Advanced R

July 2018

Charlotte Wickham
@cvwickham
cwickham@gmail.com
cwick.co.nz

Adapted from *Tidy Tools* by Hadley Wickham

CC BY SA

Welcome to the **Advanced R** Workshop!

1. Get connected to the wifi:

2. Get the course materials:
   `usethis::use_course("bit.ly/advr-jul18")`
3. Follow the instructions in **setup.R**
4. Stuck? Please ask for help!

Use this link, **not** one in handout!

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Preliminaries

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Your turn

This course is very hands on, and while we're here to help you, the best resource is often the person sitting next to you.

Introduce yourself to your neighbours. Who are you and what are you using R for?

This means that you have to work!

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Goal

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Go from code like this...

```
# Fix missing values
df$a <- df$a[df$a == -99] <- NA
df$b <- df$b[df$b == -99] <- NA
df$c <- df$c[df$c == -99] <- NA
df$d <- df$d[df$d == -99] <- NA
df$e <- df$e[df$e == -99] <- NA
df$f <- df$f[df$f == -99] <- NA
df$g <- df$g[df$g == -98] <- NA
df$h <- df$h[df$h == -99] <- NA
df$i <- df$i[df$i == -99] <- NA
df$j <- df$i[df$j == -99] <- NA
```

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

To code like this...

```
fix_missing <- function(x) {
  x[x == -99] <- NA
  x
}
df <- modify(df, fix_missing)
```

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Overall goal:

Code that expresses what you did, not the details of how you did it.

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Learning Objectives

By the end of the day you'll be able to ...

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Functions

Reduce repetitive code by extracting a common action into a function

1. Identify when to write a function, and use a strategy for how to write a function
2. Apply good design principles to make your functions easy for you and others to use.

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Functional Programming

Reduce repetitive code by having functions write for loops for you

1. Describe what it means that functions in R are first class citizens
2. Solve iteration problems using purrr::map() and friends
3. Avoid a single error stopping iteration with purrr::safely()

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Tidy Evaluation

Write functions that wrap tidyverse functions

1. Identify arguments in functions that are automatically quoted
2. Refer to a saved variable in the quoted arguments of a tidyverse function
3. Wrap a tidyverse function inside your own function by combining quo() and !!

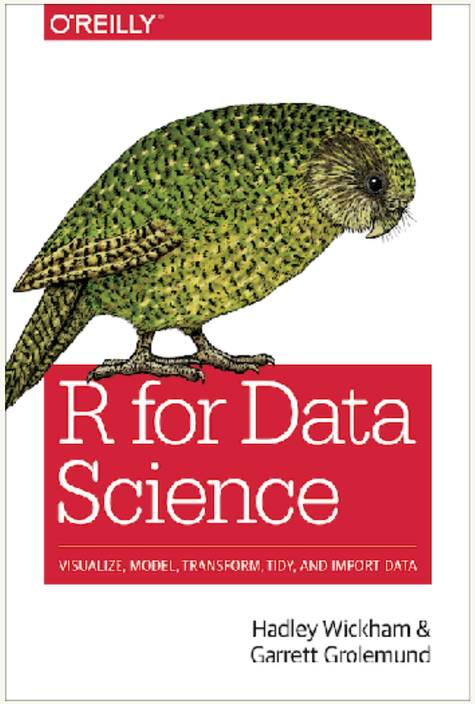
Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Schedule

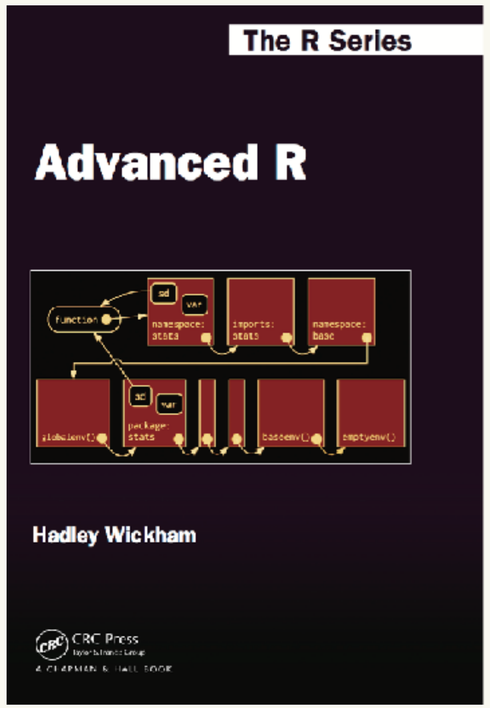
10:15-10:30	Morning break
12:30-14:00	Lunch
15:15-15:30	Afternoon break

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Much of the course is drawn from existing books



http://r4ds.had.co.nz



http://adv-r.had.co.nz/

2nd Ed. WIP:
https://adv-r.hadley.nz

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Warmups

Don’t expect to know all the answers!

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Your turn

What are the four common types of atomic vectors?
(Bonus points for the two uncommon types)

What are the three primary properties of a vector?

Four common types: logical, integer, double, character

```
typeof(TRUE)
typeof(1L)
typeof(1.5)
typeof("a")
```

Every vector has three properties:

```
x <- 1:5

# 1. Type:
typeof(x)

# 2. Length
length(x)

# 3. Attributes
attributes(x)
```

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Missing values

What does NA == NA return? Why?

What should you use instead?

There isn't a single unknown value

```
age_john <- NA
age_mary <- NA
age_john == age_mary

is.na(x)
```

```
sum(is.na(x))
mean(is.na(x))
```

Your turn

What are the six types of thing that you can put inside []?

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Number (be specific), Integer (be more specific)

1. positive integer (get those elements)
2. negative integer (exclude)
3. zero
4. logical
5. character
6. nothing

blank	include all
integer	**+ve**: include **0**: drop all **-ve**: exclude
logical	keep TRUEs
character	lookup by name

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

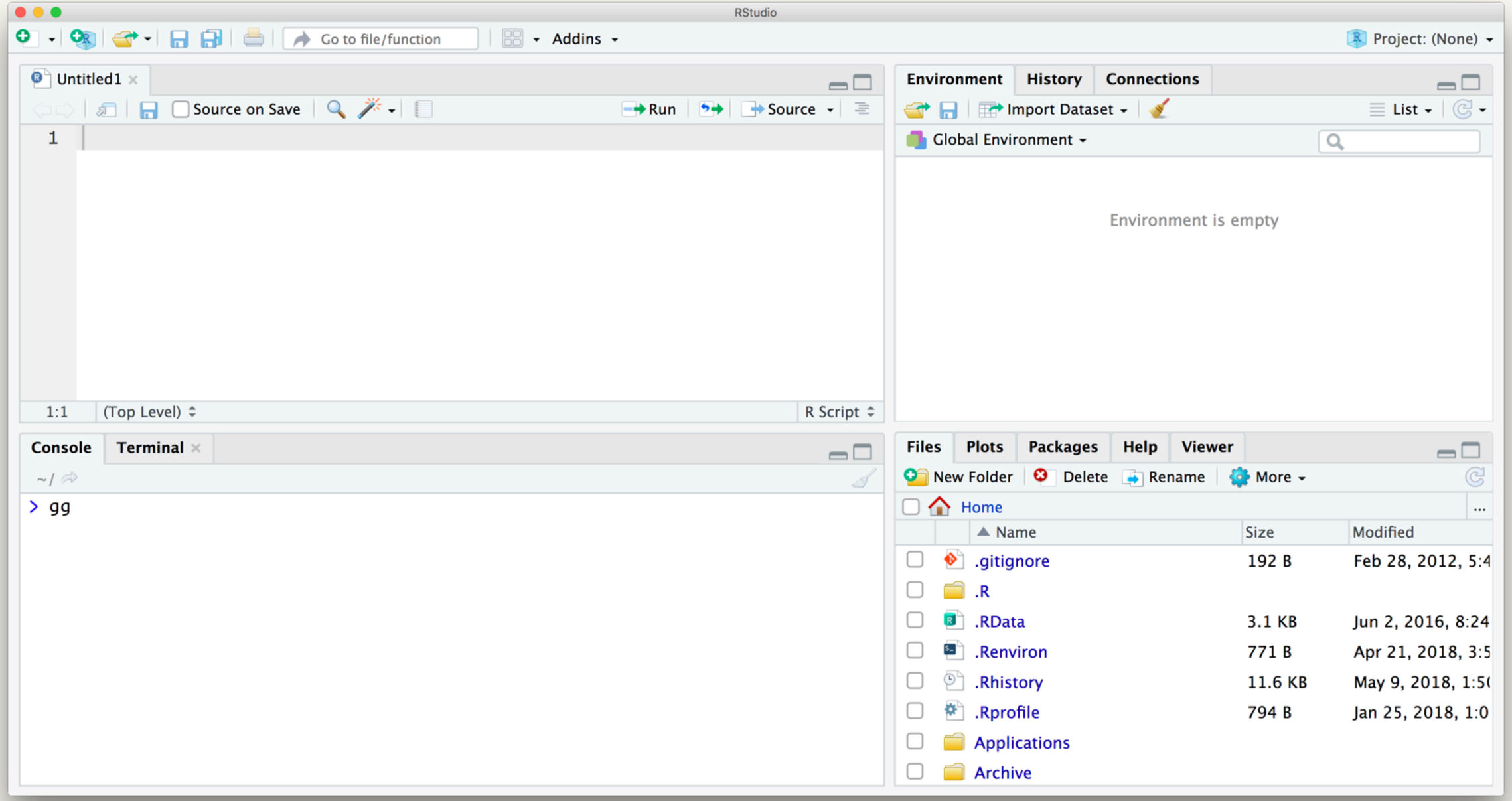
Use character subsetting for simple look ups

```
x <- c("m", "f", "u", "f", "f", "m", "m")

lookup <- c(m = "Male", f = "Female", u = NA)
lookup[x]

unname(lookup[x])
```

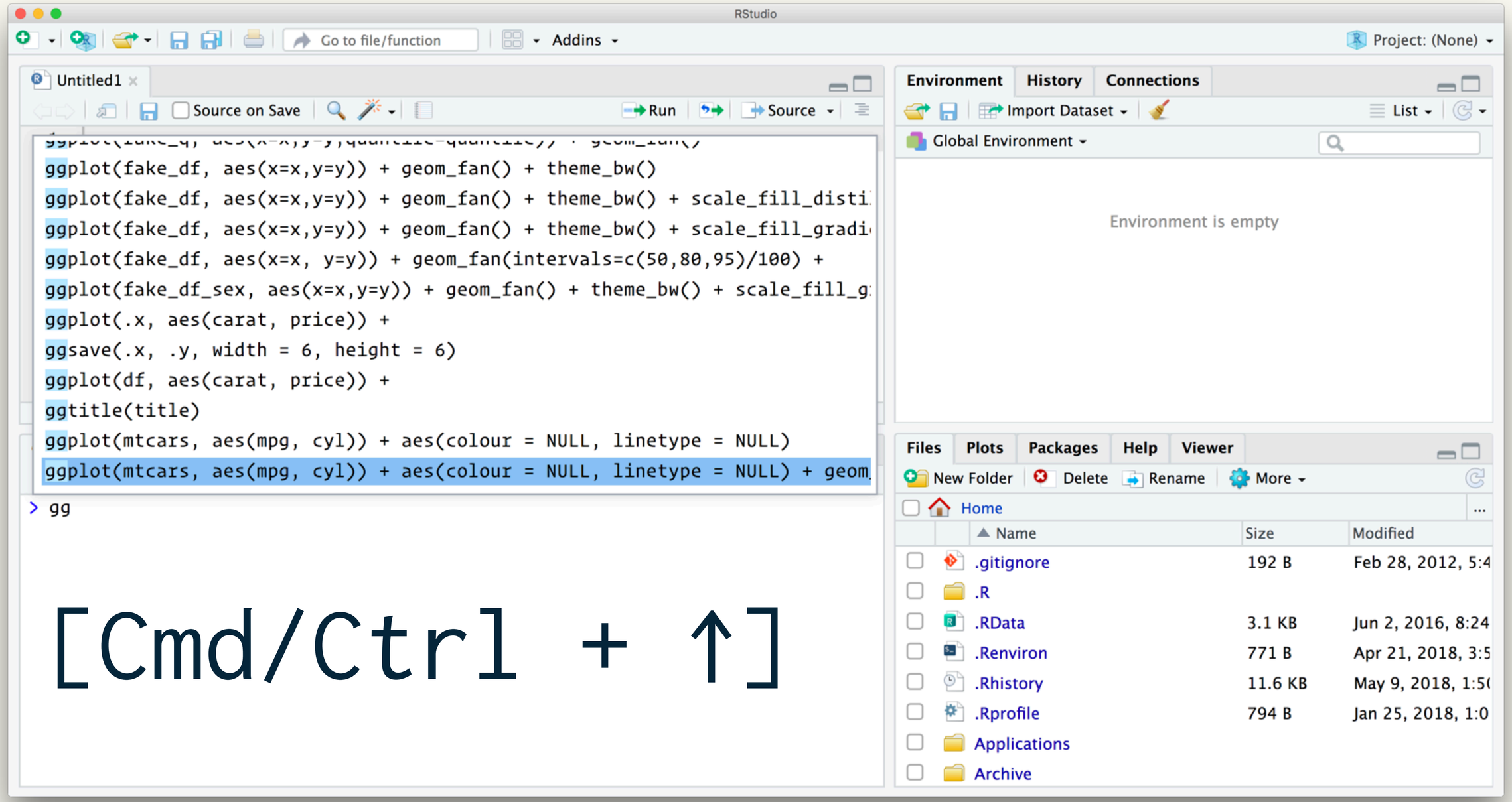
RStudio

You don't have to use RStudio,
but if you do, try to master it!

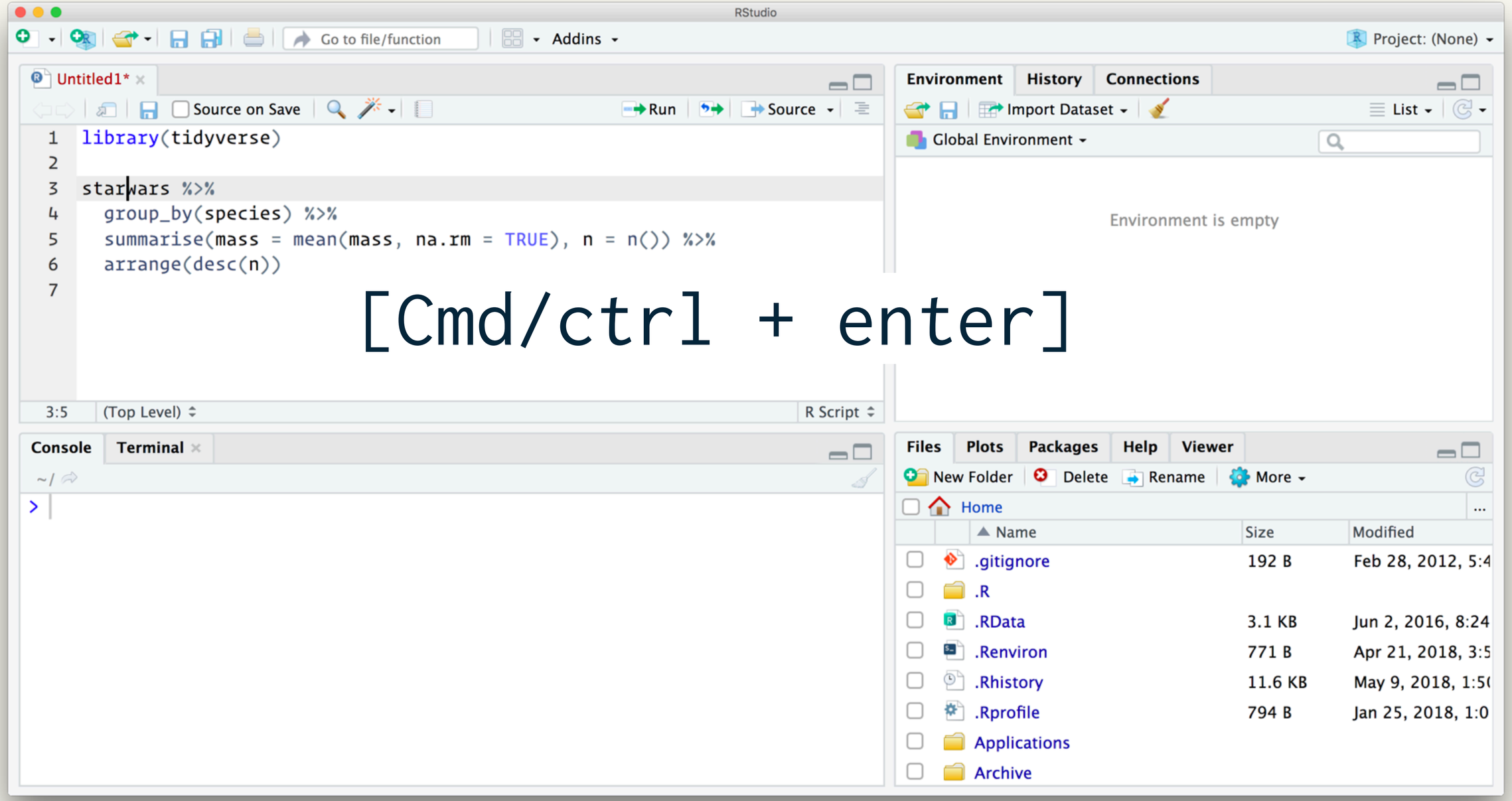
Get materials: `usethis::use_course("bit.ly/advr-jul18")`



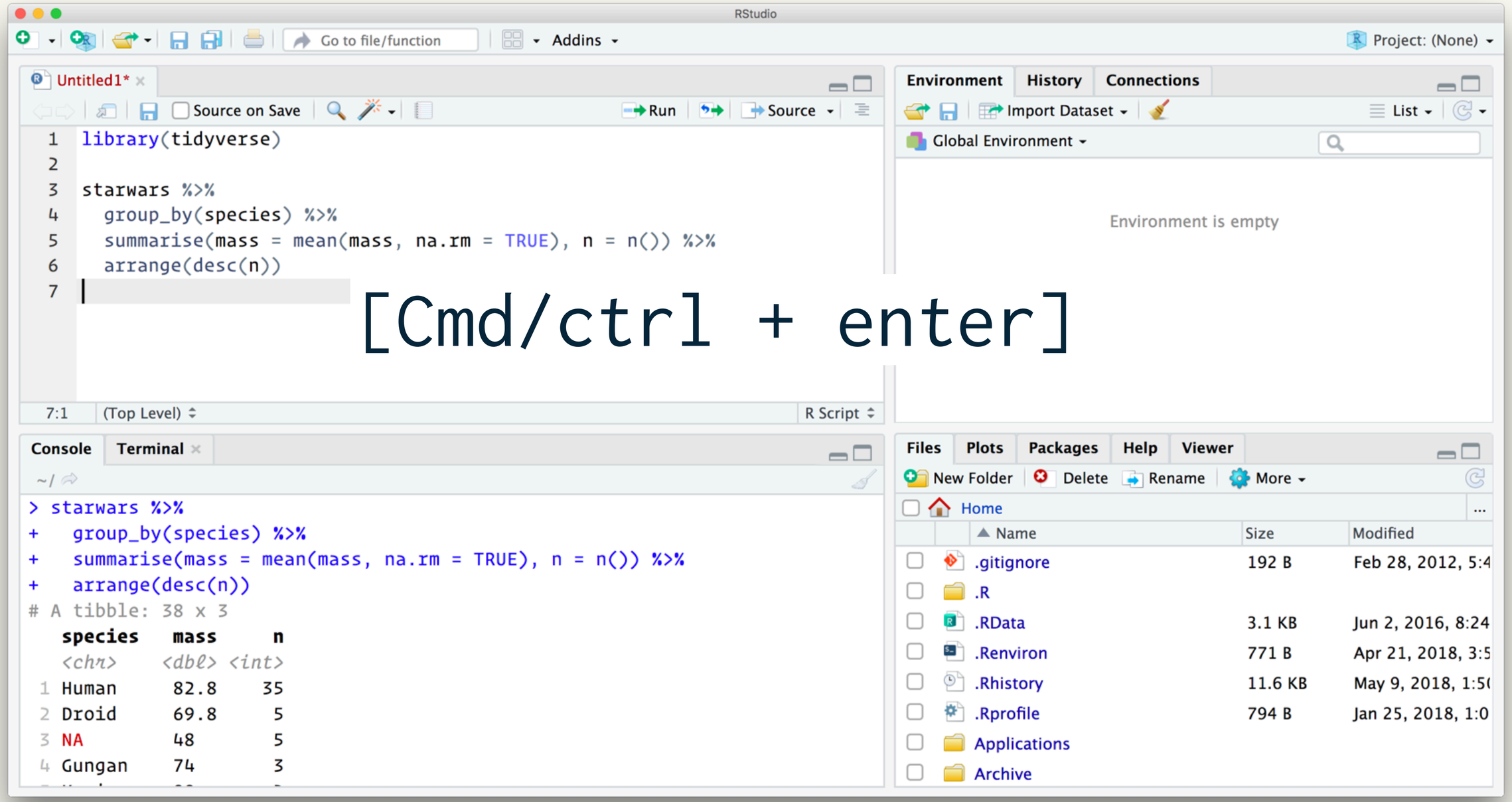
Get materials: `usethis::use_course("bit.ly/advr-jul18")`



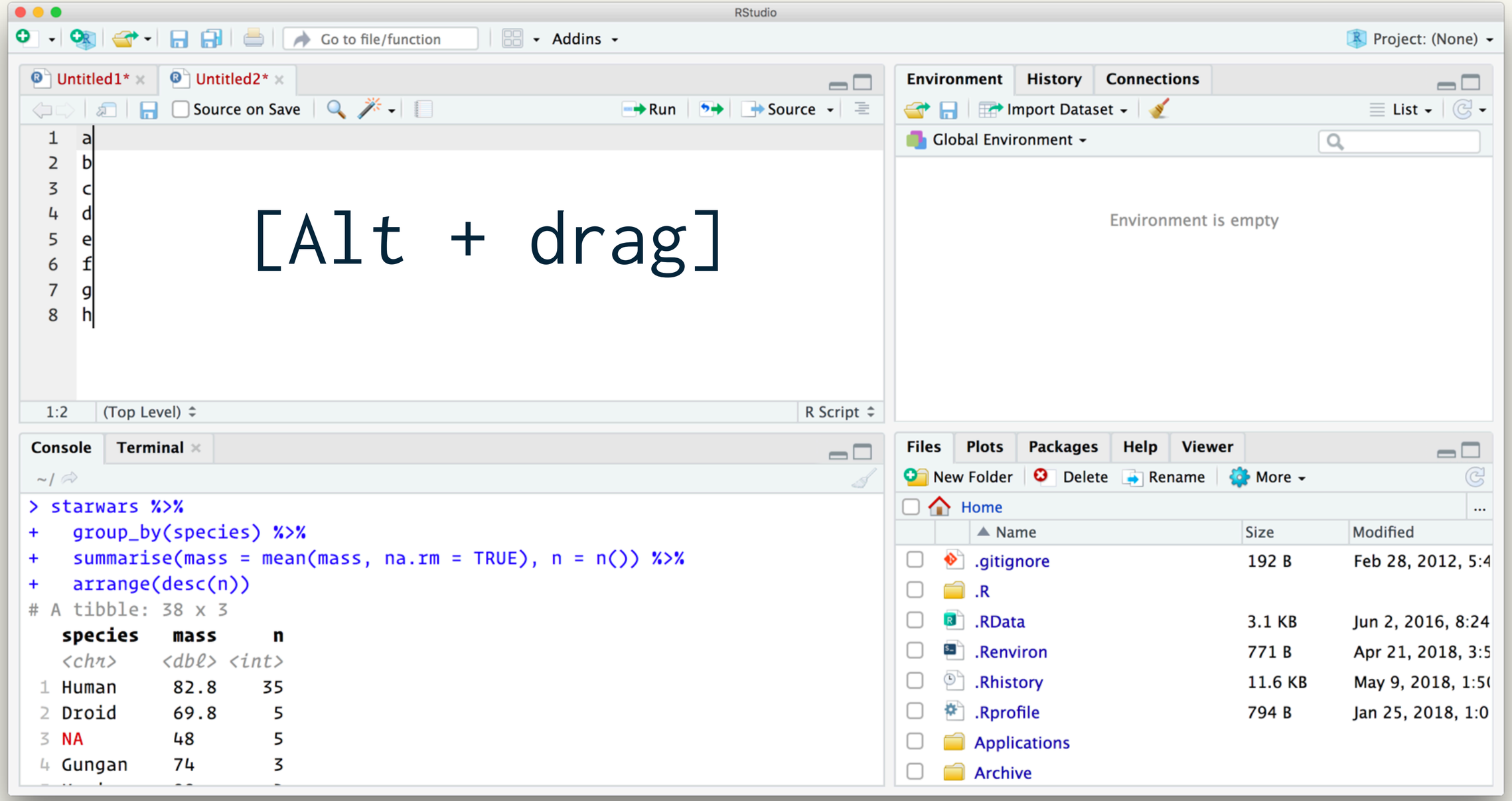
Get materials: `usethis::use_course("bit.ly/advr-jul18")`



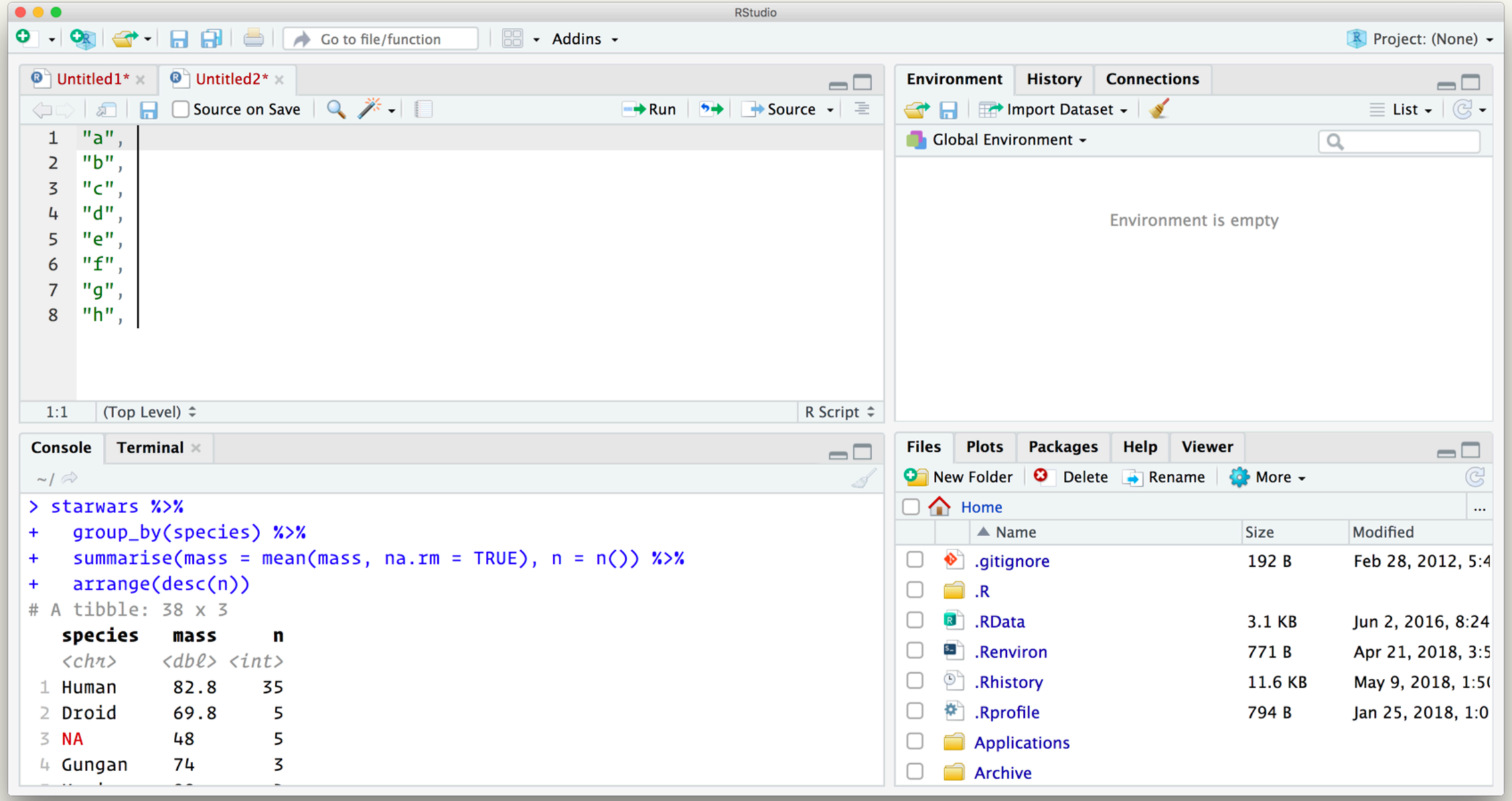
Get materials: `usethis::use_course("bit.ly/advr-jul18")`



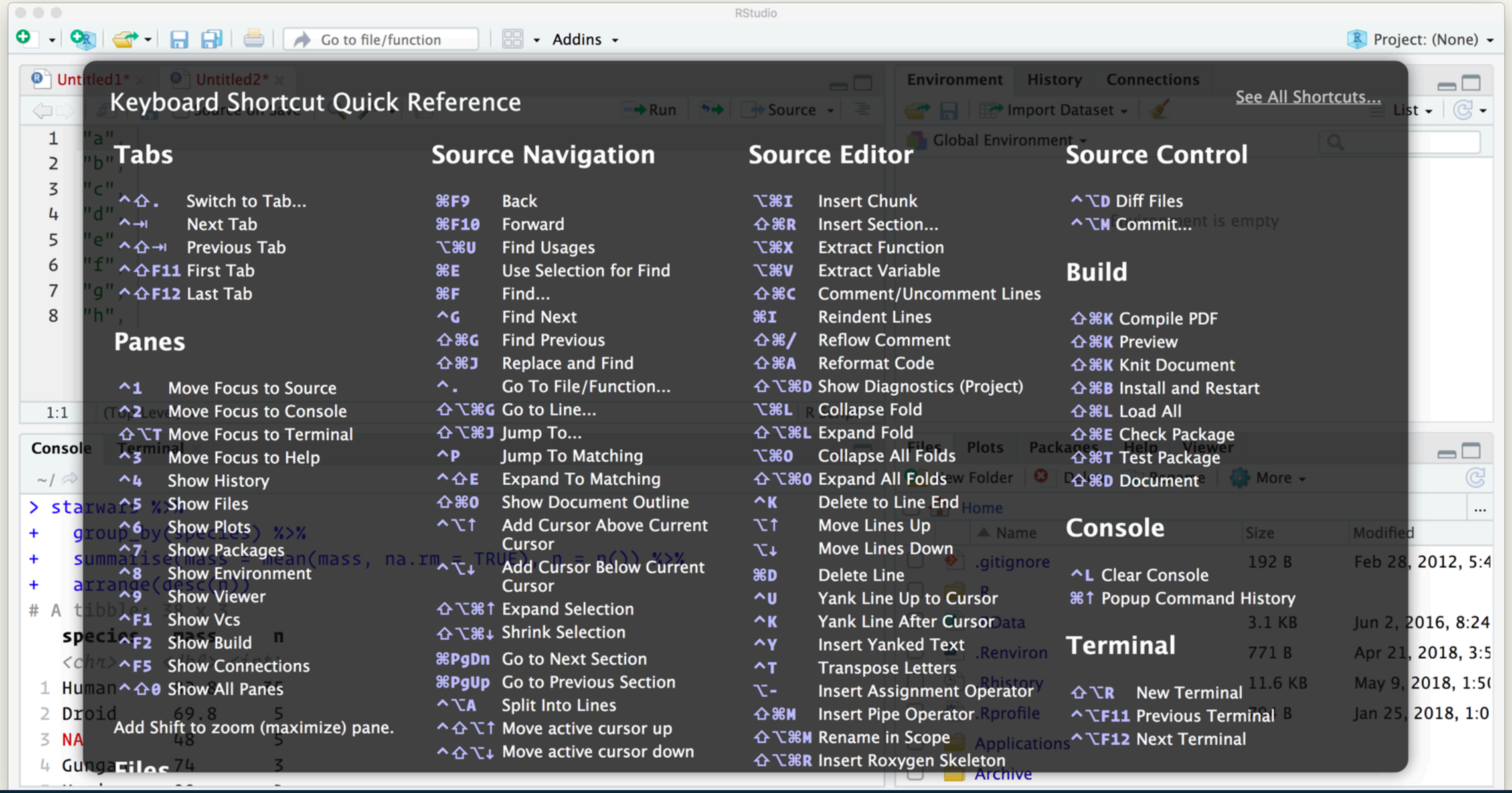
Get materials: `usethis::use_course("bit.ly/advr-jul18")`



Get materials: `usethis::use_course("bit.ly/advr-jul18")`



Get materials: `usethis::use_course("bit.ly/advr-jul18")`



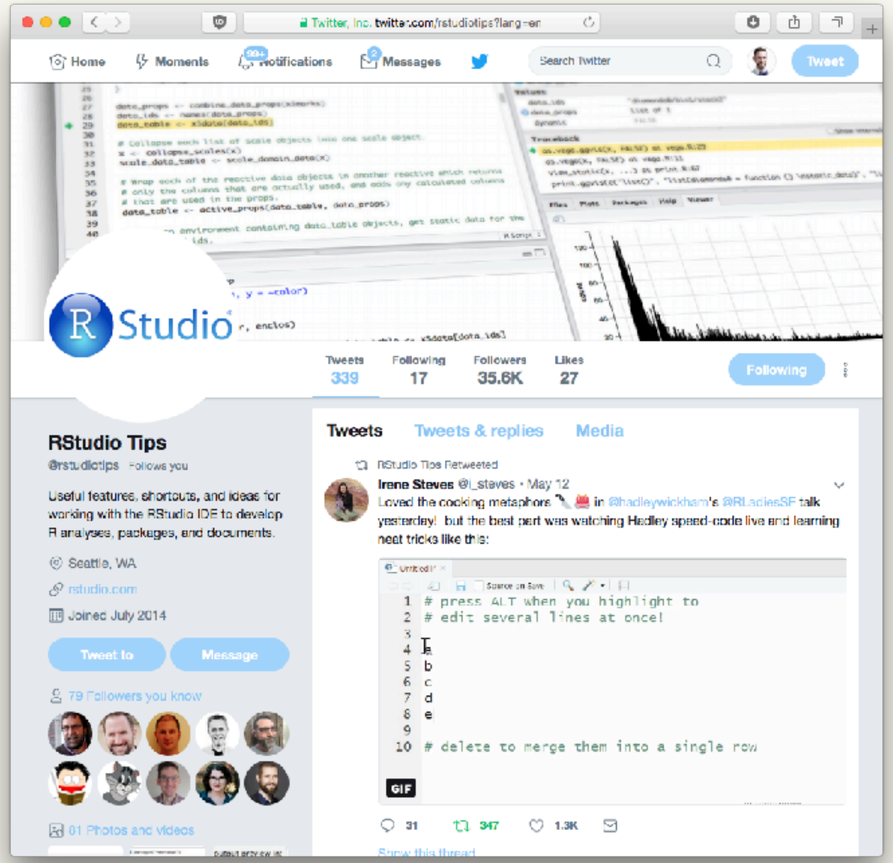
Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Your turn

What's the shortcut for <- (assignment)?

What about %>% (pipe)?

How can you quickly comment a block of lines?



@rstudiotips

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Get materials: `usethis::use_course("bit.ly/advr-jul18")`

Adapted from *Tidy Tools* by Hadley Wickham

This work is licensed as

Creative Commons
Attribution-ShareAlike 4.0
International

To view a copy of this license, visit
https://creativecommons.org/licenses/by-sa/
4.0/

Get materials: `usethis::use_course("bit.ly/advr-jul18")`