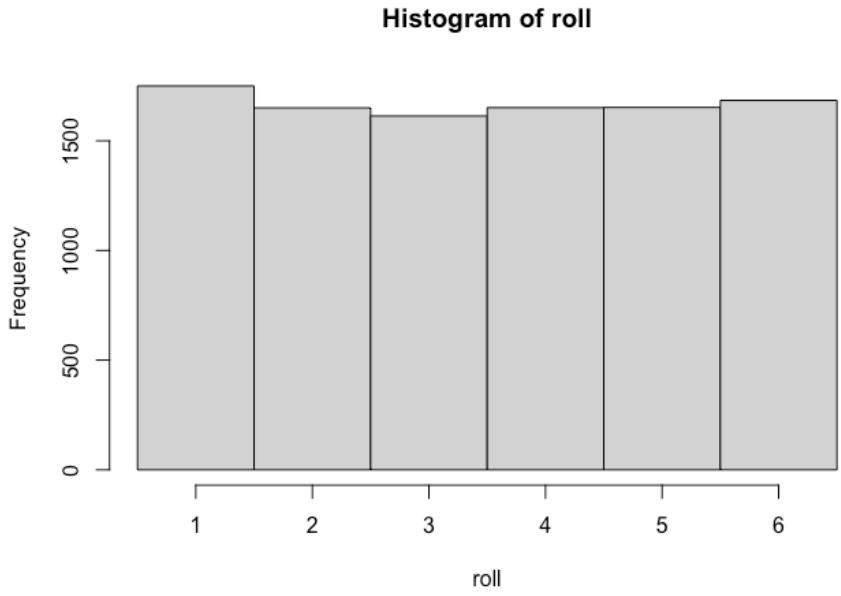



```
result <- mean(roll)
result
```

```
## [1] 3.6
```

I am pretty optimistic that this time the value of 3.6 is closer to the theoretical value then in the case above. But it does not need to be so. We increase the number of devices again, this time I am pretty sure that we will get close to 3.5.

```
n_dice <- 10000
# the sample command is our random number generator, our dice
roll <- sample(1:6,n_dice, replace=T)
hist(roll, breaks=0.5:6.5)
```




```
result <- mean(roll)
result
```

```
## [1] 3.4857
```

The more often one throws the dice experimentally, the more similar is the distribution of the sample to the population. There you have it: the value from 10,000 dices is3.4857 therefore quite close to the theoretical value. Note also the histogram: the more often we throw the dice, the more regular the histogram will look like.

The relative frequency of the random results converges against the probability of the random result

We can visualise this using a slightly different simulation. This time we are not simulating a specific parameters of our distribution, but the relative frequency of an individual result. Theoretically, the probability of throwing a 6 is 1/6. No let's watch how the frequency changes over time if we throw one dice after the other.

```
n_dice <- 10000

# We roll all devices at the same time
roll<- as.numeric(sample(1:6,n_dice, replace=T))
```

```
# Therefore we have to make them iterative
# preparing some variables to gather our results
list=0
ratio=0

# Now iteratively calculating the relative frequency
for (test in roll)
{
list<-append(list,test,length(list))
ratio<-append(ratio,sum(list==6)/length(list))
}

# And plot...
plot(ratio,type="l",ylim=c(0,1))
abline(h=1/6,lty=3)
```

Simulated dice experiment, the proportion of the number of 6 eyes is plotted, the dotted line shows the probability for 6 eyes.

Since again, this is a random experiment, I cannot guarantee but I am pretty optimistic that the solid line, representing our actual proportion of the number 6, will approach hour dashed line, the theoretical value. Note, that in the beginning we have some wiggles, but the longer of the experiment runs, the smoother and closer to the theoretical value it will get.

The law of the large number is the **bridge from the sample to the population**, it allows statements to be made about the population without knowing it.

6.5 Random variables

6.5.1 What is random at all?

Are use the term random already several times. As has been explained in the beginning, in a strict physical reality there is no such thing as randomness, except for probably quantum physics (We will not talk about quantum physics in this course!). Also, when you are using the random generator of your computer, still this random process is defined by the state in which the computer is at a given time. We should rather think of random processes as those processes that are either unknown or so complex that their outcome cannot be foreseen by us, but only estimated. A random variable is then the result of such a complex or unknown process. The values of such a random variable then represent the possible outcomes or results of this process. Whether we then really understand this as a random value is more of a philosophical question:

Coin toss: The result of a coin toss is *determined* by different physical laws (throwing force, density of air, gravity etc.)! Since we cannot control these, the result to be considered *random*!

An important step in order to statistically evaluate the results of a random variable is to convert them into the realm of real numbers (recode). In some detours, this is also possible for values that at first sight are difficult to convert into numbers. We will demonstrate this with the example of the coin toss.

6.5.2 Example for recoding (after Dolić)

A coin is flipped three times. The number of "heads" (H) is noted as a random variable. Possible results:

coin flip	x_i	$p(x_i)$
TTT	0	1/8
TTH	1	1/8
THT	1	1/8
HTT	1	1/8
THH	2	1/8
HTH	2	1/8
HHT	2	1/8
HHH	3	1/8

In this example, the results of the individual coin tosses are heads or tails. Different sequences of heads or tails, even with the same total number, represent different possible events. The total sample space of our experiment is therefore a result with respect to the order with replacement at the same time. The calculation of the number of possible outcomes therefore follows from our combination table from above and corresponds to the size of the event space for the single experiment (n=2, head or tails) to the power of the number of trials (k=3).

$n = 2; k = 3; 2^3 = 8$

Therefore, each event has a probability of 1/8. If we now look at the number of heads, it becomes clear that we can arrive at the same number of heads in different ways. There is only one possibility of not getting a head on three throws. Likewise, there is only one variation in which we get 3 heads. For one or two heads, on the other hand, there are 3 possibilities in each case.

If the number of heads represents our random variable, we arrive at the following distribution:

$$f(x_i) = \left\{ p(x_i = 0) = \tfrac{1}{8} p(x_i = 1) = \tfrac{3}{8} p(x_i = 2) = \tfrac{3}{8} p(x_i = 3) = \tfrac{1}{8} \right.$$

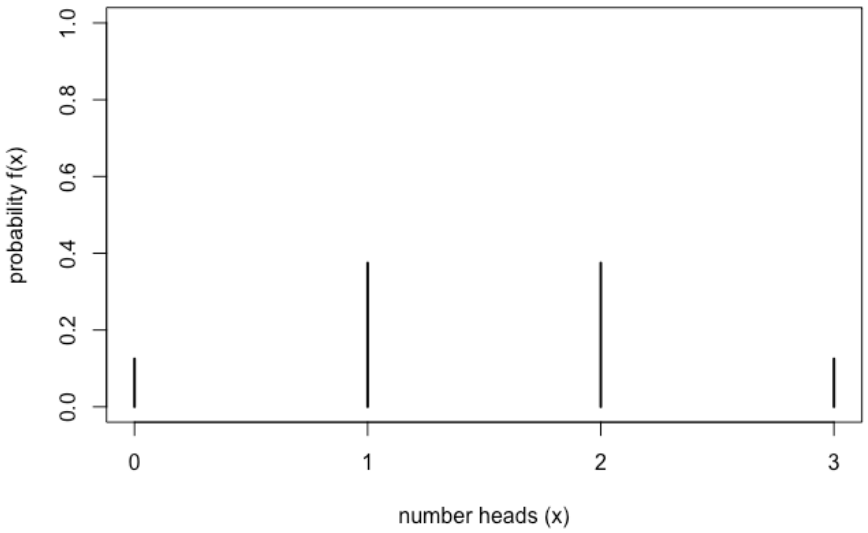
This distribution is called the **probability density function (PDF)** of our random variable. If it is a discrete (nominal) variable, it is also called the probability mass function (PMF).

Typical properties of such a function are:

Expected value: The value that is most probable.

Dispersion: The variance of distribution

As with any other distribution function, one can of course calculate Skewness and Kurtosis.



In addition to this quite intuitively understandable distribution function, there is also the **cumulative distribution function (CDF)**, which is very relevant for many statistical applications. Especially in the area of statistical tests, we will make use of it below.

This function represents the cumulative sum of the probability density function. It answers the question of how likely it is to obtain a certain value of the random variable or a smaller one. In our example: "What is the probability of having up to two heads?"

One important of its properties:

$0 \leq F(x) \leq 1$

F(x) is monotonous not falling. This means that as the value of the variable increases, the probability can never become smaller.

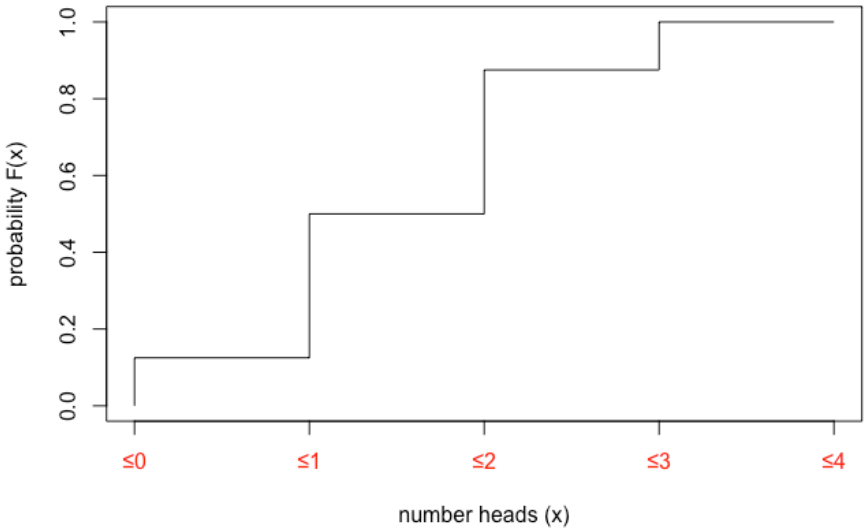
$F(x_1) \leq F(x_2) ... \leq F(x_n)$

In the example of our coin toss, this function results as follows:

$$f(x_i) = \Big\{ p(x_i < 0) = 0 p(x_i \leq 0) = \tfrac{1}{8} p(x_i \leq 1) = \tfrac{4}{8} p(x_i \leq 2) = \tfrac{7}{8} p(x_i \leq 3) = 1$$

It is obvious that values smaller than zero are impossible. Values less than or equal to zero, on the other hand, have a probability of 0.125 or one eighth, as already in the PDF. Values less than or equal to one have a probability of four-eighths, which is the sum of the probabilities for zero and one head. And so on.

This is what the graphical representation looks like:



Now, of course, the question arises, what is the point of all this! How does this knowledge of probability theory help us to better understand statistical tests? We will continue with our example of the coin toss and now develop a statistical test to assess whether it is fair or marked coins that are being played with. Our example will include all the elements of a statistical test, and the procedure is basically transferable to other statistical tests. So we will take a look under the bonnet of a statistical test.

6.6 Building a statistical test from scratch

In this section, I would like to use the knowledge, that we just gained, to build a statistical test from scratch. We will use our tools to estimate, if the coins with which we make our random experiment, are fair or marked. This will give us insight, how was statistical test actually works.

So, our scientific question is: **Is someone playing with biased coins?**

For this, we have to figure out, how one can significantly (error probability 5%) determine that the coins are biased and always show e.g. head?

At first, of course we have to construct our hypothesis and the respective null hypothesis. This could look like the following:

H_0: The coins are not biased, the distribution corresponds to the distribution of an unbiased coin toss

H_1: The coins are biased, the distribution differs significantly from the distribution of an unbiased coin toss

Let's also assume, that we do not have very much time for this test, so we limit our sample size or the number of throws to 20.

n=20 throws

No, we need a **Rejection range**: the number of heads that is high enough to be sure with a certain probability the coin is biased towards heads. We have to decide, I'll sure we need to be, before we reject the null hypothesis, that the coin is not biased. For this, we stick to the traditional 0.05 significance level. That means, using our axioms, there need to be a 95% chance that rejecting the null hypothesis is a good idea. If the probability of the random occurrence of a result is less than 5%, the occurrence of that event is not random with 95% probability.

Having figured out this, now we have to determine for how many heads results on 20 throws there is the probability 5% or lower, given unbiased coins. Basically, this question can be solved using the probability concept of Laplace:

$p(A) = \frac{Number\ of\ positive\ results}{Number\ of\ possible\ results}$

We have to count the relative frequency of an event (heads). For this, we need:

- the number of possibilities for the positive events (heads)
- the total number of possible events (total number of possible results of throws of 20 coins)

6.6.1 Number of possible events

Let's look at some examples, how such random experiment could result:

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

There is only one possibility, that we do not have any head in our sample.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
H	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
T	H	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

If we have only one head, it matters, on which throw this head falls. So there are 20 possibilities for this.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
H	H	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
H	T	H	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

If we have two head, there is much more possibility for variation. Because also here, the order of the results matters, therefore we have to consider the position of the two heads in our sample. We cannot estimate the number of possibilities here by just looking (or at least I can't do that). So either, we have to count it, or we have to calculate it.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
H	T	T	H	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
H	H	H	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T	T

With three heads, the possibilities should be even larger, because we have more heads that could take up certain positions in our sample. And this is just

beginning: for figuring out, how many possibilities in total there are, we have to repeat this exercise for all possible numbers of head, and then sum them up. Counting them directly will take quite a while, so it is a practical way. Better try calculation!

To calculate the total possible number, we can use our combinatorics table from above. We have already decided that if we have a head, it makes a difference whether it falls first, or second, or so on. I.e. that the order matters to us.

[T,T,H] represents a different case that [H,T,T]

On the other hand, we have to decide whether the probability of the next throw is affected by the previous one. If a coin shows tails, does that change the probability of the next coin to show tails (without replacement) or does this probability remain unchanged (with replacement)?

The answer is: **No**. Chances does not change, so with replacement.

To make things easier, here is the table again:

	Variation (with respect to order)	Combination (without respecting the order)
with 'putting back'; with replacement	n^k	$\frac{(n+k-1)!}{k!*(n-1)!}$
without 'putting back'; without replacement	$\frac{n!}{(n-k)!}$	$\frac{n!}{k!*(n-k)!} = \binom{n}{k}$

Our choice should be the upper left cell, that is variation with replacement.

n^k

We can also build that differently: For the first coin, there exists two possibilities, head or tail. For the second coin, there are also two possibilities, and so on. Each trial is independent from each other, so the probabilities will not change. With two coins, we have four possible results.

[H, H]

[T, H]

[H, T]

[T, T]

That means, for each additional coin we have to multiply the number of possible outcomes (head or tail = two). That means, in a two coin situation:

$2 * 2 = n^k = 2^2$

And in the three coin situation:

$2 * 2 * 2 = n^k = 2^3$

And finally in the 20 coin situation:

$2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 = n^k = 2^{20}$

Now we can assign our values to the variables:

- 2 possible cases (Heads, Tails) : n
- 20 possible positions : k

number of possible results: $n^k = 2^{20} =$ **1048576**

2 results can be distributed in 1048576 ways on 20 positions

6.6.2 Number of positive events

Now comes the tricky part: We have to calculate the number of positive events for our set up. Let's start with the easy part: we know already, that there is only one possibility to distribute zero heads on our 20 positions. We also know, that there are 20 possibilities to distribute one head on the 20 positions.

Number of positive events: How many possibilities are there to distribute a fixed number of coins with head to 20 places?

What we have done intuitively in this situation is to change our perspective: the question is now how many possibilities there are to distribute one head to 20 possible positions. This means that there are now 20 possible outcomes (n), but our number of trials is only one (one head, k).

n=20 places, k= cases head

So let's continue off for experiment using these numbers. How many possibilities are there to distribute zero heads on 20 places:

number of possible results: $n^k = 20^0 =$ **1**

That looks fine. It is the number that we expected from our counting. With this promising result let's continue:

number of possible results: $n^k = 20^1 =$ **20**

That also looks fine. We expected 20, we got 20. But now, with more than one trial, it becomes difficult. If we have already put one coin on one position, now there are lesser positions available for the next head. With one position filled, there are only 19 other positions open. Also, when we have two coins, in the first place we have also to coins to distribute. For us in the end, both quotes are indistinguishable, but for the number of events the coins matter.

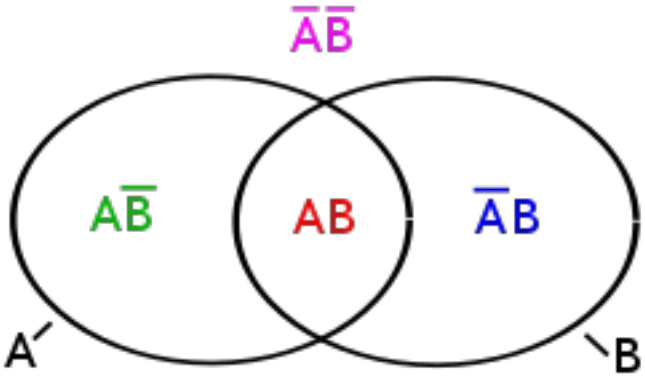
Let's demonstrate that with a more simple example having only three positions and two coins:

1	2	3
H1	H2	T
H1	T	H2
T	H1	H2
H2	H1	T
H2	T	H1
T	H2	H1

For the first coin, there are three possible options for its position. For the second coin there are only two possibilities left. On the perspective of the final result, the first three lines are identical to the last three lines. But from the perspective of the event number, they represent different events. So in this case, the total number of events including the order could be calculated like that:

$3_{first\ coin} * 2_{second\ coin} = 6$

If we want to ignore the order, we have to remove those possibilities that are equivalent to us. Remember the image from the set theory (although it is not perfectly equivalent):



Do union of the number of cases is the number of cases from A times the number of cases from B minus the number, where A and B both a present. Since in the case of the first trial, we have two equivalent coins, we have to divide the number by two. Consequently, we also have to go to divide the second number of cases by one, which in practice does not change the result.

$(3_{first\ coin}/2_{two\ coins}) * (2_{second\ coin}/1_{one\ coin}) = 3$

No we can do the same for our 20 coin example:

```
(20 / 1)
```

```
## [1] 20
```

```
(20 / 2) * (19 / 1)
```

```
## [1] 190
```

```
(20 / 3) * (19 / 2) * (18 / 1)
```

```
## [1] 1140
```

```
(20 / 4) * (19 / 3) * (18 / 2) * (17 / 1)
```

```
## [1] 4845
```

Actually, this is essentially the same like using the formula from the lower right cell of our combinatorics table:

$\frac{n!}{k!*(n-k)!} = \binom{n}{k}$

Let's try this out in R:

```
n = 20; k = 3
factorial(n)/(factorial(k) * factorial(n-k))
```

```
## [1] 1140
```

We could have gotten here much faster, admittedly, if we would have blindly trust the table. But now you know, why is this binomial coefficient works. You can also sum up all the possible outcomes from k=0 to k=20, to check if our initial calculation of the total number of possible results was correct:

```
n = 20
total = 0
for (k in 0:20) {
  total <- total + factorial(n)/(factorial(k) * factorial(n-k))
}

total
```

```
## [1] 1048576
```

```
total == 2^20
```

```
## [1] TRUE
```

6.6.3 From counts to probabilities

If you are still with me, then now it is easy to calculate the possible number of events for the different numbers of head. All we have to do is to calculate its value using the normal coefficient.

n = 20 places, k = number heads

Number of positive events: calculated according to binomial coefficient, possibilities to arrange number of outputs in number of throws

No times head: n = 20, k=0: only one possibility

1 times head: n = 20, k = 1: 20 possibilities

2 times head: n = 20, k = 2: 190 possibilities

3 times head: n = 20, k = 3: 1140 possibilities

4 times head: n = 20, k = 4: 4845 possibilities

5 times head: n = 20, k = 5: 15504 possibilities

6 times head: n = 20, k = 6: 38760 possibilities

$\frac{n!}{k!*(n-k)!} = \binom{n}{k}$

One thing by the way: Of course there is an easy way to calculate this using R. The command is `choose()`. Are used it to calculate the numbers above, but we can also demonstrate it directly:

```
n=20; k=3
choose(n,k)
```

```
## [1] 1140
```

Finally, we can calculate probabilities! We have the total number of possible results, and we can calculate our positive results. All that is left to do is to use the formula of Laplace:

$p(A) = \frac{Number\ of\ positive\ results}{Number\ of\ possible\ results}$

Number of possible events = $n^k = 2^{20} = 1048576$

number head	possibilities	positive/possible	cumulative
0	1	0.00000	0.00000
1	20	0.00002	0.00002
2	190	0.00018	0.00020
3	1140	0.00109	0.00129
4	4845	0.00462	0.00591
5	15504	0.01479	0.02069
6	38760	0.03696	0.05766

Note, that this table there is a column for cumulative. If you want to know, how many ads would indicate a biased coin, we are not interested in the probability of getting exactly the specific number, but the probability of getting this number or less. Therefore, we have to cumulatively sum up the probabilities. This is equivalent to the cumulative distribution function (CDF). Since the probability since the probabilities do not care whether we are looking for heads or tails, if there is only a 2% chance of getting five heads, there is also only a 2% chance of getting five tails. When ever in this situation we observe in our sample, that there are only five or less number of tails present, we can be rather sure that the coins are biased towards heads. There is less than 2% probability, that this situation could come from a random effect.

6.7 The Binomial Distribution

With this example we have developed a statistical distribution on our own. What we have detected is the so-called binomial distribution. This theoretical distributions are shortcuts for the calculation of probabilities. They represent model cases, against which we can compare our sample distribution. For these models it is assumed that your data following certain characteristics and have certain parameters. In case of to be normal distribution, it is a situation where you have two possible outcomes with certain probabilities. To binomial distribution then calculate the probability for the number of outcomes.

The equation for the general binomial distribution is this:

$B_{n;k;p} = \binom{n}{k} * p^k * (1-p)^{n-k}$

You can see, dead beside our well-known n and k, there is also a p. This p represents the probability of the positive event, for example the probability of getting heads with a coin flip. In case of a fair coin, p is equal to 0.5. You would assume that there is an even chance for heads and tails. Since the safe event has a probability of 1, if the probability is distributed via between the two options, it should be 0.5 each.

Now, we can directly calculate the probability using this formula: For example 2 Heads, 18 Tails, k = 2, n = 20, p = 0.5

$190/1048576 = 0.00018 = \binom{20}{2} * 0.5^2 * (1-0.5)^{20-2} = 190 * 0.25 * 0$

If you like, you can compare this number to our calculation by hand from the table above. They should be the same.

Again, in our there is a shortcut for calculating this probability, the command is `dbinom()`.

```
dbinom(6,20,0.5)
```

```
## [1] 0.036964
```

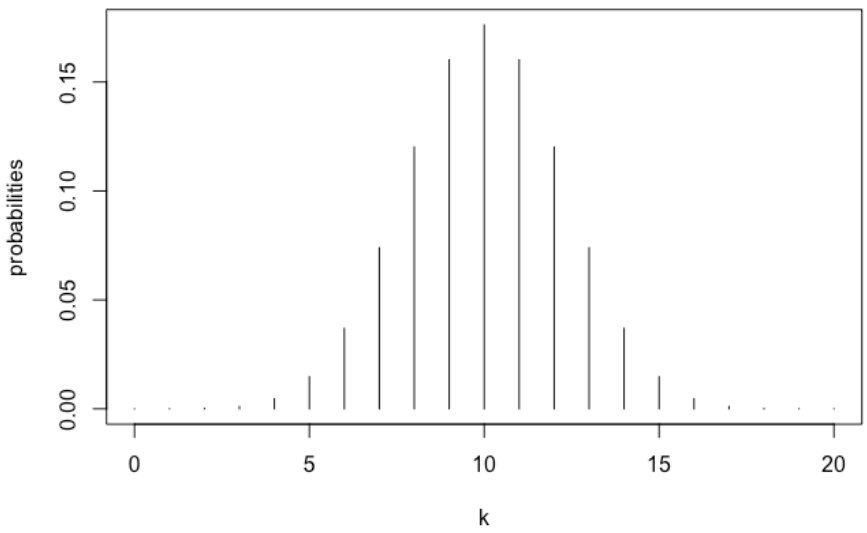
And also for the cumulative distribution function, in that case it is `pbinom()`

```
pbinom(6,20,0.5)
```

```
## [1] 0.057659
```

With these functions, now we have something to play with to explore the probabilities of the distribution and also of statistical tests in general. For example, we can plot the probability function, to get a graphical representation of our coin flips:

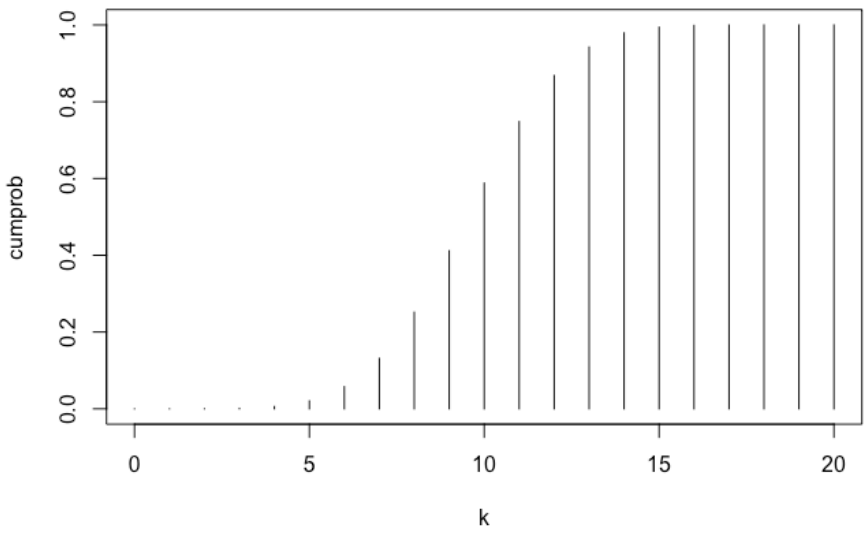
```
k <- 0:20
probabilities <- dbinom(k,20,0.5)
plot(k,probabilities, type = "h")
```



You can see, that the highest probability is to get 10 heads and 10 tails. The distribution is symmetrical, the smaller the number of heads or tails get the more unlikely this situation will be.

Also, we can plot the cumulative density function:

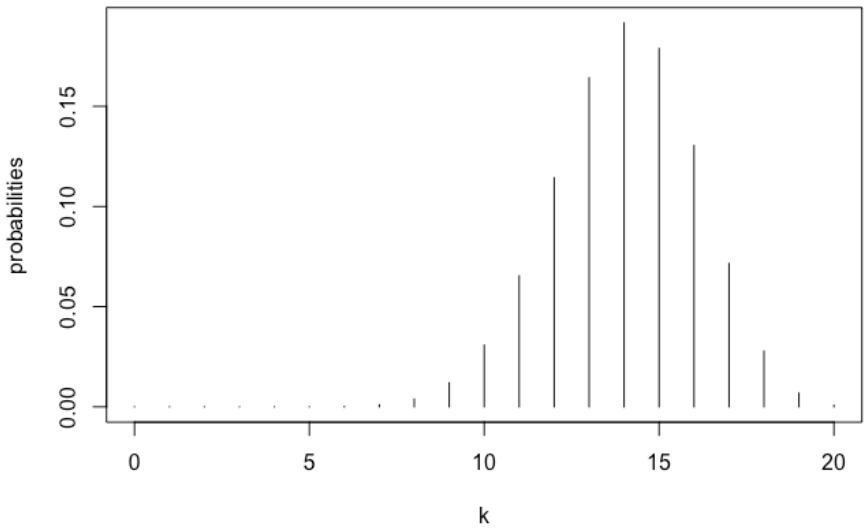
```
k <- 0:20
cumprob <- pbinom(k,20,0.5)
plot(k,cumprob, type = "h")
```



There is a 59 % probability to get 10 or less heads (or tails).

What you also can do is try out, how different probabilities of getting a head would result. This mean, you can build a model of how biased your coin is, and then try to fit this model to your data. You could also compare different models to evaluate which one is the best fitting. These are advanced statistical procedures, but nonetheless very often takes place in scientific investigations. Although the models might be more complicated than just changing a probability, the general workflow is the same. For now, we will only plot the probability density function for the situation, when a coin has a 70% chance of getting a head.

```
k <- 0:20
probabilities <- dbinom(k,20,0.7)
plot(k,probabilities, type = "h")
```



This case, you can see that the most likely number of heads would be 14.

6.7.1 The binom.test

Of course, there is also an inbuilt binomial test in R. The test takes the parameters of the number of successes, the number of trials and the probability of success. The result is whether or not the sample is significantly different from the binomial model with that probability. The command is `binom.test()`

```
binom.test(5,20,0.5)
```

```
##
##  Exact binomial test
##
## data:  5 and 20
## number of successes = 5, number of trials = 20, p-value = 0.041
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.086571 0.491046
## sample estimates:
## probability of success
##                   0.25
```

The resulting output contains a p-value. As with the other tests before, this p-value indicates statistical significance. In this case, it is statistical significance. Our sample is significantly different from a fair binomial trial. Beyond that, it also estimates the probability of success from the sample. In this case it is 25%. This is equivalent to our number of successes, divided by the number of trials. So with 20 trails, if only a quarter of the results are successes, then there is a significant divergence from the binomial distribution.

Therefore, our rejection range is Number Tails < 6: this results in 95% probability, that something is wrong with the coins. (25% of the tosses)

6.7.2 Sample size

With our toy model we can also explore the effects of sample size. Let's tenfold the number of trials, and see, what the rejection range would then be:

N=200 throws

```
binom.test(85,200,0.5)
```

```
##
##  Exact binomial test
##
## data:  85 and 200
## number of successes = 85, number of trials = 200, p-value = 0.04
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.35556 0.49670
## sample estimates:
## probability of success
##                  0.425
```

In case of 200 trials, we get a significant result already with 85 successes. This represents 43% of the coin flips. This means, that we relatively need much lesser surprise to be sure that there is something wrong. Of course, this is related to the law of the large number: the more trials we have, the more likely the sample distribution is close to the theoretical distribution. There is less influence of random effects.

Here now, the rejection range is Number Tails < 85: this results in 95% probability, that something is wrong with the coins. (42% of the tosses)

To binomial distribution will be the only distribution, that we investigate in this detail. There are other distributions, that are interesting in situations where we have different variables, and deal with different underlying processes. To binomial distribution is helpful for analysing nominal variables and test their

distribution according to specific probabilities. Other helpful distributions of examples the normal distribution, in case of metric data, who are influenced by a lot of independent effects. The Poisson distribution can be used for rare events, for example for analysing spatial or temporal distributions of events. There will be an in-depth level in which these distributions will be dealt with in more detail. However, this will not be an exam-relevant part of this exercise. Rather, those who are not directly interested can skip the next chapter and go directly to the parametric tests, in which we also deal intensively with the normal distribution, but for whose understanding the next chapter is not essential. If, however, you are interested in the concept of standard error and the question of the basis on which we can estimate mean values of populations on the basis of samples, what a confidence interval is, the next chapter is recommended, perhaps only in the second reading of this book.