Q&A Material

Ari Anisfeld & Terence Chau

9/28/2020

What's the Deal with Fall Coding Lab?

Two tracks:

Accelerated: 2 lessons

- 2 lessons covering loops and functions.
- No final project (you already did it).

Not accelerated: 5 lessons.

- 3 lessons review summer camp material.
- 2 lessons covering loops and functions.
- Final project:
  - Find a data set that speaks to you.
  - Try to uncover something interesting. Graph it and tab it.
  - We'll give you feedback.

Logistics for Both Tracks

- Instructors Ari and Terence + wonderful TAs.
- 80 minutes per week: brief review and Q&A, then work in groups.
- Not graded.
- Access to TAs for coding specific problems throughout the quarter.
  - TA office hours 30 minutes before and after lecture time.
- Github website with all material.
  - We'll post solutions (eventually).
- Use Piazza for questions.
  - Rules of engagement: coding questions only, no Stats homework!
  - How to ask a good question?

Poll: How much coding experience do you have?

- First timer.
- Beginner.
- Intermediate.
- Proficient.

Please include your email.

Class 1: Why R? & Vectors

Key Points: R Basics

- Rstudio has a console to access R and a text editor to write code for reproducible projects.
  - Analogy: R is to RStudio as Tony Stark is to Iron Man's suit.
- R extensible through packages.
  - use `install.packages("")` once and then `library()` each session.
- Use <- to assign *any* object to a name.
- Functions take inputs and return outputs.
  - Input "understood" based on position or name.
  - Find out more about functions with `?` (e.g. `?filter`).

Questions

- Any questions on this? Feel free to ask on chat.

Key points: Vectors

- Vectors are the fundamental way to store data in R.
- We can operate on vectors element-by-element without loops.
  - `dplyr` verbs rely on this!
- We introduced built-in functions to build vectors and do operations on vectors.
- `NA`s are sticky!

Key points: Data Types and Coercion

- (Atomic) Vectors have a single data type.
  - Most often: `logical`, `integer`, `double`, or `character`.
- Certain operations expect a certain data type and R will try to coerce the data if it can.
- Usually, simpler types can be coerced to more complex types.
  - `logical < integer < double < character`.
  - Example on slides: `paste0(1L, "ing")`.
- Caution! Coercion can lead to unexpected behavior such as making `NA`s.

One More Thing

Logicals are coercible to numeric or character. This is very useful!

Determine the rule for how R treats `TRUE` and `FALSE` in math.

```
TRUE + 4
```

```
## [1] 5
```

```
FALSE + 4
```

```
## [1] 4
```

Questions?

- Any questions on this? Feel free to ask on chat.

Warm up & Lab 1

Warm up

- Solve the questions at the beginning of the lab in small (random) groups.

Lab 1

- Two "types" of breakout room:
  - Work along: larger group with more guidance from a TA.
  - Small groups: 4 people, TAs will come in and out to answer questions. Use the help button!
- Add "(work along)" or "(small group)" to your Zoom name so we can sort you.
- Get as far as you can, then finish it up after class.
- Before you leave, fill out the exit poll.

Lab 1: Exit poll

- What does `hist()` return?
  - A histogram plot of the data you give it.
  - A history of the commands you've run.