

Introduction to RStudio

Mikhail Dozmorov

Summer 2018

What is it?

- **R** is the underlying statistical computing environment, but using R alone is no fun
- **RStudio** is a graphical integrated development environment that makes using R much easier.
 - RStudio offers some GUI (graphical user interface) features for R
 - The same R program is running underneath the interface
- RStudio is open-source software (and therefore is free)
 - Available for download at: <http://rstudio.com>

Panes in RStudio

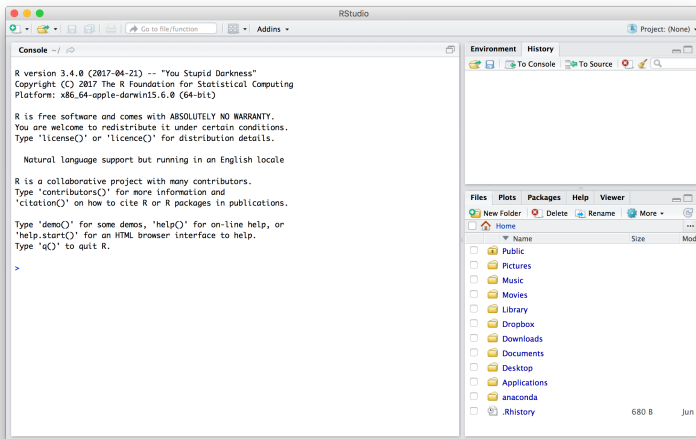
- There are four panes, and their orientation is configurable under “Tools – Global Options.” Defaults are:
 - Editor in the top left
 - Console bottom left
 - Environment/history on the top right
 - Plots/help on the bottom right.

Projects

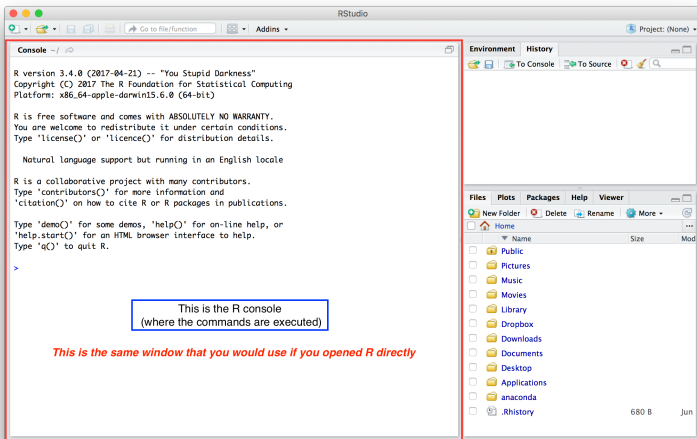
- First, start a new project in a new folder somewhere easy to remember
 - Keep project code and data in the same place
- Creating a project creates an `Rproj` file that opens R running *in that folder*. This way, when you want to read in dataset *whatever.txt*, you just tell it the filename rather than a full path. This is critical for reproducibility

RStudio

Open RStudio and this is what you will see:

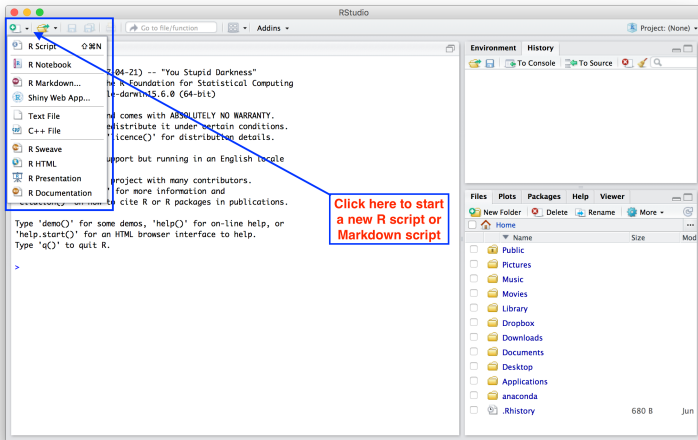


Outlined in red is the R Console:



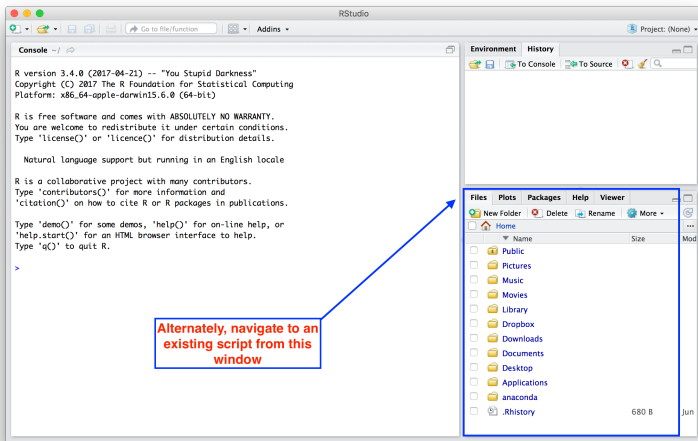
RStudio

You can open a new script in the editing window:

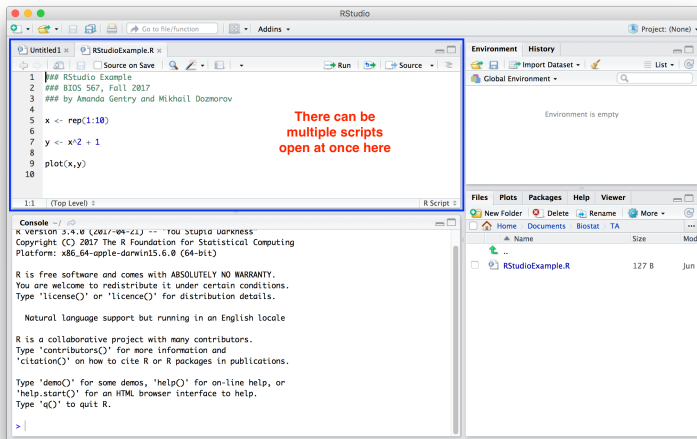


RStudio

Or you can open an existing script:



There can be multiple scripts open in the editing window:



History tab shows command run:

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains R code for a simple example:

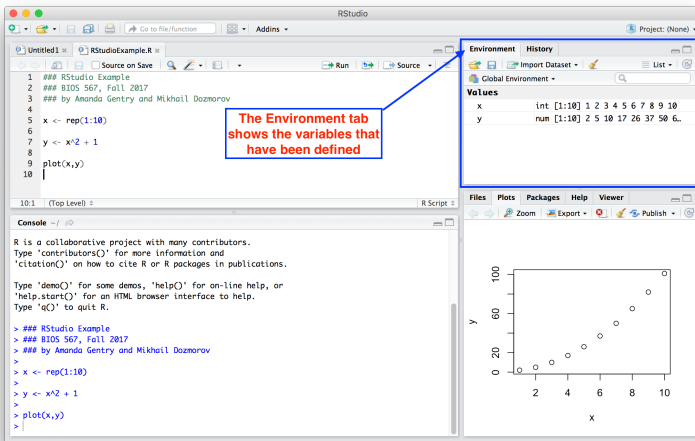
```
1 ## RStudio Example
2 ## BIOS 567, Fall 2017
3 ## by Amanda Gentry and Mikhail Dozmorov
4
5 x <- rep(1:10)
6
7 y <- x^2 + 1
8
9 plot(x,y)
10 |
```
- Console:** Shows the execution of the code and help text for the 'demo()' function:

```
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.

> ## RStudio Example
> ## BIOS 567, Fall 2017
> ## by Amanda Gentry and Mikhail Dozmorov
>
> x <- rep(1:10)
>
> y <- x^2 + 1
>
> plot(x,y)
> |
```
- History Tab:** A blue box highlights this tab, which contains the same code as the source editor, indicating the commands that have been run.
- Plots Panel:** Displays a scatter plot of the data generated by the code. The x-axis is labeled 'x' and ranges from 0 to 10. The y-axis is labeled 'y' and ranges from 0 to 100. The plot shows a clear upward trend, representing the equation $y = x^2 + 1$.

Environment tab keeps track of the things the user defines:



The screenshot shows the RStudio interface with the Environment tab selected. A blue box highlights the Environment tab, and a red text box with a blue arrow pointing to it contains the text: "The Environment tab shows the variables that have been defined".

The Environment tab displays the following variables:

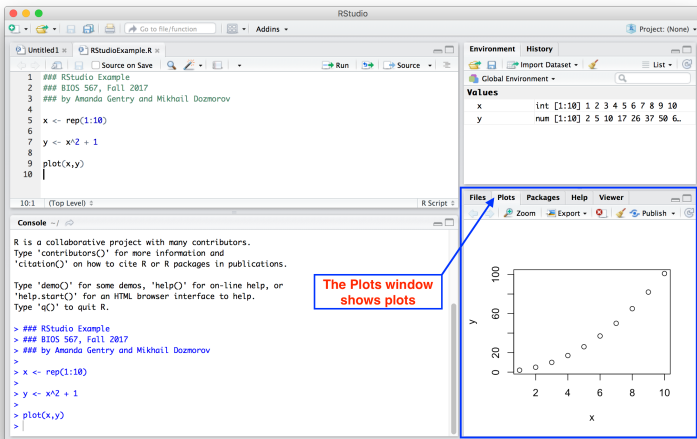
Variable	Type	Value
x	int [1:10]	1 2 3 4 5 6 7 8 9 10
y	num [1:10]	2.5 10 17 26 37 50 65

The Console shows the following R code and output:

```
> ## RStudio Example
> ## BIOS 567, Fall 2017
> ## by Amanda Gentry and Mikhail Dozmorov
>
> x <- rep(1:10)
>
> y <- x^2 + 1
>
> plot(x,y)
>
>
```

The Plot window shows a scatter plot of y versus x. The x-axis ranges from 0 to 10, and the y-axis ranges from 0 to 100. The data points are (1, 2), (2, 5), (3, 10), (4, 17), (5, 26), (6, 37), (7, 50), (8, 65), (9, 82), and (10, 101).

Plots appear in the plots tab (as expected):



The screenshot displays the RStudio interface. The main editor window shows an R script with the following code:

```
1 ## RStudio Example
2 ## BIOS 567, Fall 2017
3 ## by Amanda Gentry and Mikhail Dozmorov
4
5 x <- rep(1:10)
6
7 y <- x^2 + 1
8
9 plot(x,y)
10
```

The console window shows the execution of the script:

```
> ## RStudio Example
> ## BIOS 567, Fall 2017
> ## by Amanda Gentry and Mikhail Dozmorov
>
> x <- rep(1:10)
>
> y <- x^2 + 1
>
> plot(x,y)
>
```

The Environment pane on the right shows the values of the variables:

Variable	Class	Value
x	int [1:10]	1 2 3 4 5 6 7 8 9 10
y	run [1:10]	2 5 10 17 26 37 50 65 82 101

The Plots pane on the right shows a scatter plot of y versus x. The x-axis ranges from 0 to 10, and the y-axis ranges from 0 to 100. The data points form a parabolic curve. A blue box highlights the Plots pane, and a red arrow points to it from a text box that says "The Plots window shows plots".

Getting help

- Is there a built-in function that does x ? and if so,
- Where can i find it among the +2000 R Packages in CRAN?
- Use `?function_name` to get help on a function from a *loaded* package
- Use `??function_name` to search for the function across all installed packages, even not loaded
- Use `apropos("part_of_function_name")` if you forgot the exact name, to get suggestions what the function name may be
- `library(sos)`, `findFn` function, e.g., `findFn("cosine", maxPages=2, sortby="MaxScore")`
- Search engine is your best friend on many things

RStudio keyboard shortcuts

- `Command (Ctrl) + Enter` - Run current line/selection, go to the next line
- `Alt + Enter` - Run current line/selection, stay on the current line
- `Ctrl + 1` - Move cursor to source
- `Ctrl + 2` - Move cursor to console
- `Ctrl + L` - Clear console
- `Ctrl + Alt + I` - Create new code chunk
- `Ctrl + Alt + c` - Run current code chunk
- All the usual - `Ctrl + Z, C, X, V, S, O`

And more: <https://support.rstudio.com/hc/en-us/articles/200711853-Keybaord-Shortcuts>

Turning off annoyances

- Tools, Global Options:
 - Uncheck “Restore .RData into workspace at startup”
 - Set to Never: “Save workspace to .RData on exit: Never”
 - Uncheck “Show output inline for all R Markdown documents”

RStudio summary

Why use it?

- Project-centric work - scripts and data are organized in one folder (project), easily accessible
- Single workspace with four (rearrangeable, zoomable) panels
- Work on multiple projects simultaneously in several instances of RStudio
- Work on multiple (types of) scripts (rearrangeable tabbed interface)
- See all variables in R environment, easily visualize them
- Easy access to help, plots, packages
- Simple integration with Git version control system

RStudio Tutorials

- <http://dss.princeton.edu/training/RStudio101.pdf>
- http://stcorp.nl/R_course/rstudio_tutorial.html
 - This tutorial describes RStudio's interaction with git