Introduction to R
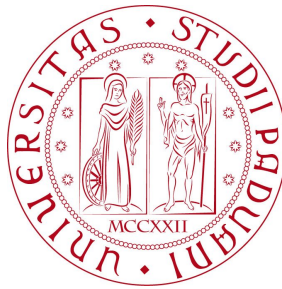
Alberto Garfagnini

Università di Padova

R lecture 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 3.6.3 (Holding the Windsock), released on February 29th, 2020 (20 years after R 1.0.0 release)

R Web Resources

- R Web Site: https://www.r-project.org/
- R source code: https://cran.r-project.org/src/base/R-3/
- 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 20 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)
- http://www.celebration2020.org/
- https://www.youtube.com/channel/UCqEdfW-1KUn_QQyQogxqLeA/

A Brief History of R and some Thoughts about Its Future

Peter Dalgaard

Center for Statistics
Copenhagen Business School

CelebRation 2020, Copenhagen, February 2020

Execution time by R version

https://jozef.io/r921-happy-birthday-r/

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 Journal

Volume 11/1, June 2019

A peer-reviewed, open-access publication of the
R Foundation for Statistical Computing

The R Manuals

edited by the R Development Core Team.

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 I PDF I EPUB	HTML I PDF I EPUB	HTML I PDF I 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 I PDF I EPUB	HTML I PDF I EPUB	HTML I PDF I EPUB
R Installation and Administration	HTML I PDF I EPUB	HTML I PDF I EPUB	HTML I PDF I EPUB
Writing R Extensions covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces.	HTML I PDF I EPUB	HTML I PDF I EPUB	HTML I PDF I EPUB
A draft of **The R language definition** documents the language *per se*. 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 I PDF I EPUB	HTML I PDF I EPUB	HTML I PDF I EPUB
R Internals: a guide to the internal structures of R and coding standards for the core team working on R itself.	HTML I PDF I EPUB	HTML I PDF I EPUB	HTML I PDF I 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

RWeekly.org

- It's a Web repository with lots of great blogs, tutorials and other formats of resources coming out every day

R Weekly 2020-10
golem, Celebration2020,
Patchwork

Highlight

Insights

R in the Real World

Resources

satRday Johannesburg

New Packages

Updated Packages

Videos and Podcasts

R Internationally

Tutorials

R Project Updates

Upcoming Events in 3 Months

Call for Participation

Quotes of the Week

- A SHINY QUIZ APP ABOUT THE RUSSIAN INFLUENCE CAMPAIGN BEFORE THE 2016 US ELECTIONS (skranz.github.io)
- Predicting the video game hype train - Playing around with Naïve Bayesian Learning (rcrastinate.rbind.io)
- COVID-19 epidemiology with R (rviews.rstudio.com)
- When does "garbage time" in an start in an NBA game? (jtcies.com)
- Modeling roll call voting behavior in the US House (jtimm.net)
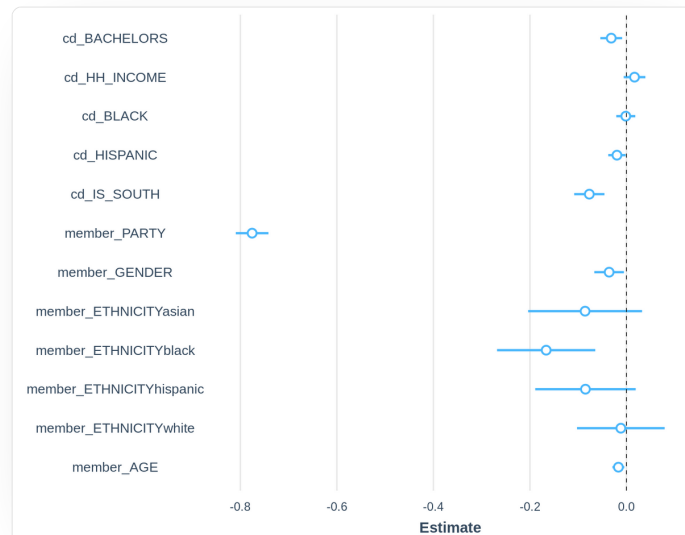
How to install R

Local Installation

- from sources: `https://cran.stat.unipd.it/src/base/R-3/R-3.6.3.tar.gz`
- or using pre-defined packages for
- Linux (check with your favorite distribution)
- mac OS X (for Catalina and 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 3.6.3, alone
- `docker pull Jupiter/r-notebook`, for R 3.6.1 integrated with Jupyter

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 0xffffff00 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
```
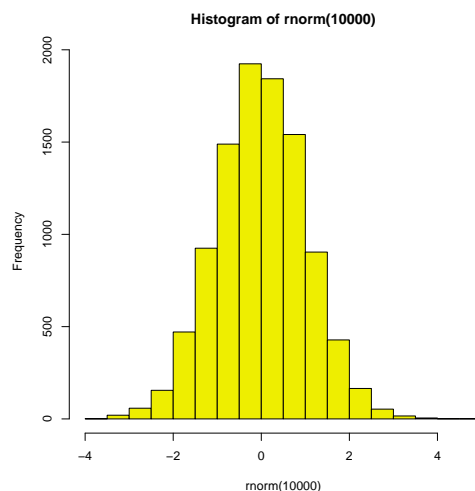
How to run R ?

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
```

Histogram of rnorm(10000)

Batch mode R

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

$ R CMD BATCH xh_plot.R
```

← **file: xh.pdf**

Starting an interactive R session

- the R program can be invoked from the bash shell

```
$ R

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)

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.

>
```

...and ending it

- 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
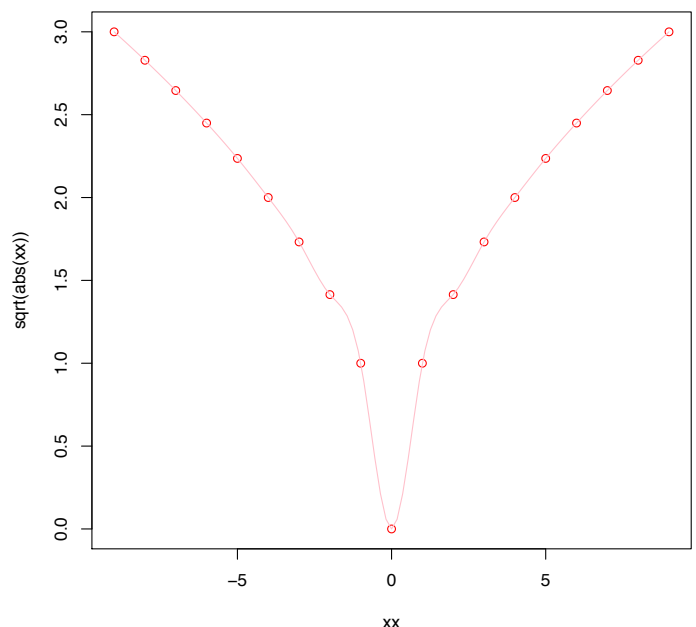
```
> example(sqrt)

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:        3.6.3
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 3.6.3
Suggests:       methods
Built:          R 3.6.3; ; 2020-02-29 10:11:03 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] 31

> df_pkg[c(1:9,25:31),c(1,3,5,10)]
             Package  Version                  Depends             License
docopt        docopt   0.6.1                     <NA> MIT + file LICENSE
littler      littler   0.3.9                     <NA>         GPL (>= 2)
base            base   3.6.3                     <NA>       Part of R 3.6.3
boot            boot  1.3-24 R (>= 3.0.0), graphics, stats         Unlimited
class          class  7.3-15       R (>= 3.0.0), stats, utils   GPL-2 | GPL-3
cluster      cluster   2.1.0                R (>= 3.3.0)         GPL (>= 2)
codetools  codetools  0.2-16                R (>= 2.1)                 GPL
compiler    compiler   3.6.3                     <NA>       Part of R 3.6.3
datasets    datasets   3.6.3                     <NA>       Part of R 3.6.3
splines      splines   3.6.3                     <NA>       Part of R 3.6.3
stats          stats   3.6.3                     <NA>       Part of R 3.6.3
stats4        stats4   3.6.3                     <NA>       Part of R 3.6.3
survival    survival   3.1-8                R (>= 3.4.0)       LGPL (>= 2)
tcltk          tcltk   3.6.3                     <NA>       Part of R 3.6.3
tools          tools   3.6.3                     <NA>       Part of R 3.6.3
utils          utils   3.6.3                     <NA>       Part of R 3.6.3
```

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

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"
```

R as a calculator

- 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
Re(z)	Extract the real part
Im(z)	Extract the imaginary part
Mod(z)	Calculate the modulus
Arg(z)	Calculate the argument : Arg(x+yi) = atan(y/x)
Conj(z)	Work out the complex conjugate
is.complex(z)	test for complex number membership
as.complex(z)	force the input as a complex number

R mathematical functions

Function	Description
log(x)	base e log of x
exp(x)	anti-log of x
log(x,n)	base n log of x
log10(x)	base 10 log of x
sqrt(x)	square root of x
factorial(x)	$x! = x(x-1)(x-2)\ldots 3 \cdot 2 \cdot 1$
choose(n,x)	binomial coefficient, $n!/(x! \cdot (n-x)!)$
gamma(x)	$\Gamma(x)$ for real x, $(x-1)!$ for integer x
lgamma(x)	natural log of $\Gamma(x)$
abs(x)	absolute value for x
floor(x)	greater integer less than x
ceiling(x)	smallest integer greater than x
trunc(x)	closest integer to x between 0 and x; it behaves as floor() for $x > 0$ and like ceiling() 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
cos(x)	cosine of x in radians
sin(x)	sine of x in radians
tan(x)	tangent of x in radians
asin(x), acos(x), atan(x)	inverse trigonometric functions for real or complex numbers
asinh(x), acosh(x), atanh(x)	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

Vector

Matrix

rows

columns

Array

Data Frame
(Table)

Lists