Biostat 561: Homework 1

Instructor: Amy Willis, Biostatistics, UW

September 28, 2017

Instructions and preliminaries

Question 0: Due by the end of office hours (Tuesday 4:00pm) via github classroom (instructions below).

Questions 1-7: Due 2:00pm Thursday 5 October via github classroom (instructions below).

- You may discuss the questions with other students or the instructor, but you should write up your own answers in your own words. The only exception to this is Question 0 Parts 1-7, which you are welcome to discuss in full detail with classmates.
- Instructions for uploading homeworks are contained in Question 0. Please complete Question 0 by Tuesday, and if you have trouble, please bring your computer to office hours on Tuesday at 3 p.m.
- Answer all questions in full sentences and paragraphs; supplying just code or its output is not a good-faith attempt to answer the question.
- Many of the questions in this homework were not discussed in class. They are research questions, intended to make you think about what happens "underneath" `R`. I expect you to make extensive use of online resources to complete these exercises.

Question 0

Please complete Steps 1-7 before Tuesday afternoon, and attend office hours if you are having trouble!

1. If you do not have `R` installed on your computer, install it. Google "how to install R" for instructions.
2. Install RStudio on your computer.
3. Download and install git and create an account on `github.com`. Choose a professional-sounding username!
4. Create a folder on your computer for this class, e.g. "Users/name/Documents/coursework/biost561/". Create 2 subfolders: "materials" and "responses".
5. In Terminal, go to "/materials/" and type `git clone http://github.com/adw96/biostat561 .`
6. Sign into our virtual classroom at `https://classroom.github.com/classrooms/32249780-biost-561`. Create a repository with your name or github username followed by "-561". eg. "adw96-561" or "amyw-561" would be ideal for user adw96.
7. Copy a blank (or random) pdf to "/responses/", and call it "homework1-response.pdf"
8. Read `https://stackoverflow.com/questions/6143285/git-add-vs-push-vs-commit` and `http://kbroman.org/github_tutorial/pages/init.html`
9. Initialise a git repository in "/responses/" (your local copy) to "[your-name-561]" in github classroom (your remote copy). Add the pdf "homework1-response.pdf", commit the pdf, and push it to your classroom repository. Google (or similar) error messages you get. You may need to add an SSH key to your computer.
10. Work on your response to Questions 1-4 in this folder. Overwrite the blank pdf as you go through your responses. Repeat Step 9 with your actual submission to submit your homework!

Our repositories on github classroom are private, so your classmates cannot see your solution (but I can). You should practice adding, committing and pushing your homework as you work (e.g. as you finish each question) to get into good habits! I will not begin grading your solutions until next Thursday,

Question 1: I don't know

`R`'s code for missing data is `NA`. Thinking of it as "I don't know" helps understand its behavior, e.g. `42+NA` returns `NA`; the sum of 42 and a number I don't know is another number I don't know.

1. Explain what these mathematical operations return, and why this is sensible:
    1. `FALSE & NA` – note that the "&" symbol is R's expression for a logical AND. See R's Logic help page if this is unfamiliar to you.
    2. `TRUE & NA`
    3. `TRUE | NA` – note that the "|" symbol is logical OR.
    4. `mean(c(3,5,7,NA))`
2. Why is the `is.na()` function needed to identify elements of a vector that are missing? In other words, explain why for vector

    ```
    myvec <- c(1,6,2,NA,500)
    ```

    we have to locate the missing value by using

    ```
    is.na(myvec)
    ```

    instead of

    ```
    myvec == NA
    ```

Question 2: Generic functions

Find and look at the code for the `plot` method for `data.frame` objects. (Hint: use the `find()` function to see which package it is in.) In your own words, briefly describe what you think each line of the code does. If you are unsure about what any individual line does, say so.

Question 3: Global environment

The Global Environment is `R`'s name for the space where objects are stored, by default. This is also the first place where `R` looks for objects specified in commands. The search path, i.e. the other places `R` looks for objects – usually packages – are listed in the output from search().

1. Start a new `R` session and load the MASS package, which contains data and functions from the classic book Modern Applied Statistics with S, by Venables and Ripley. Where in the search path is MASS added?
2. It's also possible to put data frames in the search path. For example, the command

    ```
    attach(women)
    ```

    adds the built-in data frame women to the search path. This data frame has 2 columns, height and weight, giving average weight (lbs) for women aged 30-39 of particular heights (inches). Suppose with women attached, you wanted to use heights in cm, not inches, and so used the following code;

    ```
    height <- height*2.54
    ```

By examining the contents of the environments in the search path, and using ls() with the pos argument specified, explain why these similar-looking commands then give different answers;

```
mean(height)
```

```
mean(women$height)
```

Attaching data frames is not recommended, as the user has to keep track of what's attached, and where in the search path they are. This is very error prone, and mistakes are difficult to debug. Use of the with() function is a better option – as you'll see in 514.

Question 4: Complex numbers

While you will probably use them only rarely, `R` can handle complex numbers:

```
class(0+1i)
```

Euler's formula gives us that $e^{i\pi} + 1 = 0$. Can `R` confirm this? If not, what's going on? You should research the binary accuracy of `R` to develop your answer.