Assignment1

Code

Main Topics: Data Exploration & Data Visualization

Tasks:

- Data Exploration & Visualization:

1. Download the file 'parkingLA2017.csv' from our class Blackboard site.
2. Read this file into your R environment (if it takes a while for the file to load, don't worry – this is normal. Be patient). Show the step that you used to accomplish this.

Hide

```
parkingLA2017.df <- read.csv("parkingLA2017.csv", header = TRUE) # load data
```

a. What are the dimensions of this dataframe? Show the code that you used to determine this.

Hide

```
dim(parkingLA2017.df) # find the dimensions of the dataframe
```

```
[1] 1048575      21
```

3. Filter the dataframe. Create a new object that only contains data for your assigned "Make" of car. Show the code that you used to accomplish this. For the next set of questions, use this new dataframe.

a. What are the dimensions of this new dataframe that only contains the rows for your assigned "Make" of car?

Hide

```
library(dplyr)
```

```
Registered S3 method overwritten by 'dplyr':
  method           from
  print.rowwise_df

Attaching package: ‘dplyr’

The following objects are masked from ‘package:arules’:

    intersect, recode, setdiff, setequal, union

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

Hide

```
CHRY <- filter(parkingLA2017.df, Make == "CHRY") # create a new DF with 'CHRY' as a filt
er
dim(CHRY) # find the new dimensions
```

```
[1] 11964    21
```

4. Dealing with missing values:

a. Are there any missing values in your dataset? How do you know this? Show the R code that you used to answer this question, along with the results that appearedin your console.

Hide

```
anyNA(CHRY) # determine if any missing values exist with dataframe CHRY
```

```
[1] TRUE
```

b. Does the variable RP.State.Plate contain any missing values? How do you know this? Again, show your steps and the results.

Hide

```
anyNA(CHRY$RP.State.Plate) # determine if variabble RP.State.Plate has missing values
```

```
[1] FALSE
```

c. Find and display the standard deviation for the variable Fine.amount. Does the variable Fine.amount contain any missing values? How do you know this? Again, show your steps and the results. To deal with this issue, perform an imputation by replacing the NAs with a reasonable alternative. Now, find and display the standard deviation for this variable again. What happened? Why do you think this change occurred?

Hide

```
sd(CHRY$Fine.amount) # Find the standard deviation for 'Fine.amount' variable
```

```
[1] NA
```

Hide

```
anyNA(CHRY$Fine.amount) # Alternative determine if NA's exist for var 'Fine.amount'
```

```
[1] TRUE
```

Hide

```
median(CHRY$Fine.amount, na.rm = TRUE) # Determine if the median is reasonble
```

```
[1] 68
```

Hide

```
mean(CHRY$Fine.amount, na.rm = TRUE) # Determine if the mean is reasonable
```

```
[1] 72.69187
```

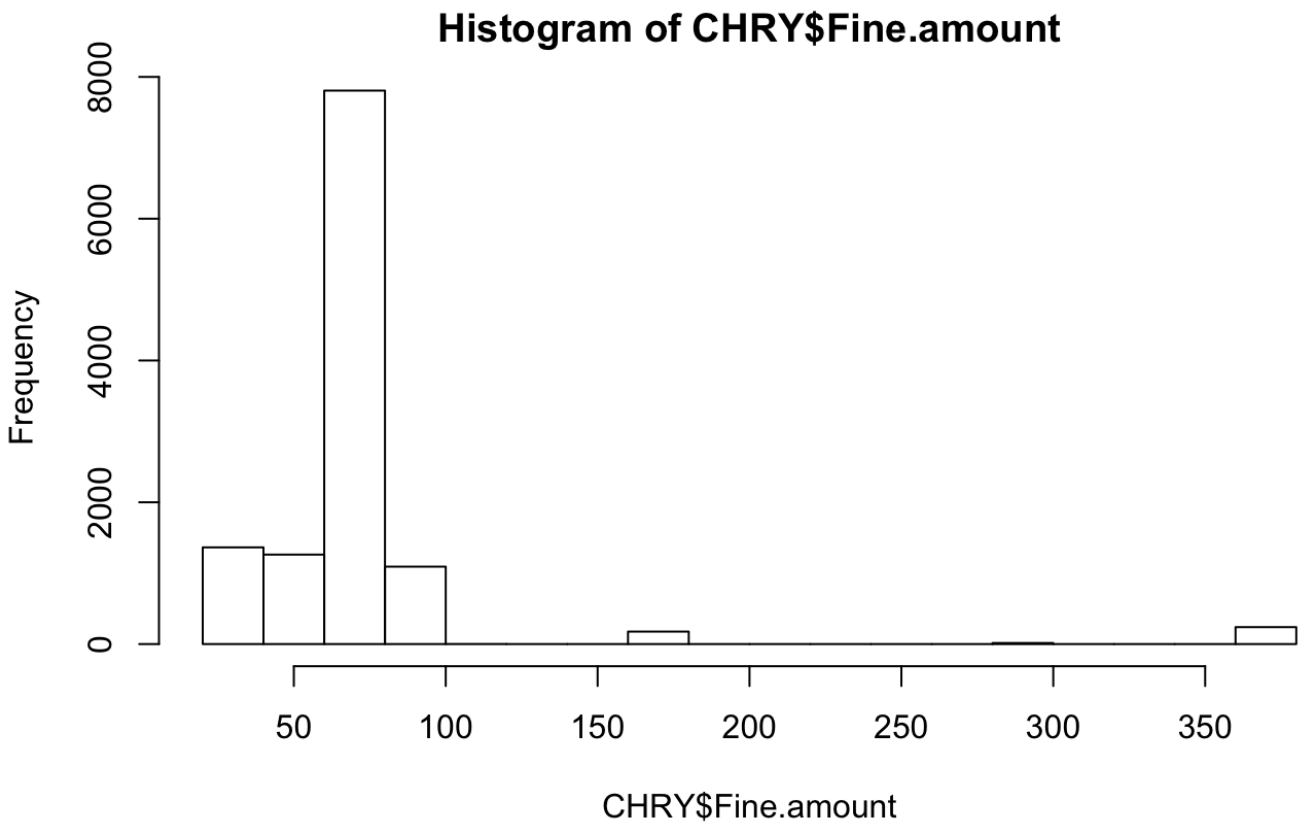
Hide

```
range(CHRY$Fine.amount, na.rm = TRUE) # Observe the range for the variable
```

```
[1]  25 363
```

Hide

```
hist(CHRY$Fine.amount) # Use the histogram to observe any outliers
```



Hide

```
CHRY$Fine.amount[is.na(CHRY$Fine.amount)] <- median(CHRY$Fine.amount, na.rm = TRUE) # us
e median
anyNA(CHRY$Fine.amount) # determine if NAs were replaced
```

```
[1] FALSE
```

Hide

```
sd(CHRY$Fine.amount) # re-run the standard deviation
```

```
[1] 47.03247
```

d. Replace any blank cells in the Location column with NA. In a sentence or two, what does this accomplish?

Hide

```
CHRY$Location[CHRY$Location==""] <- "NA" # replace blanks with NA's
```

```
invalid factor level, NA generated
```

Hide

```
anyNA(CHRY$Location) # check for any NA's
```

```
[1] TRUE
```

e. Now, remove all of the records that contain “NA” for AGENCY.SHORT.NAME from your dataset.

Hide

```
anyNA(CHRY$AGENCY.SHORT.NAME) # check to see if NA's exist within that variable
```

```
[1] TRUE
```

Hide

```
library(tidyr)
```

```


Attaching package: ‘tidyr’

The following objects are masked from ‘package:Matrix’:

    expand, pack, unpack
```

Hide

```
CHRY <- drop_na(CHRY, AGENCY.SHORT.NAME) # drop all NA's from
table(CHRY$AGENCY.SHORT.NAME) # after checking, NA's do exist
```

```


AIRPT BACK     AMTRAK BLDG & SAF    CENTRAL DOT - HLYW DOT - VALY
         0           0          17           0         514       4963
DOT - WEST DOT - WLSH     G.S.D.  HOLLYWOOD         HPV  LAPD BACK
      5221           0           1         131           0           0
   LAX CUR PUB. UTIL.    RANGERS   SOUTHERN STREET USE     VALLEY
       152           0           0           0           0          17
VN AIRPORT    WESTERN
        16         920
```

Hide

```
CHRY$AGENCY.SHORT.NAME <- droplevels(CHRY$AGENCY.SHORT.NAME) # alternate to dropping N
A's
View(CHRY)
```

5. Dealing with the Date data type A. Right now, if you call the str() function on your dataset, you’ll see that R does not recognize the Issue.Date variable as a date. Fix this by explicitly telling R to treat this variable as a date. Show the code that you used to accomplish this.

Hide

```
CHRY$Issue.Date <- as.Date(CHRY$Issue.Date, format="%Y-%m-%d" ) # change the date based
 from View
str(CHRY) # recall the structure
```

```
'data.frame':   11952 obs. of  21 variables:
 $ Agency               : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Ticket.number        : Factor w/ 1048575 levels "1001197094","1001197105",..: 32198 5
3343 54395 55516 51736 53427 69070 53348 81380 16594 ...
 $ Issue.Date           : Date, format: "2017-10-11" "2017-03-13" ...
 $ Issue.time           : int  1540 1540 1630 1455 920 1335 217 1450 1538 900 ...
 $ Meter.Id             : Factor w/ 18141 levels "","."."#1","#2",..: 1 1 1 1 1 1 1 1 21
8 1 ...
 $ Marked.Time          : int  NA NA NA NA NA NA NA NA NA NA ...
 $ RP.State.Plate       : Factor w/ 71 levels "AB","AK","AL",..: 8 8 8 8 8 8 8 8 8 ...
 $ Plate.Expiry.Date    : int  201707 201606 NA 201708 201706 201709 201711 NA 201802 NA
...
 $ VIN                  : logi  NA NA NA NA NA NA ...
 $ Make                 : Factor w/ 1076 levels "","ABAR","ABRI",..: 148 148 148 148 148
148 148 148 148 148 ...
 $ Body.Style           : Factor w/ 96 levels "","2D","2H","4D",..: 51 51 51 51 51 51 51
51 51 51 ...
 $ Color                : Factor w/ 68 levels "","AM","AP","AQ",..: 53 49 25 64 10 64 49
7 10 25 ...
 $ Location             : Factor w/ 307500 levels "",",10989 ROCHESTER AVE",..: 185483 3
04477 201295 179143 130195 33756 95692 304475 301187 304296 ...
 $ Route                : Factor w/ 3942 levels "","016V5","01A4",..: 1730 1779 1732 177
3 1 1079 3028 1779 2171 2103 ...
 $ Violation.code       : Factor w/ 220 levels "0","1","10","11",..: 82 44 78 75 163 146
6 44 201 219 ...
 $ Violation.Description: Factor w/ 384 levels "","1504A","1564052",..: 264 271 259 285
295 358 30 271 280 120 ...
 $ Fine.amount          : num  25 363 25 50 93 93 68 363 63 25 ...
 $ Latitude             : num  99999 99999 99999 6473311 6488212 ...
 $ Longitude            : num  99999 99999 99999 1825895 1841816 ...
 $ AGENCY.NAME          : Factor w/ 20 levels "51 - DOT - WESTERN",..: 20 20 20 20 20 20
20 20 20 20 ...
 $ AGENCY.SHORT.NAME    : Factor w/ 10 levels "BLDG & SAF","DOT - HLYW",..: 10 10 10 10
10 10 10 10 10 10 ...
```

B. Are the rows currently displayed in chronological (i.e. date) order? How do you know this? Use the arrange() function from dplyr to put the dates in order. Show the code that you used to accomplish this.

Hide

```
View(CHRY$Issue.Date) # filter by issue date or use View overall
CHRY <- arrange(CHRY, Issue.Date) # arrange the date in chronological order
View(CHRY) # recall View and the dates are in order
```

C. What month were you born in? Using the subset() function, make a new object that only contains dates for your particular birth month. Show the code that you used to accomplish this. We will not use this object again in any of the following steps.

Hide

```
november <- subset(CHRY, Issue.Date >= "2017-11-01" & Issue.Date <= "2017-11-30") # use
 the first/last of Nov
View(november)
```

6. We won't need to use the variable 'ticket number' in our analysis. Remove this column from your dataframe. Show the code that you used to accomplish this.

Hide

```
CHRY2 <- CHRY[,-c(2)] # use the = symbol rather than <- if you were to completely remove
on CHRY
View(CHRY2)
```

7. Using the summary() function, find out even more about the distribution of fine amounts. Show a screenshot that displays the Minimum, 1st Quartile, Median, 3rd Quartile, Maximum, and Mean values for parking fine amounts.

Hide

```
summary(CHRY2$Fine.amount)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  25.00   63.00   68.00   72.69   73.00  363.00
```

8. Identify the five most common types of violation descriptions in your dataset. Show the code that you used to accomplish this, and a screenshot that shows the names of the five most common violation descriptions.

Hide

```
summary(CHRY2$Violation.Description) # shows all summary for violations desc for top fiv
e
```

NO PARK/STREET CLEAN	METER EXP.
3717	1578
DISPLAY OF TABS	RED ZONE
910	707
NO PARKING	PREFERENTIAL PARKING
613	609
NO EVIDENCE OF REG	PARKED OVER TIME LIMIT
411	393
DISPLAY OF PLATES	EXCEED 72HRS-ST
319	238
WHITE ZONE	BLOCKING DRIVEWAY
181	163
NO STOPPING/ANTI-GRIDLOCK ZONE	STANDNG IN ALLEY
160	141
NO STOP/STANDING	18 IN. CURB/2 WAY
133	110
PARKED ON SIDEWALK	NO STOP/STAND
107	90
DISABLED PARKING/NO DP ID	FIRE HYDRANT
87	87
EXPIRED TAGS	HANDICAP/NO DP ID
78	72
STOP/STAND PROHIBIT	YELLOW ZONE
70	59
OUTSIDE LINES/METER	WHITE CURB
55	55
PK IN PROH AREA	PRIVATE PROPERTY
53	44
OFF STR/OVERTIME/MTR	DOUBLE PARKING
43	35
DSPLYPLATE A	NO STOPPING/STANDING
33	32
2251157A	22514
27	25
PREF PARKING	NO STOP/STAND PM
25	24
PARKED IN PARKWAY	OVNIGHT PRK W/OUT PE
23	22
STNDNG IN ALLEY	22500H
18	17
NO PARKING BETWEEN POSTED HOURS	PK OUTSD PK STL
17	17
22500F	EXCEED 72 HOURS
15	15
METER EXPIRED	PARK IN GRID LOCK ZN
14	14
2251157B	5204
13	13
DISABLED PARKING/CROSS HATCH	WITHIN INTERSECTION
13	13
PARKED IN BUS ZONE	22500E
12	11
4000A	RED CURB

```
                          10                          10
                       22502A           BLK BIKE PATH OR LANE
                           9                           9
                PARKING AREA             TIME LIMIT/CITY LOT
                           9                           9
          OFF STR MTR/OUT LINE             PARKED IN CROSSWALK
                           8                           8
                       22500B               18 IN. CURB/1 WAY
                           7                           6
          DP-BLKNG ACCESS RAMP                  WRG SD/NOT PRL
                           6                           6
           18 IN/CURB/COMM VEH                        5200
                           5                           5
              CITY PARK/PROHIB             DP- RO NOT PRESENT
                           5                           5
               EXCEED TIME LMT                PARK-PSTD AREAS
                           5                           5
           PARKED IN FIRE LANE              PARKING/FRONT YARD
                           5                           5
          RESTRICTED TAXI ZONE             3 FT. SIDEWALK RAMP
                           5                           4
               NO PK BET 1-3AM                NO STOP/STAND AM
                           4                           4
                PUBLIC GROUNDS                      22500I
                           4                           3
                       22502E             CLEANING VEH/STREET
                           3                           3
       COMM VEH OVER TIME LIMIT                LOAD/UNLOAD ONLY
                           3                           3
            R/PRIV PARKING AREA                       22522
                           3                           2
                       80581      DISABLED PARKING/BOUNDARIES
                           2                           2
DISABLED PARKING/OBSTRUCT ACCESS                   DP-REFUSE ID
                           2                           2
            HANDICAP/CROSS HATCH   PARKING OUTSIDE PARKING STALLS
                           2                           2
                PK OTSD PSTD AR                SAFETY ZONE/CURB
                           2                           2
             STORING VEH/ON STR
                           2                           1
                       22500A                       22502
                           1                           1
                       225078                     22523AB
                           1                           1
                       22651C                     (Other)
                           1                          19
```

9. Create a new dataframe that only contains data for the five most common violations. Show the code that you used to accomplish this. You will use this new dataframe for all the following steps in this assignment.

Hide

```
top5vio <- as.data.frame(sort(table(CHRY2$Violation.Description), decreasing = TRUE)[1:5
]) # create a new df
test <- filter(CHRY2, Violation.Description %in% top5vio$Var1) # pull the top5
dim(test) # check for varaibles
```

```
[1] 7525   20
```

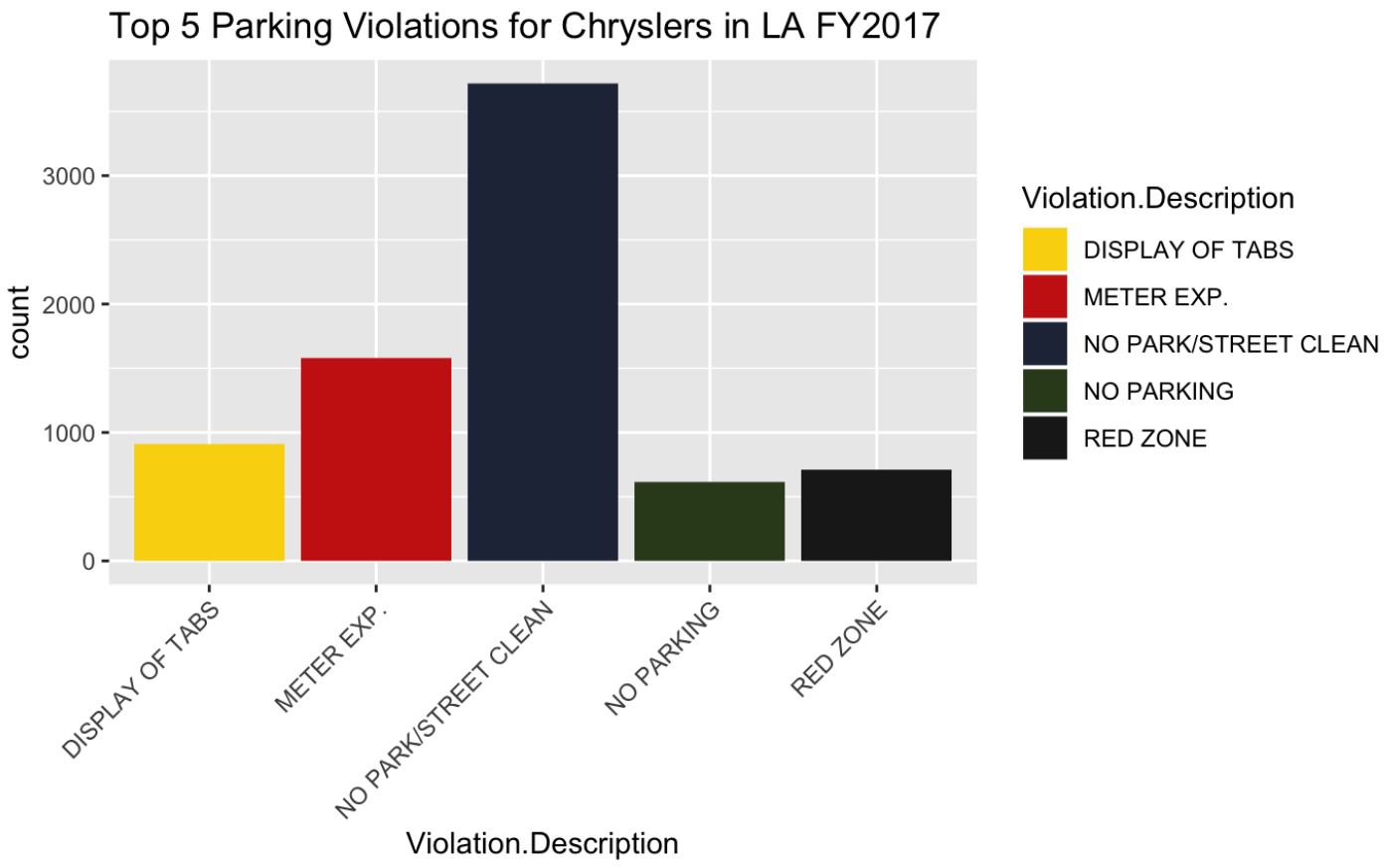
Hide

```
View(top5vio) # view your new dataframe
```

10. Using ggplot, create a barplot that displays the number of occurrences for the five most common violations. Be sure to label your axes, to give the graph a title, and to color each of your bars. In a sentence or two, describe what your barplot is showing you.

Hide

```
library(ggplot2)
library(wesanderson)
top5bar <- ggplot(test, aes(Violation.Description, fill=Violation.Description)) + geom_b
ar() + scale_fill_manual(values = wes_palette(n=5, "BottleRocket2"))
top5bar <- top5bar + labs(title = "Top 5 Parking Violations for Chryslers in LA FY2017")
# add title
top5bar <- top5bar + theme(axis.text.x = element_text(angle=45, hjust=1)) # adjust the l
abels on horizontal line
top5bar
```



11. How did the average size of a fine vary from agency to agency? (use the AGENCY.SHORT.NAME variable to make this grouping). Find the average fine Size for each agency (show a screenshot of your code plus your results) and then display your results visually with a barplot built using ggplot. Give your barplot a title, and clearly label your x and y axes. Be sure to color your bars. In a sentence or two, describe what your barplot is showing you.

Hide

```
Chrysler <- group_by(test, AGENCY.SHORT.NAME)
Chrysler2 <- summarise(Chrysler, AvgFine = mean(Fine.amount))
Chrysler2
```

AGENCY.SHORT.NAME <fctr>	**AvgFine** <dbl>
BLDG & SAF	85.00000
DOT - HLYW	64.96013
DOT - VALY	67.11465
DOT - WEST	66.28250
HOLLYWOOD	91.82353
LAX CUR	73.00000
VALLEY	93.00000
WESTERN	83.85271

8 rows

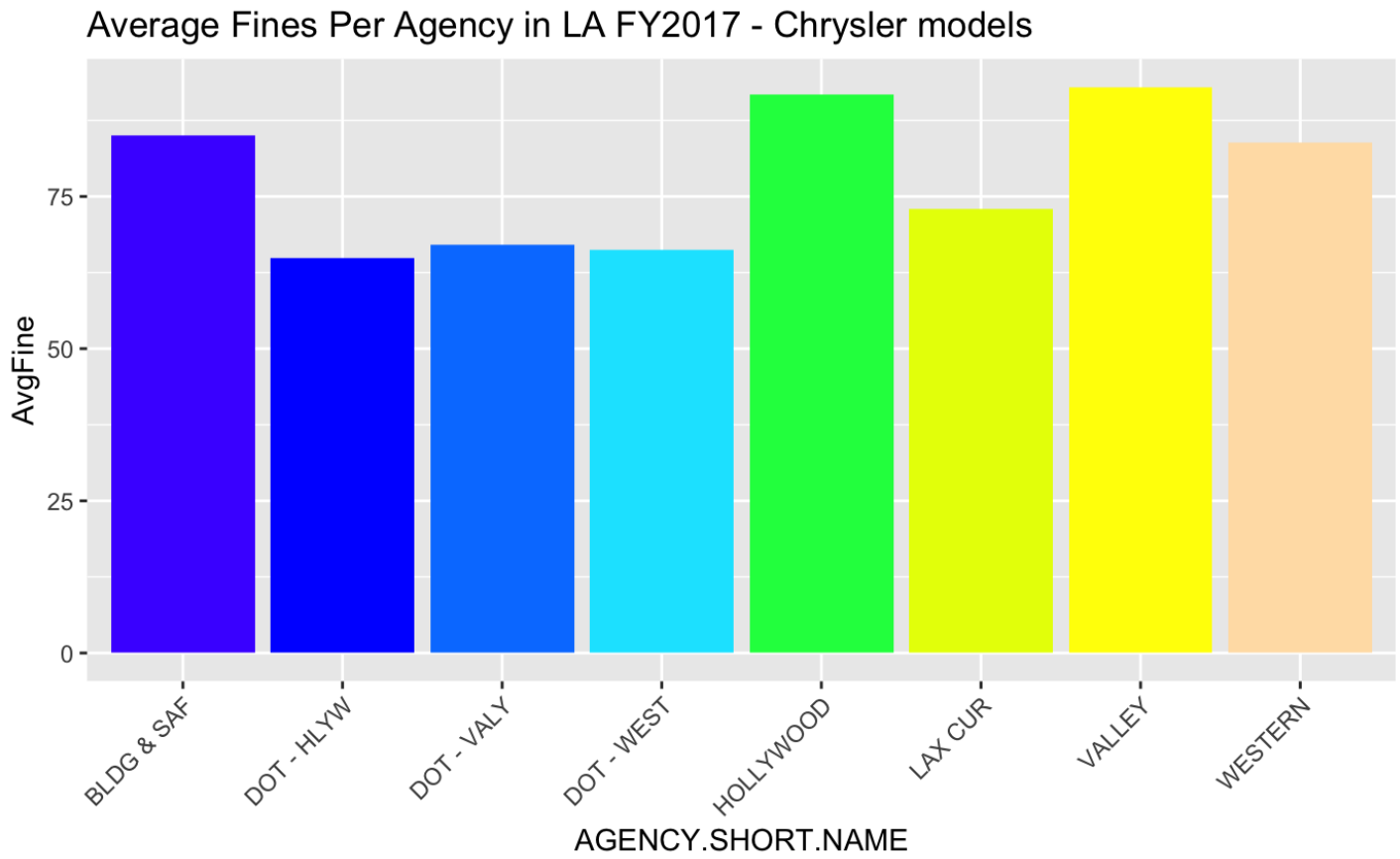
Hide

```
table(test$AGENCY.SHORT.NAME)
```

```
BLDG & SAF DOT - HLYW DOT - VALY DOT - WEST     G.S.D.  HOLLYWOOD
         5         301        3332        3738           0          17
   LAX CUR     VALLEY VN AIRPORT    WESTERN
         1           2           0         129
```

Hide

```
AvgBar <- ggplot(Chrysler2, aes(x=AGENCY.SHORT.NAME, y=AvgFine)) + geom_bar(stat="identi
ty", fill=topo.colors(n=8))
AvgBar <- AvgBar + labs(title = "Average Fines Per Agency in LA FY2017 - Chrysler model
s")
AvgBar <- AvgBar + theme(axis.text.x = element_text(angle=45, hjust=1))
AvgBar
```



12. Using ggplot, create a violin plot that shows the agencies on the x-axis, and the fine amounts on the y-axis. Give your violin plot a title, and clearly label your x and y axes. In a sentence or two, describe what your violin plot is showing you.

Hide

```
install.packages("Hmisc")
```

```
Installing package into '/Users/josemartinez/Library/R/3.6/library'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/Hmisc_4.4-0.tgz'
Content type 'application/x-gzip' length 3146788 bytes (3.0 MB)
==================================================
downloaded 3.0 MB
```

```
The downloaded binary packages are in
    /var/folders/6v/wsr694r57n9dfsdxftfdysbh0000gn/T//RtmpuQhIg0/downloaded_packages
```

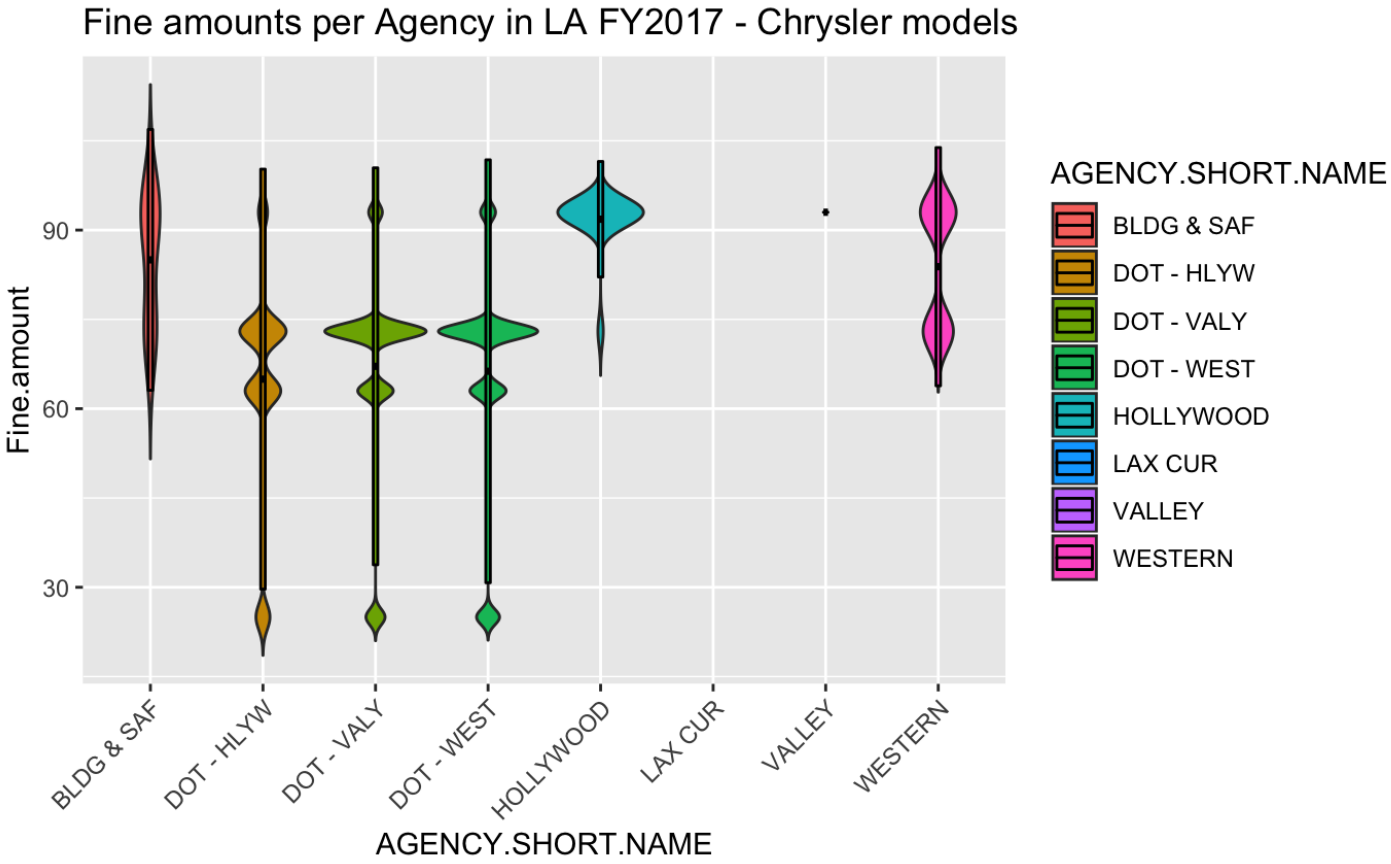
Hide

```
FineViolin <- ggplot(Chrysler, aes(x=AGENCY.SHORT.NAME, y=Fine.amount, fill=AGENCY.SHOR
T.NAME)) + geom_violin(trim=FALSE)
FineViolin <- FineViolin + stat_summary(fun.data="mean_sdl", mult=1, geom="crossbar", wi
dth=0.04 )
```

```
Ignoring unknown parameters: mult
```

Hide

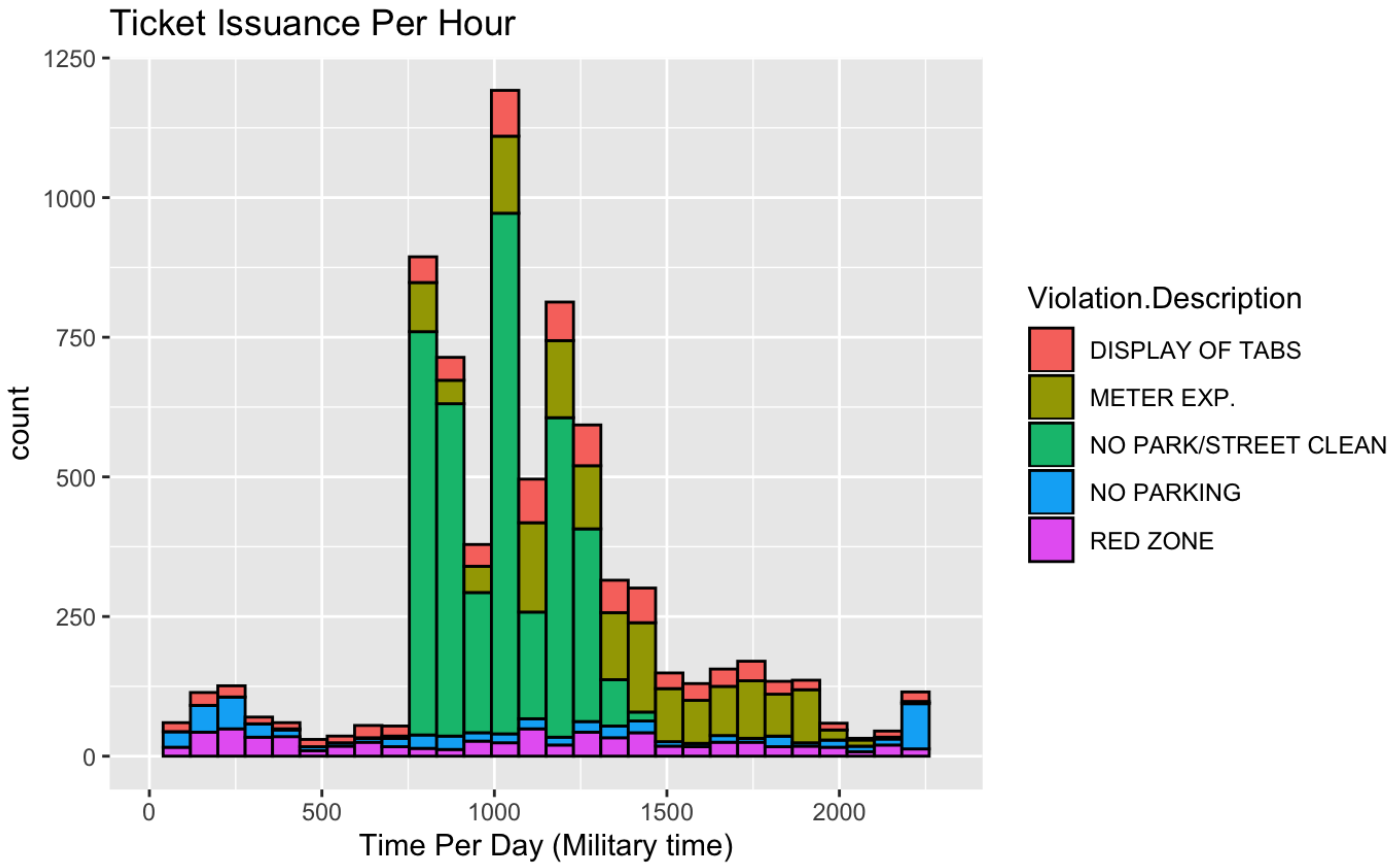
```
FineViolin <- FineViolin + labs(title = "Fine amounts per Agency in LA FY2017 - Chrysler models")
FineViolin <- FineViolin + theme(axis.text.x = element_text(angle=45, hjust=1))
FineViolin
```



13. Using ggplot, create a histogram that shows the frequency of ticket issuances per hour of the day. Use different colors to depict different types of violation descriptions. In a couple of sentences, describe what your histogram is showing you. What meaning could someone take from this? Why (or why not) does this histogram make sense, in terms of what it says about parking violations and times of day?

Hide

```
TicketHist <- ggplot(Chrysler, aes(x=Issue.time, fill=Violation.Description)) + geom_histogram(color="black")
TicketHist <- TicketHist + ggtitle("Ticket Issuance Per Hour") + xlab("Time Per Day (Military time)") + xlim(0, 2300)
TicketHist
```


Ticket Issuance Per Hour
1250
1000
750
500
250
0
count
0
500
1000
1500
2000
Time Per Day (Military time)
Violation.Description
DISPLAY OF TABS
METER EXP.
NO PARK/STREET CLEAN
NO PARKING
RED ZONE