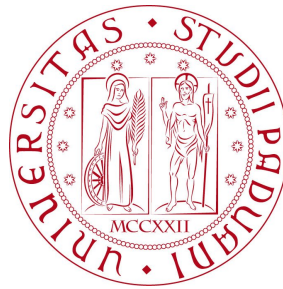


Introduction to R

Alberto Garfagnini

Università di Padova

AA 2021/2022 - Rlecture 1



What is R ?

- R is a **language and environment** for statistical computing and graphics
- similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues.
- **R provides a wide variety of statistical** (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) **and graphical techniques**
- R is **highly extensible**
- R is **available as Free Software** (GNU GPL) and it **compiles and runs on** a wide variety of **UNIX platforms, Windows and MacOS**
- The latest R version is 4.1.2 (Bird Hippie), released on November 1st, 2021
- A New R version, 4.1.3 (One Push-Up), will be released on March 10, 2022

R Web Resources

- R Web Site: <https://www.r-project.org/>
- R source code: <https://cran.r-project.org/src/base/R-4/>
- R Developer Page: <https://developer.r-project.org/>
- A list of changes in the new version can be found here:
<https://cran.r-project.org/doc/manuals/r-release/NEWS.html>



R is 22 years old

- the first version of R (1.0.0) was released on Feb 29, 2000
 - original release message: <https://stat.ethz.ch/pipermail/r-announce/2000/000127.html>
- the 20 years celebration took place in Copenhagen, (28-29 Feb 2020):
 - https://www.youtube.com/channel/UCqEdfW-1KUn_QQyQogxqLeA/
- P. Dalgaard, A Brief History of R and some Thoughts about the Future



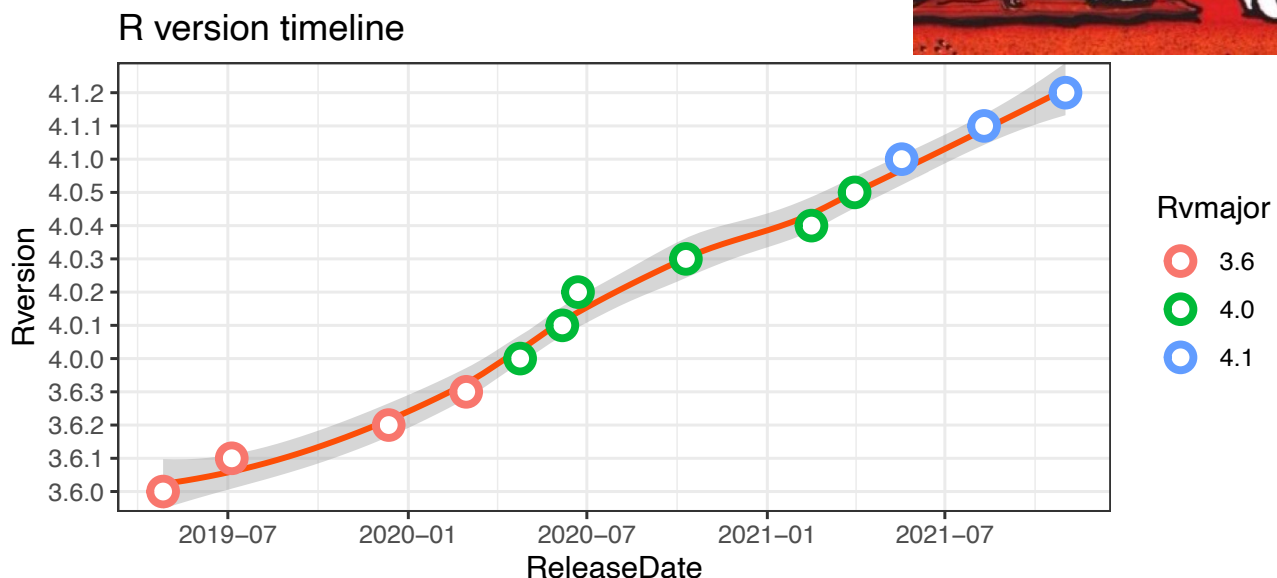
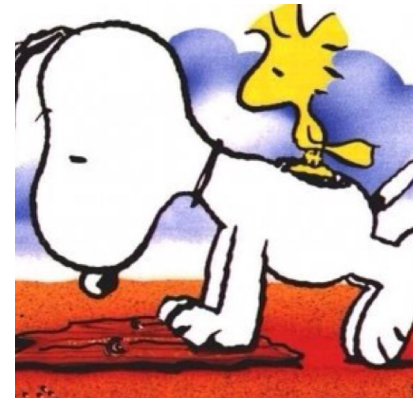
A. Garfagnini (UniPD)

AdvStat 4 PhysAna - R01

2

R versions and codenames

- R release names are taken from the Peanuts comics
- R 4.1.3 (One Push-Up)
- see Peter Dalgaard talk at useR!2018
<https://www.youtube.com/watch?v=z1vTSdRolgI>



A. Garfagnini (UniPD)

AdvStat 4 PhysAna - R01

3

CRAN: the Comprehensive R Archive Network

- Access point to R resources: [HOWTOs](#), [FAQ](#), [manuals](#), [examples](#), ...
- CRAN Web Page: <https://cran.r-project.org/>
- a list of [Frequently Asked Questions](#) is available
<https://cran.r-project.org/faqs.html>
- an [open access R journal](#) is published online once/twice per year:
<https://journal.r-project.org/>
- and several [Manuals](#) are available on CRAN Web Page:



The R Manuals			
<i>edited by the R Development Core Team.</i>			
The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms. The correct version of the manuals for each platform are part of the respective R installations. The manuals change with R, hence we provide versions for the most recent released R version (R-release), a very current version for the patched release version (R-patched) and finally a version for the forthcoming R version that is still in development (R-devel).			
Here they can be downloaded as PDF files, EPUB files, or directly browsed as HTML:			
Manual	R-release	R-patched	R-devel
An Introduction to R is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics.	HTML PDF EPUB	HTML PDF EPUB	HTML PDF EPUB
R Data Import/Export describes the import and export facilities available either in R itself or via packages which are available from CRAN.	HTML PDF EPUB	HTML PDF EPUB	HTML PDF EPUB
R Installation and Administration	HTML PDF EPUB	HTML PDF EPUB	HTML PDF EPUB
Writing R Extensions covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces.	HTML PDF EPUB	HTML PDF EPUB	HTML PDF EPUB
A draft of The R language definition documents the language <i>per se</i> . That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions.	HTML PDF EPUB	HTML PDF EPUB	HTML PDF EPUB
R Internals : a guide to the internal structures of R and coding standards for the core team working on R itself.	HTML PDF EPUB	HTML PDF EPUB	HTML PDF EPUB
The R Reference Index : contains all help files of the R standard and recommended packages in printable form. (9MB, approx. 3500 pages)	PDF	PDF	PDF

A. Garfagnini (UniPD)

AdvStat 4 PhysAna - R01

4

How to install R

Local Installation

- from sources: <https://cran.stat.unipd.it/src/base/R-4/R-4.1.2.tar.gz>
- or using pre-defined packages for
 - Linux (check with your favorite distribution)
 - mac OS X (from 10.13 above and for Legacy Os Systems)
 - Windows, (<https://cran.stat.unipd.it/bin/windows/base/>)

Anaconda distribution

- a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment
- it uses [Conda](#), an open source, cross-platform, language-agnostic package manager and environment management system.
- it is available for Linux, macOS and Windows: <https://www.anaconda.com>

Using Virtualization tools

- with Docker (<https://www.docker.com>), using predefined containers
 - `docker pull r-base` for R 4.1.2, alone https://hub.docker.com/_/r-base
 - `docker pull jupyter/r-notebook:latest`, for R 4.1.2 integrated with Jupyter <https://hub.docker.com/r/jupyter/r-notebook>

A. Garfagnini (UniPD)

AdvStat 4 PhysAna - R01

5

R console with Docker

- List images:

```
$ docker images -a
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
...
r-base               latest       3aad1ffccc53     7 days ago     679MB
```

- check computer IP and enable remote host display

```
$ open -a xquartz
$ ifconfig | grep 192
    inet 192.168.38.204 netmask 0xfffff00 broadcast 192.168.38.255
$ xhost + 192.168.38.204
192.168.38.204 being added to access control list

$ docker run -it --rm -v "$PWD":/mnt \
    -e DISPLAY=192.168.38.204:0 --name rintera r-base

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
...
> plot(1:10,1:10, col="pink")
> q()
Save workspace image? [y/n/c]: n
$
```

Jupyter notebook with Docker

- List images:

```
$ docker images -a
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
jupyter/r-notebook  latest       eca5843b30ea     2 days ago     3.37GB
...
```

- Start Docker container in 'detached' mode

```
$ docker run -d -P --rm --name nb_R01 \
    -v "$PWD":/home/jovyan/work jupyter/r-notebook
```

- Check running container. Extract HTTP port

```
$ docker ps -a
CONTAINER ID  IMAGE          [...] PORTS          NAMES
0a36c87d6cd6 jupyter/r-notebook [...] 0.0.0.0:32770->8888/tcp  nb_R01
```

- Inspect the container log file → extract the Jupyter token for Web login

```
$ docker logs --tail 3 nb_R01
Or copy and paste one of these URLs:
    http://0a36c87d6cd6:8888/?token=94ed...8b52
    or http://127.0.0.1:8888/?token=94ed...8b52
```

- Open the page in Browser. Once asked, insert the Jupyter token

```
http://127.0.0.1:32770
```

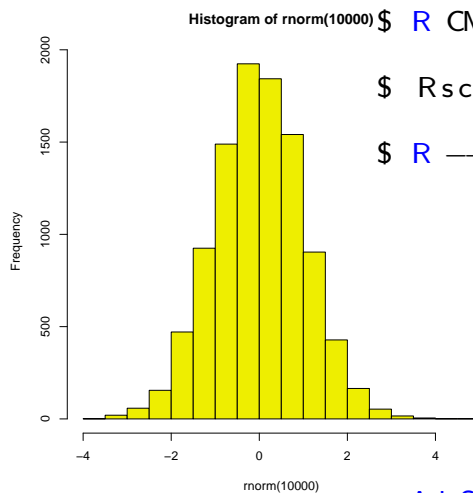
How to run R ?

Two running modes are available:

- **interactive** mode
- **batch** mode

Interactive mode R

```
$ R
> pdf("xh.pdf")
> hist(rnorm(1000),
      col="yellow")
> dev.off()
null device
      1
```



Batch mode R

```
file: xh_plot.R
pdf("xh.pdf")
hist(rnorm(1000), col="yellow")
dev.off()

$ R CMD BATCH xh_plot.R

$ Rscript xh_plot.R

$ R --no-save < xh_plot.R > xh_plot.out
```

← file: xh.pdf

Starting an interactive R session

- the R program can be invoked from the bash shell

```
$ R

R version 4.1.2 (2021-11-01) -- "Bird_Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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.

>
```

- exiting R can be done through the `q()` function or by typing `<CTRL>-d`

```
> q()
Save workspace image? [y/n/c]:
```

- at the end of an R session, the user can save an image of the current workspace that is automatically reloaded the next time R is started

Getting help in R

- the simplest way, if the [name of the function](#) we need help with is known, it to [prefix it with the question mark](#) symbol (?)

```
> ?plot
```

```
plot                                package:graphics                R Documentation
```

```
Generic X-Y Plotting
```

```
Description:
```

```
Generic function for plotting of R objects. For more details about the graphical parameter arguments, see 'par'.
```

```
For simple scatter plots, 'plot.default' will be used. However, there are 'plot' methods for many R objects, including 'function's, 'data.frame's, 'density' objects, etc. Use 'methods(plot)' and the documentation for these.
```

```
Usage:
```

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

Getting help in R

- if the [name of the function is not known](#), but only the subject on which help is needed, the `help.search()` function can be used

```
> help.search("data_input")
```

Help files with [alias](#) or [concept](#) or [title](#) matching 'data_input' using fuzzy matching:

```
utils::read.DIF          Data Input from Spreadsheet
utils::read.table        Data Input
```

Type `'?PKG::FOO'` to inspect entries `'PKG::FOO'`, or `'TYPE?PKG::FOO'` for entries like `'PKG::FOO-TYPE'`.

- or with the `find()` and `apropos()` functions

```
> find("read.table")
[1] "package:utils"
```

```
> apropos("lm")
[1] ".colMeans"      ".lm.fit"         "KalmanForecast"  "KalmanLike"
[5] "KalmanRun"      "KalmanSmooth"   "colMeans"        "confint.lm"
[9] "contr.helmert"  "dummy.coef.lm"  "getAllMethods"   "glm"
[13] "glm.control"   "glm.fit"        "kappa.lm"       "lm"
[17] "lm.fit"        "lm.influence"   "lm.wfit"        "model.matrix.lm"
[21] "nlm"          "nlminb"         "predict.glm"     "predict.lm"
[25] "residuals.glm" "residuals.lm"   "summary.glm"     "summary.lm"
```

R worked examples

- all R functions have a set of working examples that can be invoked and examined

```
> example(sqrt)
```

```
sqrt> require(stats) # for spline
```

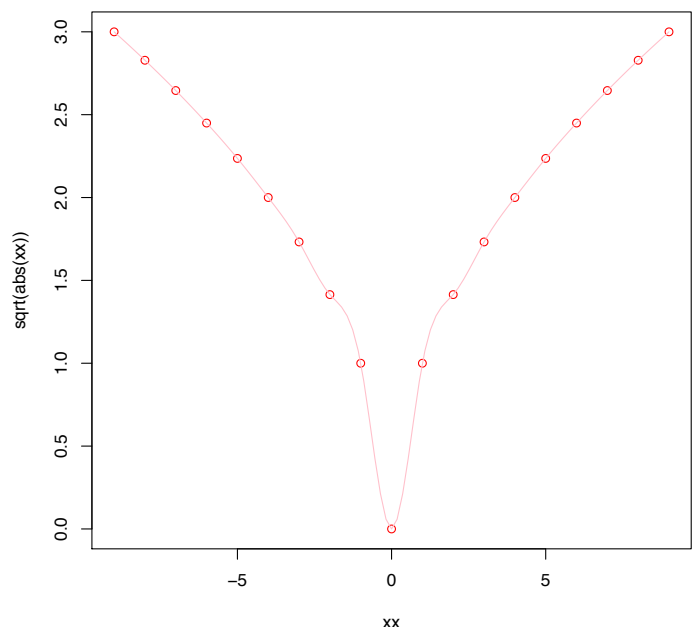
```
sqrt> require(graphics)
```

```
sqrt> xx <- -9:9
```

```
sqrt> plot(xx, sqrt(abs(xx)),
          col = "red")
```

Hit <Return> to see next plot:

```
sqrt> lines(spline(xx,
                  sqrt(abs(xx)),
                  n=101),
          col = "pink")
```



R packages - help

- `functions` and sets of `data` are organized in `packages`
- to find help and list the contents of a packages, the `library(help=package.name)` function will give details on the packages and a list of all the functions and data sets.

```
> library(help=base)
```

```

              Information on package 'base'
Description:

Package:      base
Version:      4.0.3
Priority:      base
Title:        The R Base Package
Author:        R Core Team and contributors worldwide
Maintainer:    R Core Team <R-core@r-project.org>
Description:   Base R functions.
License:       Part of R 4.0.3
Suggests:      methods
Built:         R 4.0.3; ; 2020-10-11 03:26:00 UTC; unix

Index:

.Call          Modern Interfaces to C/C++ code
...
zapsmall       Rounding of Numbers
```

R packages - listing

- with the command `installed.packages()` it is possible to retrieve a list of all the installed packages

```
> pkg <- installed.packages()
> df_pkg <- data.frame(pkg)
> names(df_pkg)
 [1] "Package"      "LibPath"      "Version"
 [4] "Priority"      "Depends"      "Imports"
 [7] "LinkingTo"    "Suggests"     "Enhances"
[10] "License"      "License_is_FOSS" "License_restricts_use"
[13] "OS_type"      "MD5sum"       "NeedsCompilation"
[16] "Built"

> length(df_pkg[,1])
[1] 299

> df_pkg[c(1:9,295:299),c(1,3,10,16)]
      Package      Version      License      Built
abind      abind      1.4-5      LGPL (>= 2) 4.0.2
askpass      askpass      1.1      MIT + file LICENSE 4.0.2
assertthat  assertthat  0.2.1      GPL-3      4.0.2
backports    backports    1.2.0      GPL-2 |    GPL-3      4.0.2
base64enc    base64enc    0.1-3      GPL-2 |    GPL-3      4.0.2
BayesFactor BayesFactor 0.9.12-4.2 GPL-2      4.0.2
bayestestR   bayestestR   0.8.0      GPL-3      4.0.2
bbmle        bbmle        1.0.23.1    GPL        4.0.2
bdsmatrix    bdsmatrix    1.3-4      LGPL-2     4.0.2
stats4       stats4       4.0.3      Part of R 4.0.3 4.0.3
survival     survival     3.2-7      LGPL (>= 2) 4.0.3
tcltk       tcltk       4.0.3      Part of R 4.0.3 4.0.3
tools       tools       4.0.3      Part of R 4.0.3 4.0.3
utils       utils       4.0.3      Part of R 4.0.3 4.0.3
```


R packages - installing

- a package can be installed from three main sources :

1. from [CRAN](#) (official stable versions)
2. from [GitHub](#) (developer versions)
3. from [other repositories](#), (for instance BioConductor)

Package: [data.table](#)

- on CRAN: <https://cran.r-project.org/web/packages/data.table/index.html>
- on GitHub: <https://github.com/Rdatatable/data.table>

README.md

data.table

CRAN OK build passing build passing codecov 99% pipeline passed downloads 523K/month
Depsy 100th percentile CRAN usage 806 BioC usage 201 indirect usage 2295



`data.table` provides a high-performance version of `base R`'s `data.frame` with syntax and feature enhancements for ease of use, convenience and programming speed.

30 January 2020

List-columns in `data.table` - Tyson Barrett, [rstudio::conf\(2020L\)](#)

R session housekeeping

- to list all the objects created with the current session, use the `ls()` or `objects()` functions

```
> objects()
[1] "Rdate"      "XXL"      "ctl"      "data"      "dc"      "diffs"
[7] "dl"        "duration" "group"    "hh"        "lm.D9"   "lm.D90"
[13] "model"     "ncol"     "op"       "opar"     "r"       "res"
[19] "st"        "t1"       "t2"       "test1"    "tf"      "times"
[25] "trt"       "weight"   "x"        "xx"       "y"       "y1"
[31] "y2"
```

- to list all the packages and data frames currently attached to the running R session, use `search()`

```
> search()
[1] ".GlobalEnv"      "package:lattice"  "times"
[4] "data"            "package:stats"    "package:graphics"
[7] "package:grDevices" "package:utils"    "package:datasets"
[10] "package:methods" "Autoloads"        "package:base"
```

- the screen prompt `>` invites to type commands and data
- the command line can be used as a calculator

```
> log(34/5.5)
[1] 1.821612
```

- each line can have up to 8192 characters, but can be continued on further lines if incomplete (the prompt will change from `>` to `+`)

```
> log(34.7) + sqrt(12) -
+ 25 / 7 * 46^3
[1] -347621.6
```

- two or more expressions can be placed on the same line, if are separated by `;`

```
> log(10); sqrt(3.75)*4.7; 2^2
[1] 2.302585
[1] 9.101511
[1] 4
```

R knows complex numbers

- complex numbers arithmetic's and elementary trigonometric, logarithmic, exponential, square root and hyperbolic functions are implemented
- a complex number has the imaginary part identified by a lower-case `'i'`

```
> 3.5 + 2i
[1] 3.5+2i
```

- special R functions can be used with complex numbers :

```
> Re(3.5 + 2i)
[1] 3.5
> Im(3.5 + 2i)
[1] 2
> Mod(3.5 + 2i)
[1] 4.031129
> Arg(3.5 + 2i)
[1] 0.5191461
> Conj(3.5 + 2i)
[1] 3.5-2i
> is.complex(3.5 + 2i)
[1] TRUE
> as.complex(3.5)
[1] 3.5+0i
```

Function	Description
<code>Re(z)</code>	Extract the real part
<code>Im(z)</code>	Extract the imaginary part
<code>Mod(z)</code>	Calculate the modulus
<code>Arg(z)</code>	Calculate the argument : $\text{Arg}(x+yi) = \text{atan}(y/x)$
<code>Conj(z)</code>	Work out the complex conjugate
<code>is.complex(z)</code>	test for complex number membership
<code>as.complex(z)</code>	force the input as a complex number

R mathematical functions

Function	Description
<code>log(x)</code>	base e log of x
<code>exp(x)</code>	anti-log of x
<code>log(x,n)</code>	base n log of x
<code>log10(x)</code>	base 10 log of x
<code>sqrt(x)</code>	square root of x
<code>factorial(x)</code>	$x! = x(x-1)(x-2)\dots 3 \cdot 2 \cdot 1$
<code>choose(n,x)</code>	binomial coefficient, $n!/(x! \cdot (n-x)!)$
<code>gamma(x)</code>	$\Gamma(x)$ for real x, $(x-1)!$ for integer x
<code>lgamma(x)</code>	natural log of $\Gamma(x)$
<code>abs(x)</code>	absolute value for x
<code>floor(x)</code>	greater integer less than x
<code>ceiling(x)</code>	smallest integer greater than x
<code>trunc(x)</code>	closest integer to x between 0 and x; it behaves as <code>floor()</code> for $x > 0$ and like <code>ceiling()</code> for $x < 0$

```
> floor(1.6); floor(-1.6)
[1] 1
[1] -2
> ceiling(1.6); ceiling(-1.6)
[1] 2
[1] -1
> trunc(1.6); trunc(-1.6)
[1] 1
[1] -1
```

R trigonometric functions

Function	Description
<code>cos(x)</code>	cosine of x in radians
<code>sin(x)</code>	sine of x in radians
<code>tan(x)</code>	tangent of x in radians
<code>asin(x)</code> , <code>acos(x)</code> , <code>atan(x)</code>	inverse trigonometric functions for real or complex numbers
<code>asinh(x)</code> , <code>acosh(x)</code> , <code>atanh(x)</code>	inverse hyperbolic trigonometric functions for real or complex numbers

- all trigonometric functions measure angle in radians. R knows the value of π as `pi`

```
> pi
[1] 3.141593
> sin(pi/2)
[1] 1
> cos(pi/2)
[1] 6.123234e-17
```

R variable names and assignments

- variable names are **case sensitive** : y different from Y
- variable names **must not begin with numbers** (4t) or symbols (%8)
- variable names **must not contain blank spaces** (use m.value instead of m value)
- object assignment is achieved using the '**<-**', **gets arrow** operator. Do not put spaces between them or a logical test will be performed (see below)

```
> x <- 5
> x
[1] 5
> x < - 5
[1] FALSE
```

- assignment can be achieved also with the '**->**', or '**=**' operators

```
> sqrt(x) + x^3 -> y
> y
[1] 127.2361
> z = x/y
> z
[1] 0.03929703
```

R arithmetic operators summary

+ - * /	sum, subtraction, multiplication, division
%% % ^	integer quotient, modulo, power
> >= < <= == !=	relational operators
! &	logical not, and, or
~	model formulae ('is modelled as a function of')
<- ->	assignment (gets)
\$	list indexing (the 'element name' operator)
:	sequence creation operator

```
> 119 %% 12 # integer part of the division
[1] 9
```

```
> 119 % 12 # reminder (modulo) of the division
[1] 11
```

```
> 15421 %% 7 == 0
[1] TRUE
```

- several of these operators have different meaning inside model formulae :
 - * indicates the main effects plus interaction (rather than multiplication),
 - : the interaction between two variables (rather than generate a sequence), and
 - ^ interactions up to the indicated power (rather than raise to the power)

R data types

- **everything** in R is an object
- the following data types are available:
- **atomic data types**: Vector (1-dim), Matrix (2-dim), Array (> 2 -dim)
- **Data Frame**: with homogeneous data type in each column
- **List**: a collection of simpler data types

