

Basic statistics using R

Jarno Tuimala (CSC)

Dario Greco (HY)

Day 1

Welcome and introductions

Learning aims

➤ **To learn R**

- Syntax
- Data types
- Graphics
- Basic programming (loops and stuff)

➤ **To learn basic statistics**

- Exploratory data analysis
- Statistical testing
- Liner modeling (regression, ANOVA)

Schedule

- **Day 1**
 - 10-16 Basic R usage
- **Day 2**
 - 10-16 Descriptive statistics and graphics
- **Day 3**
 - 10-16 Statistical testing
- **Day 4**
 - 10-16 More advanced features of R

Installing R

<http://www.r-project.org>

On Windows, in general


Downloading R I/V

The R Project for Statistical Computing - Windows Internet Explorer

http://www.r-project.org/

File Edit View Favorites Tools Help

The R Project for Statistical Computing



The R Project for Statistical Computing

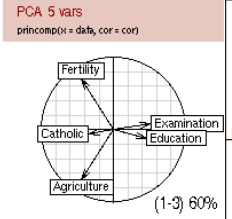
About R
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

Download
[CRAN](#)

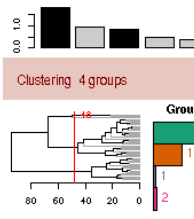
[Foundation](#)
[Members & Donors](#)
[Mailing Lists](#)
[Bug Tracking](#)
[Developer Page](#)
[Conferences](#)
[Search](#)

Documentation
[Manuals](#)
[FAQs](#)
[Newsletter](#)
[Wiki](#)
[Books](#)
[Certification](#)
[Other](#)

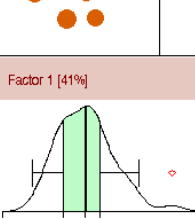
PCA 5 vars
princomp(x = data, cor = cor)



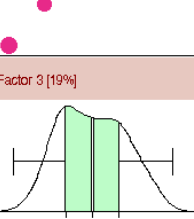
Clustering 4 groups



Factor 1 [41%]



Factor 3 [19%]

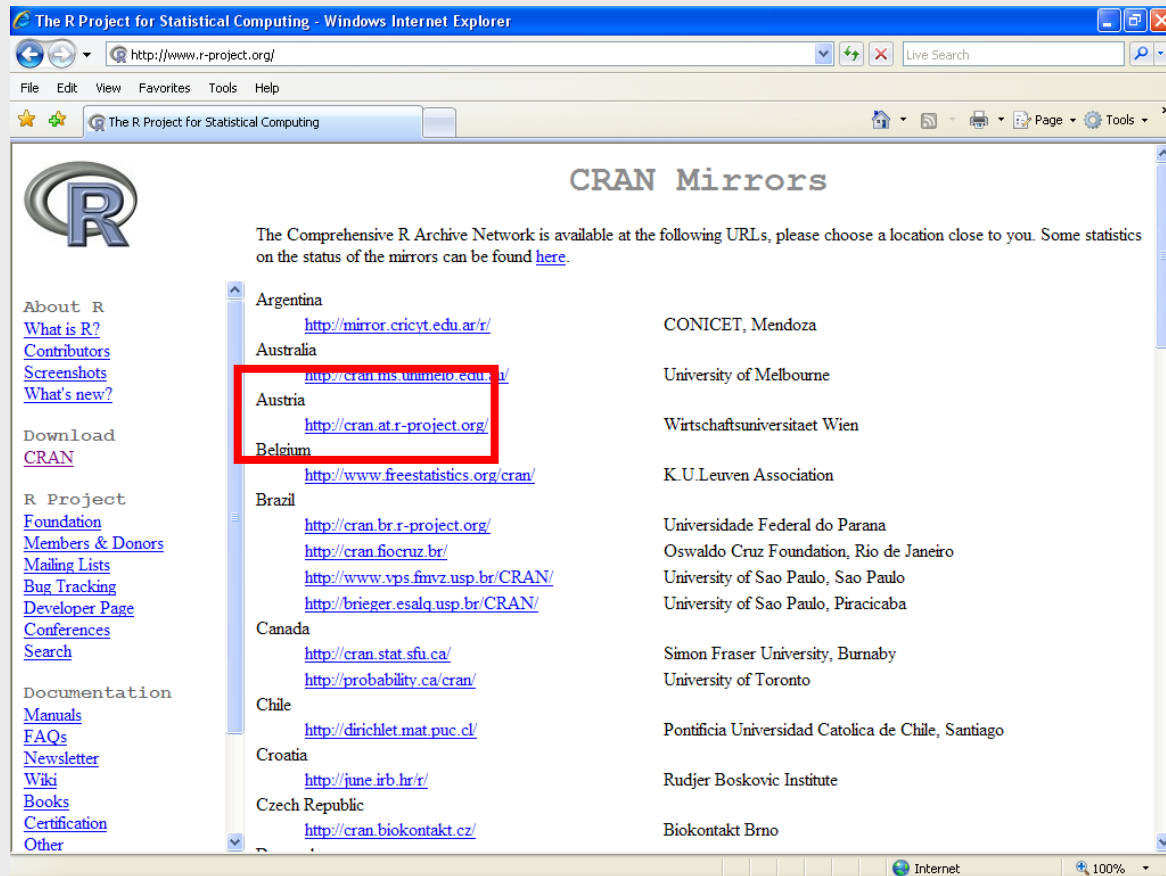


Getting Started:

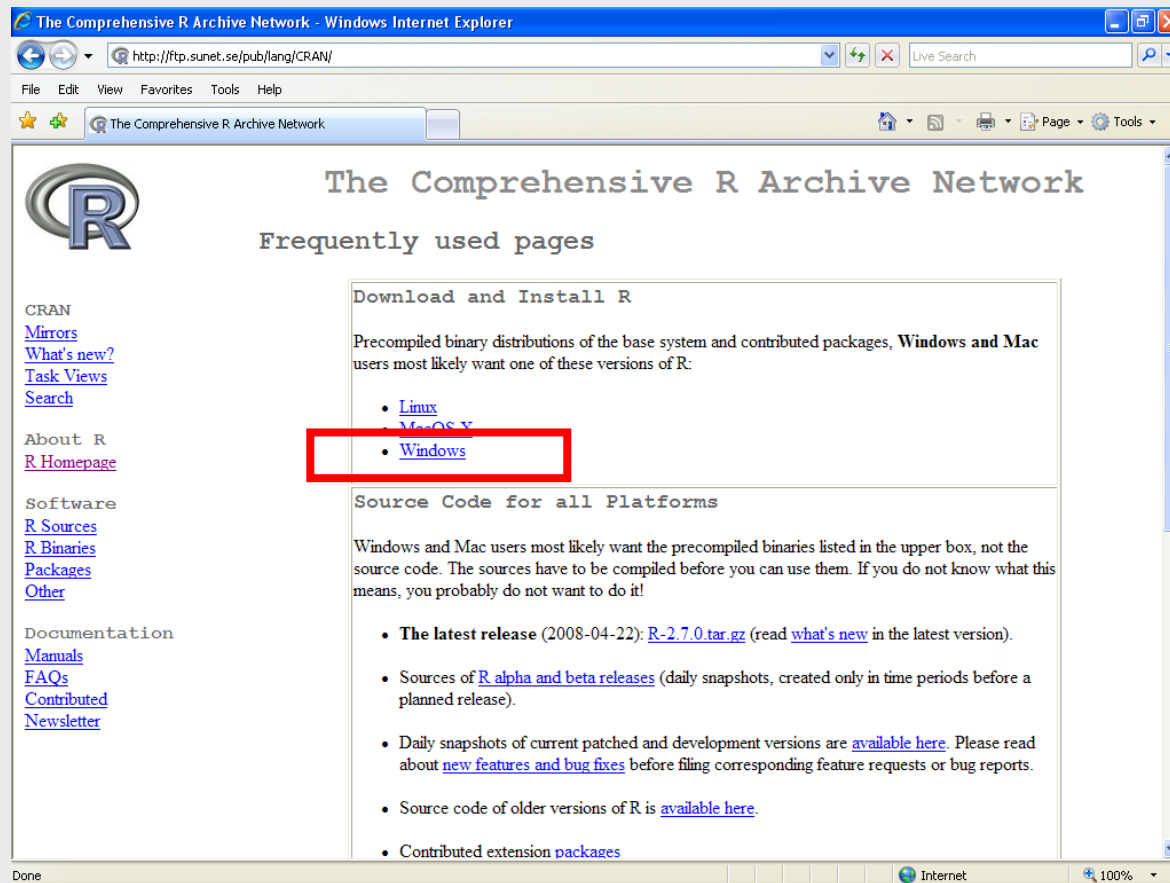
- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred [CRAN mirror](#).
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

Done Internet 100%

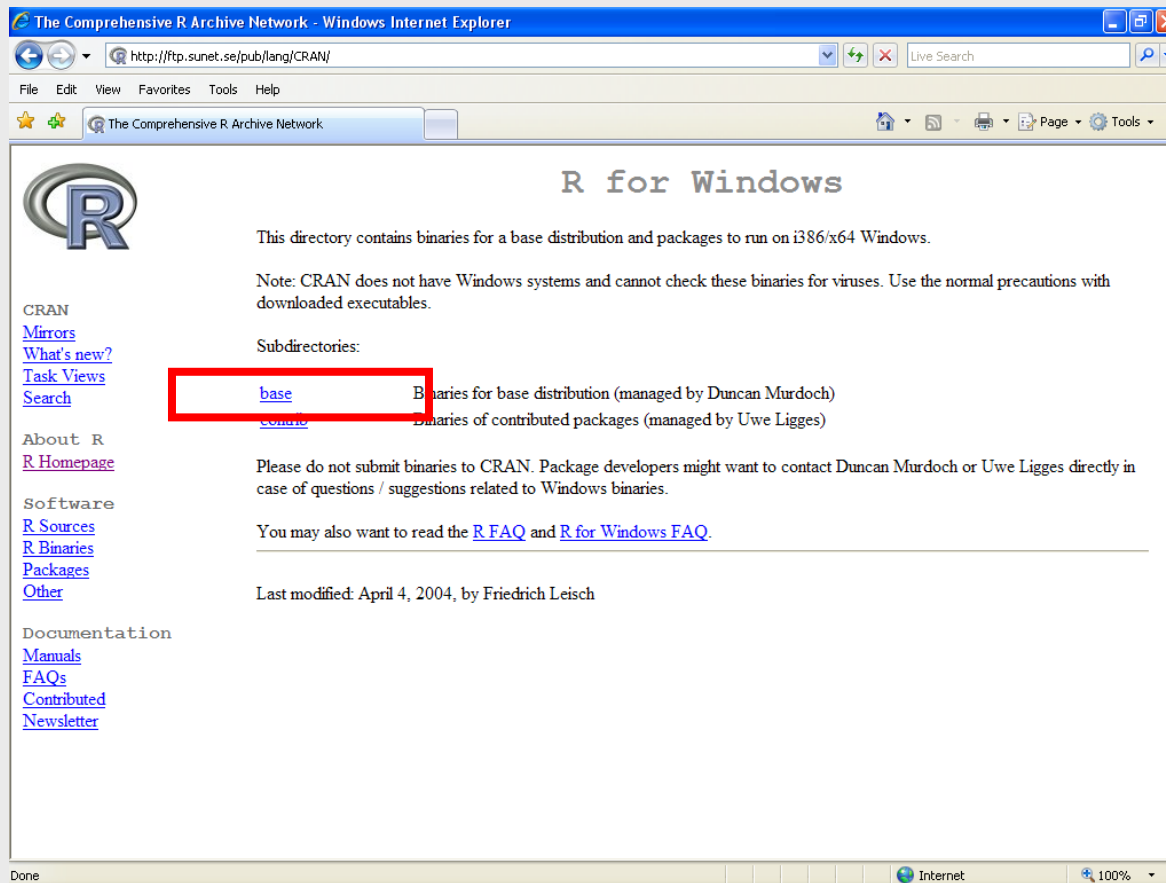
Downloading R II/V



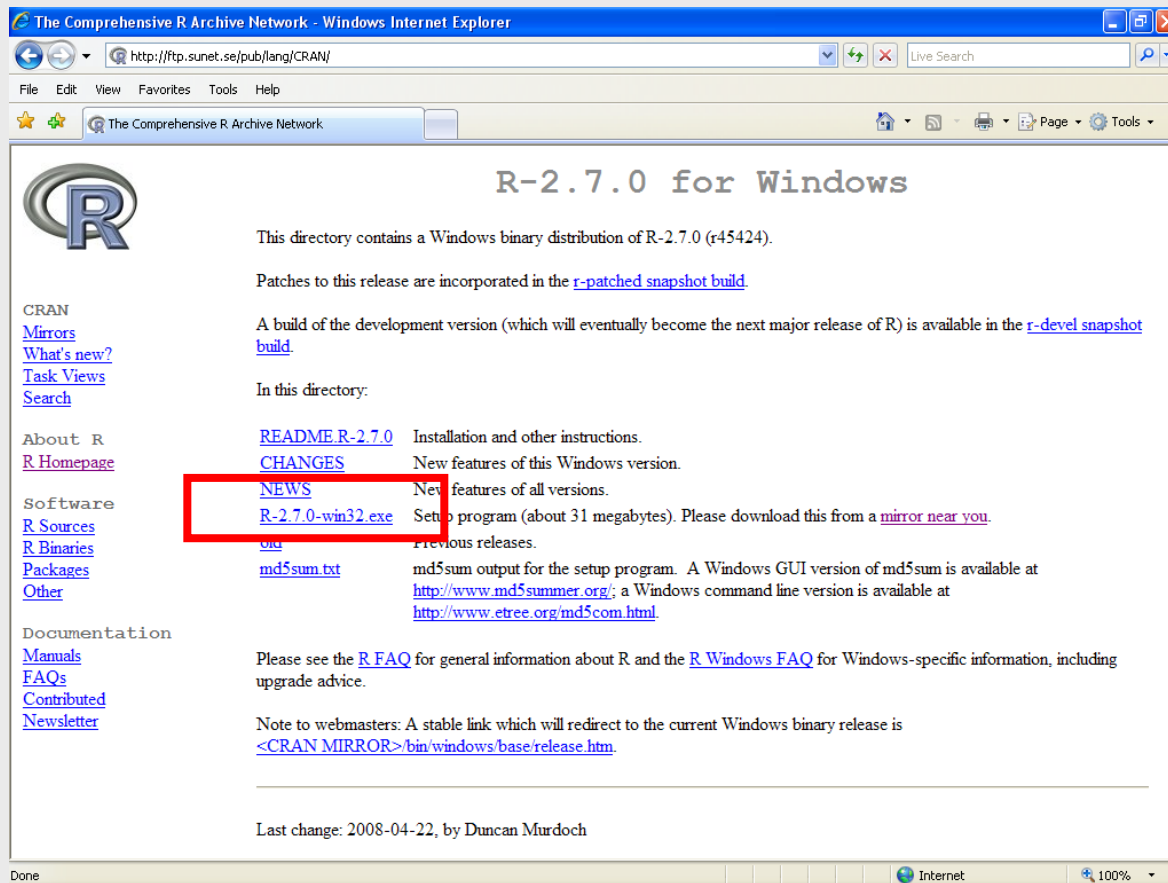
Downloading R III/V



Downloading R IV/V



Downloading R V/V



The screenshot shows a Windows Internet Explorer browser window displaying the CRAN website for R-2.7.0 for Windows. The browser's address bar shows the URL <http://ftp.su.se/pub/lang/CRAN/>. The website content includes the R logo, the title "R-2.7.0 for Windows", and a list of links for downloading and documentation. The link [R-2.7.0-win32.exe](#) is highlighted with a red box.

R-2.7.0 for Windows

This directory contains a Windows binary distribution of R-2.7.0 (r45424).

Patches to this release are incorporated in the [r-patched snapshot build](#).

A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).

In this directory:

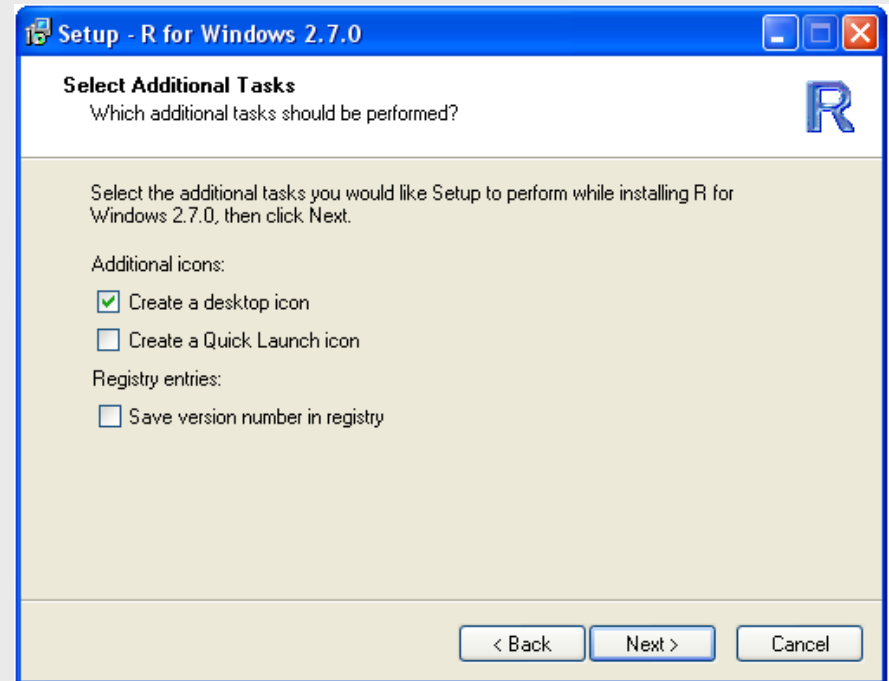
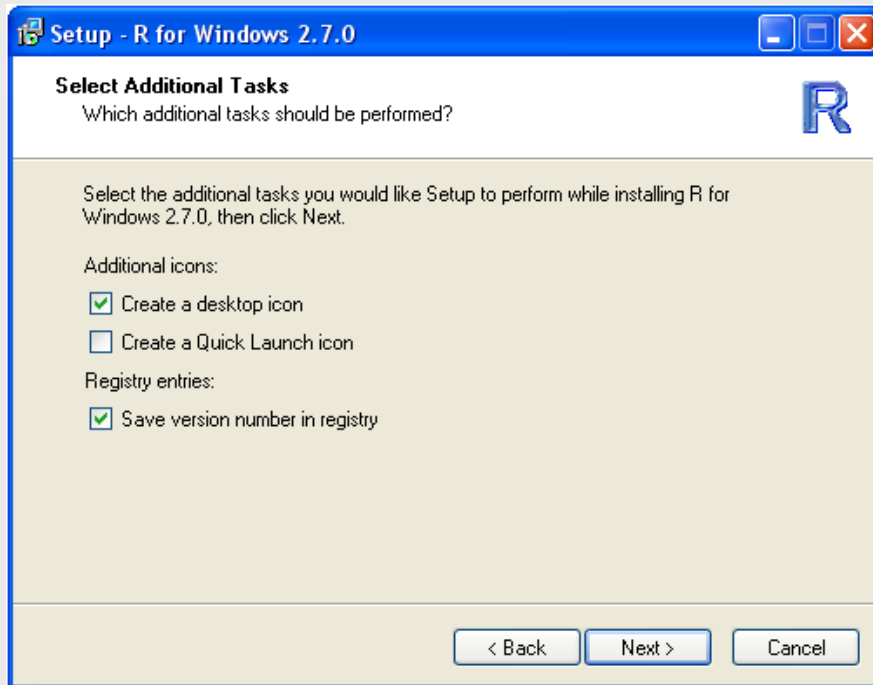
- [README R-2.7.0](#) Installation and other instructions.
- [CHANGES](#) New features of this Windows version.
- [NEWS](#) New features of all versions.
- [R-2.7.0-win32.exe](#) Setup program (about 31 megabytes). Please download this from a [mirror near you](#).
- [bin](#) Previous releases.
- [md5sum.txt](#) md5sum output for the setup program. A Windows GUI version of md5sum is available at <http://www.md5summer.org/>; a Windows command line version is available at <http://www.etree.org/md5com.html>.

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information, including upgrade advice.

Note to webmasters: A stable link which will redirect to the current Windows binary release is CRAN.MIRROR>/bin/windows/base/release.htm.

Last change: 2008-04-22, by Duncan Murdoch

Installing



Exercise I

Installing on this course

- On this course we using an easier setup, where we copy the already created R installation to each persons computer.
- This is a version where certain settings have been slightly modified.
- Go to <http://www.csc.fi/english/csc/courses/archive/R2008s>, and click on the link Download R 2.7.0. Save the file on Desktop.
- Extract the zip-file to desktop (right-click on the file, and select Winzip -> Extract to here).
- Go to folder R-2.7.0c/bin and right-click on file Rgui.exe. Select Create Shortcut.
- Copy and paste the shortcut to Desktop.

Packages

What are packages?

- **R is built around packages.**
- **R consist of a core (that already includes a number of packages) and contributed packages programmed by user around the world.**
- **Contributed packages add new functions that are not available in the core (e.g., genomic analyses).**
- **Contributed packages are distributed among several projects**
 - CRAN (central R network)
 - Bioconductor (support for genomics)
 - OmegaHat (access to other software)
- **In computer terms, packages are ZIP-files that contain all that is needed for using the new functions.**

How to get new packages?

➤ **The easiest way is to:**

1. Packages -> Select repository
2. Packages -> Install packages
 - Select the closest mirror (Sweden probably)

➤ **You can also download the packages as ZIP-files.**

- Save the ZIP-file(s) into a convenient location, and without extracting them, select Packages -> Install from a local ZIP file.

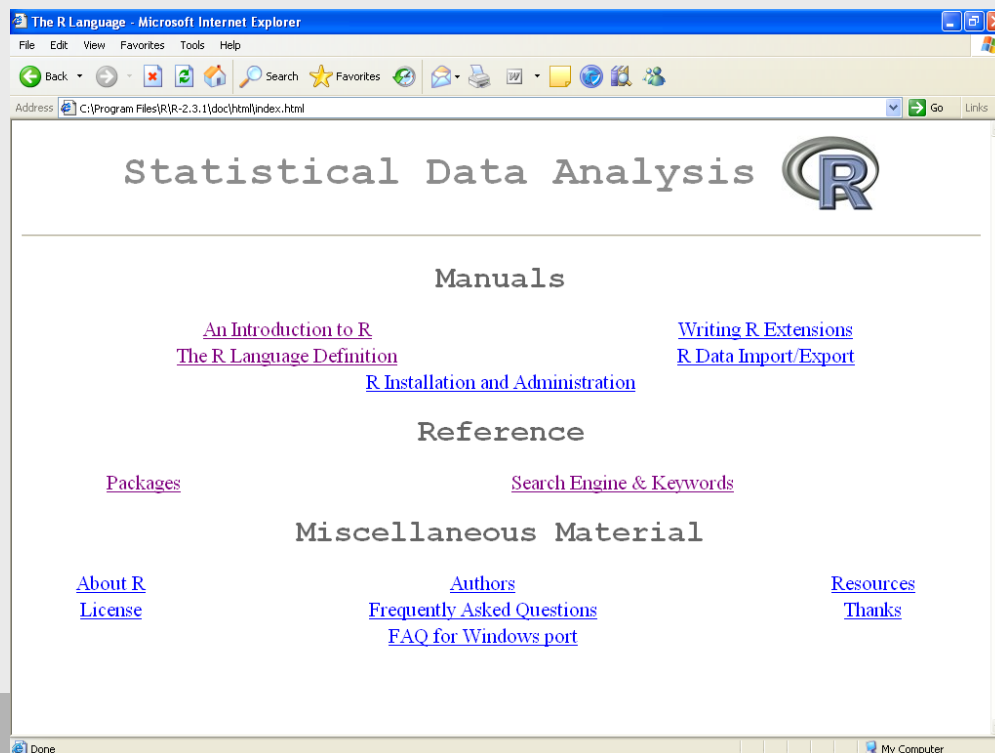
How to access the functions in packages?

- **Before using any functions in the packages, you need to load the packages in memory.**
- **On the previous step packages were just installed on the computer, but they are not automatically taken into use.**
- **To load a package into memory**
 - Packages -> Load Packages
 - Or as a command: `library(rpart)`
- **If you haven't loaded a package before trying to access the functions contained in it, you'll get an error message:**
 - Error: could not find function "rpart"

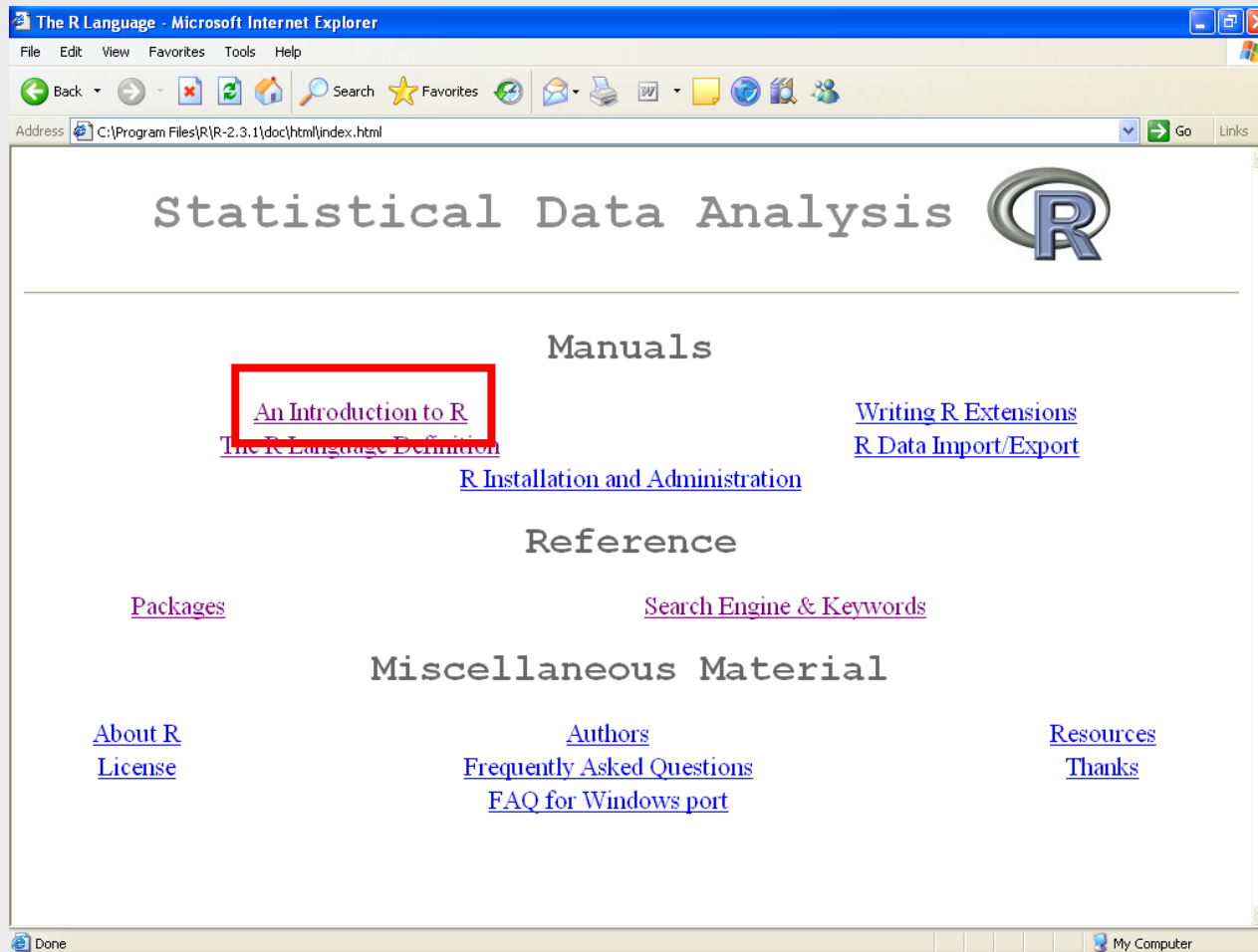
Help facilities

HTML help

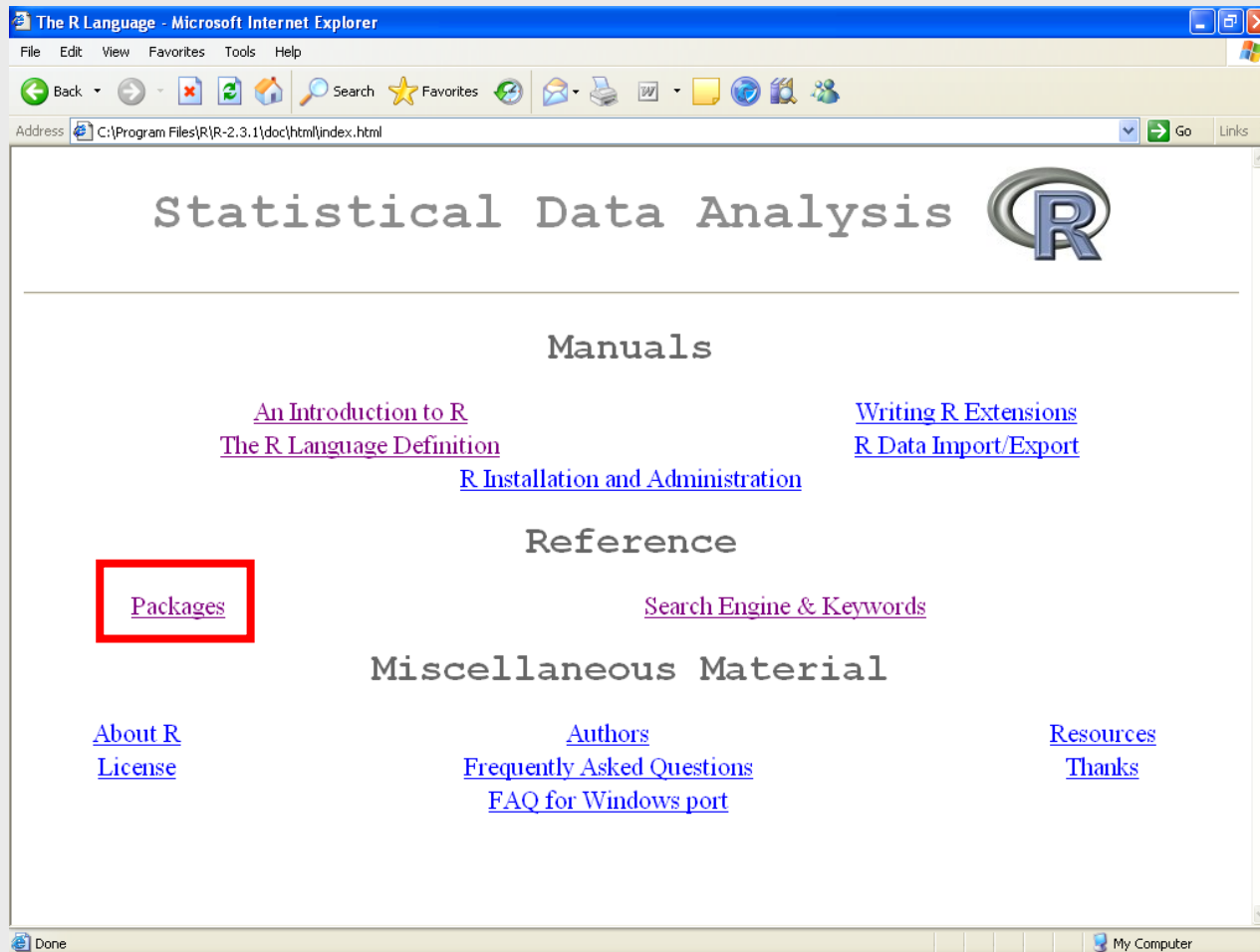
- To invoke a built-in help browser, select Help->HTML help.
 - Command: `help.start()`
- This should open a browser window with help topics:



A basic book



List of installed packages



Help for packages


R: Package Index - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address C:\Program Files\R\R-2.3.1\doc\html\packages.html

Package Index



abind	Combine multi-dimensional arrays
acepack	ace() and avas() for selecting regression
affy	Methods for Affymetrix Oligonucleotide
affydata	Affymetrix Data for Demonstration Pur
affyio	Tools for parsing Affymetrix data files
affyPLM	Methods for fitting probe-level models
annaffy	Annotation tools for Affymetrix biolog
annotate	Annotation for microarrays
aroma	An R Object-oriented Microarray Analy
aroma.light	Light-weight methods for normalizatio
	basic R data types
base	The R Base Package
Biobase	Biobase: Base functions for Bioconduc
biomaRt	Interface to BioMart databases (e.g. En
Biostrings	String objects representing biological s
boot	Bootstrap R (S-Plus) Functions (Canty
car	Companion to Applied Regression


R: Graphics related functions for Bioconductor - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address C:\Program Files\R\R-2.3.1\library\genefilter\html\00Index.html

Graphics related functions for Bioconductor



Documentation for package 'genefilter' version 1.10.0

User Guides and Package Vignettes

Read [overview](#) or browse [directory](#).

Help Pages

alongChrom	A function for plotting expression data from an exprset for a given chromosome.
amplicon.plot	Create an amplicon plot
buildACMainLabel	A function for plotting expression data from an exprset for a given chromosome.
cColor	A function for marking specific probes on a cPlot.
closeHtmlPage	Open and close an HTML file for writing.
connection-class	Virtual S4 classes for method dispatching
cPlot	A plotting function for chromosomes.
cScale	A function for mapping chromosome length to a number of points.
cullACXPoints	A function for plotting expression data from an exprset for a given chromosome.

Anatomy of a help file 1/2

R: MAS 5.0 expression measure - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address <C:\Program Files\R\R-2.3.1\library\affy\html\mas5.html> Go Links

mas5 {affy} ← R Documentation

MAS 5.0 expression measure

Description

This function converts an instance of [AffyBatch-class](#) into an instance of [exprSet-class](#) using our implementation of Affymetrix's MAS 5.0 expression measure.

Usage

```
mas5(object, normalize = TRUE, sc = 500, analysis = "absolute", ...)
```

Arguments

object	an instance of AffyBatch-class
normalize	logical. If TRUE scale normalization is used after we obtain an instance of exprSet-class
sc	Value at which all arrays will be scaled to.
analysis	should we do absolute or comparison analysis, although "comparison" is still not implemented.
...	other arguments to be passed to expresso .

Details

This function is a wrapper for [expresso](#) and [affy.scalevalue.exprSet](#).

Value

Function {package}

General description

Command and it's argument

Detailed description of arguments

Anatomy of a help file 2/2

R: MAS 5.0 expression measure - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address <C:\Program Files\R\R-2.3.1\library\affy\html\mas5.html> Go Links

sc Value at which all arrays will be scaled to.
analysis should we do absolute or comparison analysis, although "comparison" is still not implemented.
... other arguments to be passed to [expresso](#).

Details

This function is a wrapper for [expresso](#) and [affy.scalevalue.exprSet](#).

Value

[exprSet-class](#)
The methods used by this function were implemented based upon available documentation. In particular a useful reference is Statistical Algorithms Description Document by Affymetrix. Our implementation is based on what is written in the documentation and as you might appreciate there are places where the documentation is less than clear. This function does not give exactly the same results. All source code of our implementation is available. You are free to read it and suggest fixes. For more information visit this URL: <http://stat-www.berkeley.edu/users/bolstad/>

See Also

[expresso](#), [affy.scalevalue.exprSet](#)

Examples

```
data(affybatch.example)
eset <- mas5(affybatch.example)
```

[Package *affy* version 1.10.0 [Index](#)]

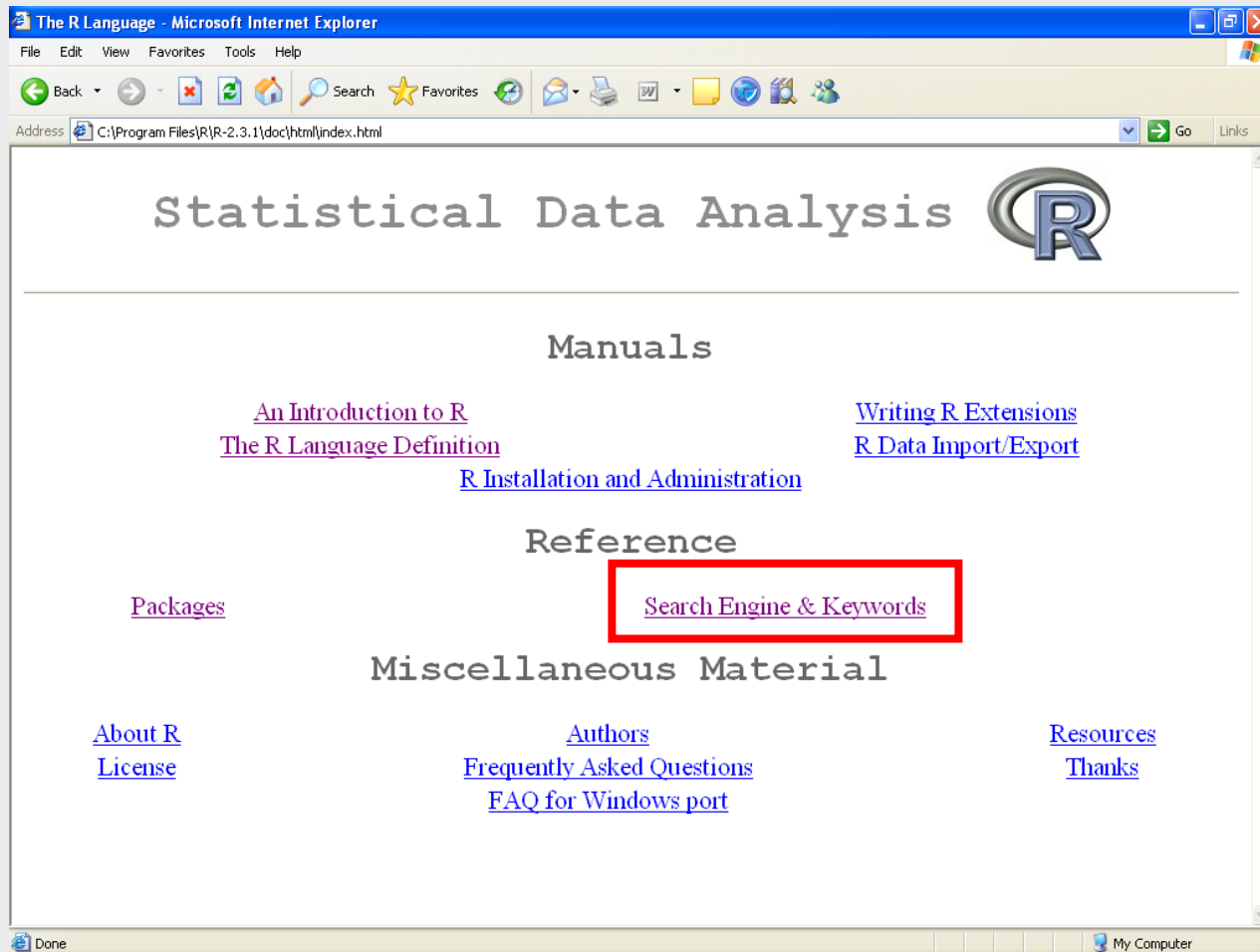
Description of how function actually works

What function returns

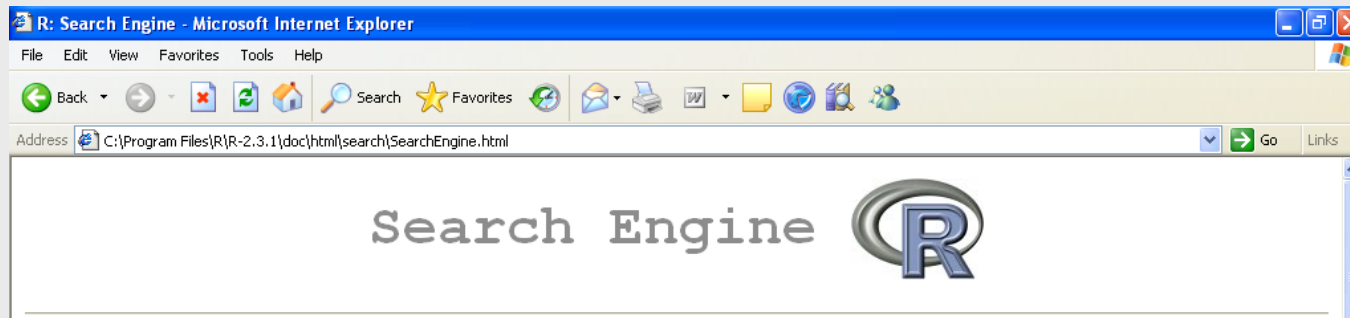
Related functions

Examples, can be run from R by:
`example(mas5)`

Search help



Search results



You can search for keywords, function and data names and text in help

Usage: Enter a string in the text field below and hit RETURN.

☒ Help page titles ☒ Keywords ☒ Object names

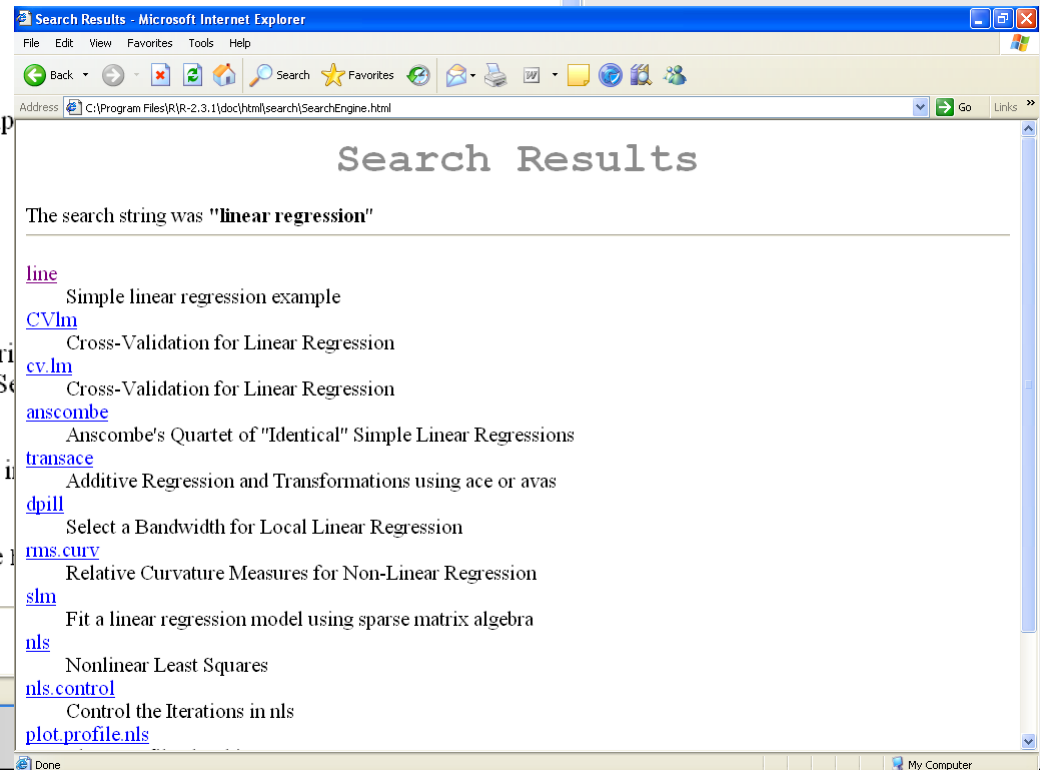
For search to work, you need Java installed and both Java and JavaScript. On the Mozilla/Netscape family of browsers you should see 'Applet Security' bar. For help consult the [R Installation and Administration](#) manual.

On Mozilla-based browsers the links on the results page will become if you can open a link in a new tab or window.

Even if this search does not work on your system, you can always use

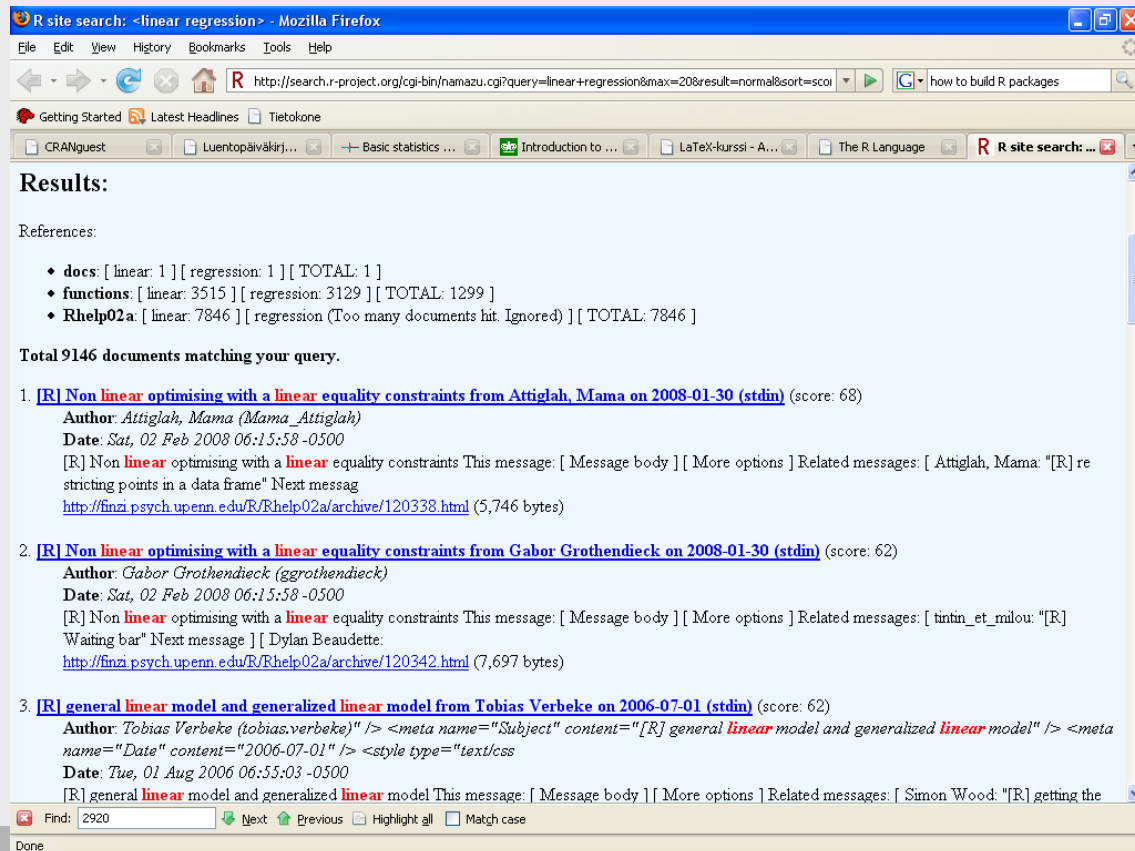
Keywords

Applet SearchEngine started



Other search possibilities I/II

➤ Help -> search.r-project.org



The screenshot shows a Mozilla Firefox browser window with the address bar displaying the URL: `http://search.r-project.org/cgi-bin/namazu.cgi?query=linear+regression&max=20&result=normal&sort=score`. The search results are displayed on the R site search page.

Results:

References:

- docs: [linear: 1] [regression: 1] [TOTAL: 1]
- functions: [linear: 3515] [regression: 3129] [TOTAL: 1299]
- Rhelp02a: [linear: 7846] [regression (Too many documents hit. Ignored)] [TOTAL: 7846]

Total 9146 documents matching your query.

1. [\[R\] Non linear optimising with a linear equality constraints from Attiglah, Mama on 2008-01-30 \(stdin\)](#) (score: 68)
Author: Attiglah, Mama (Mama_Attiglah)
Date: Sat, 02 Feb 2008 06:15:58 -0500
[R] Non **linear** optimising with a **linear** equality constraints This message: [Message body] [More options] Related messages: [Attiglah, Mama: "[R] restricting points in a data frame" Next message: <http://finzi.psych.upenn.edu/R/Rhelp02a/archive/120338.html> (5,746 bytes)
2. [\[R\] Non linear optimising with a linear equality constraints from Gabor Grothendieck on 2008-01-30 \(stdin\)](#) (score: 62)
Author: Gabor Grothendieck (ggrothendieck)
Date: Sat, 02 Feb 2008 06:15:58 -0500
[R] Non **linear** optimising with a **linear** equality constraints This message: [Message body] [More options] Related messages: [tintin_et_milou: "[R] Waiting bar" Next message: [Dylan Beaudette: <http://finzi.psych.upenn.edu/R/Rhelp02a/archive/120342.html> (7,697 bytes)
3. [\[R\] general linear model and generalized linear model from Tobias Verbeke on 2006-07-01 \(stdin\)](#) (score: 62)
Author: Tobias Verbeke (tobias.verbeke) /> <meta name="Subject" content="[R] general **linear** model and generalized **linear** model" /> <meta name="Date" content="2006-07-01" /> <style type="text/css"
Date: Tue, 01 Aug 2006 06:55:03 -0500
[R] general **linear** model and generalized **linear** model This message: [Message body] [More options] Related messages: [Simon Wood: "[R] getting the

Find: 2920 Next Previous Highlight all Match case

Done

Other search possibilities II/II

➤ <http://www.r-seek.org>

RSeek.org R-project Search Engine - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.rseek.org/?cx=010923144343702598753%3Aboaz1reyxd4&q=linear+regression&sa=Search+H

Getting Started Latest Headlines Tietokone

CRANguest Luentopäiväkirj... Basic statistics ... Introduction to ... LaTeX-kurssi - A... The R Language RSeek.org R-...

linear regression Search

Results 1 - 10 for **linear regression**. (0.49 seconds)

[CRAN Task View: Computational Econometrics](#)
Linear regression models : **Linear** models can be fitted (via OLS) with `lm()` (from stats) and standard tests for model comparisons are available in various ...
cran.r-project.org/web/views/Econometrics.html

[useR! 2006: Non-Linear Regression Models in R](#)
The tutorial aims at illustrating how to use R to fit non-**linear regression** models that consist of several curves. How to fit such models is a recurring ...
www.r-project.org/useR-2006/Tutorials/Ritz+Streibig.html

[R Guide -- the linear model](#)
 $y \sim x$ or $y \sim 1+x$ are both examples of simple **linear regression** with an implicit or explicit intercept. $y \sim 0+x$ or $y \sim -1+x$ or $y \sim x-1$ **linear regression** through the ...
www.personality-project.org/r/r_lm.html

[Bayesian Model Averaging Home Page](#)
BMA has been applied successfully to many statistical model classes including **linear regression**, generalized **linear** models, Cox **regression** models, ...
www.research.att.com/~volinsky/bma.html

[CRAN Task View: Bayesian Inference](#)
The models include **linear regression** models, multinomial logit, multinomial probit, multivariate probit, multivariate mixture of normals (including ...

[Introductions](#) [Support Lists](#) [Functions](#) [R code](#) [Books](#)

[Generalized Linear Models: logistic regression, Poisson regression ...](#)
We can consider the qualitative variable as a quantitative variable, assuming two values, 0 and 1, and perform a **linear regression** against the others. ...
zoonek2.free.fr/UNIX/48_R/12.html

[R Class Notes: Analyzing Data](#)
The `glm` function fits a general **linear** model including a logistic **regression**. In order to fit a logistic model we need to specify that the distribution of ...
www.ats.ucla.edu/stat/r/notes/analyze.htm

[Econometrics in R](#)
A simple **linear regression** might be The `glm()` command provides access to a plethora of other advanced **linear regression** methods. See the ...
cran.r-project.org/doc/contrib/Farnsworth-EconometricsInR.pdf

[R Library: Matrices and matrix computations](#)
Using matrix computations to perform a basic **linear regression** on the data set `hsb2`. Here is `hsb2.txt` as a text file for use in R. `y <- matrix(hsb2$write, ...`
www.ats.ucla.edu/stat/r/library/matrix_alg.htm

[Regression](#)
You can see **linear regression** as an orthogonal projection of Y onto the subspace generated by 1, X_1 , X_2 , etc. (here, 1 is the vector all of whose ...
zoonek2.free.fr/UNIX/48_R/D9.html

[An Introduction to R](#)

Find: 2920 Next Previous Highlight all Match case

Transferring data from bls7.books.google.com...

Exercise II

Install packages and use help

1. Install the following package(s):

- car (can be found from CRAN)

2. Load the library into memory

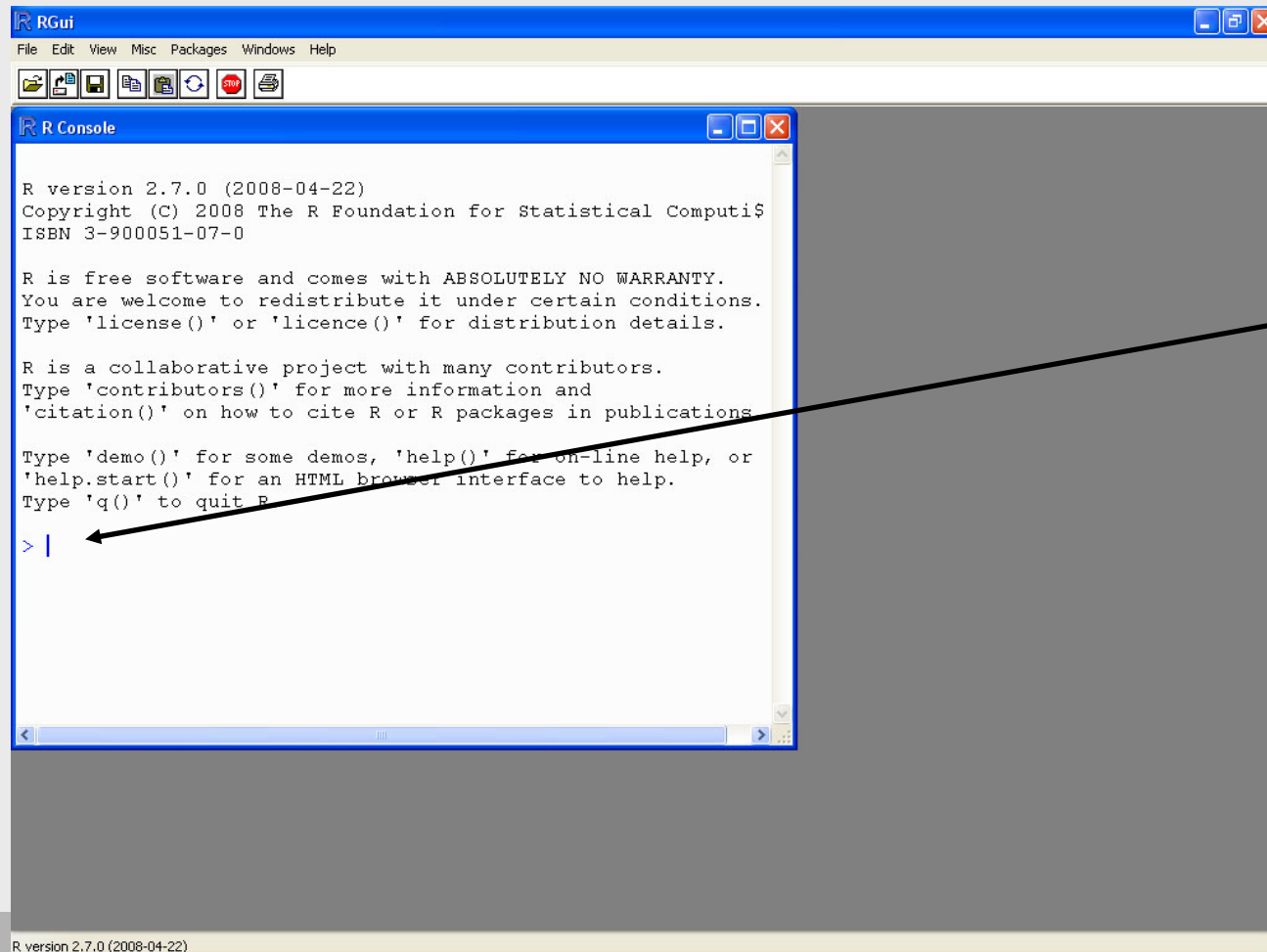
3. Consult the help files for the car package.

- What does States contain?
- What does function scatterplot do?

4. What packages are available for data analysis in epidemiology?

Basic use and data import I

Interface



- **Normal text:**
black
- **User text:**
blue
- **Prompt:**
that where
type the
commands

R as a calculator

- **R can be used as a calculator.**
- **You can just type the calculations on the prompt. After typing these, you should press Return to execute the calculation.**
 - `2+1` `# add`
 - `2-1` `# subtract`
 - `2*1` `# multiply`
 - `2/1` `# divide`
 - `2^2` `# potency`
- **Note: # is a comment mark, nothing after it on the same line is not executed**
- **Normal rules of calculation apply:**
 - `2+2*3` `# =8`
 - `(2+2)*3` `# =12`

Anatomy of functions or commands

- **To use a function in a package, the package needs to be loaded in memory.**
- **Command for this is `library()`, for example:**
 - `library(affy)`
- **There are three parts in a command:**
 - The command - `library`
 - Brackets – `()`
 - Arguments inside brackets (these are not always present) - `affy`
- **Arguments modify or specify the commands**
 - Command `library()` loads a library, but unless it is given an argument (name of the library) it doesn't know what to load.
- **R is case sensitive!**
 - `library(affy)` # works!
 - `Library (affy)` # fails

Mathematical functions

➤ **R contains many mathematical function, also.**

- `log(10)` # natural logarithm, 2.3
- `log2(8)` # 3
- `exp(2.3)` # 9.97
- `sin(10)` # -0.54
- `sqrt(9)` # square root, 3

- `sum(v)`
- `diff(v)`

Comparisons

- **Is equal**
 - ==
- **Is larger than**
 - >
- **Is larger than or equal to**
 - >=
- **Smaller than or equal to**
 - <=
- **Isnot equal to**
 - !=
- **Examples**
 - 3==3 # TRUE
 - 2!=3 # TRUE
 - 2<=3 # TRUE

Logical operators

➤ Basic operators are

- & # and
- | # or (press Alt Gr and < simultaneously)

➤ Examples

- 2==3 | 3==3 # TRUE (if either is true then print TRUE)
- 2==3 & 3==3 # FALSE (another statement is FALSE, so ->FALSE)

Creating vectors I/III

- **So far, we've been applying the function on only one number at a time.**
- **Typically we would like to do the same operation for several number at the same time.**
 - Taking a \log_2 of several numbers, for instance
- **First, we need to create a vector that holds those several numbers:**
 - `v<-c(1,2,3,4,5)`
 - Everything in R is an object
 - Here, v is an object used for storing these 5 numbers
 - `<-` is the operator that stores something
 - `c()` is a command for creating a vector by typing values to be stored.

Naming objects

- **Never use command names as object names!**
- **If you are unsure whether something is a command name, type it to the command line first. If it gives an error message, you're safe to use it.**
 - data # not good
 - dat # good
- **Object names can't start with a number**
 - 1a # not good
 - a1 # good
- **Never use special characters, such as å, ä, or ö in object names.**
- **Object names are case sensitive, just like commands**
 - A1 # object nro 1
 - a1 # object nro 2

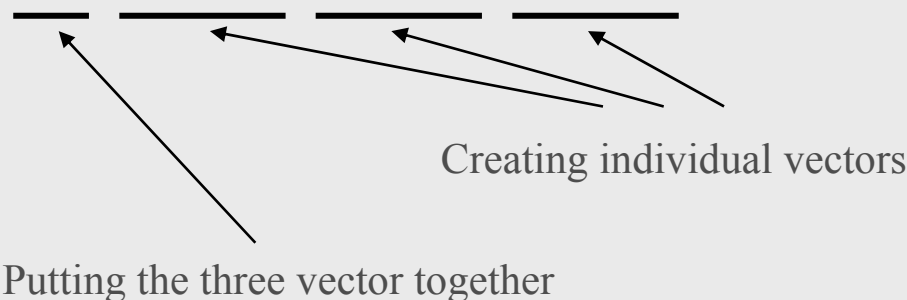
Creating vectors II/III

- **Vectors can also be created using : notation, if the values are continuous:**
 - `v <- c(1:5)`
- **For creating a vector of three 1s, four 2s, and five 3s, there are several options:**
 - `v <- c(1,1,1,2,2,2,2,3,3,3,3,3)`
 - Using `rep()`
 - `v1<-rep(1,3)` # Creates a vector of three ones
 - `v2<-rep(2,4)`
 - `v3<-rep(3,5)`
 - `v<-c(v1,v2,v3)`
 - Putting the command together:
 - `v<-c(rep(1,2), rep(2,4), rep(3,5))`

Creating vectors III/III

- Let's take a closer look at the last command:

- `v<-c(rep(1,2), rep(2,4), rep(3,5))`



- So you can nest commands, and that is very commonly done!
- But nothing prevents you from breaking these nested commands down, and running them one by one
 - That's what we did on the last slide

Applying functions to vectors

- **If you apply any of the previously mentioned functions to a vector, it will be applied separately for every observation in that vector:**

```
> log2(v)
```

```
[1] 0.000000 0.000000 0.000000 1.000000 1.000000 1.000000
```

```
[7] 1.000000 1.584963 1.584963 1.584963 1.584963 1.584963
```

- **When applied to a vector, the length of the result is as long as the starting vector.**
- **When a function is applied to a vector this way, the calculation is said to be vectorized.**

Exercise III

Import Data + some calculations

➤ **A certain American car was followed through seven fill ups. The mileage was:**

- 65311, 65624, 65908, 66219, 66499, 66821, 67145, 67447

- 1. Enter the data in R.**
- 2. How many observations there are in the data (what is the R command)?**
- 3. What is total distance driven during the follow up?**
- 4. What are the fill up distances in kilometers (1 mile = 1.6 km)?**
- 5. Use function `diff()` on the data. What does it tell you?**
- 6. What is the longest distance between two fill ups (search for a appropriate command from the help)?**

Basic use and data import II

Factors

- **In vectors you have a list of values. Those can be numbers or strings.**
- **Factors are a different data type. They are used for handling categorical variable, e.g., the ones that are nominal or ordered categorical variables.**
 - Instead of simply having values, these contain levels (for that categorical variable)
- **Examples:**
 - Male,female
 - Featus, baby, toddler, kid, teenager, young adult, middle-aged, senior, aged

Creating factors I/III

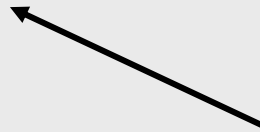
- **Factors can be created from vectors, or from a scratch.**
- **Here I present only the route from vectors.**
- **So, let's create a vector of numerical values (1=male, 2=female):**
 - `v<-c(1,2,1,1,1,2,2,2,1,2,1)`
- **To convert the vector to factor, you need to type:**
 - `f<-as.factor(v)`
- **Check what R did:**

```
> f
[1] 1 2 1 1 1 2 2 2 1 2 1
Levels: 1 2
```
- **f is now a vector with two levels (1 and 2).**

Creating factors II/III

- **Levels of factors can also be labeled. This makes using them in statistical testing much easier.**

- `f<-factor(v, labels=c("male", "female"))`



A string vector!

```
> f
```

```
[1] male  female male  male  male  female female female male  female  
     male
```

```
Levels: male female
```

- **Which order do you give the levels then?**

- Check how the values are printed in
 - `unique(sort(v))` # 1 2

Creating factors III/III

➤ **Levels of a factor can also be ordered.**

- These are similar to the unordered factors, but statistical tests treat them quite differently.

➤ **To create an ordered factor, add argument `ordered=T`:**

- `f<-factor(v, labels=c("male", "female"), ordered=T)`

`> f`

```
[1] male  female male  male  male  female female female male  female  
     male
```

Levels: male < female

- Note the < sign! That identifies the factor as ordered.

Applying functions to factors

- **You can't calculate, for example, log2 of every observation is a factor.**

```
> log2(f)
```

```
Error in Math.factor(f) : log2 not meaningful for factors
```

- **There are separate function for manipulating factors, such as:**

```
> table(f)
```

```
f
```

```
male female
```

```
6      5
```

Data frames

- **Data frames are, well, tables (like in any spreadsheet program).**
- **In data frames variables are typically in the columns, and cases in the rows.**
- **Columns can have mixed types of data; some can contain numeric, yet others text**
 - If all columns would contain only character or numerical data, then the data can also be saved in a matrix (those are faster to operate on).

	V1	V2	V3
C1	1	0	one
C2	2	1	two
C3	3	0	three

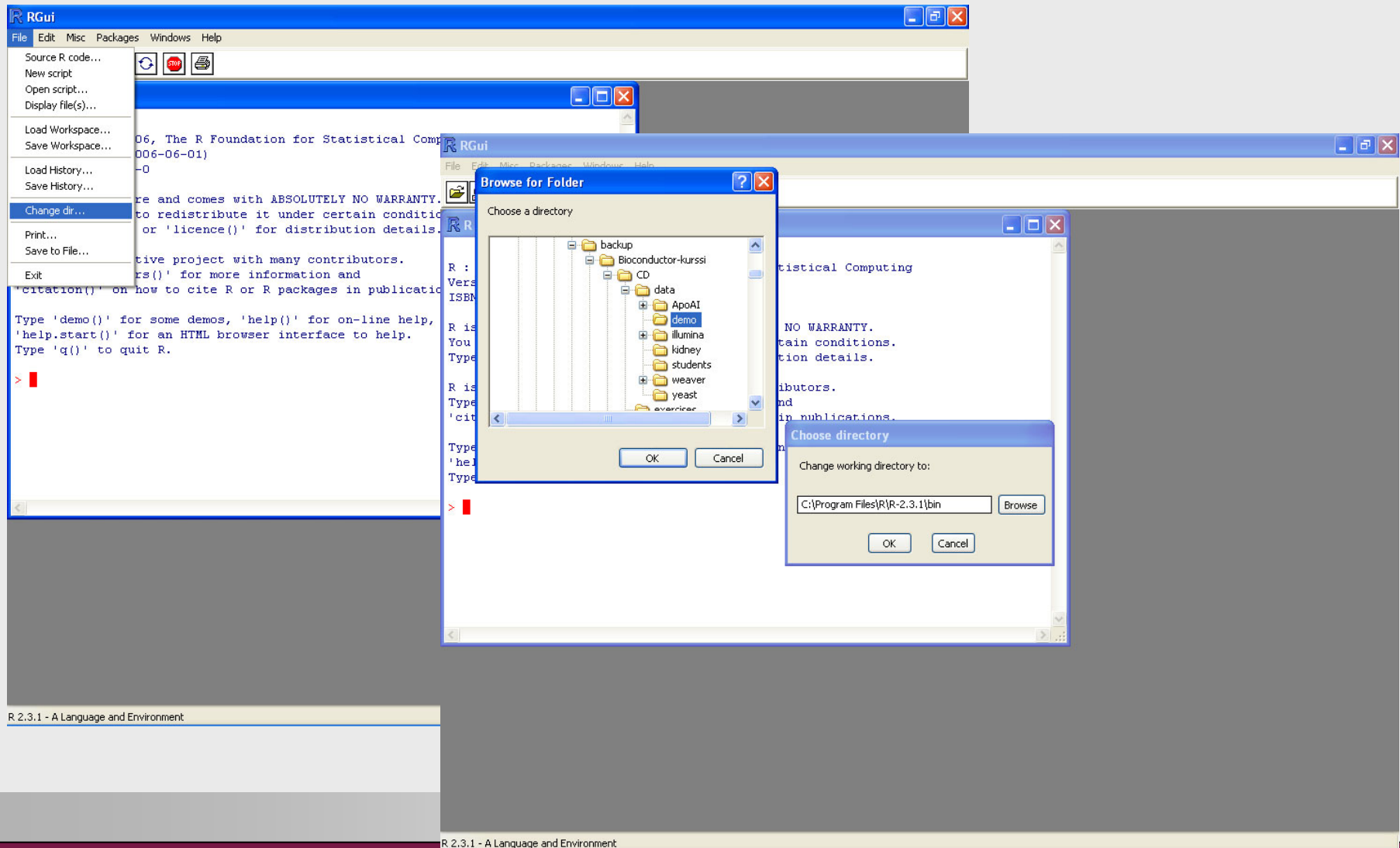
Data frames

- From previous slides we have two variable, v and f.
- To make a data frame that contains both of these variables, one can use command:
 - `d<-data.frame(v, f)`
- To bind the two variables into a table, one could also use
 - `d2<-cbind(v, f)`
- The difference between these methods is that the first creates a data frame and the second one a matrix.

Data frames and data import

- Usually when you import a data set in R, you read it in a data frame.
- This is assuming your data is in a table format.
- One can input the data in a table with some spreadsheet, but it should be saved as tab-delimited text file to make importing easy.
- This text file should not contain are (unmatched) quotation marks (' or ").
- It is best to fill in all empty fields with some value (not leave them blank in the spreadsheet).
 - Missing values (no measument): NA
 - Small values: 0?

Starting the work with R (browse to a folder)



Importing a tabular file

- **Simply type:**
 - `dat<-read.table("filename", header=T, sep="\t", row.names=1)`
- **dat** is the name of the object the data is saved in R
- **<-** is the assignment operator
- `read.table()` is the command that reads in tabular files
- It needs to get a filename, complete with the extension (Windows hides those by default)
- If every column contains a title, then argument should be `header=TRUE` (or `header=T`), otherwise `header=F`.
- If the file is tab-delimited (there is a tab between every column), then `sep="\t"`. Other options are, e.g., `sep=","` and `sep=" "`.
- If every case (row) has its own unambiguous (non-repeating) title, and the first column of the file contains these row names, then `row.names=1`, otherwise the argument should be deleted.

Importing data from web

➤ **Code can be downloaded and executed from the web with the command `source()`**

- `source("http://koti.mbnet.fi/tuimala/tiedostot/Rcourse_data.txt")`

➤ **Files can be downloaded by `download.file()`**

- `download.file("http://koti.mbnet.fi/tuimala/tiedostot/rairuoho.txt",
destfile="rairuoho.txt")`

Checking the objects and memory

➤ **To see what objects are in memory:**

- `ls()`

➤ **Length of a vector or factor**

- `length(v)`

➤ **Dimensions of a data frame or matrix:**

- `dim(d)`

➤ **Column and row names of a data frame or matrix**

- `col.names(d)`

- `row.names(d)`

Exercise IV

Import tabular data

- **Download the file from the Internet:**
 - `http://koti.mbnet.fi/tuimala/tiedostot/rairuoho.txt`
- **Put the file on desktop.**
- **See how the data looks like (use Excel and Wordpad):**
 - Are there columns headers?
 - What is the separator between the columns (space, tab, etc)?
 - Are there row names in the data?
- **Now you should know what arguments to specify in the `read.table()` command, so use it for reading in the data.**

Import the rest of the data

- **I have prepared several datasets for this course.**
- **These can be downloaded from the web:**
 - `source("http://koti.mbnet.fi/tuimala/tiedostot/Rcourse_data.txt")`
- **The datasets are written as R commands, so the command above downloads and runs this command file.**
- **Check what object were created in R memory?**
- **Run the command `showMetaData()`.**
 - This should show some information about the datasets.
 - Note that the command is written for this course only (by me), and can't be used in R in general.

Object type conversions

Converting from a data type to another

- **Certain data types can easily be converted to other data types.**
 - Vector <-> factor
 - Data frame <-> matrix
 - Data frame <-> vector / factor
 - Matrix <-> vector / factor
- **Typical need for converting a vector to a factor is when performing some statistical tests.**
- **Data frame might need to be converted into a matrix (or vice versa) when running some statistical tests or when plotting the data.**
- **Several vectors can be cleaved from a data frame or a matrix.**
- **Several vectors can be combined to a data frame or a matrix.**

Converting from a vector to a factor

- **To convert a vector to factor, do**
 - `v2<-as.factor(v, labels=c("Jan", "Feb"))`
 - Unordered factor
 - `v2<-factor(v, ordered=T, labels=c("Jan", "Feb"))`
 - Ordered factor
- **Difference between ordered and unordered factors lies in the detail that if the factor is unordered, the values are automatically ordered in plots and statistical test according to lexical scoping (alphabetically).**
- **If the factor is ordered, then the levels have an explicit meaning in the specified order, for example, January becomes before February.**

Extracting a vector from a data frame I/III

- As individual variables are stored in the columns of a data frame, it is typically of interest to be able to extract these column from a data frame.
- Columns can be addressed using their names or their position (calculated from left to right)
- Rows can be accessed similarly to columns.
- Remember how to check the names?
 - `row.names()`
 - `col.names()`

	Jan	Feb
Jarno	1	31
Dario	2	12
Panu	3	37
Vidal	4	8
Max	5	11

Extracting a vector from a data frame II/III

- This data frame is stored in an object called **dat**.
- The first column is named **Jan**, so we can get the values in it by notation:
 - `dat$Jan`
 - Name of the data frame + `$` + Name of the column
 - There are no brackets, so there is "no" command: we are accessing a data frame.

	Jan	Feb
Jarno	1	31
Dario	2	12
Panu	3	37
Vidal	4	8
Max	5	11

Extracting a vector from a data frame III/III

- This data frame is stored in an object called **dat**.
- To get the first column, one can also point to it with the notation:
 - `dat[,1]` # 1, 2, 3, 4, 5
- This is called a **subscript**.
- Subscript consists of square brackets.
- Inside the bracket there are at least one number.
- The number before a comma points rows, the number after the comma to columns
- The first row would be extracted by:
 - `dat[1,]` # 1, 31
- And the value on the first row of the first column:
 - `Dat[1,1]` # 1
- Again, no brackets -> no commands, so we are accessing an object

	Jan	Feb
Jarno	1	31
Dario	2	12
Panu	3	37
Vidal	4	8
Max	5	11

Extracting several columns of rows

- One can want to extract several columns or rows from a table.
- This can be accomplished using a vector instead of a single number.
- For example, to get the rows 1 and 3 from the previous table:
 - `dat[c(1,3),]`
- Or create the vector first, and extract after that:
 - `v<-c(1,3)`
 - `dat[v,]`
- These should give you:

	Jan	Feb
Jarno	1	31
Panu	3	37

Deleting a column or a row

- One can delete a row or a column (or several of them using a vector in the place of number) from a data frame by using a negative subscript:

- `dat[-1,]`
- `dat[-c(1,3),]`

	Jan	Feb
Dario	2	12
Panu	3	37
Vidal	4	8
Max	5	11

	Jan	Feb
Dario	2	12
Vidal	4	8
Max	5	11

Selecting a subset by some variable

- How to get those rows for which the value for February is below 20?
- Function which gives on index of the rows:
 - `which(dat$Feb<=20)`
[1] 2 4 5
- To get the rows, use then index as a subscript:
 - `i<-which(dat$Feb<=20)`
 - `dat[i,]`

	Jan	Feb
Jarno	1	31
Dario	2	12
Panu	3	37
Vidal	4	8
Max	5	11

Writing data to disk

Using sink

- **Sink prints everything you would normally see on the screen to a file.**

- **Usage:**

- `sink("output.txt")` # Opens a file
- `print("Just testing!")` # Commands
- `sink()` # Closes the file

Using write.table

- **Writing a data frame or a matrix to disk is rather straight-forward.**
 - Command `write.table()`
- **Usage:**
 - `write.table(dat, "dat.txt", sep="\t", quote=F, row.names=T, col.names=T)`
 - `dat` name of the table in R
 - `"dat.txt"` name of the file on disk
 - `sep="\t"` use tabs to separate columns
 - `quote=F` don't quote anything, not even text
 - `row.names=T` write out row names (or F if there are no row names)
 - `col.names=T` write out column names

Quitting R

Quitting R

- **Command**

- `q()`

- **Asks whether to save workspace image or not.**

- Answering yes would save all objects on disk in a file `.RData`.
- Simultaneously all the commands given in this session are saved in a file `.RHistory`.

- **These workspace files can be later-on loaded back into memory from the File-menu (Load workspace and Load history).**

Exercise V

Extracting columns and rows I/II

- **What is the size of the Students dataset (number of rows and columns)?**
- **What are column names for the Students dataset?**
- **Extract the column containing data for population. How many students are from Tampere?**
- **Extract the tenth row of the dataset. What is the shoesize of this person?**
- **Extract the rows 25-29. What is the gender of these persons?**
- **Extract from the data only those females who from Helsinki. How many observations (rows) are you left?**
- **How many males are from Kuopio and Tampere?**

Extracting columns and rows II/II

- **Examine Hygrometer dataset. Notice that the measurements were taken on two different dates (day1 and day2 – each hygrometer was read before and after a few rainy days).**
- **Modify the dataset so that the order of the measurements is retained, but the measurements for the day1 and day2 are in two separate column in the same data frame.**
- **We will later on use this data frame for running certain statistical tests (e.g., paired t-test) that require the data in this format.**

Recoding variables

Making new variables I/

- **There are several ways to recode variables in R.**
- **One way to recode values is to use command `ifelse()`.**
 - `ifelse(Students$shoesize<=40, "small", "large")`
 1. Comparison: is shoesize smaller than 40
 2. If comparison is true, return "small"
 3. If comparison is false, return "large"
- **You can combine several comparisons with logical operators**
 - `ifelse((Students$shoesize<=37 & Students$gender=="female"), "small", "large")`

Making new variables II/

- **If the coding needs to be done in several steps (e.g. we want to assign sho sizes to four classes), a better approach could be the following.**
 - `s<-Students$shoesize`
 - `s[Students$shoesize<=37]<-"minuscule"`
 - `s[Students$shoesize>37 & Students$shoesize<=39]<-"small"`
 - `s[Students$shoesize>39 & Students$shoesize<=43]<-"medium"`
 - `s[Students$shoesize>43]<-"large"`
- **At each step we select the only the observations that fulfill the comparsion.**
 - At the first step, all students who have a shoesize less than or equal to 37 are coded as minuscule.
 - At the second step, all students having shoesize larger than 37 but smaller than or equal to 39 are coded as small.
 - And so forth.

Exercise VI

Making new variables

- **Make a new vector of the shoesize measurements (extract that column from the data).**
- **Code the shoesize as it was done on the previous slides (in the range minuscule...large).**
- **Turn this character vector into a factor. Make the factor ordered so that the order of the factor levels is according to the size (minuscule, small, medium, large).**
- **Add this new factor to the Students dataset (make a new data frame).**

Day 2

Topics

- **Data exploration**
- **Graphics in R**
- **Wrap-up of the first half of the course**

Exploration

Exploration – first step of analysis

- **Usually the first step of a data analysis is graphical data exploration**
- **The most important aim is to get an overview of the dataset**
 - Where is data centered?
 - How is the data spread (symmetric, skewed...)?
 - Any outliers?
 - Are the variables normally distributed?
 - How are the relationships between variables:
 - Between dependent and independents
 - Between independents
- **Graphical exploration complements descriptive statistics**

Variable types

➤ **Continuous (vectors in R)**

- Height
- Age
- Degrees in centigrade

➤ **Categorical (factors in R)**

- Make of a car
- Gender
- Color of the eyes

Exploration – methods I/II

➤ **Single continuous variable**

- Plots: boxplot, histogram (density plot, stem-and-leaf), normal probability plot, stripchart
- Descriptives: mean, median, standard deviation, five number summary

➤ **Single categorical variable**

- Plots: contingency table, stripchart, barplot
- Descriptives: mode, contingency table

➤ **Two continuous variables**

- Plots: scatterplot
- Descriptives: individually, same as for a single variable

➤ **Two categorical variables**

- Plots: contingency table, mosaic plot
- Descriptives: individually, same as for a single variable

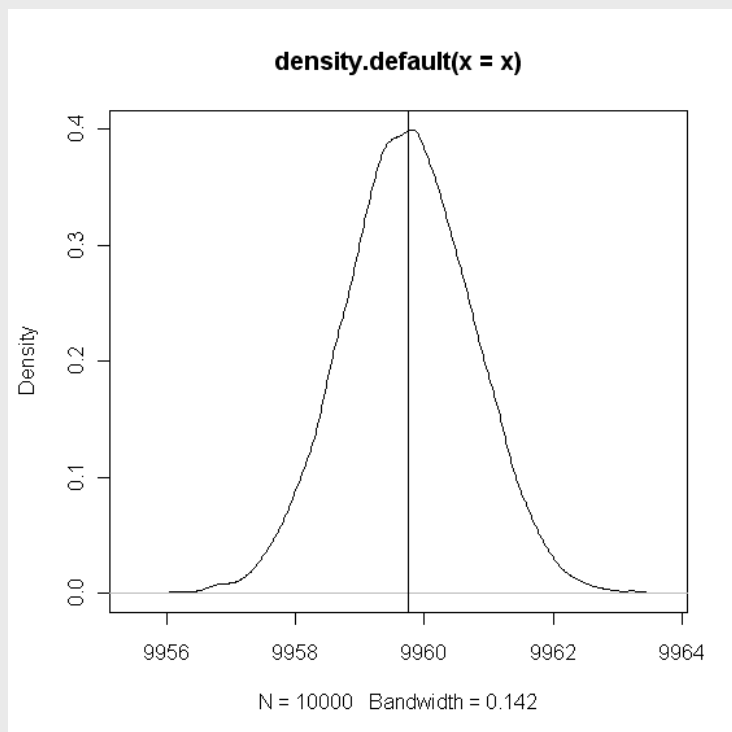
Exploration – methods II/II

- **One continuous, one categorical variable**
 - Plot: boxplot, histogram, but for each category separately
 - Descriptives: mean, median, sd..., for each category separately
- **Several continuous and / or categorical variables**
 - Plots: pairwise scatterplot, mosaic plot
 - Descriptives: as for continuous or categorical variables

Descriptive statistics

Mean

➤ **Mean** = sum of all values / the number of values



Standard deviation and variance

- **SD = each observation's squared difference from the mean divided by the number of observation minus one.**

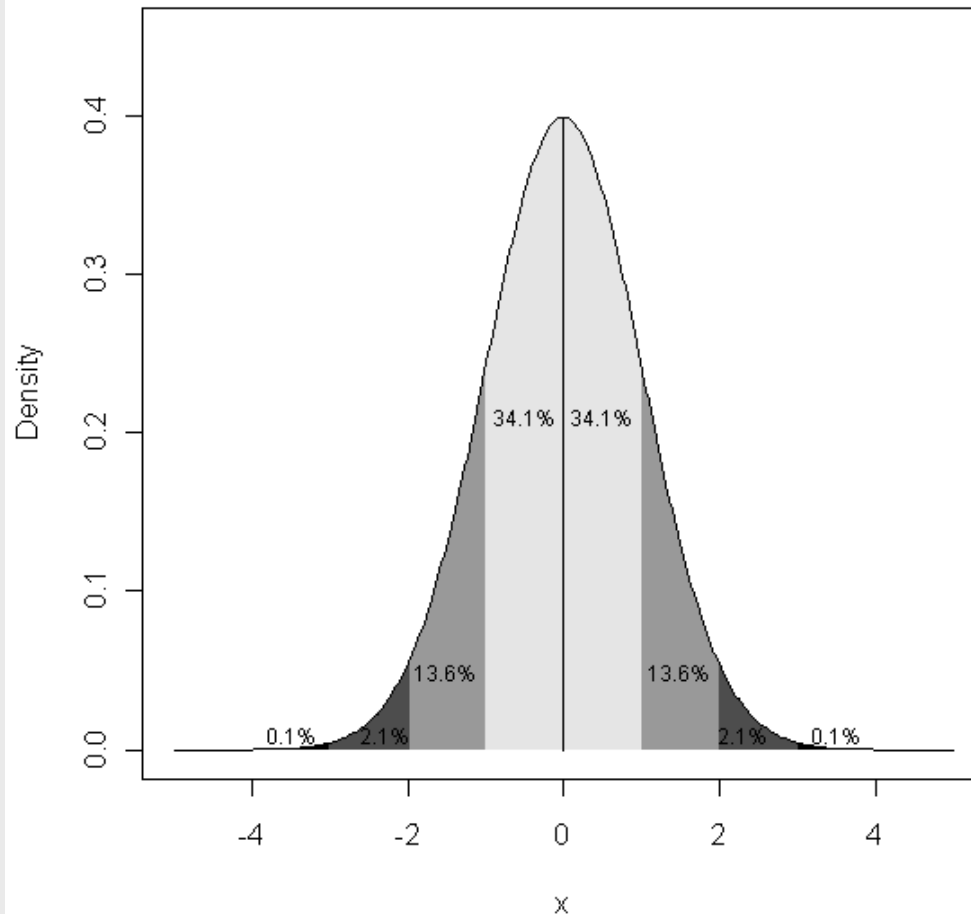
$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2},$$

- Has the same unit as the original variable

- **Var = SD*SD = SD^2**

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2 = \frac{1}{n-1} \sum_{i=1}^n y_i^2 - \frac{n}{n-1} \bar{y}^2,$$

Normal distribution I/III



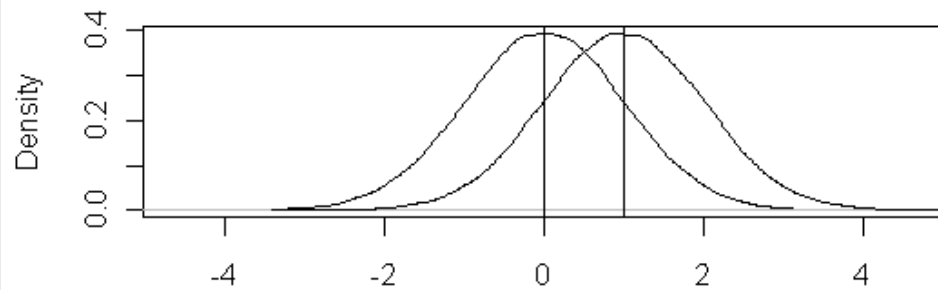
- **Some measurements are normally distributed in the real-world**
 - Height
 - Weight
- **Means of observations taken from otherwise distributed data are also normally distributed**
- **Hence, many descriptives, and statistical tests have been devised on the assumption of normality**

Normal distribution II/III

- **Normal distribution are described by two statistics:**
 - Mean
 - Standard deviation
- **These two are enough to tell:**
 - Where is the peak (center) of the distribution located
 - How the data are spread around this peak

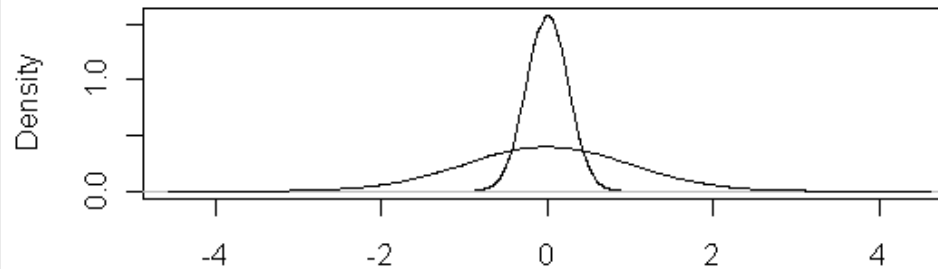
Normal distribution III/III

Different means, same SD



N = 100000 Bandwidth = 0.09013

Same means, different SDs

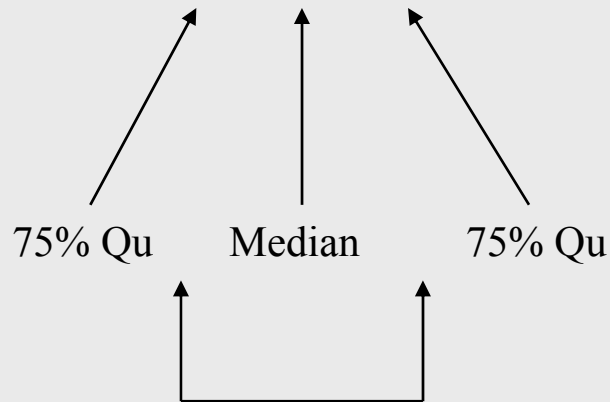


N = 100000 Bandwidth = 0.02251

Quartiles

➤ **1st quartile(25%), Median (50%), and 3rd quartile (75%)**

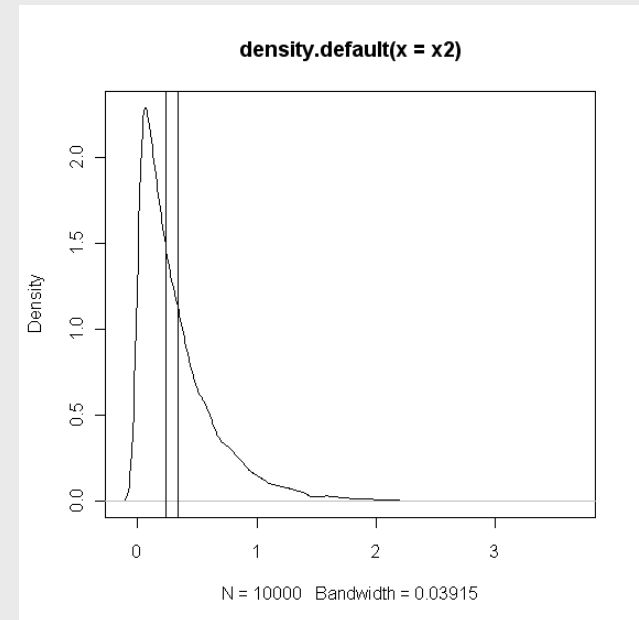
➤ **1 2 3 4 5 6 7 8 9**



Interquartile range (IQR)

➤ **Fivenum summary:**

- Minimum (1), 1st Quartile (3), Medium (5), 3rd Quartile (7), maximum (9)



What if distribution is skewed or there are outliers/deviant observation?

➤ **Use nonparametric alternatives to descriptives**

- Median instead of mean
- Inter-quartile range instead of standard deviation

Summary of a continuous variable I/II

➤ **summary()**

- `x<-rnorm(100)`
- `summary(x)`

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.005561 0.079430 0.202900 0.310300 0.401000 1.677000

➤ **median(x)**

➤ **mean(x)**

➤ **min(x)**

➤ **max(x)**

➤ **quantile(x, probs=c(0.25, 0.75))**

- 1st and 3rd quartiles

Summary of a continuous variable II/II

- **IQR(x)** **# inter-quartile range**
- **mad(x)** **# robust alternative to IQR**
- **sd(x)** **# standard deviation**
- **var(x)** **# variance**
 - **sd(x)^2**
- **table()** **# Makes a table (categ. var.)**

Outliers and missing values

What are these outliers then?

➤ **Outliers**

- Technical errors
 - The measurement is too high, because the machinery failed
- Coding errors
 - Male = 0, Female=1
 - Data has some values coded with 2

➤ **Deviant observations**

- Measurements that are somehow largely different from others, but can't be treated as outliers
- If the observation is not definitely an outlier, better treat it as a deviant observation, and keep it in the data

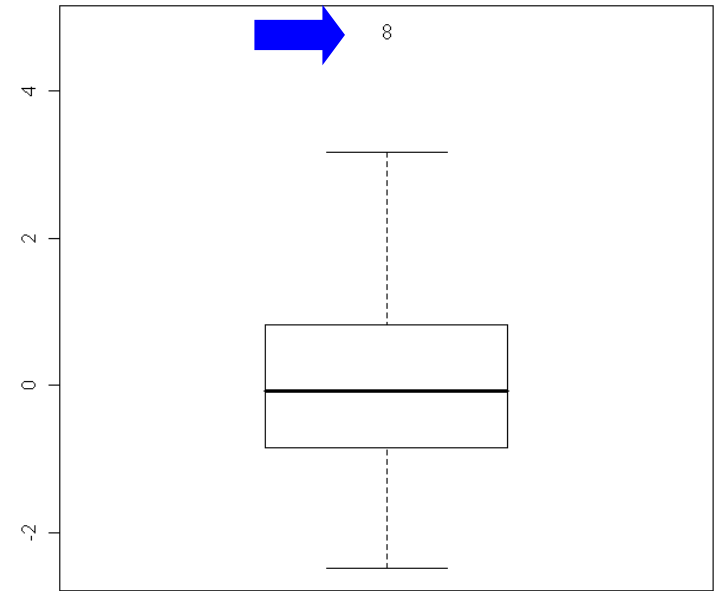
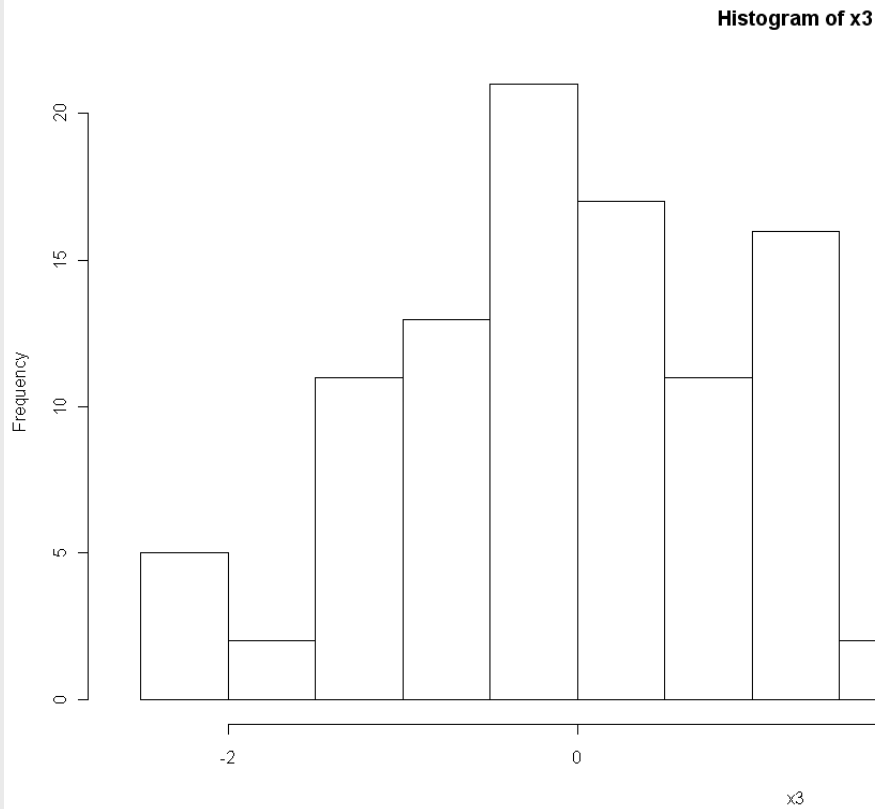
Outliers

gender

0	1	2
11	8	1

- **What are those with gender coded as 2?**
- **Probably a typing error**
 - What if they are missign values (gender is unknown)?
- **If a typing error, should be checked from the original data**
- **If a missing value, should be coded as missing value**
 - We will come to this shortly

Deviant observations



Missing values

- **Missing values are observation that really are missing a value**
 - Some samples were not measured during the experiment
 - Some students did not answer to certain questions on the feedback from
- **If the sample was measured, but the results was very low or not detectable, it should be coded with a small value (half the detection limit, or zero, or something)**
- **So, no measurement and measurement, but a small result, should be coded separately**

Missing values in R I/II

- **In R missing values are coded with NA**
 - NA = not available
- **Although it is worth treating missing measurements as missing values, they tend to interfere with the analysis**
 - Many graphical, descriptive, and testing procedures fail, if there are missing values in the data
- **An example**
 - `x<-c(NA, rnorm(10))`
 - `mean(x)`
 - `[1] NA`

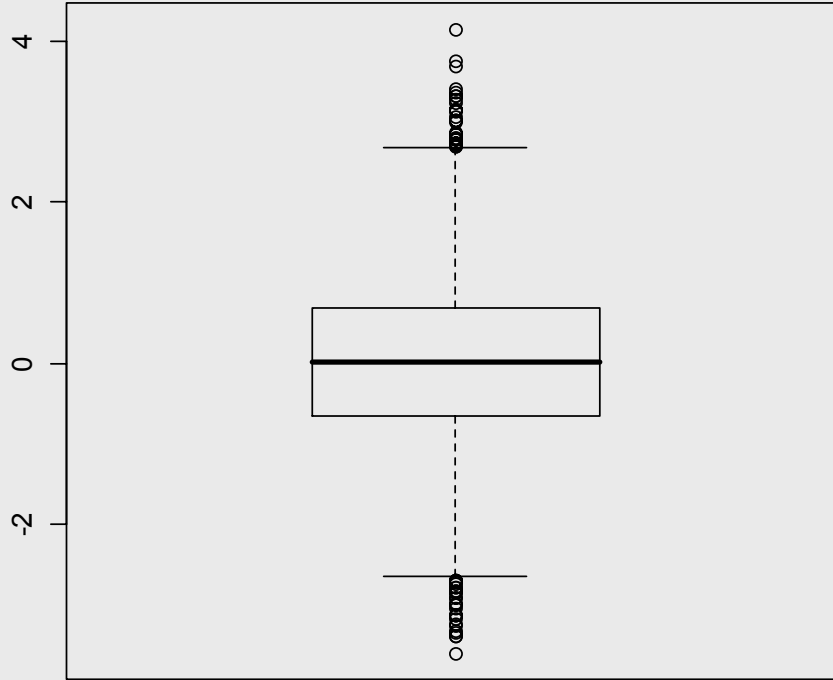
Missing values in R II/II

- **The most simple way to treat missing values is to delete all cases (rows) that contain at least one missing value.**
- **For vector this means just removing the missing values:**
 - `x2<-na.omit(x)`
 - `mean(x2)`
 - `[1] -0.1692371`
- **There are other ways to treat missing values, such as imputation, where the missing values are recoded with, e.g., the mean of the continuous variable, or with the most common observation, if the variable is categorical.**
 - `x2[is.na(x2)]<-mean(na.omit(x))`

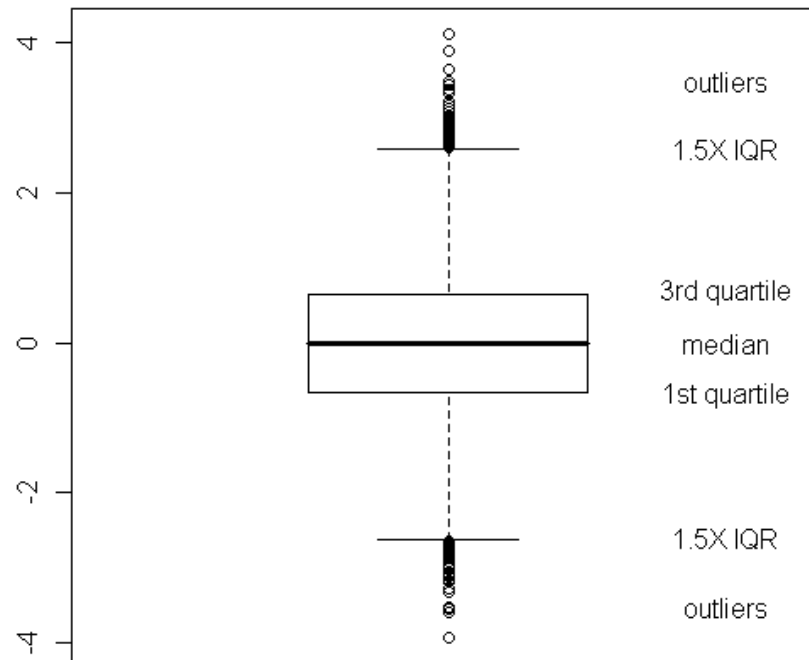
Graphical methods

Continuous variables

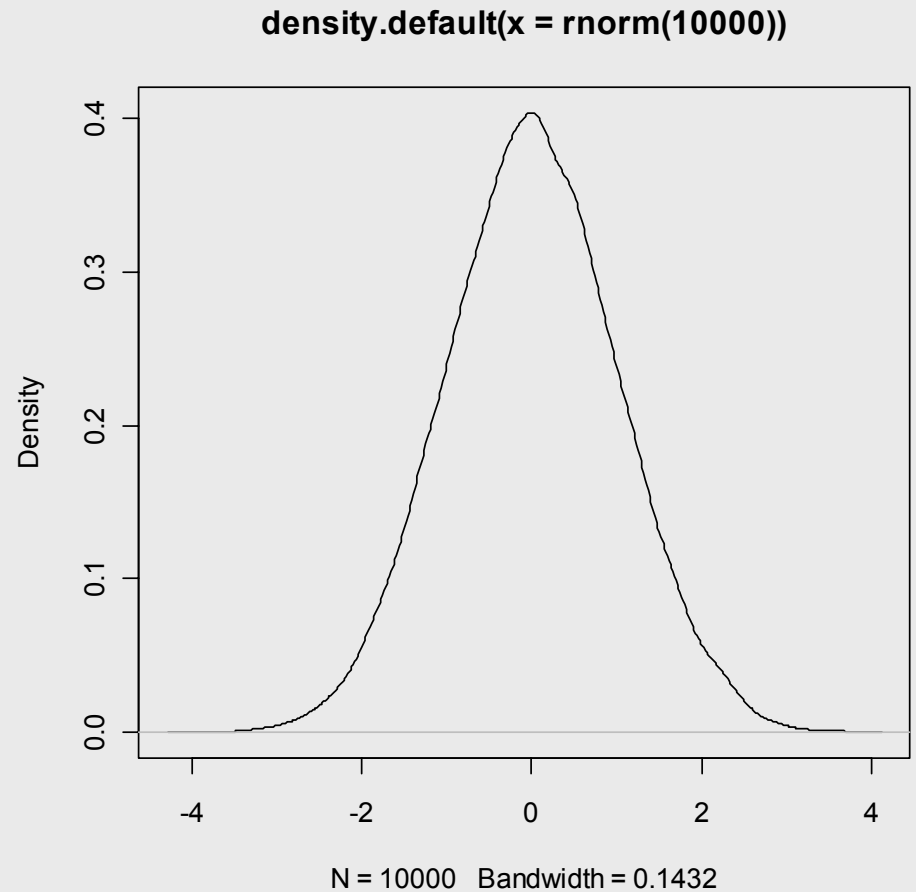
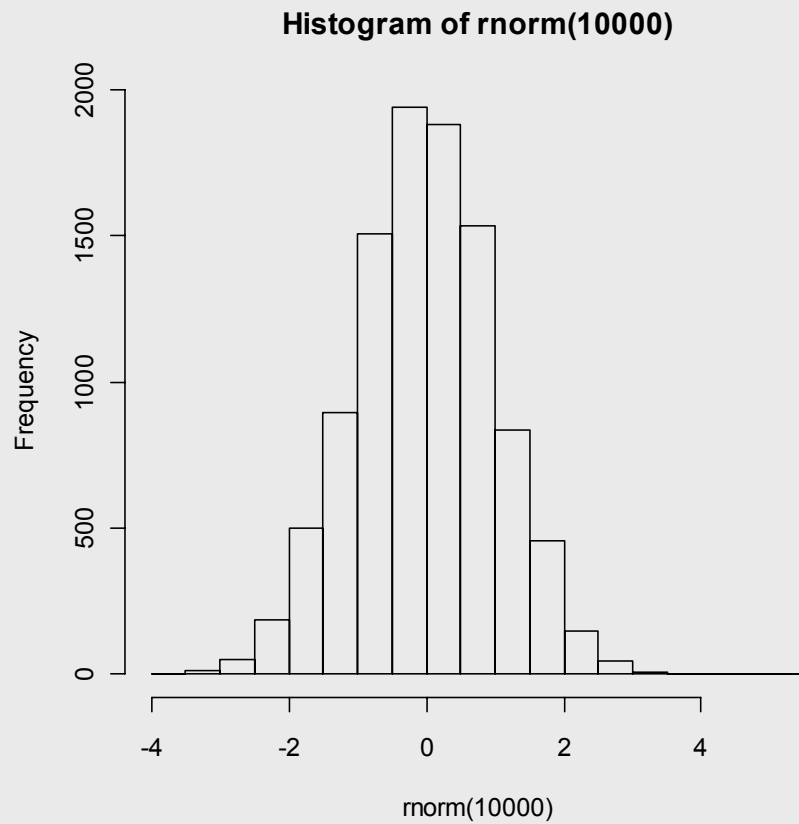
Boxplot



Link between quartiles and boxplot

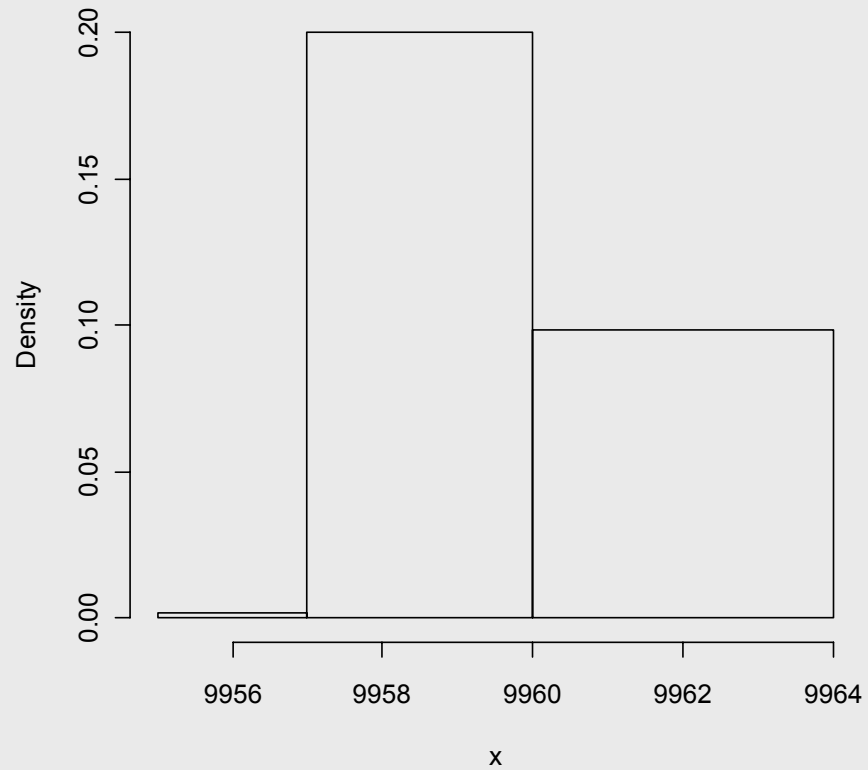


Histogram I/II

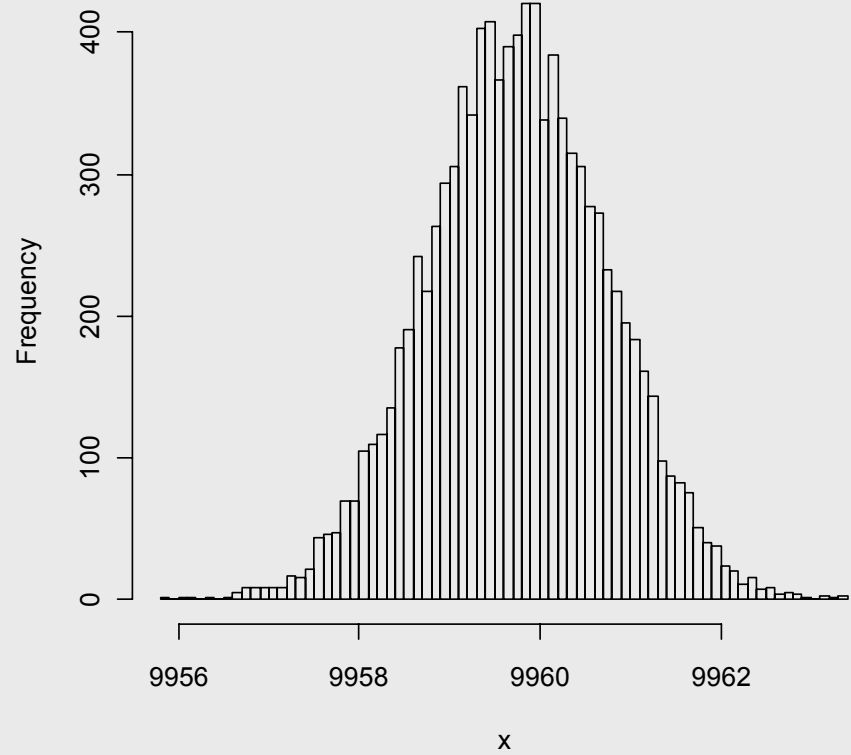


Histogram II/II

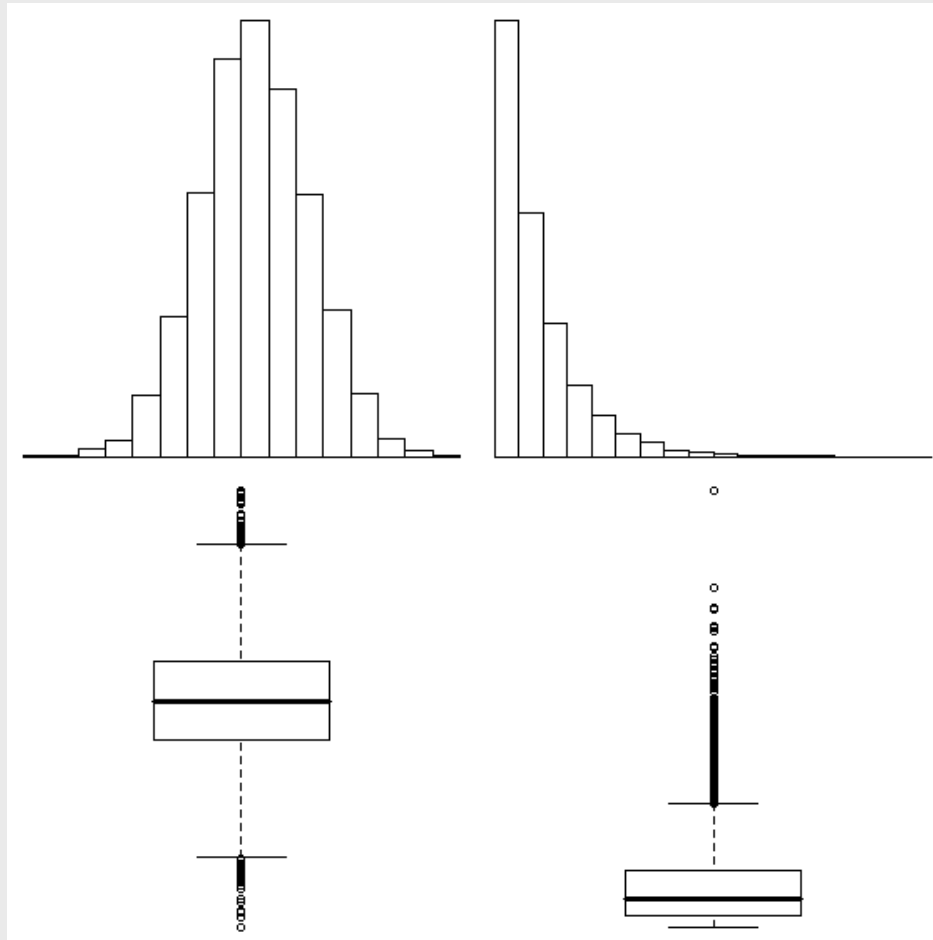
Histogram of x



Histogram of x



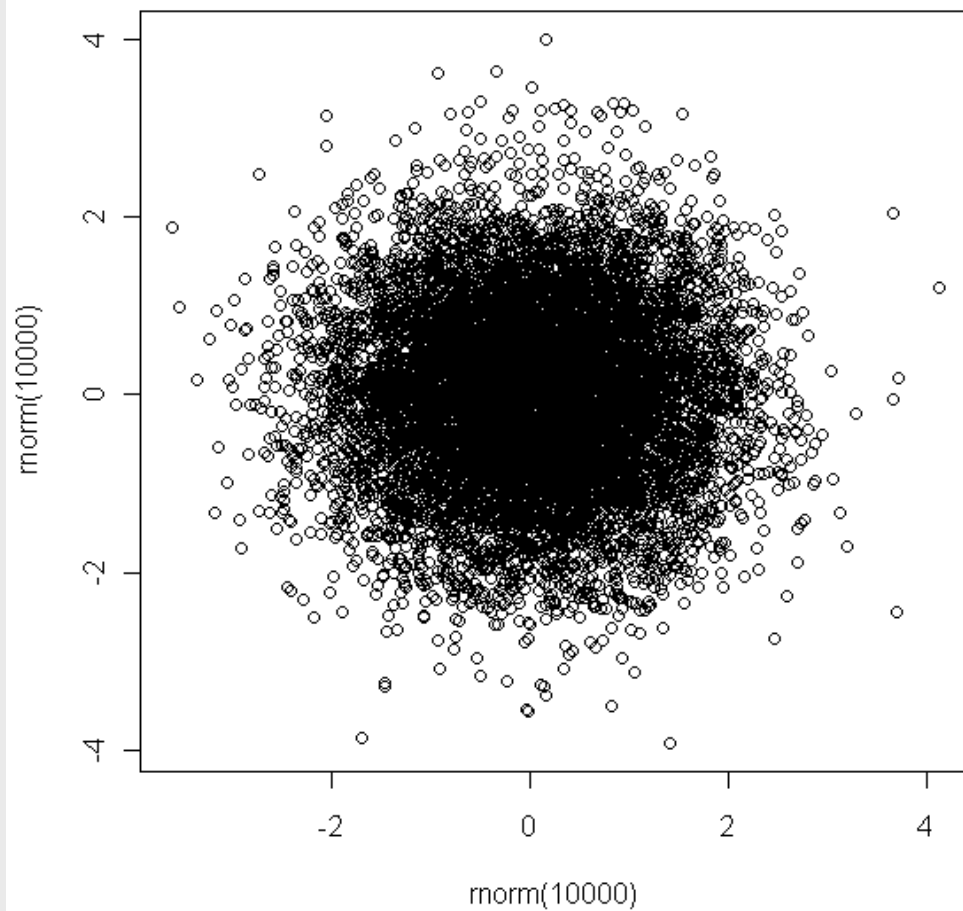
Link between histogram and boxplot



Stem-and-leaf plot

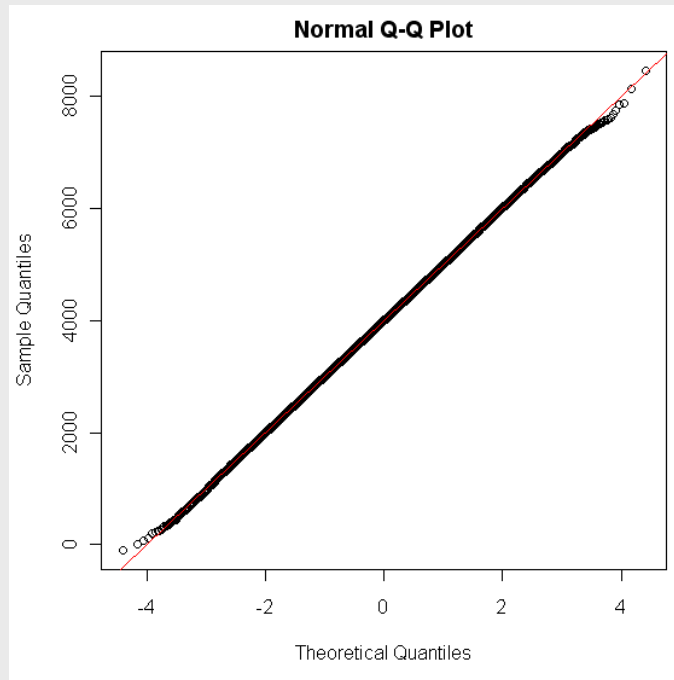
- The decimal point is at the |
- -2 | 90
- -1 | 88876664322221000
- -0 | 998886665555544444333322222211110
- 0 | 001111111112222334445667778888899
- 1 | 00112334455569
- 2 | 3

Scatterplot

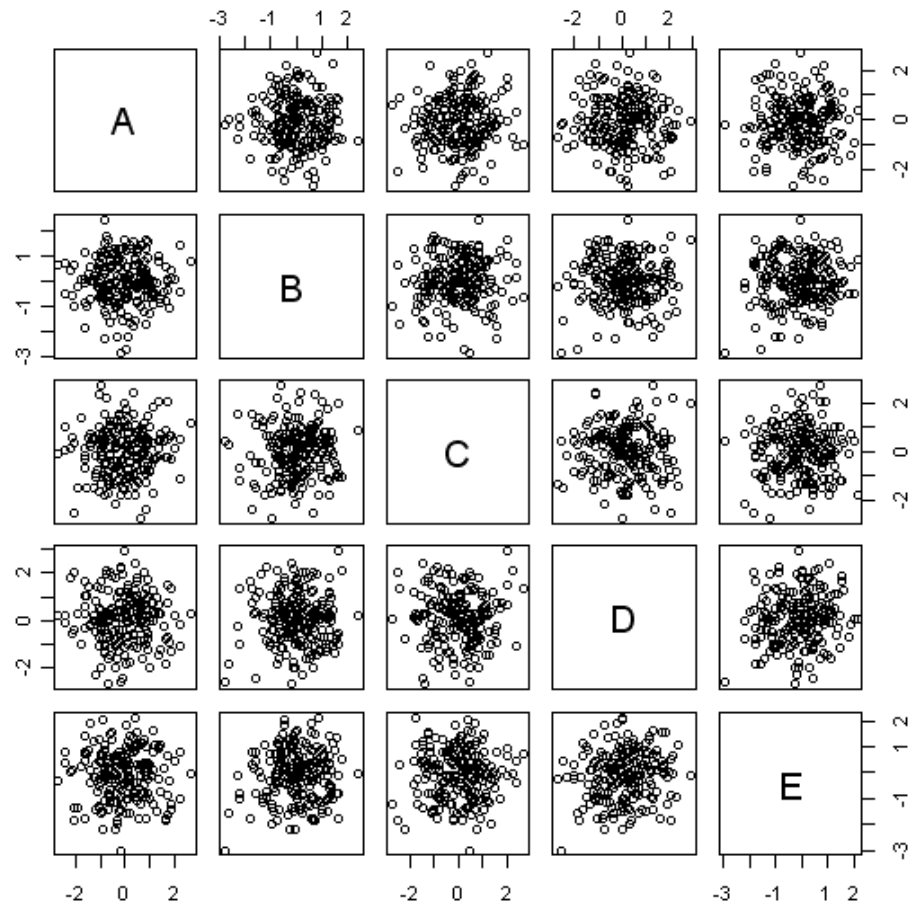


QQ-plot

- **QQ-plot is a plot that can be used for graphically testing whether a variable is normally distributed.**
 - Normal distribution is an assumption made by many statistical procedures.

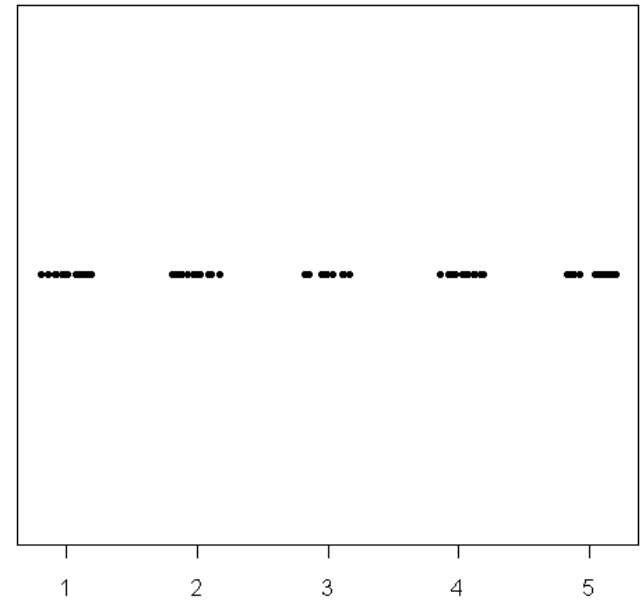
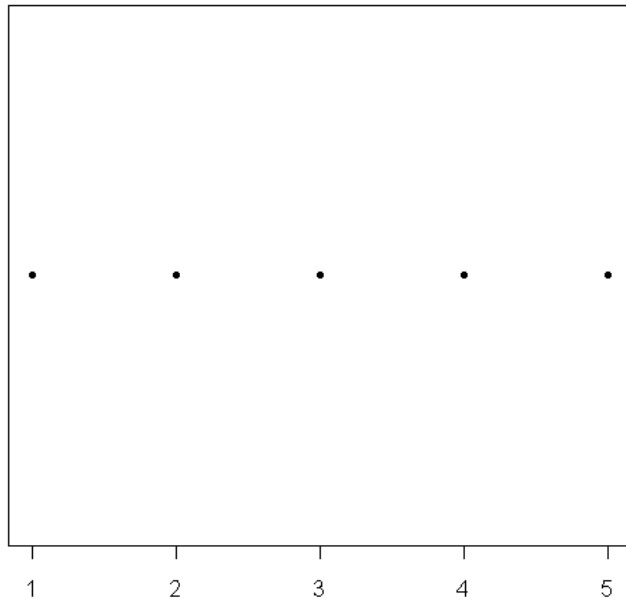


Pairwise scatterplot

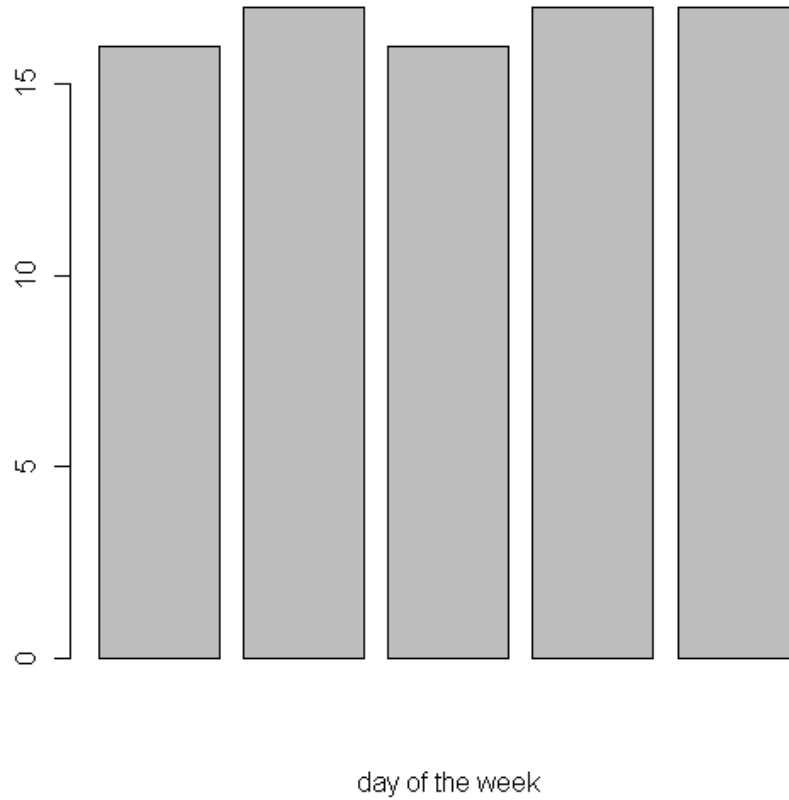


Categorical variables

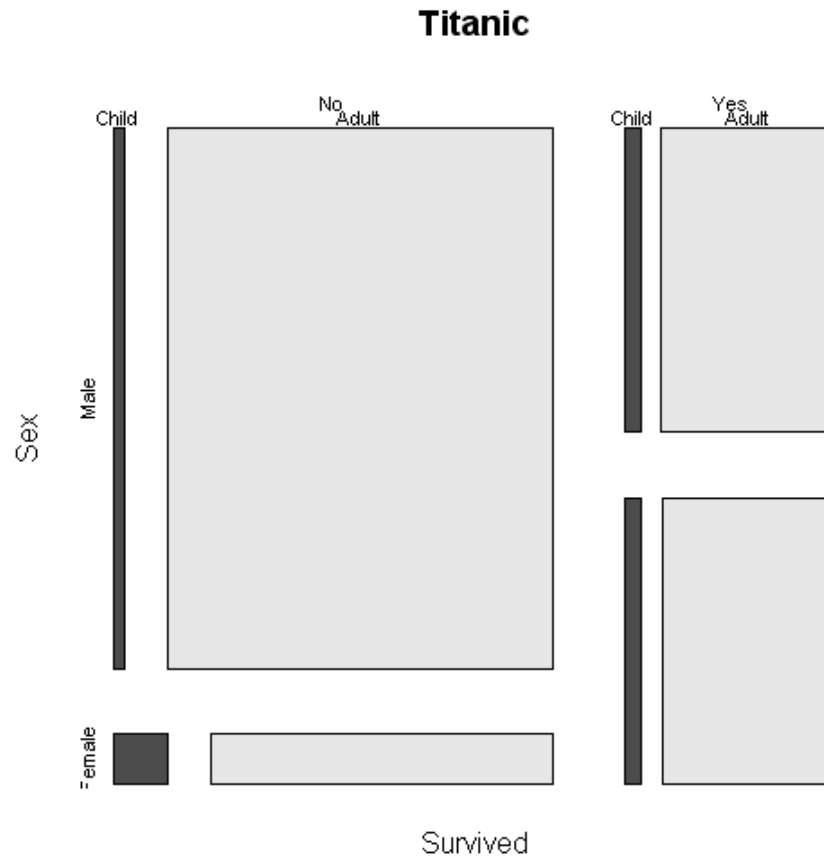
Stripchart



Barchart



Mosaicplot



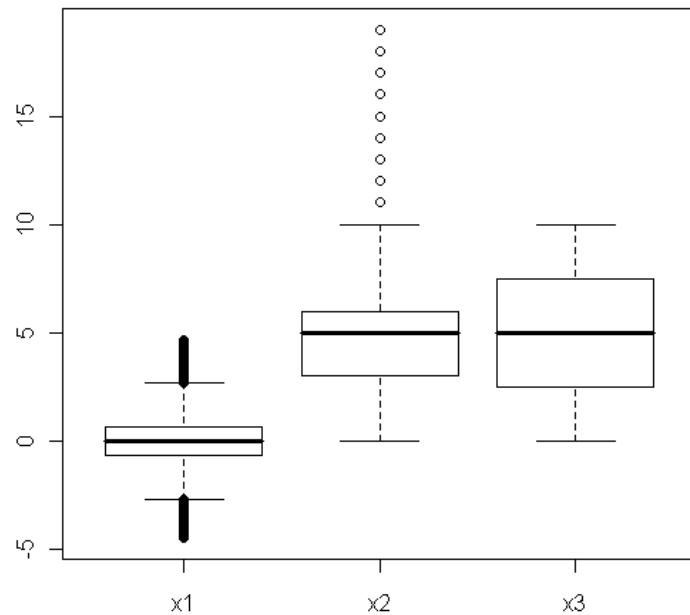
Contingency table

	January	February	March	April
Friday	4	5	3	4
Monday	4	4	4	5
Thursday	4	4	4	4
Tuesday	5	4	4	4
Wednesday	5	4	4	4

Exercise VII

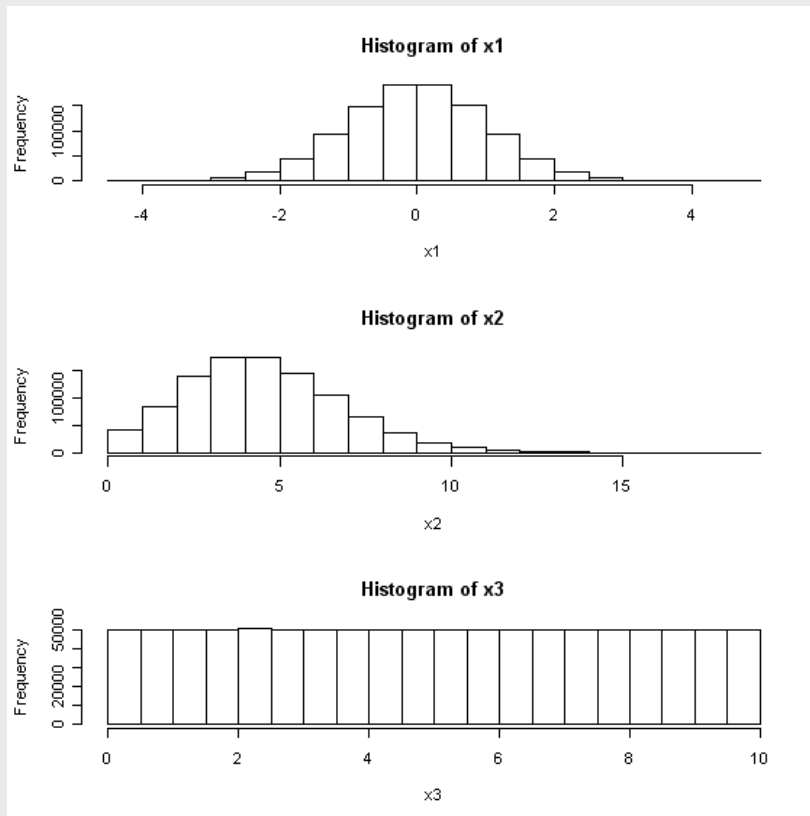
Checking distributions

➤ Are these data normally distributed?



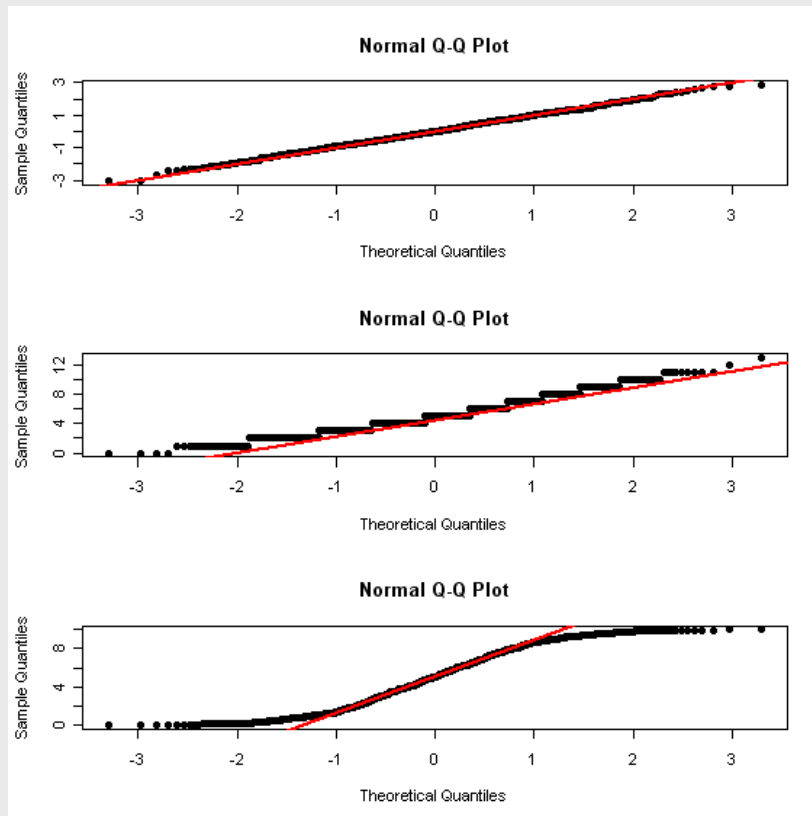
Checking distributions

➤ Are these data normally distributed?



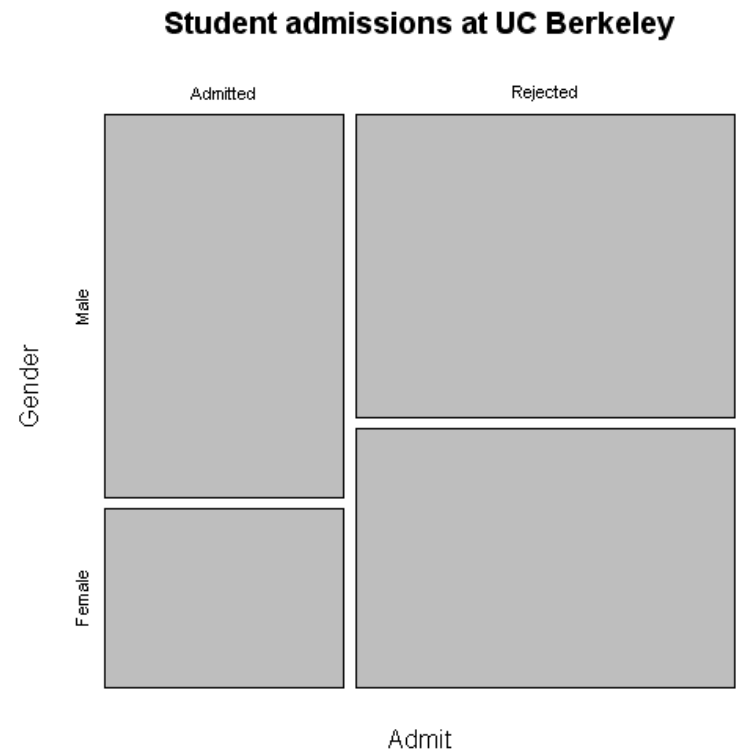
Checking distributions

➤ Are these data normally distributed?



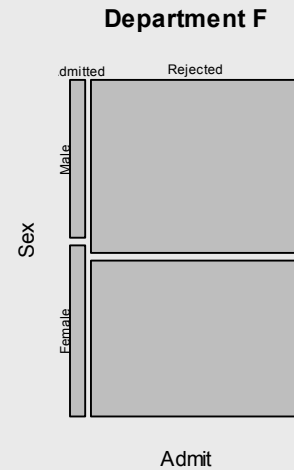
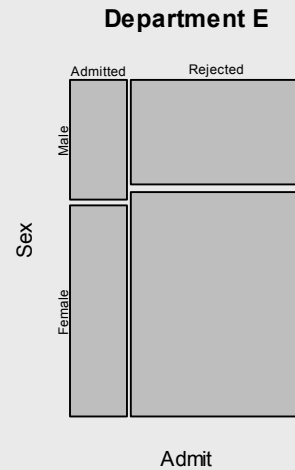
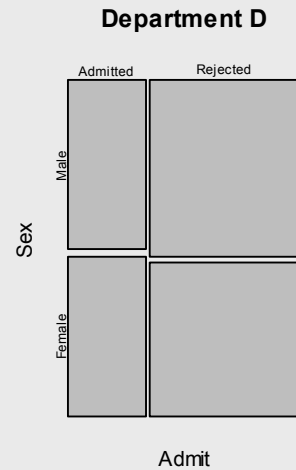
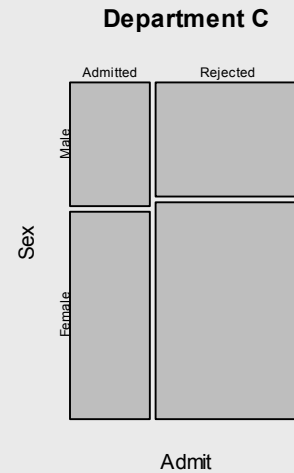
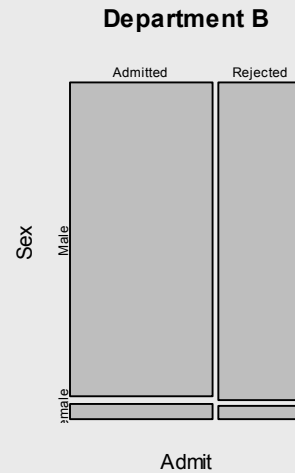
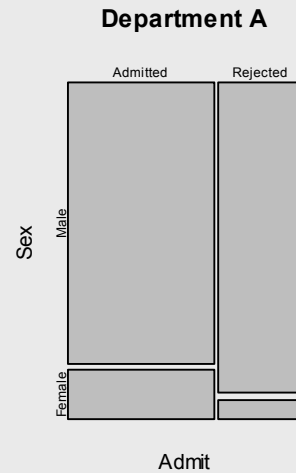
UCB admissions

- **Claim: UCB discriminates against females.**
 - I.e., More females than males are rejected, and don't get admitted to the university.
 - Does UCB discriminate?



➤ **Claim: UCB discriminates against females.**

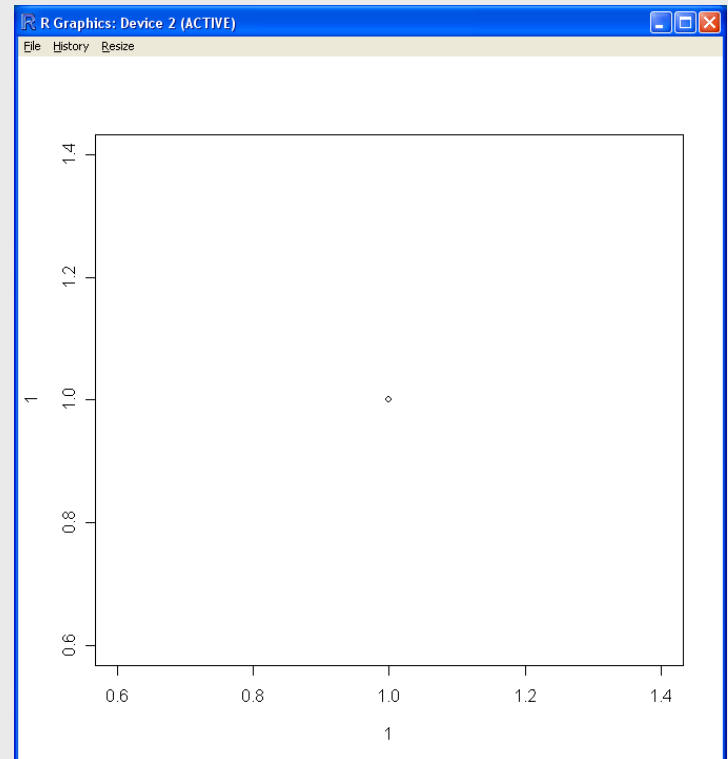
- Does it?



Graphics in R

Basic idea

- All graphs in R are displayed on a graphical device.
- If no device is open when the plotting command is called, a new one is opened, and the image is displayed in it.
- Graphics device is simply a new window that displays the graphic.
- Graphic device can also be a file where the plot is written.
 - Open it
 - Make the plot
 - Close it



Traditional graphics commands in R

➤ High level graphical commands create the plot

- `plot()` # Scatter plot, and general plotting
- `hist()` # Histogram
- `stem()` # Stem-and-leaf
- `boxplot()` # Boxplot
- `qqnorm()` # Normal probability plot
- `mosaicplot()` # Mosaic plot

➤ Low level graphical commands add to the plot

- `points()` # Add points
- `lines()` # Add lines
- `text()` # Add text
- `abline()` # Add lines
- `legend()` # Add legend

➤ Most command accept also additional graphical parameters

- `par()` # Set parameters for plotting

Graphical parameters in R

➤ **par()**

- `cex` # font size
- `col` # color of plotting symbols
- `lty` # line type
- `lwd` # line width
- `mar` # inner margins
- `mfrow` # splits plotting area (mult. figs. per page)
- `oma` # outer margins
- `pch` # plotting symbol
- `xlim` # min and max of X axis range
- `ylim` # min and max of Y axis range

A few worked examples

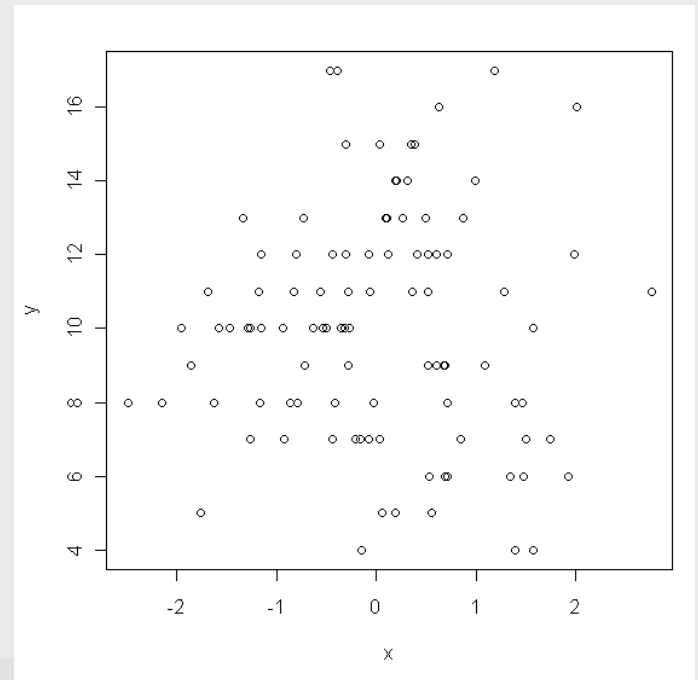
Drawing a scatterplot in R I/V

➤ Let's generate some data

- `x<-rnorm(100)`
- `y<-rpois(100, 10)`
- `g<-c(rep("horse", 50), rep("hound",50))`

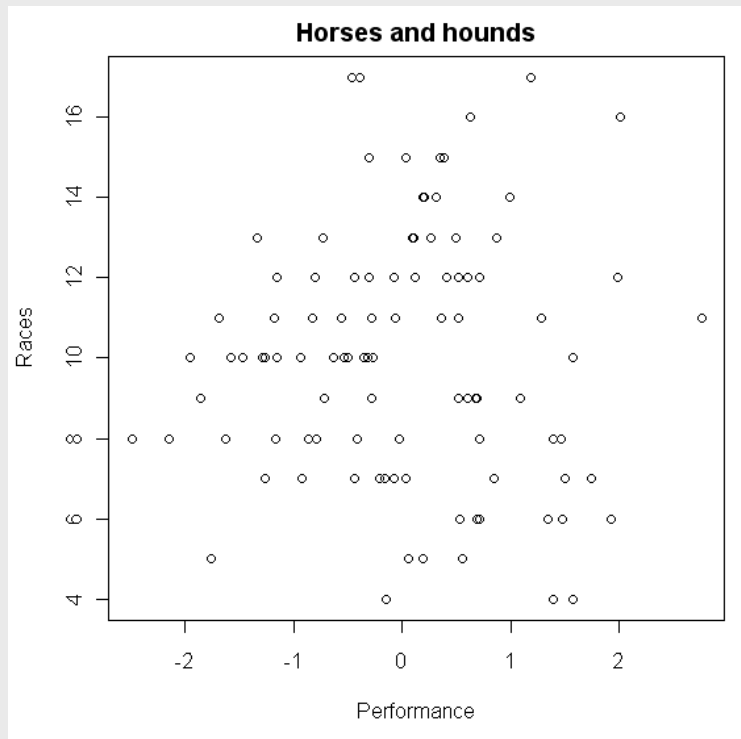
➤ Simple scatter plot

- `plot(x, y)`



Adding a title and axis labels II/V

- `plot(x, y, main="Horses and hounds",
xlab="Performance", ylab="Races")`



Drawing a scatterplot in R III/V

➤ Coloring spots according to the group (horse or hound) they belong to

- `cols<-ifelse(g=="horse", "Black", "Red")`
- `plot(x, y, main="Horses and hounds", xlab="Performance", ylab="Races", col=cols)`

`plot (graphics)`

Generic X-Y Plotting

Description

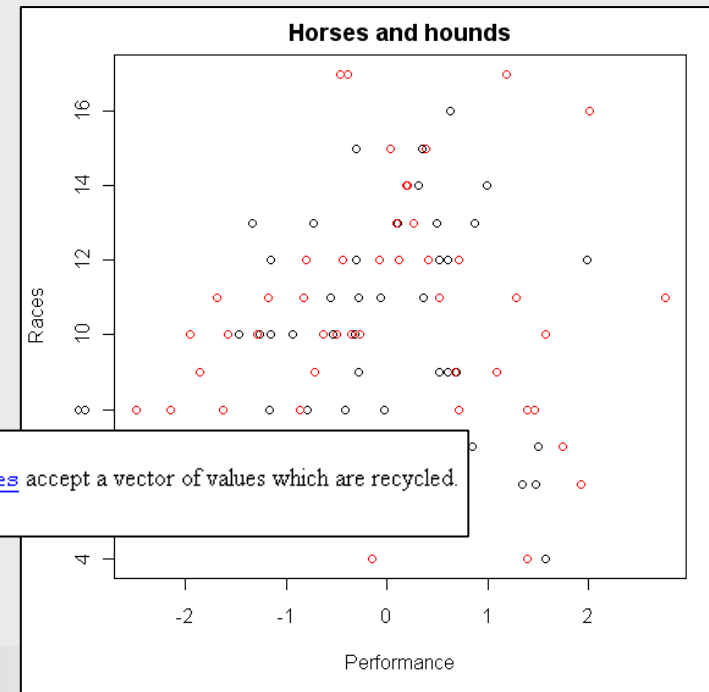
Generic function for plotting of R objects. For more details about the graphical parameter arguments, see [par](#).

Usage

`plot(x, y, ...)`

`col`

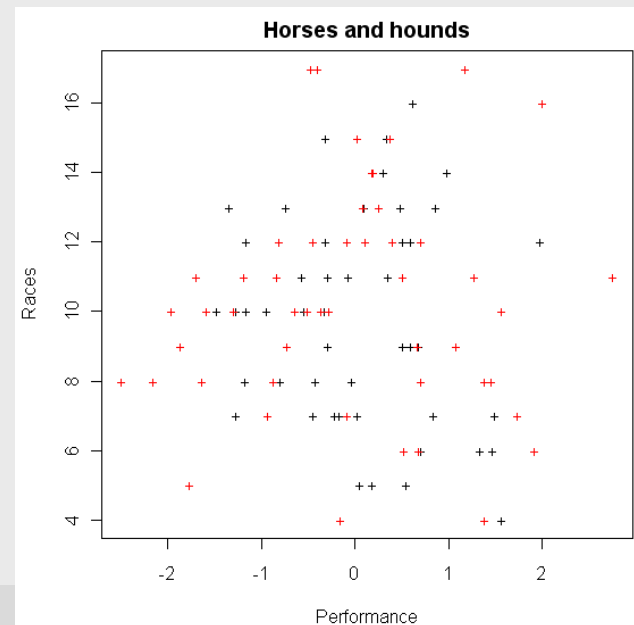
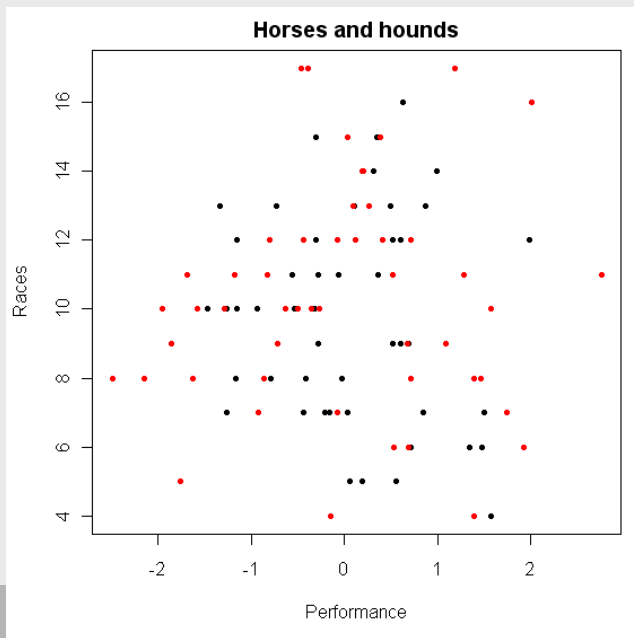
A specification for the default plotting color. See section 'Color Specification'. (Some functions such as [lines](#) accept a vector of values which are recycled. Other uses will take just the first value if a vector of length greater than one is supplied.)



Drawing a scatterplot in R IV/V

➤ Changing the plotting symbol

- `plot(x, y, main="Horses and hounds", xlab="Performance", ylab="Races", col=cols, pch=20)`
- `plot(x, y, main="Horses and hounds", xlab="Performance", ylab="Races", col=cols, pch="+")`



Drawing a scatterplot in R V/V

➤ **Saving the image**

- Menu: File -> Save As -> JPEG / BMP / PDF / postscript

➤ **Directing the plotting to a file**

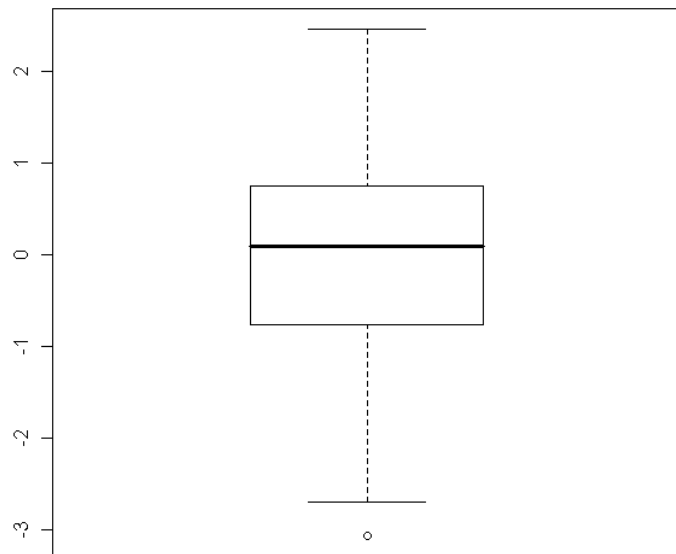
- `pdf("hnh.pdf")`
- `plot(x, y, main="Horses and hounds", xlab="Performance", ylab="Races", col=cols, pch=20)`
- `dev.off()`

➤ **Setting the size of the image in inches**

- `pdf("hnh.pdf", width=7, height=7)`
- `plot(x, y, main="Horses and hounds", xlab="Performance", ylab="Races", col=cols, pch=20)`
- `dev.off()`

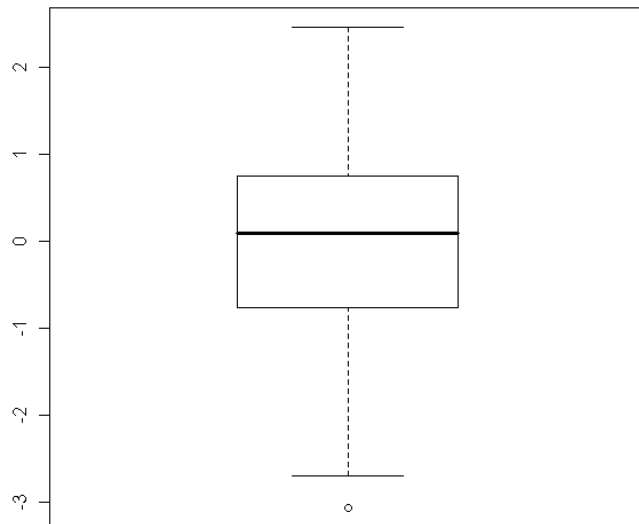
Drawing a box plot I/III

- `x<-rnorm(100)` # x is a vector
- `boxplot(x)` # makes a boxplot



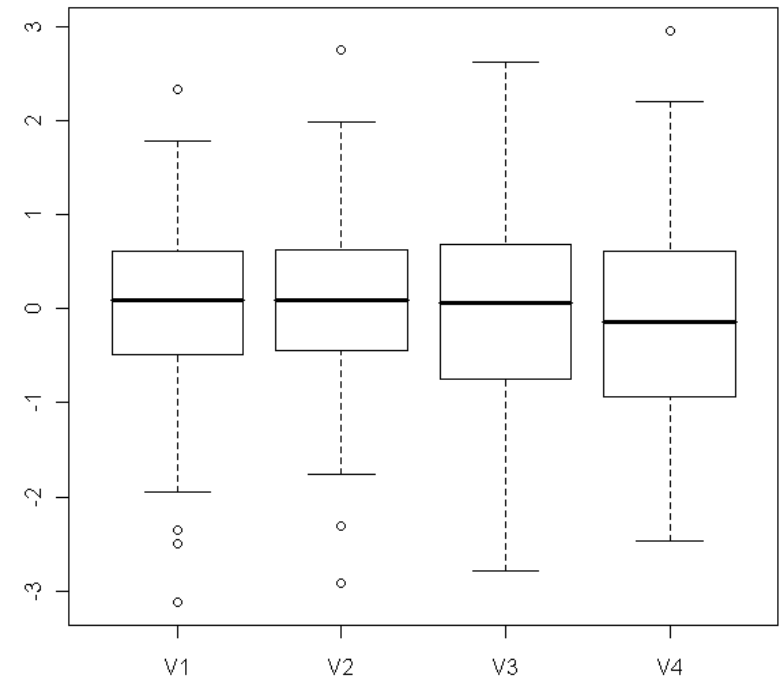
Drawing a boxplot II/III

- **# x is a matrix**
- **x<-matrix(ncol=4, nrow=100, data=rnorm(400))**
- **boxplot(x) # makes a boxplot**



Drawing a boxplot III/III

- **# x is a matrix**
- **x<-matrix(ncol=4, nrow=100, data=rnorm(400))**
- **# x is converted a data frame first**
- **x<-as.data.frame(x)**
- **# makes a boxplot**
- **boxplot(as.data.frame(x))**

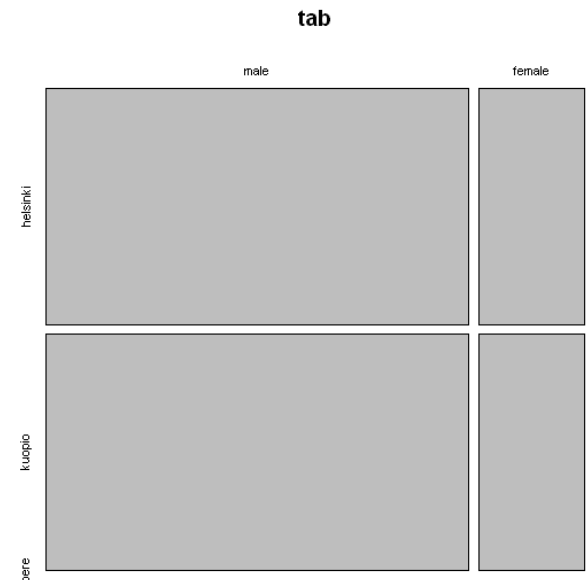


Drawing a mosaic plot I/II

- **Two or more categorical variables**
- **First make a contingency table using `table()`.**
- **Then plot the table using `mosaicplot()`.**

- For example:

```
> tab<-table(s$gender, s$population)
      helsinki kuopio tampere
male          4      4        0
female        1      1        0
> mosaicplot(tab)
```



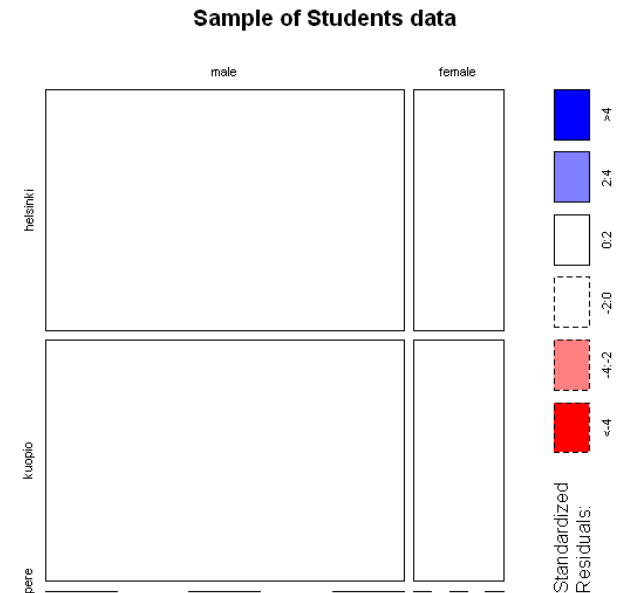
Drawing a mosaic plot II/II

➤ Adding title

```
> mosaicplot(tab, main="Sample of Students data")
```

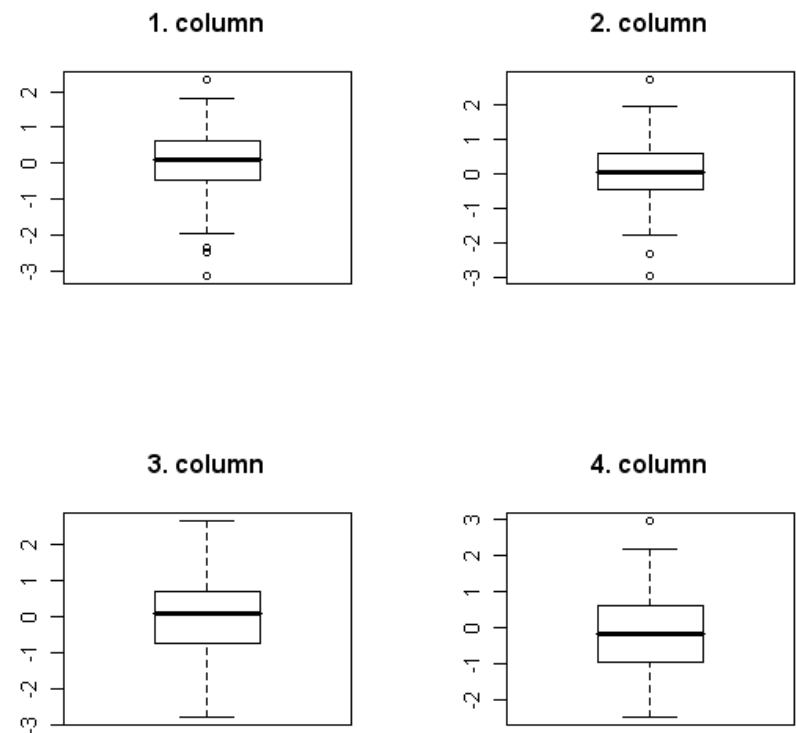
➤ Coloring by residuals

```
> mosaicplot(tab, main="Sample of Students data",  
  shade=T)
```



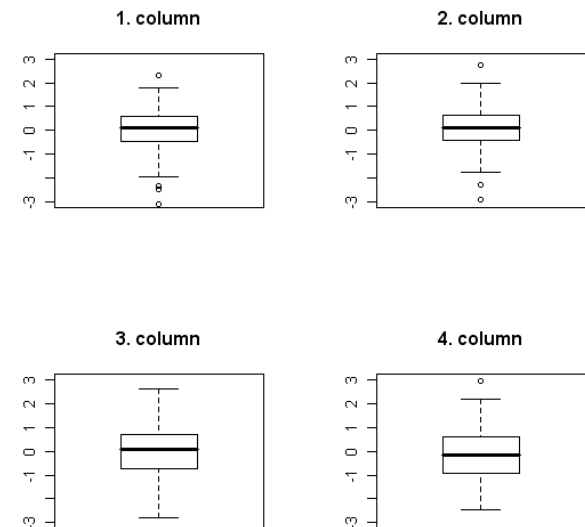
Putting several graphs on the same page I/II

- **# 2*2 figures on the same page**
- **# Setting graphical parameters**
- **`par(mfrow=c(2,2), xlim=c(-3,3))`**
- **# plotting**
- **# Every box plot has a title**
- **`boxplot(x[,1], main="1. column")`**
- **`boxplot(x[,2], main="2. column")`**
- **`boxplot(x[,3], main="3. column")`**
- **`boxplot(x[,4], main="4. column")`**



Putting several graphs on the same page II/II

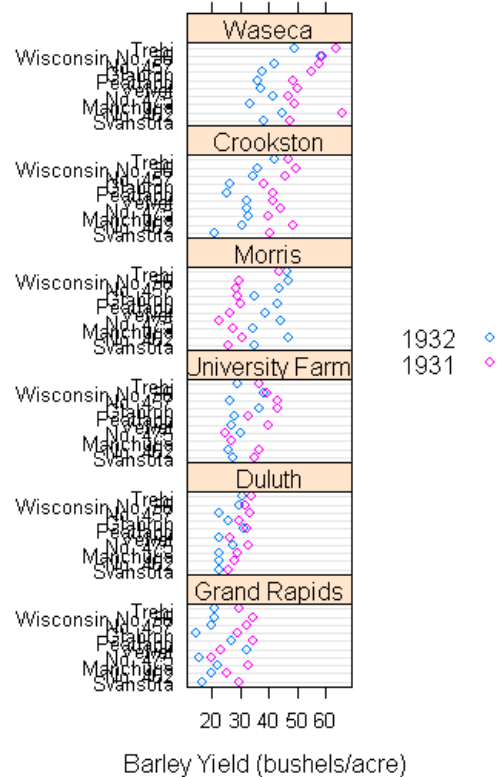
- **# 2*2 figures on the same page**
- **# Setting graphical parameters**
- **`par(mfrow=c(2,2), xlim=c(-3,3))`**
- **# plotting**
- **# Every box plot has a title and a same range**
- **`boxplot(x[,1], main="1. column", ylim=c(-3,3))`**
- **`boxplot(x[,2], main="2. column", ylim=c(-3,3))`**
- **`boxplot(x[,3], main="3. column", ylim=c(-3,3))`**
- **`boxplot(x[,4], main="4. column", ylim=c(-3,3))`**



Trellis graphics

Trellis graphics

➤ Multipanel functions for displaying data



Trellis graphics commands

➤ High level commands:

- `bwplot()` # boxplot
- `densityplot()` # "smoothed histogram"
- `histogram()` # histogram
- `xyplot()` # scatter plot

➤ Traditional graphics take arguments as

- `plot(x, y)` # scatter plot

➤ Trellis graphics take arguments as a formula

- `plot(y~x)` # scatter plot
- In formula the y (what is predicted) is on the left, then comes tilde, and then the predictors

Trellis scatter plot

➤ **Let's generate some data**

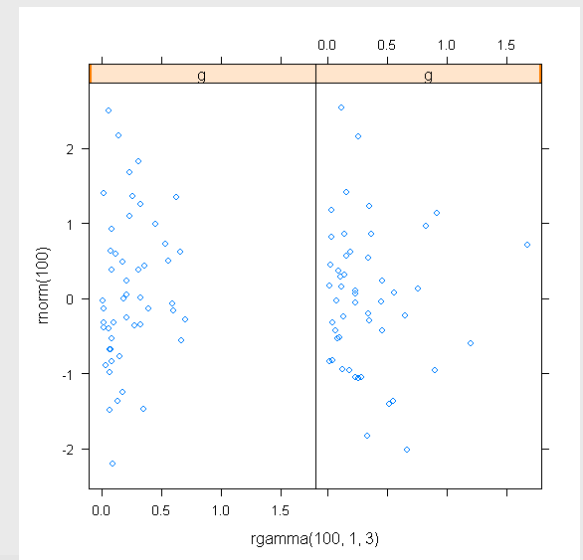
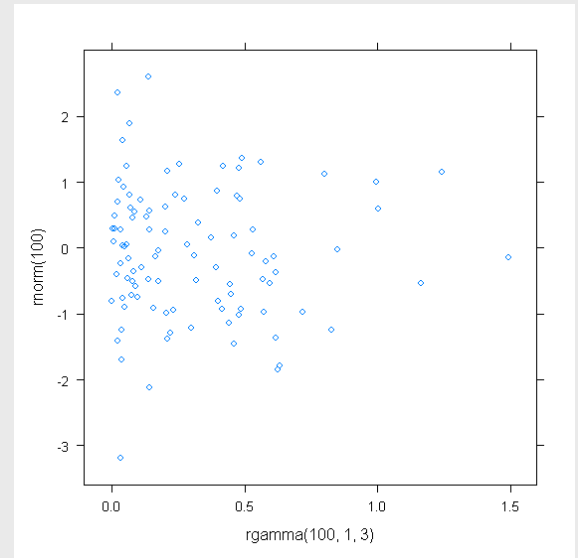
- `y<-rnorm(100)`
- `x<-rgamma(100, 1, 3)`
- `g<-c(rep(1,50), rep(2,50))`

➤ **A simple scatter plot**

- `library(lattice)`
- `xyplot(y~x)`

➤ **Split according to g**

- `xyplot(y~x | g)`



Graphics systems in R

- **Traditional graphics**
 - Package graphics
- **Grid graphics**
 - Package grid
- **Other systems (built on grid)**
 - Package lattice (Trellis graphics)
 - Package ggplot2

Exercise VII

Descriptive statistics

➤ **Examine the variable height in the Students dataset**

- What is the mean of height?
- What is standard deviation of height?
- What are minimum, maximum and range for height?
- What is the difference of mean heights between males and females?
- What is median of height?
- What is inter-quartile range of height?

➤ **Examine gender and population**

- How many females are there from helsinki?
- Many many times more females there are from Helsinki than males from Helsinki?

Graphical exploration

➤ **Examine the variable height in the Students dataset**

- Make a boxplot for all data
- Check the help file for boxplot to figure out how to split it into several distinct boxplots
 - Make a boxplot to compare males and females
 - Make a boxplot to compare different population
- In your opinion, is the height normally distributed?
 - You can also use `qqnorm()` or `hist()` to get more insight to this.

➤ **Plot a scatterplot of height against shoesize**

- Are there any obviously deviant values?
- Code all males with "o", and all females with "+" (hint: create a new vector using `command ifelse()`)
 - Make a scatterplot of height against shoesize using o/+ (the vector you just created) as the plotting symbol
 - Is there a clear distinction between males and females?
- Create the same plot again, but additionally coding populations with different colors.
 - Start with `cols<-as.vector(Students$population)`
 - Assign a different color to every population in this cols vector
- Why there are no females from Kuopio visible in the plot?
 - Are there any females from Kuopio in the dataset?

Exercise VIII

Wrap-up

- **Go through the lecture slides, exercises and your own notes.**
- **Discuss the things that were left unclear with your pair.**
- **Write the possible questions or requirements for further clarifications on a piece paper. A stack of paper is circulating in the class.**

Day 3

Today's topics

➤ **Philosophy of statistical testing**

➤ **Tests**

- One-group
 - T-test (one-sample t-test)
 - Chi square
- Two-groups
 - T-test (two-sample t-test, paired t-test)
 - F-test
 - Chi square
- More than two groups
 - Analysis of variance (ANOVA)
- Correlation
- Bivariate linear regression

Philosophy of statistical testing

Basic questions

- **Assume that we have collected sample from two groups, say, cancer patients and their healthy controls.**
- **Statistical testing tries answer the question**
 - Can the observed difference (in certain variable) between the groups be explained by chance alone?
 - How significant is this difference?
- **Statistical testing can also be viewed as hypothesis testing, where two different hypothesis are compared**
 - Hypothesis 0: There is no difference between the groups
 - Hypothesis 1: There is a difference between the groups

Statistical significance v. practical significance

- **Comparing two groups of workers exposed to styrene, we found a mean difference of 0.000001 grams.**
 - The result is statistically significant.
 - Is it of practical significance? No. The difference is too small to have effect on the workers health.
- **Is an epidemiological case-control study those who drank tap water from Helsinki area were 1.01X more prone to get cancers than their control from the metropolitan area.**
 - The result is not statistically significant.
 - Is the result of practical significance? Yes. There are 500000 individuals living in Helsinki. As a quick estimate $1.01 \times 500000 - 500000$ would mean 5000 new cases of cancer per year.
 - Had the effect also been statistically significant, it would have strengthened it, but changed the conclusions.

Phases of testing

- **Select an appropriate statistical test**
 - Compare means of groups?
 - Compare variances of groups?
 - Compare the distributions
 - Model the relationship between two variables?
- **Calculate the test statistic and p-value**
 - These are automated by the computer
- **Draw conclusions**
 - This is not automated by the computer!

How to select the test?

➤ **There are two types of test**

- **Parameteric**
 - Assume that the variable is normally distributed
- **Non-parametric**
 - Does not assume that the variable is normally distributed
 - But, can make other restricting assumptions!

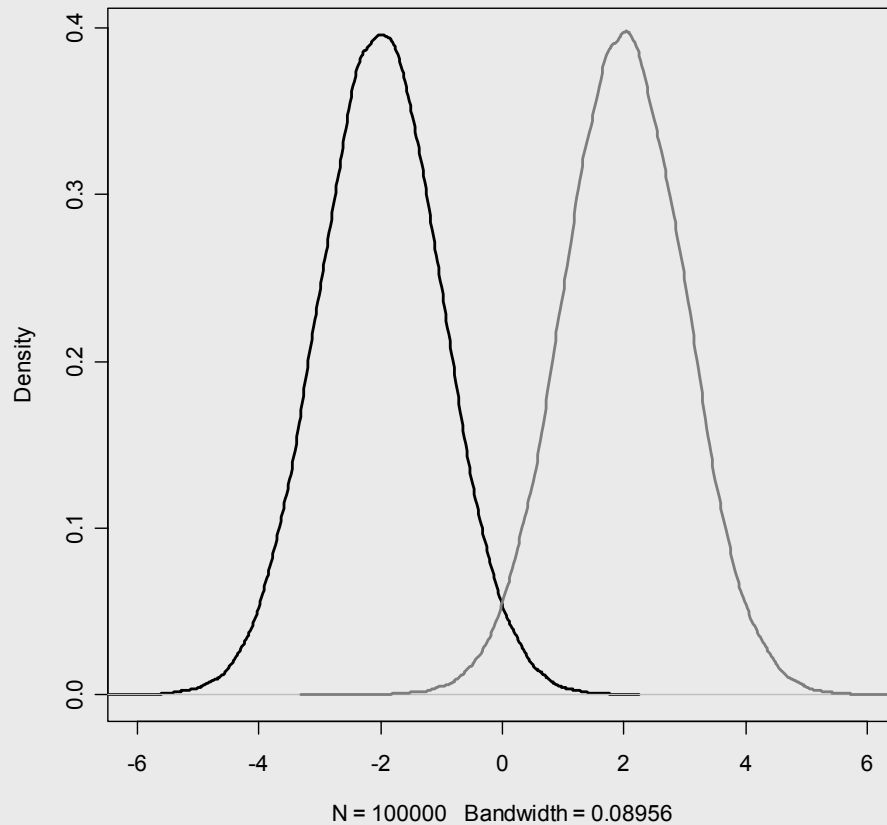
➤ **Only parametric ones are used on this course**

➤ **What kind of hypothesis you want to test?**

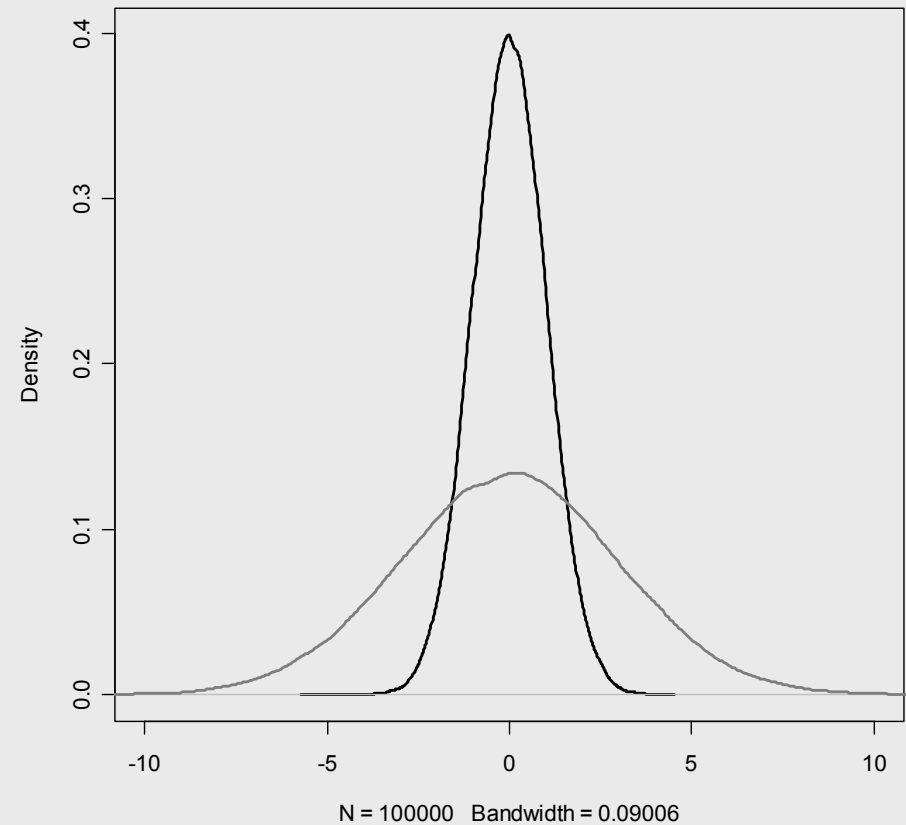
- Is the prime interest in the difference in means?
 - Are men taller than women
- Can the difference in variance be of interest?
 - Is the height of men more variable than the height of women?
- Do you want to predict the variable with another variable
 - Can a persons height be predicted from shoesize?

Mean and variance, an example

density.default(x = x1)



density.default(x = y1)



Different tests

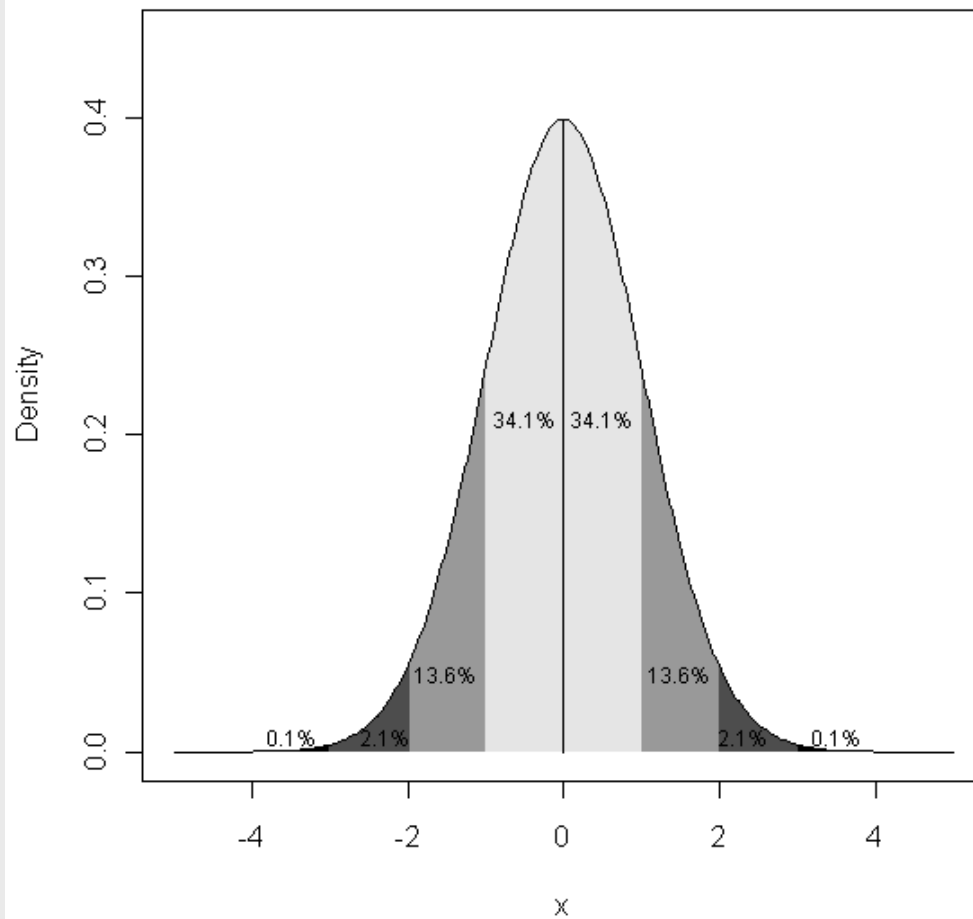
- **Compare the means of groups**
 - T-test
 - ANOVA
- **Compare the variability of groups**
 - F-test
- **Compare the distribution of categorical variables**
 - Chi Square
- **Predict a variable with another variable**
 - (Linear) Regression

Tests to compare group means

One-sample t-test I/XI

- **Comparison of the mean of the data againsts some known value of group mean.**
 - Is the mean of height of the sampled students different from the population mean (we know the population mean to be 167 cm)?
- **This simple test will act a primer to all other tests, since deep down they have the same principles:**
 - Calculate a test statistic (here, T)
 - Calculate the degrees of freedom
 - Compare the test statistic to a distribution (here, T)
 - Get the p-value

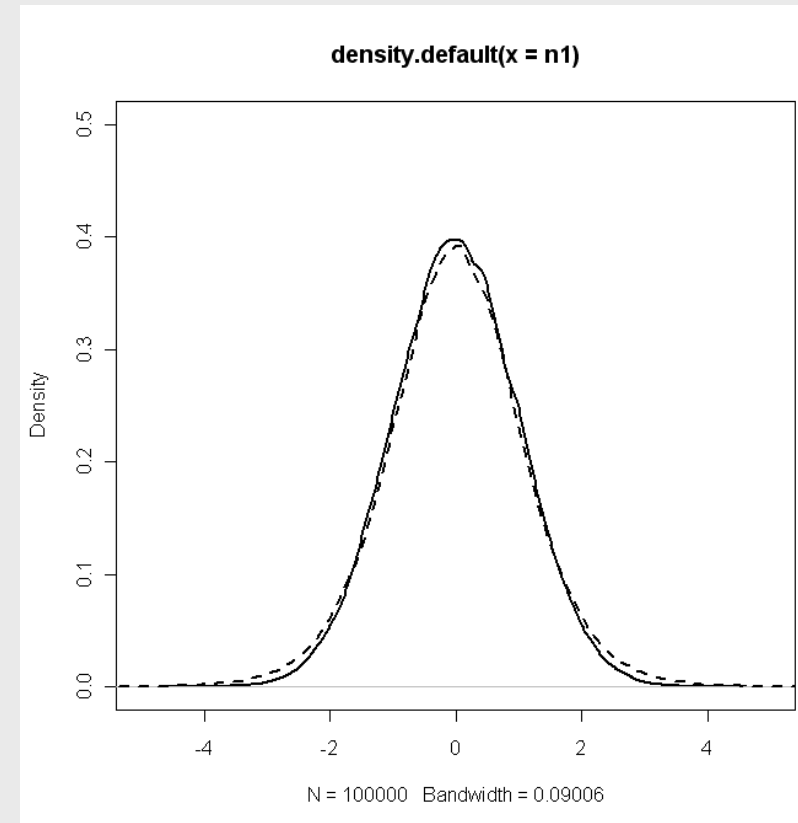
Normal distribution I/III



One-sample t-test II/XI

➤ The idea behind the t-test is the following

- Transform the variable of interest to follow a t-distribution.
 - T-distribution is very similar to a normal distribution, but with a small degrees of freedom it's tails are fatter.
 - Degrees of freedom is the parameter that defines the shape of the t-distribution.
- Compare the calculated t statistic to standard t distribution with the certain degrees of freedom.
- If the test statistic falls in the area where less than 5% of the values in the standard distribution are, the result is significant with p-value of 0.05.



One-sample t-test III/XI

➤ What are degrees of freedom?

- Assume we know three values (1,2,3) and the mean of the values (2).
- To calculate the degrees of freedom, we have to think how many of those values can we erase, and still be able to say what it was. Note that we have the mean to help as here.
- In this case, one can erase one of the values, and still be able to say what it was.
 - If we erase number 1, we have to values (2,3) left. Since we know the mean (2), we can say with confidence that the one value that was removed was 1.
 - The same goes for all other values as well.
- Since one value could be erased, we say that the degrees of freedom is 2 (equal to the number of observations left).

One-sample t-test IV/XI

➤ So, how do we get the test statistic then?

- Say we have five observations of height (160, 170, 172, 174, 181)
- The mean height of population is 167
- We first calculate a mean of the observations, that's 171.4
- Then we calculate the standard deviation, that's 7.6
- Last, we plug these into the formula:

$$T = \frac{M - \mu}{\frac{s}{\sqrt{n}}},$$

- Using the numbers we just calculated that becomes:
 - $T = (171.4 - 167) / (7.6 / 2.24) = 1.30$
- Last, the value of T is compared to a table of critical values, where we can see, that $T = 1.30$ with $df = 5 - 1 = 4$ is not statistically significant
 - We don't use a table here, but R (see the next slide)

One-sample t-test V/XI

➤ `> height<-c(160, 170, 172, 174, 180)`

➤ `> t.test(height, mu=167)`

➤ One Sample t-test

➤ data: s

➤ `t = 1.2941, df = 4, p-value = 0.2653`

➤ alternative hypothesis: true mean is not equal to 167

➤ 95 percent confidence interval:

➤ 161.9601 180.8399

➤ sample estimates:

➤ mean of x

➤ 171.4

One-sample t-test VI/XI

➤ What is that p-value anyway?

- P-value is a risk of saying that there is a difference between the groups means when there actually isn't.
- So, if there is a difference in heights, the p-value should be small, and there is not any difference, then it should be high.
- Traditionally p-values were coded with three stars:
 - 0.05 *
 - 0.01 **
 - 0.001 ***
- Nowadays it's more customary to report the p-value as such.

➤ How to interpret the p-value?

- If the p-value is less than 0.05 then the test usually said to be statistically significant.
 - This cut-off is made from the top of one's head, but it is often used, purely on traditional basis.

One-sample t-test VII/XI

- **What happens, if the difference remains at the same level, but we add more observations?**
- With 10 observations:
 - $t = 3.6565$, $df = 9$, $p\text{-value} = 0.005264$
 - With 20 observations:
 - $t = 2.474$, $df = 19$, $p\text{-value} = 0.02296$
 - With 100 observations:
 - $t = 6.6407$, $df = 99$, $p\text{-value} = 1.696e-09$

One-sample t-test VIII/XI

➤ Pay attention to the degrees of freedom!

```
➤      One Sample t-test

➤      data:  s
➤      t = 1.2941, df = 4, p-value = 0.2653
➤      alternative hypothesis: true mean is not equal to 167
➤      95 percent confidence interval:
➤      161.9601 180.8399
➤      sample estimates:
➤      mean of x
➤      171.4
```

➤ Here we had 5 observations, so the degrees of freedom should be 4, as they are.

- If they weren't, then something went wrong, and you should check your procedures.

One-sample t-test IX/XI

➤ What about the confidence interval?

```
➤      One Sample t-test

➤      data:  s
➤      t = 1.2941, df = 4, p-value = 0.2653
➤      alternative hypothesis: true mean is not equal to 167
➤      95 percent confidence interval.
➤      161.9601 180.8399
➤      sample estimates:
➤      mean of x
➤      171.4
```

➤ Confidence intervals gives a range of values. The true mean estimated from the sample is in this range with 95% probability.

- If you sample the same population again 100 times, the true mean should fall into this range about 95% of the time.

One-sample t-test X/XI

➤ How do you calculate a confidence interval?

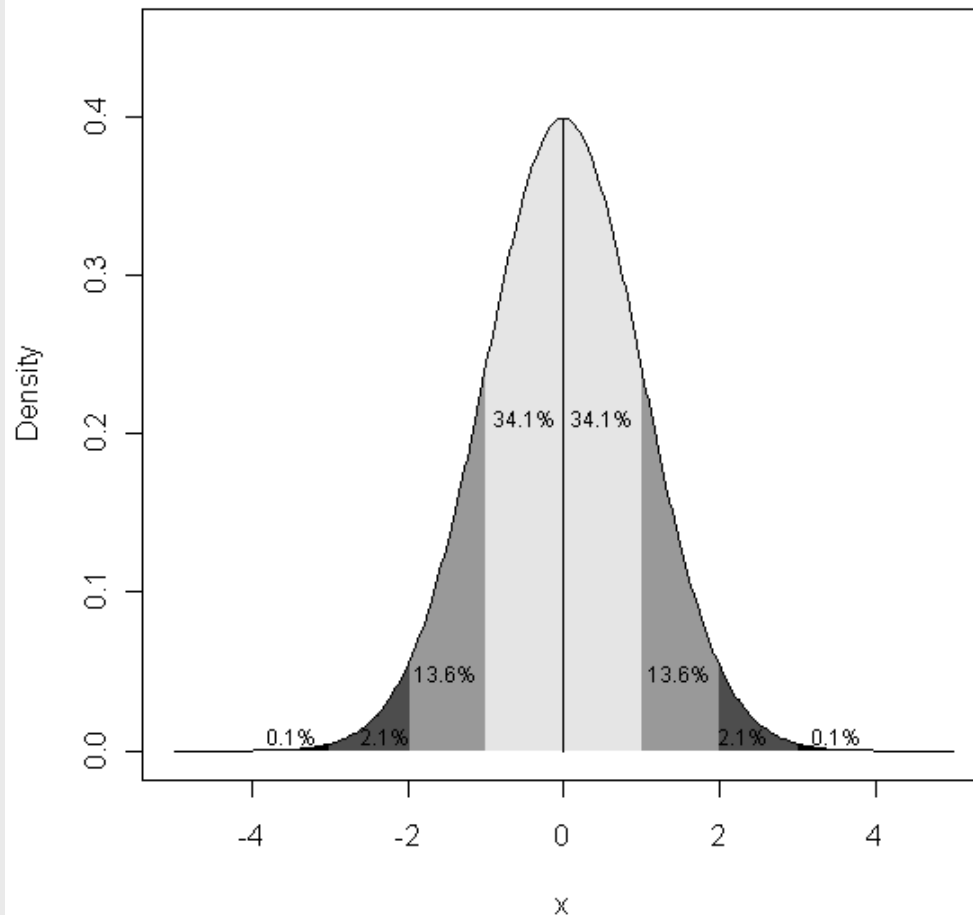
- We use normal distribution (or t-distribution) for calculations.
- Using the normal distribution, 95% of the values are in the range of ± 1.96 standard deviations from the mean.
- Since we want the estimate of the mean to be in this range, we use that 1.96 for calculations.
- First calculate a standard error (standard deviation of the estimated mean)
 - $SE = SD / \sqrt{n} = 7.6 / 3.4 = 2.235$
- The positive confidence interval is then
 - $mean + 1.96 * SE = 171.4 + 1.96 * 2.235 = 177.61$
- And the negative confidence interval is
 - $mean - 1.96 * SE = 171.4 - 1.96 * 2.235 = 167.01$

➤ These values are not equal to the ones given in the t-test output from R. The reason is that these were calculated in a slightly different way (using normal distribution instead of t distribution)

One-sample t-test XI/XI

- **Calculating the correct confidence interval by hand in R using the t distribution**
 - First check the correct quantile from the t-distribution
 - Two-tailed test, so should 0.975
 - `qt(0.975, df=4)` # 2.776445
 - The calculate the standard error
 - `sd(height) / sqrt(5)` # 3.4
 - Calculate the positive confidence interval
 - `171.4 + 2.776445 * 3.4` # 180.8399
 - Calculate the negative confidence interval
 - `171.4 - 2.776445 * 3.4` # 161.9601
- **Now these are the same values as output by `t.test()` in R.**
- **Note that the confidence intervals calculated on the basis of t-distribution are slightly wider than those based on the normal distribution.**
 - That's how it should be.

Normal or t distribution



➤ Two-tailed test:

- Both ends taken into account (5% of the values are in both ends)
- In the two-tailed test, the cut-off point for quantile from the distribution is 0.975

➤ One-tailed test:

- Only one end taken into account
- The quantile is 0.95.

Exercise IX

T-test and height

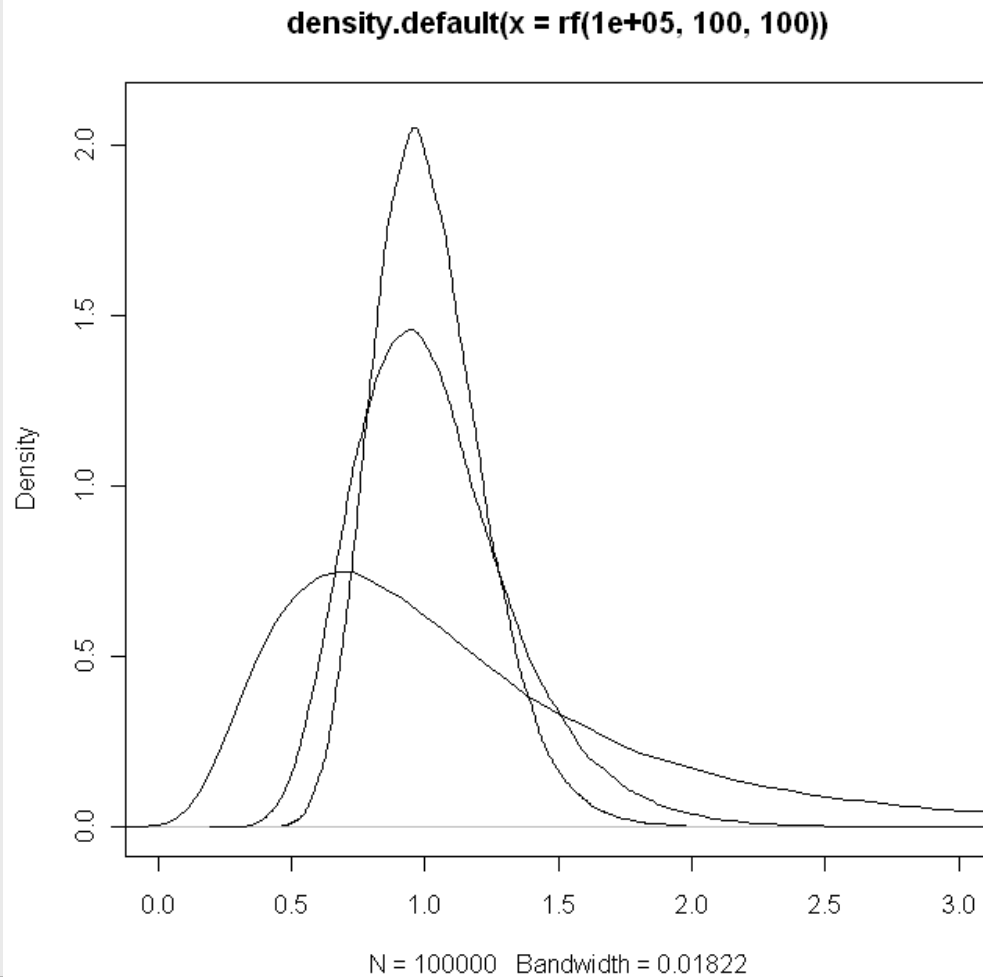
- **Compare the mean height of the students to the known mean height of Finnish population (167 cm) using the one-sample t-test.**
 - Is there a significant difference?
 - Are the bioinformatics student in average longer than the Finnish population
 - What might explain the situation?

Tests to compare group variances

F-test I/IV

- **F-test is used for comparing variances of two groups.**
 - More generally F-test is any test that uses F distribution.
- **Hypothesis are usually:**
 - $H_0: \text{Var1} = \text{Var2}$
 - $H_1: \text{Var1} > \text{Var2}$
- **Test statistic is the larger of**
 - $\text{Var1} / \text{Var2}$
 - $\text{Var2} / \text{Var1}$
- **The stronger the ratio deviates from 1, the stronger the evidence for unequal variances is.**
- **Degrees of freedom are calculated as for two-sample t-test.**

F-test II/IV



- **F-distribution is defined by two degrees of freedom.**

F-test III/IV

- `> x<-rnorm(10, mean=0, sd=1)`
- `> y<-rnorm(10, mean=3, sd=1)`

- `F test to compare two variances`

- `data: x and y`
- `F = 0.7891, num df = 9, denom df = 9, p-value = 0.73`
- `alternative hypothesis: true ratio of variances is not equal to 1`
- `95 percent confidence interval:`
- `0.1960066 3.1769977`
- `sample estimates:`
- `ratio of variances`
- `0.7891213`

F-test IV/IV

- **F-test can be used prior to t-test to check whether the variances of the groups are equal, and then to adjust the test accordingly.**
- **It is safe to use setting unequal variance in every situation, but the test is more powerful (finds statistically significant difference more often) if the correct setting is used.**

Exercise X

F-test

- **Compare variance of heights and shoesizes between**
 - Males and females
 - Kuopio and Helsinki

Tests to compare group means

Two-sample t-test

- **Two-sample t-test compares means of two groups.**
- **The idea is the same as in the one-sample t-test.**
 - First we calculate the difference in group means.
 - Then we divide it by the standard error.
 - There are different ways to estimate the standard error depending on whether the variances in the groups can be assumed to be equal or unequal.
 - Thus, we get the test statistic, and we conclude testing as with one-sample t-test.

Two-sample t-test in R

```
➤ > x<-rnorm(10, mean=0, sd=1)
➤ > y<-rnorm(10, mean=3, sd=1)
➤ > t.test(x, y)
```

```
➤ Welch Two Sample t-test
```

```
➤ data: x and y
➤ t = -10.7387, df = 17.753, p-value = 3.416e-09
➤ alternative hypothesis: true difference in means is not equal to 0
➤ 95 percent confidence interval:
➤ -4.217709 -2.836288
➤ sample estimates:
➤ mean of x mean of y
➤ -0.3181124 3.2088861
➤ > t.test(x, y, var.equal=T)
```

```
➤ Two Sample t-test
```

```
➤ data: x and y
➤ t = -10.7387, df = 18, p-value = 2.95e-09
➤ alternative hypothesis: true difference in means is not equal to 0
➤ 95 percent confidence interval:
➤ -4.217021 -2.836976
➤ sample estimates:
➤ mean of x mean of y
➤ -0.3181124 3.2088861
```


Note on degree of freedom

- **Note that in the two-sample test assuming equal variances, the degrees of freedom are calculated as a sum of**
 - Number of observation in group A -1
 - Number of observation in group B -1
- **So the df should always two less than the number of observations in the whole data set.**

Exercise XI

Two-sample t-test

- **Compare mean heights and shoesizes between**
 - Males and females
 - Kuopio and helsinki
- **When running the t-test, taken into account the results from the Exercise X (F-test), apply a suitable form of t-test.**

Paired t-test

- **Paired t-test is applied in situations where there is a paired setting.**
 - The samples were measured before and after some treatment.
- **The demodata Hygrometer contains paired data**
 - There are two observations per every hygrometer.
 - Each one of them was read before and after a longer rainy period.
 - Note that after preprocessing done on the first day, the data are now in two different columns is R. The order of the hygrometers is exactly the same in both columns, otherwise the pairing would be meaningless.
- **Paired t-test equal running a one-sample t-test on the differences between the two observations.**
 - Subtract the observation for hygrometer 1 on day 1 from the observation for hygrometer 1 on day 2.
 - Do this for all hygrometer, and run a one-sample t-test on these differences.

Paired t-test in R

- `> x<-rnorm(10, mean=10, sd=1)`
- `> y<-x+rnorm(10, mean=0, sd=1)`
- `> t.test(x, y, paired=T)`
- Paired t-test
- data: x and y
- `t = 0.5283, df = 9, p-value = 0.6101`
- alternative hypothesis: true difference in means is not equal to 0
- 95 percent confidence interval:
- `-0.5609109 0.9026993`
- sample estimates:
- mean of the differences
- `0.1708942`

Running the paired t-test by hand

➤ `> dif<-x-y`

➤ `> t.test(dif, mu=0)`

➤ One Sample t-test

➤ data: dif

➤ `t = 0.5283, df = 9, p-value = 0.6101`

➤ alternative hypothesis: true mean is not equal to 0

➤ 95 percent confidence interval:

➤ `-0.5609109 0.9026993`

➤ sample estimates:

➤ mean of x

➤ `0.1708942`

Exercise XII

Paired t-test

- **Use Hygrometer dataset for this exercise.**
- **Is there a difference in mean humidity between before the rain and after the rain measurements?**

Analysis of variance

ANOVA I/

- **ANOVA compares the means of three or more groups.**
- **It tells us whether there is a statistically significant difference between any of the groups, but it does not tell the groups that are different.**
 - After running ANOVA, there are ways to find the groups that differ. Those are called post-hoc tests.
- **ANOVA can be thought of as a generalization of a two-sample t-test.**
- **Only one-way ANOVA will be presented here.**
 - In one-way ANOVA, there is one dependent variable (e.g. height) and a categorical variable (e.g. population) giving grouping of observations of the dependent variables.

ANOVA II/

- **The variance in the dependent variable can be partitioned into two parts:**
 - Variance within groups
 - Individual differences
 - Measurement error
 - Variance between groups
 - Effect of the grouping variable
 - Individual differences
 - Measurement error
- **The actual test is based on comparing the magnitudes of these variances using the F-test.**
 - If the between groups variance is large enough compared to the variance within groups ("error variance"), the test will come up as significant.

ANOVA III/

➤ Calculations

- Variance within groups
 - Calculate an individual estimate of variance inside every group using group specific means.
 - Variance in every group has $n-1$ degrees of freedom.
 - Thus, in total this variance estimate has $n-k$ (k =number of groups) degrees of freedom.
- Variance between groups
 - This means the variance between group-wise means and the grand mean of the whole dataset (weighted using the group sizes).
 - The degrees of freedom are $k-1$.

➤ **These two variances are two different estimates of population variance.**

Calculation of ANOVA by hand

	Bacteria 1	Bacteria 2	Bacteria 3
Observation 1	2	6	4
Observation 2	3	7	5
Observation 3	1	5	3
Mean	2	6	4
Overall mean		4	

Calculation of ANOVA by hand

Table 8.3: *Calculation of error terms.*

	Bacteria 1	Bacteria 2	Bacteria 3
Observation 1	$2-2=0$	$6-6=0$	$4-4=0$
Observation 2	$2-3=-1$	$6-7=-1$	$4-5=-1$
Observation 3	$2-1=1$	$6-5=1$	$4-3=1$
Sum of squares	$0+1+1$	$0+1+1$	$0+1+1$
SS within groups	$2+2+2=6$		

Table 8.4: *Calculation of treatment effects.*

	Bacteria 1	Bacteria 2	Bacteria 3
Observation 1	$4-2=2$	$4-6=-2$	$4-4=0$
Observation 2	$4-3=1$	$4-7=-3$	$4-5=-1$
Observation 3	$4-1=3$	$4-5=-1$	$4-3=1$
Sum of squares	$4+1+9=14$	$4+9+1=14$	$0+1+1=2$
SS total	$14+14+2=30$		

Calculation of ANOVA by hand

Table 8.5: *ANOVA table presents a summary of the results.*

	SS	df	MS	F	p-value
Effect	24	2	12	12	<0.01
Error	6	6	1		

ANOVA IV/

➤ ANOVA in R

- `> x1<-rnorm(10, mean=0, sd=1)`
- `> x2<-rnorm(10, mean=0, sd=1.5)`
- `> x3<-rnorm(10, mean=2, sd=1)`
- `> x<-c(x1,x2,x3)`
- `> group<-c(rep(1, 10), rep(2, 10), rep(3, 10))`
- `> group<-as.factor(group)`
- `> a1<-aov(x~group)`
- `> a1`
- Call:
- Terms:
 - group Residuals
- Sum of Squares 15.88236 44.95124
- Deg. of Freedom 2 27
- Residual standard error: 1.290295
- Estimated effects may be unbalanced

ANOVA VI/

➤ ANOVA in R

```

■ > summary(a1)
■           Df Sum Sq Mean Sq F value    Pr(>F)
■ group         2  15.882     7.941   4.7699 0.01683 *
■ Residuals    27  44.951     1.665
■ ---
■ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

➤ What are those Sum of Squares?

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2 = \frac{1}{n-1} \sum_{i=1}^n y_i^2 - \frac{n}{n-1} \bar{y}^2,$$

➤ **Where do the degrees of freedom (Df) come from?**

- Check the slide ANOVA III
- Those are the numbers used as denominator in the variance formula

ANOVA VI/

➤ ANOVA in R

```
▪ > summary(a1)
▪               Df Sum Sq Mean Sq F value    Pr(>F)
▪ group          2 15.882     7.941   4.7699 0.01683 *
▪ Residuals     27 44.951     1.665
▪ ---
▪ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

➤ What about Mean Sq?

- That's the estimate of variance
 - Group = variance between groups
 - Residuals = error variance (variance within groups)

➤ F Value?

- That's the ratio between the two variance estimates = F test statistic

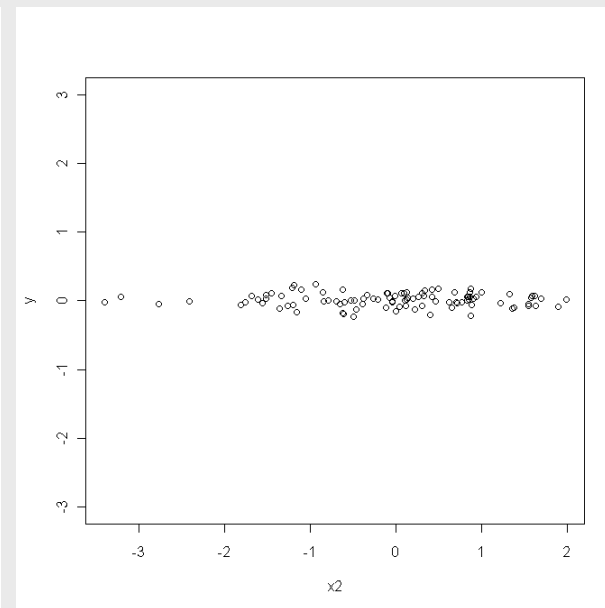
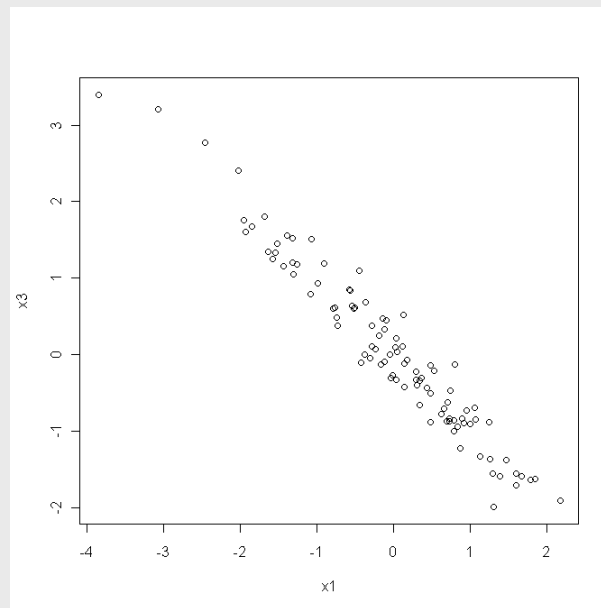
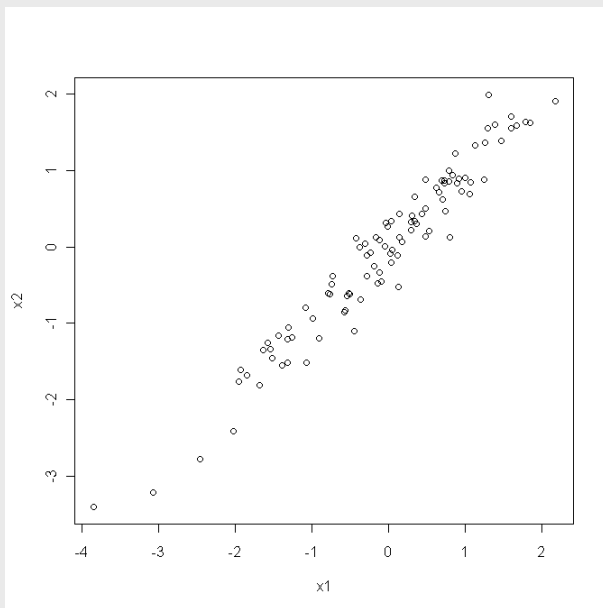
Exercise XIII

ANOVA

- **Test is there any difference between the mean height or shoeseize between different student populations.**
 - Are there any significant differences?
 - How does the boxplot look like for the same data?
- **Dataset Customer lists the number of customer questions to helpdesk during a period of four months.**
 - Assuming the variable questions is normally distributed, is there any difference in the mean number of questions on different days of week or different months?
 - Does a boxplot support this result?
- **Clover dataset contains leaf area measurements with different nitrogen and sulfur treatments.**
 - Do these treatments (analyze independently) affect the leaf area?

Linear regression

Correlation I/VI



Correlation II/VI

- **Correlation coefficient varies between -1 (perfect negative relationship) and +1 (perfect positive relationship).**
- **$r = s_{xy} / s_x * s_y$**
- **where s_{xy} = covariance**
 - For every value of X, subtract from it the mean of all X values. Do the same for every Y value. Multiple there results so that each centered X value is multiplied by the concomittant centered Y value. Sum over the multiplication results. Divide the sum by the number of observation subtracted by one.
- **Correlation is usable only for data that are linearly dependent (check the plots). Correlation can be calculated for non-linear datasets, but it has no meaning.**
- **Correlation can't be used the other way around. If the correlation is high, it does not necessarily mean that the variables are linearly dependent.**

Correlation III/VI

➤ Testing the correlation coefficient

- T statistic is calculated as the square root of (number of observations - 2) / square root of (1 – squared correlation coefficient) multiplied by the correlation coefficient. This is compared to the t-distribution with n-2 degrees of freedom.

➤ Calculations in R by hand

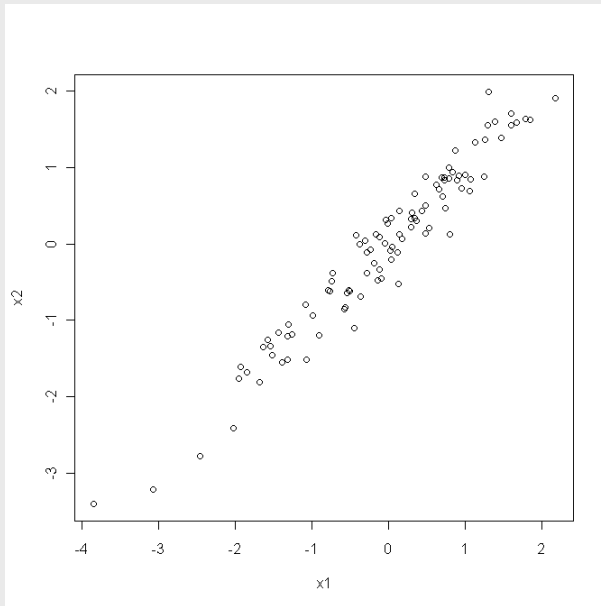
- `> x1<-rnorm(100)`
- `> x2<-x1+rnorm(100, mean=0, sd=0.25)`
- `> y<-rep(0, 100)`
- `> y<-y+rnorm(100, mean=0, sd=0.25)`
- `> cor(x1, x2)`
- `[1] 0.9719912`
- `> 0.9719912 * (sqrt(98)/sqrt(1-0.9719912^2))`
- `[1] 40.94262`

Correlation IV/VI

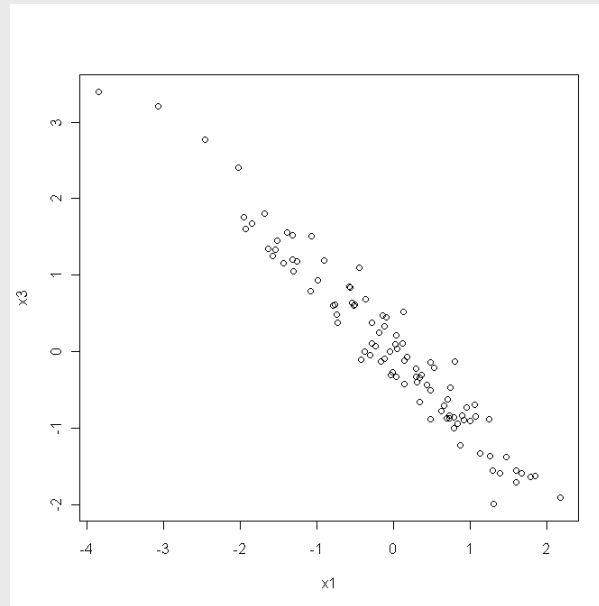
➤ Testing in R automatically

- `> cor.test(x1, x2)`
- Pearson's product-moment correlation
- data: x1 and x2
- `t = 40.9426, df = 98, p-value < 2.2e-16`
- alternative hypothesis: true correlation is not equal to 0
- 95 percent confidence interval:
- `0.9585825 0.9811008`
- sample estimates:
- `cor`
- `0.9719912`

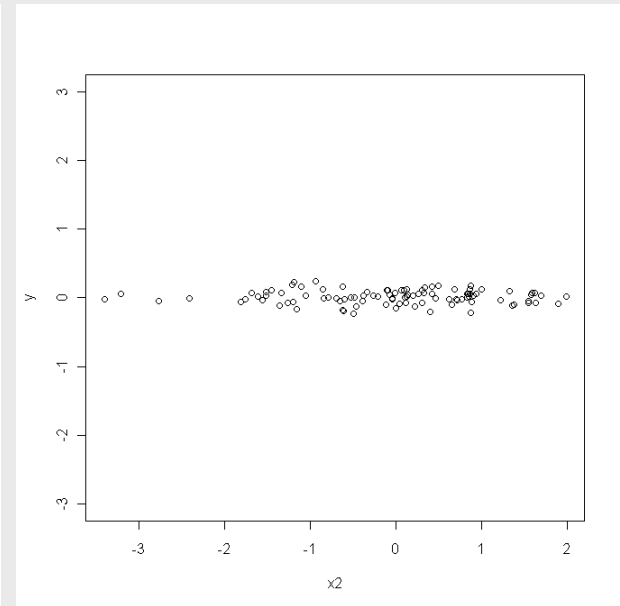
Correlation V/VI



$r = 0.9719912$
p-value < 2.2e-16



$r = -0.9719912$
p-value < 2.2e-16



-0.01394637
p-value = 0.8905

Correlation VI/VI

➤ Caveats of testing the correlation coefficients

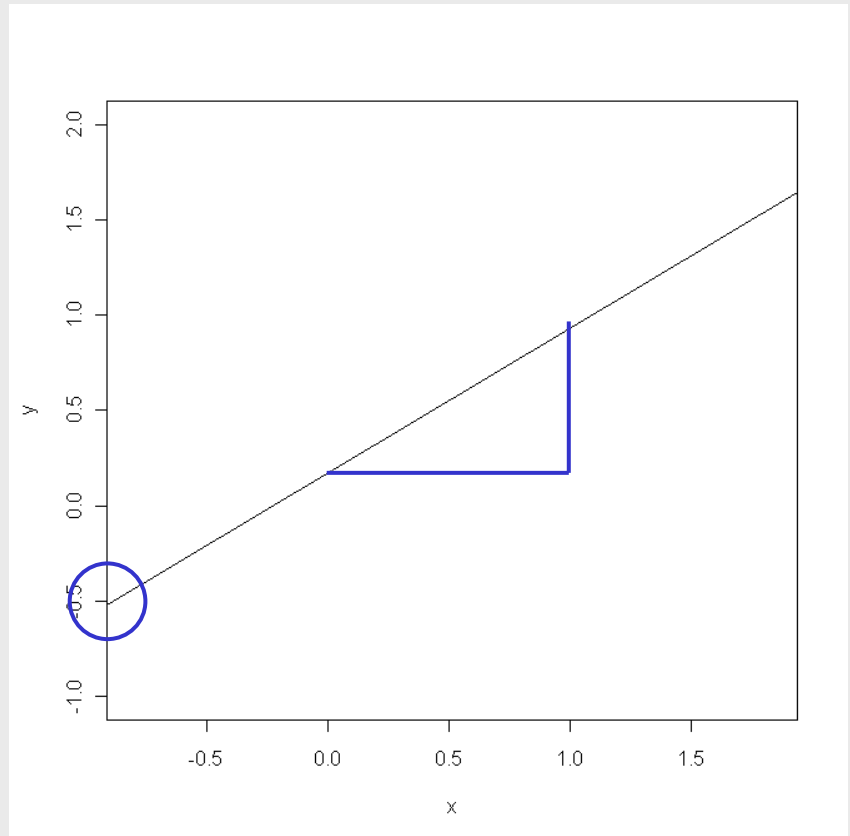
- If there are enough observations, say 1000, for the compared variables, even very small coefficients ($r = 0.1$ or $r = 0.01$) might come as significant.
- Such small coefficients, even if statistically significant, don't typically imply that the relationship between the variables would be strong.
- This is equivalent to the already discussed situation of statistical significance versus practical significance.
- Correlation coefficients can't directly be thought to represent causal relationships between the variables.
 - The correlation coefficient is exactly the same, even if the order of the variables in the test is reversed.

Linear regression I/

- **Correlation quantifies the strength of association between two linearly dependent variables.**
 - Using correlation, it is impossible to predict which is the value for the second variables, if we know the value of the first variable.
- **Linear regression tries to build a predictive model that can**
 - be used for predicting the second variable from the first variable
 - describe the relationship between the variable in a more formal fashion
- **In linear regression the first variable is called the predictor (or independent variable) and the second is called the predicted (or dependent variable)**
 - So, there is already a postulated division into predicted and predicting variables – this was not the case with correlation

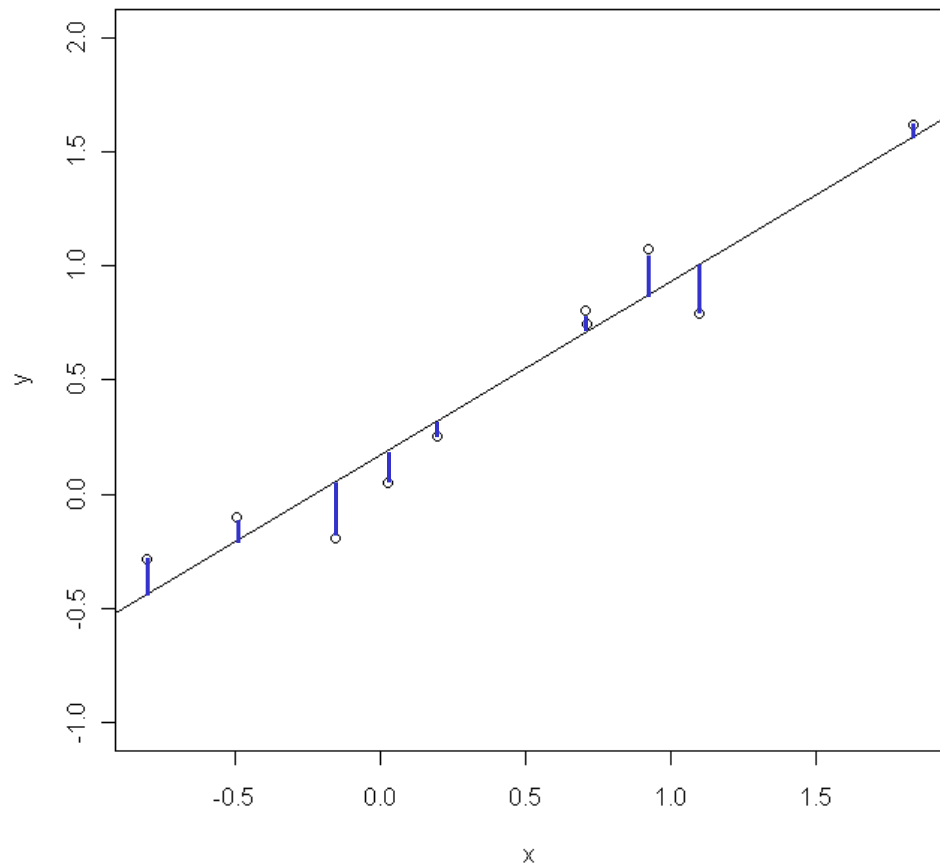
Linear regression II/

- **Linear regression uses a formula for a simple line fitted into the dataset.**
- **Line can be expressed mathematically as**
 - $y = a + bx$
- **Often in statistics this is written as**
 - $y = b_0 + b_1X$
- **In order to fit the line, we need to estimate a and b from our data.**
 - This is done using the least squares approach.



Linear regression III/

- We fit the line so that the sum of squared distances between the line and the observations is as small as possible.
- Sum of squares... sounds a bit like ANOVA... and it is!
 - The error variance in ANOVA is the same as the summed squared distance between the line and the observations.



Linear regression IV/

➤ Calculation in R

```
> y<-rnorm(10, sd=1, mean=0)
> x<-y+rnorm(10, sd=0.25, mean=0)
> lm(y~x)
```

Call:

```
lm(formula = y ~ x)
```

Coefficients:

(Intercept)	x
0.1681	0.7590

Linear regression V/

➤ Calculation in R

```
> y<-rnorm(10, sd=1, mean=0)
> x<-y+rnorm(10, sd=0.25, mean=0)
> summary(lm(y~x))
```

Call:

```
lm(formula = y ~ x)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.2433	-0.1201	0.0488	0.1018	0.2004

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.16807	0.05907	2.845	0.0216 *
x	0.75899	0.06874	11.041	4.03e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1647 on 8 degrees of freedom

Multiple R-squared: 0.9384, Adjusted R-squared: 0.9307

F-statistic: 121.9 on 1 and 8 DF, p-value: 4.033e-06

P-values for predictors

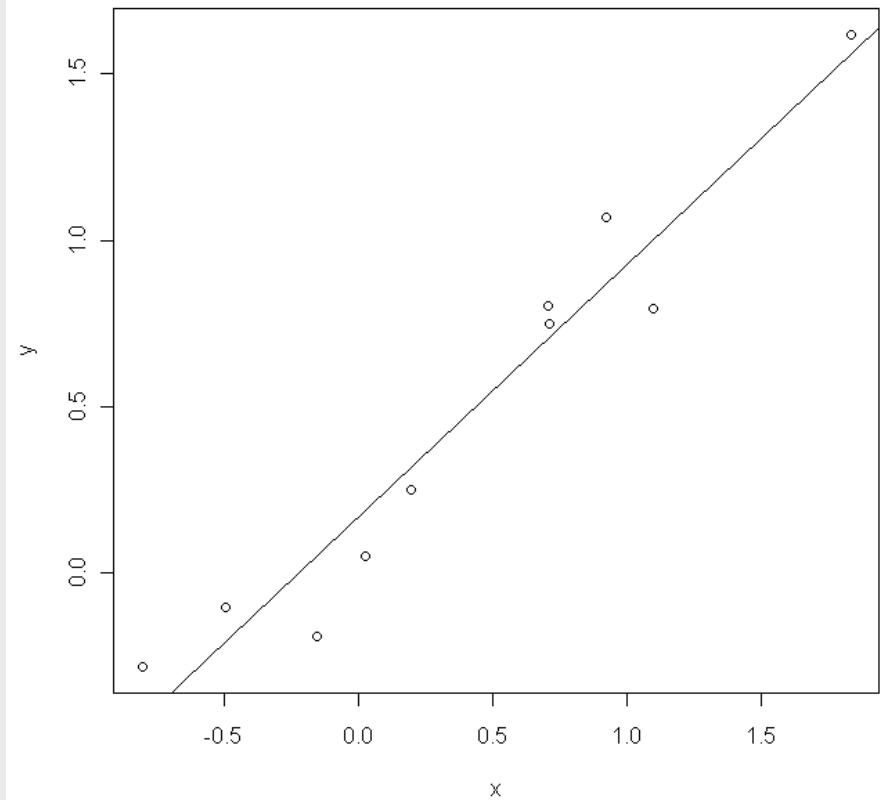
$\text{cor}(x,y) \cdot \text{cor}(x,y)$

P-value for the model

Linear regression VI/

➤ Plotting the results

- `> plot(x, y)`
- `> abline(lm(y~x))`



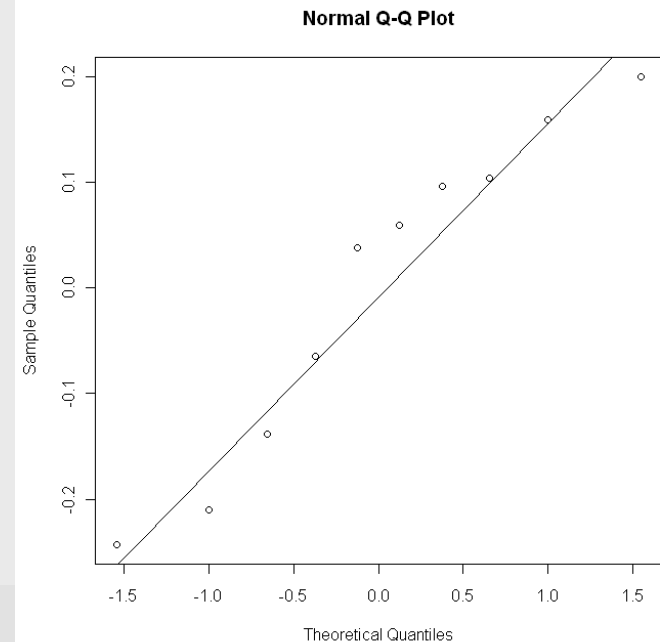
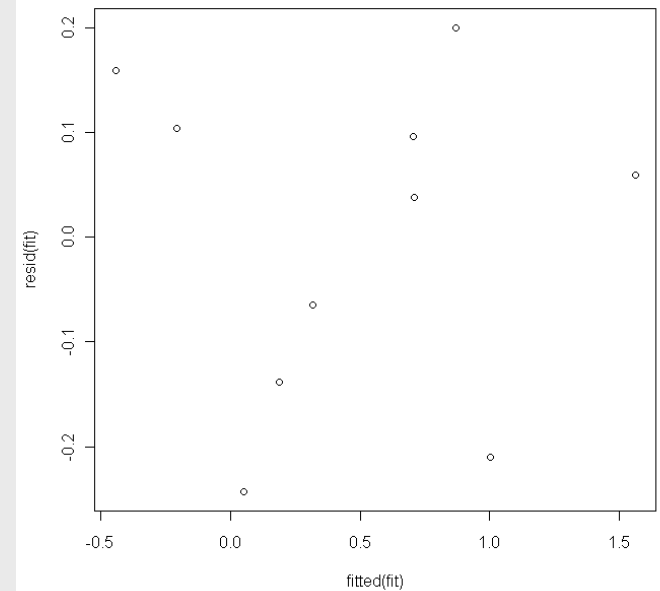
Linear regression VII/

➤ Diagnostic plots

- Does the model fit the data?

➤ In R

- `> fit<-lm(y~x)`
- `> plot(fitted(fit), resid(fit))`
- `> qqnorm(resid(fit))`
- `> qqline(resid(fit))`



Linear regression VIII/

➤ Linear regression with a categorical variable

```
> group<-factor(c(rep(1, 10), rep(2, 10), rep(3, 10)))  
> y1<- c(rnorm(10, mean=0, sd=1), rnorm(10, mean=2, sd=1), rnorm(10, mean=2, sd=2))  
> summary(lm(y1~group))
```

Call:

```
lm(formula = y1 ~ group)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.0970	-1.2270	0.2277	1.2265	2.4167

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.1964	0.5425	0.362	0.72016
group2	1.6629	0.7672	2.168	0.03918 *
group3	2.5437	0.7672	3.316	0.00261 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.715 on 27 degrees of freedom

Multiple R-squared: 0.2958, Adjusted R-squared: 0.2436

F-statistic: 5.67 on 2 and 27 DF, p-value: 0.008791

Linear regression IX/

```
> group<-factor(c(rep(1, 10), rep(2, 10), rep(3, 10)), labels=c("child", "adult",  
  "senior"))  
> y2<- c(rnorm(10, mean=0, sd=1), rnorm(10, mean=2, sd=1), rnorm(10, mean=2, sd=2))  
> summary(lm(y2~group))
```

Call:

lm(formula = y2 ~ group)

Residuals:

Min	1Q	Median	3Q	Max
-2.9529	-0.5609	0.1827	0.9495	1.8933

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.4036	0.4039	0.999	0.3266
groupadult	1.4471	0.5713	2.533	0.0174 *
groupsenior	1.0295	0.5713	1.802	0.0827 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Where has the groupchild disappeared?

Residual standard error: 1.277 on 27 degrees of freedom

Multiple R-squared: 0.2012, Adjusted R-squared: 0.142

F-statistic: 3.4 on 2 and 27 DF, p-value: 0.04820

Exercise XIV

Linear regression

- **Using the bioinformatics students dataset model the dependence of shoesize on height.**
 - What is correlation between height and shoesize. Is it statistically significant? Is it also practically significant?
 - How much of the variation in shoesize does height explain?
 - Does the model fit the data well?
 - Is there collinearity in the residuals?
 - Are the residuals normally distributed?

Comparing categorical variables

Chi square test I/VIII

➤ **There are two flavors of Chi Square tests**

- Goodness of fit test
 - In general: are observed frequencies as they are expected on the basis of some theory?
 - Comparing whether the frequency of heads and tails acquired with a coin is as expected (half and half)?
 - Is the observed distribution of the three possible genotypes of a gene as expected ($p^2 + 2pq + q^2$)?
- Test of independence
 - In general: is the distribution to the groups random?
 - Is the observed distribution of the genotypes of one gene equal in cancer cases and their healthy controls?

Chi square test II/VIII

➤ Goodness of fit

- Calling heads and tail 100 times on the same nickel, the following result was obtained:

head	tail
46	54

- If the coin is fair (not biases towards either result), the expected frequency of both heads and tails is 50%, i.e. 50 heads and 50 tails in this case.
- The Chi Square test statistic is calculated as the observed frequency minus expected frequency squared divided by the expected frequency. This is calculated for all classes, and summed together.
- Here: $(46-50)^2/50 + (54-50)^2/50 = 16/50 + 16/50 = 32/50 = 0.64$.
- This test statistic is compared to Chi Square distribution. This distribution is defined by its degrees of freedom. For this test the degrees of freedom are the number of classes (here two) minus 1, i.e. $2-1 = 1$.

Chi square test III/VIII

➤ Goodness of fit in R

- Calling heads and tail 100 times on the same nickel, the following result was obtained:

```
head tail
```

```
46    54
```

- Defining this in R can be done in two different ways. Either using the original variable:

```
x<-round(runif(100, min=0, max=1))
xx<-factor(x, labels=c("head", "tail"))
chisq.test(table(xx))
```

```
Chi-squared test for given probabilities
```

```
data:  table(xx)
```

```
X-squared = 0.64, df = 1, p-value = 0.4237
```

- Or typing in the table:

- `table1<-as.table(c(46,54))`
- `names(table1)<-c("heads", "tails")`
- `chisq.test(table1)`

Chi square test IV/VIII

➤ **Goodness of fit in R**

- By default R expects that we want to run a goodness of fit test against a uniform distribution.
 - Every class is equally probable = they have the same expected frequency.
 - Therefore, we do not need to specify the expected values.

Chi square test V/VIII

➤ Test of independence

- Calling heads and tail 100 times on two nickels, the following result was obtained:

- `> x<-round(runif(200, min=0, max=1))`
- `> c1<-x[1:100]`
- `> c2<-x[101:200]`
- `> c11<-factor(c1, labels=c("head", "tail"))`
- `> c22<-factor(c2, labels=c("head", "tail"))`

- **Coin 1**

head tail

50 50

- **Coin 2**

head tail

53 47

- Is the distribution of heads and tails for these two coins the same?

Chi square test VI/VIII

➤ Observed:

➤ Coin	Heads	Tails	Sum
➤ 1	50	50	100
➤ 2	53	47	100
➤ Sum	103	97	200

➤ Expected:

➤ Coin	Heads	Tails
➤ 1	$100 \cdot 103 / 200$	$100 \cdot 97 / 200$
➤ 2	$103 \cdot 100 / 200$	$97 \cdot 100 / 200$

➤ Expected

➤ Coin	Heads	Tails
➤ 1	51.5	48.5
➤ 2	51.5	48.5

Chi square test VII/VIII

➤ Test of independence

- The test statistic is calculated as for the goodness of fit test, but the degrees of freedom are calculated differently.
 - $Df = (\text{number of columns} - 1) * (\text{number of rows} - 1)$
 - Here $Df = (2-1)*(2-1) = 1$

➤ Test of independence in R

- If there are two vector of equal length, then

```
> chisq.test(c11, c22)
```

Pearson's Chi-squared test with Yates' continuity correction

data: c11 and c22

X-squared = 1.4452, df = 1, p-value = 0.2293

Chi square test VIII/VIII

➤ Test of independence in R

- If the vector are not of equal lenght, then we need to provide the command with a table:

```
> table(c11, c22)
```

```
      c22  
c11    head tail  
  head    30   20  
  tail    23   27
```

```
> chisq.test(table(c11, c22))
```

Pearson's Chi-squared test with Yates' continuity correction

```
data:  table(c11, c22)
```

```
X-squared = 1.4452, df = 1, p-value = 0.2293
```

Exercise XV

Compare several dice

- **The Dice dataset contains 120 rolls for four different dice.**
- **Use Chi square test for goodness of fit to see whether some of these dice are biased.**
 - These dice (red and blue) have been extensively tested using the board game Risk, and would appear to be biased towards higher numbers. At least when attacked using these dice, the lecturer has consistently always lost the combat even with favourable odds.