Minute Paper after Class 3
Edited Summary / Analysis for class discussion 2018-09-06

Update: 2018-09-06, n=52 responses from 52 students

THANK YOU for responding and for your many kind comments about TEL and the TAs.

How confident are you about your ability to successfully complete Homework 1, due on Friday 2018-09-05 at noon?

Not at all confident	Somewhat confident	Very confident
29		23 (44%)

What was the most important thing you learned during the 431 class so far?

- Data Cleaning. The need to clean/scrub/tidy the data before proceeding to an analysis
- Data Visualization. The need to visualize or at least look at the data before proceeding to an analysis
- Doing something in R. How to do <specific thing> using R, R Studio, R Markdown
    - Creating a Histogram
    - Counting, finding means and other numerical summaries
    - Tidying the Data with filter, select, other verbs, plus the pipe %>%
    - Importing files, creating projects, establishing a consistent workflow
- R Markdown and using it to integrate all parts of a complete and replicable research document, rather than just coding in the R Console and copy-and-pasting results into some other document.
- Types of Data. The distinction between quantitative and categorical data (A related question: are haircut prices were discrete or continuous? - that's a distinction here without a meaningful difference in the eventual analysis.)
- Getting Help. All of the ways in which I can get help.
- Incomplete > Late for HW. The importance of getting HW in on time in this class so that the answer sketches can be provided quickly.
- A few pain points:
    - I cannot yet get <specific thing> to work properly on my laptop.
    - I am having trouble working with GitHub files.
    - There are many different codings to accomplish <some thing>.

What question (or questions) remain uppermost in your mind at this point?

1. What's the best way to learn how to program in R?
    - Do it yourself, and make lots of mistakes in small-stakes situations.
    - Review your work afterwards, and identify ways to improve your code.
    - Don't fear asking for help.
    - Recognize that, as with any language, pain accompanies your learning.
2. How do I memorize <this thing I need to do in R>?
    - Why memorize something? Either you're going to use it so many times that it will be something you memorize whether you want to or not, or you will have resources available to you any time you are coding something so that you can look it up, or tweak existing code.
    - Dr. Love uses Google every day to look things up in R.
    - So does Hadley Wickham, who wrote the initial versions of most of the tidyverse, which actually used to be called the Hadleyverse.
    - **Understanding** what you're doing, though, that is a big part of this course, but have a little patience. We're just getting started.
3. I could spend hours making graphs prettier. How much time should I invest in making them beautiful (publication level) for the homeworks?
    - It's a homework assignment. I wouldn't spend more than 10 minutes.
4. I have <some prior R experience>. Is it OK to reuse code from that experience in this class? Is it OK to do something in the Homework using R code we haven't learned yet in this class if we get to the same final destination?
    - Absolutely yes, on both counts.
5. There's so much material on the web site that I don't know where to start.
    - Outside of class, I encourage you to start with the Course Notes, for now particularly the Chapters in Part A, which will track with the class more starting today. Don't forget that there is a search function there.
    - In class, I encourage you to start with the README page (in the Slides section) for that day's class.
6. How do I include the "black and white" theme in an R plot?
    - What you want is the option theme_bw() as part of your graph. We've got an example in the Day 1 Survey analysis today, for example. Or you can look at an application in our Course Notes, for instance in section 3.4.2.
    - There are many available "themes" in ggplot2 - for a demonstration, see https://ggplot2.tidyverse.org/reference/ggtheme.html

7. Could you walk us through the verbs that the tidyverse uses?
    - Yes. I'll describe them a bit today in class and list them explicitly in the Day 1 Survey analysis we'll discuss, but I also strongly recommend reading the more thorough treatment in the section on Data Transformation in the R for Data Science textbook, and, for another example, see the NHANES material in Chapters 3-6 of our Course Notes.
8. Could you explain the difference between the two types of pipes we've seen, which are %>% and %$%?
    - Sure. These are special operators, called pipes, which pipe their left-hand side values into expressions that appear on the right side. The point of the pipe is to help you write code in a way that is easier to read/understand.
    - The %>% operator is the main one we will use all the time. It loads automatically as part of the tidyverse, and basically means "then".
    - The %$% operator is a specialty operator that "explodes" out the variables in a data frame so that you can refer to them explicitly. This is needed when you're working with functions, like `cor`, outside of the tidyverse.
    - For more details, see the Pipes chapter in the R for Data Science textbook.
9. Missing Data: mean(varname, na.rm = TRUE) - What's that na.rm = TRUE stuff?
    - That's a bit of code that tells R "in finding the mean of this thing called varname, remove the missing values before doing the calculation."
    - For more on missing data, see Missing Values in R for Data Science.
10. Missing Data: What does filter(!is.na(varname)) mean?
    - It means filter out all observations where varname has a missing value.
    - Actually, filter to keep the observations where varname is NOT missing..
    - More commonly, we'll use filter(complete.cases(varname)) which is at least a little easier for human beings to read, I think.
11. Why do I need two "=" signs sometimes, and not others?
    - The statement x == 1 is evaluated by asking if x is equal to 1.
    - The statement x = 1 sets x equal to 1.
    - Also, not all R code was written by the same person at the same time, so this is absolutely a potential pain point.
12. A list of commonly used codes would be great - reasoning behind them?
    - In combination with each class' Slides, the Course Notes describe and demonstrate all code you'll need to succeed in this course.
    - That's not a "list" of codes, you understand, more a set of analyses using these codes. I expect you'll generate a list of codes you personnally find most useful as Part A moves on, and you complete Homeworks 1-4.

13. **What is the best way to earn class participation credit?**
    - **Unless you're about to apply to medical school, please try to relax about grades. It's graduate school, and grades are no longer the key issue.**
    - **But the best way is to actually participate in the class, in any way that you feel comfortable doing.**
    - **I am delighted to take questions during class time, as well as before and after, although in these early sessions, it's occasionally important to postpone discussion and move forward.**
14. **A Windows problem: I'd like to know to download R Markdown files that were not included in the initial zip download. I'm able to download them as txt files, but is there a way to download it with the .rmd extension?**
    - **If you right-click on the raw version of an .Rmd or .csv file, you should be able to download the file with the extension intact.**
15. **Can R be used to do <this analysis we haven't even approached discussing yet>?**
    - **I expect the answer is Yes. That's my default answer.**
16. **Why are we learning R in this course, instead of <some other software>?**
    - **Because it is by far the better choice for what we're trying to do, which is to help you become effective data scientists. And effective scientists, period. There are lots of specific reasons.**
    - **Here is a quote from Greg Snow, comparing R to SPSS...**
      "When talking about user friendliness of computer software I like the analogy of cars vs. buses:
        - Buses are very easy to use, you just need to know which bus to get on, where to get on, and where to get off (and you need to pay your fare).
        - Cars, on the other hand, require much more work, you need to have some type of map or directions (even if the map is in your head), you need to put gas in every now and then, you need to know the rules of the road (have some type of driver's license).
        - The big advantage of the car is that it can take you a bunch of places that the bus does not go and it is quicker for some trips that would require transferring between buses.
        - Using this analogy, programs like SPSS are buses, easy to use for the standard things, but very frustrating if you want to do something that is not already pre-programmed.
        - R is a 4-wheel drive SUV (though environmentally friendly) with a bike on the back, a kayak on top, good walking and running shoes in the passenger seat, and mountain climbing and spelunking gear in the back.

      R can take you anywhere you want to go if you take time to learn how to use the equipment, but that is going to take longer than learning where the bus stops are in SPSS." Source: http://stackoverflow.com/questions/3787231/r-and-spss-difference

17. This seems hard. Will I be able to do this?
    - Sure. Why not? You've done many harder things.
    - Others have been in the same situation you find yourself in, and they have succeeded. You are not an imposter, and you are welcome here.
    - And we're here to help you.
18. Are you a fox or a hedgehog?
    - Like everyone else, it depends on the decision I'm faced with at the time, and on other things.
    - I aspire to think like a fox most of the time in my professional work.
    - This should be clearer after you read some of *The Signal and the Noise*.