R for Biologists

Version 1.1 – December 2009

NIMBioS
National Institute for Mathematical
and Biological Synthesis

Marco Martinez

Supported by the University of Tennessee, Knoxville and the National Science Foundation through Cooperative Agreement EF-0832858 to the University of Tennessee.

Preface

This material is intended as an introductory guide to data analysis with the R system, to assist in statistical computing training for life science researchers. It was produced as companion material for a seminar (R Tutorial for Life Sciences) given at The University of Tennessee in the spring of 2009, sponsored by The National Institute for Mathematical and Biological Synthesis (NIMBioS), for graduate students in different biological areas. This material was produced to serve as a basic introduction to R for researchers and students visiting NIMBioS. Many more advanced guides are available both through the R web site and various books.

The principal aim is to provide a step-by-step guide on the use of R to carry out statistical analysis and techniques widely used in the life sciences. In each section, we give a detailed explanation of a command in R, followed by a biological example with all the instructions (in red) needed to run the test and with the corresponding output in R (in blue). In several sections we left some questions or additional analysis as an exercise. Also at the end of some sections we give a list of other commands in R related to the topics explained in the corresponding section. We assume some previous knowledge in statistics and experimental design, essentially corresponding to a basic undergraduate introductory statistics course.

These notes were written to take advantage of R version 2.8.1 or later, under a Windows operating system. This is version 1.1 of these notes, generated in May 2009 and edited for style and content by NIMBioS Director Louis Gross in December 2009. This document is available for download from the NIMBioS.org site and is provided free-of-charge with no warrantee for its use. It is not to be modified from this form without explicit authorization from the author.

Marco Martinez
Graduate Student - Mathematical Ecology
Department of Mathematics
The University of Tennessee - Knoxville
mmarti52@utk.edu
December 2009

Contents

1. An introduction to R

1.1 What is R?

R is a statistical computer program made available through the Internet under the General Public License (GPL). That is, it is supplied with a license that allows you to use it freely, distribute it, or even sell it, as long as the receiver has the same rights and the source code is freely available. It is available for Microsoft Windows XP or later, for a variety of Unix and Linux platforms, and for Apple Macintosh OS X (Dalgaard, 2002).

R is an integrated suite of software facilities for data manipulation, calculation and graphical display (Venables *et al.* 2009). There is a difference in philosophy between R and some other statistical software, since in R a statistical analysis is normally done as a series of steps, with intermediate results being stored as objects. Thus whereas SAS and SPSS will give copious output from a regression, R will give minimal output and store the results in an object (a statistical "fit") for subsequent interrogation by further R functions (Venables *et al.* 2009).

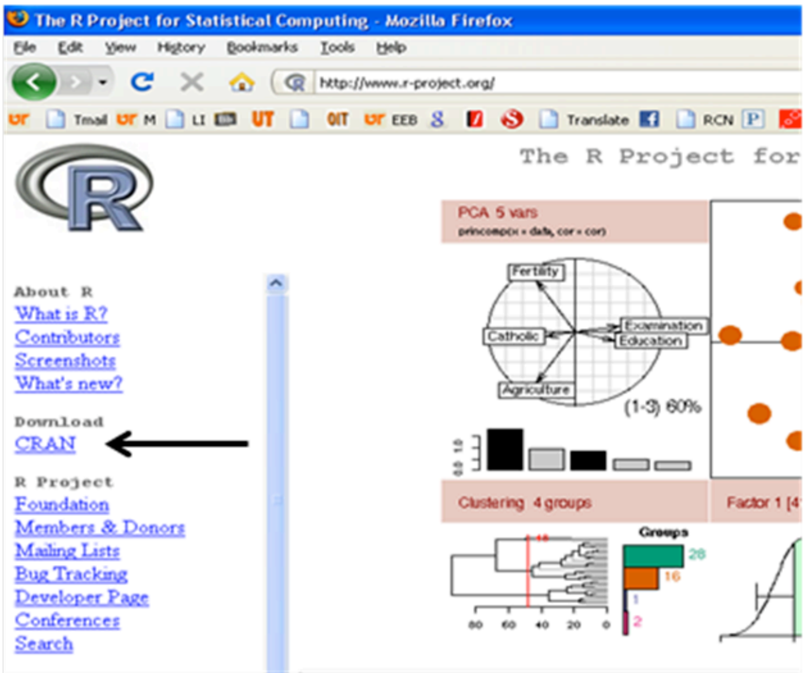
1.2 How to install R?

These instructions are given for R version 2.8.1

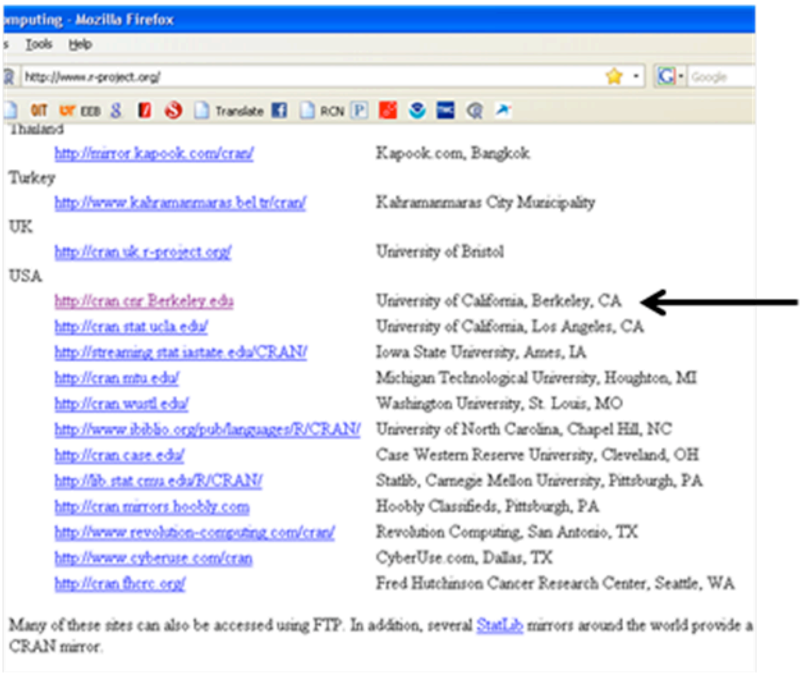
1.2.1 The base system

Here we give detailed instructions to download R. Please be aware that new versions can be released with some differences – the main R web page has details on new versions.

1. Go to the web page of R: http://www.r-project.org/
2. In the left part, find CRAN and click there.



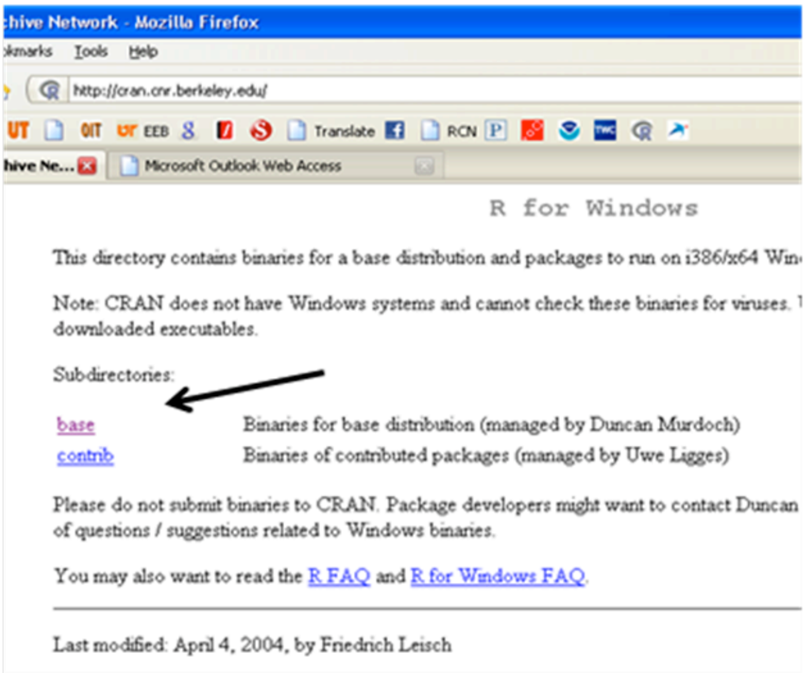
3. Now you can choose a mirror site to download the program. You can choose any mirror - here we illustrate using the first one in the USA, University of California, Berkeley.



Thailand		
	http://mirror.kapook.com/cran/	Kapook.com, Bangkok
Turkey		
	http://www.kahramanmaras.bel.tr/cran/	Kahramanmaras City Municipality
UK		
	http://cran.uk.r-project.org/	University of Bristol
USA		
	http://cran.cnr.Berkeley.edu	University of California, Berkeley, CA
	http://cran.stat.ucla.edu/	University of California, Los Angeles, CA
	http://streaming.stat.iastate.edu/CRAN/	Iowa State University, Ames, IA
	http://cran.mtu.edu/	Michigan Technological University, Houghton, MI
	http://cran.wustl.edu/	Washington University, St. Louis, MO
	http://www.ibiblio.org/pub/languages/R/CRAN/	University of North Carolina, Chapel Hill, NC
	http://cran.case.edu/	Case Western Reserve University, Cleveland, OH
	http://lib.stat.cmu.edu/R/CRAN/	Statlib, Carnegie Mellon University, Pittsburgh, PA
	http://cran.mirrors.hoobly.com	Hoobly Classifieds, Pittsburgh, PA
	http://www.revolution-computing.com/cran/	Revolution Computing, San Antonio, TX
	http://www.cyberuse.com/cran	CyberUse.com, Dallas, TX
	http://cran.fhcrc.org/	Fred Hutchinson Cancer Research Center, Seattle, WA

Many of these sites can also be accessed using FTP. In addition, several StatLib mirrors around the world provide a CRAN mirror.

4. Then we choose our operating system (e.g. Windows in our example). Then click on "base".


hive Network - Mozilla Firefox
okmarks Tools Help
http://cran.cnr.berkeley.edu/
UT OIT EEB Translate RCN P
hive Ne...
Microsoft Outlook Web Access

R for Windows

This directory contains binaries for a base distribution and packages to run on i386/x64 Win

Note: CRAN does not have Windows systems and cannot check these binaries for viruses. downloaded executables.

Subdirectories:

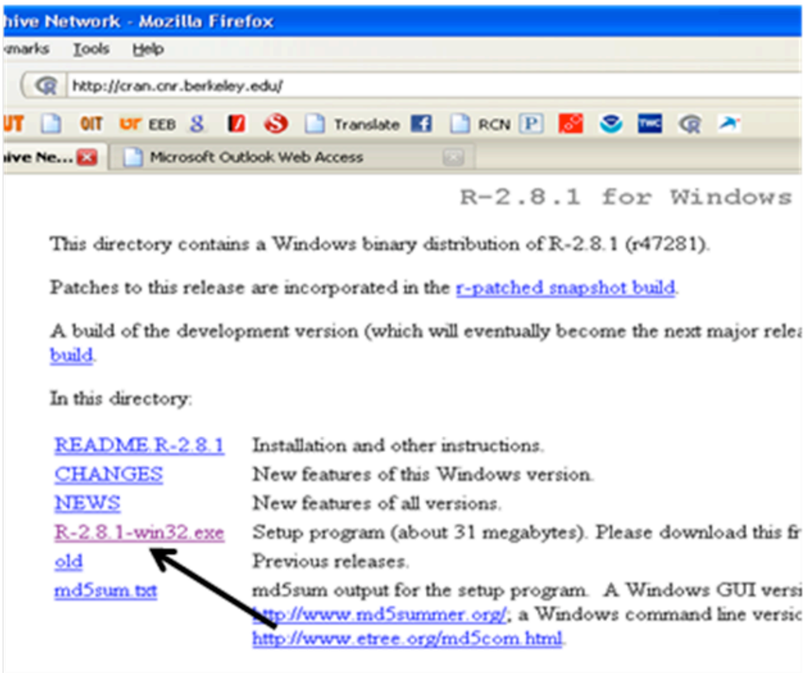
base	Binaries for base distribution (managed by Duncan Murdoch)
contrib	Binaries of contributed packages (managed by Uwe Ligges)

Please do not submit binaries to CRAN. Package developers might want to contact Duncan of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Last modified: April 4, 2004, by Friedrich Leisch

5. Finally we download the file R-2.8.1-win32.exe which executes R.


hive Network - Mozilla Firefox
marks Tools Help
http://cran.cnr.berkeley.edu/
UT OIT UT EEB Translate RCN
ive Ne...
Microsoft Outlook Web Access

R-2.8.1 for Windows

This directory contains a Windows binary distribution of R-2.8.1 (r47281).

Patches to this release are incorporated in the r-patched snapshot build.

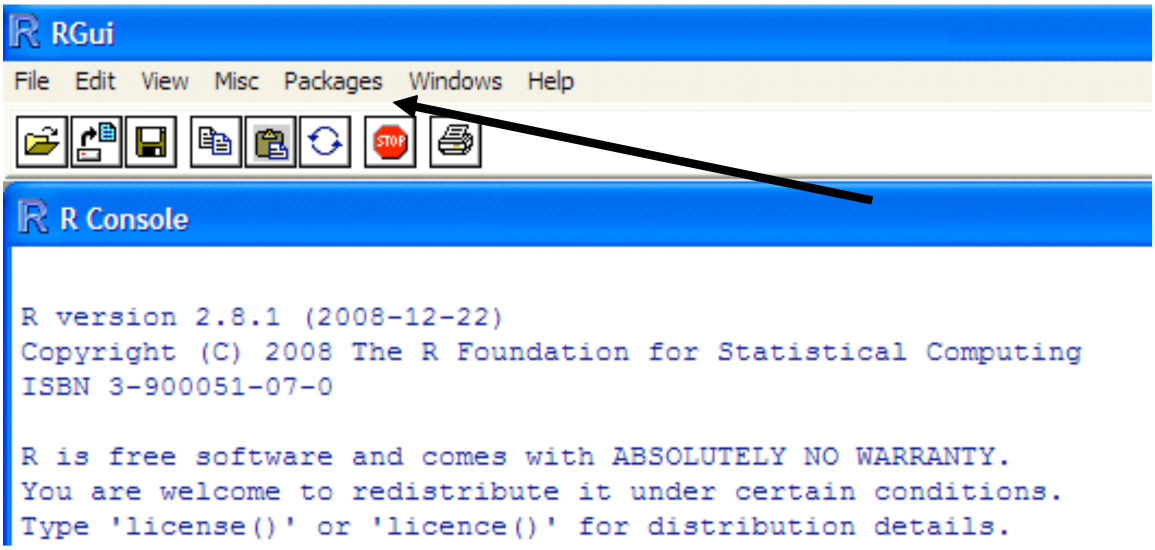
A build of the development version (which will eventually become the next major rele build.

In this directory:

README R-2.8.1	Installation and other instructions.
CHANGES	New features of this Windows version.
NEWS	New features of all versions.
R-2.8.1-win32.exe	Setup program (about 31 megabytes). Please download this fr
old	Previous releases.
md5sum.txt	md5sum output for the setup program. A Windows GUI versi http://www.md5summer.org/; a Windows command line versic http://www.etree.org/md5com.html.

1.2.2 Packages
An R installation contains one or more libraries of packages. Some of these packages are part of the base installation. Others can be downloaded from CRAN (see Appendix A), which currently hosts over 1000 packages for various purposes (Dalgaard, 2002). A package can contain functions written in the R language, and data sets. Most packages implement functionality that users will probably not need to have loaded all the time (Dalgaard, 2002).

Once a base installation is finished, you can install packages in R using: Open R and from the R window, go to the menu Packages.



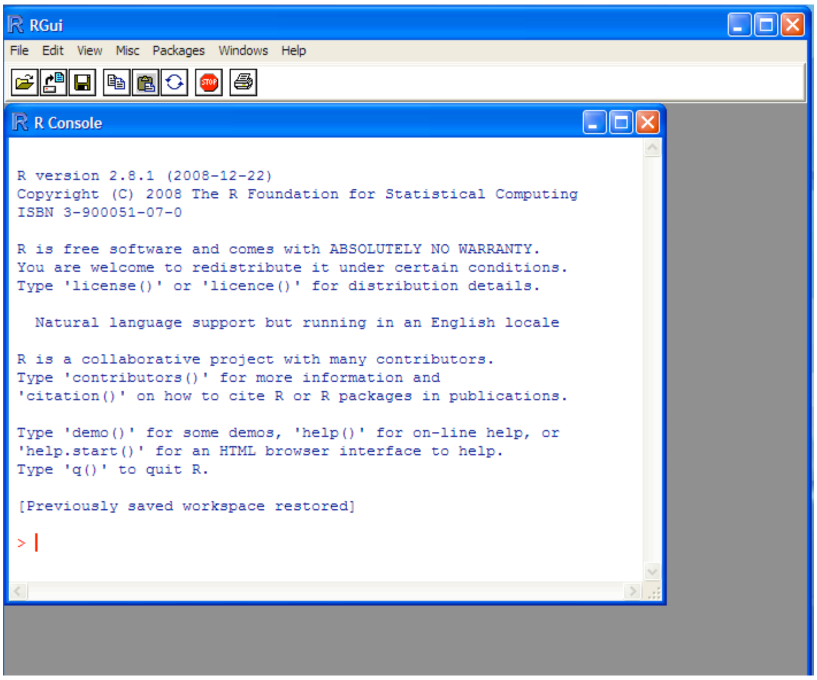
```
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
```

Then select Install package(s), choose a mirror and then select the package(s) that you need to install. This process needs to be done once for each package.

To use a package, you need to load the package. To do that, go to the menu Packages, then select Load package and choose the package(s) that you need. This process has to be done each time that you open a new session and wish to use a specific function in a package that is not in the base system.

1.3 A sample session

Before starting with this section be sure that you have a working installation of R. When you open R you should see the console window:



```
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> |
```

R works fundamentally using a question-and-answer model: You enter a line with a command and press Enter. Then the program does something, prints the result if relevant, and asks for more input. When R is ready for input, it prints out its prompt, a ">" symbol (Dalgaard, 2002).

One of the simplest possible tasks in R is to enter an arithmetic expression and receive a result (Dalgaard, 2002). The first line in red (font courier new) are inputs or instructions that we type, the second line is in blue (font courier new) are the outputs or answers from R.

```
> 3+2                   "Instruction or inputs"
[1] 5                   "Answers or Outputs"
```

We also can perform other standard arithmetic calculations:

```
> 4^2
[1] 16
```

```
> sqrt(36)
[1] 6
> pi
[1] 3.141593
> exp(1)
[1] 2.718282
```

The number in brackets is the index of the first number on that line (Dalgaard, 2002). Consider the case of generating the sequence of integers from 50 to 100

```
> 50:100
 [1]  50  51  52  53  54  55  56  57  58  59  60  61  62  63
[15]  64  65  66  67  68  69  70  71  72  73  74  75  76  77
[29]  78  79  80  81  82  83  84  85  86  87  88  89  90  91
[43]  92  93  94  95  96  97  98  99 100
```

Here [15] indicates that 64 is the fifteen element in the vector of output from this command.

One of the most common procedures in R is to store numbers or results. R, like other computer languages, has symbolic variables that are names that can be used to represent values (Dalgaard, 2002). To assign the value 10 to the variable *a*:

```
> a<-10
```

The two characters <- should be read as a single symbol: an arrow pointing to the variable to which the value is assigned. This is known as the assignment operator. Spacing around operators is generally disregarded by R, but notice that adding a space in the middle of a <- changes the meaning to "less than" followed by "minus" (Dalgaard, 2002). Also be aware that there is no immediately visible result, but from now on, *a* has the value 10 and can be used in subsequent arithmetic expressions.

```
> a
[1] 10
> a*2
[1] 20
> a/5
[1] 2
> a+2
[1] 12
```

R allows overwriting variables, without providing any warning that you are redefining a variable that had previously been assigned a value.

```
> a<-234
> a
[1] 234
> a<-456.43
> a
[1] 456.43
```

Technically R is an expression language with a very simple syntax. It is case sensitive, so *A* and *a* are different symbols and would refer to different variables. The set of symbols which can be used in R names depends on the operating system. Normally all alphanumeric symbols are allowed plus ‘.’ and ‘_’ (Venables *et al.* 2009). There is, however, the limitation that the name must not start with a digit or a period followed by a digit (Dalgaard, 2002).

```
> A
Error: object "A" not found
> 2<-10
Error in 2 <- 10 : invalid (do_set) left-hand side to assignment
> .3<-10
Error in 0.3 <- 10 : invalid (do_set) left-hand side to assignment
```

For entering small data sets we can use the function `c()`. This is a generic function which combines its arguments into a single data set.

```
> c(2,43,56,43,12,34,56,76)
[1]  2 43 56 43 12 34 56 76
```

This is useful as a method to generate variables as a vector or list of data

```
> x<-c(21,23,45,32,12,34,56,7,8,98)
> x
 [1] 21 23 45 32 12 34 56  7  8 98
```

1.4 How to get help

R has several ways to help the user. Some of these are:

Command in R	Result
`help(t.test)` or `?t.test`	These functions provide access to documentation. In the example R will provide documentation for the function t.test
`help.search(“anova”)`	Searches the help system for documentation matching a given character string. Names and titles of the matched help entries are displayed. In this example, a list of functions that are related to anova will be returned.
`apropos("test")`	Provides a list of command names that contain the pattern in quotes. This example lists commands that contain the word “test”.
`example(t.test)`	This initiates running an example, if available, of the use of the function specified by the argument function.
`help.start()`	Start the hypertext (currently HTML) version of R's online documentation.
`RSiteSearch("")`	Search for key words or phrases in the R-help mailing list archives, or R manuals and help pages, using the search engine at http://search.r-project.org and view them in a web browser.

1.5 Documentation

Additional documentation on R is available from several sources including:

1.5.1 Free documentation
Edited by the R Development Core Team. http://cran.r-project.org/manuals.html
Manuals, tutorials, etc. provided by users of R. http://cran.r-project.org/other-docs.html

1.5.2 Books
A list of books that are related to R. http://www.r-project.org/doc/bib/R-books.html

1.6 Data import

Large data objects will usually be read as values from external files rather than entered during an R session at the keyboard. R input facilities are simple with strict and somewhat inflexible requirements. There is a clear presumption by the designers of R that you will be able to modify your input files using other tools, such as file editors, to fit the requirements of R (Venables *et al.* 2009).

The most convenient way to read data into R is via the function called read.table. It requires that data be in "ASCII format"; that is, a "flat file" as created with Windows' NotePad or any plain-text editor, with extension .txt. The first line of the file can contain a header giving the names of the variables, a practice that is highly recommended (Dalgaard, 2002). Each subsequent line contains a row of data.

Command in R:
read.table(file, header = FALSE)

file: the name of the file from which the data are to be read. Each row of the table appears as one line of the file. If it does not contain an absolute path, the file name is relative to the current working directory.

header: logical value indicating whether the file contains the names of the variables as its first line. Default is FALSE

The working directory can be changed by using the menu item File, then change dir, and select the directory that you want. Also the current working directory can be obtained by getwd() and changed by setwd(mydir), where mydir is a character string containing the path to the desired working directory (Dalgaard, 2002).

Other commands in R to read data include: read.csv, read.csv2, read.delim, read.delim2 and the foreign package that are functions for reading and writing data stored by statistical packages such as Minitab, S, SAS, SPSS, Stata, Systat and others.

2. Descriptive Statistics

In sampled and whole population data, a measure of central tendency provides an assessment of an “average” of the data. A measure of dispersion, or a measure of variability, is an indication of the spread of measurements around the center of the distribution (Zar, 1999).

Example (from Samuels & Witmer 2003, p. 29)
Male mormon crickets (*Anabrus simplex*) sing to attract mates. A field researcher measured the duration of 51 unsuccessful songs, that is, the time until the singing male gave up and left his perch. Below are examples of computing basic descriptive statistics for such a dataset.

4.3	24	6.6	7.3	1.5	2.6	5.6	3.9	9.4	6.2	1.6	6.5	0.2	2.7	17	4	2
0.7	1.6	2.3	3.7	0.8	0.5	4.5	12	3.5	0.8	5.2	3.9	0.7	1.7	3.8	5	2
4.5	1.8	1.2	0.7	0.7	4.2	4.7	2.2	1.4	2.8	8.6	3.7	3.5	1.2	3.7	14	4

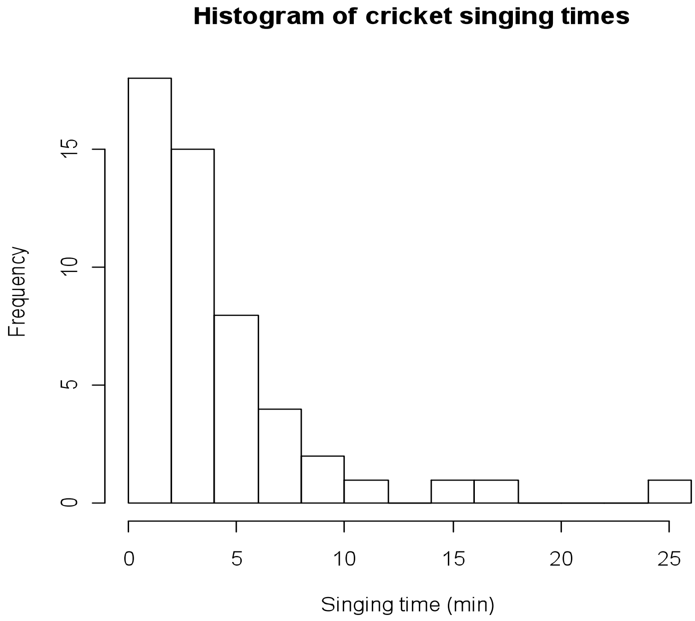
```
> cricket<-c(4.3,24.1,6.6,7.3,1.5,2.6,5.6,3.9,9.4,6.2,1.6,6.5,0.2,2.7,
+ 17.4,4, 2, 0.7,1.6,2.3,3.7,0.8,0.5,4.5,11.5,3.5,0.8,5.2,3.9,0.7,1.7,
+  3.8,5,2,4.5,1.8,1.2,0.7,0.7,4.2,4.7,2.2,1.4,2.8,8.6,3.7,3.5,1.2,3.7,
+ 14.1,4)
> cricket
 [1]  4.3 24.1  6.6  7.3  1.5  2.6  5.6  3.9  9.4  6.2  1.6  6.5  0.2
[14]  2.7 17.4  4.0  2.0  0.7  1.6  2.3  3.7  0.8  0.5  4.5 11.5  3.5
[27]  0.8  5.2  3.9  0.7  1.7  3.8  5.0  2.0  4.5  1.8  1.2  0.7  0.7
[40]  4.2  4.7  2.2  1.4  2.8  8.6  3.7  3.5  1.2  3.7 14.1  4.0

> mean(cricket)
[1] 4.335294
> median(cricket)
[1] 3.7

> var(cricket)
[1] 19.64793
> sd(cricket)
[1] 4.432598
> range(cricket)
[1]  0.2 24.1

> summary(cricket)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.200   1.600   3.700   4.335   4.850  24.100 
```

Histograms
Command in R:

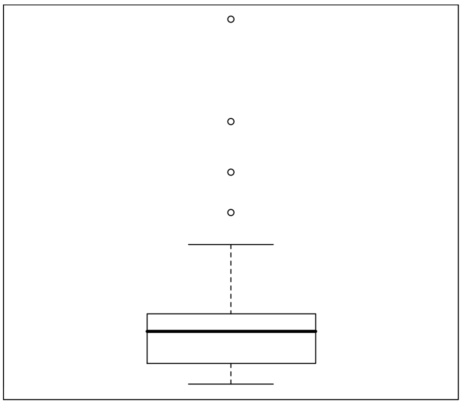
hist(x, breaks = "Sturges", freq = NULL, main , xlab , ylab,)
x: vector of data values for which the histogram is to be constructed.
breaks: a vector giving the breakpoints between histogram cells or a single number giving the number of cells for the histogram. Default is the algorithm of Sturges.

freq: logical; if TRUE, the histogram graphic is a representation of frequencies, the counts of data values in each cell of the histogram; if FALSE, the histogram is the fraction of data within each histogram cell (e.g. probability density)
main: The main title .
xlab: X axis label.
ylab Y axis label.

```
> hist(cricket, breaks =10,freq = T,
+ main = "Histogram of cricket singing times",
+ xlab = "Singing time (min)",
+ ylab = "Frequency")
```



```
> boxplot(cricket)
```



Note that a “+” at the start of a line indicates that this line is a continuation of the previous command line. As an exercise, add labels to the above boxplot using the same format as for the hist() command.

3. One and two sample tests

3.1 One sample test

R provides methods to use statistical tests to compare an observed mean value to a known or hypothetical mean, denoted μ_0 (Selvin, 2004).

The t-test: the command in R for a t-test:
t.test(x, alternative = c("two.sided", "less", "greater"), mu = 0, conf.level = 0.95)
x: vector containing the observations
alternative: character string specifying the alternative hypothesis. Default is two.sided
mu: hypothetical mean
conf.level: confidence level of the interval. Default is 0.95

Example (from Selvin 2004, p 209)
An experimental process employed to purify drinking water, to be useful, must not change the acidity of the treated water (ideally it would maintain a neutral pH of 7.0). To assess the process, the mean of a sample of pH-values is compared to the hypothetical mean pH of 7.0. The experimental process applied 24 times produces 24 observations which are assumed to be independent and normally distributed.

5.95	7.39	6.88	6.54	6.50	6.73	6.69	6.95
7.58	6.62	6.96	6.90	6.93	6.32	7.22	6.36
6.54	6.67	7.25	6.94	7.21	6.83	6.80	6.59

```
> observation<- c(5.95, 7.39, 6.88, 6.54, 6.50, 6.73, 6.69, 6.95,
+ 7.58, 6.62, 6.96, 6.90, 6.93, 6.32, 7.22, 6.36,
+ 6.54, 6.67, 7.25, 6.94, 7.21, 6.83, 6.80, 6.59)

> t.test(observation,mu=7.0)

        One Sample t-test
data:  observation
t = -2.591, df = 23, p-value = 0.01633
alternative hypothesis: true mean is not equal to 7
95 percent confidence interval:
 6.651562 6.960938
sample estimates:
mean of x
  6.80625
```

In conclusion the null hypothesis is rejected at the 5% level, because the p-value is below 0.05. The mean of the population that generated the sample is different than 7.

Other commands in R for single population tests are: wilcox.test, prop.test. Another very common test of one population is the Z-test. We are unaware of a command in R for this test, but you can find a complete procedure in Verzani 2002, p. 62.

3.2 Two sample test

One of the most common procedures in biostatistics is the comparison of two samples to infer whether differences exist between two observed populations (Zar, 1999). One command for this is:

Command in R:
t.test(x, y , alternative = c("two.sided", "less", "greater"), mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95)

x: vector with observations of sample 1
y: vector with observations of sample 2
alternative: character string specifying the alternative hypothesis. Default is two.sided
mu: difference in means. Default is 0
paired: a logical indicating whether you want a paired t-test. Default is false
var.equal: a logical indicating whether to treat the variances as equal. Default is false
conf.level: confidence level of the test. Default 0.95

3.2.1 t test unequal variance
Example (from Quinn & Keough 2002, p 40)
Furness & Bryant (1996) studied energy budgets of breeding northern fulmars (*Fulmarus glacialis*) in Shetland. As part of their study, they recorded various characteristics of individually labeled male and female fulmars. We will focus on differences in metabolic rate between sexes. There were eight males and six females labeled. The H_0 was that there is no difference between the sexes in the mean metabolic rate of fulmars. This is an independent, non-paired comparison because individual fulmars can only be either male or female.

Male	2950	2308.7	2135.6	1945.6	1195.5	843.3	525.8	605.7
Female	1956.1	1490.5	1361.3	1086.5	1091	727.7	-	-

Note that the ranges (and variances) are very different in these two samples. The small and unequal sample sizes, in conjunction with the unequal variances, indicate that a t test based on different variances is more appropriate (Quinn & Keough 2002). That is why in the command t.test, we don't use the argument var.equal, because the default is false. Note that in the below, the length of the two samples are the same but we use "NA" to indicate there is no data value.

```
> Male<-c(2950,2308.7,2135.6,1945.6,1195.5,843.3,525.8,605.7)
> Female<-c(1956.1,1490.5,1361.3,1086.5,1091,727.7, NA, NA)
> t.test(Male, Female)

        Welch Two Sample t-test
data:  Male and Female
t = 0.7732, df = 10.468, p-value = 0.4565
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -518.8042 1075.3208
sample estimates:
mean of x mean of y
 1563.775  1285.517
```

We would not reject the H_0 at the 95% confidence level and thus conclude there was no statistically significant difference in mean metabolic rate of fulmars between sexes (Quinn & Keough 2002).

3.2.2 Paired t test

Now we consider the comparison of two samples that are not independent but are paired. In this design, the observations occurs in pairs, the observational units in a pair are linked in some way, so they have more in common with each other that with other members of another pair (Samuels & Witmer, 2003).

Example (from Quinn & Keough 2002 , p 41)
Elgar *et al.* (1996) exposed 17 orb spiders each to dim and higher light conditions and recorded two aspects of web structure under each condition. The H_0's are that the two variables (vertical diameter and horizontal diameter of the orb web) were the same in dim and higher light conditions. Because the same spider spun their web in both light conditions, then it is appropriate to use a paired comparison.

PAIR	**VERTDIM**	**HORIZDIM**	**VERTLIGH**	**HORIZLIGH**
1	300	295	80	60
2	240	260	120	140
3	250	280	170	160
4	220	250	90	120
5	160	160	150	180
6	170	150	110	90
7	300	290	260	120
8	180	120	240	220
9	200	210	190	210
10	80	120	120	150
11	190	240	160	160
12	270	270	300	330
13	130	150	160	100
14	190	210	300	240
15	190	200	280	190
16	120	160	190	170
17	180	160	100	100

```
> HORIZDIM<-c(295,260,280,250,160,150,290,120,210,120,240,270,150,210,
+200,160,160)
> HORIZLIGH<-c(60,140,160,120,180,90,120,220,210,150,160,330,100,240,
+190,170,100)
> t.test(HORIZDIM, HORIZLIGH, paired=T)
        Paired t-test
data:  HORIZDIM and HORIZLIGH
t = 2.1482, df = 16, p-value = 0.04735
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  0.6085725 91.7443687
sample estimates:
mean of the differences
               46.17647
```

So we would reject the H_0 at the 95% level and conclude that, for the population of orb spiders, there is a difference in the mean horizontal diameter of spider webs between higher light and dim conditions (Quinn & Keough 2002). We leave the comparison of the vertical diameter as an exercise.

Other commands in R for two-sample tests include: wilcox.test, prop.test.

4. Single factor Analysis of variance

The above methods are appropriate to analyze measurements of a single variable from two samples. For observations of a variable using three or more samples, multi-sample analysis is required. To test the null hypotheses H_0: $\mu 1 = \mu_2 = \ldots = \mu_k$, where k is the number of experimental groups, or samples, a standard method is to carry out an analysis of variance (Zar, 1999).

4.1 Parametric Analysis of variance (ANOVA)

There are several R commands to perform this analysis. We here use the command aov, because using various formulas in the arguments allows several different types of design to be considered. The end of this section mentions other commands for ANOVA.

Command in R:
aov(formula, data = NULL)

formula: A formula specifying the model. A formula is an expression of the form *y ~ model* that is interpreted as a specification that the response *y* is modeled by a linear predictor specified symbolically for the desired model.

data: A data set in R in which the variables specified in the formula will be found.

Example (from Zar 1999, p. 180)
Nineteen pigs are assigned at random among four experimental groups. Each group is fed a different diet. The data are pig body weights, in kilograms, after being raised on these diets. We wish to ask whether pig weights are the same for all four diets.

Feed1	**Feed2**	**Feed3**	**Feed4**
60.8	68.7	102.6	87.9
57	67.7	102.1	84.2
65	74	100.2	83.1
58.6	66.3	96.5	85.7
61.7	69.8		90.3

Below illustrates the correct arrangement of the data in a .txt file and then we import the data to R (see section 1.6). The data format in the .txt file is:

```
weights     diet
60.8        Feed1
57          Feed1
65          Feed1
58.6        Feed1
61.7        Feed1
68.7        Feed2
67.7        Feed2
74          Feed2
```

66.3	Feed2
69.8	Feed2
102.6	Feed3
102.1	Feed3
100.2	Feed3
96.5	Feed3
87.9	Feed4
84.2	Feed4
83.1	Feed4
85.7	Feed4
90.3	Feed4

We create this file in our working directory (assumed to be C:) with the name pigs.txt and read it into the R variable "pigs" (see section 1.6 for more details). We will use this example also for section 4.2.

```
> pigs<-read.table("pigs.txt", header=T)
> exit_pigs<-aov(weights~diet, data=pigs)
> summary(exit_pigs)
            Df Sum Sq Mean Sq F value    Pr(>F)
diet         3 4226.3  1408.8  164.64 1.061e-11 ***
Residuals   15  128.4     8.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can reject H_0, implying that the mean weights of pigs on the four diets are not equal. Note that we don't know from this analysis which of the treatments is the one providing highest pig weights, or a ranking of the various diets for this objective. A multiple comparison test (see section 5) would be used for further analysis of these additional questions.

Other commands in R: anova, Anova, lm, anova.lm.

4.2 Assumptions

The decompositions of the variability in the observations through an analysis of variance is just an algebraic relationship. To use the method to test formally for no differences in the means of various treatments requires that certain assumptions be satisfied. Specifically, these assumptions are that the residuals are normally and independently distributed with mean zero and constant but unknown variance (Montgomery, 2001).

4.1.1 Normality

Here we explain how to test the hypothesis that the residuals follow a normal distribution. We give details for the Shapiro Wilk test and the normal quantile quantile plots or q-q plots. The first step to test the assumption of normality is to obtain the residuals, using the following command (we continue with the last example)

```
> exit_pigs$residuals

   1     2     3     4     5     6     7     8     9    10    11
```

```
 0.18 -3.62  4.38 -2.02  1.08 -0.60 -1.60  4.70 -3.00  0.50  2.25
   12    13    14    15    16    17    18    19
 1.75 -0.15 -3.85  1.66 -2.04 -3.14 -0.54  4.06
```

Command in R:
Shapiro.test(x)
qqnorm(x) following with qqline(x).
  x: Numeric vector of data values. Missing values are allowed.

```
> shapiro.test(exit_pigs$residuals)
        Shapiro-Wilk normality test
data:  exit_pigs$residuals
W = 0.9511, p-value = 0.4132
```

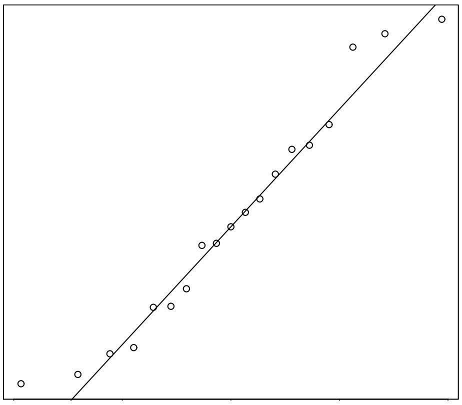
Given the value of p we cannot reject the H_0, and thus the evidence is that the data have a normal distribution.

For the graphical procedure:

```
> qqnorm(exit_pigs$residuals)
```

**Don't close the window with the graph

```
> qqline(exit_pigs$residuals)
```



Given that in general the dots are close to the line, it is reasonable to infer that the data follow a normal distribution. We have to be careful because purely graphical methods are always subject to the interpretation of the researcher.

Other commands in R: The package nortest has several tests of normality including Anderson-Darling, Cramer-von Mises and Lilliefors.

4.1.2 Homogeneity of variances (Homoscedasticity)

If we have three or more samples, and we compute a variance for each, then we can test the hypothesis that all sample came from populations with identical variances (Zar, 1999). Here we explain Bartlett's test and the plot of residuals against fitted values; we also use the data from section 4.1.

Command in R:
bartlett.test(formula, data)

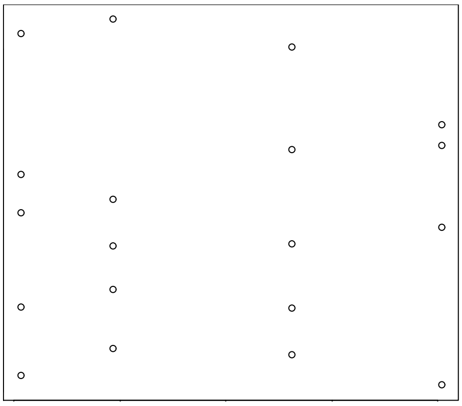
formula: Formula of the form lhs ~ rhs where lhs gives the data values and rhs the corresponding groups.
data: Data frame containing the variables in the formula.

```
> bartlett.test(weights~diet, data=pigs)
        Bartlett test of homogeneity of variances
data:  weights by diet
Bartlett's K-squared = 0.0328, df = 3, p-value = 0.9984
```

From this test we can say that we have homogeneity of variances.

For the graphical procedure:

```
> plot(exit_pigs$fitted.values, exit_pigs$residuals, main = "
+ Residuals vs Fitted", xlab="Fitted", ylab="Residuals")
```



Constancy of the residual variance is shown in these plots by the plots having about the same extension of scatter of the residual around zero for each factor level or treatment (Kutner *et al.* 2005). Then in this particular case the graph indicates that we have homogeneity of variances.

Other commands in R: levene.test (Package car)

4.3 Nonparametric Analysis (Kruskal- Wallis)

If a set of data is collected according to a completely randomized design, it is possible to test nonparametrically for difference among groups. This may be done by the Kruskal-Wallis test, often called an analysis of variance by ranks. This test may be used in situations where the parametric single-factor ANOVA is not applicable (Zar, 1999).

Command in R:
kruskal.test(formula, data)

formula: Formula of the form lhs ~ rhs where lhs gives the data values and rhs the corresponding groups.
data: Data frame containing the variables in the formula.

Example (Zar 1999, p. 197)
An entomologist is studying the vertical distribution of a fly species in a deciduous forest and obtains five collections of the flies in three different vegetation layers: herb, shrub and tree. The entomologist want to test the H_0: The abundance of the flies is the same in all three vegetation layers.

Herbs	14	12.1	9.6	8.2	10.2
Shrubs	8.4	5.1	5.5	6.6	6.3
Trees	6.9	7.3	5.8	4.1	5.4

The data have to be in a txt file in the following way:

abundance	layers
14	Herbs
12.1	Herbs
…	…
4.1	Trees
5.4	Trees

We create this file in our working directory with the name flies (see section 1.6):

```
> flies<-read.table("flies.txt", header=T)
> bartlett.test(abundance~layers, data=flies)

        Bartlett test of homogeneity of variances
data:  abundance by layers
Bartlett's K-squared = 1.7057, df = 2, p-value = 0.4262
```

Since we don’t have homogeneity of variance, we should perform a Kruskal-Wallis test.

```
> kruskal.test(abundance~layers, data=flies)

        Kruskal-Wallis rank sum test
data:  abundance by layers
Kruskal-Wallis chi-squared = 8.72, df = 2, p-value = 0.01278
```

We can conclude that the abundance of the flies is different in the three layers of vegetation.

5. Multiple Comparison Tests

Suppose that in conducting an analysis of variance the null hypothesis is rejected. Thus, there are differences between the treatment means, but exactly which means differ is not specified. Sometimes in this situation, further comparison and analysis among groups of treatment means may be useful. The procedures for making these comparisons are usually called multiple comparison methods (Montgomery, 2001). In general these methods consider the null hypothesis H_0: $\mu_A = \mu_B$ versus the alternative hypothesis, where the subscripts denote any possible pair of groups (Zar, 1999)

5.1 Tukey test

A much-used multiple comparison procedure is the Tukey test, also know as the honestly significant difference test (Zar, 1999).

Command in R:

```
HSD.test(y, trt, DFerror, MSerror, alpha = 0.05, group=TRUE)
  y: Variable response
  trt: Treatments
  DFerror: Degrees of freedom of the residuals. Take from the ANOVA table
  MSerror: Mean Square Error of the residuals. Take from the ANOVA table
  alpha: Significant level.
  group: TRUE or FALSE. Use always TRUE.
```

For this command we need the package agricolae (see section 1.2.2).

Example (Quinn & Keough 2002, p. 174)

Medley & Clements (1998) sampled a number of stations (between four and seven) on six streams known to be polluted by heavy metals in the Rocky Mountain region of Colorado, USA. They recorded zinc concentration, and species richness and species diversity of the diatom community and proportion of diatom cells that were the early-successional species, *Achanthes minutissima*. The first analysis compares mean diatom species diversity (response variable) across the four zinc-level groups (categorical predictor variable), zinc level treated as a fixed factor. The H_0 was no difference in mean diatom species diversity between zinc-level groups.

ZINC	**DIV**	**ZINC**	**DIV**	**ZINC**	**DIV**	**ZINC**	**DIV**	**ZINC**	**DIV**
BACK	2.27	MED	2.19	LOW	1.83	MED	1.75	HIGH	1.04
HIGH	1.25	MED	2.1	LOW	1.88	LOW	2.83	LOW	2.18
HIGH	1.15	BACK	2.2	MED	2.02	BACK	1.53	BACK	1.89
MED	1.62	MED	2.06	MED	1.94	BACK	0.76	HIGH	1.37
BACK	1.7	HIGH	1.9	LOW	2.1	MED	0.8	LOW	1.4
HIGH	0.63	HIGH	1.88	LOW	2.38	LOW	1.66	BACK	1.98
BACK	2.05	HIGH	0.85	HIGH	1.43	MED	0.98		

The data have to be in a txt file in the following way:

```
ZINC   DIV
BACK   2.27
HIGH   1.25
…      …
```

LOW	1.66
MED	0.98

We create this file in our working directory with the name streams (see section 1.6).

```
> streams<-read.table("streams.txt", header=T)
> exit_streams<-aov(DIV~ZINC, data=streams)
> summary(exit_zinc)
            Df Sum Sq Mean Sq F value   Pr(>F)  
ZINC         3 2.5666  0.8555  3.9387 0.01756 *
Residuals   30 6.5164  0.2172                  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can reject H0. Now to see the difference between each level of zinc we perform a Tukey test.

```
> HSD.test(streams$DIV, streams$ZINC, 30, 0.2172, group=T)

HSD Test for streams$DIV 
                                        ......
Alpha                                 0.050000
Error Degrees of Freedom             30.000000
Error Mean Square                     0.217200
Critical Value of Studentized Range   3.845401

Treatment Means
  streams.ZINC streams.DIV   std.err replication
1         BACK    1.797500 0.1715658           8
2         HIGH    1.277778 0.1422906           9
3          LOW    2.032500 0.1573298           8
4          MED    1.717778 0.1676701           9

Honestly Significant Difference 0.6157647 
Harmonic Mean of Cell Sizes  8.470588 

Different HSD for each comparison
Means with the same letter are not significantly different.

Groups, Treatments and means
a        LOW     2.0325 
ab       BACK    1.7975 
ab       MED     1.717778 
 b       HIGH    1.277778 
   trt    means  M        N   std.err
1  LOW 2.032500  a 8.470588 0.1573298
2 BACK 1.797500 ab 8.470588 0.1715658
3  MED 1.717778 ab 8.470588 0.1676701
4 HIGH 1.277778  b 8.470588 0.1422906
```

The only H_0 to be rejected is that of no difference in diatom diversity between sites with low zinc and sites with high zinc (Samuels & Witmer, 2003). We leave as exercise the following: check the assumptions (normality and homoscedasticity) of the model and perform the Tukey test for the example in section 4.1.

5.2 Least significant difference (LSD) test

Command in R:

LSD.test(y, trt, DFerror, MSerror, alpha = 0.05, group=TRUE)

y: Variable response
trt: Treatments
DFerror: Degrees of freedom of the residuals. Take from the ANOVA table
MSerror: Mean Square Error of the residuals. Take from the ANOVA table
alpha: Significant level.
group: TRUE or FALSE. Use always TRUE.

For this command we need the package agricolae (see section 1.2.2).

Example (Zar 1999, p. 210)

Researchers want to perform an ANOVA table and a multiple test comparison. For the following data:

Grayson's Pond	28.2	33.2	36.4	34.6	29.1	31
Beaver Lake	39.6	40.8	37.9	37.1	43.6	42.4
Angler's Cove	46.3	42.1	43.5	48.8	43.7	40.1
Appletree Lake	41	44.1	46.4	40.2	38.6	36.3
Rock River	56.3	54.1	59.4	62.7	60	57.3

The data are strontium concentrations (mg/ml) in five different bodies of water. The data have to be in a txt file in the following way:

strontium	bodies
28.2	Grayson
33.2	Grayson
…	…
60	Rock
57.3	Rock

We create this file in our working directory with the name water (see section 1.6).

```
> water<-read.table("water.txt", header=T)
> exit_water<-aov(strontium~bodies, data=water)
> summary(exit_water)
            Df  Sum Sq Mean Sq F value     Pr(>F)    
bodies       4 2193.44  548.36  56.155 3.948e-12 ***
Residuals   25  244.13    9.77                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Because a significant p value resulted from the analysis of variance, the LSD test is now applied on the means.

```
> LSD.test(water$strontium, water$bodies, 25, 9.77, group=T)

LSD t Test for water$strontium
                            ......
```

```
Alpha                     0.050000
Error Degrees of Freedom 25.000000
Error Mean Square         9.770000
Critical Value of t       2.059539

Treatment Means
  water.bodies water.strontium  std.err replication
1       Angler        44.08333 1.257621           6
2        Apple        41.10000 1.496663           6
3       Beaver        40.23333 1.033011           6
4      Grayson        32.08333 1.308540           6
5         Rock        58.30000 1.239624           6

Least Significant Difference 3.716692
Means with the same letter are not significantly different.

Groups, Treatments and means
a        Rock    58.3
 b       Angler          44.08333
 bc      Apple   41.1
  c      Beaver          40.23333
   d     Grayson         32.08333
      trt    means    M N  std.err
1    Rock 58.30000    a 6 1.239624
2  Angler 44.08333    b 6 1.257621
3   Apple 41.10000   bc 6 1.496663
4  Beaver 40.23333    c 6 1.033011
5 Grayson 32.08333    d 6 1.308540
```

We leave as exercise the following: give the appropriate conclusions from this analysis of last example, check the assumptions (normality and homoscedasticity) of the model and perform the Tukey test for the example in this section.

Other commands in R: TukeyHSD, waller.test (package agricolae)

6. Other Analysis of variance

6.1 Randomized block design

In any experiment, variability arising from a nuisance factor can affect the results. We define a nuisance factor as a design factor that probably has an effect on the response, but we are not interested in that effect. When the nuisance source of variability is known and controllable, a design technique called blocking can be used to systematically eliminate its effect on the statistical comparison among treatments (Montgomery, 2001). The statistical term "block" is conceptually an extension of the term "pair" introduced in section 3.2.2 (Zar, 1999).

6.1.1 Parametric Analysis of variance

The randomized block ANOVA is an extension of the one – way ANOVA presented in section 4.1.

Command in R:

```
aov(formula, data = NULL)
```

formula: A formula specifying the model. Formula of the form a ~ b + c, where a, b and c give the data values and corresponding groups and blocks, respectively.

data: A data frame in which the variables specified in the formula will be found.

Example (from Samuels & Witmer 2003, p. 487)

Researchers were interested in the effect that acid has on the growth rate of alfalfa plants. They created three treatment groups in an experiment: low acid, high acid and control. The response variable in their experiment was the average height of the alfalfa plants in a Styrofoam cup after five days of growth. The observational unit was a cup, rather than individual plants. They had 5 cups for each of the 3 treatments, for a total of 15 observations. However, the cups were arranged near a window and they wanted to account for the effect of differing amounts of sunlight. Thus they created 5 blocks and randomly assigned the 3 treatments within each block. The data are given in the following table:

	treatments		
block	**low**	**high**	**control**
1	1.58	1.1	2.47
2	1.15	1.05	2.15
3	1.27	0.5	1.46
4	1.25	1	2.36
5	1	1.5	1

The data have to be in a txt file in the following way:

```
height  trt     block
2.47    control block1
2.15    control block2
1.46    control block3
2.36    control block4
1       control block5
…       …       …
1.1     high    block1
```

1.05	high	block2
0.5	high	block3
1	high	block4
1.5	high	block5

We create this file in our working directory with the name alfalfa (see section 1.6).

```
> alfalfa<-read.table("alfalfa.txt", header=T)
> exit_alfalfa<-aov(height~trt + block, data=alfalfa)
> summary(exit_alfalfa)
            Df  Sum Sq Mean Sq F value  Pr(>F)
trt          2 1.98601 0.99301  5.4709 0.03182 *
block        4 0.83963 0.20991  1.1565 0.39740
Residuals    8 1.45205 0.18151
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p value for the treatment (trt) is small, indicating that the differences between the three sample means are greater than would be expected by chance alone (Samuels & Witmer, 2003).

6.1.2 Nonparametric analysis of variance (Friedman)

Friedman’s test is a nonparametric analysis that may be performed on a randomized block experimental design, and it is especially useful with data which do not meet the parametric analysis of variances assumptions of normality and homoscedasticity (Zar, 1999).

Command in R:

friedman.test(formula, data)

formula: Formula of the form a ~ b | c, where a, b and c give the data values and corresponding groups and blocks, respectively.

data: Data frame in which the variables specified in the formula will be found.

Example (from Zar 1999, p. 264)

We want to investigate the H_0 that the mean weight gain of guinea pigs is the same on each of four specified diets. Each guinea pig is housed in a separate cage. A block consists of a group of four animals that we can be reasonably assured will experience identical environmental conditions (light, temperature, draft, noise, etc). Each block has each of its four animals assigned at random to one of the four experimental diets, so that each animal in a given block is to receive a different diet. The data (weight gains, in grams) are summarized in the following table.

	diets			
Blocks	**1**	**2**	**3**	**4**
1	7	5.3	4.9	8.8
2	9.9	5.7	7.6	8.9
3	8.5	4.7	5.5	8.1
4	5.1	3.5	2.8	3.3
5	10.3	7.7	8.4	9.1

The data have to be in a txt file in the following way:

```
weight diets   blocks
7      diet1   block1
9.9    diet1   block2
…      …       ….
3.3    diet4   block4
9.1    diet4   block5
```

We create this file in our working directory with the name guinea (see section 1.6).

```
> guinea<-read.table("guinea.txt", header=T)
> friedman.test(weight~ diets | blocks, data=guinea)

        Friedman rank sum test
data:  weight and diets and blocks
Friedman chi-squared = 10.68, df = 3, p-value = 0.01359
```

Therefore, we reject H_0.

6.2 Factorial structure

Many experiments involve the study of the effects of two or more factors. In general, factorial designs are most efficient for this type of experiment. By a factorial design, we mean that in each complete trial or replication of the experiment all possible combinations of the levels of the factors are investigated (Montgomery, 2001).

Command in R:

aov(formula, data = NULL)

formula: A formula specifying the model. Formula of the form a ~ b * c, where a, b and c give the data values and corresponding factor1 and factor2, respectively.

data: A data frame in which the variables specified in the formula will be found.

Example (from Samuels & Witmer 2003, p. 6)

Before new drugs are given to humans subjects, it is common practice to test them first in dogs or other animals. In part of one study, a new drug under investigation was given to 4 male and 4 female dogs, at doses 8mg/kg and 25mg/kg. Alkaline phosphatase level (measured in U/Li) was measured from blood samples in order to screen for toxicity problems in dogs before starting with humans. The design of this experiment allows for the investigation of the interaction of two factors: sex of the dog and dose. Data are shown in the following table:

Dose	**Male**	**Female**
8	191	150
	154	127
	194	152
	183	105
25	80	141
	49	153
	78	171
	71	197

The data have to be in a txt file in the following way:

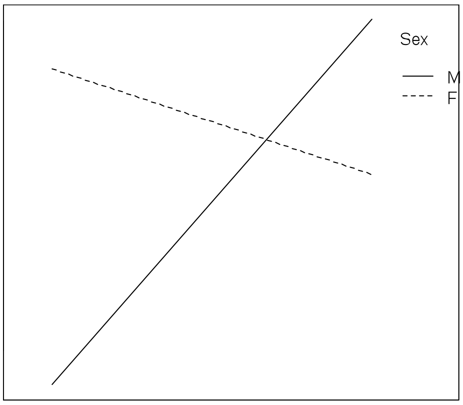
Sex	Dose	Level
M	dose8	191
M	dose8	154
M	dose8	194
M	dose8	183
…	…	…
F	dose25	141
F	dose25	153
F	dose25	171
F	dose25	197

We create this file in our working directory with the name dogs (see section 1.6).

```
> dogs<-read.table("dogs.txt", header=T)
> exit_dogs<-aov(Level~ Dose*Sex, data=dogs)
> summary(exit_dogs)
            Df  Sum Sq Mean Sq F value    Pr(>F)
Dose         1  6241.0  6241.0 15.4289  0.002006 **
Sex          1  2401.0  2401.0  5.9357  0.031367 *
Dose:Sex     1 20449.0 20449.0 50.5538 1.230e-05 ***
Residuals   12  4854.0   404.5
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

From the last analysis we can reject all H_0. The factors sex and dose interacted in the following sense: For females the effect of increasing the dose from 8 to 25 was positive (the average increases), but for males the effect of increasing the dose from 8 to 25 was negative (the average decreases). We can see this in the interaction plot.

```
> interaction.plot(dogs$Dose, dogs$Sex, dogs$Level,
+ xlab="Dose", ylab="Level", trace.label="Sex")
```



Example (from Samuels & Witmer 2003, p. 490)
A plant physiologist investigated the effect of mechanical stress on the growth of soybean plants. Individually potted seedlings were randomly allocated to four treatment groups of 13 seedlings each. Seedlings in two groups were stressed by shaking for 20 minutes twice daily, while two control groups were not stressed. Thus, the first factor in the experiment was presence or absence of stress. Also, plants were growth in either low or moderate light. Thus the second factor was amount of light. This experiment is an example of a 2*2 factorial experiment.

Low		**moderate**	
Control	**stress**	**control**	**stress**
264	235	314	283
200	188	320	312
225	195	310	291
268	205	340	259
215	212	299	216
241	214	268	201
232	182	345	267
256	215	271	326
229	272	285	241
288	163	309	291
253	230	337	269
288	255	282	282
230	202	273	257

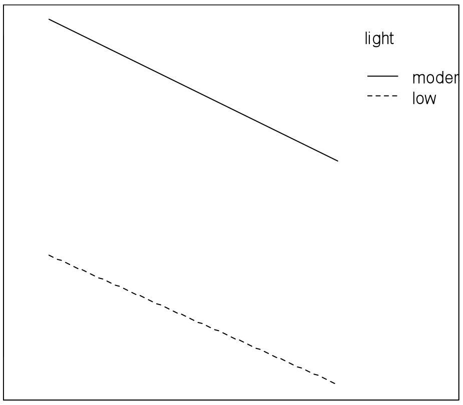
The data have to be in a txt file in the following way:

```
area    shaking   light
264     control    low
200     control    low
225     control    low
268     control    low
…       …          …
291     stress     moderate
269     stress     moderate
282     stress     moderate
257     stress     moderate
```

We create this file in our working directory with the name soybean (see section 1.6).

```
> soybean<-read.table("soybean.txt", header=T)
> exit_soybean<-aov(area~shaking*light, data=soybean)
> summary(exit_soybean)
              Df Sum Sq Mean Sq F value    Pr(>F)    
shaking        1  14858   14858 16.5954 0.0001725 ***
light          1  42752   42752 47.7490 1.010e-08 ***
shaking:light  1     26      26  0.0294 0.8645695    
Residuals     48  42976     895                      
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> interaction.plot(soybean$shaking, soybean$light, soybean$area,
+ xlab="shaking", ylab="area", trace.label="light")
```



Conclusions from this example are left as an exercise.

7. Regression and correlation

7.1 Regression

In many problems there are two or more variables that are related, and it is of interest to model and explore this relationship. In general, suppose that there is a single dependent variable or response y that depends on k independent variables, for example, $x_1, x_2, \ldots, x_k$. The relationship between these variables is characterized by a mathematical model called a regression model (Montgomery, 2001).

Command in R:
lm(formula, data = NULL)
  formula: A formula specifying the model.
  data: A data frame in which the variables specified in the formula will be found.

Example (from Quinn & Keough 2002, p.79)
Christensen *et al.* (1996) studied the relationships between coarse woody debris (CWD) and shoreline vegetation and lake development in a sample of 16 lakes in North America. The main variables of interest are the density of cabins (no. km^{-1}), density of riparian trees (trees km^{-1}), the basal area of riparian trees (m^2 km^{-1}), density of coarse woody debris (no. km^{-1}) and basal area of coarse woody debris (m^2 km^{-1}). The researchers are interested in fitting a linear regression model to CWD basal area against riparian tree density (RTD).

LAKE	CWD	RTD	LAKE	CWD	RTD	LAKE	CWD	RTD
Bay	121	1270	Palmer	65	1330	Lake_hills	97	976
Bergner	41	1210	Street	52	964	Towanda	1	771
Crampton	183	1800	Laura	12	961	Black oak	4	833
Long	130	1875	Annabelle	46	1400	Johnson	1	883
Roach	127	1300	Joyce	54	1280	Arrowhead	4	956
Tenderfoot	134	2150						

```
> CWD<-c(121,41,183,130,127,134,65,52,12,46,54,97,1,4,1,4)
> RTD<-c(1270,1210,1800,1875,1300,2150,1330,964,961,1400,1280,976,771,
+ 833,883,956)
> exit_lakes<-lm(CWD~RTD)
> summary(exit_lakes)
Call:
lm(formula = CWD ~ RTD)

Residuals:
   Min      1Q Median      3Q     Max 
-38.62 -22.41 -13.33  26.15  61.36 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -77.09908   30.60801  -2.519 0.024552 *  
RTD           0.11552    0.02343   4.930 0.000222 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1 

Residual standard error: 36.32 on 14 degrees of freedom
Multiple R-squared: 0.6345,     Adjusted R-squared: 0.6084
```

```
F-statistic:  24.3 on 1 and 14 DF,  p-value: 0.0002216
```

The t test and the ANOVA F test cause us to reject the H_0 that β_1 equals zero. We would also reject the H_0 that β_0 equals zero, although this test is of little biological interest. The r^2 value (0.634) indicates that we can explain about 63% of the total variation in CWD basal area by the linear regression with riparian tree density. We can predict CWD basal area for a new lake with 1500 trees km^{-1} in the riparian zone. Plugging 1500 into our fitted regression model:

CWD basal area = -77.099 + 0.116*1500

The predicted basal area of CWD is 96.901 m^2 km^{-1} (Quinn & Keough 2002).

7.2 Correlation

In many kinds of biological data, the relationship between two (or more variables) is not of clearly of strict dependence between the variables. In such cases, the magnitude of one of the variables changes as the magnitude of the second variable changes, but it is not reasonable to consider there to be an independent variable and a dependent variable whch is causally related to it. In such situations, correlation, rather than regression, analysis is called for (Zar, 1999).

Command in R:
cor.test(x, y, method = c("pearson", "kendall", "spearman"), conf.level = 0.95)
x, y: Numeric vectors of data values. x and y must have the same length.
method: Character string indicating which correlation coefficient is to be used for the test. One of "pearson", "kendall", or "spearman", can be abbreviated. Default Pearson.
conf.level: confidence level for the returned confidence interval. Default 0.95.

Example (from Quinn & Keough 2002, p.73)
Green (1997) studied the ecology of red land crabs on Christmas Island and examined the relationship between the total biomass of red land crabs and the density of their burrows within 25 m^2 quadrats (sampling units) at five forested sites on the island. We will look at two of these sites: there were ten quadrats at Lower Site (LS) and eight quadrats at Drumsite (DS). Pearson's correlation coefficient was considered appropriate for these data although more robust correlations were calculated for comparison.

DS	**TOTMASS**	2.15	2.27	4.31	2.58	3.23	1.83	1.54	2		
	BURROWS	39	38	61	79	35	39	45	28		
LS	**TOTMASS**	4.36	4.01	3.33	2.63	4.46	3.96	4.18	4.21	2.54	4.29
	BURROWS	38	37	27	18	41	33	40	29	25	38

```
> BURROWS<-c(39,38,61,79,35,39,45,28)
> TOTMASS<-c(2.15,2.27,4.31,2.58,3.23,1.83,1.54,2)
> cor.test(BURROWS, TOTMASS)

        Pearson's product-moment correlation
data:  BURROWS and TOTMASS
t = 1.0428, df = 6, p-value = 0.3372
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4322803  0.8592175
sample estimates:
      cor
0.3917155

> cor.test(BURROWS, TOTMASS, method=c("spearman"))

        Spearman's rank correlation rho
data:  BURROWS and TOTMASS
S = 69.9159, p-value = 0.6915
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.1676677

> cor.test(BURROWS, TOTMASS, method=c("kendall"))

        Kendall's rank correlation tau
data:  BURROWS and TOTMASS
z = 0.1247, p-value = 0.9008
alternative hypothesis: true tau is not equal to 0
sample estimates:
       tau
0.03636965
```

The H_0 of no linear relationship between total crab biomass and number of burrows at DS could not be rejected. The same conclusion applies for monotonic relationships measured by Spearman and Kendall's coefficients. So there was no evidence for any linear or more general monotonic relationship between burrow density and total crab biomass at site DS (Quinn & Keough 2002). We leave the analysis of site LS as exercise.

Other commands in R: cor.

Bibliography

Christensen, D.L., Herwig, B.R., Schindler, D.E. and Carpenter, S.R. 1996. Impacts of lakeshore residential development on coarse woody debris in north temperate lakes. Ecological Applications 64: 1143–1149.

Dalgaard, P. 2002. Introductory Statistics with R. Springer. USA. 267p.

Elgar, M.A., Allan, R.A. and Evans, T.A. 1996. Foraging strategies in orb-spinning spiders: ambient light and silk decorations in *Argiope aetherea Walckenaer* (Araneae: Araneoidea). Australian Journal of Ecology 21: 464–467.

Furness, R.W. and Bryant, D.M. 1996. Effect of wind on field metabolic rates of breeding Northern Fulmars. Ecology 77: 1181–1188.

Green, P.T. 1997. Red crabs in rain forest on Christmas Island, Indian Ocean: activity patterns, density and biomass. Journal of Tropical Ecology 13: 17–38.

Kutner, M., Nachtsheim, C., Neter, J. and Li, W. 2005. Applied linear statistical models. Fifth edition. McGraw-Hill/Irwin. USA. 1396p.

Montgomery, D. 2001. Design and analysis of experiments. Fifth edition. John Wiley & sons, INC. USA. 684p.

Quinn, G. and Keough, M. 2002. Experimental Design and Data Analysis for Biologists. Cambridge University Press. USA. 557p.

Samuels, M. and Witmer, J. 2003. Statistics for The Life Sciences. Third Edition. Pearson Education. USA. 724p.

Selvin, S. 2004. Biostatistics How it works. Pearson Education. USA. 394p.

Venables W. N., Smith D. M. and R Development Core Team. 2009. An Introduction to R. Notes on R: A Programming Environment for Data Analysis and Graphics Version 2.9.0. URL http://cran.cnr.berkeley.edu/doc/manuals/R-intro.pdf

Verzani, J. 2002. Simple R – Using R for introductory Statistics. URL http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf

Zar, J. 1999. Biostatistical Analysis. Fourth Edition. Prentice Hall. USA. 663p.