Programming in Statistics: Final Project

Due: 13:00 on February 15, 2019
We must meet to discuss the project no later than February 27, 2019

The project has two parts: a data exercise and understanding/explaining more complex code. The data exercise will be similar to homework 2 except that you also need to do some simple modeling and I have higher expectations for quality of work. It would behoove you to discuss your data sets with me. In the second part, you will explain code that I have written that implements a revisiting testing algorithm. As it is my own unpublished code, you won't find anything helpful online.

You should turn in the following: a zipped folder of your data, an .Rmd and .html file for your data analysis, an .Rmd and .html file for your explanation of my code, and an updated projectPoly.R file which includes your comments. If you answer any of the extra credit questions, please use a separate .Rmd/.html file.

Grading Schema:

- ...doing it correctly. I will be more strict than with homework, but there will be some acceptable variability in answers.
- Clarity of project and code: clear structure, appropriate text mixed with code, commented code when helpful.
- Quality of discussion. I'm not grading the writing itself (per-se), but the thoroughness and detail. If the writing is not clear enough for me to follow, however, that is still a problem.
- Coding style: naming, structure, format (both large scale and small scale), using loops and functions to reduce duplication, indentation, modularized code, etc.
- Graphics and communication of results.
- If desired, you can write your project in German. A word of warning: don't assume that by writing in German you can "slip-by" during grading. Given my German skills, I will take *more* time to grade projects in German. You also won't have the excuse of "this isn't so clear because my English isn't great."

1. (15%) Finding data! The hope is that you are able to find data on something you are passionate about. Football? Board games? Travel? Sustainability? Education? Psychology? Biology? I guarantee you can find something worthwhile. Re-read homework 2 and consider the tasks of question 2 below. If you choose a bad data set, the remainder of the project will be difficult and boring. This is actually worth a decent amount of credit because finding good data is hard. You also need to describe/introduce your data in the introduction. Your data must satisfy the following requirements:

   (a) Your data "set" needs to include at least 2 related files (preferably 3, depending on the types of joins required). You do not need to use data which all come from a single source. Be creative. For example, if you are analyzing country data, you could perform a "per continent" analysis. You can find a new file with country-continent information. A single data base can contain multiple tables, and these will be considered distinct as you still need to merge them for your analysis.

   (a2) *If* your data includes significant unstructured data (text from which you must extract information using regular expressions), then you do not need to have multiple files.

(b) You need to be able to ask "statistically relevant" questions: how are X and Y related to Z? How can I visualize the joint distribution of X and Y? Does the distribution of Y depend on class membership C? How does missing data affect the analysis? etc.

(c) Potential data sources include:

- http://data.un.org/Explorer.aspx?d=UNESCO
- https://data.oecd.org/searchresults/?r=+f/type/datasets
- https://www.data.gov/
- https://data.worldbank.org/
- https://www.weather.gov/
- www.google.com :)
- Though I used it for class, you *cannot* use https://www.kaggle.com/datasets, because there is often code available for the type of analysis I'm expecting (see below). The only exception is if you can prove that you are doing something that hasn't been done before and posted online in the kernels. This is possible if you focus on a suitable subset of the data for example.
- You *cannot* use data from packages in R; you must load data from files or data bases.

2. (50%) You have data...now analyze them :) Some pointers:

(a) This is an *exploratory data analysis* exercise. There is an additional chapter on this in R for Data Science if you would like further discussion. We have done this multiple times throughout the course, though I may not have called it EDA.
https://r4ds.had.co.nz/exploratory-data-analysis.html

(b) Think about the *structure* of homework 2: merge data sets, consider missing data, simple graphs to understand your data distributions, generate some hypotheses about the relationships between your variables, draw appropriate graphs.

(c) Justify and estimate some linear models to explore your hypotheses in more detail.

(d) Communicate your results. This requires both graphs and text. Your graphs need to be good. Each group will be analyzing a separate data set and *you* will be explaining it to *me* through your project.

(e) Here are two examples from Kaggle that you should consider as "inspiration." They spend more time on modeling than you need to, and use graphs that we have not discussed; do not worry about these things. Focus on the structure: introduction, preparation, exploration, modeling; note how each section includes code (when shown), output, and discussion/results; observe that they "walk you through" the data.
https://www.kaggle.com/headsortails/shopping-for-insights-favorita-eda
https://www.kaggle.com/headsortails/steering-wheel-of-fortune-porto-seguro-eda

3. (35%) I have posted a file with code online (projectPoly.R). I have commented and explained the statistical components of the algorithm. Your task is to explain *how* computations are managed/performed. Warning: it's easy to "half-ass" the answers here, but that won't get you half the points!

(a) In a second .Rmd file, explain what every function does. Importantly, you need to include the *environment* in which the function is bound and the other variables in this environment. Alternatively, you could draw a picture of all of the environments which enclose functions which are bound in the expert list. Both capture the same information (though the picture is more succinct).

(b) Comment the code thoroughly. After part 1, you should understand how the code works. Now comment it to help explain it to others. Essentially every line should be commented. Remember, you aren't explaining what the code does, eg, for line 42, "ifelse checks the condition, bids wealth*frac if it holds, otherwise wealth." Instead, explain *why* things are done as they are, eg, "ifelse statement ensures termination by eventually bidding remaining wealth". Note that some things are worth explaining a bit of the *how*. For example, line 95, "+1 accounts for the zero at the beginning of the activeColumns vector".

(c) The state function returns the values of the private variables. Why is this a function that returns a list and not just a list of the values of the private variables?

(d) Extra credit (10%). Generalize the code to use multiple experts (store the experts in a list). While the code should be general, test it using two experts; one expert should use a geometric bidder and the other should use the constant bidder. Start each expert with wealth = omega/2. The expert with the maximum bid "wins", and gets to perform its test. Note that each expert has private bidder and constructor objects (why?). Only use *one* makeExpert function (you should pass it an argument for which bidder function to use).

(e) Extra credit (5%). Suppose I start the two experts from part d with the *same* constructor object. How does this set of experts behave compared to the ones in part d? How does this help you design bidding strategies?

(f) Extra credit (10%). Make both experts from the part d use a single wealth object. This will require you to create a separate wealth object that is passed to both of the bidders, and modify the bidder objects to change the wealth object appropriately.