Find us...

📅 Events
https://www.meetup.com/rladies-tbilisi/

📝 Materials !!
https://github.com/rladies/meetup-presentations_tbilisi

👍 Social Media
Twitter @rladiestbilisi
Facebook @rladiestbilisi/

tbilisi@rladies.org

emtg.co@gmail.com ✉

If you haven't yet

- Download R - choose your operating system
  https://cran.r-project.org/

- Download R Studio - choose your OS
  https://www.rstudio.com/products/rstudio/download/

What is R? Why R?

- Free software for statistical analysis and graphics

- Programme own methods & use existing tools (packages) for own jobs/ tasks/ purposes

- R is command-driven → type in a command, execute it

- Allows independence and flexibility

- Big supporter-developer community

What is R? Why R?

1976   S programme for statistics and graphics developed by John Chambers, Rick Becker and Allan Wilks (Bell Laboratories)

1992   Implementation of S by Ross Ihaka and Robert Gentleman (University of Auckland, New Zealand), renamed it R

Currently run by the R Development Core Team

Some things you can do with R

Reading & creating data

Data wrangling, cleaning, manipulation

Statistical analysis

Spatial analysis, mapping

Data visualisation, plotting

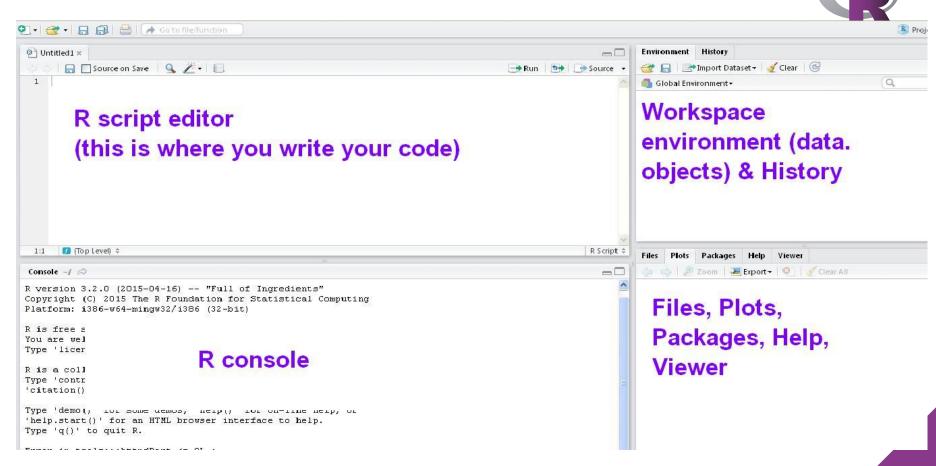**Web scraping**

Machine learning

RStudio - Integrated Development Environment (IDE) for R

Specifically designed interface for using R

Makes it easier to write code (auto completion, code highlight), jump between projects and working directories, look up definitions of functions, etc.

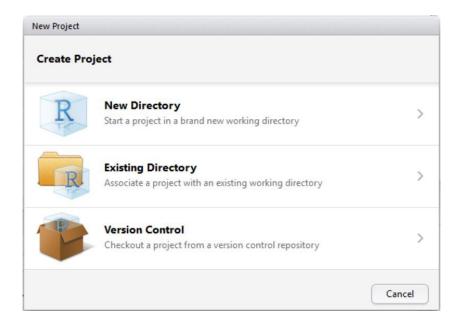# The RStudio environment

R and RStudio

Latest version of R (3.4.4) https://r-project.org/

IDE RStudio https://www.rstudio.com/

Another way to start

Go to: File → New project….

Setting your working directory

- Working directory - the folder where you will be keeping your files
- If you don't know your current directory, type "getwd()":

```
getwd()
```

```
[1] "C:/Documents and Settings/User/My Documents"
```

- To change that or to specify the folder name you will be working in:

```
setwd("C:/Documents and Settings/User/Something")
```

R requires **/** for Windows   (try \ or \\ for other OS)

Starting to code

- Hashtags **#** for writing comments and notes

```
#This is a comment
This is not a comment
```

- "<-" is the usual assign operator to create **objects**

```
a <- 5
b <- "string object"
```

- Get the output by printing the name of the object ("a" or "b")

```
[1] 5

> b
[1] "string object"
```

Object names are case sensitive (a≠A)!

Everything is an object

We created object called "a" already which consists of one single value

```
a <- 5
```

A **vector** is an object that consists of same type of data elements (one-dimensional object). To create a vector we can combine a list of elements using the **function c( )**.

```
c(1,2,3,4,5)
```

You can assign a name to this vector

```
V1 <- c(1,2,3,4,5)
```

Operators in R

ARITHMETIC

+ addition

- subtraction

* multiplication

/ division

^ power

%% remainder (after division)

COMPARISON

< lesser than

> greater than

<= lesser than or equal to

>= greater than or equal to

== equal

!= different

LOGICAL

! NOT

& AND

| OR

In-built functions, some of them

GENERAL

c( )	cbind( )
ls( )	list.files( )
seq( )	sort( )
paste( )	rep( )
data( )	q( )
rm(list=ls( ))	

MATH

log( )

sqrt( )

sum( )

STATISTICS

cor.test( )

t.test( )

mean( )

sd( )

lm( )

GRAPHICAL

plot( )

hist( )

abline( )

png( )

Help function

`help(foreign)` Reading and writing data stored by some versions of 'Epi Info', 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', and for reading and writing some 'dBase' files.

or

?

Packages == extensions, add-ons

Core functions built in

function1( )

function2( )

function3( )

Base R

function4( )

functionA( )

functionB( )

functionC( )

Package A

functionD( )

functionE( )

functionF( )

Package B

Using Packages

1. Install files into computer (package name in the brackets)

```
install.packages("foreign")
```

→ **1 X per computer**

2. "Load" the library/ package

```
library("foreign")
```

→ **1 X per R session**

Sometimes you can use the menu for installing packages that you need

Errors, errors, errors

- Read your error
- R requires balanced brackets ()
- Check with or without " "
- Completed expressions   + *at the end of the line*

- Google your error - use `r, rstat, rlang, cran` keywords
- Copy-pasting R code from a website to terminal is not recommended

Your turn

1. Set your working directory.

2. Create six objects named var0, *var1, var2, var3, var4, var5, var6, var7* with the following content: 5, 2+2, 4*5, 240/60, 4^15, "this is", "my object", "5"

3. What is the difference between var0 and var7?

4. Do a summing operation with var0 and var3.

5. Paste var5 and var6 into *var56*.

6. Install the following packages: *foreign*, *readr*. Load them.

7. Read about these packages - what are they for?

8. Save your R file in your working directory.

Data types, objects, structures and handling data

Data types

Most common data types in R:

- Numeric
- Logical
- Character
- Integer
- Factor

- Missing values labelled: **NA**

Factors

- vector contains a set of numeric codes with character-valued levels

 > kids = factor(c(1,0,1,0,0,0), levels = c(0, 1),

          labels = c("boy", "girl")

Often factors will be automatically created (from character or string data) when you read a dataset in using a function like `read.table()`

Data structures

	Homogeneous	Heterogeneous
1d	Atomic vector	List
2d	Matrix	Data frame
nd	Array	

Source: Data structures in Advanced R by Hadley Wickham

Variable names

- These are reserved for other purposes, don't use them as variable names: c q t C D F(ALSE) I T(RUE) Inf NaN NA

- Start your variable name with a letter

- Variable names are case sensitive (a≠A)

Vector

To create a vector with more than one element

```
V1 <- c(1,2,3,4,5)
```

Create a vector with character elements

```
V2 <- c("red","blue","green")
```

The length differs because the number of elements differs

```
length(v1)          length(v2)
```

Check the class of both vectors

```
class(v1)          class(v2)
```

Matrix

A matrix is a two-dimensional rectangular data set with same type of elements.

```r
M1 <- matrix(c("a","a","b","c","b","a"), nrow = 2, ncol = 3)
```

| Fill/ content | Number of rows | Number of columns |

```r
M2 <- matrix(0, 3, 2)
```

Count the dimensions of matrix

```r
dim(M1)
```

Data frame

Tabular data objects where each column can contain different modes of data. It is a list of vectors of equal length.

```r
df1 <- data.frame(v1, v1 * 10)
    V3 <- c(6,7,8,9,10)
df2 <- data.frame(v1,v3)
```

Count the dimensions of a data frame
```r
dimnames(df1)        nrow(df1)              ncol(df1)
```

List

Lists are a special type of vector that can contain elements of different classes, including lists.

```r
mylist <- list(1, "a", TRUE, 1 + 4i)
```

Count the dimensions of a list

Using in-built datasets

To see which datasets are available within R
`data()`

Load a dataset called Titanic
`data(Titanic)`

Read about the dataset, its variables
`?Titanic`

Exploring data

`View(Titanic)`   *#View opens the data in a separate window*

`class(Titanic)`   *#class tells you the class/format of the data*

`ftable(Titanic)`   *#gives a flat contingency table*

`prop.table(Titanic)`   *#gives a table of proportions*

`table(age, sex)`   *#frequency table function*

Exploring data

```
Titanic2 <- data.frame(Titanic)   #Change it into a data frame (new object)
```

```
head(Titanic2)         #Shows 6 first observations

tail(Titanic2)         #Shows 6 last observations

str(Titanic2)          #Structure of the dataset (eg variable types)

summary(Titanic2)      #Summarises main indicators of variables (eg median, mean)

names(Titanic2)        #Lists all the variable names in the data
```

Extracting and subsetting

Data objects in R are indexed. These indices can be used to extract/subset vectors, matrices, data frames and lists. Named dimensions can be extracted using the operator $.

[returns an object of the same class; can be used to select multiple elements of an object

[[extract elements of a list or a data frame; can only be used to extract a single element and the class of the returned object will not necessarily be a list or data frame

$ extract elements of a list or data frame by literal name

Extracting and subsetting

```
V1[2]          #extract 2nd element of vector

V1[2:5]        #extract 2nd to 5th  elements of vector

M1[2,3]        #extract element from 2nd row, 3rd column of matrix

df1[4,1]       #extract element from 4th row, 1st column of data frame

df1[,1]        #extract the1st column of data frame (and all rows)

mylist[[3]]    #extract the 3rd element of list xy

df1$v1         #values of vector v1 from data frame df1 are selected
```

Importing data

read...( ) function

`read.csv( )` comma delimited

`read.csv2( )` semi-colon delimited

`read.table("file_name.txt")` reads a
text file into a table

Package "readr"

`read_csv( )`

`read_csv2( )`

`read_tsv( )` tab delimited

Package "foreign" for SPSS, Stata ("haven" + SAS)

`read.spss("file_name.sav", sep="", header==T)`

No space = no separation mark

1st row - variable name

`read.dta( )`

Package "readxl" for xls and xlsx

`read_excel( )`

Exporting data

#Into SPSS or Stata format

```
write.foreign(cars, "cars.csv", "cars.sav",   package="SPSS")
write.dta(cars, "cars.dta")
```

#Into csv format

```
write.csv(cars, file = "cars_new.csv", row.names = FALSE)
write.csv(cars, "cars_new.csv",row.names=FALSE, na="")
write.table(cars, "cars_new.csv",row.names=FALSE, na="",
col.names=FALSE, sep=",")
```

Your turn

1. Load data "iris" from the R datasets
2. Explore the dataset. How many variables does it include? What is the factor variable called? How many levels does the factor variable have?
3. What species are the first observations in the dataset from?
4. What is the maximum value of each variable in the dataset?
5. Run a flat contingency table of the "Petal.Width" variable
6. Create a new object "sep_length" with the first 10 values of "Sepal.Length" variable.
7. Save the file as a csv file in your working directory.

Resources

[Advanced R](#) by Hadley Wickham

[Data Visualization for Social Science](#) by Kieran Healy (draft)

[RStudio cheat sheets](#)

Resources

- R-bloggers

  Stackoverflow

  Rseek - look for a package

- Online coding school Codeschool

  Learn R on youtube

  Instructions for installing the swirl package

- Google group for ggplot2 (visualisation package)