STAT 29000 Project 7

Topics: Bash Scripts

Motivation: We know quite a bit about how bash can be used to help with data wrangling, especially during the early parts of the data analysis cycle. Now we learn how to wrap bash into a script.

Context: We often have very refined processes in bash, which we might want to utilize over and over again. Using a bash script, we can use our bash skills systematically.

Scope: We write some bash scripts to be used (in a very elementary way) in the analysis of the flights data and the election data. We focus on scripts that promote reproducibility and systematic analysis. Wrapping our bash commands into a script makes our data analysis more transparent to our colleagues, with whom we might share our methods of analysis.

To write a script in bash, we usually open a plain text file, either using `Applications > Text Editor` or by typing: `gedit&` in the terminal. A typical bash script looks like this:

```
#!/bin/bash
```

```
.....put the code here.....
```

After we save the code in a bash script and we are ready to test it, we need to save the script. We usually use `.sh` for the file extension. We also need to set the permissions for the file after we save it (this only needs to be done once), like this:

```
chmod 755 myscript.sh
```

Use this template to submit your solutions:

```
/class/datamine/data/examples/stat29000project7tempate.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&`

Question 1: Analyzing Indianapolis flights

1a. Write a script called `indyflights.sh` that takes a file from this directory as its input:

```
/class/datamine/data/flights/
```

and returns the number of flights that have `IND` as the origin or destination.

Hint: To test your file, there are 17524 `IND` flights in 1987. If you saved your `indyflights.sh` file in your home directory, you might type:

```
cat 1987.csv | ~/indyflights.sh
```

and it should return 17524 as the solution.

or if you have indyflights.sh saved your current directory, you might type:

```
cat 1987.csv | ./indyflights.sh
```

(Remember that the "tilde" stands for your home directory and the "dot" stands for the current directory.)

1b. Use your bash script `indyflights.sh` to determine how many flights have origin or destination `IND` altogether (across all years in all of the flights files). Your solution should match your solution from Project 3, Question 3a/3b.

Question 2: Analyzing flights for any airport

2a. Copy `indyflights.sh` to a new file called `anyflights.sh` and, instead of searching for flights with origin or destination `IND`, allow the user to send the airport location as the 1st parameter.

Hint: To test your file, there are 135046 `ORD` flights in 1987. So if you type

```
cat 1987.csv | ~/anyflights.sh ORD
```

or perhaps this (if `anyflights.sh` is in your current directory)

```
cat 1987.csv | ./anyflights.sh ORD
```

it should return 135046 as the solution.

Hint: The first variable passed to the bash shell is `$1`

You can read more here:

The Linux Command Line by William E. Shotts, Jr.

https://sourceforge.net/projects/linuxcommand/files/TLCL/19.01/TLCL-19.01.pdf

p. 452 (middle of page), p. 453 (middle of page), and p. 455 (top)

Question 3: A script for summarizing election campaign contributions

3a. Write a bash script called electionscript.sh that takes a year and a person's name and returns the amount of money that was given by this person in that election campaign year. (It is OK if the script combines all of the contributions from people with that name.)

For instance, if you type:

```
./electionscript.sh "FARHAT, LILLIAN" 2020
```

it should return: 10538

or if you type:

```
./electionscript.sh "GILMAN, PRISCILLA" 2016
```

it should return: 344495

Hint 1: Try to get this working without using a bash script. After you have things working, then build your bash script, and replace:

```
/class/datamine/data/itcont2020.txt
```

by

```
/class/datamine/data/election/itcont${2}.txt
```

and replace:

```
grep "FARHAT, LILLIAN"
```

by

```
grep "${1}"
```

Hint 2: This Stack Overflow page might help!

https://stackoverflow.com/questions/8748831/when-do-we-need-curly-braces-around-shell-variables

Project Submission:

Submit your solutions for the project at this URL: https://classroom.github.com/a/Qagk44YL using the instructions found in the GitHub Classroom instructions folder on Blackboard.