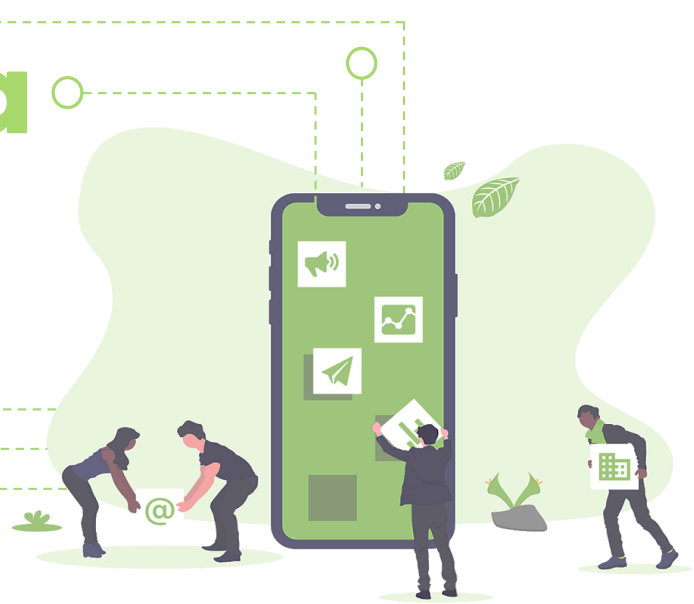
The Data Science Track



Prepared By: R. Daynolo

1

1

7. Overview and History of R

2

Getting Started with R

- To enable us to use R, we firstly discuss its capabilities (as well as its costs and benefits), then we describe how to install it, and then we illustrate some basic commands.
- Note that we have various ways of "communicating" with R
  - Interactively: (through console)
  - Batch Processing: (through scripts)
  - Point and Click: (through packages Rcmdr, rattle, deducer)

What is R?

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hardcopy

What is R?

- R is a statistical programming environment for performing standard and specialized statistical methods
  - "environment" : a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software
- R is a free/open source statistical package
  - based on S language developed at Bell Laboratories

What is S?

- S is a language that was developed by John Chambers and others at Bell Labs.
- S was initiated in 1976 as an internal statistical analysis environment – originally implemented as Fortran libraries.
- Early versions of the language did not contain functions for statistical modeling.
- In 1988 the system was rewritten in C and began to resemble the system that we have today (this was Version 3 of the language). The book ***Statistical Models in S*** by Chambers and Hastie (the white book) documents the statistical analysis functionality.

What is S?

- Version 4 of the S language was released in 1998 and is the version we use today. The book ***Programming with Data*** by John Chambers (the green book) documents this version of the language.

7

Historical Notes

1991: Created in New Zealand by Ross Ihaka and Robert Gentleman of University of Auckland. Their experience developing R is documented in a 1996 *JCGS* paper.
1993: First announcement of R to the public.
1995: Martin Mächler convinces Ross and Robert to use the GNU General Public License to make R free software.
1996: A public mailing list is created (R-help and R-devel)
1997: The R Core Group is formed (containing some people associated with S-PLUS). The core group controls the source code for R.
2000: R version 1.0.0 is released.
2018: R Version 3.5.0 is released.

8

Back to R

- Although R is a programming language, i.e. generating computer code to complete tasks is required, there are now Graphical User Interface (GUI) Add Ons like R Commander, and rattle, which allow users to "point and click".
- Initially developed by Robert Gentleman and Ross Ihaka and now maintained by the "R core development team"
- Cross platform compatibility: Windows, MacOS, Linux

Costs and Benefits of R

ADVANTAGES	DISADVANTAGES
• Fast and free.	• Not user friendly, i.e. steep learning curve with minimal GUI.
• State of the art: Statistical researchers provide their methods as R packages.	• No commercial support; figuring out correct methods or how to use a function on your own can be frustrating.
• Some packages, such as Mx, WinBugs, and other programs use or will use R.	• Easy to make mistakes and not know.
• Active user community	• Some users complain about hostility on the R listserve
• Excellent for computer intensive analyses, etc., and Interfaces with database storage software (SQL)	• Working with large datasets is limited by RAM
• Forces you to think about your analysis.	• Data prep & cleaning can be messier & more mistake prone in R vs. SPSS or SAS

Costs and Benefits of R

Free Software!

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

https://www.fsf.org/

Design of the R System

The R system is divided into 2 conceptual parts:

1. The "base" R system that you download from CRAN
2. Everything else

Design of the R System

R functionality is divided into a number of packages:

- The "base" R system contains, among other things, the **base** package which is required to run R and contains the most fundamental functions.
- The other packages contained in the "base" system include **utils, stats, datasets, graphics, grDevices, grid, methods, tools, parallel, compiler, splines, tcltk, stats4**.
- There are also "Recommend" packages: **boot, class, cluster, codetools, foreign, KernSmooth, lattice, mgcv, nlme, rpart, survival, MASS, spatial, nnet, Matrix.**

R Packages

- When you download R from the Comprehensive R Archive Network (CRAN), you get that "base" R system
- The base R system comes with basic functionality; implements the R language
- One reason R is so useful is the large collection of packages that extend the basic functionality of R
- R packages are developed and published by the larger R community

Obtaining R Packages

- The primary location for obtaining R packages is **CRAN**.
- For biological applications, many packages are available from the **Bioconductor Project**.
- You can obtain information about the available packages on CRAN with the `available.packages()` function.

```
a <- available.packages()
head(rownames(a), 3)  ## Show the names of the first few packages
```

```
## [1] "A3"        "abc"       "abcdeFBA"
```

- Currently, the CRAN package repository features **13,626** available packages, covering a wide range of topics.

Installing R Packages

- Packages can be installed with the `install.packages()` function in R.
- To install a single package, pass the name of the package to the `install.packages()` function as its first argument.
- The following code installs the **ggplot2** package from CRAN

```
install.packages("ggplot2")
```

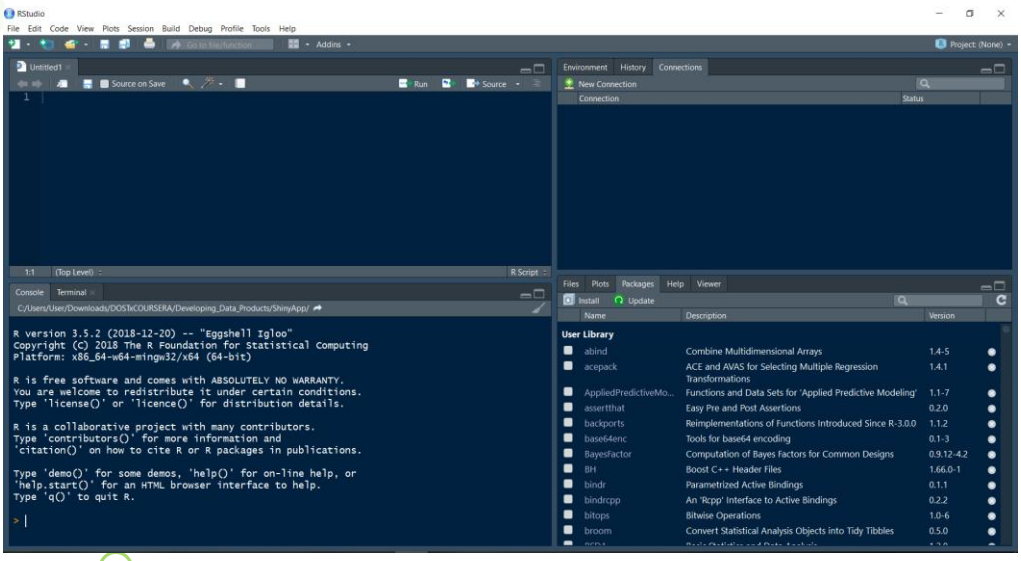
- The command downloads the hhplot2 package from CRAN and installs it on your computer
- Any packages on which this package depends will also be downloaded and installed.

Installing R Packages

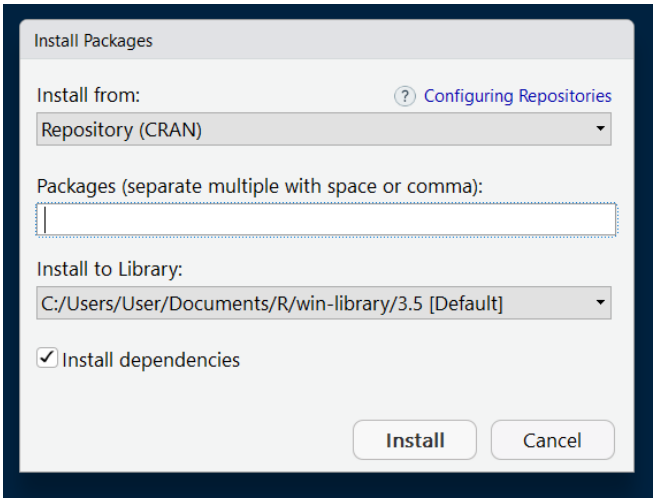
- You can install multiple R packages at once with a single call to `install.packages()`
- Place the names of the R packages in a character vector

```
install.packages(c("ggplot2", "lattice", "devtools"))
```


Installing R Packages in RStudio



Installing R Packages in RStudio



Installing an R Package from Bioconductor Project

- To install core packages, type the following in an R command window:

```
if (!requireNamespace("BiocManager"))
    install.packages("BiocManager")
BiocManager::install()
```

- To install specific packages, e.g., “GenomicFeatures” and “AnnotationDbi”, with

```
BiocManager::install(c("GenomicFeatures", "AnnotationDbi"))
```

http://bioconductor.org/install/

Installing an R Package from Bioconductor Project

- Installing a package does not make it immediately available to you in R; you must load the package
- The `library()` function is used to load packages into R
- The following code is used to load the ggplot3 package into R

```
libary(ggplot2)
```

Note: Do not put the package name in quotes!

Some R Resources

Available from CRAN (https://cran.r-project.org/)

- An Introduction to R
- Writing R Extensions
- R Data Import/Export
- R Installation and Administration (mostly for building R from sources)
- R Internals (not for the faint of heart)

Some Useful Books on S/R

Standard Texts

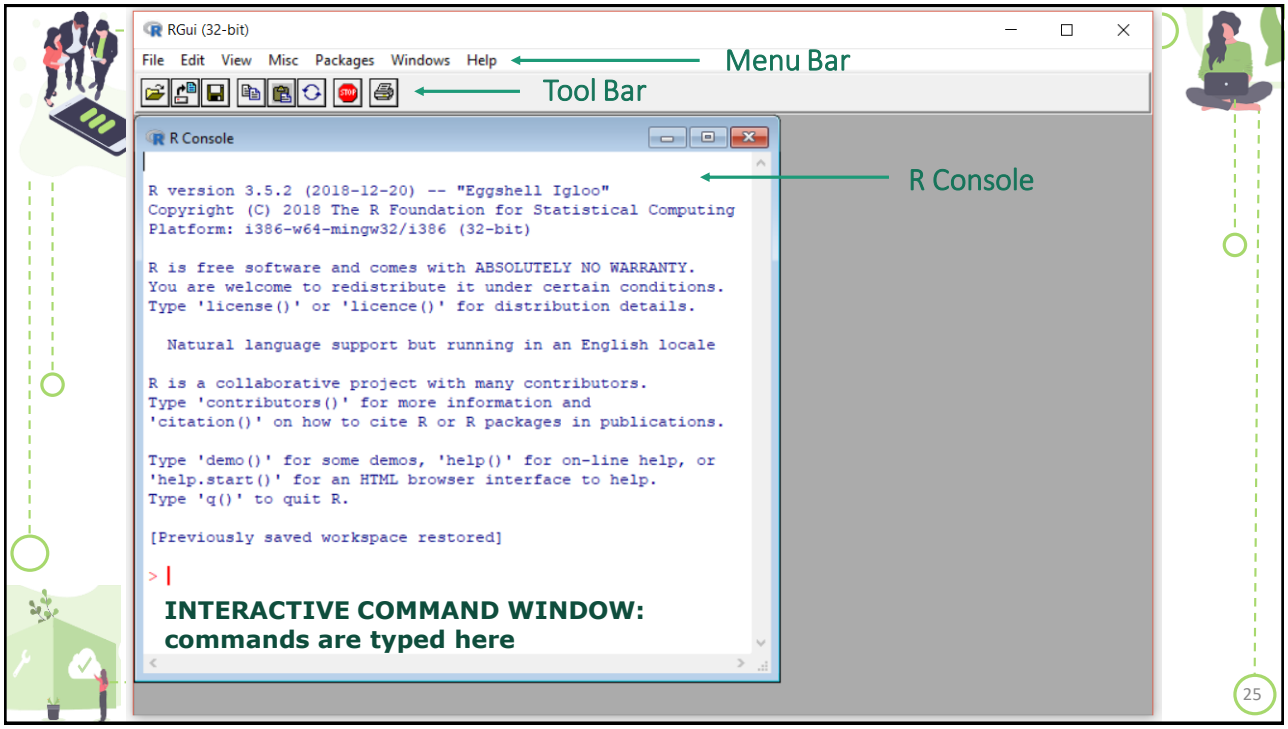
- Chambers (2008). *Software for Data Analysis*, Springer. (your textbook)
- Chambers (1998). *Programming with Data*, Springer.
- Venables & Ripley (2002). *Modern Applied Statistics with S*, Springer.
- Venables & Ripley (2000). *S Programming*, Springer.
- Pinheiro & Bates (2000). *Mixed-Effects Models in S and S-PLUS*, Springer.
- Murrell (2005). *R Graphics*, Chapman & Hall/CRC Press.

Other Resources:

- A longer list of books is at http://www.r-project.org/doc/bib/R-books.html

R Environment

- Start-up window
  - Menu bar
  - Tool bar
  - Command window or R console
- Graphics device
- R codes
  - Expression language with simple syntax
  - Case-sensitive
  - Use of text editor (e.g. R editor, or Notepad, but not MS Word) for ease of management of codes
  - Always save R codes in a text editor rather than saving your work space



25

Setting up R

- To change the default working directory:
  setwd(file="drive:/path/folder/")
- To access help files for a function:
  ?function
- To access help files for a text:
  ??text or ??"text phrase"
- To view an example:
  example(function)
- To add comments:
  #commentline

Prepared By: R. Daynolo

26

26

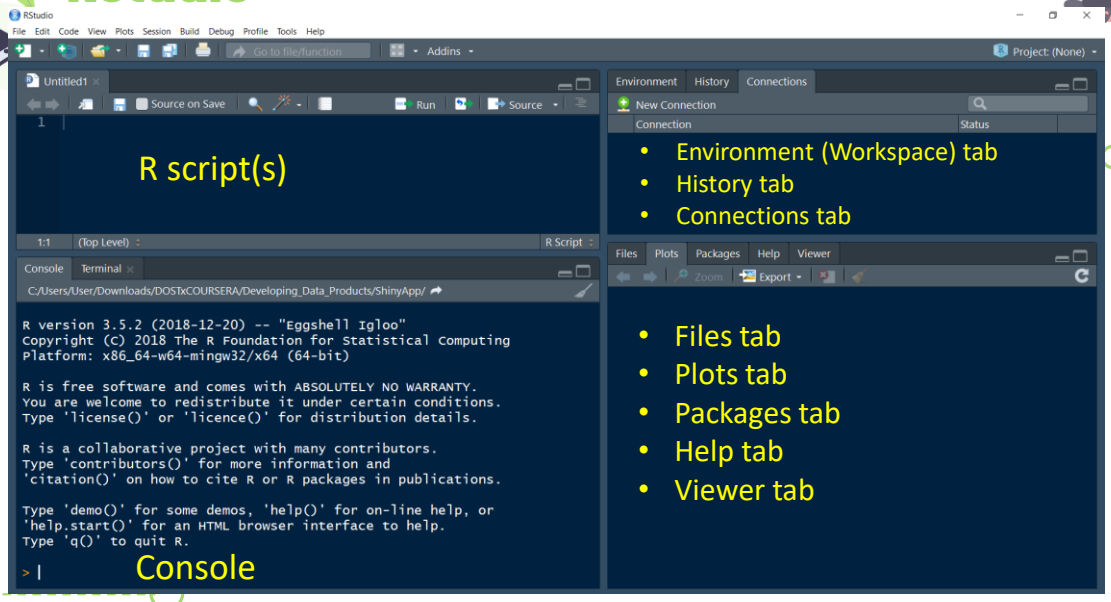
Interfaces

- For Windows and MAC OS, the standard R download comes with an R GUI, which is adequate for simple tasks
- RStudio. Very popular, with a nice interface and well thought out, especially for more advanced usage: can be a bit buggy, so make sure you update it regularly. Available in all platforms.

RStudio

- RStudio allows the user to run R in a more user-friendly environment. It is open-source (i.e. free) and available at http://www.rstudio.com/

RStudio



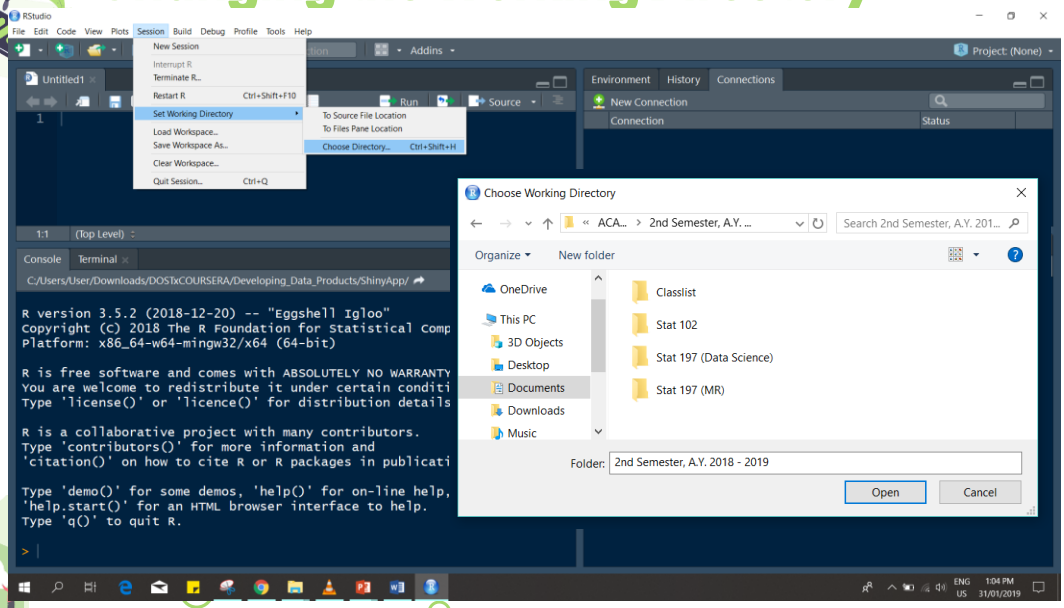
Changing the Working Directory

- If you have different projects you can change the working directory for that session from the menu bar

- Or you can type:

```
# Shows the working directory (wd)
    getwd()
# Changes the wd
    setwd("C:/myfolder/data")
```

Changing the Working Directory

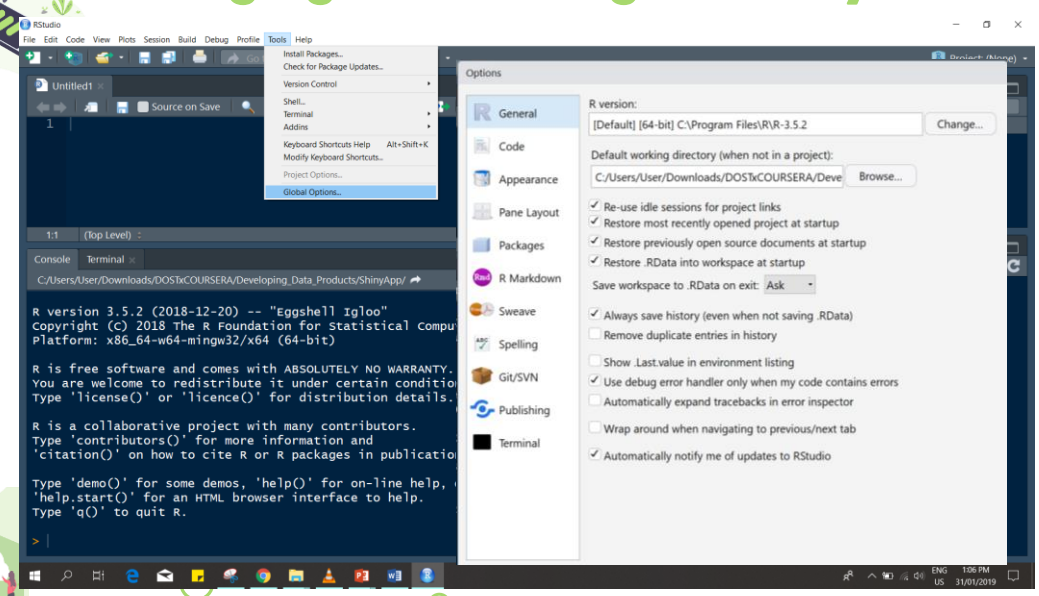


Prepared By: R. Daynolo

31

31

Changing the Working Directory



Prepared By: R. Daynolo

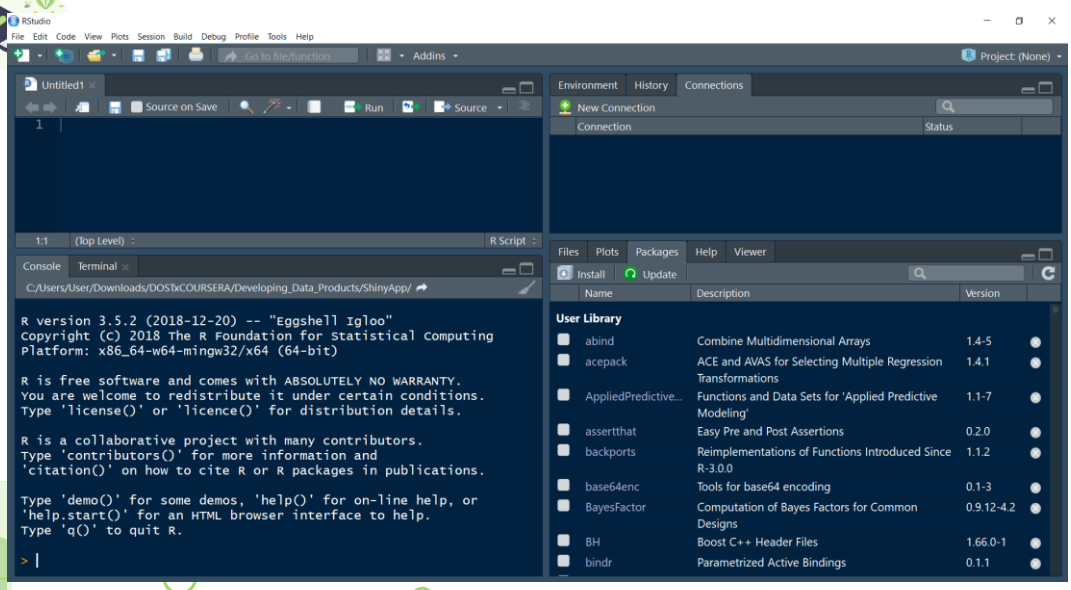
32

32

Package Tab

- The package tab shows the list of add-ons included in the installation of RStudio. If checked, the package is loaded in R, if not, any command related to that package won't work, you will need to select it. You may also install other add-ons by clicking on the Install Packages icon.
- Another way to activate a package is by typing, for example, library(foreign). This will automatically check the –foreign package (it helps bring data from proprietary formats like Stata, SAS, or SPSS)

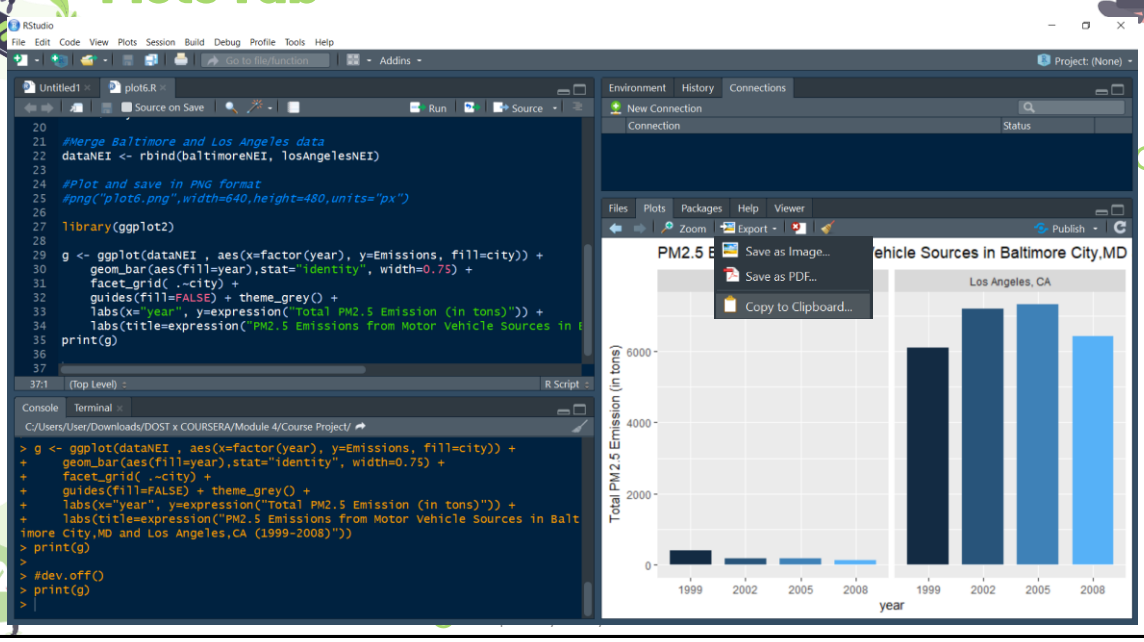
Package Tab



Plots Tab

- The plots tab will display the graphs.
- To extract the graph, click on Export where you can save the file a an image (PNG, JPG, etc.) or as PDF, these options are useful when you only want to share the graph. Probably the easiest way to export a graph is by copying it to the clipboard and then paste it directly into your Word document.

Plots Tab



Help and Documentation

- There is a large amount of (free) documentation and help available. Some help is automatically installed. Typing in the console window the command
  > `help(rnorm)`
- gives help on the `rnorm()` function. It gives a description of the function, possible arguments and the values that are used as default for optional arguments. Typing
  > `example(rnorm)`
- gives some examples of how the function can be used

R Scripts

- R is an interpreter that uses a command line based environment.
- This means that you have to type commands, rather than use the mouse and menus.
- This has an advantage that you do not always have to retype all commands.
- You can store your commands in files, the so-called scripts.
- These scripts have typically file names with extension .R , e.g. foo.R.

R Scripts

- You can run (send to the console window) part of the code by selecting the lines and pressing CTRL + ENTER or click Run in the editor window.
- If you do not select anything, R will run the line your cursor is on.
- You can always run the whole script with the console command source, so e.g. for the script in the file foo.R you type:

```
> source("foo.R")
```

- You can also click Run all in the editor window or type CTRL+SHIFT+S to run the whole script at once.

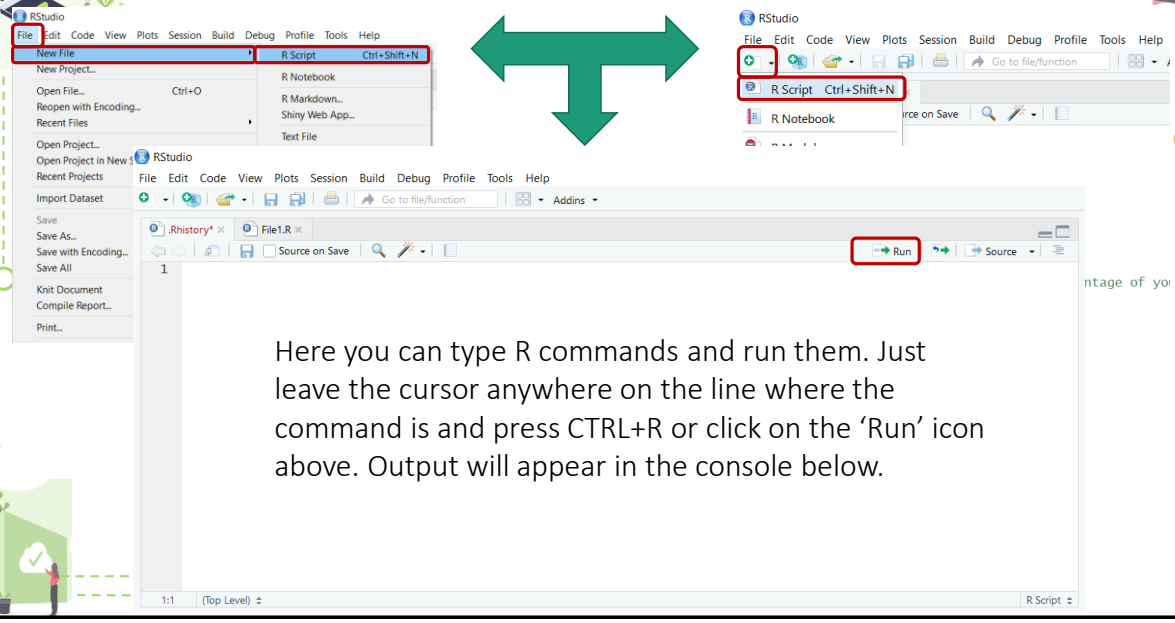
39

R Scripts

- To create a new R script you can either go to File > New R Script, or click on the icon with the "+" sign and selct "R Script", or simply press CRTL+SHIFT+N.
- Make sure to save the script.

40

R Scripts



Here you can type R commands and run them. Just leave the cursor anywhere on the line where the command is and press CTRL+R or click on the 'Run' icon above. Output will appear in the console below.

41

41

Three Ways of Quitting from R Session

1. Enter in Command Window:
   `> quit()`
2. Click on
   File ► Exit
3. Click on Close button (X at upper right hand corner of R console window).

Prepared By: R. Daynolo

42

42