Welcome to the Tidy Tools Workshop!

1. Get connected to the wifi
   rstudioconf19 password: tidyverse
2. Get the course materials
   `usethis::use_course("rstd.io/tidytools19")`
3. Check you have the needed packages by running `setup.R`
4. Stuck? Please ask for help!

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Preliminaries

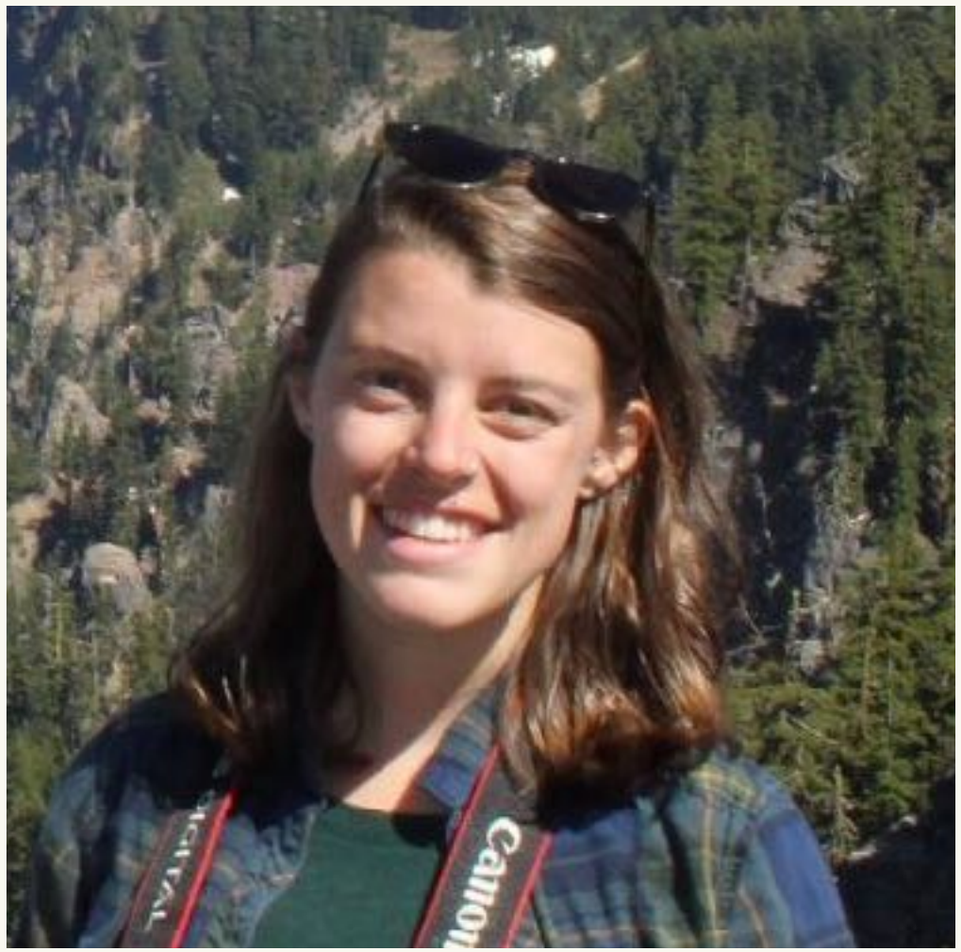
Get materials: `usethis::use_course("rstd.io/tidytools19")`

HELLO my name is

Charlotte

cwick.co.nz @cvwickham

Get materials: `usethis::use_course("rstd.io/tidytools19")`



Erin Howard

PhD student

Oregon State University

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Miles McBain

Research Associate, ACEMS, Queensland University of Technology.

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Dana Seidel

Ph.D. candidate
University of
California, Berkeley

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Hadley Wickham

Chief Scientist
RStudio

Joining us in the afternoons.

Get materials: `usethis::use_course("rstd.io/tidytools19")`

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.

This means that you have to work!

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

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Goal: help build tidy tools

Writing functions

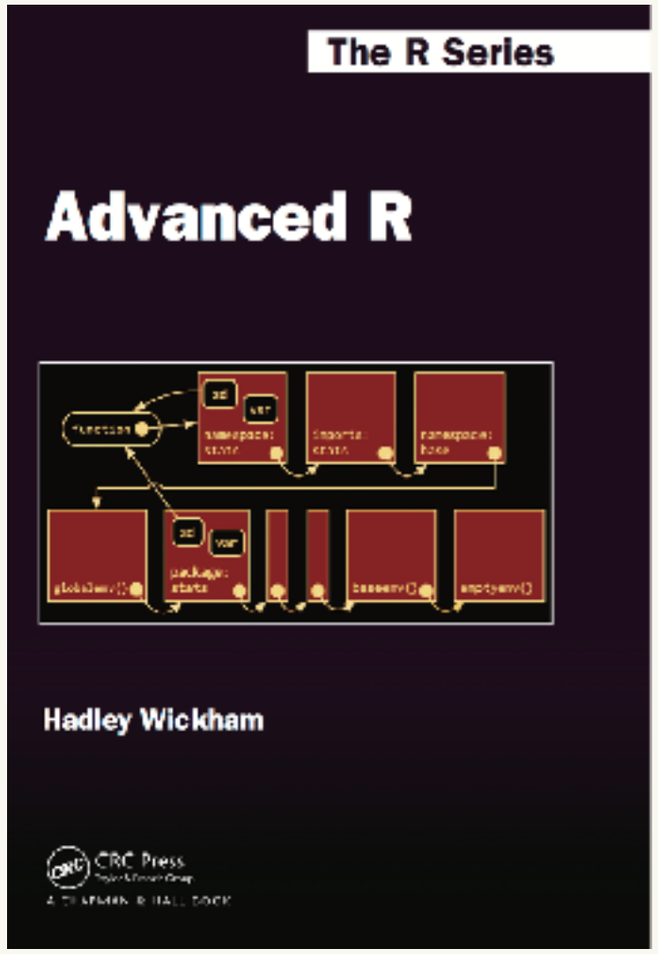
Individual functions to solve individual problems

Designing APIs

Family of functions that work together to solve family of problems

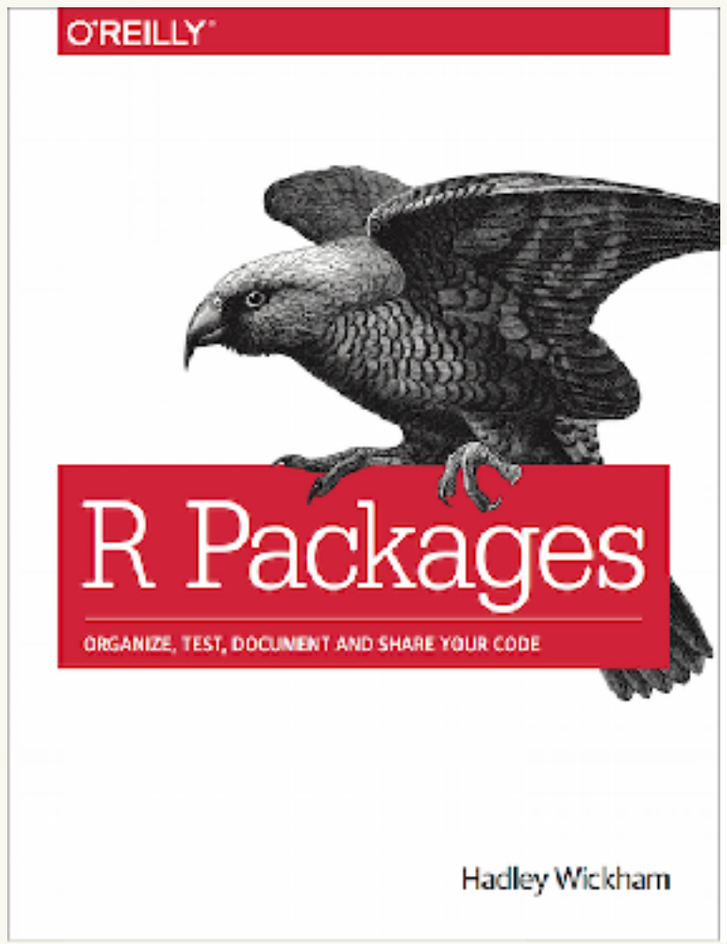
Get materials: `usethis::use_course("rstd.io/tidytools19")`

Much of the course is drawn from existing books



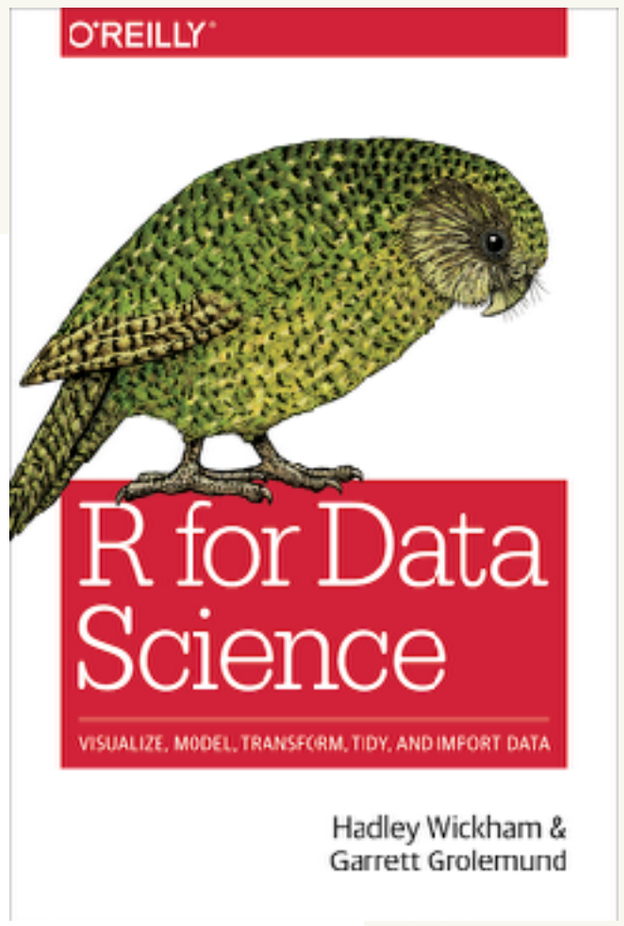
http://adv-r.hadley.nz/

Working on 2nd ed



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

https://amzn.com/1491910399

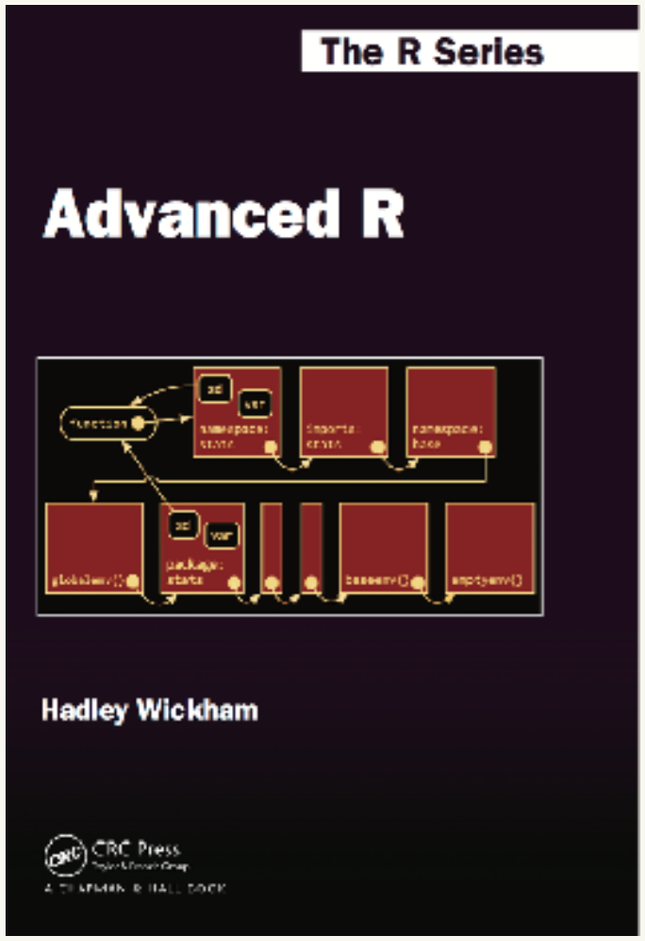


http://r4ds.had.co.nz

https://amzn.com/1491910399

Get materials: `usethis::use_course("rstd.io/tidytools19")`

But the primary book does not yet exist



How R works

Tidy tools

Hadley Wickham

How to solve real problems with R

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Schedule

Day 1	Day 2
Packages	**Errors**
10:30-11am Morning Break	10:30-11am Morning Break
Testing	**Object Oriented Programming**
12:30-2pm Lunch	12:30-2pm Lunch
API Design	**Tidy Evaluation**
3:30-4pm Afternoon Break	3:30-4pm Afternoon Break
Functional Programming	**Document/Share**

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Warmups

Don’t expect to know all the answers!

Get materials: `usethis::use_course("rstd.io/tidytools19")`

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?

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Four common types: logical, integer, double, character

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

```
# We'll talk about this (S3) later:
typeof(factor(1:10))
typeof(Sys.Date())
```

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Every vector has three properties:

```
x <- 1:5

# 1. Type:
typeof(x)

# 2. Length
length(x)

# 3. Attributes
attributes(x)
# (we'll come back to those later)
```

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Missing values

What does NA == NA return? Why?

What should you use instead?

Get materials: `usethis::use_course("rstd.io/tidytools19")`

There isn't a single unknown value

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

```
is.na(x)
```

Get materials: `usethis::use_course("rstd.io/tidytools19")`

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

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Your turn

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

Get materials: `usethis::use_course("rstd.io/tidytools19")`

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

Get materials: `usethis::use_course("rstd.io/tidytools19")`

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])
```

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Your turn

```
x <- runif(1e6)
lobstr::obj_size(x)
#> 8,000,040 B

y <- list(x, x, x)
lobstr::obj_size(y)
#> ???

y[[1]][[1]] <- NA
lobstr::obj_size(y)
#> ???
```

https://lobstr.r-lib.org/

Get materials: `usethis::use_course("rstd.io/tidytools19")`

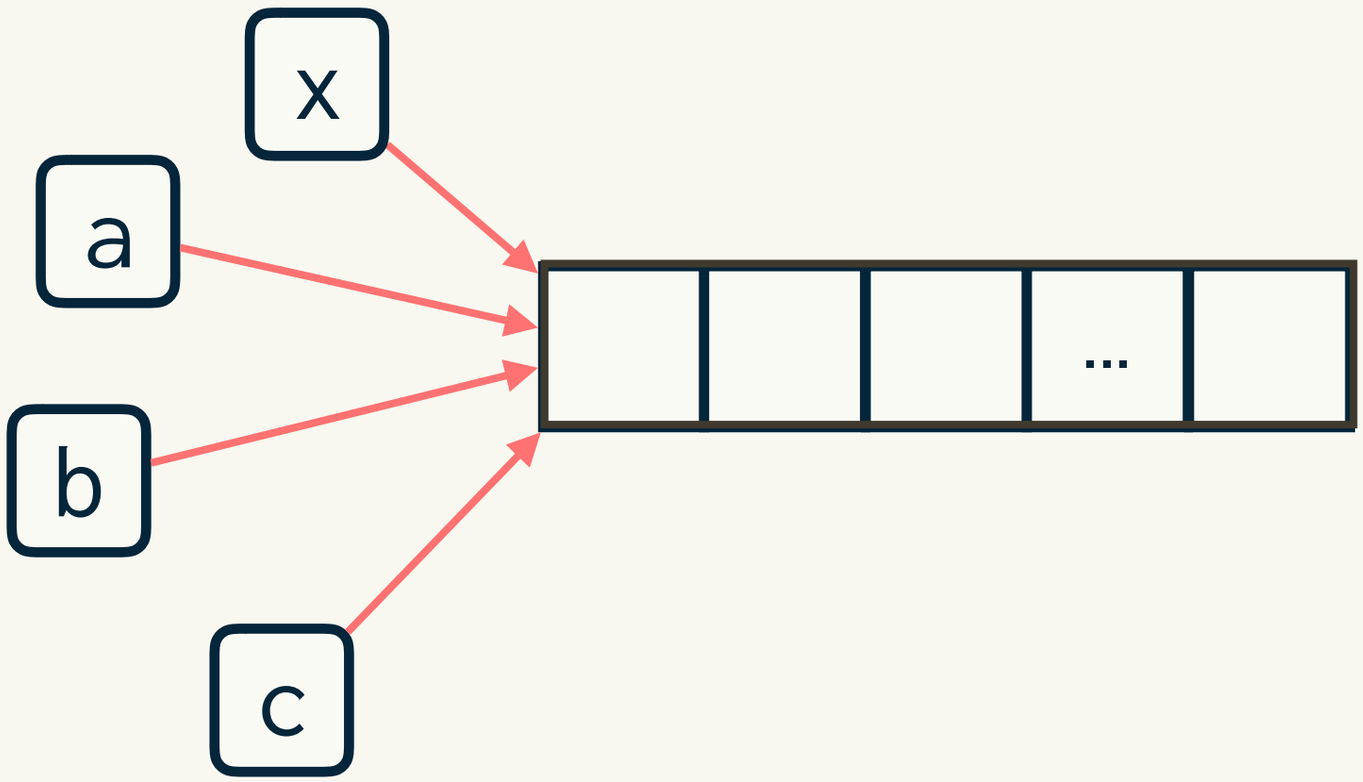
A **name** is a reference to a **value**

```
x <- runif(1e6)
```

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Many references can point to one object

```
a <- b <- c <- x
```

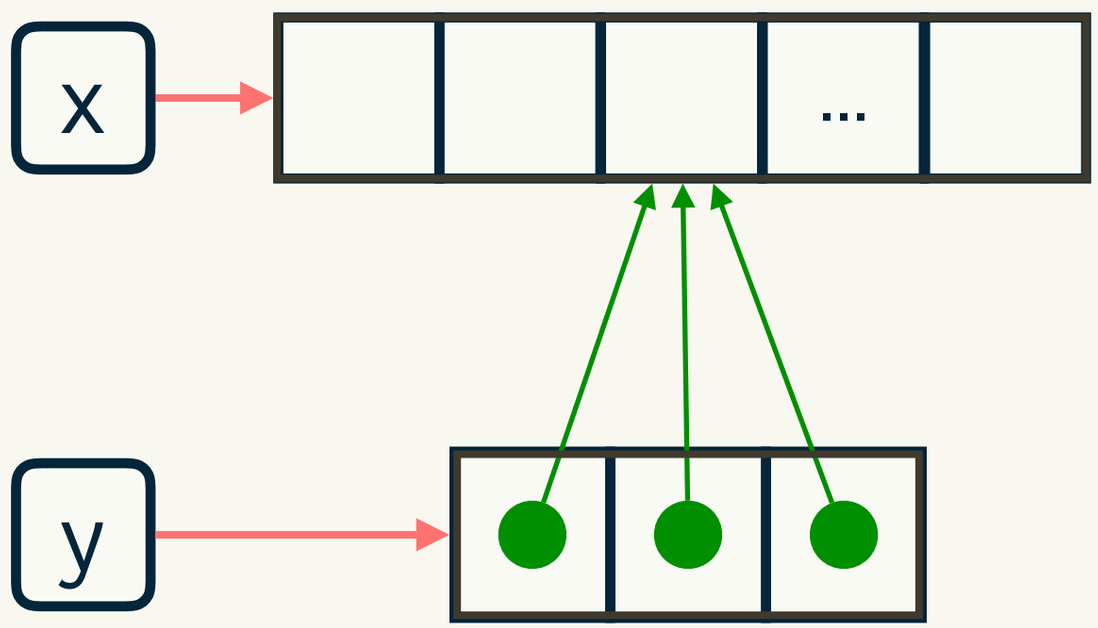


How big is **a**?
How big is **b**?
How big are **a** and **b** together?

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Elements of lists are also references

```
y <- list(x, x, x)
```

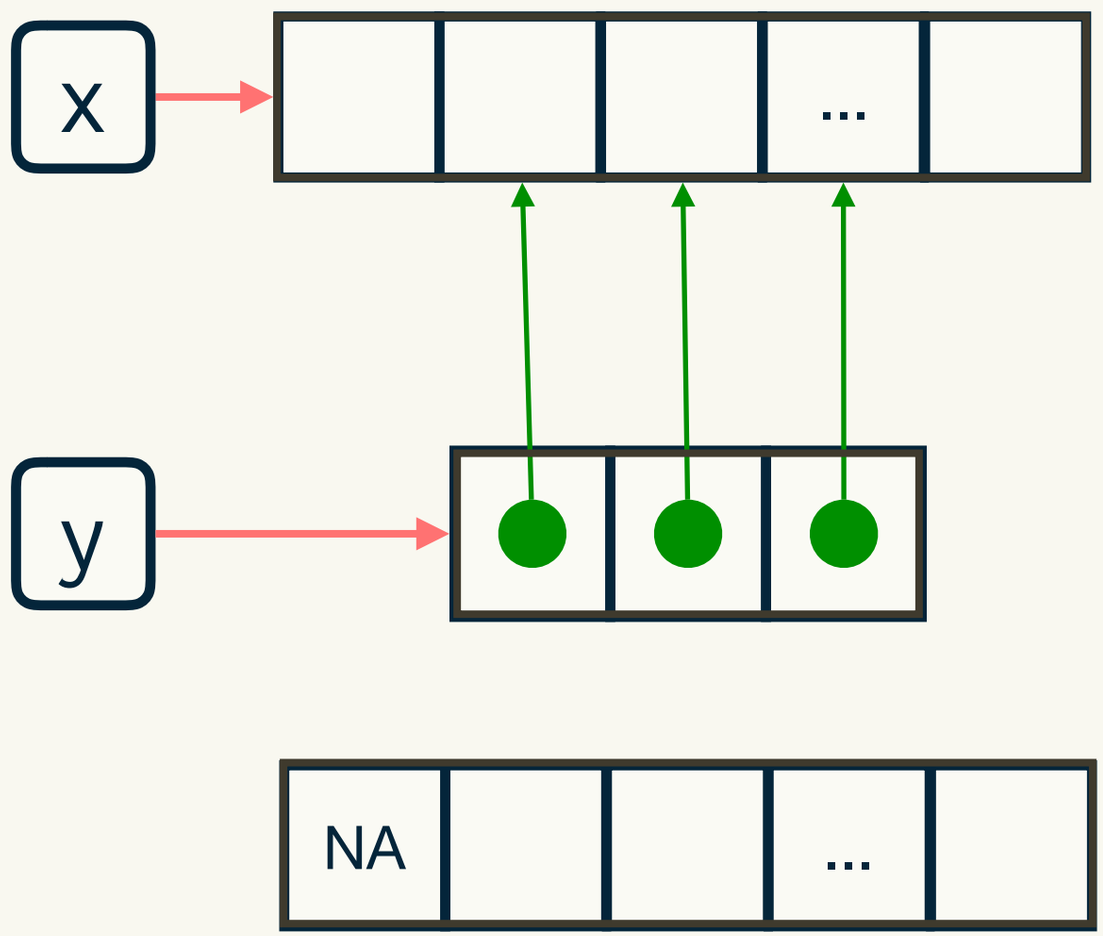


How big is **x**?
How big is **y**?
How big are **x** and **y** together?

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Modifying an object creates a copy

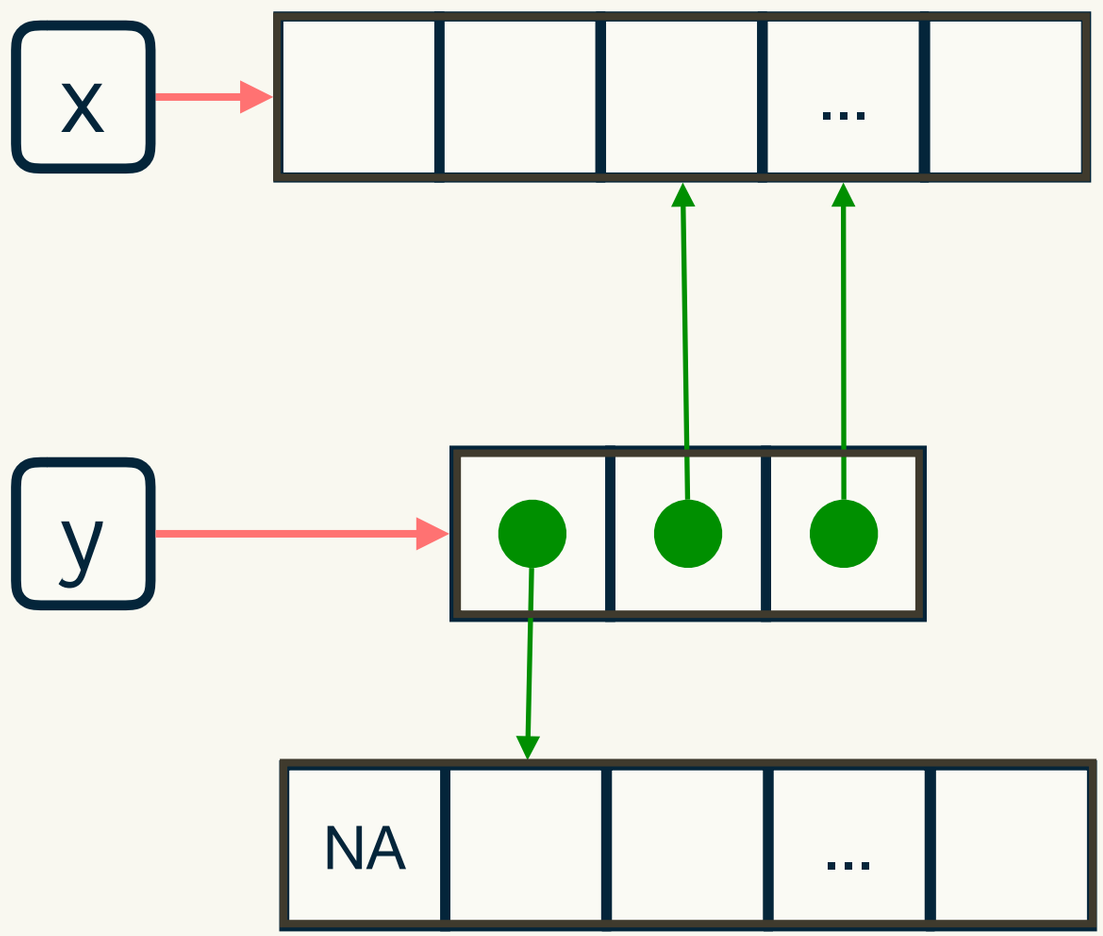
```
y[[1]][[1]] <- NA
```



Get materials: `usethis::use_course("rstd.io/tidytools19")`

Modifying an object creates a copy

```
y[[1]][[1]] <- NA
```

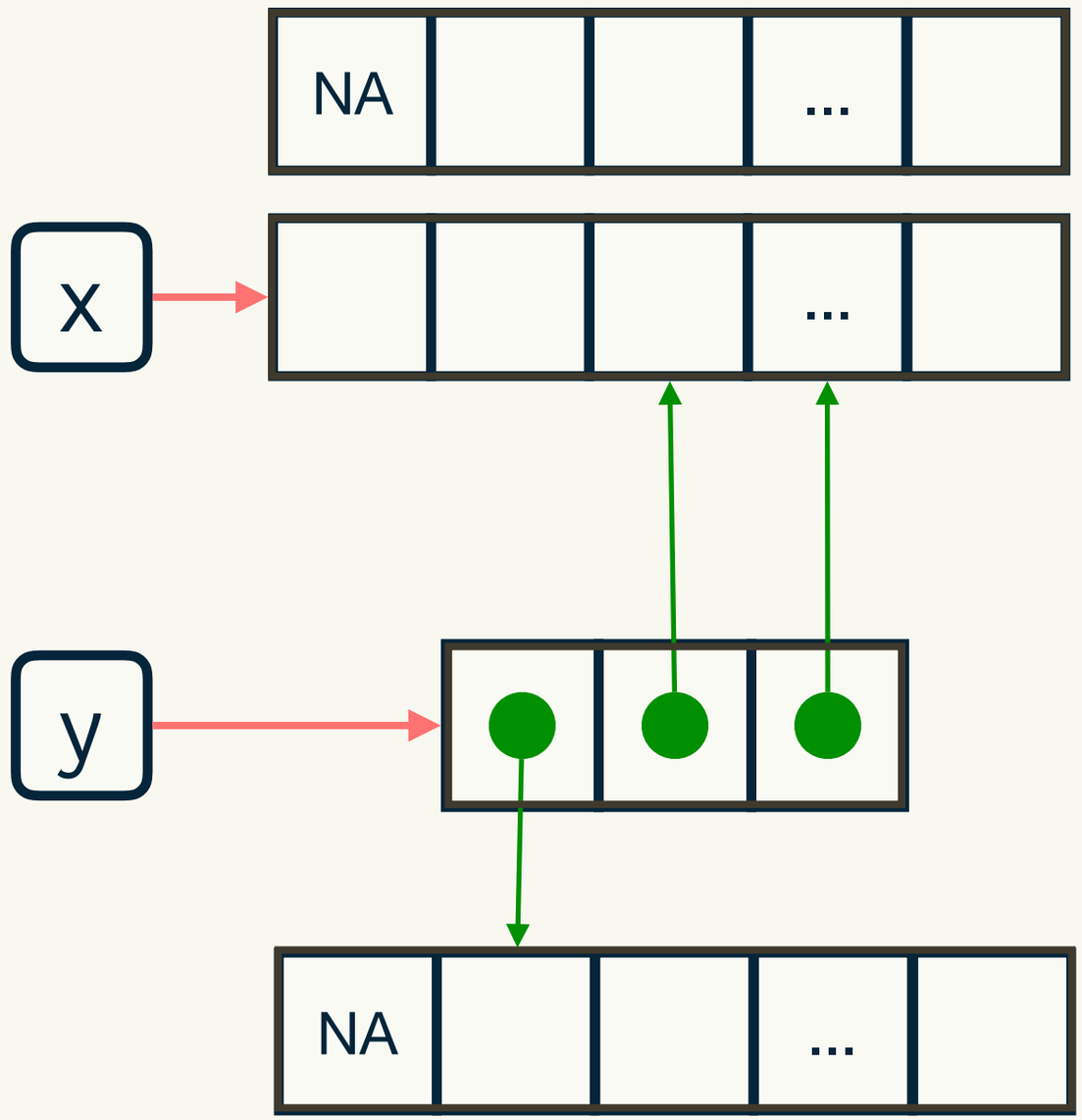


How big is **x**?
How big is **y**?
How big are **x** and **y** together?

Get materials: `usethis::use_course("rstd.io/tidytools19")`

Modifying an object creates a copy

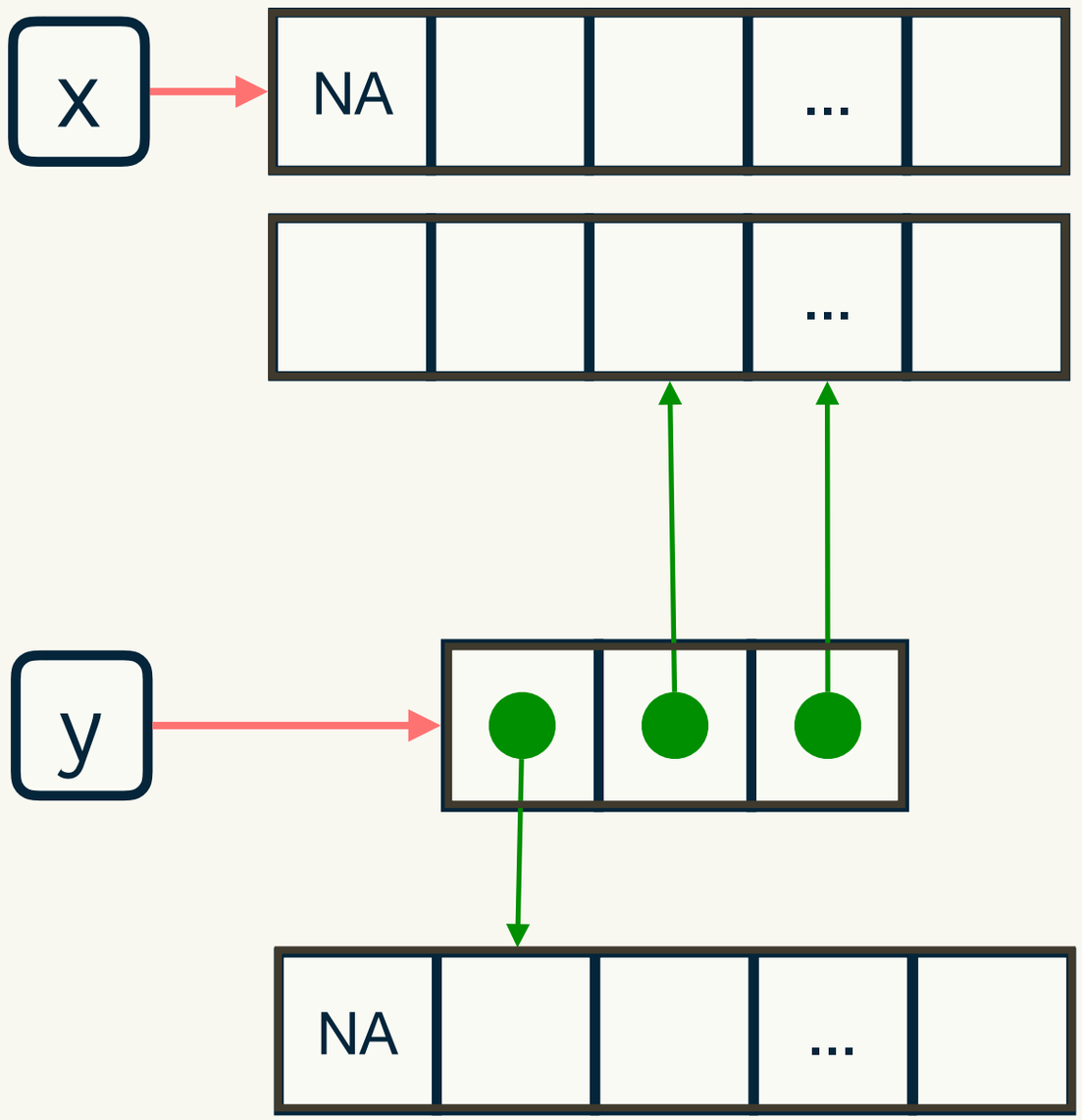
```
x[[1]] <- NA
```



Get materials: `usethis::use_course("rstd.io/tidytools19")`

Modifying an object creates a copy

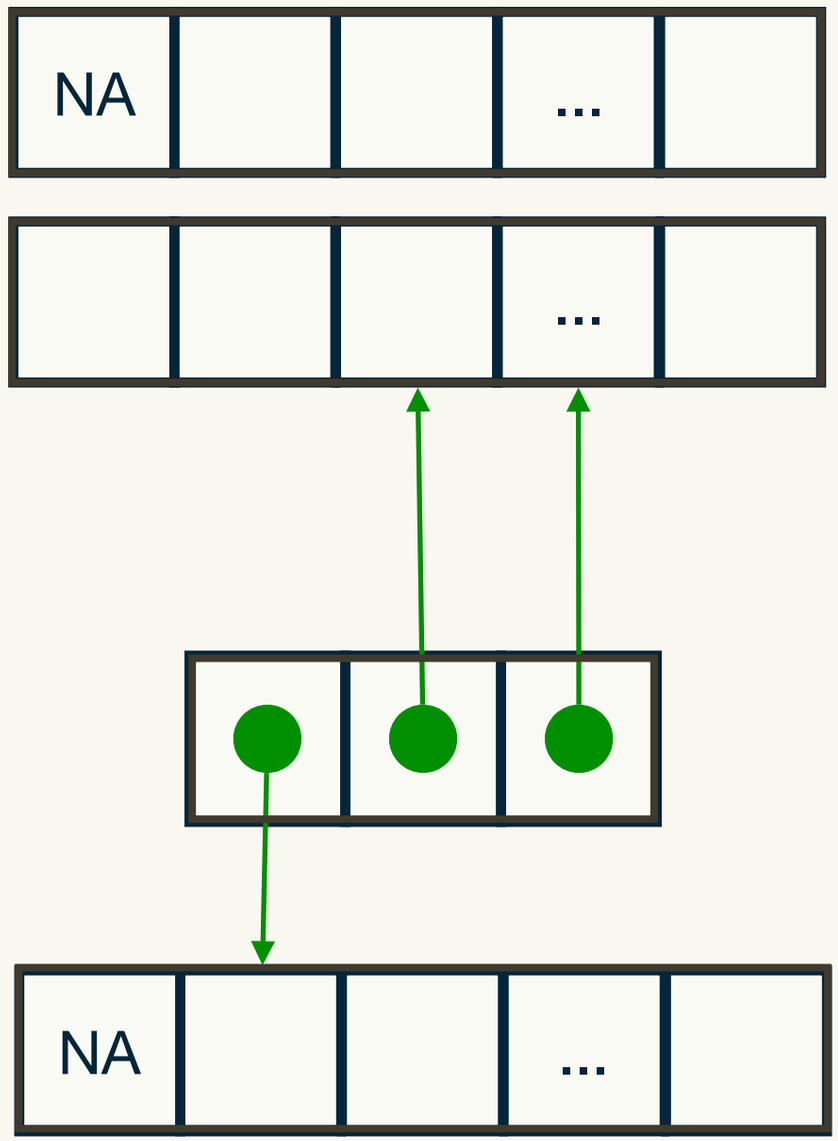
```
x[[1]] <- NA
```



Get materials: `usethis::use_course("rstd.io/tidytools19")`

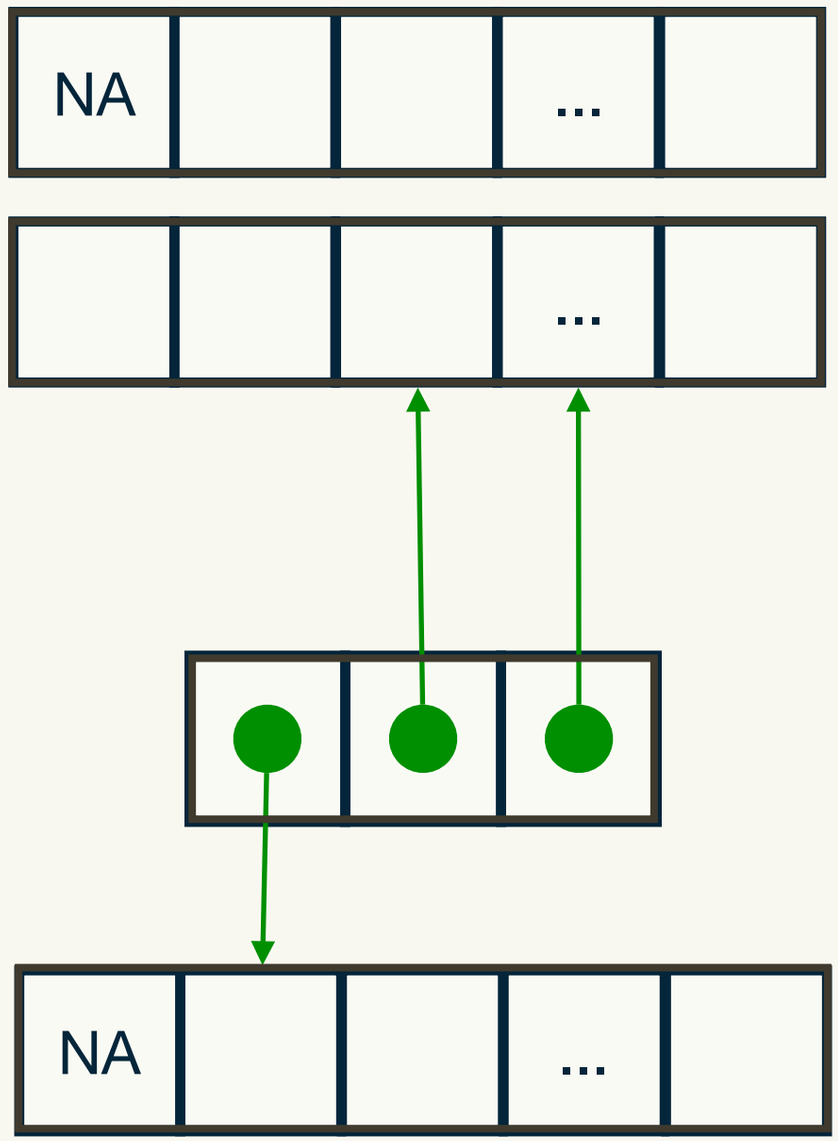
rm() removes references

```
rm(x, y)
```



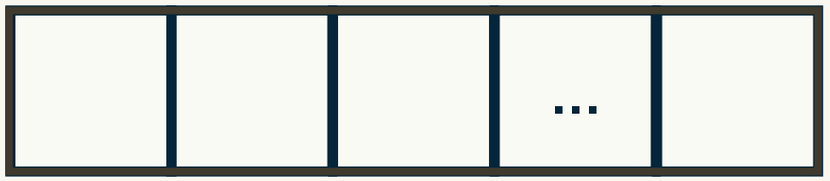
Get materials: `usethis::use_course("rstd.io/tidytools19")`

The garbage collector removes values



Get materials: `usethis::use_course("rstd.io/tidytools19")`

The garbage collector removes values

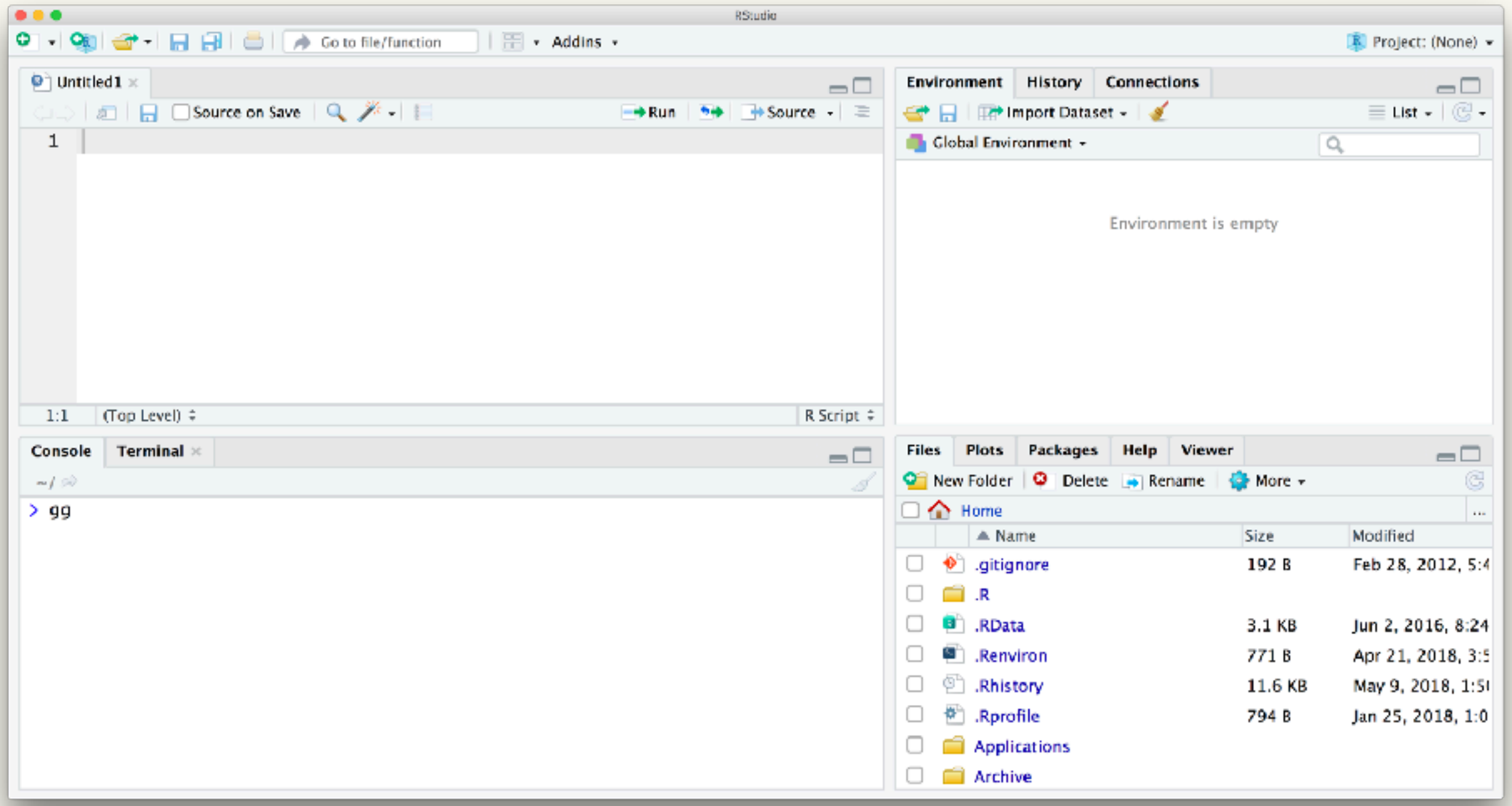


Get materials: `usethis::use_course("rstd.io/tidytools19")`

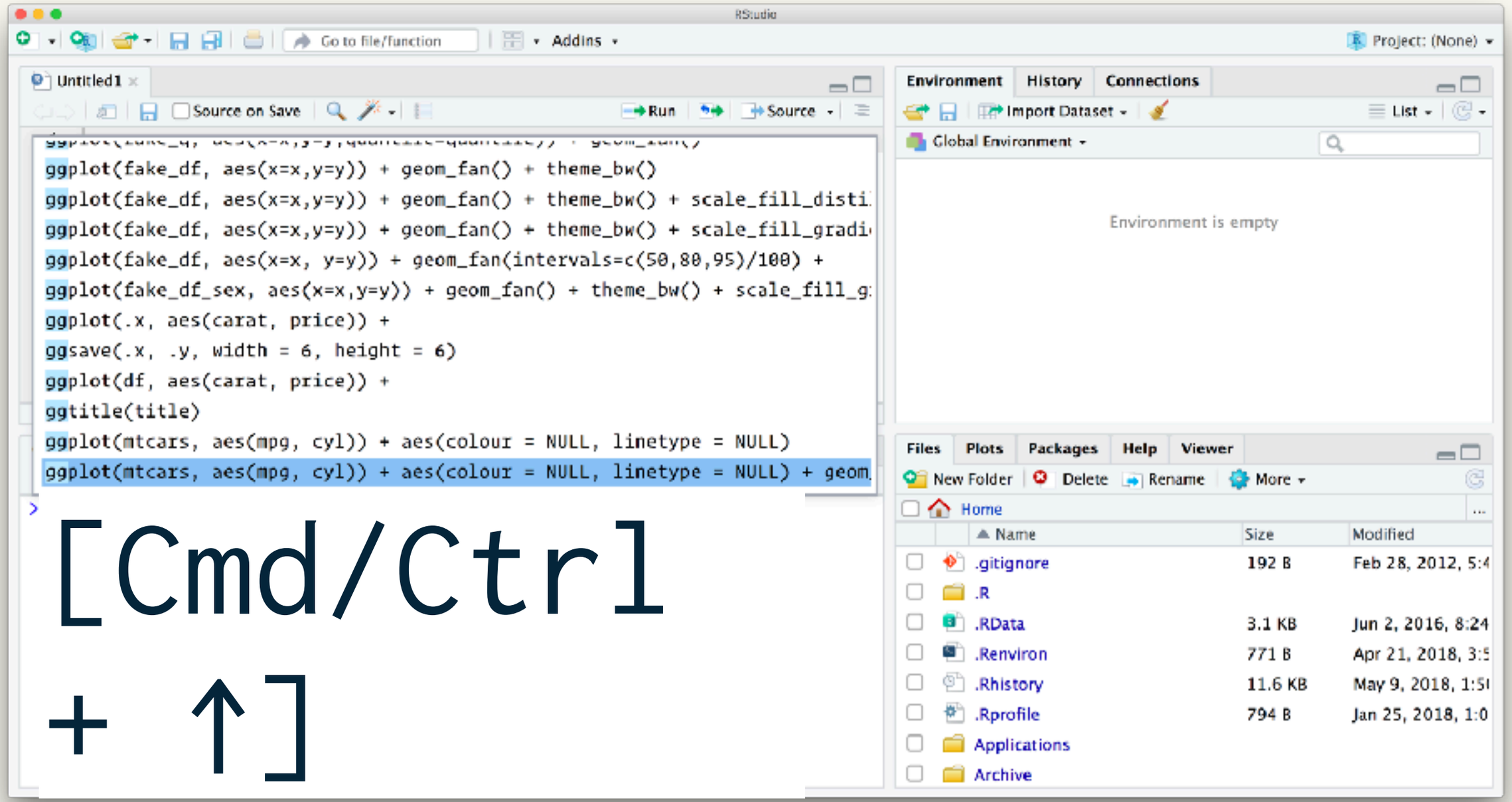
RStudio

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

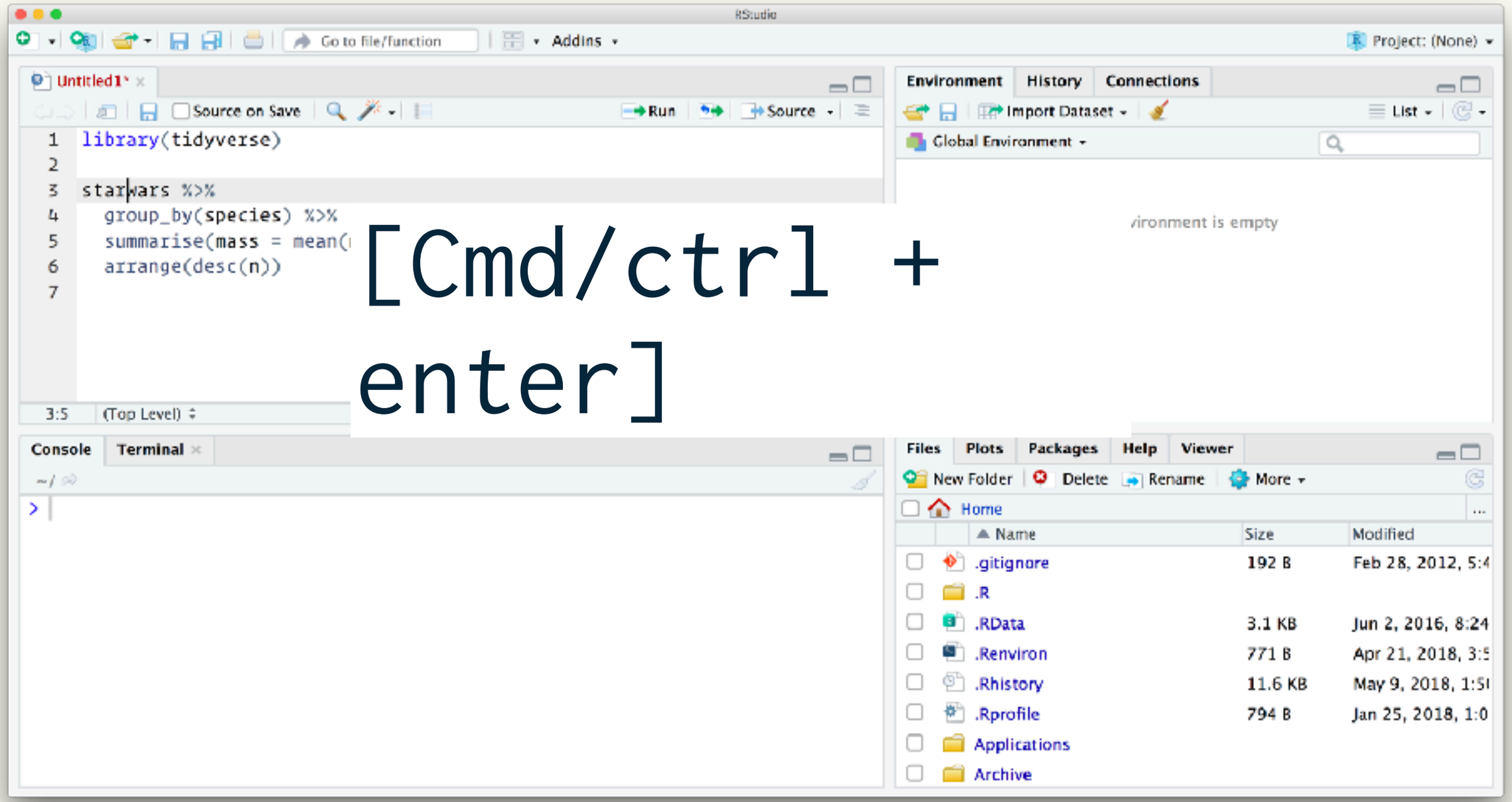
Get materials: `usethis::use_course("rstd.io/tidytools19")`



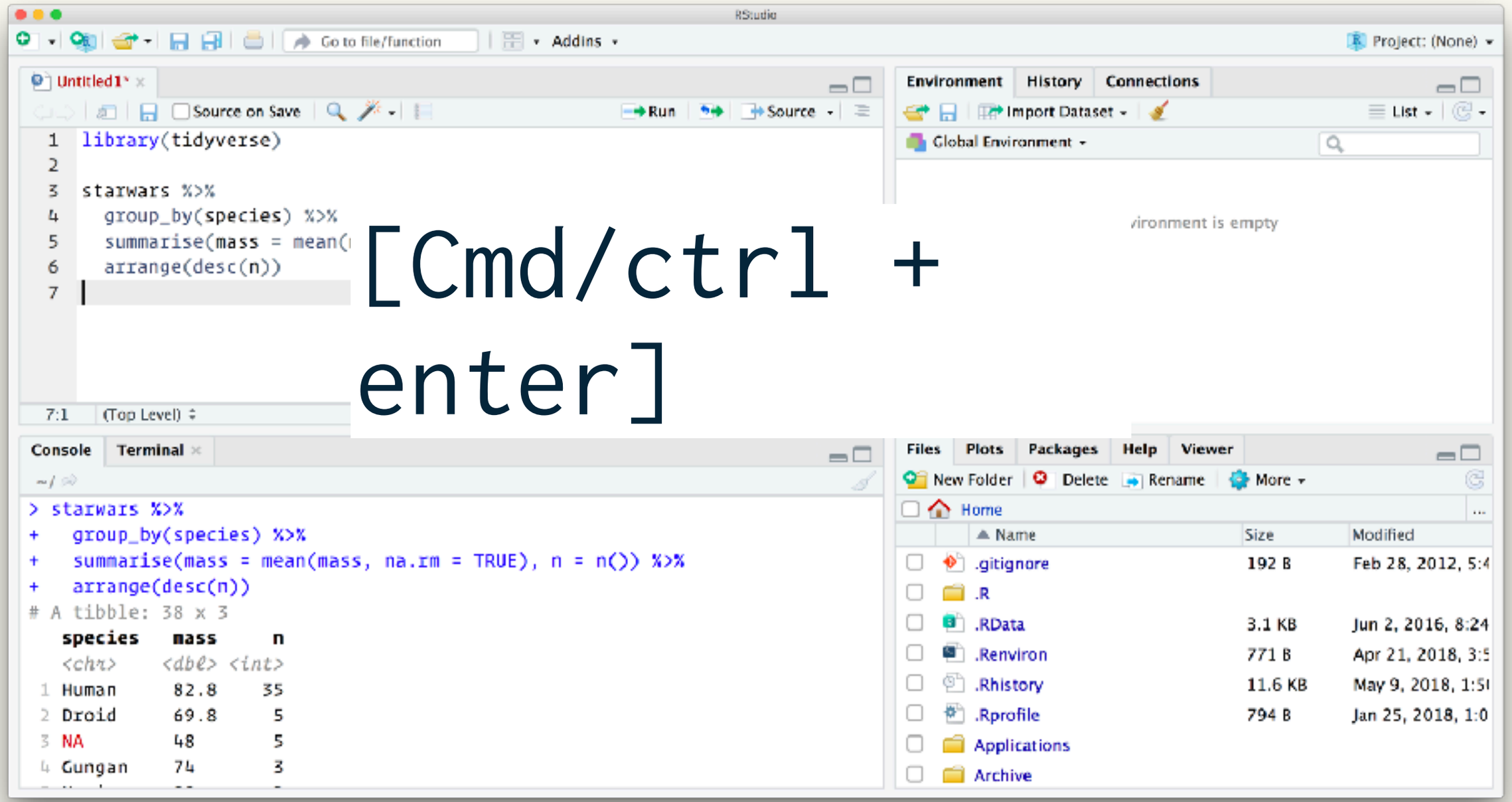
Get materials: `usethis::use_course("rstd.io/tidytools19")`



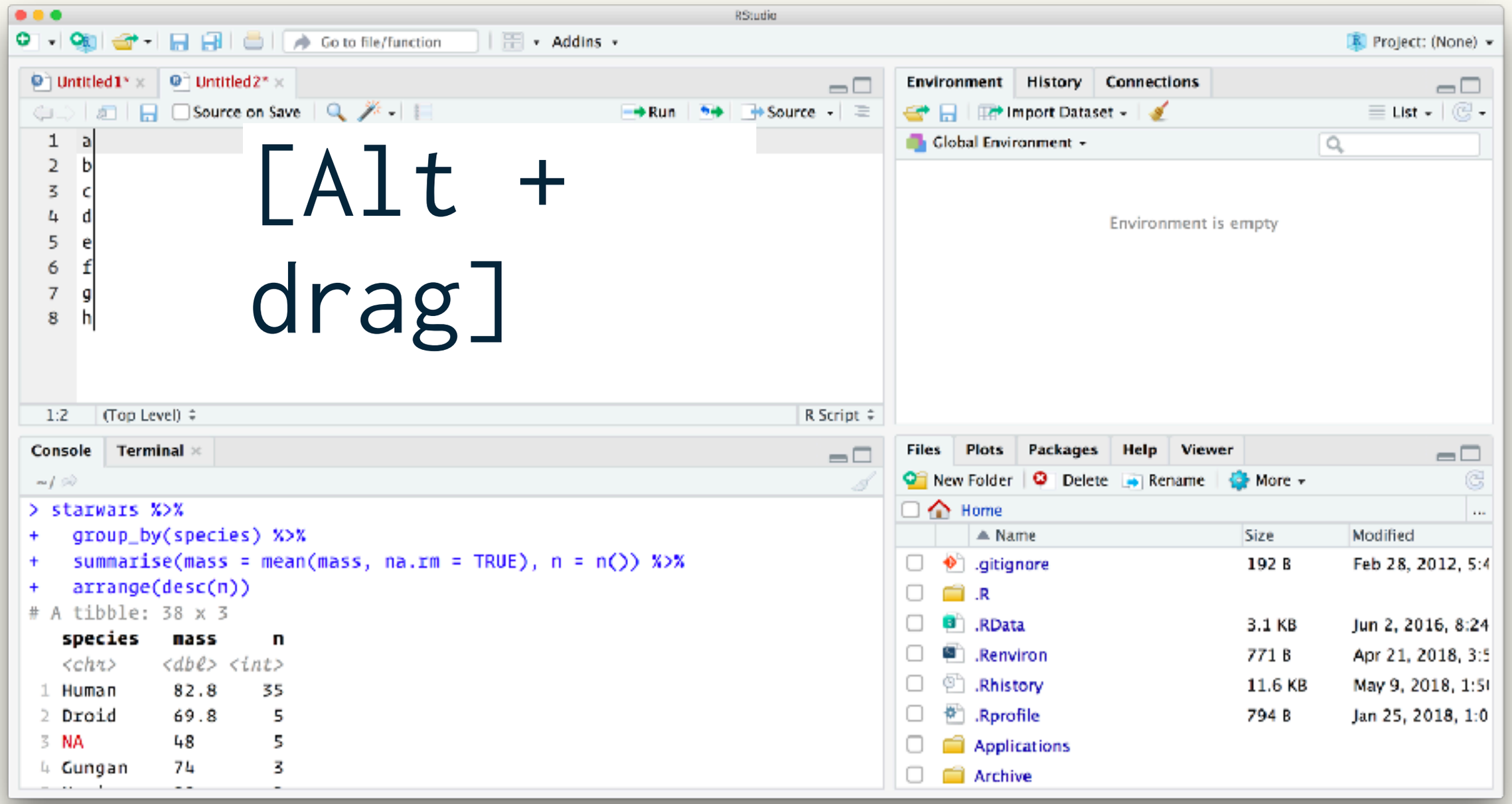
Get materials: `usethis::use_course("rstd.io/tidytools19")`



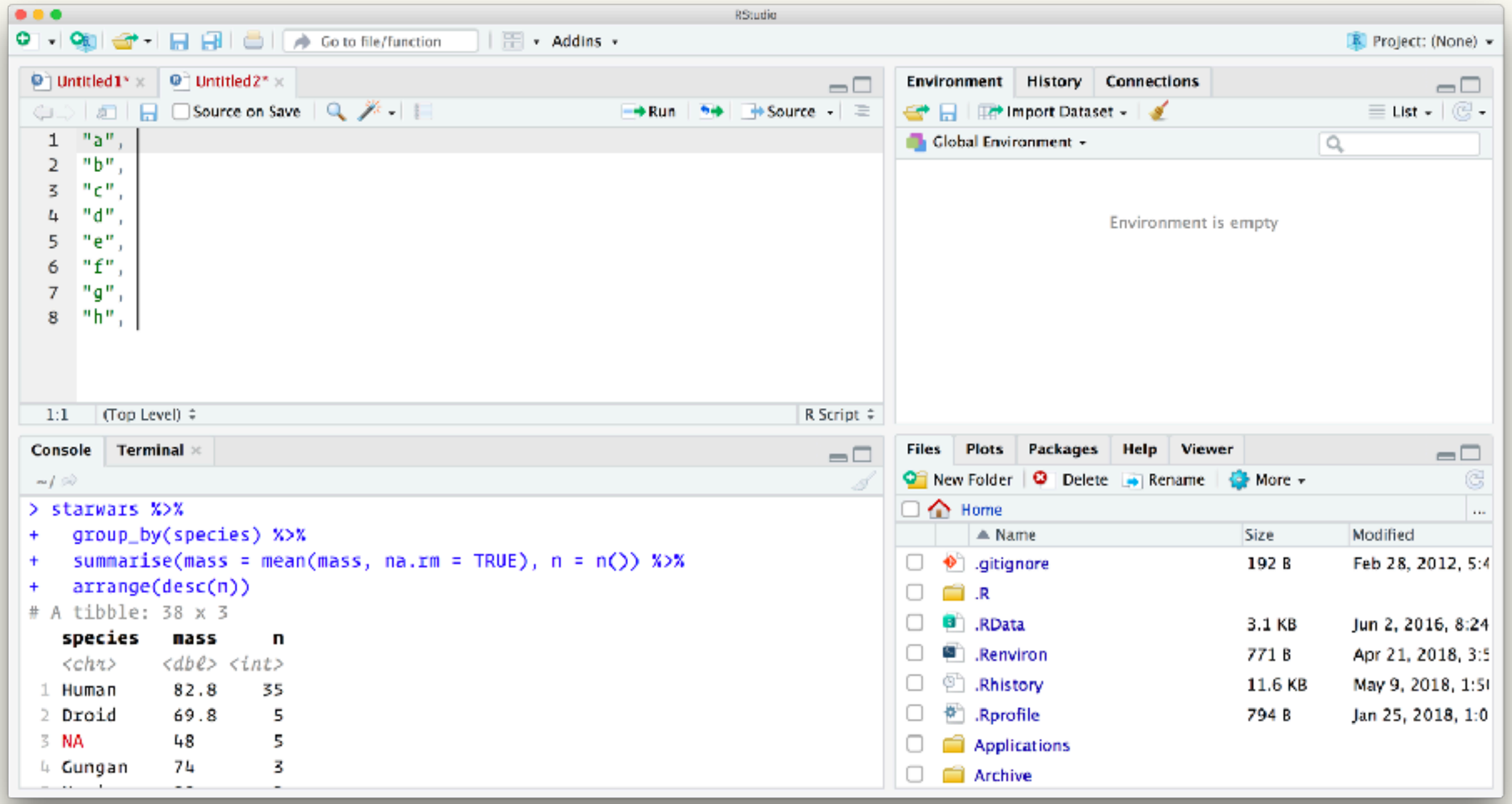
Get materials: `usethis::use_course("rstd.io/tidytools19")`



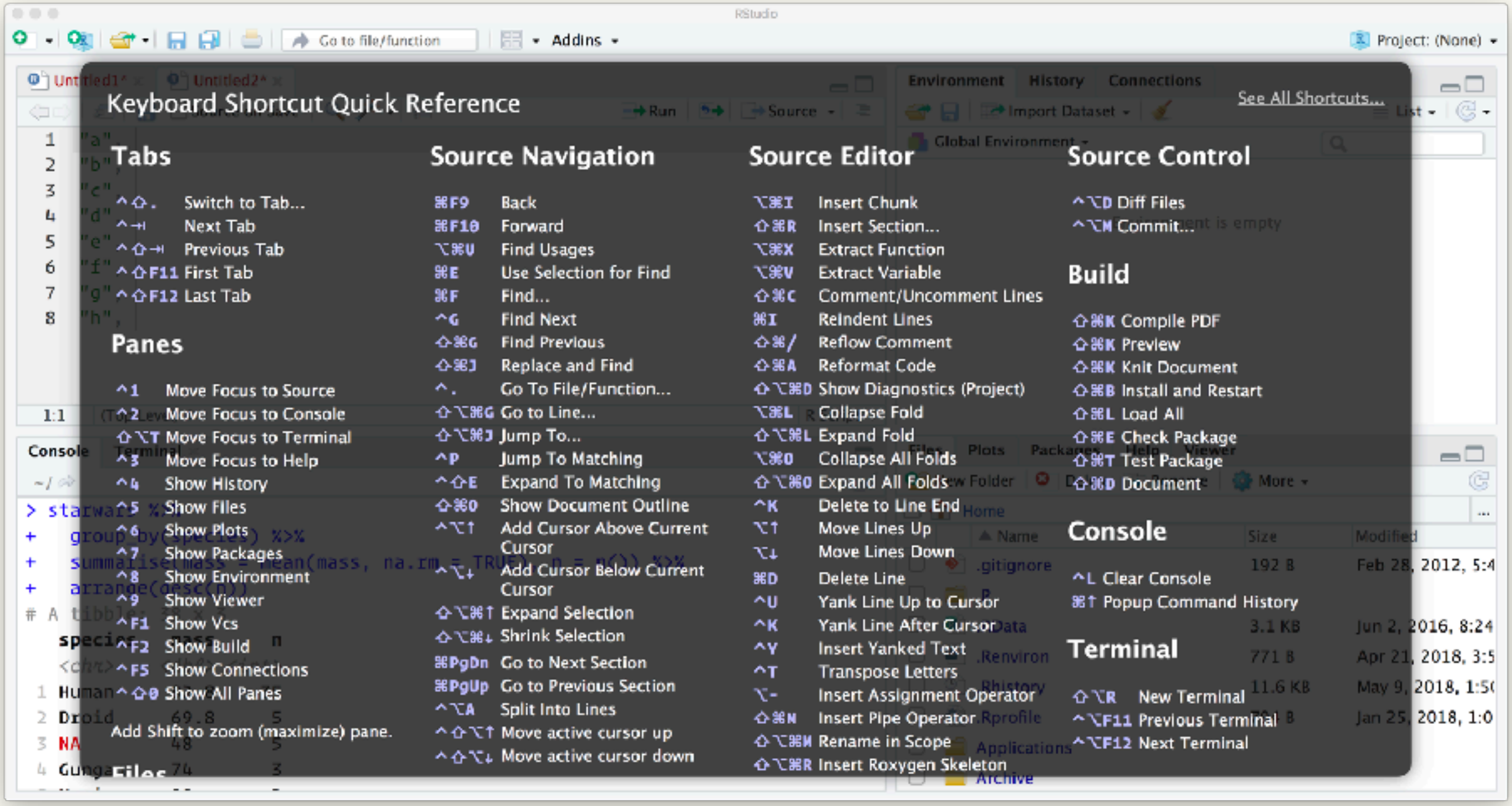
Get materials: `usethis::use_course("rstd.io/tidytools19")`



Get materials: `usethis::use_course("rstd.io/tidytools19")`



Get materials: `usethis::use_course("rstd.io/tidytools19")`



Get materials: `usethis::use_course("rstd.io/tidytools19")`

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("rstd.io/tidytools19")`

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/