

Introduction to Rstudio

Anders K. Krabberød

a.k.krabberod@ibv.uio.no

https://github.com/krabberod/UNIS_AB332_2023

UNIS - AB332 - 2023

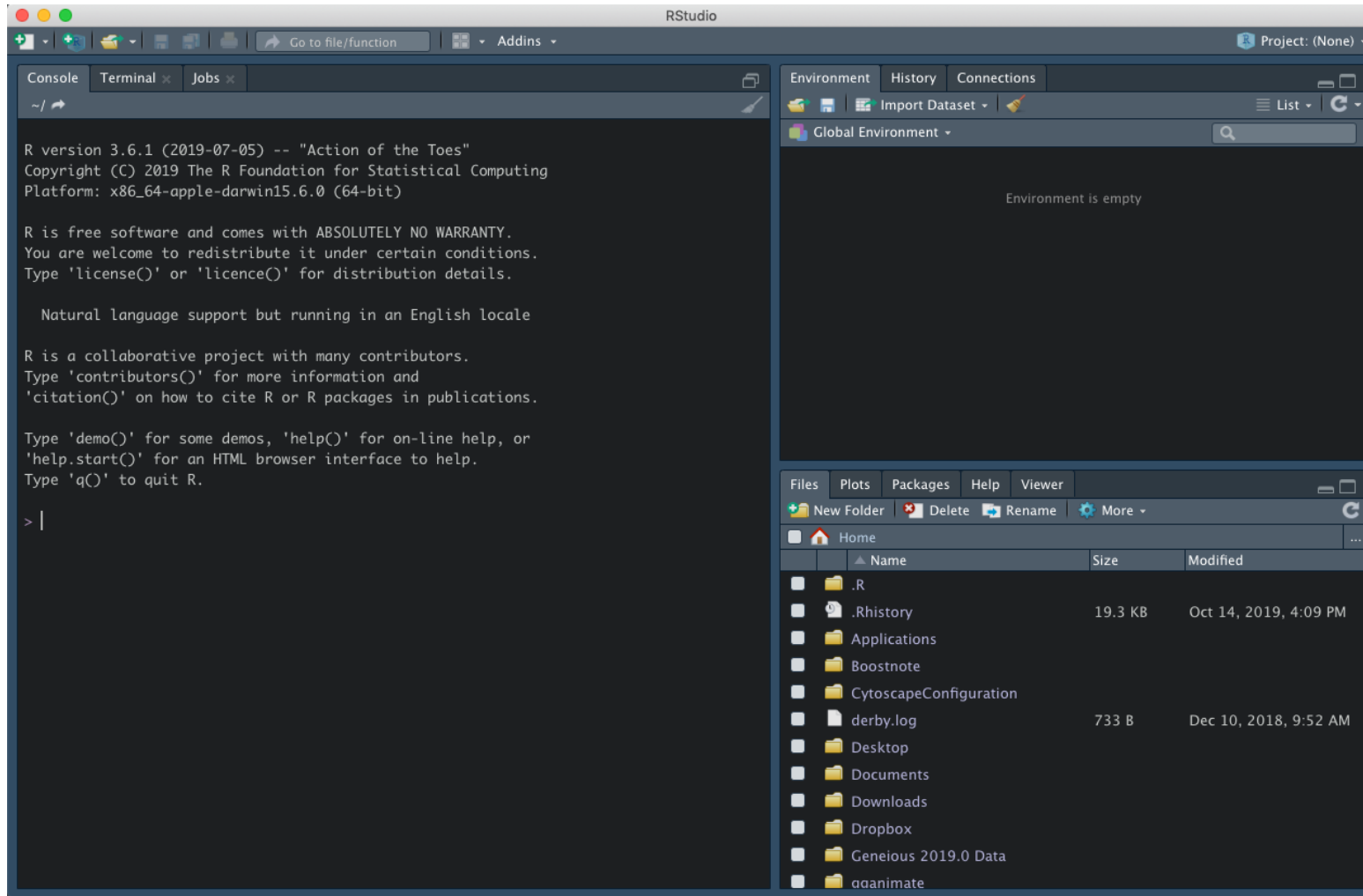


Rstudio

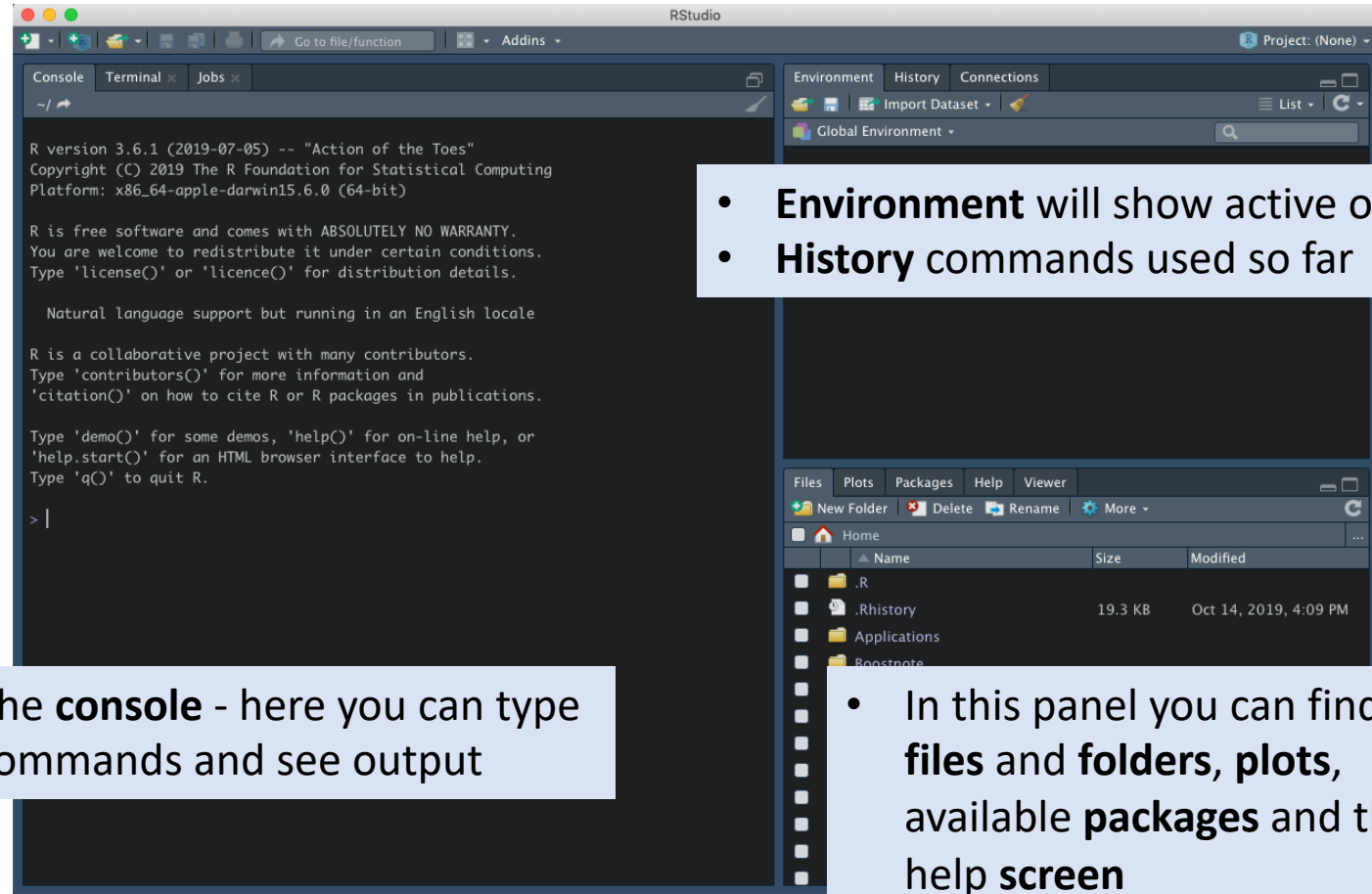
- RStudio is an Integrated Development Environment (IDE) for R, a programming language for statistical computing and graphics.
- Customizable workbench with all of the tools required to work with R in one place (console, source, plots, workspace, help, history, etc.).
- Syntax highlighting editor with code completion.
- Execute code directly from the source editor (line, selection, or file).
- Runs on Windows, Mac, and Linux, and has a community-maintained FreeBSD port.
- Can also be run as a server, enabling multiple users to access the RStudio IDE using a web browser.
- (Source <https://github.com/rstudio/rstudio>)



Rstudio - Graphical interface



Rstudio



- **Environment** will show active objects
- **History** commands used so far

The **console** - here you can type commands and see output

- In this panel you can find **files** and **folders**, **plots**, available **packages** and the **help screen**



RStudio

Go to file/function Addins Project: (None)

Source

Console Terminal x Jobs x

~/Documents/Dada2_workshop/

```
> db<-as.data.frame(matrix(c(1,2,3,4,5,6,7,8), ncol=4,nrow=2))
> db
  V1 V2 V3 V4
1  1  3  5  7
2  2  4  6  8
>
>
>
>
>
>
>
>
```

- **Creating** an object in the console

Environment History Connections

Import Dataset List

Global Environment

Data

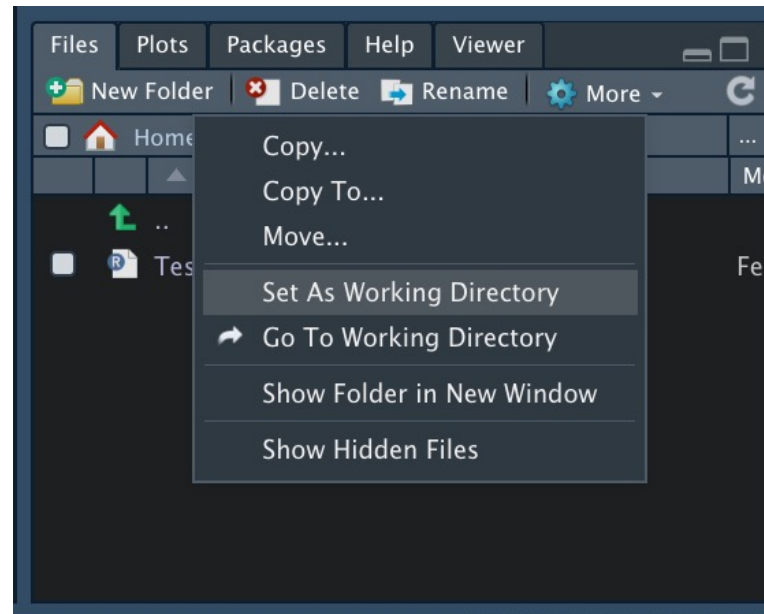
db 2 obs. of 4 variables

- **Environment** shows the object



Setting working directory

- Navigate to correct folder under the “files” tab
- Click “Set As Working Directory” (under *More*)



Setting working directory

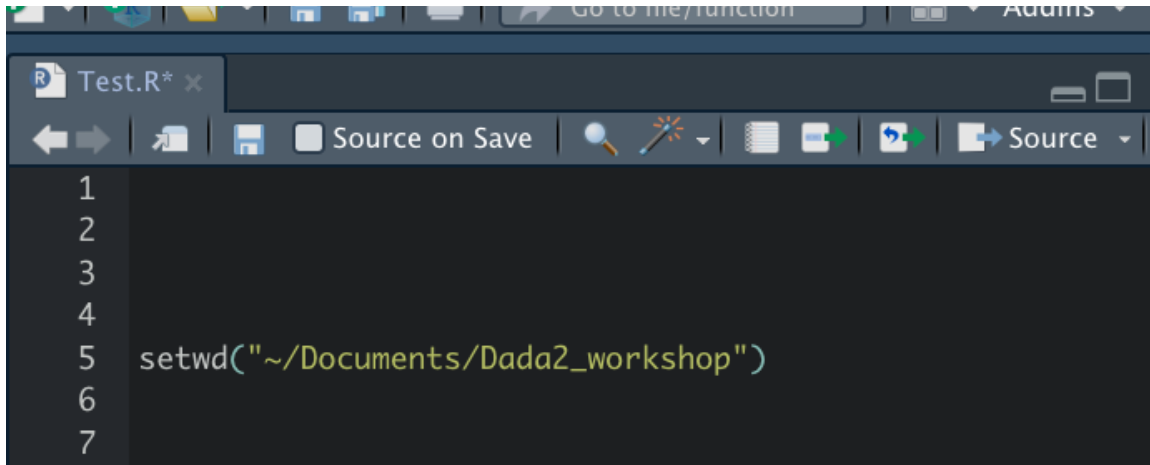
- Alternatively write

MAC:

```
setwd("~/path/to/my/folder")
```

WINDOWS

```
setwd("C:/path/to/my/folder")
```

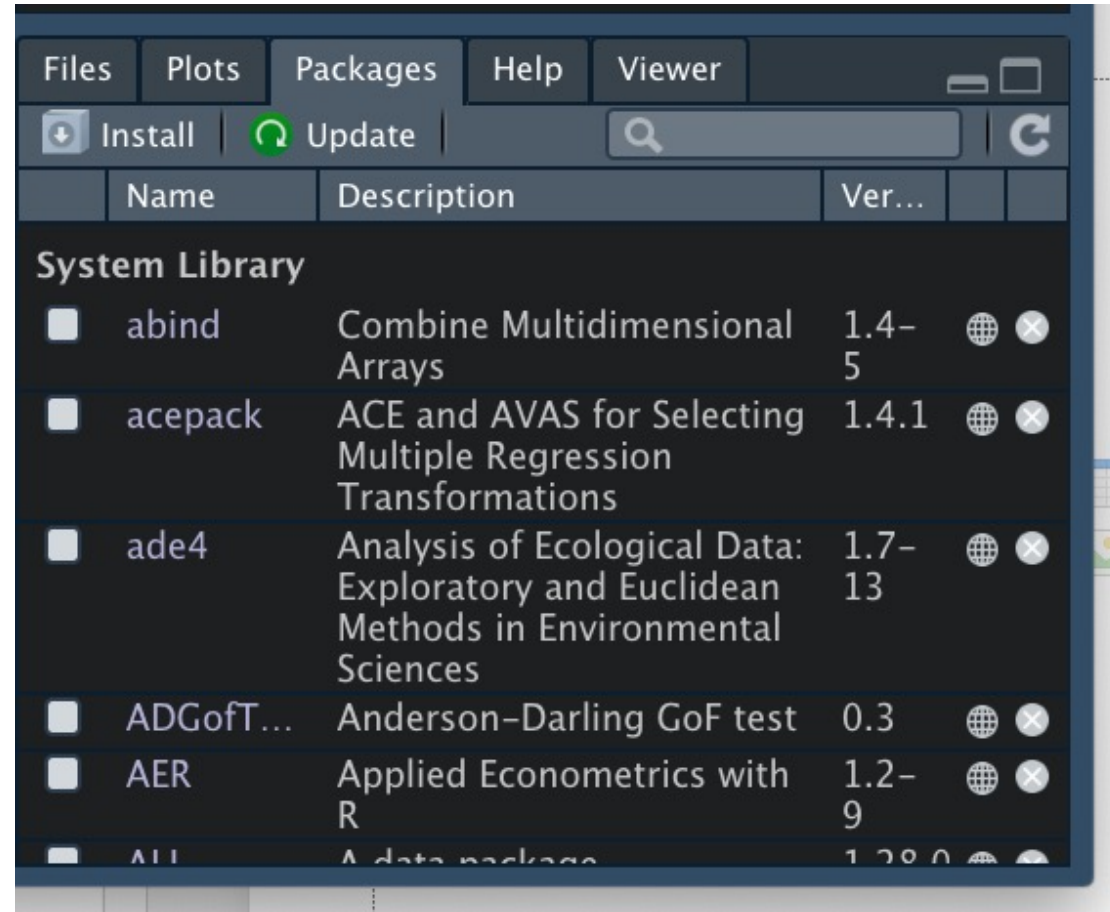


The screenshot shows the RStudio interface with a script editor open. The file is named 'Test.R*'. The script contains the following code:

```
1  
2  
3  
4  
5 setwd("~/Documents/Dada2_workshop")  
6  
7
```

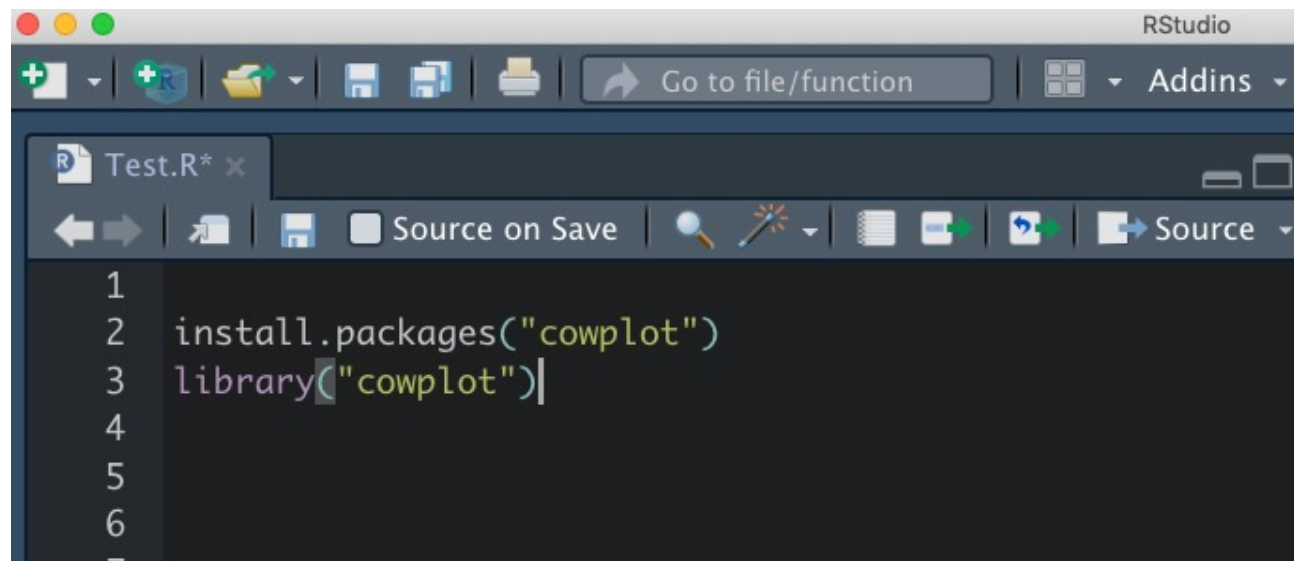


Installing packages



Installing packages 2

- Or use the command (with cowplot as example)
 - `install.packages("cowplot")`
- Installed packages can be loaded with the command
 - `library("cowplot")`



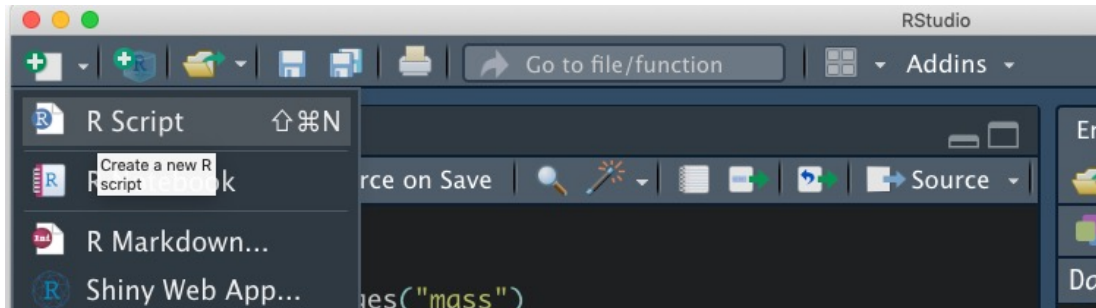
The screenshot shows the RStudio interface. The top toolbar includes icons for file operations and a search bar. The editor window, titled 'Test.R*', contains the following R code:

```
1  
2 install.packages("cowplot")  
3 library("cowplot")  
4  
5  
6
```

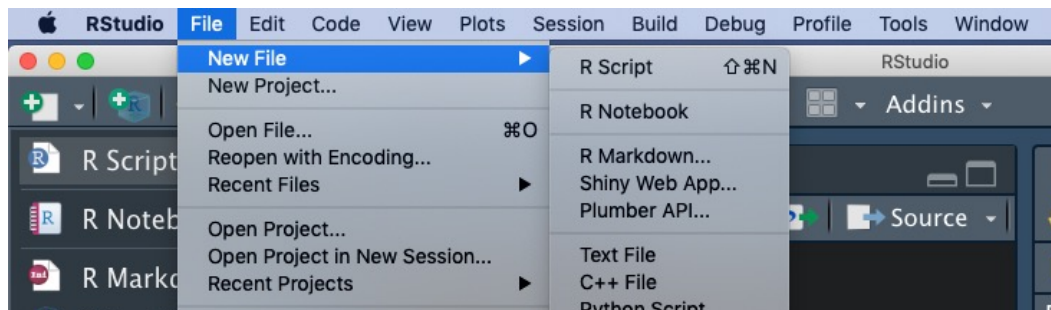


Using Scripts

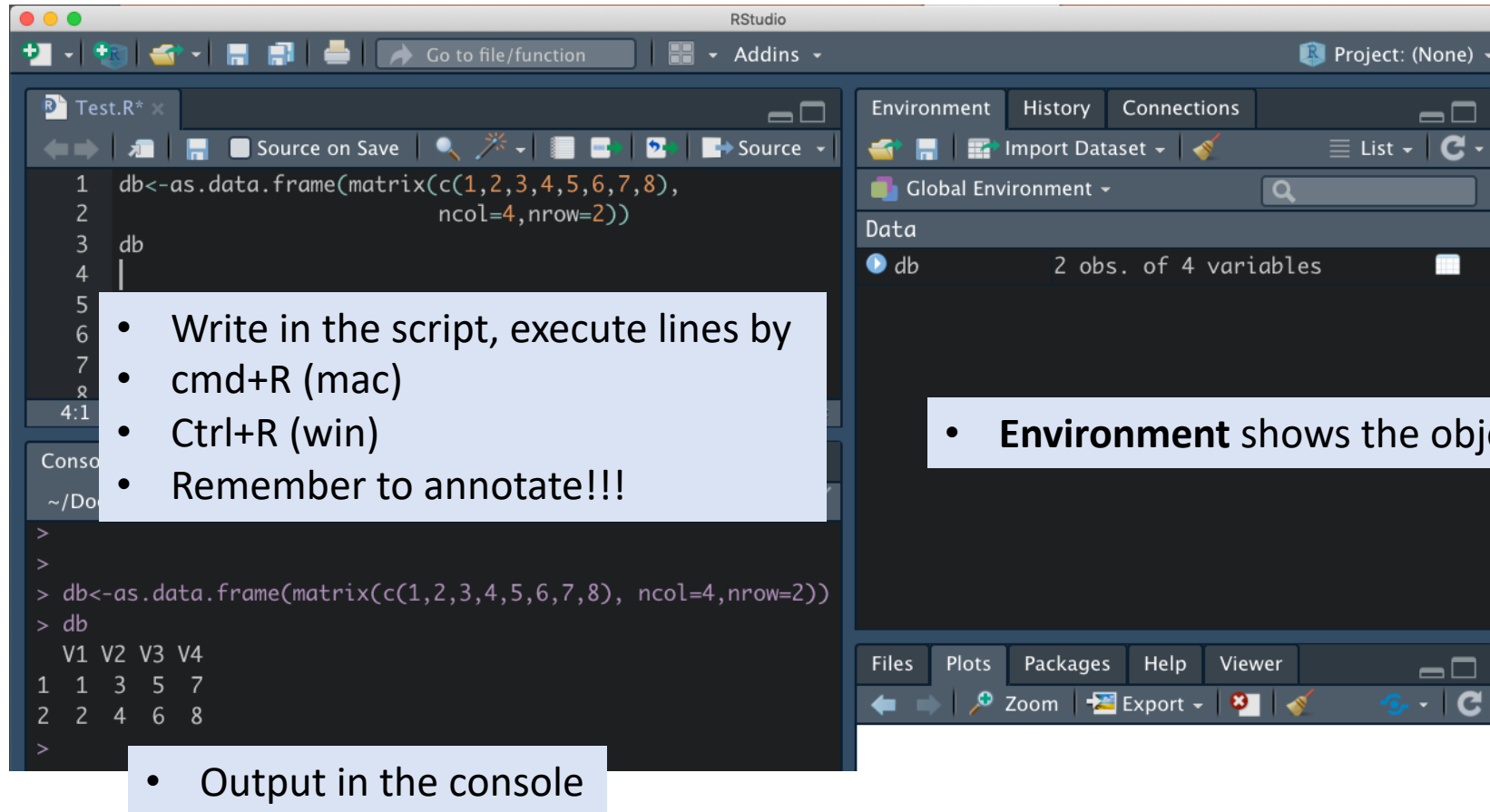
Click icon with a document and a + sign



OR click File -> New File -> R Script



Using Scripts



The screenshot shows the RStudio interface with a script named 'Test.R' open in the Source pane. The script contains the following R code:

```
1 db<-as.data.frame(matrix(c(1,2,3,4,5,6,7,8),
2                             ncol=4,nrow=2))
3 db
4 |
5
6
7
8
9 4:1
```

The Environment pane on the right shows the object 'db' with 2 observations and 4 variables. The console at the bottom shows the output of the script:

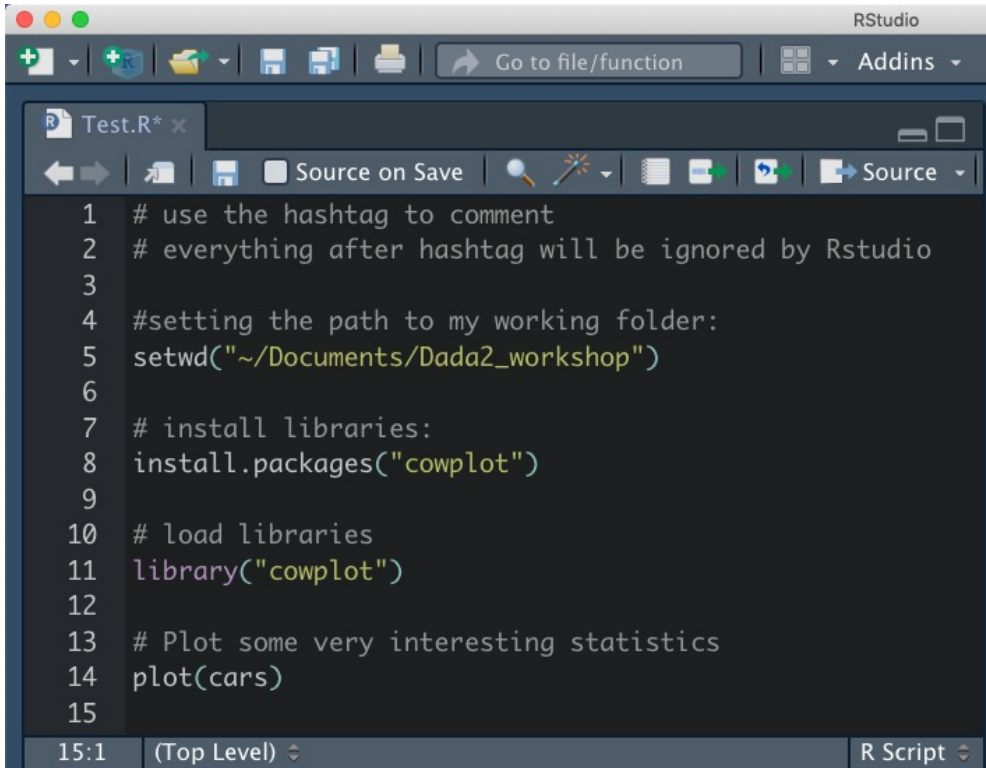
```
> db<-as.data.frame(matrix(c(1,2,3,4,5,6,7,8), ncol=4,nrow=2))
> db
  V1 V2 V3 V4
1  1  3  5  7
2  2  4  6  8
>
```

Annotations in the image highlight key aspects of using scripts:

- Write in the script, execute lines by
• cmd+R (mac)
• Ctrl+R (win)
• Remember to annotate!!!
- **Environment** shows the object
- Output in the console



Comment and annotate your script!!!



```
1 # use the hashtag to comment
2 # everything after hashtag will be ignored by Rstudio
3
4 #setting the path to my working folder:
5 setwd("~/Documents/Dada2_workshop")
6
7 # install libraries:
8 install.packages("cowplot")
9
10 # load libraries
11 library("cowplot")
12
13 # Plot some very interesting statistics
14 plot(cars)
15
```

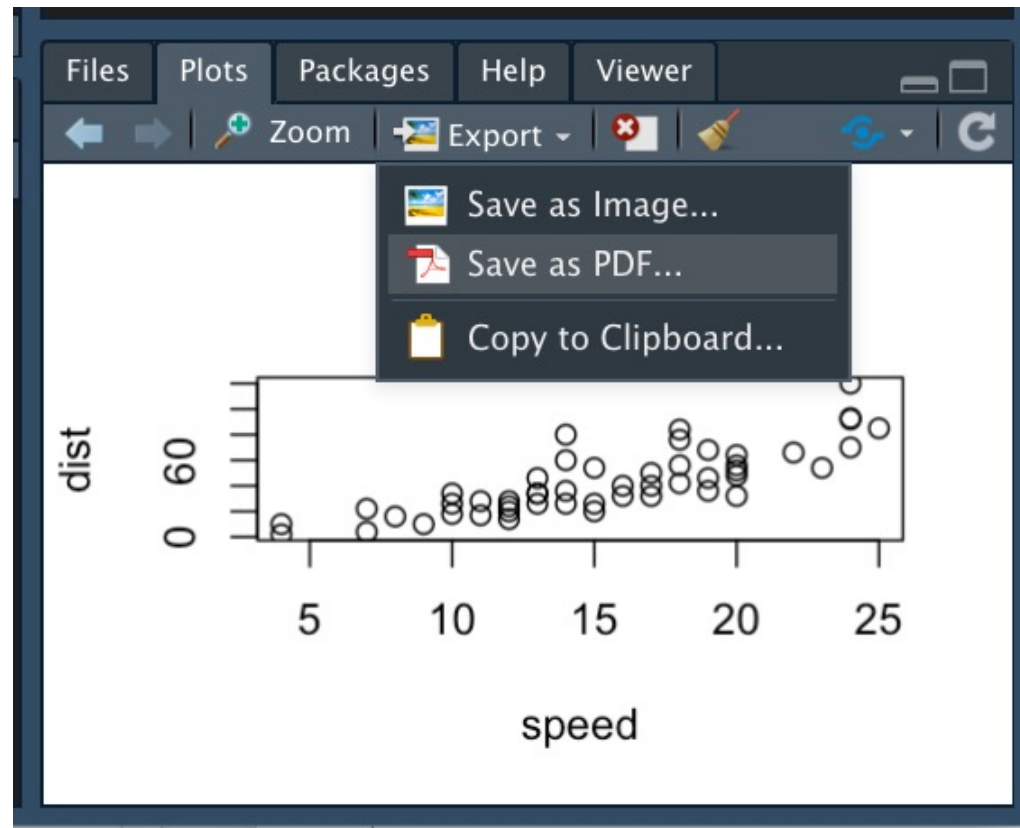
The screenshot shows the RStudio interface with a source editor window titled 'Test.R*'. The editor contains 15 lines of R code. Lines 1-3 are comments explaining the use of the hashtag. Lines 4-6 show the working directory being set. Lines 7-9 show the installation of the 'cowplot' package. Lines 10-12 show the loading of the 'cowplot' library. Lines 13-15 show the plotting of 'cars' data. The status bar at the bottom indicates '15:1 (Top Level)' and 'R Script'.

- What the code does
- How the code does it
- How to use the code



Plotting plots and other dots

- Plots will appear in the *plots* tab and can be exported in various formats

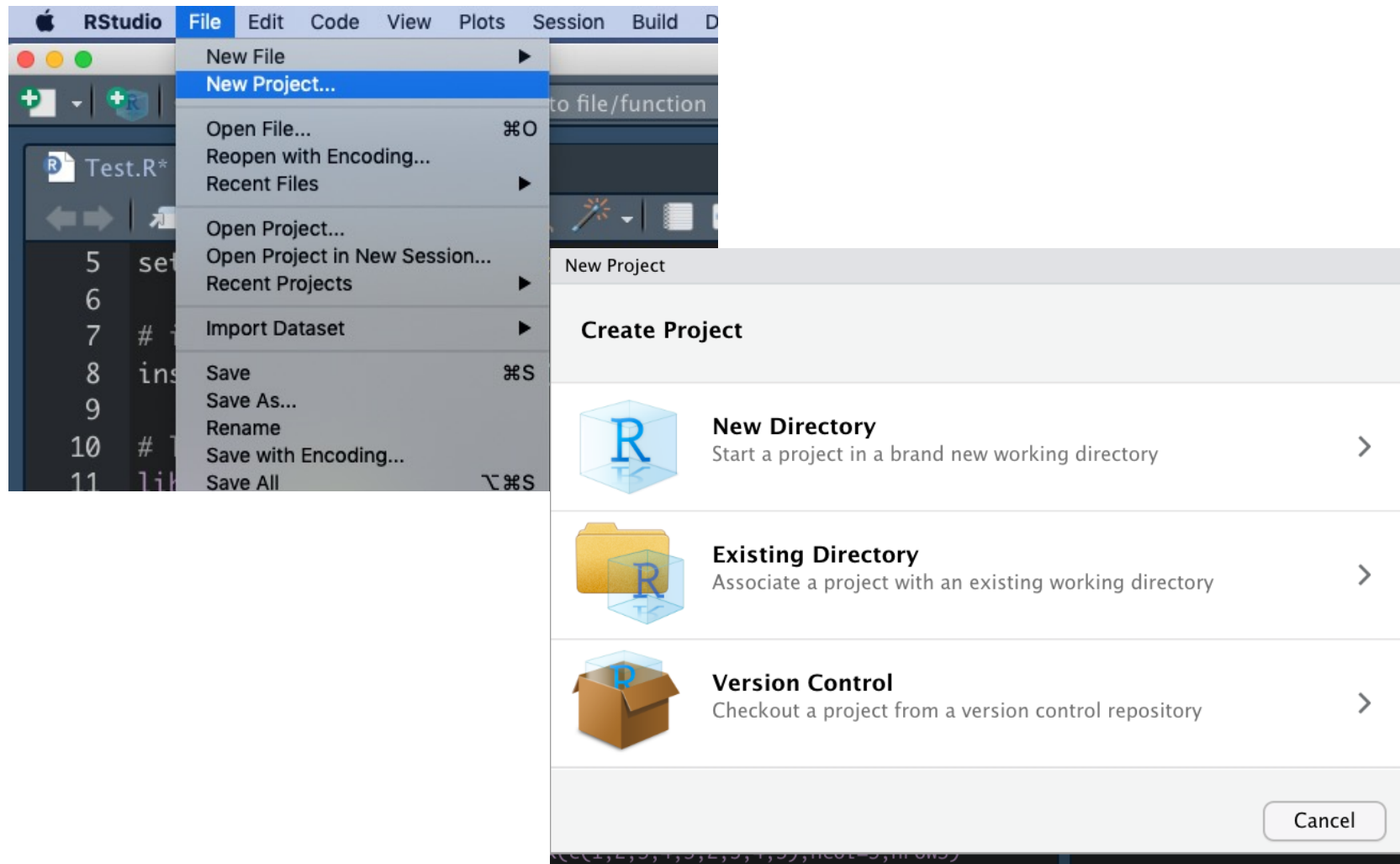


Use R-projects!

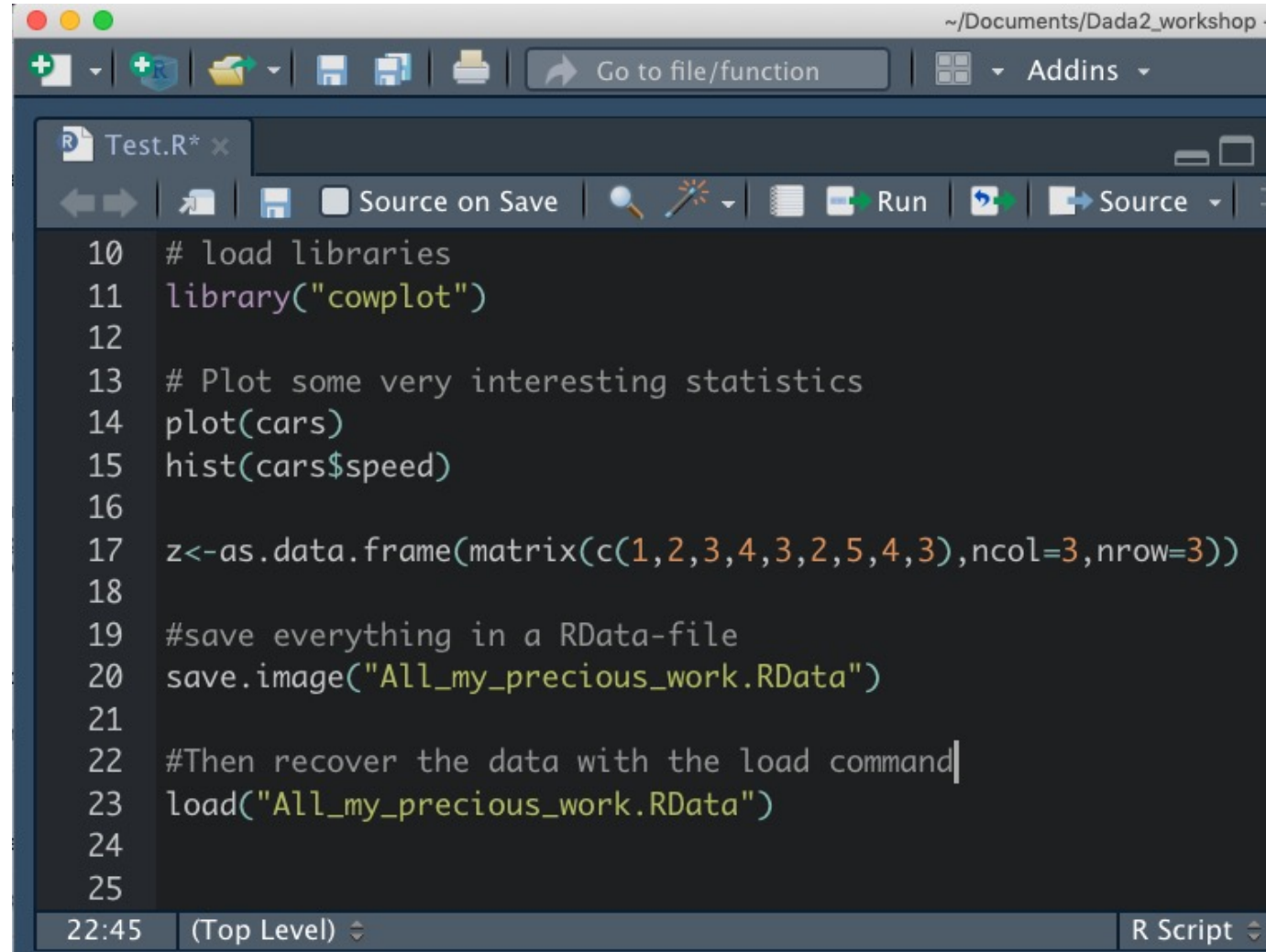
- This will set the default working directory for the particular project, and makes it easy to save everything in the same folder.
- Very helpful when working on several different projects
- Also very easy to integrate with *github* and version control with the option to push and pull repositories (not covered in this lecture)
- Or for sharing all data with somebody else using RStudio



Use R-projects



Use R-projects



```
10 # load libraries
11 library("cowplot")
12
13 # Plot some very interesting statistics
14 plot(cars)
15 hist(cars$speed)
16
17 z<-as.data.frame(matrix(c(1,2,3,4,3,2,5,4,3),ncol=3,nrow=3))
18
19 #save everything in a RData-file
20 save.image("All_my_precious_work.RData")
21
22 #Then recover the data with the load command
23 load("All_my_precious_work.RData")
24
25
```

The screenshot shows the RStudio interface. The top toolbar includes icons for file operations and a search bar. The editor pane displays the R script 'Test.R*' with line numbers 10 through 25. The script contains comments and code for loading the 'cowplot' library, plotting 'cars' data, creating a histogram of 'cars\$speed', saving the current session to 'All_my_precious_work.RData', and then loading it back. The status bar at the bottom shows the time '22:45', the environment '(Top Level)', and the file type 'R Script'.



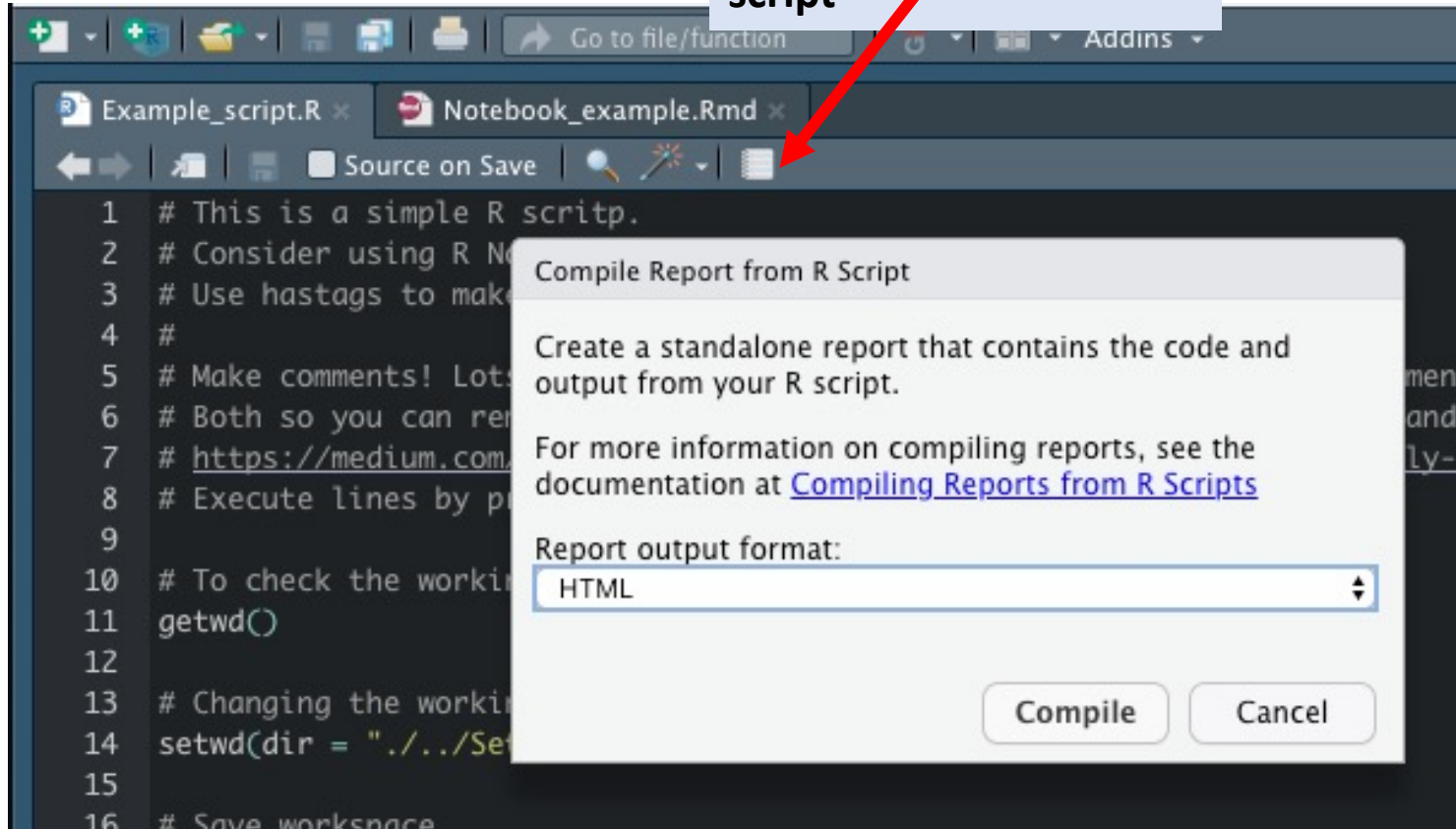
R Markdown and R notebooks

- An alternative to “simple” script in Rstudio.
- Advantage: easy to export in other easy-to-read formats (i.e. html, pdf, word, presentations).
- Markdown language is an easy way of formatting using plain text
- R Notebook is somewhat more powerful with additional options for formatting.
- Can run chunks of code from other languages *within* Rstudio
- **Disadvantage:** Not compatible with (standalone aka. vanilla) R, which is often used on clusters and servers.

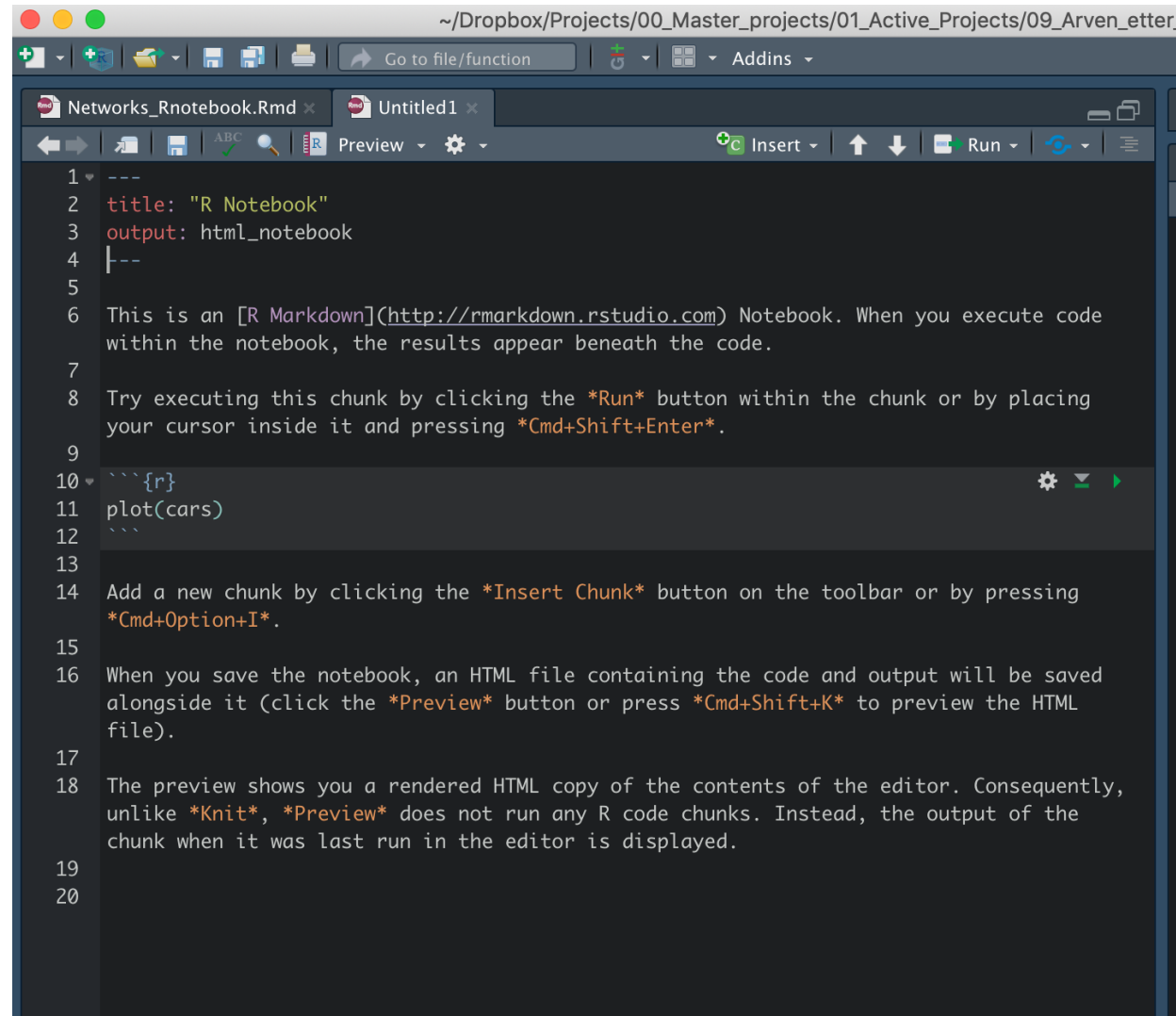


Export a report

This button will help to generate a pdf, html, or word document of your script



R Notebook



```
1 ---
2 title: "R Notebook"
3 output: html_notebook
4 |---
5
6 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code
7 within the notebook, the results appear beneath the code.
8
9 Try executing this chunk by clicking the *Run* button within the chunk or by placing
10 your cursor inside it and pressing *Cmd+Shift+Enter*.
11
12 ```{r}
13 plot(cars)
14 ```
15
16 Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing
17 *Cmd+Option+I*.
18
19 When you save the notebook, an HTML file containing the code and output will be saved
20 alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML
21 file).
```

The screenshot shows a web browser window with the address bar displaying a path to a project directory. The browser tab is titled 'Networks_Rnotebook.Rmd'. The notebook interface includes a toolbar with buttons for 'Go to file/function', 'Addins', 'Insert', 'Run', and 'Preview'. The main content area displays the R Markdown code as shown in the pre block. The code includes a title, output format, and a code chunk for plotting cars. The notebook also contains explanatory text about R Markdown and the 'Run' button.

