Project 07 Answer Key and Grading Guide

https://datamine.purdue.edu/seminars/fall2019/stat19000project7.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

Accpeted 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 (1 pt)

Display the stanza of poetry written in this file: `/class/datamine/data/hidden/poem.txt`

```
# Use cat to list the contents of poem.txt
cat /class/datamine/data/hidden/poem.txt
```

```
>>>
Do not go gentle into that good night,
Old age should burn and rave at close of day;
Rage, rage against the dying of the light.
```

Question 1b (1 pt)

Download the 2006 flights from the 2009 ASA Data Expo, using the method that was demonstrated in the project 7 examples. How many flights are found in the 2006 file?

```
# Use wget to download 2006.csv.bz2
wget http://stat-computing.org/dataexpo/2009/2006.csv.bz2
# Use bzip2 with -d to expand the downloaded file
bzip2 -d 2006.csv.bz2
# Use wc with -l to get the line count of 2006.csv
wc -l 2006.csv
```

```
>>>
7141923 2006.csv
```

Taking into account the file header, there are 7141922 recorded flights.

Question 2a (2 pts)

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`.

```
# Use grep with to search for IND in indyflights.csv
# Optionally pipe the output into head to see only the first 10 lines
grep IND 2006.csv > indyflights.csv | head
```

```
>>>
2006,1,11,3,1949,1950,2115,2134,US,432,N781AU,86,104,69,-19,-1,CLT,IND,428,6,11,0,,0,0,0,0,0,0
2006,1,11,3,1120,1110,1305,1253,US,509,N441US,105,103,80,12,10,CLT,IND,428,10,15,0,,0,0,0,0,0,0
2006,1,11,3,1624,1625,1809,1814,US,101,N506AU,105,109,84,-5,-1,DCA,IND,499,9,12,0,,0,0,0,0,0,0
2006,1,11,3,647,655,835,841,US,141,N426US,108,106,83,-6,-8,DCA,IND,499,8,17,0,,0,0,0,0,0,0
2006,1,11,3,841,845,1019,1015,US,51,N375US,98,90,80,4,-4,IND,CLT,428,5,13,0,,0,0,0,0,0,0
2006,1,11,3,701,710,828,840,US,1092,N530AU,87,90,70,-12,-9,IND,CLT,428,5,12,0,,0,0,0,0,0,0
2006,1,11,3,1332,1335,1503,1503,US,1445,N441US,91,88,73,0,-3,IND,CLT,428,6,12,0,,0,0,0,0,0,0
2006,1,11,3,1837,1845,1958,2013,US,1110,N506AU,81,88,70,-15,-8,IND,DCA,499,3,8,0,,0,0,0,0,0,0
2006,1,11,3,926,930,1055,1100,US,1275,N426US,89,90,76,-5,-4,IND,DCA,499,4,9,0,,0,0,0,0,0,0
2006,1,11,3,1108,1105,1303,1242,US,1675,N812MD,115,97,79,21,3,IND,PHL,587,4,32,0,,0,0,0,21,0,0
```

Question 2b (2pts)

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`.

```
# Use grep to search for lines containing '01-JAN-17' in the transactions data
# Use > to pipe the output into a file called newyearsday.csv
grep 01-JAN-17 /class/datamine/data/8451/The_Complete_Journey_2_Master/5000_tra
nsactions.csv > newyearsday.csv
# Use head to peek at the first 10 lines of your output
head newyearsday.csv
```

```
>>>
518325                ,2820           ,01-JAN-17,5180178                ,      2.68,
518338                ,3524           ,01-JAN-17,0088558                ,      2.69,
518357                ,2195           ,01-JAN-17,5571584                ,      3.99,
518372                ,1751           ,01-JAN-17,0714604                ,         1,
519598                ,4945           ,01-JAN-17,5205991                ,        .5,
519633                ,1315           ,01-JAN-17,0792859                ,      5.99,
519645                ,3105           ,01-JAN-17,0699309                ,      1.49,
519743                ,1918           ,01-JAN-17,0965989                ,      3.29,
520686                ,2710           ,01-JAN-17,0085348                ,      2.19,
520754                ,4182           ,01-JAN-17,6115990                ,      2.99,
```

Question 2c (1 pt)

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`.

```
# Use grep to search for PURDUE in the 2018 election data
# Use > to pipe the output to a file called purduedonations.txt
grep PURDUE /class/datamine/data/election/itcont2018.txt > purduedonations.txt
# Use head to peek at the first 10 lines of this new file
head purduedonations.txt
```

```
>>>
C00540443|N|YE|P|201801299090882789|15E|IND|MYKYTIUK, LAWRENCE|WEST LAFAYETTE|IN|479064127|PURDUE UN
C00327023|N|YE|P|201801319091044128|15|IND|PURDUE, PAULA|CHICAGO|IL|60613|RETIRED|RETIRED|10162017|1
C00365536|N|YE|P|201801319091211714|15|IND|BODNER, GEORGE|WEST LAFAYETTE|IN|479072084|PURDUE UNIVERS
C00365536|N|YE|P|201801319091211715|15|IND|BODNER, GEORGE|WEST LAFAYETTE|IN|479072084|PURDUE UNIVERS
C00654442|N|YE|P|201801319091182652|15|IND|GRIFFITH, JULIE K|CARMEL|IN|460328536|PURDUE UNIVERSITY|A
C00030676|N|YE|P|201801239090516743|15|IND|PURDUE, WILLIAM|PITTSBURGH|PA|15219|UNITED STATES STEEL C
C00166504|N|YE|P|201801309090926391|15|IND|FLANNERY, MICHAEL|VALPARAISO|IN|463831911|PURDUE NORTHWES
C00166504|N|YE|P|201801309090926474|15|IND|SKOZEN, CONSTANCE M.|DYER|IN|46311|PURDUE UNIVERSITY NORT
C00193433|N|YE|P|201801239090524787|15|IND|BONEM, EMILY|WEST LAFAYETTE|IN|47906|PURDUE UNIVERSITY|IN
C00193433|N|YE|P|201801239090527069|15|IND|SCHWEICKERT, RICHARD MR.|LAFAYETTE|IN|47905|PURDUE UNIVER
```

NOTE: The `grep` command will indiscriminately search for `'PURDUE'`. This means that your output will also contain donations made by people named `PURDUE`. You can actually see one such person on the second line in the output above: `PURDUE, PAULA` from Chicago. For the purposes of this problem, it's OK to include these entries!

Question 2d (1 pt)

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

```
Use wc with -l to get the line count of the purudue donors file
wc -l purduedonations.txt
```

```
>>>
2237 purduedonations.txt
```

Question 3a (2 pts)

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`.

```
# Use cut with -d, to specify a comma delimiter and -f17,18 to get fields 17 and 18
# Use > to pipe the output to a file called originsdestinations.csv
cut -d, -f17,18 2006.csv > originsdestinations.csv
# Use head to peek at the first 10 lines of the new file
head originsdestinations.csv
```

```
>>>
Origin,Dest
ATL,PHX
ATL,PHX
ATL,PHX
AUS,PHX
AUS,PHX
BDL,CLT
BDL,CLT
BDL,CLT
BDL,CLT
```