Preliminaries in R

More on talking to R

“Composing” to combine function calls

What if you want to “compose” a call from more than one function call?

One way to do it is to assign the output from the first function call to a name and then use that name for the next call.

For example:

```
message <- paste("Hello", "world")
print(x = message)
```

```
## [1] "Hello world"
```

“Composing” to combine function calls

You can also “nest” one function call inside another function call. For example:

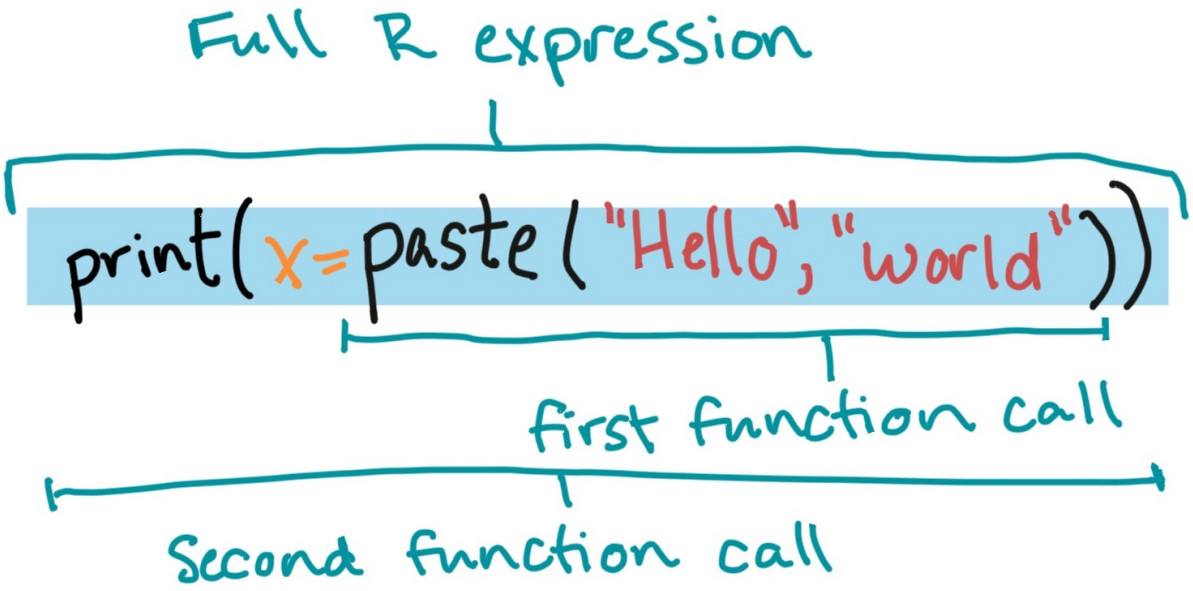
```
print(x = paste("Hello", "world"))
```

```
## [1] "Hello world"
```

Just like with math, the order that the functions are evaluated moves from the inner set of parentheses to the outer one.

There’s one more way we’ll look at later...

“Composing” to combine function calls



1. paste ("Hello", "world") ⇨ "Hello world"

2. print(x = "Hello world")

Using R scripts

The console can be great for quick functions to explore the data.

However, for most data analysis work you'll want to use a script, so you can save all the expressions you used for the analysis.

This improves the *reproducibility* of your analysis.

R scripts

An **R script** is a plain text file where you can write down and save R code.

When you write, run, and save your R code in a script rather than running it one line at a time in the console, you can easily go back and re-do exactly what you did again later.

You can also share the script for someone else to use, or run it on a different computer.

R scripts

RStudio has one pane that shows any R scripts you have open. If you'd like to create new R scripts, you can do that in RStudio with the following steps:

- Open a new script file in RStudio: `File -> New File -> R Script`.
- I recommend that you make an "R" folder in all of the R project directories that you create and save all your script files in that folder.
- Save scripts using the extension `.R`

R scripts

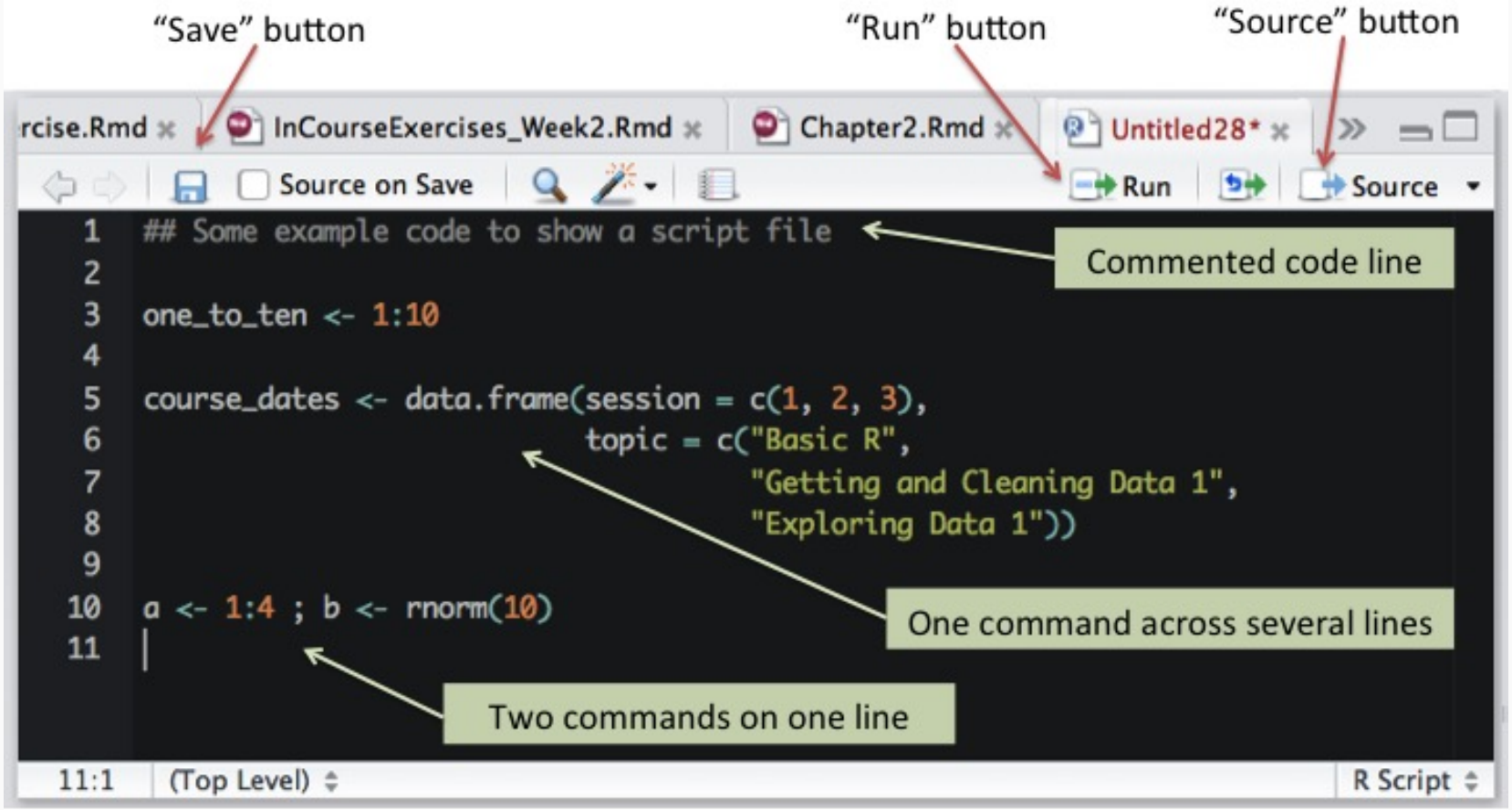
Running code in R scripts line-by-line:

- To run code from an R script file in RStudio, you can use the `Run` button (or `Command-R`).
- This will run whatever's on your cursor line or whatever's highlighted.

Sourcing an R script (i.e., running all the code saved in the script):

- To run the whole script, you can also use the `source` function with the filename.
- You can also use the "Source" button on the script pane.

R scripts


"Save" button
"Run" button
"Source" button
rcise.Rmd
InCourseExercises_Week2.Rmd
Chapter2.Rmd
Untitled28*
Source on Save
Run
Source
1 ## Some example code to show a script file
2
3 one_to_ten <- 1:10
4
5 course_dates <- data.frame(session = c(1, 2, 3),
6 topic = c("Basic R",
7 "Getting and Cleaning Data 1",
8 "Exploring Data 1"))
9
10 a <- 1:4 ; b <- rnorm(10)
11
Commented code line
One command across several lines
Two commands on one line
11:1
(Top Level)
R Script

Comment characters

Sometimes, you'll want to include notes in your code. You can do this in all programming languages by using a **comment character** to start the line with your comment.

In R, the comment character is the hash symbol, #. R will skip any line that starts with # in a script.

```
# Don't print this.
"But print this"
```

```
## [1] "But print this"
```

Closing an R session

Do **not** save the history of your R session when you close RStudio.
Instead, get in the habit of writing your R code in reproducible formats (R scripts, RMarkdown documents)