

ASMS 2019
ANNUAL
CONFERENCE
WORKSHOP



GETTING STARTED WITH R & RSTUDIO

BEFORE YOU START

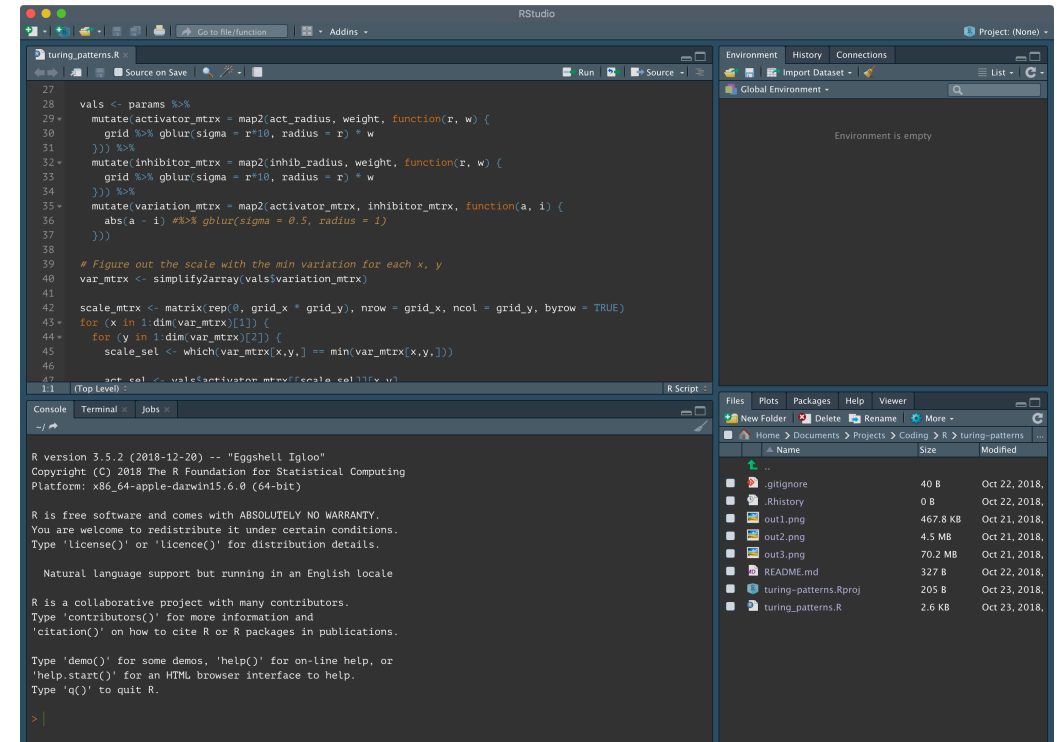
- ▶ Confirm you have R and RStudio installed and working
- ▶ For help installing R and RStudio, check out the slides:

[Installing R & RStudio](#)

R IS NOT RSTUDIO (AND VICE VERSA)



First, you'll install R
R is the "engine"



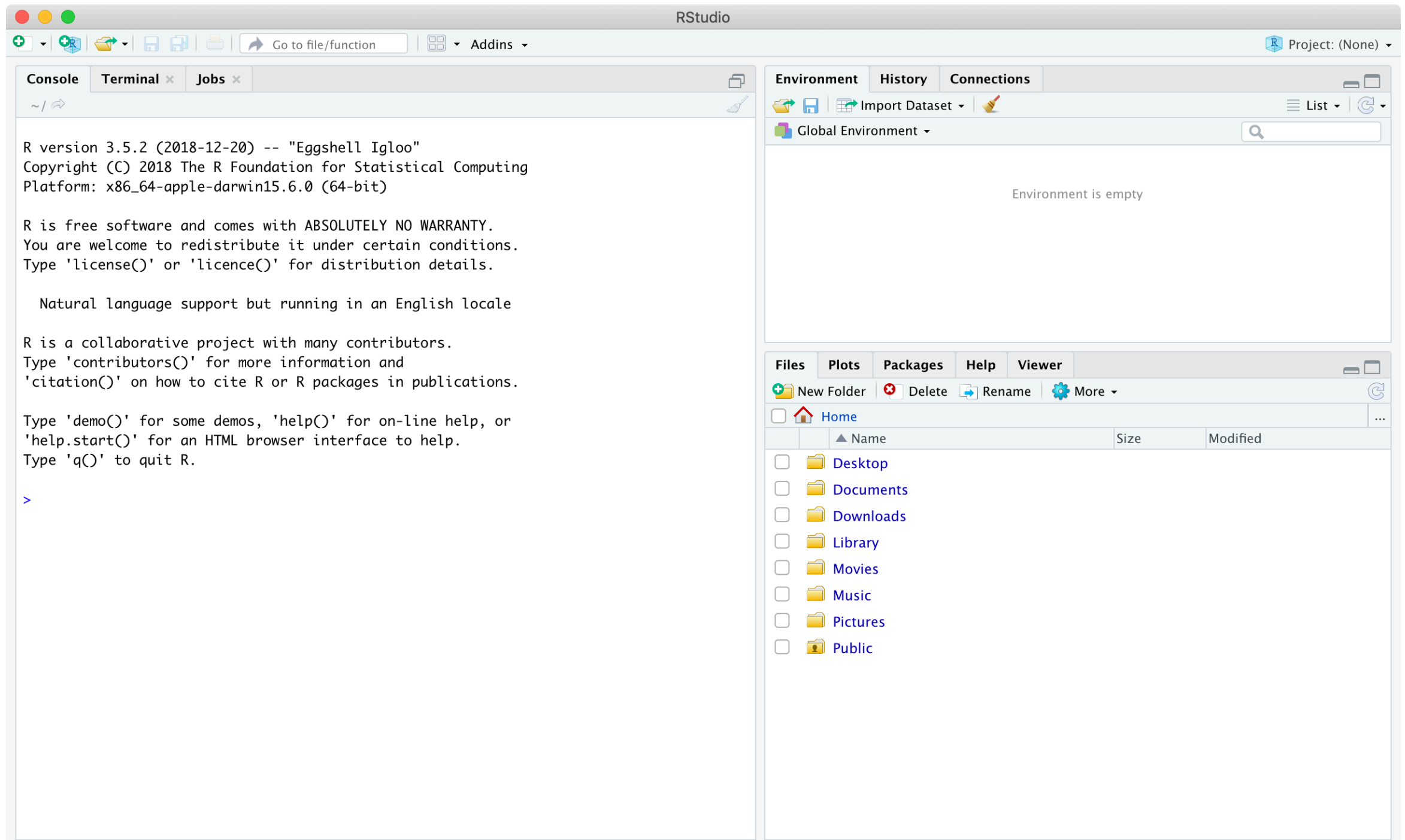
Then, you'll install RStudio
RStudio is the awesome
"car body and paint job"

There are some other fiddly bits to install too depending on the platform...

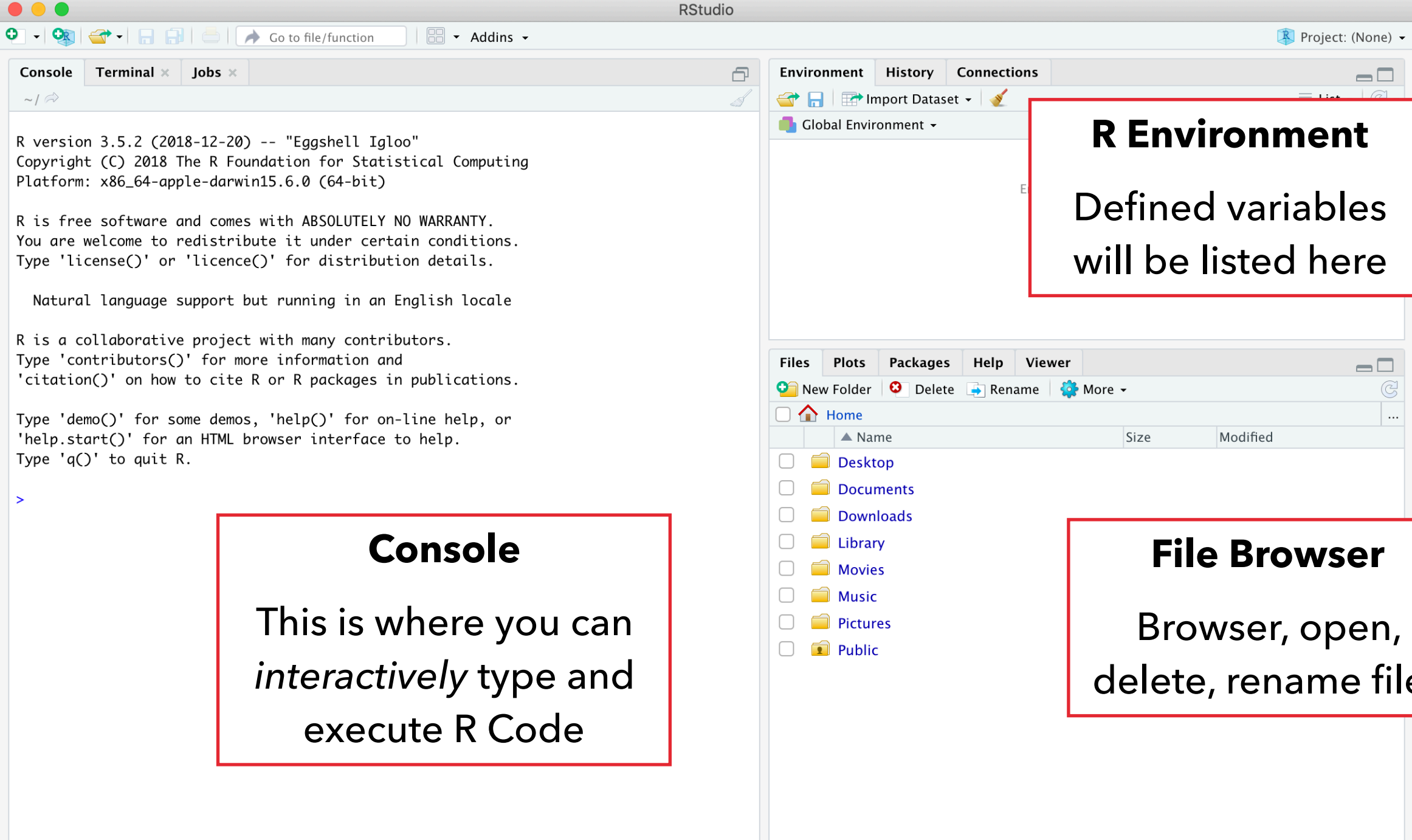
HIGH-LEVEL: WHAT YOU'LL BE DOING WHEN YOU USE R

- ▶ Writing R code to solve a problem (i.e. a data analysis task)
- ▶ Two typical ways you'll write R code
 - ▶ *Interactively* via the R console
 - ▶ *In a script file* that can then be run using the R interpreter
- ▶ One of your first hurdles will be learning the R language
 - ▶ Syntax: the set of rules and symbols that define the language
 - ▶ Getting familiar with R capabilities, functions, packages

FIRE-UP RSTUDIO — IT SHOULD LOOK SOMETHING LIKE THIS



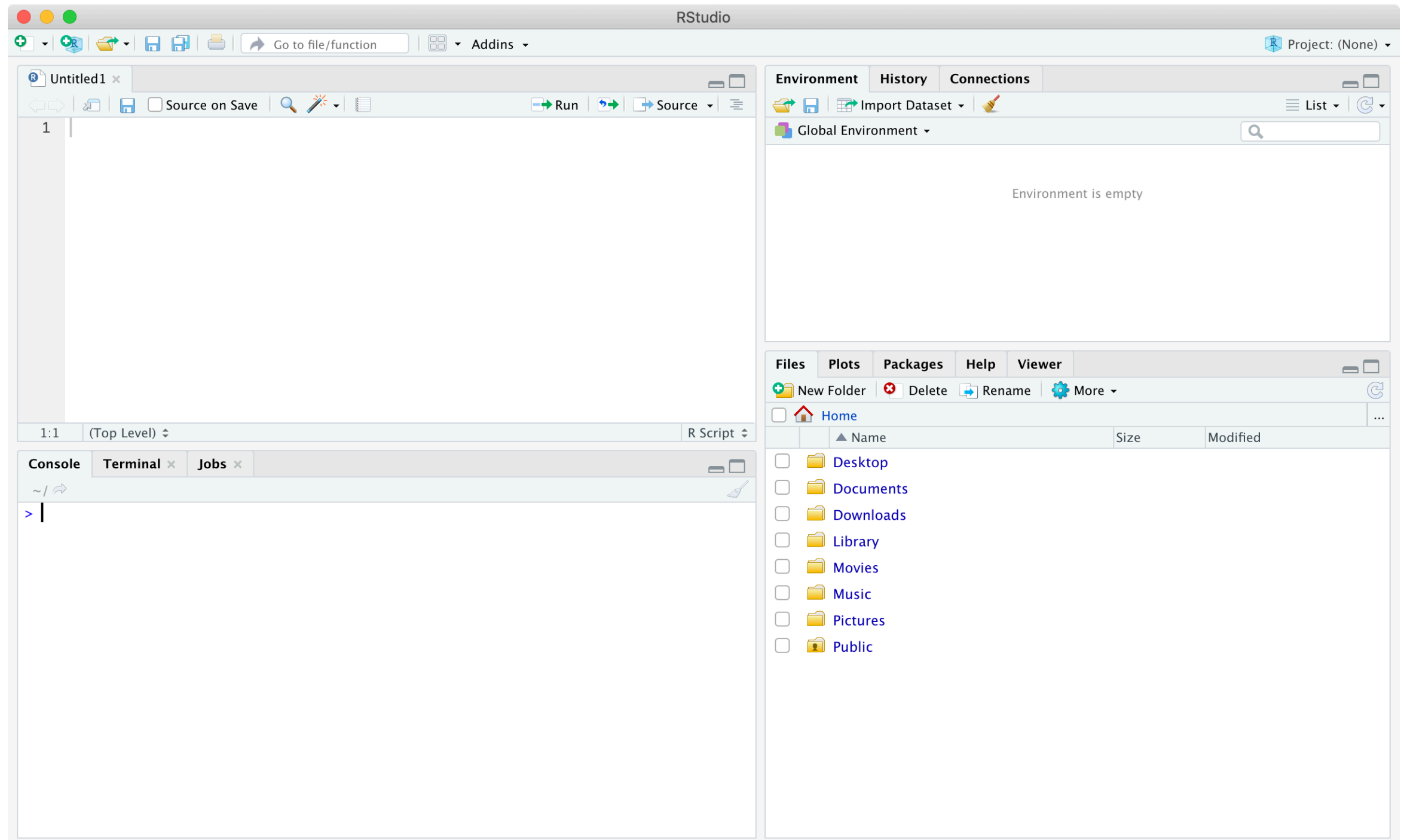
FIRE-UP STUDIO — IT SHOULD LOOK SOMETHING LIKE THIS



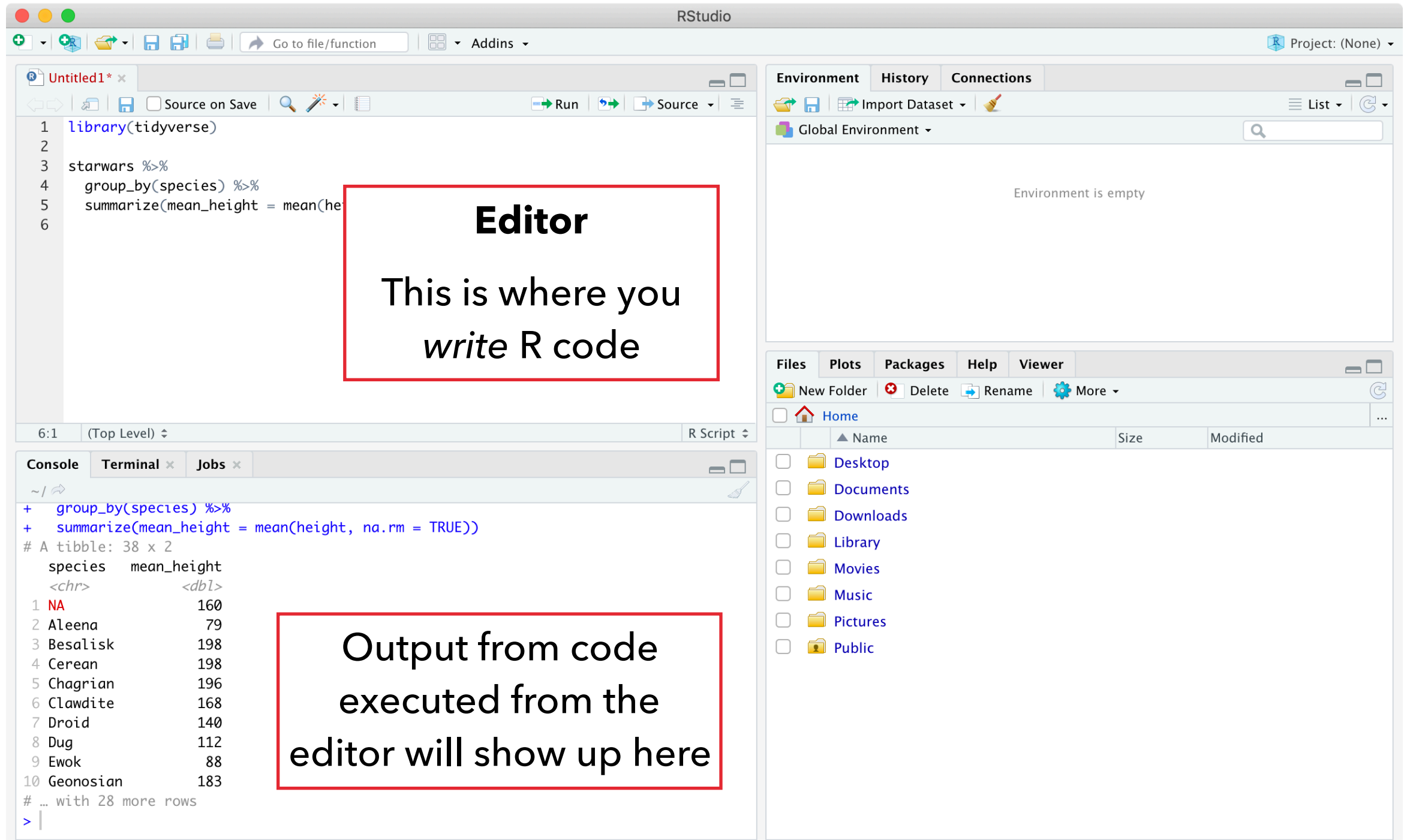
The screenshot shows the RStudio application window. The top toolbar includes icons for file operations and a search bar. The main interface is divided into four panes: Console, Environment, Files, and Plots. The Console pane on the left displays the R startup message and a prompt. The Environment pane on the top right shows the Global Environment. The Files pane on the bottom right shows the file browser. Three red-bordered callout boxes provide additional information:

- Console**
This is where you can *interactively* type and execute R Code
- R Environment**
Defined variables will be listed here
- File Browser**
Browser, open, delete, rename files

MAKE AN NEW R SCRIPT: FILE → NEW FILE → R SCRIPT



MAKE AN NEW R SCRIPT: FILE → NEW FILE → R SCRIPT



The screenshot displays the RStudio application window. The main Editor pane on the left contains R code for loading the tidyverse library and summarizing the mean height of Star Wars characters by species. A red-bordered box with the text "Editor" and "This is where you write R code" is overlaid on the code. The Environment pane on the right shows the Global Environment, which is currently empty. The Files pane at the bottom right shows the file explorer with a list of folders including Desktop, Documents, Downloads, Library, Movies, Music, Pictures, and Public. The Console pane at the bottom left shows the output of the executed code, displaying a tibble with 38 rows and 2 columns: species and mean_height. A red-bordered box with the text "Output from code executed from the editor will show up here" is overlaid on the console output.

Editor
This is where you write R code

```
1 library(tidyverse)
2
3 starwars %>%
4   group_by(species) %>%
5   summarize(mean_height = mean(height))
6
```

Environment
Global Environment
Environment is empty

Files
New Folder Delete Rename More
Home
Name Size Modified
Desktop
Documents
Downloads
Library
Movies
Music
Pictures
Public

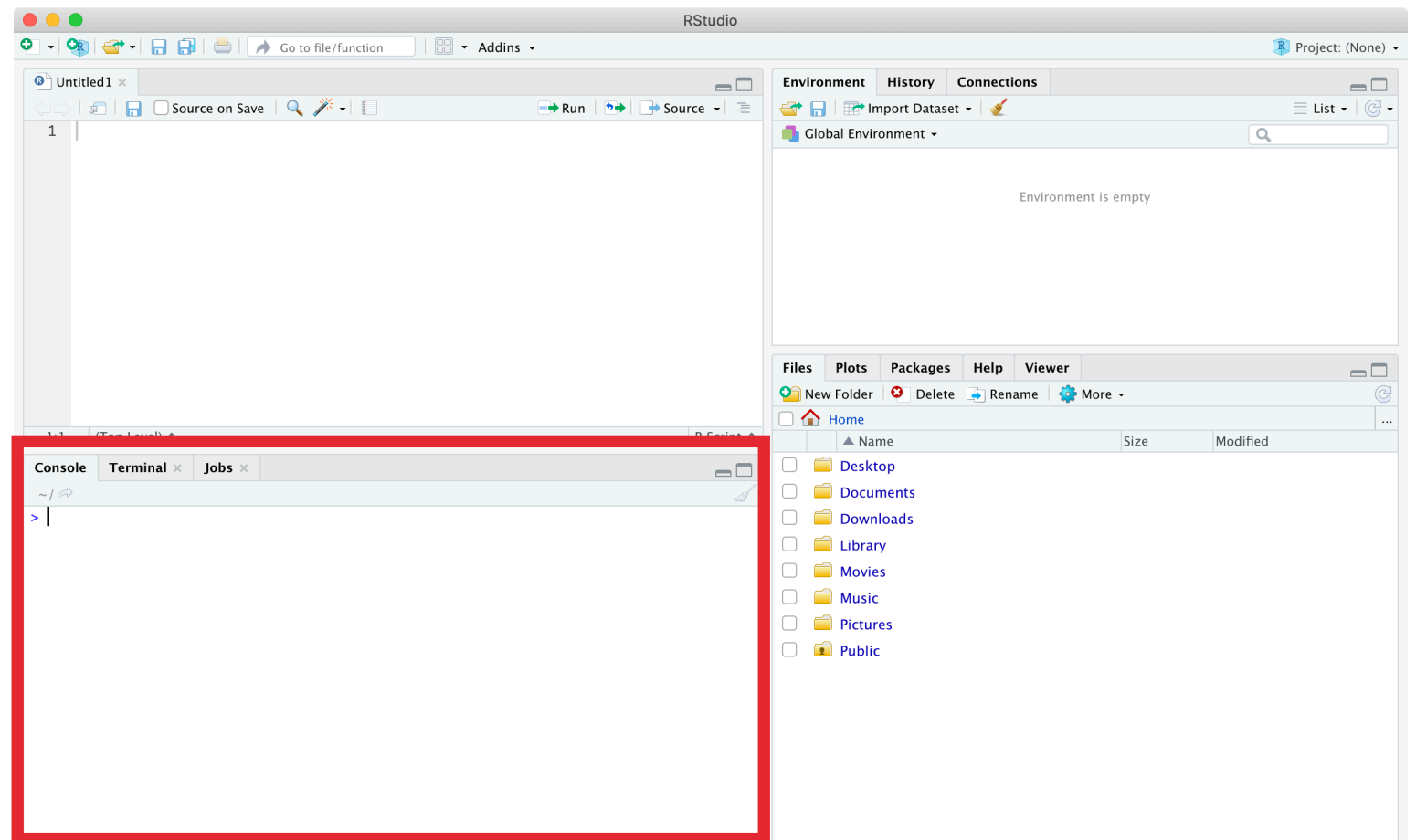
Console
~/
+ group_by(species) %>%
+ summarize(mean_height = mean(height, na.rm = TRUE))
A tibble: 38 x 2
 species mean_height
 <chr> <dbl>
1 NA 160
2 Aleena 79
3 Besalisk 198
4 Cerean 198
5 Chagrian 196
6 Clawdite 168
7 Droid 140
8 Dug 112
9 Ewok 88
10 Geonosian 183
... with 28 more rows
>

Output from code executed from the editor will show up here

THE CONSOLE VS. THE EDITOR

You'll use the *Console* to:

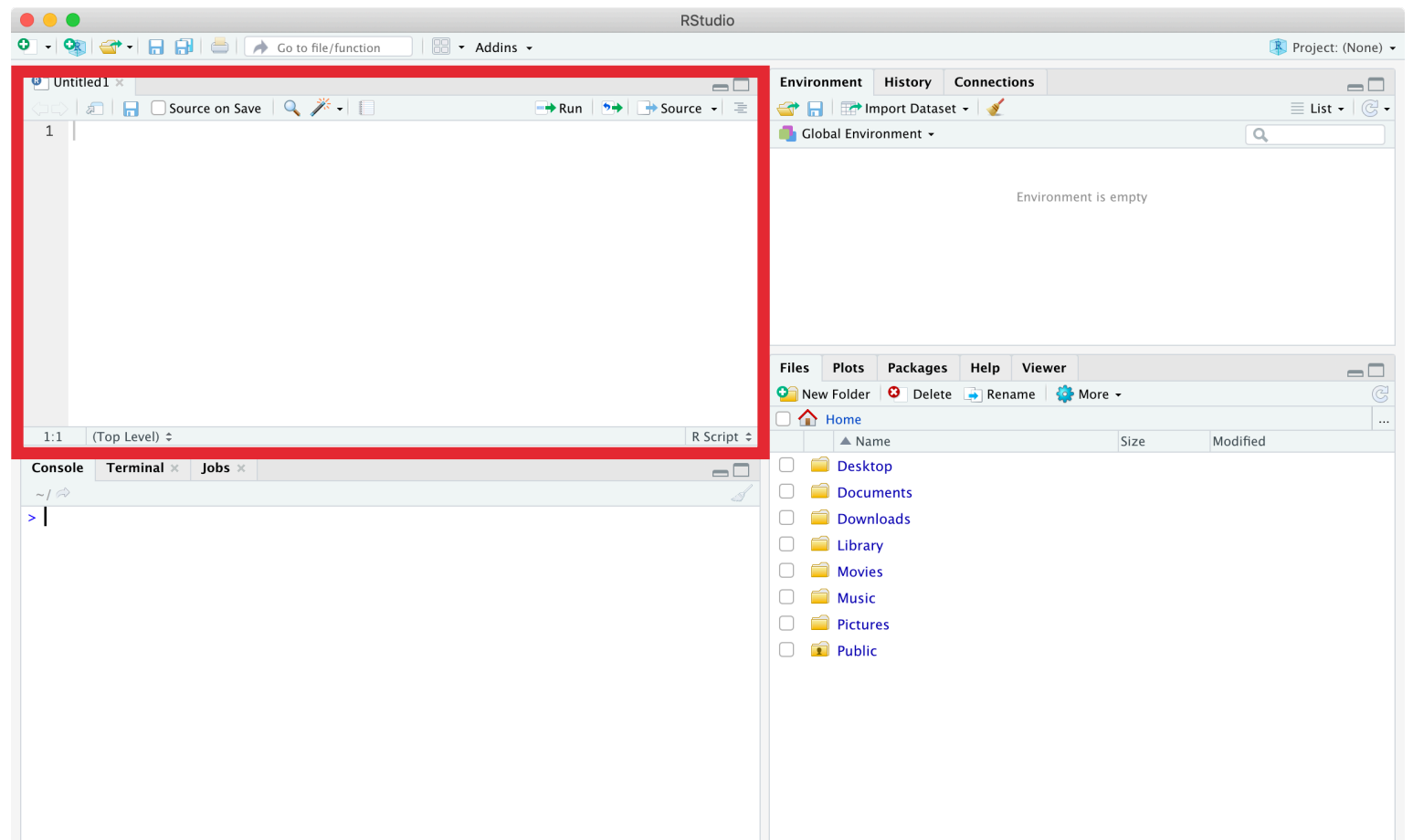
- ▶ Interactively work with R
- ▶ Try things out & experiment
- ▶ Test & check things
- ▶ Run code from the *Editor*



THE CONSOLE VS. THE EDITOR

You'll use the *Editor* to:

- ▶ Write R code
- ▶ Keep a persistent record of your analysis (i.e. save your work to a file)
- ▶ View a complete analysis script in one place



THE CONSOLE VS. THE EDITOR

For a typical analysis workflow

- ▶ Try to work mostly in the *editor* and save your work
- ▶ Treat anything you write in the console as ephemeral – code written in the console is not a real record of your work!
- ▶ Try to encapsulate your *entire* analysis in a script or set of scripts; avoid one-off analysis steps executed in the console
- ▶ Execute code from the editor by pressing CMD/CTRL–RETURN, great for testing your code as you write it

RSTUDIO PROJECTS ARE A GREAT WAY TO ORGANIZE YOUR WORK

- ▶ R Projects: organize and encapsulate your analysis projects
- ▶ File → New Project... → New Directory → New Project
name your project & choose where you want to save it
- ▶ This will create a new directory with a *.Rproj* file,
double click this file to get back to your work at any time
- ▶ R Projects help manage your R Studio session and your
working directory

A SUGGESTED WORKFLOW

Imagine you need to start a new analysis project

1. Create a new RStudio Project
2. Assemble the required input data & info you need, put them inside the project directory*
3. Write analysis scripts that do the work; consider naming them with a numeric prefix (e.g. 01_..., 02_...) indicating the order they should be executed in
4. Write, test, write test, write, test...
5. Double click the .Rproj file when you want to get back to work

* can depend on how you store data (e.g. networked storage)

Demo Time

1. Create an RStudio Project
2. Review of R basics example code [*00_quick_r_intro.R*]