Project 10 Answer Key and Grading Guide

https://datamine.purdue.edu/seminars/fall2019/stat19000project10.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 the UNIX cut or awk tool to extract (only) the dates and transaction amounts from the election data, across all years. Save this data into a file in your home directory.

Solution using cut

```
# Use cat to list the contents of all the election data files.
# Use cut with -d\| to specify a | delimiter and -f14,15 to extract the 14th and
# 15th fields.
# Use > to write the resulting output to a file called project10_1a.txt.
cat /class/datamine/data/election/*.txt | cut -d\| -f14,15 > project10_1a.txt
```

Solution using AWK

```
# Use cat to list the contents of all the election data files.
# Use awk with -F\| to specify a | delimiter.
# BEGIN{} will be empty.
# {} we will print the 14th and 15th fields for each line in the data, with a
# "|" thrown in as a delimiter.
# END will be empty.
# Use > to write the resulting output to a file called project10_1a.txt.
cat /class/datamine/data/election/*.txt |
awk -F\| '{ print $14"|"$15 }' > project10_1a.txt
```

Question 1b (1 pt)

> Remove the lines from the file that came from the headers of each election file, i.e., remove the 21 lines of the form: TRANSACTION_DT|TRANSACTION_AMT Save the result into a new file.

```
# Use cat to list the contents of the previously-created project10_1a.txt.
# Use grep with -v to print out only lines that do not contain the string
# "TRANSACTION_DT|TRANSACTION_AMT".
# Use > to write the resulting output to a file called project10_1b.txt
cat project10_1a.txt | grep -v "TRANSACTION_DT|TRANSACTION_AMT" > project10_1b.txt
```

Question 2a (1 pt)

> Import into R the resulting file that you prepared from question 1.

```
# Use read.delim() to read in the file created in the previous problem.
transactions = read.delim("./project10_1b.txt", sep="|", header=FALSE, colClasses=c("character", "in
```

Question 2b (2 pts)

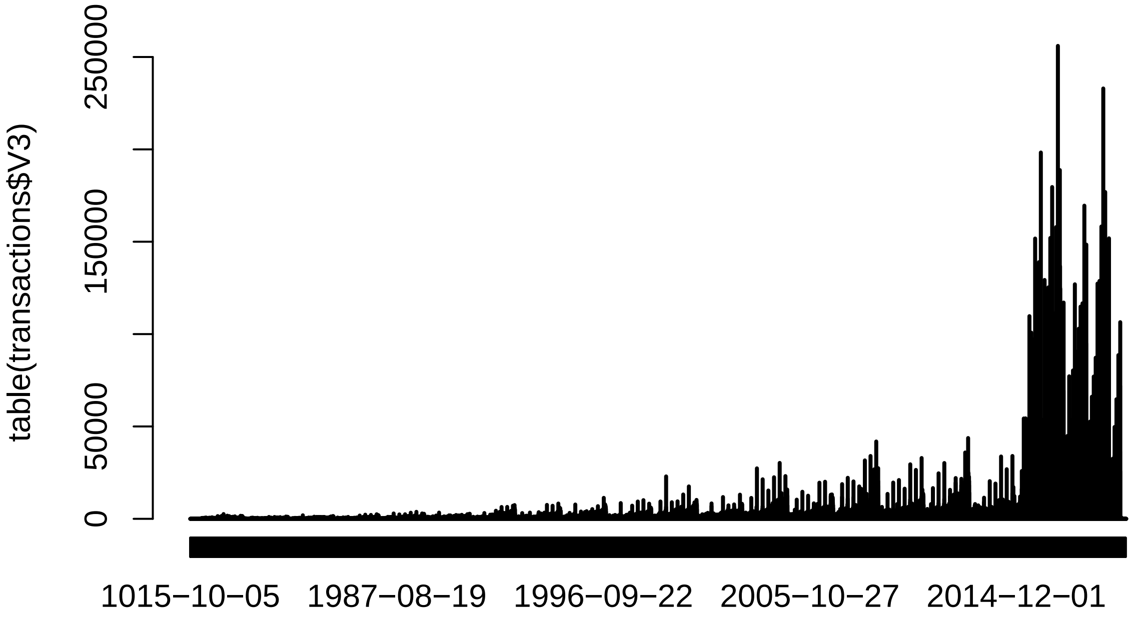
> Make a new third column of the data.frame that has the same data from column 1, but is stored in Date format.

```
# Use as.Date with date format "%m%d%Y" to convert the elements in column 1 to
# date objects.
transactions$V3 = as.Date(transactions$V1, format="%m%d%Y")
```

Question 3a (2 pts)

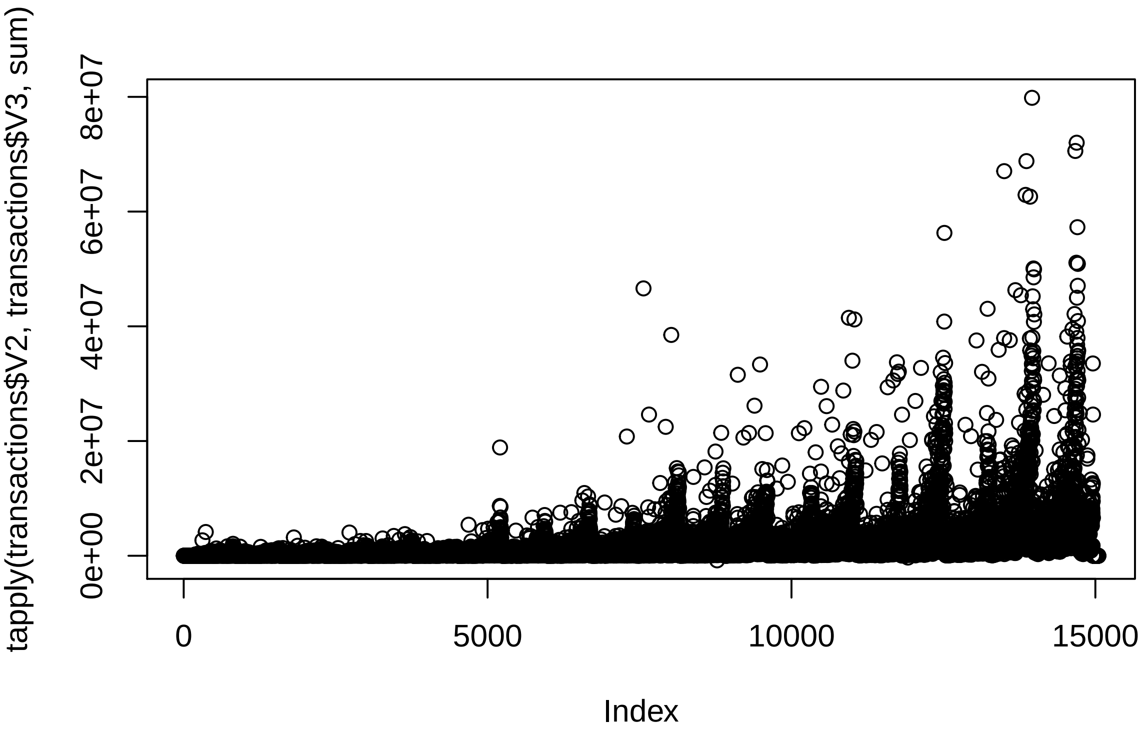
> Plot a table of the number of donations given per day. You might want to cut off some of the first several and last several dates, which are probably erroneous dates.

```
# Use table() to get a the frequencies of donations for each date.
# Use plot() to plot date v. number of donations
plot(table(transactions$V3))
```



Question 3b (2 pts)

```
# Use tapply() to get the total transaction amount for each date.
# Use plot() to plot date v. total transaction amount
plot(tapply(transactions$V2, transactions$V3, sum))
```


tapply(transactions$V2, transactions$V3, sum)
0e+00
2e+07
4e+07
6e+07
8e+07
0
5000
10000
15000
Index