Homework 6

Max Ryoo (hr2ee)

library(ggplot2) library(car) library(gcookbook) library(MASS) library(Hmisc)

Problem 1

A

```
set.seed(12181998)
library(ggplot2)
library(car)
```

```
## Loading required package: carData
```

```
library(gcookbook)
library(MASS)
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
setwd("/Users/maxryoo/Documents/Fall 2018/STAT3080/HW6")
crash <- read.csv("fatal accidents.csv")
states <- as.vector(unique(crash[1]))
statenames <- states[[1]]
state.list <- lapply(1:length(statenames)
                     , function(x) crash[crash$State == statenames[x],])
```

First I read in the csv file of fatal accidents. I then found the names of the different states and stored it into a states vector. I then stored that into a statenames vector. I then used the lapply function to loop through the statenmaes vector and use the function of finding which entries coreespond to the statename and stored that data frame in indicies of the state.list list.I will not print this due to it being too long

B

```
firstfew <- lapply(1:length(statenames), function(x) state.list[[x]][1:3,] )
print(firstfew)
```

```
## [[1]]
##                  State Case.number Vehicle.count Person.count Day Month
## 1 District of Columbia      110001             1            1   1     1
## 2 District of Columbia      110002             1            1  22     2
## 3 District of Columbia      110003             2            2  25     3
##   Year Day.of.week Hour Minute
## 1 2016           6    4      8
## 2 2016           2   18     44
## 3 2016           6   15     47
##
## [[2]]
##       State Case.number Vehicle.count Person.count Day Month Year
## 27 Maryland      240001             2            2  14     1 2016
## 28 Maryland      240002             1            1  21     2 2016
## 29 Maryland      240003             1            2   4     1 2016
##    Day.of.week Hour Minute
## 27           5   17     23
## 28           1   13     39
## 29           2   19     12
##
## [[3]]
##              State Case.number Vehicle.count Person.count Day Month Year
## 499 North Carolina      370001             2            3  20     1 2016
## 500 North Carolina      370002             1            2  21     1 2016
## 501 North Carolina      370003             2            3  20     1 2016
##     Day.of.week Hour Minute
## 499           4   14     10
## 500           5   14     42
## 501           4   14     18
##
## [[4]]
##         State Case.number Vehicle.count Person.count Day Month Year
## 1847 Virginia      510001             3            3   3     1 2016
## 1848 Virginia      510002             3            6   2     1 2016
## 1849 Virginia      510003             2            2   4     1 2016
##      Day.of.week Hour Minute
## 1847           1   15     27
## 1848           7   14      4
## 1849           2    8     38
##
```

```
## [[5]]
##              State Case.number Vehicle.count Person.count Day Month Year
## 2569 West Virginia      540001             1            1   1     1 2016
## 2570 West Virginia      540002             1            1   2     1 2016
## 2571 West Virginia      540003             1            1   2     1 2016
##      Day.of.week Hour Minute
## 2569           6   10     54
## 2570           7    0      1
## 2571           7    1     10
```

I applied the function of printing the first three indicies of index x, which has a value between 1 and length of the vector oc statenames. then i printed the firstfew vector holding these dataframes.

C

```
numvhe <- lapply(1:length(statenames),
                 function(x) as.data.frame(
                   table(state.list[[x]]$Vehicle.count)))
addstate<-lapply(1:length(statenames),
                 function(x) cbind(
                   numvhe[[x]], State = statenames[x]))
vehicles_bystate <- lapply(addstate,
                           setNames,
                           nm = c("Number of vehicles involved",
                                  "Frequency", "State"))
print(vehicles_bystate)
```

```
## [[1]]
##   Number of vehicles involved Frequency               State
## 1                           1        16 District of Columbia
## 2                           2         8 District of Columbia
## 3                           4         1 District of Columbia
## 4                           7         1 District of Columbia
##
## [[2]]
##   Number of vehicles involved Frequency    State
## 1                           1       257 Maryland
## 2                           2       157 Maryland
## 3                           3        42 Maryland
## 4                           4         9 Maryland
## 5                           5         2 Maryland
## 6                           6         2 Maryland
## 7                           7         2 Maryland
## 8                           9         1 Maryland
```

```
##
## [[3]]
##   Number of vehicles involved Frequency          State
## 1                           1       763 North Carolina
## 2                           2       497 North Carolina
## 3                           3        72 North Carolina
## 4                           4         9 North Carolina
## 5                           5         3 North Carolina
## 6                           6         2 North Carolina
## 7                           7         1 North Carolina
## 8                           9         1 North Carolina
##
## [[4]]
##   Number of vehicles involved Frequency    State
## 1                           1       450 Virginia
## 2                           2       226 Virginia
## 3                           3        35 Virginia
## 4                           4         9 Virginia
## 5                           7         1 Virginia
## 6                           8         1 Virginia
##
## [[5]]
##   Number of vehicles involved Frequency         State
## 1                           1       159 West Virginia
## 2                           2        81 West Virginia
## 3                           3         3 West Virginia
## 4                           4         5 West Virginia
## 5                           6         1 West Virginia
## 6                          12         1 West Virginia
```

I first made numvhe list which as the dataframe of number of vehicles involved as well as the frequency. I then made another list addstate, which adds the corressponding states for each item in the list. I then added the column names for more description and stored into a list called vehicles_bystate, which i printed.

D

```
numaccstate <- table(crash$State,
                     crash$Vehicle.count)
print(numaccstate)
```

```
##
##                          1   2   3   4   5   6   7   8   9  12
##   District of Columbia  16   8   0   1   0   0   1   0   0   0
##   Maryland             257 157  42   9   2   2   2   0   1   0
```

```
##   North Carolina       763 497  72   9   3   2   1   0   1   0
##   Virginia             450 226  35   9   0   0   1   1   0   0
##   West Virginia        159  81   3   5   0   1   0   0   0   1
```

For D I simply used the table function to show a table of accidents with certain vehicle count for each state. Each table entry is a frequency. For example 16 instances of 1 vehicle crash for District of Columbia.

E

```
totalacc<-lapply(1:length(statenames),
                 function(x)
                   sum(as.vector(
                     numaccstate[x,])) )
print(totalacc)
```

```
## [[1]]
## [1] 26
##
## [[2]]
## [1] 472
##
## [[3]]
## [1] 1348
##
## [[4]]
## [1] 722
##
## [[5]]
## [1] 250
```

Doing numaccstate[x,] will give me the row of the table (row 1 if x = 1). I then took the sum of the vector form of that result. I used the lapply function to do it for all the different states, which I stored the list into a totalacc.

F

```
percent <- sapply(1:length(statenames),
                  function(x)
                    round(numaccstate[x, ]/totalacc[[x]],
                          digits = 1))
percentage <- t(percent)
tablep <- as.table(percentage)
row.names(tablep) <- statenames
print(tablep)
```

```
##                          1   2   3   4   5   6   7   8   9  12
## District of Columbia 0.6 0.3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## Maryland             0.5 0.3 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## North Carolina       0.6 0.4 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## Virginia             0.6 0.3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
## West Virginia        0.6 0.3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

I used sapply to divide each entry in numaccstate by the total accidents. The numacccstate is the table used in previous questions and totalacc is the list containing the total number of accidents. I then rounded the answers to the 0.1. I stored it into percentage and made it into a table using as.table(). I then proceeded to give names for the rows (which were the steates names)

G

For all states there were more accidents for 1 vehicles involved. The more vehicles involved the lesser the counts of accidents for all states

H

```
numvehday <- lapply(1:length(statenames),
                    function(x)
                      crash[crash$State ==
                              statenames[x],])
listtable <- lapply(1:length(numvehday),
                    function(x) table(
                      numvehday[[x]]$Day.of.week,
                      numvehday[[x]]$Vehicle.count) )
print(listtable)
```

```
## [[1]]
##
##     1 2 4 7
##   1 2 1 0 0
##   2 2 0 0 0
##   3 4 0 0 0
##   4 1 3 0 0
##   5 2 1 0 0
##   6 3 1 1 0
##   7 2 2 0 1
##
## [[2]]
##
##      1  2  3  4  5  6  7  9
```

```
##   1 40 17  9  0  0  1  0  0
##   2 42 14  5  2  1  0  0  0
##   3 31 14  6  1  0  0  0  0
##   4 30 19  4  2  0  0  1  0
##   5 27 28  3  2  0  1  1  1
##   6 37 30  6  1  0  0  0  0
##   7 50 35  9  1  1  0  0  0
##
## [[3]]
##
##       1   2   3   4   5   6   7   9
##   1 149  50   6   1   0   0   0   0
##   2  93  73  10   0   0   0   1   0
##   3  83  60  12   2   0   1   0   0
##   4  94  85  12   0   2   0   0   1
##   5 107  74  12   2   1   1   0   0
##   6 105  85   6   1   0   0   0   0
##   7 132  70  14   3   0   0   0   0
##
## [[4]]
##
##      1  2  3  4  7  8
##   1 76 29  3  1  0  1
##   2 54 30  3  1  0  0
##   3 51 36  4  1  0  0
##   4 50 28  6  3  0  0
##   5 59 29  7  1  0  0
##   6 63 39  6  1  1  0
##   7 97 35  6  1  0  0
##
## [[5]]
##
##      1  2  3  4  6 12
##   1 22  7  0  1  0  0
##   2 16 13  0  0  0  0
##   3 21 11  1  1  1  0
##   4 29 11  0  1  0  0
##   5 21 14  0  1  0  1
##   6 21 16  2  1  0  0
##   7 29  9  0  0  0  0
```

I first made the list called nuvehday. This list contains the dataframe for each state. List of dataframes. I then used the lapply function to make tables for each state, a table of Days of week and vehicle count and their frequencies for each state. I put it into listtable. This is a list of tables. I then printed the result.

I

There seems to be more accidents in the weekend for some states, which is more probable since people tend to go out to new roads during the weekends to travel. Meanwhile the weekdays its the same regular route, which results in lesser accidents

Problem 2

A

```
meanpop =62.9
sdpop = 13.3
samp <- as.vector(rnorm(25,
                        mean= meanpop,
                        sd=sdpop))
print(samp)
```

```
##  [1] 47.79761 53.13846 50.97998 72.63301 62.69512 70.32539 54.24801
##  [8] 62.40433 64.90221 59.67706 57.51740 61.02747 52.20757 68.74374
## [15] 43.37271 57.00155 61.89380 74.28509 55.58223 69.39055 73.02462
## [22] 49.06476 79.00343 59.30240 79.10613
```

I set manpop and sdpop as the parameters given. I then did a rnorm of 25 entries with the given mean and sd and saved it into samp as a vector. ### B

```
s_mean <- mean(samp)
s_sd <- sd(samp)
p_val <- 2*pnorm(
  abs((s_mean-meanpop)/(sdpop/sqrt(25)))
                  , lower.tail=FALSE)
print(p_val<0.1)
```

```
## [1] FALSE
```

I set s_mean as mean of the samp vectgor. I set s_sd as the sd of samp. I set the p_val with the pnorm with the equation for ztest. I then printed whether the p_val was lower than 0.1, which was false meaning fail to reject. ### C

```
rep_sample <- replicate(10000,
                        rnorm(25,
                              mean=meanpop,
                              sd =sdpop))
rep_mean <- sapply(1:10000, function(x)
  mean(rep_sample[,x]))
rep_p <- sapply(1:10000, function(x)
  2*pnorm(abs((rep_mean[x]-meanpop)/(sdpop/sqrt(25))), lower.tail=FALSE))
rejects <- as.vector(which(rep_p < 0.1))
```

```
proportion <- length(rejects)/10000
print(proportion)
```

```
## [1] 0.0988
```

Reg_sample contains rnorm of 25 entries being done 10,000 times. I then used sapply to this list to find the mean of the 10000 trials. I used the sapply function again to find the pnorm with the new mean that we found previously in rep_mean. I then found which entries have a p value less than 0.1 and stored it as a vector into rejects. I found the length of the rejects function and divided by 10000 to get the proportion.

D

Theoretically, this value should be 0.1 Since the upper and lower bound is each 0.05, which together would be 0.1. The possibility of getting a value that rejects the null hypothesis is 0.05 for the upper and 0.05 for the lower, which together makes up a probability of 0.1. This is similar to the value calcualted by C, although this will change every time you run it it is close to 0.1

References

1. <https://stackoverflow.com/questions/ 10234734/converting-a-numeric-matrix-into-a-data-table-or-data-frame>