Stat 27850/30850: Group project # 1

Data The data set for the first project is the Capital Bikeshare data set, from `https://www.capitalbikeshare.com/system-data`. This data comes from Washington D.C.'s bikeshare program, which records every individual ride taken. Each ride is tagged with its start and end time, start and end station (there are hundreds of locations where bikes can be rented across the city), as well as whether the rider has bought a one-time rental or is a member of the bikeshare program.

On Canvas, we provide a script named `getdata_bikeshare.R` to download and clean the data so that it's ready for R, and organized in a simple format. (A Python script is also available, `getdata_bikeshare.ipynb`.) The variables you will have after running the script are:

- Data for all $n = 1342364$ rides in 2010 & 2011.
- `starttime`, a $n \times 6$ matrix with the start time of each ride. The columns are: year/month/day/hour/minute/second.
- `duration`, a length-n vector indicating the duration of the bike ride, in seconds.
- `station_start` & `station_end`, length-n vector giving the station number where the ride began / ended.
- `stations`, a matrix matching station numbers to physical locations (you can choose to use this information if you would like to incorporate geographic information / calculate distance between stations / restrict to a particular geographic region / etc). Note that stations get added to the program over time (i.e., not all stations were already in use at the beginning of 2010).
- `bike_num`, a length-n vector giving the ID number of the bike used on each ride. Some are NA (unknown).
- `member`, a True/False length-n vector indicating whether each ride was taken by a paid member of the bikeshare program or a one-time purchase.
- `days_since_Jan1_2010`, a length-n vector measuring the date (i.e., the date at the start of the ride) in terms of days since Jan 1 2010.
- `day_of_week`, a length-n vector indicating the day of the week, Monday/Tuesday/etc.

The script will then save these variables in a file.

Assignment Your task is to study the following question: can you detect any routes (i.e., a particular combination of start & end station) where the average time it takes to travel the route, changes over the course of the time period? For example, if a bike lane is added to a major road, this may reduce the travel time for that particular route.

One simple approach we could take, would be to find routes that seem to show an increase (or decrease) in duration over time, then permute the vector of durations for that route and see if the apparent increase (or decrease) is likely to appear equally strong even when the data is randomly shuffled.

However, there are many possible confounders that are not accounted for by a simple permutation scheme. For example, members may ride faster than one time users; the day of the week can affect traffic and therefore change the speed of the ride; difficult weather can slow things down; etc. This is only a suggested list—you're welcome to frame the problem differently, and your group might think of other important confounders as well—be sure to explain your decisions in your report. (For weather—you are welcome to look up weather records from those dates, but this is completely optional and may not be feasible given the limited time—we recommend that you treat weather as unknown but constant over any single day / any single hour / etc.)

Your task is to find a way to reliably identify which routes (if any) show changes. There will be many possible ways to address this. You can think about how to use task-specific permutation strategies (rather than permuting at random), or you can use regression models or other tools instead of permutations. Your final report should give a thoughtful discussion of the issues you have identified, and the strategies and methods you developed to address them. There might be confounding effects you identify that it's not possible to address with the limited data available (or simply due to time constraints); your report can also mention issues that you were not able to address, and assess to what

extent you think it might affect the validity of the analysis. Depending on your approach, it may happen that you are or are not able to detect routes with significant changes. Either outcome is fine, as long as your conclusions and methods are well explained and justified.

Guidelines Groups of size 2, 3, or 4 are allowed for the mini-project. The extent of the project (e.g., the range of questions explored / methods tried / etc) should be proportional to the size of the group. What you hand in:

- Each group should hand in a written report and either include code throughout the report and/or include the code as an appendix. Please designate a single group member to submit everything on Gradescope, and add the other students in the team group members.
- For your code, it should be clearly organized and commented— for example, you may want to label sections of the code so that we can see which part of the report or which plot/table it corresponds to, add comments to explain steps where notation / variable names / nature of the calculation aren't obvious, etc.
- There are no page length or formatting requirements for the written report. Your report should describe the problems and questions you posed, the details of any methods you implemented / models fitted / hypotheses tested, describe your findings and show plots or numerical results as appropriate, and should discuss some interesting issues relating to inference (for example, multiple testing / appropriately controlling for confounding factors / reducing a high dimensional model to a manageable size / etc). You can also include a discussion of open questions and issues that were not addressed (due to time limitations and/or limitations of the available data).