

## Biostatistics & Epidemiological Data Analysis using R

# 1

## Introduction to R & RStudio

Stefan Konigorski

Health Intervention Analytics Group, HPI

October 28, 2021

# Learning objectives

## Introduction to R, RStudio:

- Overview, what R as programming language can be used for.
- Get an overview how the graphical interface RStudio looks like and what it can be used for.
- Understand what kind of "R files" there are (.RData, .R, .Rmd).
- Be able to access the help pages for an R function and R package, and understand what these two mean.

- 1 Background
  - Empirical research process
  - R & RStudio
- 2 Introduction to R & RStudio
  - Overview of RStudio
  - Data frames
- 3 Documentation with R
  - R Scripts
  - R Markdown

## But before we get started ...

... is everybody's installation of R and RStudio running?

## But before we get started ...

... is everybody's installation of R and RStudio running?

Who is working on Windows? Mac? Linux? Others?

## But before we get started ...

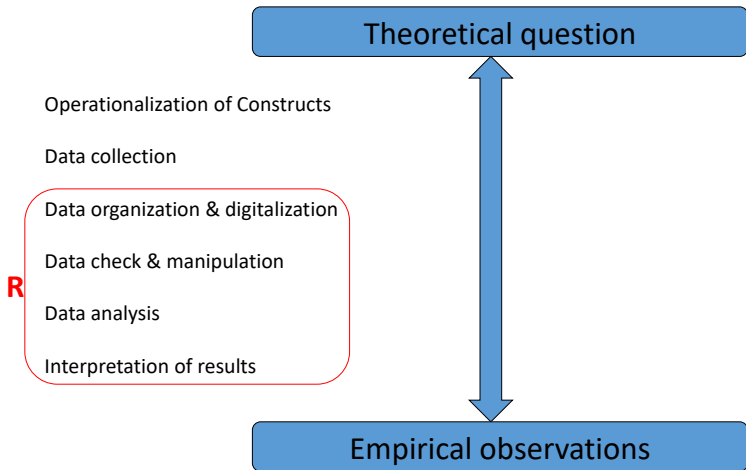
... is everybody's installation of R and RStudio running?

Who is working on Windows? Mac? Linux? Others?

Who has a second screen?

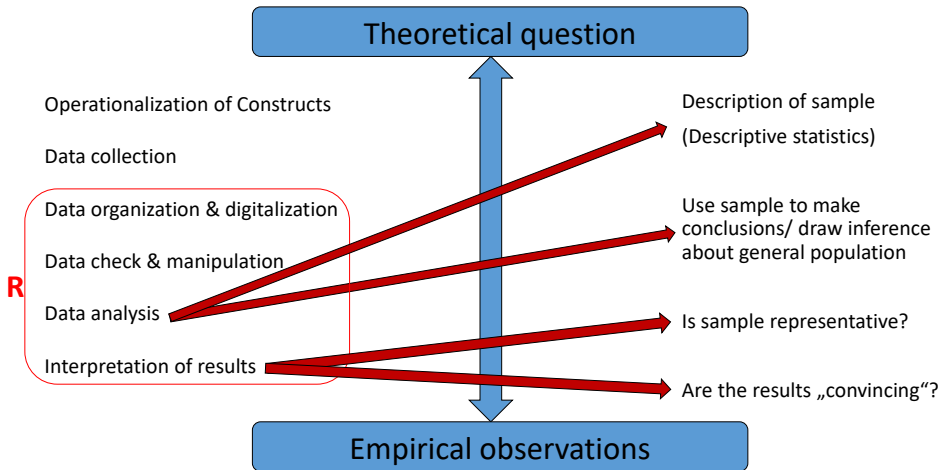
## Background

# Empirical research process - overview

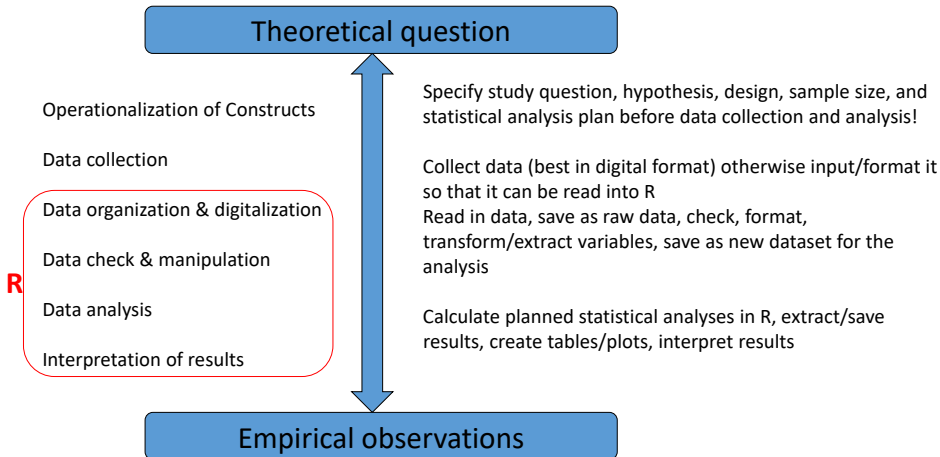




# Empirical research process - analysis & interpretation



# Empirical research process - steps in/around R



# Empirical research process - example

*International Journal of Epidemiology*, 2019, 148–156 doi: 10.1093/ije/dyy118

## **Are birthweight and postnatal weight gain in childhood associated with blood pressure in early adolescence? Results from a Ugandan birth cohort**

**Background:** In Africa, where low birthweight (LBW), malnutrition and high blood pressure (BP) are prevalent, the relationships between birthweight (BW), weight gain and BP later in life remain uncertain. We examined the effects of early life growth on BP among Ugandan adolescents.

**Methods:** Data were collected prenatally from women and their offspring were followed from birth, with BP measured following standard protocols in early adolescence. Weight-for-age Z-scores (WAZ) were computed using World Health Organization references. Linear regression was used to relate BW, and changes in WAZ between birth and 5 years, to adolescents' BP, adjusting for confounders.

**Results:** Among 2345 live offspring, BP was measured in 1119 (47.7%) adolescents, with mean systolic BP 105.9 mmHg and mean diastolic BP 65.2 mmHg. There was little evidence of association between BW and systolic [regression coefficient  $\beta = 0.14$ , 95% confidence interval (CI) (-1.00, 1.27)] or diastolic [ $\beta = 0.43$ , 95% CI (-0.57, 1.43)] BP. ...

# Empirical research process - example

*International Journal of Epidemiology*, 2019, 148–156 doi: 10.1093/ije/dyy118

## **Are birthweight and postnatal weight gain in childhood associated with blood pressure in early adolescence? Results from a Ugandan birth cohort**

**Background:** In Africa, where low birthweight (LBW), malnutrition and high blood pressure (BP) are prevalent, the relationships between birthweight (BW), weight gain and BP later in life remain uncertain. We examined the effects of early life growth on BP among Ugandan adolescents.

**Methods:** Data were collected prenatally from women and their offspring were followed from birth, with BP measured following standard protocols in early adolescence. Weight-for-age Z-scores (WAZ) were computed using World Health Organization references. Linear regression was used to relate BW, and changes in WAZ between birth and 5 years, to adolescents' BP, adjusting for confounders.

**Results:** Among 2345 live offspring, BP was measured in 1119 (47.7%) adolescents, with mean systolic BP 105.9 mmHg and mean diastolic BP 65.2 mmHg. There was little evidence of association between BW and systolic [regression coefficient  $\beta = 0.14$ , 95% confidence interval (CI) (-1.00, 1.27)] or diastolic [ $\beta = 0.43$ , 95% CI (-0.57, 1.43)] BP. ...

Which steps lie behind this description and these results?  
(How) Can they be reproduced?

# Statistical software

- R
- SPSS (PSPP)
- SAS (SAS Studio)
- Excel
- Stata
- Python
- Specialized software, e.g. Mplus, PLINK, ...
- ...

# History of R

- 1980s: S, S-PLUS developed by Becker & Chambers.<sup>1</sup>
- Ross Ihaka & Robert Gentleman (University of Auckland): development of reduced version of S: R.<sup>1</sup>
- 1995: start of initiative to publish R under the GPL and Formation of core team with 19 members.<sup>1</sup>
- 2000: release of version 1.0.0.<sup>1</sup>
- Current version: 4.1.1 (1 year ago: 4.0.3)
- New version approx. every 2-3 months, mostly small updates that do not make any/a big difference in standard analyses.

---

<sup>1</sup>Dalgaard (2008). Introductory Statistics with R.

# History of R

- 1980s: S, S-PLUS developed by Becker & Chambers.<sup>1</sup>
- Ross Ihaka & Robert Gentleman (University of Auckland): development of reduced version of S: R.<sup>1</sup>
- 1995: start of initiative to publish R under the GPL and Formation of core team with 19 members.<sup>1</sup>
- 2000: release of version 1.0.0.<sup>1</sup>
- Current version: 4.1.1 (1 year ago: 4.0.3)
- New version approx. every 2-3 months, mostly small updates that do not make any/a big difference in standard analyses.

## So what is R?

- Statistical computer program
- Complete programming language
- Environment to perform statistical analyses, produce graphics

---

<sup>1</sup>Dalgaard (2008). Introductory Statistics with R.

# Why R?

- Free, open source (General Public License) - i.e. can see what is done in the analysis, adapt and contribute. Also, easy transport of files between programs/to other programs.
- Active methods development in/for R: if you find a paper with a new statistical method you want to apply, there is a very good chance it is implemented in R.
- Large active online community, tutorials, documentation, manuals, help forums etc. → if you have problems in your analysis, almost always you can find an answer.



# Why R?

- Free, open source (General Public License) - i.e. can see what is done in the analysis, adapt and contribute. Also, easy transport of files between programs/to other programs.
- Active methods development in/for R: if you find a paper with a new statistical method you want to apply, there is a very good chance it is implemented in R.
- Large active online community, tutorials, documentation, manuals, help forums etc. → if you have problems in your analysis, almost always you can find an answer.
- Increasing number of users in education, research, industry.

# Why R?

- Free, open source (General Public License) - i.e. can see what is done in the analysis, adapt and contribute. Also, easy transport of files between programs/to other programs.
- Active methods development in/for R: if you find a paper with a new statistical method you want to apply, there is a very good chance it is implemented in R.
- Large active online community, tutorials, documentation, manuals, help forums etc. → if you have problems in your analysis, almost always you can find an answer.
- Increasing number of users in education, research, industry.
- Runs (to my experience) robustly on many operating systems, e.g. Windows, Mac, Linux, ...
- Can be run on servers, through bash scripts, and used for large multi-center studies (but has challenges for really large data).

# Why RStudio?

- Nice graphical interface, with additional options and functionalities that make working easier (e.g. documentation)

# Why RStudio?

- Nice graphical interface, with additional options and functionalities that make working easier (e.g. documentation)
- Runs "on top of" R (e.g. you can also do all analyses and creation of plots and tables without RStudio).

# Why RStudio?

- Nice graphical interface, with additional options and functionalities that make working easier (e.g. documentation)
- Runs "on top of" R (e.g. you can also do all analyses and creation of plots and tables without RStudio).
- Very good people working on the development and adding new things (e.g. Hadley Wickham) – e.g. work in the cloud (<https://rstudio.cloud/>), write packages, create reports, apps, presentations, websites, CV ...

## And why not others?

- SPSS: can do all analyses by "clicking" without programming (coding is also possible), with all upsides and downsides. Not free (free replica PSPP, with limited functionality), not open source, only restricted analyses possible, not running on all systems, cannot run on servers (at least very hard to).

## And why not others?

- SPSS: can do all analyses by "clicking" without programming (coding is also possible), with all upsides and downsides. Not free (free replica PSPP, with limited functionality), not open source, only restricted analyses possible, not running on all systems, cannot run on servers (at least very hard to).
- SAS: Can do some analyses by clicking, generally made for programming. Very high costs (but since recent years free SAS Studio version available), only restricted analyses possible in base version, problems on different servers etc.

## And why not others?

- SPSS: can do all analyses by "clicking" without programming (coding is also possible), with all upsides and downsides. Not free (free replica PSPP, with limited functionality), not open source, only restricted analyses possible, not running on all systems, cannot run on servers (at least very hard to).
- SAS: Can do some analyses by clicking, generally made for programming. Very high costs (but since recent years free SAS Studio version available), only restricted analyses possible in base version, problems on different servers etc.
- Python: Similar to R: increasing user base and applications, especially in machine learning.

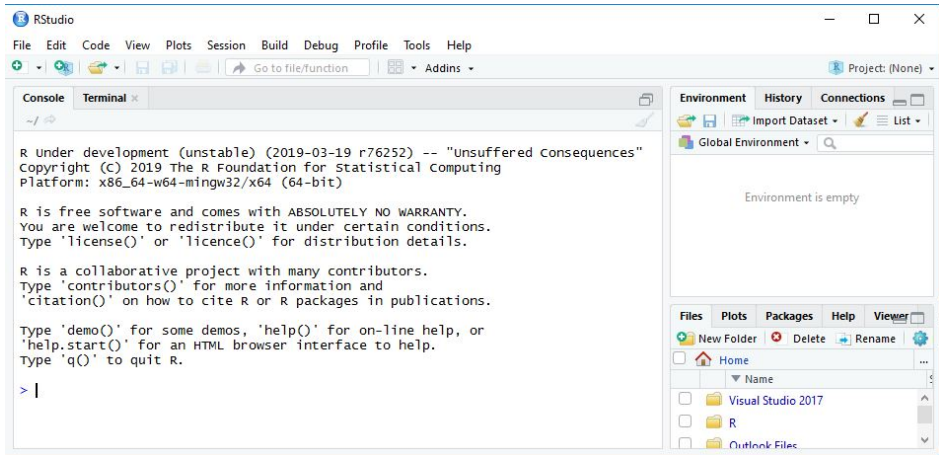


# Installation of R and RStudio

- R: Download and install most recent, archived, and development versions from <https://cran.r-project.org/>.
- RStudio: Download and install from [www.rstudio.com/products/rstudio/download/](http://www.rstudio.com/products/rstudio/download/).
- Help for the installation can be found e.g. at <http://r-tutorial.nl/>.
- Update R e.g. using the functions in the `installr` package (<https://cran.r-project.org/web/packages/installr>).

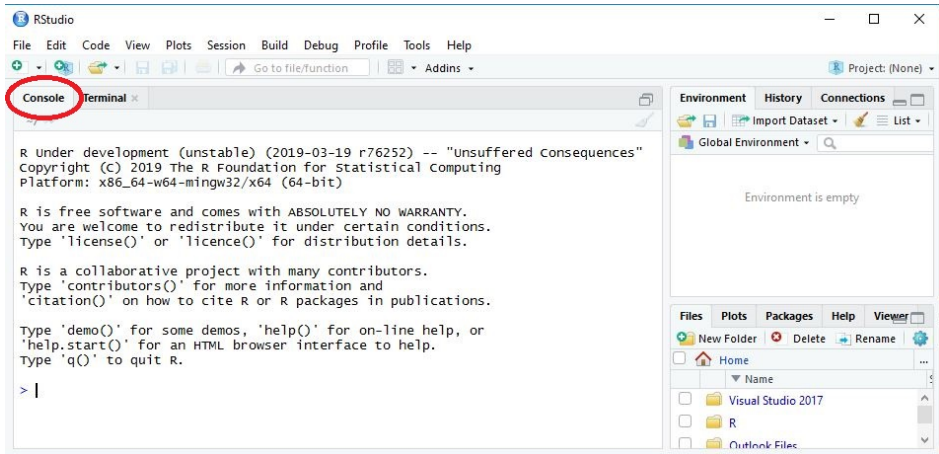
## Overview of windows and functionalities in RStudio

# First look at RStudio



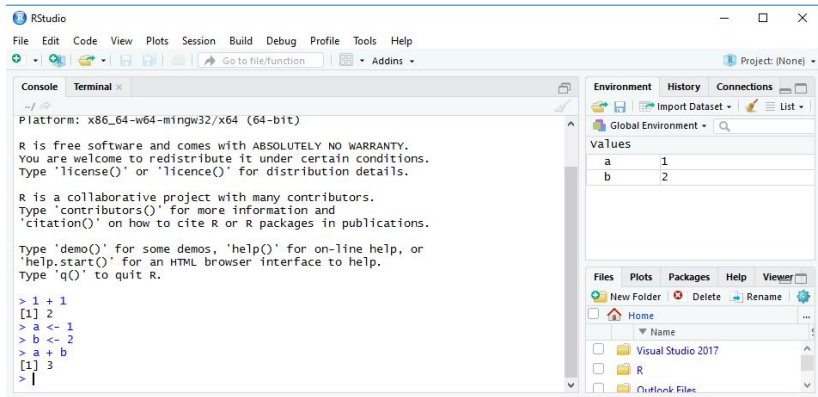
# R Console

Here you can type in your R commands and execute them by pressing Enter:



# R Console

For example:



The screenshot shows the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar. The main window is divided into three panes. The left pane is the Console, which shows the R startup message and the results of several commands. The right pane is the Environment pane, which shows the Global Environment and a table of values for variables 'a' and 'b'.

```
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.

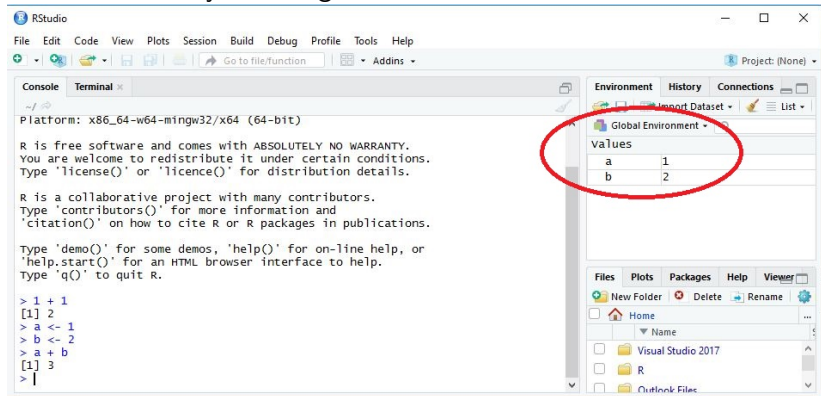
> 1 + 1
[1] 2
> a <- 1
> b <- 2
> a + b
[1] 3
> |
```

Values	
a	1
b	2

Many further options to use R as a calculator, e.g. `log`, `exp`, ...

# R Console

After running these lines of R code, the objects `a` and `b` are created, and they are assigned the values 1 and 2:



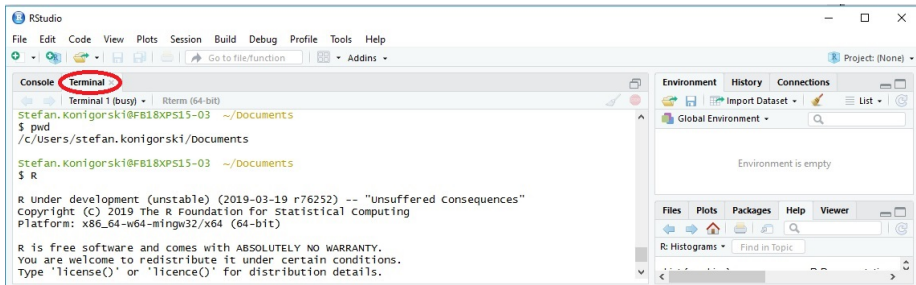
They are in the global environment, can be used in the analyses, and they can also be saved, for example.

# Exercise 1

- Open the R script 'R\_1\_exercise\_1.R' in RStudio by double clicking on the file, or open it with File ► Open.
- Go through points (1) - (7) and run the commands by copy-pasting them into the Console and pressing Enter.
- Look at the results and try to understand/take a guess what you have computed.

# Terminal

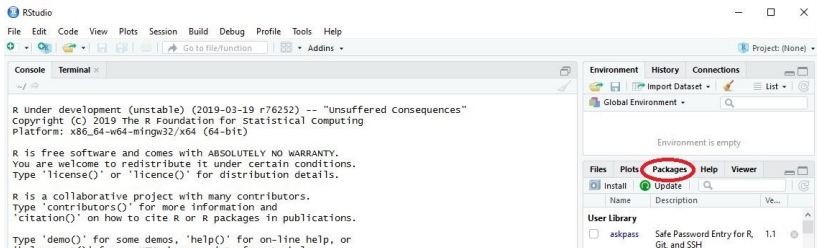
"Linux-style command line" to do stuff:





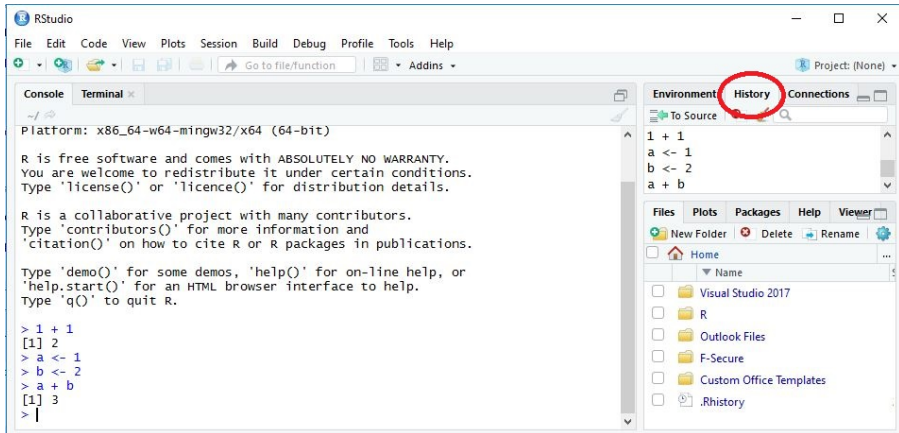
# R packages

- Analyses in R are done using functions, e.g. `+` or `t.test`.
- Many are in the automatically loaded base, stats packages.
- Functions in other packages must be loaded, and the packages installed before use - either through the package tab, or through `install.packages("packagename")`
- More documentation for all packages is available in the manual and (for some packages) vignette at <https://cran.r-project.org/package=packagename>.



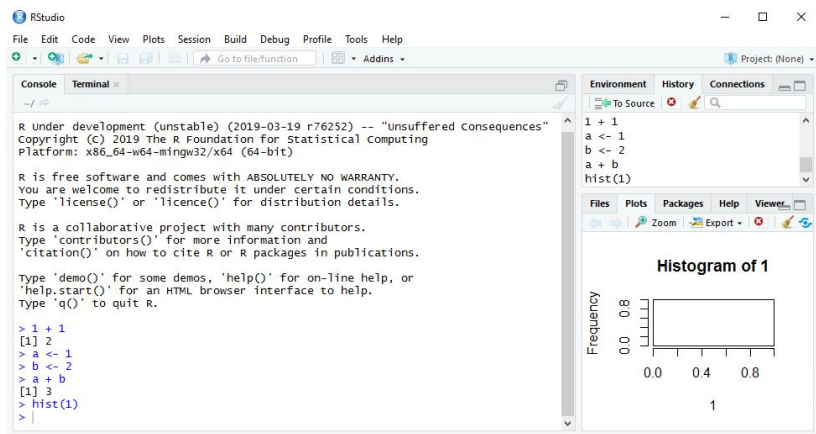
# History

All code that has been run can be viewed in the History panel:



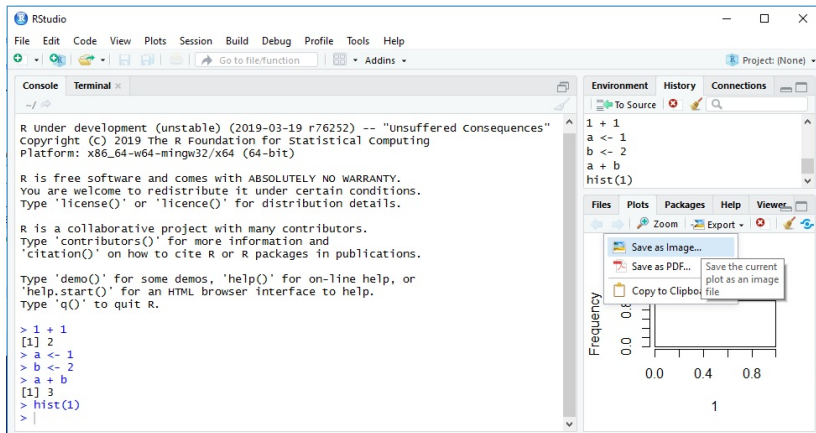
# Graphics

Plots can be generated by typing commands in the console and pressing enter. The results are shown in the "Plots" Tab:



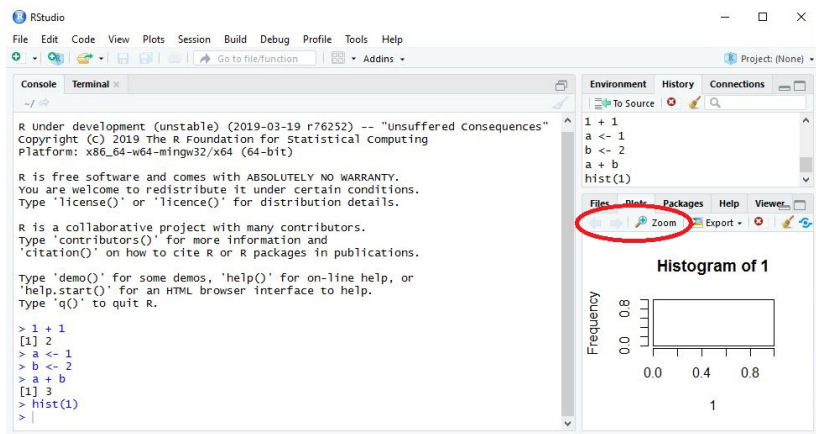
# Save images

The image can be exported and saved:



# Further handling of images

Images can be opened in a separate window to zoom in.  
Also, buttons allow to go back and forth between created images.



The screenshot shows the RStudio environment. The console window displays the R startup messages, including the version (2019-03-19 r76252) and the platform (x86\_64-w64-mingw32/x64 (64-bit)). The plot window shows a histogram titled "Histogram of 1" with a frequency axis ranging from 0.0 to 0.8 and an x-axis ranging from 0.0 to 0.8. The "Zoom" button in the plot window's toolbar is circled in red.

```
R Under development (unstable) (2019-03-19 r76252) -- "Unsuffered Consequences"
Copyright (C) 2019 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.

> 1 + 1
[1] 2
> a <- 1
> b <- 2
> a + b
[1] 3
> hist(1)
> |
```

Histogram of 1

Frequency

0.0 0.8

0.0 0.4 0.8

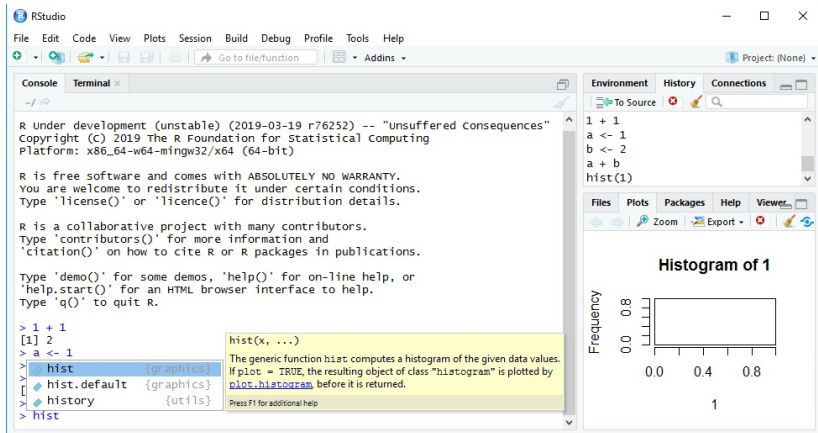
1

## Exercise 2

- Generate a new histogram with the command `hist(c(1,1,1,2,4,2,5,2,6,7,8,9,101,100))`.
- Try to understand what information is shown in the plot that you have created.
- Save the graphic as a pdf file on your desktop.
- See 'R\_1\_exercise\_2.R'.

# Help with typing and functions

Help for the functions can be seen while typing:



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Project: (None)

Environment History Connections

To Source

Files Plots Packages Help Viewer

Zoom Export

Histogram of 1

Frequency

0.0 0.8

0.0 0.4 0.8

1

1 + 1  
[1] 2  
a <- 1  
hist {graphics}  
hist.default {graphics}  
history {utils}

hist(x, ...)

The generic function hist computes a histogram of the given data values. If plot = TRUE, the resulting object of class "histogram" is plotted by plot.histogram before it is returned.

Press F1 for additional help

R Under development (unstable) (2019-03-19 r76252) -- "unsuffered Consequences"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (64-bit)

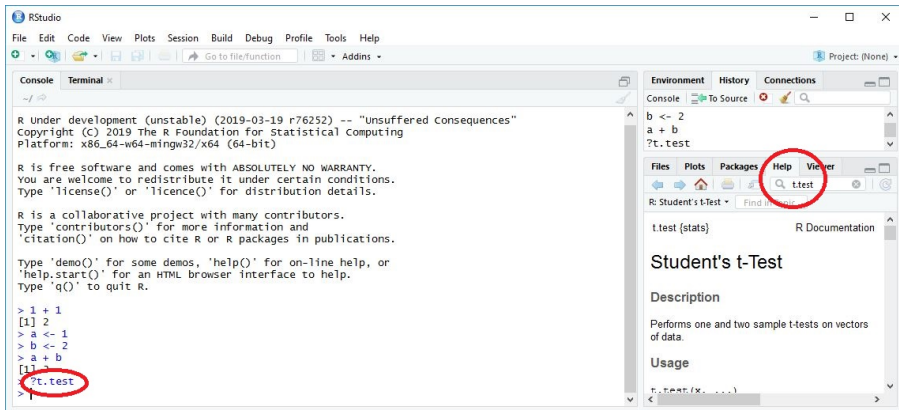
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

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.

# General help

The help sites for functions and packages can be reached by typing `?functionname` in the console (and pressing enter) or through the help panel:





## First contact with data frames

# Load data

In order to work with data in R, the data can be:

- Imported from external files (e.g. xls, txt, ... files).
- Typed in directly into R (e.g. in the console).
- Loaded, if the data has already been saved in R format: `.RData` (or `.rdata`, `.rda`, `.rds`).

# Existing datasets in R

Also, several toy datasets are always loaded in R, e.g. the `mtcars` datasets of 32 cars. Show the head (i.e. first 6 rows) of the data:

The screenshot shows the RStudio interface. The console window displays the following text:

```

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.

> 1 + 1
[1] 2
> a <- 1
> b <- 2
> a + b
[1] 3
> hist(1)
> head(mtcars)
      mpg  cyl  disp  hp drat   wt  qsec vs  am  gear  carb
Mazda RX4    21.0   6  160 110 3.90 2.620 16.46 0   1    4    4
Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02 0   1    4    4
Datsun 710    22.8   4  108  93 3.85 2.320 18.61 1   1    4    1
Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44 1   0    3    1
Hornet Sportabout 18.7  8  360 175 3.15 3.440 17.02 0   0    3    2
valiant      18.1   6  225 105 2.76 3.460 20.22 1   0    3    1
  
```

The Environment pane on the right shows the following objects:

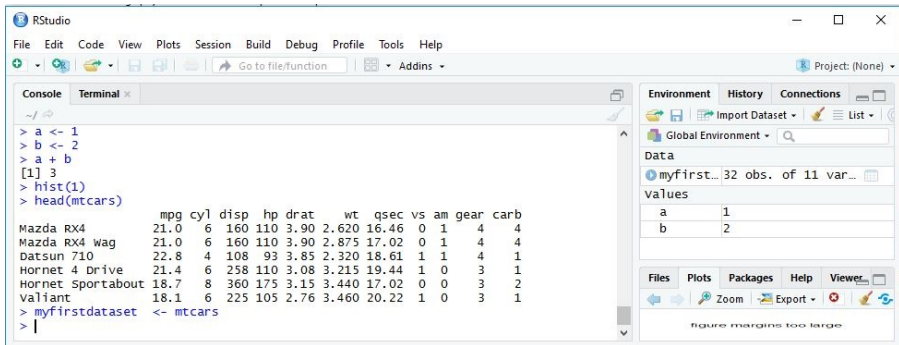
```

a <- 1
b <- 2
a + b
hist(1)
head(mtcars)
  
```

The Plots pane on the right shows a histogram titled "Histogram of 1". The x-axis is labeled "1" and ranges from 0.0 to 0.8. The y-axis is labeled "Frequency" and ranges from 0.0 to 0.8. The histogram shows a single bar with a frequency of 1.

# First contact with data frames

Copy mtcars data frame into new object myfirstdataset:



The screenshot shows the RStudio interface. The Console pane on the left contains the following R code and its output:

```
> a <- 1
> b <- 2
> a + b
[1] 3
> hist(1)
> head(mtcars)
```

The output of `head(mtcars)` is a data frame with 6 rows and 11 columns:

	mpg	cyl	displacement	horsepower	drat	weight	quarter mile time	vs	am	gear	carburetors
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

The Environment pane on the right shows the Global Environment with the object `myfirstdataset` containing 32 observations of 11 variables. The values of `a` and `b` are also displayed in the Environment pane.

# First contact with data frames

Show data frame with nicer formatting:

The screenshot shows the RStudio interface. The main editor displays a data frame with 32 entries. The columns are: mpg, cyl, disp, hp, drat, wt, qsec, vs, am, gear, carb. The rows include Mazda RX4, Mazda RX4 Wag, Datsun 710, Hornet 4 Drive, Hornet Sportabout, Valiant, Duster 360, Merc 240D, Merc 230, Merc 280, Merc 280C, Merc 450SE, and Merc 450SL. The status bar indicates 'Showing 1 to 14 of 32 entries'.

The Environment pane on the right shows the data frame 'myfirstd' with 32 observations and 11 variables. The 'Data' section is circled in red. Below it, the 'values' section shows the first two rows of the data frame.

The Files pane on the right shows a histogram titled 'Histogram of 1'. The x-axis is labeled '1' and ranges from 0.0 to 0.8. The y-axis is labeled 'Frequency' and ranges from 0.0 to 0.8. The histogram is currently empty.

The Console pane at the bottom shows the following commands:

```
> myfirstdatasets = mtcars
> view(myfirstdatasets)
```

The 'view(myfirstdatasets)' command is circled in red.

## Documentation with R

# What is an R Script? Why use it?

- R script = text file where R code can be written, saved, and read into R.

# What is an R Script? Why use it?

- R script = text file where R code can be written, saved, and read into R.
- Open new R script: File ► New File ► R Script.
- Can write code to remember it, to document it, and to run it from there.
- Can also add comments: everything after '#' is treated as a comment (marked in green) and is not be run.



# What is an R Script? Why use it?

- R script = text file where R code can be written, saved, and read into R.
- Open new R script: File ► New File ► R Script.
- Can write code to remember it, to document it, and to run it from there.
- Can also add comments: everything after '#' is treated as a comment (marked in green) and is not be run.
- Run code by highlighting some parts of the code and clicking on "Run" (or pressing Ctrl+Enter).

# What is an R Script? Why use it?

- R script = text file where R code can be written, saved, and read into R.
- Open new R script: File ► New File ► R Script.
- Can write code to remember it, to document it, and to run it from there.
- Can also add comments: everything after '#' is treated as a comment (marked in green) and is not be run.
- Run code by highlighting some parts of the code and clicking on "Run" (or pressing Ctrl+Enter).
- Save as .R (or .r) file.

# Create new R Script

The screenshot shows the RStudio interface. The 'File' menu is open, and the 'R Script' option is highlighted with a red circle. The keyboard shortcut 'Ctrl+Shift+N' is also visible. Below the menu, a portion of the 'mtcars' dataset is visible in the console.

**RStudio File Menu Options:**

- New File
- New Project...
- Open File... (Ctrl+O)
- Recent Files
- Open Project...
- Open Project in New Session...
- Recent Projects
- Import Dataset
- Save (Ctrl+S)
- Save As...
- Save All (Ctrl+Alt+S)
- Print...
- Close (Ctrl+W)
- Close All (Ctrl+Shift+W)
- Close All Except Current (Ctrl+Alt+Shift+W)
- Close Project
- Quit Session... (Ctrl+Q)
- R Script (Ctrl+Shift+N)
- R Notebook
- R Markdown...
- Shiny Web App...
- Text File
- C++ File
- R Sweave
- R HTML
- R Presentation
- R Documentation

**Console Output:**

```

> myfirstdataset <- mtcars
> View(myfirstdataset)
>

```

**Environment Panel:**

Global Environment

myfirstdataset 32 obs. of 11 variables

# Example of a simple R Script

The screenshot displays the RStudio interface. The main editor window shows a script file named 'Untitled1\*' with the following content:

```
1 ### This is a test file to call the help for the 'hist' function
2
3 ?hist
```

A red oval highlights the comment on line 1. The 'Source' pane on the right shows the help documentation for the `hist` function:

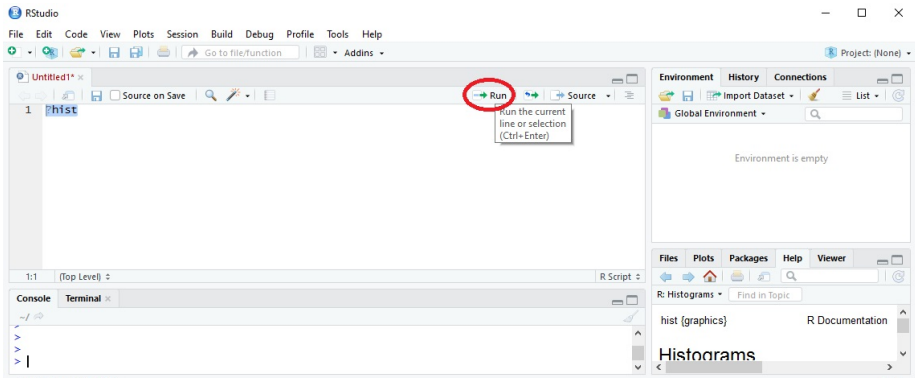
**hist**

The generic function `hist` computes a histogram of the given data values. If `plot = TRUE`, the resulting object of class "histogram" is plotted by `plot.histogram` before it is returned.

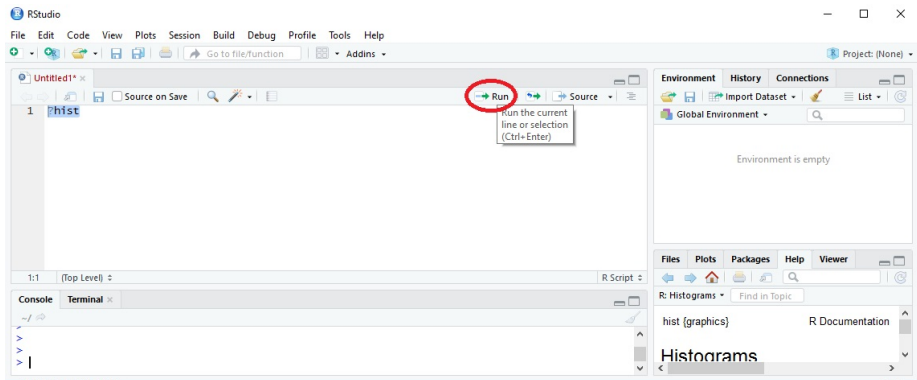
Press F1 for additional help

The 'Environment' pane on the right shows 'Global Environment' with the message 'Environment is empty'. The 'Plots' pane on the right shows 'R: Histograms' with the message 'Find in Topic' and a list of topics including 'hist (graphics)' and 'Histograms'.

# Run R Script



# Run R Script



Save R script at your desired location, then open it the next time you start RStudio.

# R Markdown

## What is Markdown?

- Simple language for formatting text input and creating HTML, PDF, and MS Word documents.

# R Markdown

## What is Markdown?

- Simple language for formatting text input and creating HTML, PDF, and MS Word documents.

## What is R Markdown?

- "Enhanced" R script.
- Interface of R with Markdown, to combine text with pieces of R code ("chunks") in one document.
- From this R Markdown file, a formatted pdf/html/word file can be generated by "knitting" the file through "knit", which contains the text, R code, and the results from these analyses.

→ more next week



# Summary: general overview how to work with R

## General steps

- Load (.Rdata) or import data.
- Load R packages and R scripts, or open R script (.R) / Markdown (.Rmd) file that includes functions for the analysis.
- Perform data analyses by running the R functions step-by-step: compute 1, use results from 1 in 2 to compute 3, ...
- Save edited R script/Markdown file, save results.

## Run R code

- by writing code in "Console" and running it there (press enter)
- by writing code in R script, and run by clicking on "Run" (or pressing Ctrl+Enter)
- (by writing code in R Markdown file and running it)

## Homework

# Homework

See file `R_1_homework.R`

Questions?