STAT 19000 Project 8

Topics: Pipes in bash

Motivation: Pipes in bash allow us to combine the strength of several UNIX tools, used in tandem. They also allow data to flow readily from one tool to another, without intermediate files or hurdles of any kind. The data science workflow becomes more seamless, when we start to think about how natural pipes are used.

Context: We begin to think about pipes as follows: We take a large data set, which is too unwieldy to display all at once time. We cut some of the data that is interesting for us, sort that data and find the number of elements of each type, and finally sort the results in numeric order. Walking through some examples is helpful in illustrating the power of such an approach. Moreover, we can do such analysis without loading full data sets into R (which takes a great deal of time).

Scope: After we become familiar with the concept of pipes, the idea of the data analysis cycle becomes readily apparent. Each tool has its own role to play in the larger process of analyzing a huge data set. When used together, pipes allow us to harness a great deal of power, directly from the terminal.

Question 1: Learning about pipelines

Before you start question 1, please consider the examples given here:

`/class/datamine/data/examples/project8examples.txt`

Use this template to submit your solutions:

`/class/datamine/data/examples/project8tempate.txt`

You can either use Accessories > Text Editor in the Applications menu, to open that template file for your solutions, or you can open an editor in the Terminal using this command: `gedit&`

This is a week of review, in which you can use the terminal to verify the work that you accomplished earlier in the semester.

1a. Use a pipeline in the terminal to solve Project 3, Question 1a, namely:

Make a table of the number of transactions that occur in each of the four stores in the 84.51 data set called `5000_transactions.csv`, found in the folder The_Complete_Journey_2_Master.

1b. Use wget to download the file:

http://stat-computing.org/dataexpo/2009/airports.csv

Then use a pipeline in the terminal to solve Project 4, Question 1c, namely:

Find the 4 cities that have the most airports.

Question 2: More learning about pipelines

2a. Use a pipeline in the terminal to solve Project 4, Question 2c, namely:

Which pickup location ID was the most popular for yellow taxi cab rides in June 2019?

2b. Use a pipeline in the terminal to solve Project 4, Question 3c, namely:

Which city has the largest number of donations so far, in the 2020 election season?

Hint 1: Remember to use the city and state together.

Hint 2: As we did in the last example from the example file, remember that the election data is not comma-delimited, so we do not use `-d,` but rather we use `-d\|` to specify the delimiter.

Question 3: Comparing the speed of bash pipelines versus R

When I run the code for questions 1a, 1b, 2a, 2b in the terminal, on Scholar, all at once, it took less than 45 seconds altogether. (You might want to try this yourself!) This is a big reason that we sometimes need to write code the terminal. It is extremely fast and low-level.

For comparison, paste the code need to run all 4 of these questions in R, into a new R document:

Project 3, Question 1a

Project 4, Question 1c

Project 4, Question 2c

Project 4, Question 3c

(Please paste the R code for all 4 of these, into your solution template.) Then run those lines of code in R.

3a. The question is: How long does the code for these for solutions take to run (altogether) in R?

(Hint: It should take much longer in R than the 45 seconds used to get the solutions in the terminal.)

Final Reflection/Observation

The reason is that R is a high-level language. Once we know how to code in R, it is slower to run but much more powerful. R has analysis and graphical capabilities that far exceed the primitive things that we can do in the terminal.

This is one of the main reasons why we have a data analysis cycle. We want to cut the data into pieces, using the terminal, in a very fast way. Afterwards, we want to be able to import much smaller pieces of data into R, so that we can do powerful analysis. There are more parts of the data analysis cycle, but this is your first glimpse at some of the reasons why we combine the power of multiple tools, when we are doing data analysis.

More exciting things to come!!

Project Submission:

Submit your solutions for the project at this URL: https://classroom.github.com/a/kKvYxIp1 using the instructions found in the GitHub Classroom instructions folder on Blackboard.