Project 09 Answer Key and Grading Guide

https://datamine.purdue.edu/seminars/fall2019/stat19000project9.html

General guidelines

Generally we don't want to penalize incorrect answers too heavily. What's important is that the student makes an honest attempt at a solution and provides rationale for their methods. Remember, it's all about the learning.

- Each assignment is worth 10 points

Accepted file formats

To receive full credit, students must use the provided project template.

- If a solution's formatting deviates significantly from that of the template, deduct 0.5 points.

Adding comments to student assignments

Create a text file called `grader_notes.txt` in each student's project folder. Put any comments or corrections in there.

Project-specific guidelines

For any given problem...

- deduct 0.5 points for missing code (if code is required to solve this problem)
- deduct 0.5 points for missing output (if output is required to solve this problem)
- deduct 0.5 points for missing comments
- deduct 0.5 points for incorrect solutions

... for a minimum score of 0 on the individual problem.

Question 1a (2 pts)

Use awk to solve Project 3, Question 2b, namely: How many passengers (altogether) rode in yellow taxi cab rides in New York City in June 2019?

```
# Use cd to navigate to the yellow taxi data directory
cd /class/datamine/data/taxi/yellow

# Use cat to list the contents of the desired file
# Use awk with -F, to specify a comma delimiter
# For each row in the file, add the 4th field (passenger_count) to a new
# variable called 'total_passengers'.
# Once we are done passing over each row, print the value of 'total_passengers'
cat yellow_tripdata_2019-06.csv |
awk -F, 'BEGIN   {                        }
                 { total_passengers += $4 }
         END     { print total_passengers }'
```

```
>>>
10878820
```

Question 1b (2 pts)

Use awk to solve Project 4, Question 2b, namely: What is the mean total number of passengers in a New York City yellow taxi cab ride in June 2019?

```
# Use cat to list the contents of the desired file
# Do the same thing as in (1a), with the addition of a new variable called
# 'total_rides', which will represent the total number of rides in this data.
# Since each row in this data represents a ride, add 1 to total_rides for each
# row in the data.
# At the END, calculate and print the mean number of passengers per ride by
# dividing 'total_passengers' by 'total_rides'
cat yellow_tripdata_2019-06.csv |
awk -F, 'BEGIN  {                                          }
                { total_passengers += $4; total_rides += 1 }
         END    { print total_passengers/total_rides       }'
```

```
>>>
1.56732
```

Question 2a (2 pts)

Use awk to analyze the 8451 transactions data: find the total amount (in dollars) spent on grocery purchases (altogether) on 23 December 2017.

```
# Use cd to navigate to the transactions data directory
cd /class/datamine/data/8451/The_Complete_Journey_2_Master
# Use cat to list the contents of the transaction data
# Use awk with -F, to specify a comma delimiter, adding together the
# transaction amounts for every transaction made on 23-DEC-17 in a variable
# called 'total_spent'.
# When done, print 'total_spent'.
cat 5000_transactions.csv |
awk -F, 'BEGIN  {                                          }
                { if($3 == "23-DEC-17") {total_spent += $5} }
          END   { print total_spent                        }'
```

```
>>>
108408
```

Question 2b (2 pts)

Use awk to find the average amount (in dollars) spent in a transaction, on 23 December 2017.

```
# Repeat the process from (2a), but also keep track of the number of
# transactions that occurred on 23-DEC-17 using a variable called 'n'.
# When done, print the average dollar amount spent per transaction.
awk -F, 'BEGIN  {                                            }
                { if($3 == "23-DEC-17"){total_spent += $5; n += 1} }
          END   { print total_spent/n                        }'
```

```
>>>
4.07045
```

Question 3a (2 pts)

Use awk to find the average amount (in dollars) in a donation in the 2018 election campaign.

```
# Use cd to navigate to the election data directory
cd /class/datamine/data/election
# Use awk with -F\| to specify a '|' delimiter and add up the TRANSACTION_AMTs
# of each donation in a variable called 'total_donation_amt'.
# When done, print 'total_donation_amt'.
cat itcont2018.txt |
awk -F\| 'BEGIN {                                            }
                { total_donation_amt += $15; total_donations+=1 }
           END  { print total_donation_amt/total_donations     }'
```

```
>>>
257.348
```