R in 02402: Introduction to Statistic

Per Bruun Brockhoff
DTU Informatics , DK-2800 Lyngby

17. januar 2012

Indhold

1 Using R on the Databar-System at DTU

1.1 Access

A description of the databar-system can be found at `http://www.gbar.dtu.dk`.
In order to use the G-bar, a login (student number) and a password is needed. All students admitting DTU get a student number and a password. Login can be done

- via thin client (a terminal on DTU)
- over the internet - use ThinLinc

See `http://www.gbar.dtu.dk/introguide/introguide_en.pdf`) for a further description of this.

1.2 R

After login, a menu with various programs appears by clicking on the middle button on the mouse when the pointer is held against the background. S-PLUS can be found under 'Statistics'. DTU has also a campus license for S-PLUS, see Section **??**.
The program R is an open source statistic program, that to a great extent is a copy of the commercial program Splus. A short introduction is found in Appendix C in the textbook for the course 02402. It can be run from the G-bar, but it is fast and easy to download the program to your own laptop, wether you use Windows, Mac or Linux: `http://www.r-project.org`.
It is recommended to download the program to your own laptop, since you do not have access to the G-bar at the exam, but it is okay to bring you own laptop for the exam and use the program.
The program can be run from the command window (R Console), where you can run commands/functions by the prompt. In the Gbar the program can also be run "interactively". In the windows version the console is automatically packed in a user interface, where the console is the only active window in the start-up, but with a few overall menus. This version of the program can also be found in the Gbar.
With advantage you can always start a script-editor ('File'→'New Script') in R, from where it is easy to submit commands without "loosing them again".
In the course 02441 Applied Statistics and Statistical Software, which is run as a 3-weeks course in January every year, you will get the chance to work with R for practical statistical problems - a direct project oriented way to get a more advanced level of working with the program, see `http://www.imm.dtu.dk/courses/02441`), where you also can find references to good (online) textbook material. Also in the new version of the course 02418 (substitute for 02416) you will get your competences for using R to do more advanced statistical analysis further devveloped.

1.2.1 R Commander

In spite of that the program is run by menus, then there are no menus used in this basic form to carry out the statistical analysis, which are known from most standard commercial statistic programs. Therefore we will use an additional advanced menu based level to do statistic in R. There are several of such advanced levels of R. We will use the package "Rcmdr" also called

”R Commander”. This is ready for use in the R version of the databar - to start: by the prompt type:

```
> library(Rcmdr)
```

You can also install ”Rcmdr” on your own laptop. Then you just have to do the following in R: (internet access is required)

1. Click ’Packages’ →’Install Packages(s)’
2. Choose ”Mirror Site” - e.g ”Denmark”
3. Find and choose the package ”Rcmdr” at the list - so that the package will be installed(that is, copied to your computer)
4. Run: library(Rcmdr) by the prompt. (This will have to be typed every time you start R and in order also to start the R Commander) (The first time you will probably be asked to install a few packages - just say yes to install all necessary packages).

Please note about the installation and the first load of R Commander: For certain platforms, there may be details making a bit of trouble. For Windows 7 and Vista: If you install R, as probably suggested by the computer, under ”Pogramme Files”, then you must run R ”as administrator” to be allowed to install packages. Either right-click when starting R to do this OR simply once and for all install R somewhere else, e.g. in the root directory. For Mac-users there are a couple of additional challenges: You must make sure that ”X Windows” is available and that ”Tcl/Tk for X Windows” is installed before doing the R installation. For any platform, please check the following web-place for details (giving Mac-users direct links tp what is necessary):
(`http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/installation-notes.html`).

Note, that R Commander gives an extra program window, this includes a script-editor and an output window respectively. When you do graphics, the the graphs will automatically pop up in separate windows. Note also that: For all menu based choices, you will not only get the result for these choices but also the R-scripts, that correspond to these choices.

1.2.2 Import af data

Data is imported into R by using the ”read.table” function, see Appendix C. Or you can use the menus in R commander: ’Data’ →’Import Data’. Different choices of file formats are possible - default setting for ”text-files” usually works for .TXT-files, which are used in the text book (www.pearsonhughered.com/datasets) or more directly by (as long as the browser is run from a computer at DTU)
`http://www.imm.dtu.dk/courses/02402/Bookdata8ED`
(7ed: `http://www.imm.dtu.dk/courses/02402/Bookdata`). When importing data from a spreadsheet, it can sometimes be better to export the data from the spreadsheet to comma delimited (CSV) format before importing it into R.(The package `RODBC` provides a more advanced way of directly importing e.g. excell files but we will not go into this here)

1.2.3 Using the program

1.2.4 Storage of Text and Graphics

Text from windows in R can be copied into other programs in the usual manner:

- mark the text by holding the left mouse button down and drag it over the desired text.
- open another program (e.g. StarOffice), place the pointer at the desired location and press the middle mouse button.

All text in the 'Commands Window' or the 'Report Window' can be stored in a text-file by activating the window and choose 'File' $\rightarrow$ 'Save As ...'.
Graphics can be stored in a graphics-file by activating the graphic window and choose 'File' $\rightarrow$ 'Save As ...'. It is possible to choose from a range of graphics formats (JPEG is the default).

2 R in 02402

2.1 Syllabus

The R part included in the syllabus are the sections in this notes that are marked as 'thorough' ('t') reading in the lecture plan. This part of the syllabus will be tested in the 'Test assignments' after each main section of the notes. Neither these exercises nor the exam demand access to the program, BUT understanding the different aspects of the program is necessary to solve them. This understanding is achieved by self-training with the program during the course. In each main section of these notes, some exercises from the book that can be solved using the program are be listed that you can use for self-training. Twice in the semester, exercises in the Databar are scheduled where it is possible to work with the program under guidance. Apart from that, you need to work with the program by yourself.
By installing R on your own laptop you can replace your calculator with the program - you are allowed to bring your laptop with you to the exam. It must be stressed that even though the program is able to calculate things for the user, understanding the details of the calculations must NOT be forgotten - understanding is an important part of the syllabus.
Please also note that Sectioin 9 in this note is an actual syllabus-chapter with theory and methods for the so-called simulation based methods.

2.2 Introductory R Exercise

It is possible to use R in two different ways as described earlier. 1) As a Menu-based data analysis program, 2) As an interactive calculator with a wide variety of built-in functions and procedures. We will mostly use 1) in this exercise, but it is also recommended to try 2) when proposed! (Should the technical challenges in making the R Commander work turn out to be too big on the day of the course start, then it is still a nice exercise to base it on method 2).)

1. Start R

2. Download via Campusnets file sharing (course 02402) the Excel-file: `karakterer2004.csv`, which contains 10 variables (columns) and 1555 observations (rows), corresponding to 1555 schools.

Number	Name of variable	Description (grades summer 2004)
Variable 1	Skole	Name of school
Variable 2	Type	Type of school
Variable 3	Type2	Type of school
Variable 4	Amt	Amt. name
Variable 5	Kommune	Kommune name
Variable 6	Dansk.Eks	Mean exam-score of school in 9th grade Danish
Variable 7	Dansk.Aars	Mean year-score of school in 9th grade Danish
Variable 8	Mat.Eks	Mean exam-score of school in 9th grade math
Variable 9	Mat.Aars	Mean year-score of school in 9th grade math
Variable 10	Antal	Number of students in 9th grade

3. Import the data into R, Data,Import Data, From text file... In the menu window poping up, choose (yourself) an (R-)name for the data set, change ”Field Separator” to ”Other” and ”Specify”the semicolon: ”;”. And as ”Decimal-Point Character” choose ”Comma” - click OK at the end.

4. Look at the data:(Click: ”View Data Set”)

5. Fill in the following table of summary school statistics: (We look at the school numbers WITHOUT taking the different number of students in the schools into account.

   - Either: Use the menu: ”Statistics”, ”Summaries”, and then either ”Actice data set” or ”Numerical Summaries...”. Mark the relevant variables, click OK.
   - Or: Use the functions listed on the top of page 529 in Appendix C. (remember to write: **attach(karakterer2004)** first)

	Dansk.Eks	Dansk.Aars	Mat.Eks	Mat.Aars
Mean				
Median				
Variance				
Standard deviation				
Upper quartile Q_3				
Lower quartile Q_1				

6. What ’story’ does this tell?

7. Compare the histograms of the four distributions.
   - Either: Use the menu Graph, Histogram
   - Or: Use the function **hist()**
8. Make boxplots for each of the distributions.
   - Either: Use the menu Graph, Boxplot
   - Or: Use the function **boxplot()**
9. Try to visualize the number of schools of each type. (Bar graph and/or pie graph) (Choose the Graph menu)
10. Compare the exam-score distributions in math between the school types (variable: Type). (Use the Graph menu, Boxplot, select Type as "plot by groups" variable)
11. Examine whether there is a connection between the scores. (Graph, scatterplot, select x and y-variables)

3 Discrete Distributions, week 2

3.1 Introduction

Commands are written after the prompt ">".
The command `3:7`, generates a vector of integers from 3 to 7 and `7:3` generates them in reverse order.

```
> 3:7
[1] 3 4 5 6 7
> 7:3
[1] 7 6 5 4 3
```

The command `prod(x)` multiples all the numbers in the vector `x`:[1]

```
> prod(2:3)
[1] 6
```

The following distributions are considered:

R	Distribution
`binom`	Binomial
`pois`	Poisson

The hypergeometric distribution is also available in R (`hyper`). However, the parametrization is different from the textbook and will not be considered here. For every distribution, there are 4 functions available, as described in the book. The functions appear by adding one of the following letters in front of the names given in the table above.

[1] Similarly, `sum(x)` adds all the numbers in the vector `x`

d Probability distribution $f(x)$.

p Cumulative distribution function $F(x)$.

r Random number generator (To be used in section 9 of this note).

q Quantile of distribution

3.1.1 The Binomial Distribution:

- $b(x; n, p)$ as on page 86 (7ed: 107) in the text book is written in R as `dbinom(x, n, p)`.
- $B(x; n, p)$ as on page 87 (7ed: 107) in the text book is written in R as `pbinom(x, n, p)`.

3.1.2 The Poisson Distribution:

- $f(x; \lambda)$ page 104 (7ed: 127) in the textbook, in R: `dpois(x, lambda)`.
- $F(x; \lambda)$ page 105 (7ed: 128) in the textbook, in R: `ppois(x, lambda)`.

3.2 Self-Training Using Exercises from the Textbook

- Solve exercise 4.15 by using both `dbinom` and `pbinom`.
- Solve exercise 4.19 by using both `dbinom` and `pbinom`.
- Solve exercise 4.57 using R.
- Solve exercise 4.59 using R. Try to use both `dpois` and `ppois` when solving question (a).
- Solve the extra exercises 4.2, 4.16 and 4.21using R.

3.3 Test-Assignments

3.3.1 Exercise

Let X be a stochastic variable. When running the R-command `dbinom(4,10,0.6)` R returns 0.1114767, written as:

```
> dbinom(4,10,0.6)
[1] 0.1114767
```

What distribution is applied and what does 0.1114767 stand for?

3.3.2 Exercise

Let X be the same stochastic variable as above. The following are results from R:

```
> pbinom(4,10,0.6)
[1] 0.1662386
> pbinom(5,10,0.6)
[1] 0.3668967
```

Calculate the following probabilities: $P(X \leq 5)$, $P(X < 5)$, $P(X > 4)$ and $P(X = 5)$.

3.3.3 Exercise

Let X be a stochastic variable. From R we get:

```
> dpois(4,3)
[1] 0.1680314
```

What distribution is applied and what does 0.1680314 stand for?

3.3.4 Exercise

Let X be the same stochastic variable as above. The following are results from R:

```
> ppois(4,3)
[1] 0.8152632
> ppois(5,3)
[1] 0.916082
```

Calculate the following probabilities: $P(X \leq 5)$, $P(X < 5)$, $P(X > 4)$ and $P(X = 5)$.

4 Continuous Distributions, the Normal Distribution, Week3

4.1 Introduction

Look at the beginning of Appendix C in the textbook, specially 'Probability Distributions' and 'Normal Probability Calculations', page 530 (7ed: 611) (not `qqnorm`). The following distributions are considered:

R	Distribution
`norm`	Normal
`unif`	Uniform
`lnorm`	Log-normal
`exp`	Exponential

As mentioned above, there are 4 functions available for every distribution. The functions appear by adding one of the following letters in front of the names given in the table above.

`d` Probability distribution $f(x)$.

`p` Cumulative distribution function $F(x)$.

`r` Random number generator (To be used in section 9 of this note).

`q` Quantile of distribution

4.1.1 The Normal Distribution:

- $f(x; \mu, \sigma^2)$ page 125 (7ed: 154) in the textbook, in R: `dnorm(x, μ, σ)`.
- The distribution function for a normal distribution with mean μ and variance σ^2, in R: `pnorm(x, μ, σ)`. That is $F(z)$ page 126, in R: `pnorm(z, 0, 1)`[2]
- Assume that Z is a standard normally distributed stochastic variable. The value z for $P(Z \leq z) = p$ is achieved by `qnorm(p)`. This value is called the p-quantile in the standardized normal distribution.

Note that R uses σ, not σ^2.

4.2 Self-Training Using Exercises from the Textbook

- Solve exercise 5.19 using `pnorm`.
- Solve exercise 5.21 using `qnorm`.
- Solve exercise 5.27 using R.
- Solve exercise 5.33 using R.
- Solve exercise 5.114 (7Ed: 5.113) using R.

4.3 Test Assignments

4.3.1 Exercise

The following R commands and results are given:

```
> pnorm(2)
[1] 0.9772499
> pnorm(2,1,1)
[1] 0.8413447
> pnorm(2,1,2)
[1] 0.6914625
```

Specify which distributions are used and explain the resulting probabilities (preferably by a sketch).

4.3.2 Exercise

What is the result of the following command: `qnorm(pnorm(2))`?

[2] or simply `pnorm(z)` since R uses the standardized normal distribution as default.

4.3.3 Exercise

The following R commands and results are given:

```
> qnorm(0.975)
[1] 1.959964
> qnorm(0.975,1,1)
[1] 2.959964
> qnorm(0.975,1,2)
[1] 4.919928
```

State what the numbers represent in the three cases (preferably by a sketch).

5 Continuous Distributions, Week 4

5.1 Introduction

Look at the beginning of Appendix C in the textbook, specially 'Probability Distributions' and 'Normal Probability Calculations', page 530 (7ed: 611).

5.1.1 The log-normal distribution:

- $f(x)$ page 136 (7ed: 166)in the textbook, in R: `dlnorm(x, α, β)`.
- The probability in the example page 137 (7ed: 167) in the textbook, in R: `plnorm(8.2,2,0.1)-plnorm(6.1,2,0.1)`.
- Can also be achieved by: `pnorm(log(8.2),2,0.1)-pnorm(log(6.1),2,0.1)`.
- And finally with: `pnorm((log(8.2)-2)/0.1)-pnorm((log(6.1)-2)/0.1)`.
- Note that the natural logarithm is named `log` in R.
- Note also that the calculated probability in the textbook is a bit different. This is because numbers inserted into the standard normal distribution are rounded before looking up in the table. The results from R are therefor more correct than in the textbook.

5.1.2 The Uniform Distribution:

- $f(x)$ page 135 (7ed: 165) in the textbook, in R: `dunif(x, α, β)`.

5.1.3 The Exponential Distribution:

- $f(x)$ page 140 (7ed: 170) in the textbook, also in R as `dexp(x, 1/β)`.

5.1.4 The Normal Scores Plot

The R function `qqnorm` can be used to make a normal scores plot, as described on page 530 (7ed: 611) in the textbook. However, the method used by R is a bit different from the one described in the book. In what way the two methods differ is described in the solution to exercise 5.120. In addition, R exchanges the x-and y-axes. If you import the data to be used in exercise 5.120 (”2-66.TXT”) AND attach it, then the plot is produced simply by typing `qqnorm(speed)`.

5.2 Self-Training Using Exercises from the Book

- Solve exercise 5.46 using `punif`.
- Solve exercise 5.51 using `plnorm`.
- Solve exercise 5.58 using `pexp`.
- Solve exercise 5.38 using `pnorm`.
- Solve exercise 5.111 using `punif` (7Ed: 5.110)
- Solve exercise 5.120 using `qqnorm`.(7Ed: 5.119)

5.3 Test-Assignments

5.3.1 Exercise

Write down the equation and/or sketch the meaning of the following R function and result:

```
> punif(0.4)
[1] 0.4
```

5.3.2 Exercise

Write down the equation and/or sketch the meaning of the following R functions and results:

```
> dexp(2,0.5)
[1] 0.1839397
> pexp(2,0.5)
[1] 0.6321206
```

5.3.3 Exercise

Write down the equation and/or sketch the meaning of the following R function and result:

```
> qlnorm(0.5)
[1] 1
```

6 Sampling Distributions, Week 5 and 8

6.1 Description

Look at the beginning of Appendix C in the textbook, specially 'Sampling Distributions' page 530 (7ed: 612). The sampling distributions introduced in the textbook are:

R	Distribution
`t`	t
`chisq`	χ^2
`f`	F

As mentioned above, there are 4 functions available for every distribution. The functions appear by adding one of the following letters in front of the names given in the table above.

`d` Probability distribution $f(x)$.

`p` Cumulative distribution function $F(x)$.

`r` Random number generator (To be used in section 9 of this note).

`q` Quantile of distribution

6.1.1 The t-Distribution

- The numbers in Table 4, page 516 (7ed: 587) in the textbook are given by: `qt(`$1-\alpha$`,`ν`)` and the corresponding α-values by: `1-pt(x,`ν`)`, where x are the numbers in the table and ν are the number of degrees of freedom.
- The probability of being below -3.19, in the example page 188 (7ed: 218) in the textbook, in R: `pt(-3.19,19)` and the corresponding probability for being above: `1-pt(-3.19,19)`.

6.1.2 The χ^2-Distribution, Week 8

- The numbers in Table 5, page 517 (7ed: 588) in the textbook are given by: `qchisq(`$1-\alpha$`,`ν`)` and the corresponding α-values by: `1-pchisq(x,`ν`)`, where x are the numbers in the table.
- The probability in the example page 190 (7ed: 219-220) in the textbook, in R: `1-pchisq(30.2,19)`

6.1.3 The F-Distribution, Week 8

- The numbers in Table 6, page 518-519 (7ed: 589-590) in the textbook are given by: `qf(`$1-\alpha$`,`ν_1`,`ν_2`)` and the corresponding α values by: `1-pf(x,`ν_1`,`ν_2`)`, where x are the numbers in the table.
- The probability 0.95 in the example page 221 in the the textbook, in S-PLUS: `1-pf(0.36,10,20)` or as `pf(2.77,20,10)`.

- It is possible the find the values in the table for $\alpha = 0.95$ by `qf(1-0.95,10,20)` or `1/qf(0.95,20,10)`.

6.2 Test-Assignments

6.2.1 Exercise

Write down the equation and/or sketch the meaning of the following R functions and results:

```
> qt(0.975,17)
[1] 2.109816
> qt(0.975,1000)
[1] 1.962339
```

6.2.2 Exercise

Write down the equation and/or sketch the meaning of the following R function and result:

```
> pt(2.75,17)
[1] 0.993166
```

7 Hypothesis Tests and Confidence Intervals Concerning One and Two Means, Chapter 7, Week 6-7

7.1 Introduction

Look at the beginning of Appendix C in the textbook, specially 'Confidence Intervals and Tests of Means', page 531 (7ed: 612). The R function `t.test` can be used to test one and two mean values as described on page 531 (7ed: 612) in the textbook. The function can also handle paired measurements. The function performs both hypotheses test and calculates confidence intervals. As the name indicates, the function only performs tests based on the t-distribution, NOT a z-test. In real life problems, the t-test is most often the appropriate test to use. Also, when n is large enough to allow the use of the z-test, the results using the `t.test` are almost identical to the z-test. If the function is called with only one set of numbers, for example, `t.test(x)`, where x is a row of numbers, the function will automatically act as in Sections 7.2 and 7.6 in the textbook. The default is two sided test with $\alpha = 5\%$. If a one sited test and/or another level of significance is wanted, this should be stated in the function call, e.g.: `t.test(x,alt=''greater'',conf.level=0.90)`. Note that the level of significance is $= 1 - \alpha$.

If the function is called with two set of numbers, e.g.`t.test(x1,x2)`, where x1 is one row of numbers and x2 is another another row of numbers, the function will automatically act as in Section 8 in the book, that is consider the two rows of numbers as independent samples. The default is two sided test with $\alpha = 5\%$. If a one sited test and/or another level of significance is wanted, this should be stated in the function call, e.g.:
`t.test(x1,x2,alt=''less'',conf.level=0.90)`.

If the samples are paired, the function is called the same way, BUT an option is added to the call:: `t.test(x1,x2,paired=T)`. This gives exactly the same results as calling the function with the difference of the two set of numbers as: `t.test(x1-x2)`. Regarding one-sided/two-sided and level of significance, the same is valid as above.

If the function is called with the alternative (`alt=''greater''` or `alt=''less''`), another confidence interval it produced, a so called one-sided confidence interval. One-sided confidence intervals are NOT considered in the course.

7.1.1 One-Sample t-Test/Confidence Intervals

The results on top of page 210 can be achieved by:

1. Import `C2sulfur.dat` (using the file-menu). Call it (e.g.) `sulfur`.
2. Attach the data-set: `attach(sulfur)`.
3. Use the function: `t.test(emission,conf.level=0.99)`.

Note that the mean value and the variance are calculated incorrectly in the book! Note also that a a two sited t-test for the hypotheses $\mu = 0$ is always reported in the output whether you need it or not. In this case the test is NOT of interest.

7.1.2 Two-Sample t-Test/Confidence Intervals

The results of the example on page 254-255 (7ed: 266-267) can be achieved by:

1. Import `C2alumin.dat` (using the file-menu). Name it (e.g.) `alumin`.
2. Attach the data-set: `attach(alumin)`.

In this example, the data is stored in a typical (and sensible) way. However, the way it is stored makes it a bit difficult to use the `t.test` function. The strength measurements for the two alloys are stored in one variable `strength` and then there is another variable `alloy`, that is used to identify the measurements as coming from alloy1 or alloy2. That is, the variable `alloy` consists of 85 1's and 2's. New variables `x1` and `x2`, can be constructed using:

```
x1=strength[alloy==1]
x2=strength[alloy==2]
```

Now `x1` has values for `alloy1` and `x2` for `alloy2`. Now the results from page 255 (7ed: 267) can be achieved by calling the function as described above: `t.test(x1,x2)`.

It is also possible to use the data as it is using the menu. Before you do this, you have to tell R that the alloy-variable is a group variable. This can be done with the command: `C8alloy$alloy=factor(C8alloy$alloy)` or by the menus:'Data'→'Manage Data in active Data set'→'Convert numeric variables to factors' and select `alloy` (choose some variable names, e.g. 1 and 2). Now you are ready to do the statistical analysis be using the menus: 'Statistics'→'Means'→'Independent Samples t-test..'. Note that it is also possible to carry out one-sample calculations through the menu.

If "alloy" is a factor variable, then you can run the `t.test` function directly as: `strength alloy,data=C8alloy`, where the use of the "tildesign ("~") means that strength is a function of alloy

7.1.3 Paired t-test/confidence interval:

No further comments.

7.2 Self-Training Using Exercises from the Textbook

In most of the exercises, the distribution functions (as described earlier) can be used in stead of looking up in the tables (the t-table or the z-table).
Using the `t.test` function (and/or the menus) the raw data needs to be available - that is only so in some of the exercises:

- Solve exercise 7.61 (Data from exercise 2.41: Import "2-41.TXT"). (7ed: 7.42)
- Solve exercise 7.63 and 7.64. The data can easily be entered into the program: `x=c(14.5,14.2,14.4,14.3,14.6)`. (7Ed: 7.48 and 7.49)
- Solve exercise 8.21 (Import "8-21.TXT"). (7Ed: 7.72)
- Solve exercise 8.10 and 8.11. (Data can easily be entered into the program)(7Ed: 7.68 and 7.69)

7.3 Test-Assignments

7.3.1 Exercise

We have the following R commands and results:

```
> x=c(10,13,16,19,17,15,20,23,15,16)
> t.test(x,mu=20,conf.level=0.99)

        One-sample t-Test

data:  x
t = -3.1125, df = 9, p-value = 0.0125
alternative hypothesis:  mean is not equal to 20
99 percent confidence interval:
 12.64116 20.15884
sample estimates:
 mean of x
      16.4
```

Write down the null and the alternative hypothesis, α and n corresponding to this output. What is the estimated standard error of the mean value? What is the maximum error with 99% confidence? (To answer the last question, the following can be used:)

```
> qt(0.995,9)
[1] 3.249836
> qt(0.975,9)
[1] 2.262157
> qt(0.95,9)
[1] 1.833113
```

7.3.2 Exercise

We have the following R commands and results:

```
> x1=c(10,13,16,19,17,15,20,23,15,16)
> x2=c(13,16,20,25,18,16,27,30,17,19)
> t.test(x1,x2,alt=''less'',conf.level=0.95,var.equal = TRUE))
         Two Sample t-test

data:  x1 and x2
t = -1.779, df = 18, p-value = 0.04606
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
        -Inf -0.09349972
sample estimates:
mean of x mean of y
     16.4      20.1
```

Write down the null and the alternative hypothesis, α, n_1 and n_2 corresponding to this output. What is the estimated standard error of the difference between the mean values? What R command would you use to find the critical vale for the hypothesis used?

7.3.3 Exercise

We have the following commands and results:

```
> x1=c(10,13,16,19,17,15,20,23,15,16)
> x2=c(13,16,20,25,18,16,27,30,17,19)
> t.test(x1,x2,paired=T,alt=''less'',conf.level=0.95)

        Paired t-test

data:  x1 and x2
t = -5.1698, df = 9, p-value = 0.0002937
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -2.388047
sample estimates:
mean of the differences
                  -3.7
```

Write down the null and the alternative hypothesis, α, n_1 and n_2 for this output. What is the estimated standard error of the difference in the mean values? What R command would you use to find the critical value for this hypothesis?

8 Hypothesis Test and Confidence Intervals for Proportions, Chapter 9, Week 9

8.1 Description

As described in appendix C, page 531 (7ed: 612), two R functions are available: `prop.test` and `chisq.test` (there are more relevant functions but they will not be considered here).

These functions can be used for hypotheses testing and confidence intervals for proportions.

8.1.1 Confidence Intervals for Proportions, Section 10.1

The 95% confidence interval on page 280 (7ed: 295) can be achieved by running:
`prop.test(36,100)`. The result is a bit different from the book. The reason is that R uses a so called continuous-correction such as the one used to approximate the binomial distribution with the normal distribution on page 132 (7ed: 160). It is possible to turn this is off by writing:
`prop.test(36,100,correct=F)`. The results are still a bit different from the book since R uses another approximation that makes the interval similar to an exact interval, given in Table 9 in the textbook. We will NOT consider the details here.

8.1.2 Hypotheses Concerning Proportions, Section 10.2

The results in the example page 299 in the textbook can be achieved by running:
`prop.test(48,60,p=0.7,correct=F,alternative=''greater'')`
Note that we do NOT get a Z-test but a χ^2-test in stead. Note that the following relationship is valid: $Z^2 = \chi^2$.
If the function is called with the alternative (`alt=''greater''` or `alt=''less''`), another confidence interval it produced, an so called one-sided confidence interval. One-sided confidence intervals are NOT considered in the course.

8.1.3 Hypothesis Concerning One or Two Proportions, Section 10.3

The results from the example page 286-287 (7ed: 302) (the example used on page 531 (7ed: 612)) can be achieved by running:

```
crumbled=c(41,27,22)
intact=c(79,53,78)
prop.test(crumbled,crumbled+intact)
```

It is also possible to use the function `chisq.test` and run
`chisq.test(matrix(c(crumbled,intact),ncol=2))`

Note that the R notation is a bit different from the R-notation, shown in the textbook page 531 (7ed: 612).

8.1.4 Analysis of $r \times c$ Tables, Section 10.4

The results from the exercise page 295 (7ed: 310) can be achieved by running:

```
poor=c(23,60,29)
ave=c(28,79,60)
vgood=c(9,49,63)
chisq.test(matrix(c(poor,ave,vgood),ncol=3))
```

If the data is on raw format, as the exam-data used in the introductory example, it is possible to use the menu bar to get cross-tabulations and χ^2-test to test for possible relationships: 'Statistics'→'Contingency Tables'→'Two-way table'

8.2 Self-Training Using Exercises from the Book

For most of the exercises, the distribution functions can be used in stead of looking up in the tables (the z- or the χ^2 distribution), as described earlier. The following exercises can be solved using either `prop.test` or `chisq.test`:

- Solve exercise 10.1 (7Ed: 9.1)
- Solve exercise 10.28 (7Ed: 9.28)
- Solve exercise 10.29 (7Ed: 9.29)
- Solve exercise 10.40 (7Ed: 9.40)
- Solve exercise 10.41 (7Ed: 9.41)

8.3 Test-assignments

No text-assignments since this part is only for orientation, or 'o' in the reading plan.

9 Simulation based statistical methods, Week 10

9.1 Introduction

One of the really big gains for statistics and modeling of random phenomena provided by computer technology during the last decades is the ability to simulate random systems on the computer. This provides the ability to calculate things that otherwise from a mathematical analytical point of view would be impossible to find. And even in cases where the highly educated mathematician/physicist might be able to find solutions, the simulation tool is a general and simple calculation tool for all of us who do not have this theoretical insight.

The direct reason for going in that direction here in our course is the "missing link" in the situations covered in Chapter 7 and 8 (7Ed: Ch. 7). The situations covered by the book in these chapters are given in summary form in Table 8.1, page 267 (7Ed: Table 7.1). In short, it's here about the statistics in connection with the average(s) of one or two samples. If you look closely at what tools the table provides us with, it appears that as long as there is a large sample size ($n \geq 30$), then we have tools for what we want, because we can do hypothesis testing and confidence intervals using the normal distribution, which due to the central limit theorem (Chapter 6) is a good approximation to the relevant sampling distributions. When you're in situations with small sample sizes, then, Table 8.1 (and Chapters 7-8) make the additional assumption that the populations, where the data comes from MUST be normal distributions. So in practice you should try to ensure that the data you're analyzing behaves like a normal distribution: symmetric and bell-shaped histogram. In Chapter 5 we also learned that you can make a normal probability

plot to verify this assumption in practice, and possibly transform the data to get them to look as normal as possible. The problem with small samples is that it even with these diagnostic tools can be difficult to know whether the underlying distribution really is "normal" or not. And in many cases the assumption of normality after all simply may be obviuosly wrong. For example, when the response scale we work with is far from being quantitative and continuous - it could be a scale like "small", "medium" and "larg" - coded as 1, 2 and 3. We need a tool that can do statistical analysis for us WITHOUT the assumption that the normal distribution is the right model for the data we observe and work with.

The book cover the missing link by Chapter 14: Nonparametric Tests. And in previous versions of this course (until 2010) parts of this chapter have been the content of this week 10 lecture. Here the traditional so-called nonparametric statistical tests are treated. In short it is a collection of methods that make use of data at a more coarse level, typically by focusing on the rank of the observations instead of the actual values of the observations. So in a paired t-test situation, for example, one would just count how many times one is bigger than the other instead of calculating the differences. In that way you can make statistical tests without using the assumption of an underlying normal distribution. There are a large number of such non-parametric tests for different situations. Historically, before the computer age, it was the only way to really handle this situation in practice. These tests are all characterized by the fact that they are given by relatively simple computational formulas which in earlier times easily could be handled.

The simulation based methods that we now present instead have several advantages to the tradional nonparametric methods:

- Confidence intervals are much easier to achieve
- They are much easier to apply in more complicated situations
- They better reflect today's reality: they are simply now used in many contexts

9.2 What is simulation really?

Basically, a computer obviously cannot create a result/number, which is random. A computer can give an output as a function of an input. (Pseudo) random numbers from a computer are generated from a specially designed algorithm - called a random number generator, which once started can make the figure x_{i+1} from the figure x_i. The algorithm is designed in such a way that when looking at a sequence of these figures, in practice one cannot tell the difference between these and a sequence of real random numbers. However, the algorithm needs a start input, called the "seed". The seed is typically generated by the computer using the inbuilt clock. Usually you can manage just fine without having to worry about the seed issue since the program itself finds out how to handle it appropriately. Only if you want to be able to recreate exactly the same results you need to save and set seed values - you can find `R` features for that also. For details about this and the random number generators used in `R`, type `?Random`.

We have already seen that `R` can generate random numbers from any of the distributions implemented in the program. The following are the relevant ones for this course:

`rbinom`	Binomial distribution
`rpois`	Poisson distribution
`rhyper`	The hypergeometric distribution
`rnorm`	normal distribution
`rlnorm`	log-normal distributions
`rexp`	exponential
`runif`	The uniform distribution
`rt`	t-distribution
`rchisq`	χ^2-distribution
`rf`	F distribution

Actually, a basic random number generator typically generates (pseudo) random numbers between 0 and 1 in the sense that figures in practice follows the uniform distribution on the interval 0 to 1, cf. Section 5.14, page 167-168 in the textbook (8th Ed). This means that regardless of which sub-interval we consider, the number of observations in the sub-interval will correspond to the width of the sub-interval. Actually, there is a simple way how to come from these to any kind of distribution: (cf. Figure 5.30, page 168)

If $U \sim$ Uniform(0.1) and F is a distribution function for any probability distribution, then $F^{-1}(U)$ follow the distribution given by F

Recall that the distribution function F in `R` is given by the `p` versions of the distributions, while R^{-1} is given by the `q` versions. But since `R` has already done this for us, we do not really need this as long as we only use distributions that have already been implemented in `R`. One can use the help function for each function, for example. `?rnorm`, to check exactly how to specify the parameters of the individual distributions. The syntax follows 100% what is used in `p`, `d` and `q` versions of the distributions.

9.2.1 Example

We can generate 100 normally distributed $N(2, 3^2)$ numbers by `rnorm(100,mean=2,sd=3)`. The same could be achieved with `qnorm(runif(100),mean=2,sd=3)`.

9.3 Simulation as a general computational tool

Basically, the strength of the simulation tool is that one can compute arbitrary functions of random variables and their outcomes. In other words one can find probabilities of complicated outcomes. As such the tool is really not a statistical one, but rather a probability calculus tool. But since statistics essentially is about analysing and learning from real data in the light of certain probabilities, the simulation tool indeed becomes of statistical importance, which we will exemplify very specifically below. Let us first exemplify the power of simulation as a general computational tool.

9.3.1 Example

A company produces rectangular plates. The length of plates (in meters), X is assumed to follow a normal distribution $N(2, 0.1^2)$ and the width of the plates (in meters), Y are assumed to follow

a normal distribution $N(3, 0.2^2)$. We're hence dealing with plates of size 2×3 meters but with errors in both length and width. Assume that these errors are completely independent. We are interested in the area of the plates which of course is given by $A = XY$. This is a nonlinear function of X and Y, and actually it means that with the tools we learn in the current course, we cannot figure out what the mean area really is, and not at all what the standard deviation would be in the areas from plate to plate, and we would definitely not know how to calculate the probabilities of various possible outcomes. For example, how often such plates have an area that differ by more than $0.1m^2$ from the targeted $6m^2$? One statement summarizing all our lack of knowledge is: we do not know the probability distribution of the random variable A and we do not know how to find it! With simulation, it is straightforward: One can find all relevant information about A by just simulating the X and Y a high number of times, and from this compute A just as many times, and then observe/record what happens to the values of A. The first step is then given by:

```
k = 10000 # Number of simulations
X = rnorm(k, 2,0.1)
Y = rnorm(k, 3,0.2)
A = X*Y
```

The `R` object `A` now contains 10.000 observations of A. The expected value and the standard deviation for A are simply found by calculating the average and standard deviation for the simulated A-values:

```
mean(A)
[1] 5.999061
sd(A)
[1] 0.5030009
```

And the desired probability, $P(|A - 6| > 0.1) = 1 - P(5.9 \leq A \leq 6.1)$ is found by counting how often the incident actually occurs among the k outcomes of A:

```
sum(abs(A-6)>0.1)/k
[1] 0.8462
```

The code `abs(A-6)>0.1` creates a vector with values `TRUE` or `FALSE` depending on whether the absolute value of $A - 6$ is greater than 0.1 or not. When you add (sum) these the `TRUE` is automatically translated into 1 and `FALSE` automatically set to 0, by which the desired count is available, and divided by the total number of simulations k. Note that if you do this yourself will not get exactly the same result as the seed value in your specific run will be different than the one used here. It is clear that this simulation uncertainty is something we must deal with in practice. The size of this will depend on the situation and on the number simulations k. You can always get a first idea of it in a specific situation simply by repeating the calculation a few times and note how it varies. Indeed, one could then systematize such an investigation and repeat the simulation many times to get an evaluation of the simulation uncertainty. We will not formalize this here. When the target of the computation is in fact a probability, as in the latter example here, you can alternatively use standard binomial statistics, which are covered in Chapter 4 and 10. For example, with $k = 100000$ the uncertainty for a calculated proportion of around 0.85 given by: $\sqrt{\frac{0.85(1-.85)}{100000}} = 0.0011$. Or for example, with $k = 10000000$ the uncertainty is

0.00011. The result using such a k was 0.8414536 and because we're a bit unlucky with the rounding position we can in practice say that the exact result rounded to 3 decimal places are either 0.841 or 0.842. In this way, a calculation which is actually based on simulation is turned into an exact one in the sense that rounded to 2 decimal places, the result is simply 0.84.

9.4 Propagation of error

Within chemistry and physics one speaks of measurement errors and how measurement errors propagate/accumulate if we have more measurements and/or use these measurements in subsequent formulas/calculations. First of all: The basic way to "measure an error", that is, to quantify a measurement error is by means of a standard deviation.The standard deviation expresses, as we know, the average deviation from the mean. It is clear it may happen that a measuring instrument also on average measures wrongly (off the target). This is called "bias", but in the basic setting here, we assume that the instrument has no bias. An error propapgation problem is thus reformulated a questions about how the standard deviation of some function of the measurements depends on the standard deviations for the individual measurements: Let $X_1, \ldots, X_n$ be n measurements with standard deviations (measurement errors) $\sigma_1, \ldots, \sigma_n$. For everything going on in this course, we assume that these measurement errors are independent of each other. There are extensions of the formulas that can handle dependices, but we omit those here. We must then in a general formulation be able to find:

$$\sigma^2_{f(X_1,\ldots,X_n)} = \text{Var}(f(X_1, \ldots, X_n)) \tag{1}$$

Actually, we have already in this course seen the linear error propagation rule, which is expressed in the box on page 154 in Chapter 5 of the book (8Ed):

$$\sigma^2_{f(X_1,\ldots,X_n)} = \sum_{i=1}^{n} a_i^2 \sigma_i^2, \text{if} f(X_1, \ldots, X_n) = \sum_{i=1}^{n} a_i X_i$$

There is a more general non-linear extension of this, albeit theoretically only an approximate result, which involves the partial derivative of the function f with respect to the n variables:

$$\sigma^2_{f(X_1,\ldots,X_n)} = \sum_{i=1}^{n} \left(\frac{\partial f}{\partial X_i} \right)^2 \sigma_i^2 \tag{2}$$

In practice we insert the actual measurement values of $X_1, \ldots, X_n$ in the partial derivatives. This is a pretty powerful tool for the general finding of (approximate) uncertainties for complicated functions of many measurements or for that matter: complex combinations of various statistical quantities. When the formula is used for the latter, it is also in some contexts called the "delta rule" (which is mathematically speaking a so-called first-order (linear) Taylor approximations to the nonlinear function f). We bring it forward here, because as an alternative to this approximate formula one could use simulation in the following way:

Simulate k outcomes of all n measurements as $N(X_i, \sigma_i^2)$: $X_i^{(j)}, j = 1 \ldots, k$
Calculate the standard deviation directly as the observed standard deviation of the k values for f:

$$\sigma_{f(X_1,\ldots,X_n)} = \sqrt{\frac{1}{k-1} \sum_{i=1}^{k} (f_j - \bar{f})^2}$$

$$f_j = f(X_1^{(j)}, \ldots, X_n^{(j)})$$

9.4.1 Example

Let us continue the example with $A = XY$ and X and Y defined as in the example above. To use the approximate error propagation rule, we must differentiate the function $f(x, y) = xy$ with respect to both x and y:

$$\frac{\partial f}{\partial x} = y \quad \frac{\partial f}{\partial y} = x$$

With two specific measurements of X and Y, for example $X = 2.05m$ and $y = 2.99m$ the error propagation law would provide the following approximate calculation of the variance of $A = 2.05 \times 2.99 = 6.13$:

$$\sigma_A^2 = y^2 \times 0.1^2 + x^2 \times 0.2^2 = 2.99^2 \times 0.1^2 + 2.05^2 \times 0.2^2 = 0.2575$$

So with the error propagation law we could manage a part of the challenge without simulating. Actually we are pretty close to be able to find the correct variance of $A = XY$ using tools provided in this course. For by definition and the following fundamental relationship: (which IS a part of the course syllabus)

$$\text{Var}(X) = \text{E}(X - \text{E}(X))^2 = \text{E}(X^2) - \text{E}(X)^2$$

So one can actually deduce the variance of A theoretically, only you must in addition know that for independent random variables: $\text{E}(XY) = \text{E}(X)\text{E}\,(Y)$: (Which by the way then also tells us that $\text{E}(A) = \text{E}(X)\text{E}(Y) = 6$)

$$\begin{aligned}
\text{Var}(XY) &= \text{E}\left[(XY)^2\right] - [\text{E}(XY)]^2 \\
&= \text{E}(X^2)\text{E}(Y^2) - \text{E}(X)^2\text{E}(Y)^2 \\
&= \left[\text{Var}(X) + \text{E}(X)^2\right]\left[\text{Var}(Y) + \text{E}(Y)^2\right] - \text{E}(X)^2\text{E}(Y)^2 \\
&= \text{Var}(X)\text{Var}(Y) + \text{Var}(X)\text{E}(Y)^2 + \text{Var}(Y)\,\text{E}(X)^2 \\
&= 0.1^2 \times 0.2^2 + 0.1^2 \times 3^2 + 0.2^2 \times 2^2 \\
&= 0.0004 + 0.09 + 0.16 \\
&= 0.2504
\end{aligned}$$

Note how the approximate error propagation rule actually corresponds to the two latter terms in the correct variance, while the first term - the product of the two variances is ignored. Fortunately, this term is the smallest of the three in this case. It does not always have to be like that. A theoretical derivation of the density function for $A = XY$ could be done if you take the course 02405 on Probability.

9.5 Confidence intervals using simulation: Bootstrapping

Generally, a confidence interval for an unknown parameter μ is a way to express uncertainty using the sampling distribution of $\hat{\mu} = \bar{x}$. Hence, we use a distribution that expresses how our calculated value would vary from sample to sample. As indicated, we have so far no method to do this if we only have a small sample size ($n < 30$), and the data cannot be assumed to follow a normal distribution. In principle there are two approaches for solving this problem:

1. Find/identify/assume a different and more suitable distribution for the population ("the system")

2. Do not assume any distribution whatsoever

The simulation method called bootstrapping, which in practice is to simulate many samples, exists in two versions that can handle either of these two challenges:

1. Parametric bootstrap: Simulate multiple samples from the assumed distribution.

2. Non-parametric bootstrap: Simulate multiple samples directly from the data.

Actually the parametric bootstrap handles in addition the situation where data could perhaps be normally distributed, but where the calculation of interest is quite different than the average, for example, the coefficient of variation (standard deviation divided by average). This would be an example of a nonlinear function of data thus not having a normal distribution as a sampling distribution. And the parametric bootstrap is basically just an example of the use of simulation as a general calculation tool, as introduced above. Both methods are hence very general and can be used in virtually all contexts. However, we will below only give detailed methods for our one and two-sample situations, focusing on average values - and only for the non-parametric bootstrap, because we do not in the course have so much focus on statistics using alternative distributions for continuous quantitative data. We have met a few of these alternative distributions, eg. log-normal, uniform and exponential distribution, but have not really learned how to do "classical" (small sample) statistics for data coming from such distributions. The parametric bootstrap is a way to do this without relying on theoretical derivations of things.

9.5.1 Non-parametric bootstrap for the one-sample situation

We have the random sample (the data): $x_1, \ldots, x_n$. The $100(1-\alpha)\%$ confidence interval for μ determined by the non-parametric bootstrap is defined as:

Simulate k samples of size n by randomly sampling among the available data
(with replacement - large k, e.g. $k > 1.000$)
Calculate the average in each of the k samples $\bar{x}_1^*, \ldots, \bar{x}_k^*$
Calculate the $100\alpha/2\%$ - and $100(1-\alpha/2)\%$ percentiles for these
The confidence interval is: $\left[\text{quantile}_{100\alpha/2\%}, \text{quantile}_{100(1-\alpha/2)\%}\right]$

There are other versions of the specific construction of the confidence interval than this one, but this is the most direct and follows a principle that can easily be generalized to other situations.

Example
In a study women's cigarette consumption before and after giving birth is explored. The following observations of the number of smoked cigarettes per day were the results:

before	after	before	after
8	5	13	15
24	11	15	19
7	0	11	12
20	15	22	0
6	0	15	6
20	20		

This is a typical paired t-test setup, as discussed in Chapter 8.4, which then was handled by finding the 11 differences and thus transforming it into a one-sample situation, which was dealt with in chapter 7.2. You get the data into R and calculate the differences by the following code:

```
x1 = c(8,24,7,20,6,20,13,15,11,22,15)
x2 = c(5,11,0,15,0,20,15,19,12,0,6)
dif = x1-x2
dif
 [1] 3 13 7 5 6 0 -2 -4 -1 22 9
```

There is a random-sampling function in R (which again is based on a uniform random number generator): sample. Eg. you can get 5 repeated samples (with replacement - replace=TRUE) by:

```
> t(replicate(5,sample(dif,replace=TRUE)))
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
[1,]    6    0    6    3    5   -2    5    9   -2     7     6
[2,]   22   22    3    9   13    5   -2    0   13     3     3
[3,]    9    3    3    7    6    6    7   -2    5    -2     3
[4,]   13    5    9   22   13    9   13   13    5     6     6
[5,]    9   -2   -1    6    3   -4   -1   -4    9    -2     3
```

Explanation: replicate is a function that repeats the call to sample - in this case 5 times. The function t simply transposes the matrix of numbers, making it 5×11 instead of 11×5(only used for showing the figures in slightly fewer lines than otherwise necessary)
One can then run the following to get a 95% confidence interval for μ based on $k = 10.000$:

```
k = 10000
mysamples = replicate(k, sample(dif, replace = TRUE))
mymeans = apply(mysamples, 2, mean)
quantile(mymeans, c(0.025,0.975))
    2.5% 97.5%
1.363636 9.727273
```

Explanation: The sample function is called 10.000 times and the results collected in an 11×10.000 matrix. Then in a single call the 10.000 averages are calculated and subsequently the relevant percentiles found. Actually, there is a bootstrap package in R called bootstrap which includes a function (also) called bootstrap. First install this package (click the Packages $\rightarrow$ "Install Packages" and find the package on the list, or easier: simply type: install.packages(''bootst
Then load the package:

```
library(bootstrap)
```

Now the calculation can be performed in a single call:

```
quantile(bootstrap(dif,k,mean)$thetastar,c(0.025,0.975))
    2.5% 97.5%
1.361364 9.818182
```

This bootstrap function is advantageous to use when looking for confidence intervals for more complicated functions of data.

9.5.2 Two-sample situation

We now have two random samples: $x_1, \ldots, x_{n_1}$ and $y_1, \ldots, y_{n_2}$ The $100(1-\alpha)\%$ confidence interval for $\mu_1 - \mu_2$ determined by the non-parametric bootstrap is defined as:

Simulate k sets of 2 samples of size n_1 and n_2 by sampling randomly from the respective groups (with replacement - large k, eg.$k > 1.000$)
Calculate the difference between the averages for each of the k sample pairs: $\bar{x}_1^* - \bar{y}_1^*, \ldots, \bar{x}_k^* - \bar{y}_k^*$
Calculate the $100\alpha/2\%$ - and $100(1-\alpha/2)\%$ percentiles for these
The confidence interval is: $\left[\text{quantile}_{100\alpha/2\%}, \text{quantile}_{100(1-\alpha/2)\%}\right]$

Example
In a study it was explored whether children who received milk from bottle as a child had worse or better teeth health conditions than those who had not received milk from the bottle. For 19 randomly selected children is was recorded when they had their first incident of caries:

bottle	age	bottle	age	bottle	Age
no	9	no	10	yes	16
yes	14	no	8	yes	14
yes	15	no	6	yes	9
no	10	yes	12	no	12
no	12	yes	13	yes	12
no	6	no	20		
yes	19	yes	13		

One can then run the following to obtain a 95 % confidence interval for $\mu_1 - \mu_2$ based on $k = 10.000$:

```
x = c(9,10,12,6,10,8,6,20,12) # no group
y = c(14,15,19,12,13,13,16,14,9,12) # yes group

k = 10000 # Number of bootstrap samples
xsamples = replicate(k, sample (x, replace = TRUE)) # Sample of no group
ysamples = replicate(k, sample (y, replace = TRUE)) # Sample of yes-group
mymeandifs = apply(xsamples, 2, mean)-apply(ysamples, 2, mean) # Calculating the diff
quantile(mymeandifs, c(0.025,0.975)) # percentiles
      2.5% 97.5%
-6.2222222 -0.1777778
```

9.6 Hypothesis testing using simulation

We shall see two ways in which we can do hypothesis testing through simulation.

9.6.1 Hypothesis testing using bootstrap confidence intervals

Hypotheses that can be formulated by a single parameter - or a direct relationship between two parameters can be tested using the usual relationship between confidence intervals and hypothesis testing - here attempted expressed in general terms:

$$H_0 : \theta = \theta_0 \text{ accepted} \Leftrightarrow \theta_0 \text{ is in the confidence interval for } \theta$$

If you then use the non-parametric bootstrap-based confidence interval as a criterion, then you automatically have a simulation-based hypothesis testing procedure. A little wrinkle here is that we in this course usually only works with 2-sided confidence intervals, which then can provide a 2-sided hypothesis test. Although we would otherwise not operate with 1-sided confidence intervals, one can define correspondingly a one-sided hypothesis test using the bootstrap in the obvious way, eg.:

$$H_0 : \theta = \theta_0 \text{ versus } H_1 : \theta > \theta_0 \text{ accepted} \Leftrightarrow$$

$$\theta_0 > 100\alpha\%\text{-percentile of the bootstrap values for } \theta$$

9.6.2 One-sample setup, Example

We continue the cigarette consumption example. We would now like to show that cigarette consumption has decreased after giving birth. We would therefore like to perform the one-sided hypothesis test: $H_0 : \mu_1 - \mu_2 = 0$ versus the alternative: $H_1 : \mu_1 - \mu_2 > 0$. Taking the bootstrap confidence interval above we can see so much as, since 0 is outside the interval, we reject the hypothesis when we test at level $\alpha = 0.025$. You can also find the P-value by observing, where 0 is in the bootstrap sample distribution - in other words, how often it happens that a difference becomes less than 0:

```
sum(mymeans<0)/k
[1] 0.0022
```

The P-value is therefore around 0.002, so a relatively clear indication that cigarette consumption actually decreased.

9.6.3 Hypothesis testing using permutation tests

There are situations where you cannot easily express the relevant hypothesis in a meaningful univariate scale where a confidence interval can do the job for us. It is typically when testing hypotheses involving more than 2 parameters. We have seen an example of this in relation to $r \times c$ tables in chapter 10, and we will see some more in Chapter 12, where we extend exactly the two-group situations from Chapters 7 and 8, which we are focussing on here to multiple group setups. The permutation test idea is in all its simplicity, that we basically "shake the bag" and see what happens. More precisely, you try many times to draw lots on which groups the individual data points belong to, and then calculate the test statistic each time. If the actually observed test statistic, which measures the group difference, is unusually large compared to the many simulated versions, then the hypothesis is rejected. In practice each new simulated situation contains exactly the same data points as the original dataset, but differently grouped. We also say that we have permuted the data on the groups. Or again in other words, we have sampled without replacement.

9.6.4 Two-sample situation

We now have the two samples: $x_1, \ldots, x_{n_1}$ and $y_1, \ldots, y_{n_2}$ A permutation test for the hypothesis $\mu_1 = \mu_2$ is defined by:

Simulate k sets of 2 samples of size n_1 and n_2 by permuting the available data
(Large k, eg. $k > 1.000$)
Calculate the difference between the averages for each of the k sample pairs: $\bar{x}_1^* - \bar{y}_1^*, \ldots, \bar{x}_k^* - \bar{y}_k^*$
Find the P-value from the position of $\bar{x} - \bar{y}$ in this distribution
(2-sided or 1-sided - in the usual manner)

Example
We continue the tooth health example. We want to perform a two-sided test for the hypothesis: $\mu_1 = \mu_2$. The following R-code implements the calculations:

```
x = c(9,10,12,6,10,8,6,20,12) # no group
y = c(14,15,19,12,13,13,16,14,9,12) # yes group

 k = 100000
perms = replicate(k,sample(c(x,y)))
mymeandifs = apply(perms[1:9,], 2, mean)-apply(perms[10:19,], 2, mean)
sum(abs(mymeandifs)> abs(mean(x)-mean(y)))/k
[1] 0.05132
```

Explanation: First, set a large k, since the P-value turns out to be around 5%. Next we run `sample(c(x,y))` that permutes the 19 observations. In `mymeandifs` the 100.000 sets of the difference between the average of the first 9 the last 10 numbers. In the last line the counting of how often these simulated differences in absolute terms are larger than the actual observed difference in the data is carried out.

The observant reader might have thought about a little curiousity in what we do here: in fact we can calculate exactly how many different permutations that actually exist in this case. In the example here it is the number of different samples of 9 you can take out of 19. That is exactly the so-called binomial coefficient:

$$\text{number of different permutations} = \left(\begin{array}{c} 19 \\ 9 \end{array} \right) = \frac{19!}{(10!)(9!)} = 92378$$

In R this figure can for example be found as:

```
factorial(19)/(factorial(10)*factorial(9))
[1] 92378
```

There are actually only a total of 92.378 different possible outcomes for the difference between the two averages. A bit more intelligent approach would simply be to list all of them exactly once and then do the count based on those instead. It would provide a so-called exact P-value for the permutation test. Sometimes the number of different permutations is unrealistically high, and thus we turn to the simulations instead. And the R-technical implementation of the test is actually easier by the simulation here, so in this course we will stick to the simulation version described in the box above.

9.7 Exercises

For this subject there are only exercises here in this note. Most are made with the premise that t themselves must have R running to solve them. Even if you do not ACTUALLY run the stuff

in R you CAN still consider and describe HOW you would do it - symbolically - step by step. For the final exam the assignments within this topic will be made in a manner so as to NOT be required to actually use the `R` in the exam room. Experience and insight obtained by solving these assignments will make one able to solve any. exam papers (jointly with the review above, obviously)

9.7.1 Exercise

(Simulation as a computation tool). A system consists of three components A, B and C serially connected such that A is positioned before B again positioned before C. The system will function only so long as A, B and C all function. The lifetime in months of the three components are assumed to follow exponential distributions with means 2 months, 3 months and 5 months, respectively.

1. Generate, by simulation, a large number (at least 1000) of system lifetimes.
2. Estimate the mean system lifetime.
3. Estimate the standard deviation of system lifetimes.
4. Estimate the probability that the system fails within 1 month.
5. Estimate the median system lifetime
6. Estimate the 10th percentile of system lifetimes
7. What seems to be the distribution of system lifetimes? (histogram etc)

9.7.2 Exercise

(Non-linear error propagation). The pressure P, and the volume V of one mole of an ideal gas are related by the equation $PV = 8.31T$, when P is measured in kilopascals, T is measured in kelvins, and V is measured in liters.

1. Assume that P is measured to be 240.48kPa and V to be 9.987L with known measurement errors (given as standard deviations): 0.03kPa and 0.002L. Estimate T and find the uncertainty in the estimate.
2. Assume that P is measured to be 240.48kPa and T to be 289.12K with known measurement errors (given as standard deviations): 0.03kPa and 0.02K. Estimate V and find the uncertainty in the estimate.
3. Assume that V is measured V to be 9.987L and T to be 289.12K with known measurement errors (given as standard deviations): 0.002L and 0.02K. Estimate P and find the uncertainty in the estimate.
4. Try to answer one or more of these questions by simulation (assume that the errors are normally distributed)

9.7.3 Exercise

(Can be handled without using R) The following measurements were given for the cylindrical compressive strength (in MPa) for 11 prestressed concrete beams: 38.43, 38.43, 38.39, 38.83, 38.45, 38.35, 38.43, 38.31, 38.32, 38.48, 38.50. 1000 bootstrap samples (each sample hence consisting of 11 measurements) were generated from these data, and the 1000 bootstrap means were arranged on order. Refer to the smallest as $\bar{x}^*_{(1)}$, the second smallest as $\bar{x}^*_{(2)}$ and so on, with the largest being $\bar{x}^*_{(1000)}$. Assume that $\bar{x}^*_{(25)} = 38.3818$, $\bar{x}^*_{(26)} = 38.3818$, $\bar{x}^*_{(50)} = 38.3909$, $\bar{x}^*_{(51)} = 38.3918$, $\bar{x}^*_{(950)} = 38.5218$, $\bar{x}^*_{(951)} = 38.5236$, $\bar{x}^*_{(975)} = 38.5382$, and $\bar{x}^*_{(976)} = 38.5391$.

1. Consider why it may be questionable to use the t-distribution based confidence interval method for these data! (Look at the data, e.g. Plot the data - e.g. a box plot)

2. Compute a 95% bootstrap confidence interval for the mean compressive strength.

3. Compute a 90% bootstrap confidence interval for the mean compressive strength.

9.7.4 Exercise

Consider the data from the exercise above. These data are entered into R as:

```
x=c(38.43, 38.43, 38.39,38.83, 38.45,38.35,38.43,38.31,38.32,38.48,38.50)
```

Now generate 1000 bootstrap samples and compute the 1000 means.

1. What is the 2.5%, and 97.5% percentiles?

9.7.5 Exercise

A TV producer had 20 consumers evaluate the quality of two different TV flat screens - 10 consumers for each screen. A scale from 1 (worst) up to 5 (best) were used and the following results were obtained:

TV screen 1	TV screen 2
1	3
2	4
1	2
3	4
2	2
1	3
2	2
3	4
1	3
1	2

1. Carry out the test of the null hypothesis that TV screen2 has a quality of at most 2.5 versus the alternative of a larger than 2.5 quality. Use $\alpha = 0.01$. (Use the bootstrap approach)

2. Carry out the test of the hypothesis that the two TV screens have the same quality. Use $\alpha = 0.05$.

(a) By the bootstrap confidence interval method

(b) By the permutation test method.

3. How many different permuations are there really?

4. Compare the results with using t-test procedures!

10 Linear Regression, Chapter 11, Week 11

10.1 Introduction

Look at Appendix C in the textbook, specially 'Regression' page 513 (7ed: 613). Data relevant for this section can be downloaded from:
`http://www2.imm.dtu.dk/courses/02402/Bookdata8ED`
or if you work with 7. edition of the book:
`http://www2.imm.dtu.dk/courses/02402/Bookdata8`
and imported the usual way. We will use the example page 303, 305, 306, 311, 312, 315 (7ed: 341, 347, 349, 351). The data can be downloaded from:
`http://www2.imm.dtu.dk/courses/02402/Bookdata8/C11evap.dat` We will assume that the data is stored as `C11evap`:

```
> C11evap
   velocity evap
 1       20 0.18
 2       60 0.37
 3      100 0.35
 4      140 0.78
 5      180 0.56
 6      220 0.75
 7      260 1.18
 8      300 1.36
 9      340 1.17
10      380 1.65
```

We can plot the relationship using:

```
> attach(C11evap)
> plot(evap ~ velocity)
```

The basic regression function described here is `lm`. We fit the line and store the results of the calculations using:

```
> fit.evap <- lm(evap ~ velocity)
```

and as described on page 613 in the textbook the results are summarized using:

```
> summary(fit.evap)

Call:
lm(formula = evap ~ velocity)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.20103 -0.14671  0.05261  0.12318  0.17473

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.0692424  0.1009737   0.686    0.512
velocity    0.0038288  0.0004378   8.746 2.29e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1591 on 8 degrees of freedom
Multiple R-squared: 0.9053,     Adjusted R-squared: 0.8935
F-statistic: 76.49 on 1 and 8 DF,  p-value: 2.286e-05
```

In the output, the parameter estimates are given, their standard error along with a t-test for whether they are equal to zero or not (equivalent to the boxes on page 310-311 (7ed: 346)). Estimates for s_e og R^2 are also given (compare with results in the book).

Regression can also be done using the menus: 'Statistics'→'Fit models'→'Linear regression'. Or in a scatterplot as: 'graph'→'Scatterplot'.

10.2 Self-Training Using Exercises from the Book

Solve the following exercises: 11.4, 11.5 and possibly exercise 11.6 using R.

10.3 Test-Assignments

10.3.1 Exercise

If you run an analysis of the math exam score as a function of the math year score, from the exam data used in Section 2, the output is:

```
> attach(karakterer2004)
> summary(lm(Mat.Eks~Mat.Aars))


Call:
lm(formula = Mat.Eks ~ Mat.Aars)

Residuals:
      Min        1Q    Median        3Q       Max
-2.450505 -0.266329  0.009203  0.281145  2.181145

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4952     0.1750   14.26   <2e-16 ***
Mat.Aars      0.7194     0.0222   32.41   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4575 on 1553 degrees of freedom
Multiple R-squared: 0.4035,     Adjusted R-squared: 0.4031
```

```
F-statistic:  1051 on 1 and 1553 DF,  p-value: < 2.2e-16
```

Write down the model and give estimates for the regression line. Are these estimates different from zero? What is the correlation between the two scores? How big is the confidence interval for the slope? Finally, what is the upper quartile for the exam-scores.

11 Analysis of Variance, Sections 12.1 and 12.2, Week 12

11.1 Introduction

Look at Appendix C in the textbook, specially 'One-way Analysis of Variance (ANOVA)' page 532 (7ed: 613). The data used in this section can be downloaded from:
`http://www2.imm.dtu.dk/courses/02402/Bookdata8ED`
or if you have 7. edition of the book:
`http://www2.imm.dtu.dk/courses/02402/Bookdata`
and imported as usual. The data must be structured as shown in the example below, where one column consists the actual data while the second column indicates the group each observation belongs to.

```
 G       Material
 6.683     Gold
 6.681     Gold
 6.676     Gold
 6.678     Gold
 6.679     Gold
 6.661   Platin
 6.661   Platin
 6.667   Platin
 6.667   Platin
 6.664   Platin
 6.678     Glass
 6.671     Glass
 6.675     Glass
 6.672     Glass
 6.674     Glass
```

The data from the example page 363 (7ed: 408) can be downloaded from:
`http://www2.imm.dtu.dk/courses/02402/Bookdata8ED/C12tin.TXT`
Or if you have 7. edition of the book:
`http://www2.imm.dtu.dk/courses/02402/Bookdata/C12tin.dat`
It is assumed here that the data is stored as `C12tin`. As described on page 532 (7ed: 613) in the textbook, the analysis is done using the function `lm`:

```
> attach(C12tin)
> Lab <- factor(Lab)
> anova(lm(weight~Lab))
```

```
Analysis of Variance Table

Response: weight
          Df   Sum Sq   Mean Sq F value  Pr(>F)
Lab        3 0.013006 0.0043354  2.8097 0.05038 .
Residuals 44 0.067892 0.0015430
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The second command, `Lab <- factor(Lab)` forces the program to consider the laboratorium column as a grouping factor and NOT as an quantitative variable. In this case, the laboratoriums are identified with the numbers 1, 2, 3 and 4.
The result of the F-test are a bit different from the ones shown in the book. This is because rounded numbers are used in the book. The consequence in this case is that the p-value is a bit bigger than 5% but not below as stated in the book. This show that there is not an "evidence based" difference between a p-value equal to 4.9% and 5.1%!

Analysis of variance can be done by the menus:(When the Lab-variable is changed to a factor) 'Statistics'→'Means'→'One-way ANOVA'.

11.1.1 Supplement: General Analysis of Variance ("For Orientation")

In one-sided ANOVA the factor variable can be considered as an explanatory variable that can take a few number of values. A variable of this kind is called a categorical variable. Analysis of variance can also be performed where there are more than one categorical variables. Section 12.3 in the textbook gives an example of this.
More general analysis of variances can be carried out from the menus:
'Statistics'→'Means'→'Multiway ANOVA'.

11.2 Self-Training Using Exercises from the Textbook

- Solve exercise 12.10 using R
- Solve exercise 12.6 using R

Fill out an ordinary ANOVA table. Try to understand the number of degrees of freedom, the test statistic and the p-value (you can use `pf(q,df1,df2)`).

11.3 Test-Assignments

11.3.1 Exercise

Two analysis of the math year-grates (see Section 2) are performed. The R commands and results are:

```
> anova(lm(Mat.Eks~Kommune))
> anova(lm(Mat.Eks~Amt))
```

```
Analysis of Variance Table

Response: Mat.Eks

Terms added sequentially (first to last)
            Df Sum of Sq   Mean Sq  F Value      Pr(F)
  Kommune  269  106.4311 0.3956547 1.159594 0.05405999
Residuals 1285  438.4435 0.3412012


Analysis of Variance Table

Response: Mat.Eks

Terms added sequentially (first to last)
            Df Sum of Sq  Mean Sq  F Value        Pr(F)
      Amt   15   16.2862 1.085744 3.161173 3.822564e-05
Residuals 1539  528.5885 0.343462
```

Write down the hypothesis and give p-values for these. Try to interpret the results: is there a difference in relation to math-exam grate between `Kommune` and `Amt` respectively. How many 'kommuner' and 'amts' are included in the study? How big is the variation within `Kommune` and `Amt`?

12 Analysis of Variance, Section 12.3, Week 12

12.1 Introduction

The data used in this section should be on the same format as in the case of one-sided ANOVA but with an extra column vector that containing the 'block' information. For the example page 373-375 (7ed: 420-422) the data can be read into the program using:

```
example <- data.frame(y = c(13,7,9,3,6,6,3,1,11,5,15,5),
treatm = c(1,1,1,1,2,2,2,2,3,3,3,3),
block  = c(1,2,3,4,1,2,3,4,1,2,3,4))
```

that corresponds to the following structure:

```
> example
    y treatm block
 1 13      1     1
 2  7      1     2
 3  9      1     3
 4  3      1     4
 5  6      2     1
 6  6      2     2
 7  3      2     3
 8  1      2     4
 9 11      3     1
10  5      3     2
11 15      3     3
12  5      3     4
```

The analysis is performed the same way as the one-sided analysis of variance, but with the block factor added:

```
> attach(example)
> treatm <- factor(treatm)
> block <- factor(block)
> anova(lm(y~treatm+block))

Analysis of Variance Table

Response: y

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value     Pr(F)
   treatm  2        56 28.00000 3.230769 0.1116192
    block  3        90 30.00000 3.461538 0.0913831
Residuals  6        52  8.66667
```

This corresponds to the ANOVA table on page 422.

12.2 Self-training Using Exercises from the Textbook

- Solve exercise 12.47 using R (7ed: 12.50)

The data can either be entered in a spreadsheet and imported as usual or by using the following commands (use 'cut & paste' to get them into R):

```
dat.12.47 <- data.frame(conc.ppm = c(
                          23.8, 7.6, 15.4, 30.6, 4.2,
                          19.2, 6.8, 13.2, 22.5, 3.9,
                          20.9, 5.9, 14.0, 27.1, 3.0),
                        agency = rep(paste(''Agency'', 1:3), rep(5,3)),
                        site = rep(paste(''Site'', LETTERS[1:5]), 3))
```

12.3 Test-Assignments

12.3.1 Exercise

Consider the following R command and results: (these are breaking strength measurements (`strength`) for some threads (`thread`) using different instruments (`instrument`) from exercise 12.20)

```
> anova(lm(strength~thread+instrument))
Analysis of Variance Table

Response: strength

Terms added sequentially (first to last)
           Df Sum of Sq  Mean Sq  F Value     Pr(F)
    thread  4    70.173 17.54325 8.315979 0.0018781
instrument  3     0.330  0.11000 0.052143 0.9835259
 Residuals 12    25.315  2.10958
```

Write down the hypothesis and give p-values for these. Try to interpret the results: are the threads different and is there any difference between the instruments? How many kind of threads and instruments are included in the study? What is the standard deviation of the strength measurements after correcting for systematic difference between types of threads and instruments?