STAT 19000 Project 7

Topics: Getting familiar with UNIX/bash

Motivation: We are familiar with the ability of R to import data that can readily be analyzed and visualized. For comparison, it takes a lot of time to import data into R, and sometimes we do not need an entire data set. R also needs extra memory to store data, so that it is prepared to perform its many types of operations on the data directly.

Context: Using the terminal, we can write commands in bash (Bourne-again shell). This allows us to be more nimble with our analysis. We can quickly answer simply questions in bash, without the overhead of loading the data into R.

Scope: The commands in bash each seem to be simple, but there are several different commands, each with very different purposes. We will only study some basic bash commands, one at a time, in this project… but in the next project, we will learn how to use several commands in tandem, which gives us the ability do a great deal of data wrangling.

Using some data analysis and data wrangling in bash–before importing the data into R–will help us to better understand the full scope of the data analysis cycle.

Question 1: Learning about the cat, wget, wc commands

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

`/class/datamine/data/examples/project7examples.txt`

Use this template to submit your solutions:

`/class/datamine/data/examples/project7tempate.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&`

1a. Display the stanza of poetry written in this file:

`/class/datamine/data/hidden/poem.txt`

Hint: The `cat` command should be helpful for this purpose.

1b. Download the 2006 flights from the 2009 ASA Data Expo, using the method that was demonstrated in the project 7 examples:

```
wget http://stat-computing.org/dataexpo/2009/2006.csv.bz2
```

```
bzip2 -d 2006.csv.bz2
```

How many flights are found in the 2006 file?

Hint 1: The `wc` command should be helpful for this purpose.

Hint 2: Don't forget to check the head of the file–the first line of the file is the header, and you do not want to count that in your total.

Question 2: Learning about the grep command

2a. Using the flights from 2006 that were downloaded in question 1b, save all of the information about the flights that departed or arrived at `IND`, into a new file called `indyflights.csv`.

Hint 1: The `grep` command should be helpful for this purpose.

Hint 2: The right carrot is used for saving the output into a new file.

2b. Using the `5000_transactions.csv` file from 8451, save all of the information about the purchases from January 1, 2017 (but no other information), into a new file called `newyearsday.csv`.

Hint: You might want to (first) check the head of the file, so that you get the format of the dates correct.

2c. Using the data from the 2018 election campaign donations, save all of the information about the donors that were somehow affiliated with Purdue, into a new file called `purduedonations.txt`.

Hint: All of the contents of the file are in capital letters, so search for PURDUE rather than Purdue.

2d. How many such donations were made in the 2018 election campaign, from Purdue-related donors?

Question 3: Learning about the cut command

3a. Using the flights from 2006 that were downloaded in question 1b, save all of the information about the origins and destinations of the flights (but none of the other information from the other variables), into a new file called `originsdestinations.csv`.

Hint: The `cut` command should be helpful for this purpose.

Project Submission:

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