EU Twinning Project JO 21 ENI ST 01 22

Component 2:

Methodology for producing Small Area Statistics

Strengthening the capacity of Jordan's Department of Statistics in terms of compilation, analysis and reporting of statistical data in line with International and European best practices.

INTRODUCTION TO 

Tomas Rudys, Statistics Lithuania. State Data Agency

12-16 February 2023, Amman, Hashemite Kingdom of Jordan

Outline

- What is R (and CRAN)?
- Installing Base R (console, first script)
- R GUI (RStudio introduction)
- GSBPM and available R modules and libraries
- Introduction to R
- Getting data into R and outputting results
- Reading. Conferences
- Useful to know

Are You ready?

You ready?

What is R?

- **R** is a **free** language and environment for statistical computing and graphics.
- It is **object-oriented** programming language.
- **Designed by:** Ross Ihaka and Robert Gentleman.
- **First appeared:** August 1993; 28 years ago
- **Developers:** R Core Team.
- **Operating systems:** Windows, Linux, MacOS.
- R's capabilities are extended through user-created **packages**, which offer statistical techniques, graphical devices, import/export, reporting (RMarkdown, knitr), etc.
- Webpage: **https://www.r-project.org/**

CRAN

- **The Comprehensive R Archive Network (CRAN)** is R's central software repository, supported by the R Foundation.
- It contains an archive of the latest and previous versions of the R distribution, documentation, and contributed **R packages**.
- CRAN: **https://cran.r-project.org/**
- CRAN Mirrors: **https://cran.r-project.org/mirrors.html**

Installing Base R

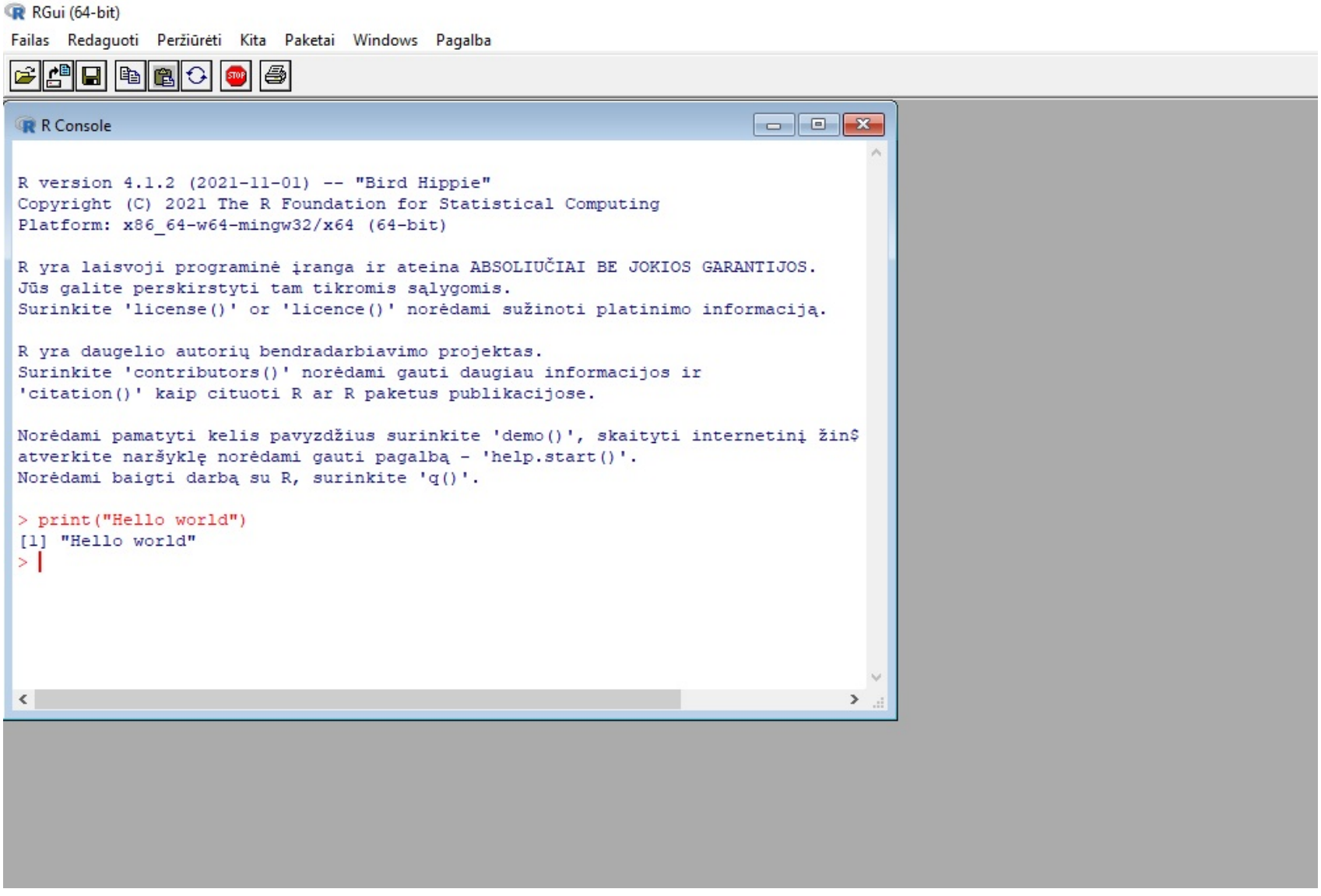
- Go to the R Project for Statistical Computing webpage: **https://www.r-project.org/**
- Find link to CRAN and choose your preferred **CRAN mirror**
- Press **Download R for Windows**
- Choose **base** for base R installation
- Choose **Download R 4.2.2 for Windows**

OR

- Go directly to CRAN webpage **https://cran.r-project.org/index.html**
- Press **Download R for Windows**
- Choose **base** for base R installation
- Choose **Download R 4.2.2 for Windows**

The R *.exe file for installation will be downloaded to your computer (eg. R-4.2.2-win.exe)

R console (first script example)



R graphical user interface (GUI)

- **RStudio**
- Rattle
- StatET for R
- RKWard
- JGR
- **R Commander** (package Rcmdr)
- Deducer
- JASP
- Tinn-R
- BlueSky Statistics

Different RStudio versions

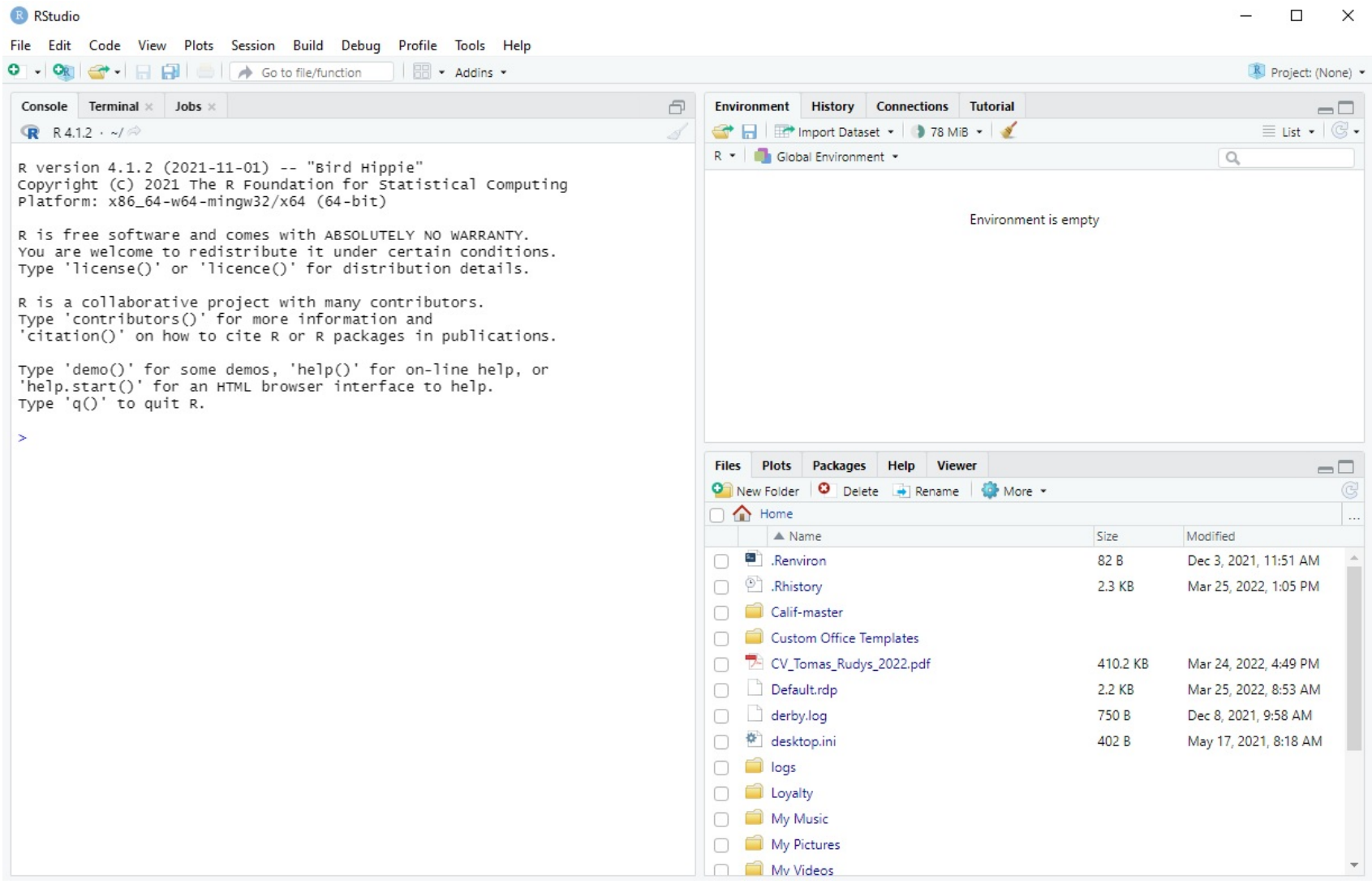
There are two versions of RStudio:

RStudio installation

- **RStudio**: **https://posit.co/**
- Products -> RStudio IDE -> Download RStudio
- Find RStudio Desktop
- Download RStudio
- Two steps:
  1. Download and install R (if you don‘t have it)
  2. Install Rstudio: press Download RStudio desktop for windows

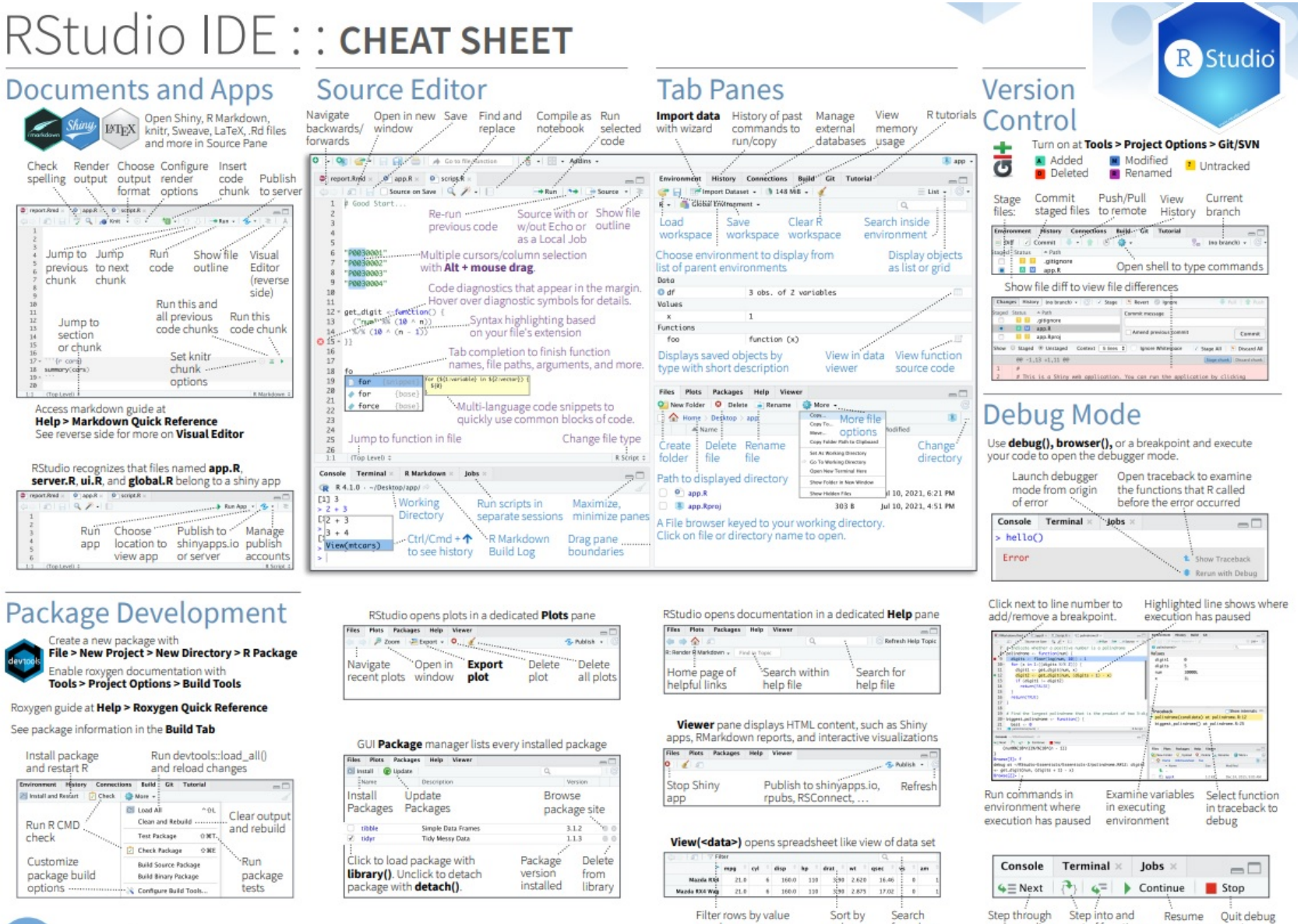
The RStudio *.exe file for installation will be downloaded to your computer (eg. RStudio-2022.12.0-353.exe)

RStudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Console Terminal Jobs
R 4.1.2 · ~/
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.
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.
>
Environment History Connections Tutorial
Import Dataset
78 MiB
List
R
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder
Delete
Rename
More
Home
Name
Size
Modified
.Renviron 82 B Dec 3, 2021, 11:51 AM
.Rhistory 2.3 KB Mar 25, 2022, 1:05 PM
Calif-master
Custom Office Templates
CV_Tomas_Rudys_2022.pdf 410.2 KB Mar 24, 2022, 4:49 PM
Default.rdp 2.2 KB Mar 25, 2022, 8:53 AM
derby.log 750 B Dec 8, 2021, 9:58 AM
desktop.ini 402 B May 17, 2021, 8:18 AM
logs
Loyalty
My Music
My Pictures
My Videos

RStudio Cheatsheets

https://posit.co/wp-content/uploads/2022/10/rstudio-ide-1.pdf

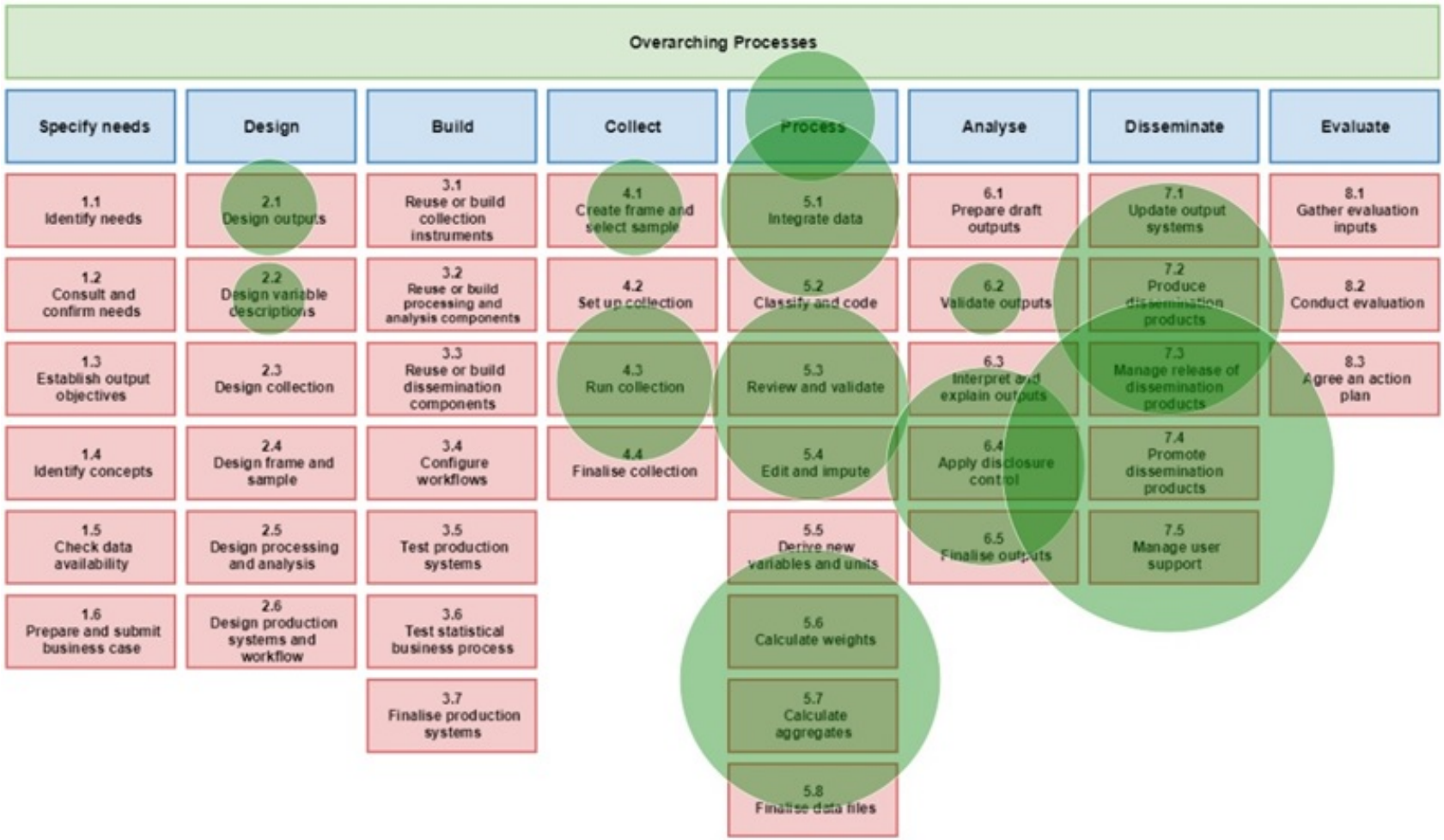


RStudio: an introduction

Live demo

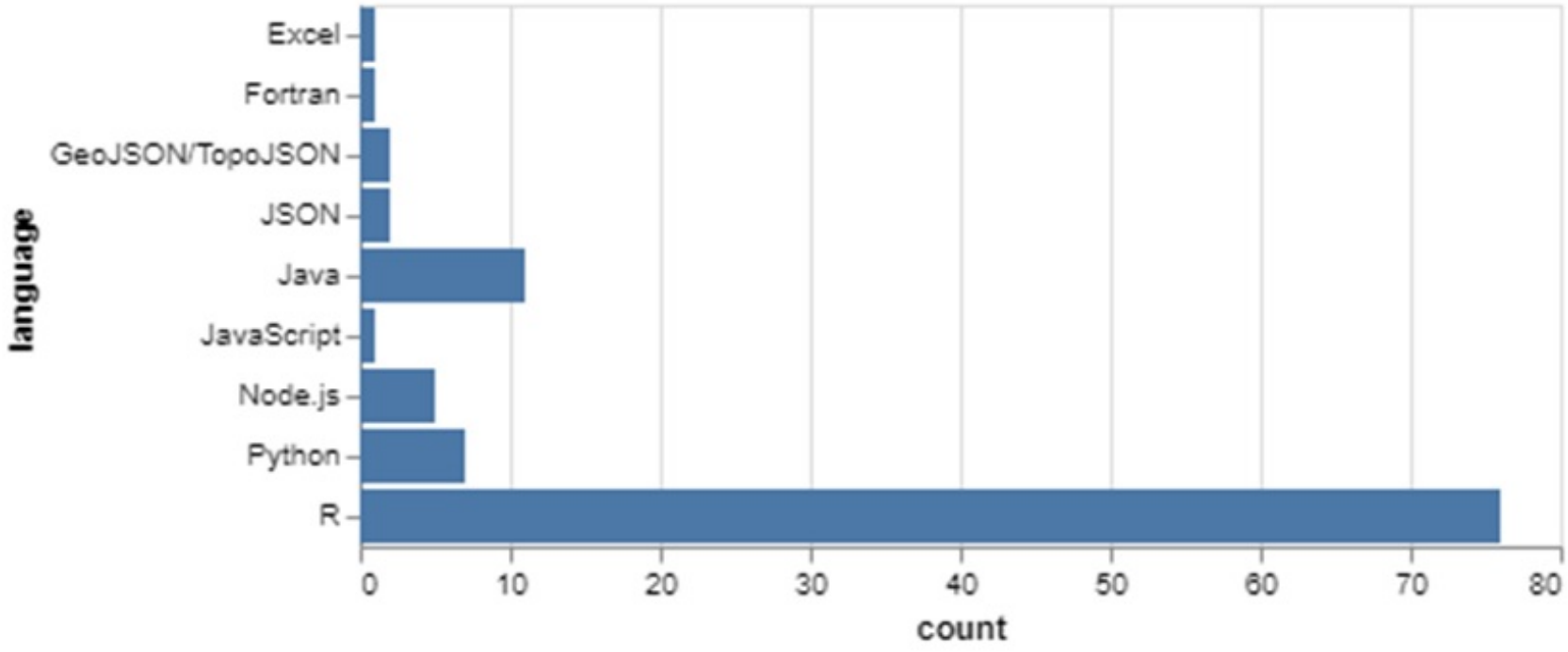
GSBPM and R

Awesome official statistics software (Olav ten Bosch):
https://github.com/SNStatComp/awesome-official-statistics-software



GSBPM and R

Not just R packages!!!



CRAN task views

- The **"Task Views"** page (subject list) on the CRAN website lists a wide range of tasks (in fields such as Finance, Genetics, High Performance Computing, Machine Learning, Medical Imaging, Oficial statistics) for which R packages are available.
- Link: **https://cran.r-project.org/web/views/**

CRAN Task View: Official Statistics & Survey Statistics

Maintainer:	Matthias Templ, Alexander Kowarik, Tobias Schoch
Contact:	matthias.templ at gmail.com
Version:	2022-11-17
URL:	https://CRAN.R-project.org/view=OfficialStatistics
Source:	https://github.com/cran-task-views/OfficialStatistics/
Contributions:	Suggestions and improvements for this task view are very welcome and can be made through issues o the maintainer address. For further details see the Contributing guide.
Citation:	Matthias Templ, Alexander Kowarik, Tobias Schoch (2022). CRAN Task View: Official Statistics & https://CRAN.R-project.org/view=OfficialStatistics.
Installation:	The packages from this task view can be installed automatically using the ctv package. For example, `coreOnly = TRUE)` installs all the core packages or `ctv::update.views("OfficialStatistics")` install up-to-date. See the CRAN Task View Initiative for more details.

This CRAN Task View contains a list of packages with methods typically used in official statistics and survey statistic than one of the topics listed below. Therefore, this list is not a strict categorization and packages may be listed more th

The task view is split into several parts

R packages

- **R packages** are extensions to the R statistical programming language. R packages contain code, data, and documentation in a standardised collection format that can be installed by users of R, typically via a centralised software repository such as CRAN.
- **https://cran.r-project.org/index.html** -> **Packages**
- Currently, the CRAN package repository features **19144** available packages.
- Available Packages: **https://cran.r-project.org/web/packages/available_packages_by_name.html**
- Every package has its own webpage: **https://cran.r-project.org/web/packages/sae/index.html**

R packages (instalation)

- Install R package with function **install.packages("name_of_the_package")** in your R script:

```
install.packages("sae")
```

- In RStudio use the tab **Packages** -> **Install**. The window for installation will appear.
- Or install from downloaded file.

R packages (loading)

- Once installed R packages have **to be loaded into the session** to be used.

```
library(xyz)
```

The **library()** by default returns an error if the requested package does not exist.

```
require(xyz)
```

The **require()** is designed to be used inside functions as it gives a warning message and returns a logical value say, FALSE if the requested package is not found and TRUE if the package is loaded.

R packages (loading example)

```
library(sae)
```

```
## Įkeliamas reikalingas paketas: MASS

## Įkeliamas reikalingas paketas: lme4

## Įkeliamas reikalingas paketas: Matrix
```

R packages (example of using package functions)

```
# Load data set with synthetic income data for provinces (
data(incomedata)
# Load population sizes of provinces
data(sizeprov)
# Compute Horvitz-Thompson direct estimator of mean income
# province under random sampling without replacement withi
result1 <- direct(y=income, dom=prov, sweight=weight,
domsize=sizeprov[,2:3], data=incomedata)
head(result1)
```

```
##   Domain SampSize    Direct        SD        CV
## 1      1       96  7121.005  978.9280 13.747049
## 2      2      173 13091.669 1380.7139 10.546508
## 3      3      539 13054.001  697.0416  5.339678
## 4      4      198 13158.890 1242.0200  9.438638
## 5      5       58 10592.494 1802.3255 17.015120
## 6      6      494 13213.102  759.5069  5.748134
```

Working with R packages

- The **pacman** package provides a function to automatically install a package if it is not available locally. In addition to CRAN it also tries to install from Bioconductor.

```
#install.packages("pacman")
library(pacman)
p_load("survey", "sampling", "ggplot3")
```

```
## Warning: package 'ggplot3' is not available for this ver
##
## A version of this package for your version of R might be
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin

## Warning: nepavyko pasiekti saugyklos http://www.stats.ox
##   nepavyko atverti URL 'http://www.stats.ox.ac.uk/pub/RW

## Warning: 'BiocManager' not available.  Could not check B
##
## Please use `install.packages('BiocManager')` and then re
```

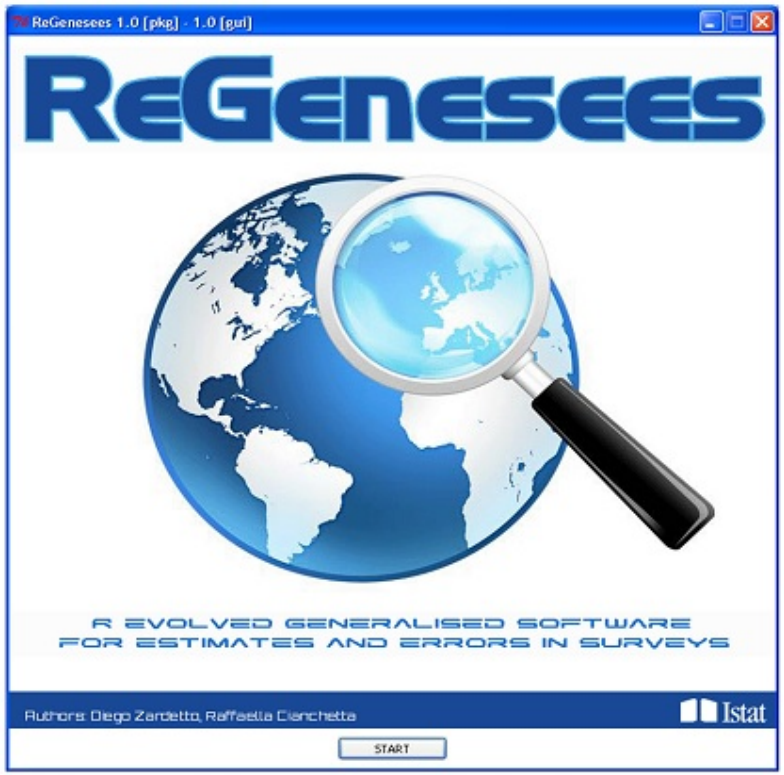
R tools outside CRAN

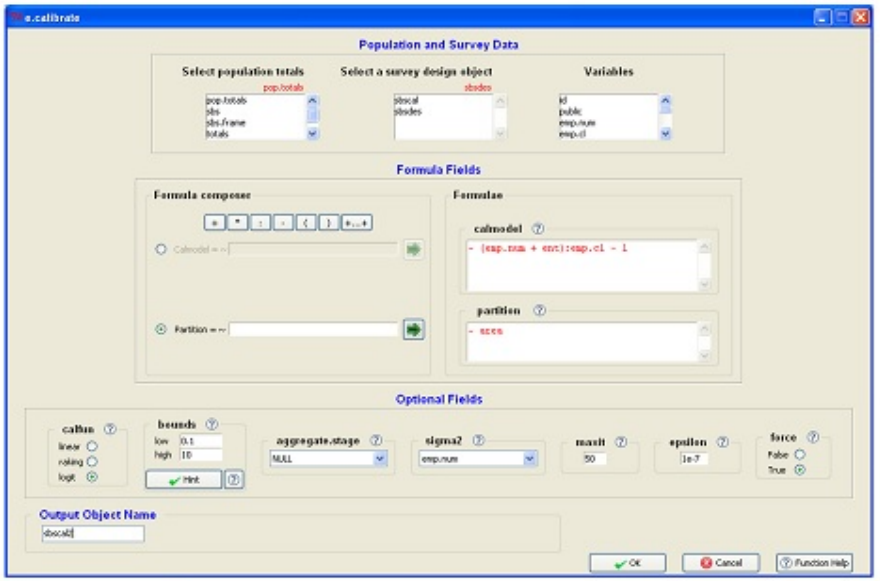
REGENESEES (R EVOLVED GENERALISED SOFTWARE FOR SAMPLING ESTIMATES AND ERRORS IN SURVEYS) BY ISTAT (Diego Zardetto).

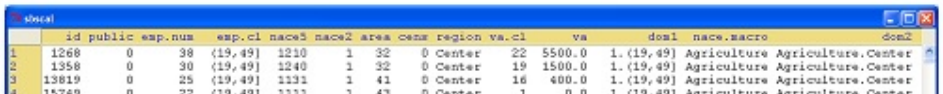
- **https://www.istat.it/en/methods-and-tools/methods-and-it-tools/process/processing-tools/regenesees**
- webpage: **https://diegozardetto.github.io/ReGenesees/**
- Main Statistical Functions:
  - Complex Sampling Designs
  - Calibration
  - Basic Estimators
  - Variance Estimation
  - Estimates and Sampling Errors (standard error, variance, coefficient of variation, confidence interval, design effect)
  - Estimates and Sampling Errors for Complex Estimators

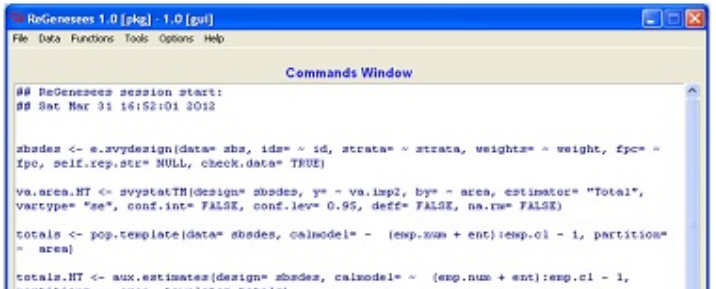
R tools outside CRAN

ReGenesees.GUI provides a Graphical User Interface for the ReGenesees package, based on tcltk. - It has been developed for users who might prefer to interact with ReGenesees through a user-friendly mouse-click graphical interface (rather than through R's command line).







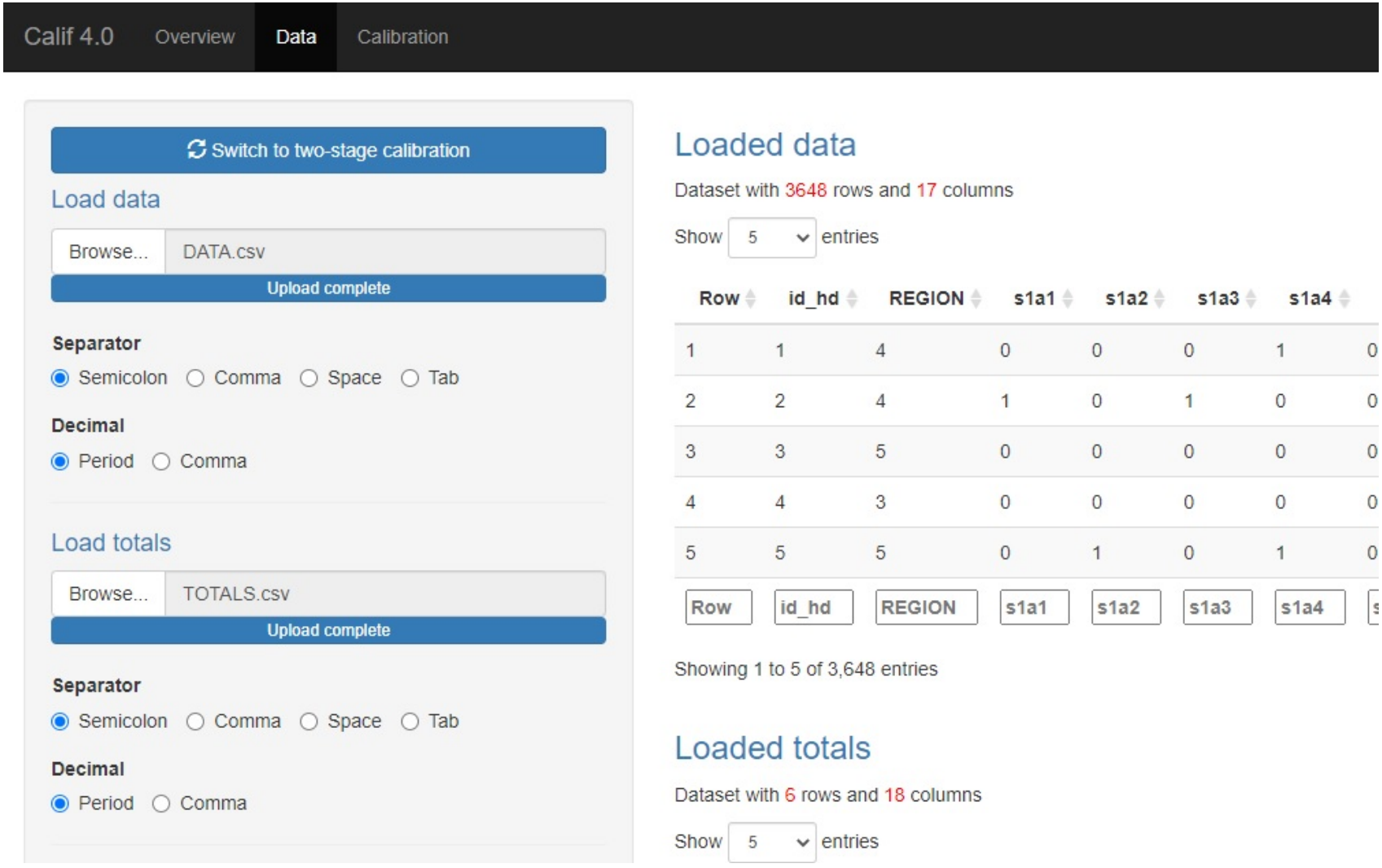


R tools outside CRAN

Calif - Calibration of weights of statistical surveys.

- It is a Shiny web app for calibration of weights of statistical surveys prepared by Statistical Office of the Slovak Republic.
- **https://tinyurl.com/ywxy78rv**
- GitHub Repository: **https://github.com/SO-SR/Calif**

R tools outside CRAN



Introduction to R

"An Introduction to R" by W. N. Venables, D. M. Smith and the R Core Team.

- Getting help with functions and features

```
help(mean)
```

An alternative is

```
?mean
```

In RStudio use Help Tab.

Introduction to R

- R commands **are case sensitive**

```
a = 5
A = 2
a + A
```

```
## [1] 7
```

- Comments

```
# This is comment
```

- Commands are separated either by a semi-colon (';'), or by a newline.

```
a = 5; A = 2
a; A
```

```
## [1] 5
```

```
## [1] 2
```

Introduction to R

- The entities that R creates and manipulates are known as **objects**.

These may be variables, arrays of numbers, character strings, functions, or more general structures built from such components.

```
a = 5; A = 2
objects()
```

```
## [1] "a"           "A"           "incomedata" "result1"
```

```
ls()
```

```
## [1] "a"           "A"           "incomedata" "result1"
```

Introduction to R

- To remove objects the function **rm** is available:

```
a = 5; A = 2
ls()
```

```
## [1] "a"           "A"           "incomedata" "result1"
```

```
rm(a)
ls()
```

```
## [1] "A"           "incomedata" "result1"    "sizeprov"
```

- remove all objects:

```
rm(list = ls())
ls()
```

```
## character(0)
```

Introduction to R

- If commands are stored in an external file, say **script_1.R** and **script_2.R**:

```
source("C:\\Users\\TomasR\\Desktop\\Jordan\\R_intro\\script
```

```
## [1] 5
## [1] 2
```

```
source("C:\\Users\\TomasR\\Desktop\\Jordan\\R_intro\\script
```

```
## [1] 7
```

Introduction to R

- Set working directory setwd()
- Get working directory getwd()

```
setwd("C:\\Users\\TomasR\\Desktop\\Jordan\\R_intro")
getwd()
```

```
## [1] "C:/Users/TomasR/Desktop/Jordan/R_intro"
```

```
source("script_1.R")
```

```
## [1] 5
## [1] 2
```

```
source("script_2.R")
```

```
## [1] 7
```

- **Recommendation:** in RStudio good practice to work by creating R projects

Introduction to R

- R operates on named data structures. The simplest such structure is the numeric vector.
- Assignment operators: <- usualy = works also
- Assignment: use function **c()** or **assign()**

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

```
## [1] 1 2 3 4 5
```

```
c(1, 2, 3, 4, 5) -> x
x
```

```
## [1] 1 2 3 4 5
```

Introduction to R

```
assign("x", c(1, 2, 3, 4, 5))
x
```

```
## [1] 1 2 3 4 5
```

```
y = c(x, 0, x)
y
```

```
##  [1] 1 2 3 4 5 0 1 2 3 4 5
```

```
# arithmetic: +, -, *, / and ^
v = 2*x + 1
v
```

```
## [1]  3  5  7  9 11
```

```
v1 = x + v
v1
```

```
## [1]  4  7 10 13 16
```

Introduction to R

```
x = c(1, 2, 3, 4, 5)
min(x)
```

```
## [1] 1
```

```
max(x)
```

```
## [1] 5
```

```
length(x)
```

```
## [1] 5
```

```
sum(x)
```

```
## [1] 15
```

Introduction to R

```
x = c(1, 2, 3, 4, 5)
# sum(x)/length(x)
mean(x)
```

```
## [1] 3
```

```
# sum((x-mean(x))^2)/(length(x)-1)
var(x)
```

```
## [1] 2.5
```

```
x = c(2, 1, 3, 5, 4)
sort(x)
```

```
## [1] 1 2 3 4 5
```

Introduction to R

- Generating regular sequences

```
# seq(1,10), seq(from=1, to=10) and seq(to=10, from=1)
seq(1:10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
seq(-2, 2, by=.5)
```

```
## [1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0
```

```
seq(length=10, from=-2, by=.5)
```

```
##  [1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0  2.5
```

Introduction to R

- Generating regular sequences

```
x = c(2, 3)
rep(x, times=3)
```

```
## [1] 2 3 2 3 2 3
```

```
rep(x, each=3)
```

```
## [1] 2 2 2 3 3 3
```

Introduction to R

- Logical vectors are generated by conditions:

```
x = c(5, 6, 7, 2, 3, 7)
log_v = x > 5
log_v
```

```
## [1] FALSE  TRUE  TRUE FALSE FALSE  TRUE
```

The logical operators are $<$, $<=$, $>$, $>=$, $==$ for exact equality and $!=$ for inequality.

Introduction to R

If c1 and c2 are logical expressions:

- c1 & c2 is their intersection ("and")
- c1 | c2 is their union ("or")
- !c1 is the negation of c1

```
x = c(5, 6, 7, 2, 3, 7)
c1 = x == 7
c2 = x <= 2
log_v = c1 | c2
log_v
```

```
## [1] FALSE FALSE  TRUE  TRUE FALSE  TRUE
```

Introduction to R

- Missing values.
- **NA** - "not available" or a "missing value"

```
z = c(1:3, NA, 4, NA)
z
```

```
## [1]  1  2  3 NA  4 NA
```

```
ind = is.na(z)
ind
```

```
## [1] FALSE FALSE FALSE  TRUE FALSE  TRUE
```

- **NaN** - Not a Number

```
0/0
```

```
## [1] NaN
```

```
Inf - Inf
```

```
## [1] NaN
```

Introduction to R

- Missing values.
- **NaN** - Not a Number

```
z = c(1, 2, NA, 0/0, 0/0, Inf - Inf)
z
```

```
## [1]   1   2  NA NaN NaN NaN
```

```
is.na(z)
```

```
## [1] FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

```
is.nan(z)
```

```
## [1] FALSE FALSE FALSE  TRUE  TRUE  TRUE
```

Introduction to R

- Character vectors

```
v = c("a", "b", "c")
v
```

```
## [1] "a" "b" "c"
```

```
labs = paste(c("X"), 1:5, sep="")
labs
```

```
## [1] "X1" "X2" "X3" "X4" "X5"
```

Introduction to R

- Index vectors

```
x = c(1:3, NA, 4, NA)
x
```

```
## [1]  1  2  3 NA  4 NA
```

```
x[2]
```

```
## [1] 2
```

```
y = x[!is.na(x)]
y
```

```
## [1] 1 2 3 4
```

```
x[is.na(x)] <- 0
x
```

```
## [1] 1 2 3 0 4 0
```

Introduction to R

Other types of objects:

- **matrices** or more generally **arrays** are multi-dimensional generalizations of vectors
- **factors** provide compact ways to handle categorical data
- **lists** are a general form of vector in which the various elements need not be of the same type, and are often themselves vectors or lists
- **data frames** are matrix-like structures, in which the columns can be of different types
- **functions** are themselves objects in R which can be stored in the project's workspace

Introduction to R

R objects modes: numeric, complex, logical, character and raw

```
z = c(0:9);z
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

```
mode(z)
```

```
## [1] "numeric"
```

```
digits = as.character(z); digits
```

```
##  [1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

```
mode(digits)
```

```
## [1] "character"
```

```
d = as.integer(digits)
mode(d)
```

```
## [1] "numeric"
```

Introduction to R

A factor is a vector object used to specify a discrete classification (grouping) of the components.

```
nace = c("4929","8610","8610","5010","7512","5012","5010")
nace_fac = factor(nace)
nace_fac
```

```
## [1] 4929 8610 8610 5010 7512 5012 5010
## Levels: 4929 5010 5012 7512 8610
```

```
levels(nace_fac)
```

```
## [1] "4929" "5010" "5012" "7512" "8610"
```

Introduction to R

```
x <- array(1:20, dim=c(4,5)); x
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    5    9   13   17
## [2,]    2    6   10   14   18
## [3,]    3    7   11   15   19
## [4,]    4    8   12   16   20
```

```
x <- array(1:20, dim=c(3,3)); x
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
x[2,3]
```

```
## [1] 8
```

Introduction to R

```
m = matrix(1:9, nrow = 3, ncol = 3); m
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
m1 = matrix(1:9, nrow=3, byrow=TRUE); m1
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
```

Introduction to R

Elements can be accessed as **var[row, column]**

```
m[1,]
```

```
## [1] 1 4 7
```

```
m[,1]
```

```
## [1] 1 2 3
```

```
# select rows 1 & 2 and columns 2 & 3
m[c(1,2),c(2,3)]
```

```
##      [,1] [,2]
## [1,]    4    7
## [2,]    5    8
```

Introduction to R

- List

```
x = c(5, 6, 7, 2, 3, 7)
v = c("a", "b", "c")
m = matrix(1:4, nrow = 2, ncol = 2)
l <- list("id" = x, "nace" = v, "m" = m)
l
```

```
## $id
## [1] 5 6 7 2 3 7
##
## $nace
## [1] "a" "b" "c"
##
## $m
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```
l[[2]]
```

```
## [1] "a" "b" "c"
```

```
l[[1]][2]
```

```
## [1] 6
```

```
l[[3]][,1]
```

```
## [1] 1 2
```

Introduction to R

A data frame is a list with class "data.frame".

```
id = c(1, 2, 3)
nace = c("a", "b", "c")
emp = c(10, 5, 8)
df = data.frame(ent_ID=id, NACE=nace, num_emp=emp)
df
```

```
##   ent_ID NACE num_emp
## 1      1    a      10
## 2      2    b       5
## 3      3    c       8
```

Introduction to R

```
df$num_emp
```

```
## [1] 10  5  8
```

```
attach(df)
num_emp
```

```
## [1] 10  5  8
```

```
detach(df)
```

Introduction to R

Conditional execution: if statements

```
if (test_expression) {
  statement
}
```

```
x <- 5
if(x > 0){
print("Positive number")
}
```

```
## [1] "Positive number"
```

Introduction to R

Conditional execution: if. . . else statements

```
if (test_expression) {
statement1
} else {
statement2
}
```

```
x <- -5
if(x > 0){
print("Non-negative number")
} else {
print("Negative number")
}
```

```
## [1] "Negative number"
```

Introduction to R

Repetitive execution: for loops, repeat and while

```
for(i in 1:5) {
  x1 <- i ^ 2
  print(x1)
}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
```

Introduction to R

```
x <- 1
repeat{
  print(x)
  x = x+1
    if (x == 5){
    break
                }
      }
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

Introduction to R

```
i <- 1
while (i < 6) {
  print(i)
    i = i+1
                }
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

Introduction to R

- Writing your own functions. Syntax:

```
func_name <- function (argument) {
statement
}
```

```
# function to print x raised to the power y

pow <- function(x, y) {
  result <- x^y
  print(paste(x,"raised to the power", y, "is", result))
}

pow(x = 2, y = 2)
```

```
## [1] "2 raised to the power 2 is 4"
```

Introduction to R

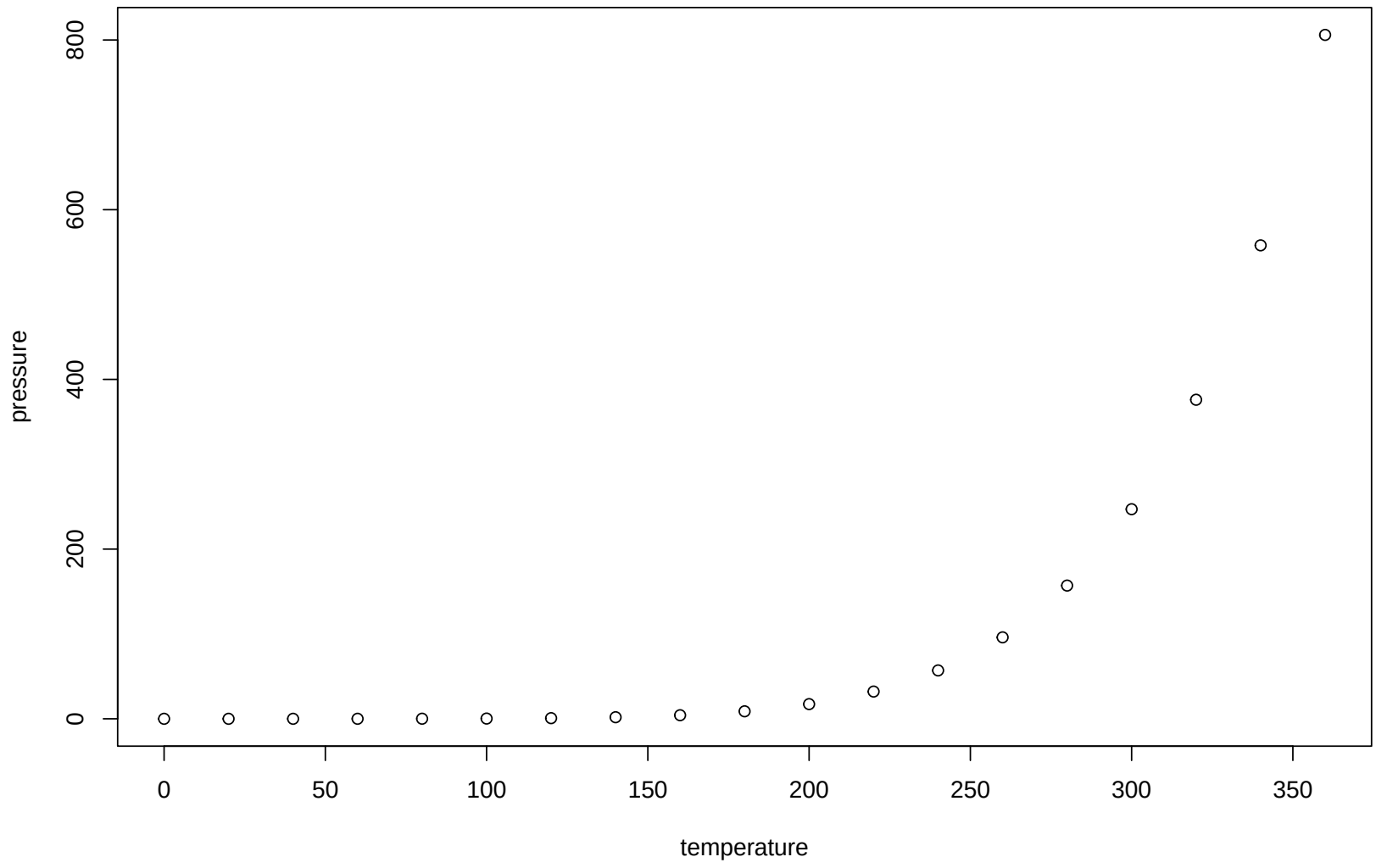
Plot data: plot() function

```
pres_data = pressure
head(pres_data)
```

```
##   temperature pressure
## 1           0   0.0002
## 2          20   0.0012
## 3          40   0.0060
## 4          60   0.0300
## 5          80   0.0900
## 6         100   0.2700
```

Introduction to R

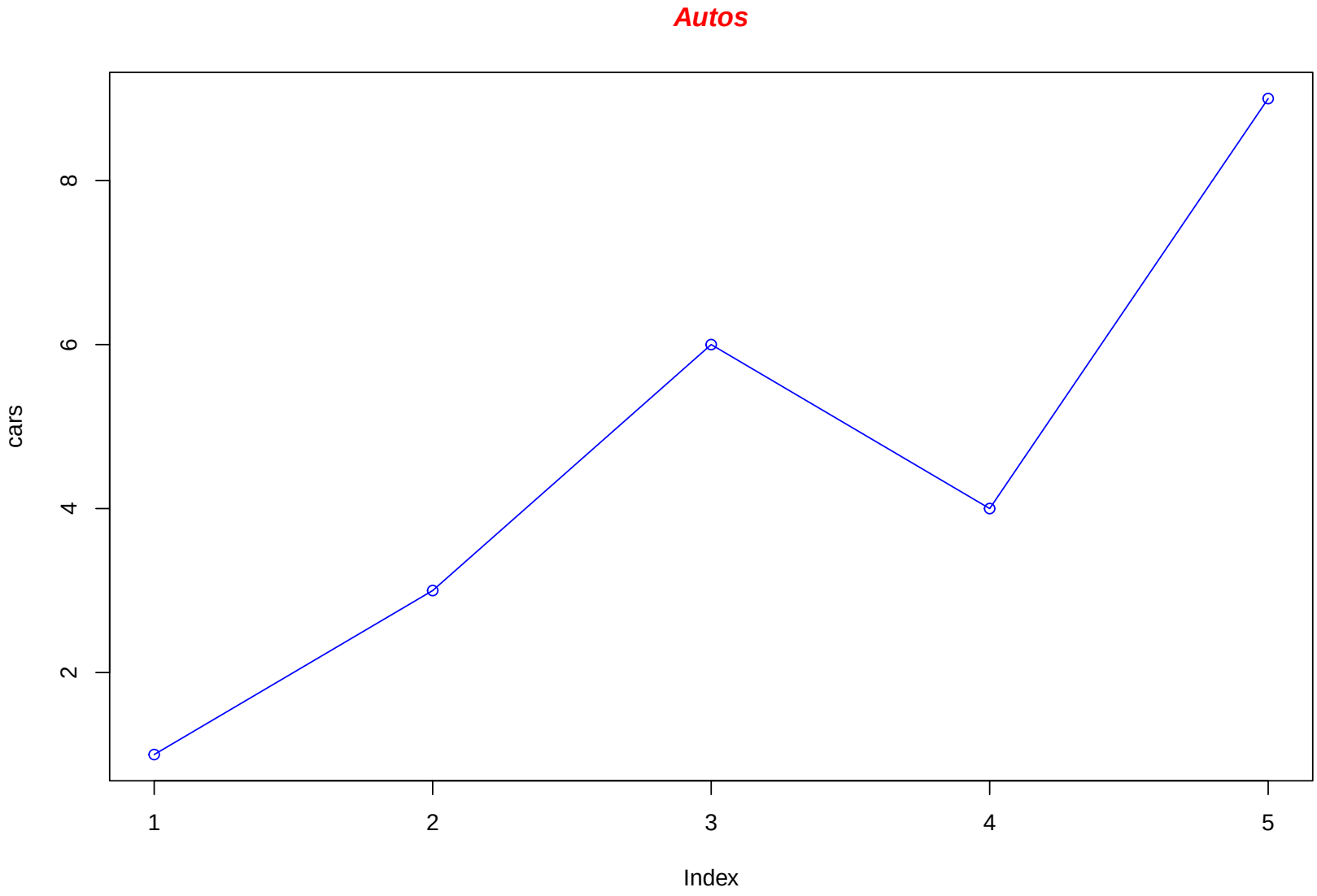
```
plot(pres_data)
```



Introduction to R

```r
# Define the cars vector with 5 values
cars <- c(1, 3, 6, 4, 9)
# Graph cars using blue points overlayed by a line
plot(cars, type="o", col="blue")
# Create a title with a red, bold/italic font
title(main="Autos", col.main="red", font.main=4)
```

Introduction to R



R Data Import

To read an entire data frame directly, the external file will normally have a special form.

```
HousePrice <- read.table("data.txt")
HousePrice
```

```
##       V1    V2   V3    V4  V5         V6
## 1 Price Floor Area Rooms Age Cent.heat
## 2 52.00 111.0  830     5 6.2         no
## 3 54.75 128.0  710     5 7.5         no
## 4 57.50 101.0 1000     5 4.2         no
```

```
HousePrice <- read.table("data.txt", header=TRUE)
HousePrice
```

```
##   Price Floor Area Rooms Age Cent.heat
## 1 52.00   111  830     5 6.2         no
## 2 54.75   128  710     5 7.5         no
## 3 57.50   101 1000     5 4.2         no
```

R Data Import

```
new_df <- data.frame(HousePrice,
           price_one = HousePrice$Price / HousePrice$Rooms)
new_df
```

```
##    Price Floor Area Rooms Age Cent.heat price_one
## 1 52.00    111  830     5 6.2         no     10.40
## 2 54.75    128  710     5 7.5         no     10.95
## 3 57.50    101 1000     5 4.2         no     11.50
```

```
write.table(new_df, "new_df.txt")
```

R Data Import

```
# csv files
df <- read.csv('file.csv')
write.csv(df, 'file.csv')

#Rdata files
save(df, file = 'file.Rdata')
load('file.RData')
```

R Data Import

- Import Spreadsheets

```
read_excel(path, sheet = NULL)
# Specify which sheet to read by position or name.
```

```
excel_sheets(path)
# Get a vector of sheet names.
```

- "R Data Import/Export" by R Core Team
  https://cran.r-project.org/doc/manuals/r-release/R-data.pdf
- possible to import different data formats and types

R Data Import (example)

```
library(readxl)
excel_sheets("data.xlsx")
```

```
## [1] "duomenys" "Sheet1"   "Sheet2"   "Sheet3"   "Sheet4"
```

```
data = read_excel("data.xlsx", sheet = "data")
head(data)
```

```
## # A tibble: 6 x 6
##      id nace         turn_q  hours      vat_q   emp
##   <dbl> <chr>         <dbl>  <dbl>      <dbl> <dbl>
## 1     1 grupe_32 245748507 475216 253114282 1024.
## 2     2 grupe_44  99654980 917213 104517461 2028.
## 3     4 grupe_9   86238221 944252  87607361 2216.
## 4     5 grupe_43  67514000  70962  68144446  151.
## 5     6 grupe_43  56587531 272702  58011296  607.
## 6     7 grupe_5   56240816 808947  55653567 2756.
```

Reading. Conferences

- **R bloggers:** https://www.r-bloggers.com/
- **The R Journal:** https://journal.r-project.org/
- **useR! — International R User Conference:** https://www.r-project.org/conferences/
- **The R Project - The Use of R in Official Statistics:** https://r-project.ro/conferences.html#uRos_Conferences; https://github.com/uRosConf
- **RStudio** -> Tab **Tutorial**

Useful to know

- **Graphs with R**: **https://r-graph-gallery.com/index.html**
- The **Tidyverse** is an opinionated collection of R packages designed for data science: **https://www.tidyverse.org/**
- Fast aggregation of large data package **data.table**: **https://rdatatable.gitlab.io/data.table/**
- **Shiny** is an R package that makes it easy to build interactive web apps straight from R: **https://shiny.rstudio.com/**
- **R Markdown** for interactive documents: **https://rmarkdown.rstudio.com/**
- R interface to Apache Spark (**sparklyr**): **https://spark.rstudio.com/**
- **Databases with R: https://cran.r-project.org/web/views/Databases.html**
- **RStudio Cloud: https://posit.cloud/**

Use R

keep-calm.net

?