STAT 19000 Project 10

Topics: R versus bash/awk

Motivation: In this project, we synthesize the topics that we have learned so far this semester. We cut some data from some very large files, gather the results from across these files and store the desired data into a smaller file, and finally import the data to R for visualization.

Context: It is often the case that we do not want to import enormous data sets into R directly. Instead, we want to do some pre-processing with other tools (for instance, with the terminal) that are extremely fast.

Scope: The full data analysis relies on learning several kinds of tools, and using the tools in tandem to accomplish large tasks.

How to format your solution: Use the `project10template.R` file from the `/class/datamine/data/examples` folder. (Of course, question 1 will have UNIX commands and questions 2 and 3 will have R commands.)

Question 1: Cutting data down, using UNIX

1a. Use the UNIX `cut` or `awk` tool to extract (only) the dates and transaction amounts from the election data, across all years. Save this data into a file in your home directory.

Hint: You can use: `cat /class/datamine/data/election/*.txt` to display all of the data from all of the election years, and then use a pipe to send this data to the `cut` command. You can use a right carrot to save the resulting data to a file. There should be 69018140 lines in the resulting file. You can check this with the `wc` command.

1b. Remove the lines from the file that came from the headers of each election file, i.e., remove the 21 lines of the form: `TRANSACTION_DT|TRANSACTION_AMT` Save the result into a new file.

Hint: You can do this use something like the following: `grep -v TRANSACTION myfile.txt >mynewfile.txt` because the grep command will avoid all patterns that you specified with `-v`

Question 2: Preparing data to display in R

2a. Import into R the resulting file that you prepared from question 1.

Hint: Please use `read.delim` instead of `read.csv` and use `sep="|"` as an option in the `read.delim` command. Use `colClasses=c('character','integer')` because you want to ensure that we keep the format of the dates from the first column of your text file. Use `header=F` because our data does not have a header.

2b. Make a new third column of the `data.frame` that has the same data from column 1, but is stored in Date format: To do this, you need to use the `as.Date` function. It is helpful to read the examples at the end of the documentation: `?as.Date`

Question 3: Displaying data in R

3a. Make a table of the number of donations given per day. Then plot the values in this table. You might want to cut off some of the first several and last several dates, which are probably erroneous dates.

3b. Use the tapply function to sum the values `in myDF$V2`, according to the values in `myDF$V3`. Then make a plot of the resulting values. Again, you might want to cut off some of the first several and last several dates, which are probably erroneous dates.

Project Submission:

Submit your solutions for the project at this URL: https://classroom.github.com/a/JSnP_rFU using the instructions found in the GitHub Classroom instructions folder on Blackboard.