Project 6 Solutions

(Abhimanyu Agarwal)

Collaborators: N/A

TA help:

1) Melissa : Helped me go through Question 4 and 5.

Online resources used: N/A

Question 1

```
#Loads into dataframe called "accidents" using read.csv()
dat <- read.csv("/class/datamine/data/fars/7581.csv")

#tapply
tapply(dat$PERSONS, dat$DRUNK_DR, mean, na.rm = T)
```

```
       0        1        2        3        4        6 
2.615540 2.474079 3.660711 5.197917 5.250000 6.000000 
```

```
#Method Preferred
#From question 4 of Project 5, where we worked on solving the problem iteratively. Here the tapply() fu
#It can observed that using tapply() function did a quicker job than using loops to estimate the mean.
#Moreover, it significantly reduces the code complexity. Therefore, using tapply() instead of using loo

# Read in data that maps state codes to state names
state_names <- read.csv("/class/datamine/data/fars/states.csv")

# Create a vector of state names called v
v <- state_names$state

# Set the names of the new vector to the codes
names(v) <- state_names$code

# Create a new column in the dat dataframe with the actual names of the states
dat$mystates <- v[as.character(dat$STATE)]
```

Question 2

```
sort(tapply(dat$DRUNK_DR, dat$mystates, mean))
```

```
       West Virginia          Mississippi                Texas 
           0.1672332            0.1688661            0.1852601 
            New York             Missouri              Alabama 
           0.1983089            0.2078921            0.2136050 
            Arkansas       North Carolina              Indiana 
           0.2650494            0.2678010            0.2717200 
```

```
        North Dakota              Florida       South Carolina
           0.2887538            0.2898366            0.3052830
              Kansas District of Columbia                 Ohio
           0.3133971            0.3153409            0.3161686
          New Mexico            Louisiana        Massachusetts
           0.3184573            0.3241348            0.3308242
             Georgia             Illinois                 Utah
           0.3309584            0.3366005            0.3385707
            Maryland             Virginia             Oklahoma
           0.3422666            0.3426975            0.3484964
                Iowa             Kentucky         Pennsylvania
           0.3609572            0.3637387            0.3793978
               Idaho              Wyoming              Arizona
           0.4049811            0.4110644            0.4126347
            Nebraska            Tennessee         Rhode Island
           0.4146229            0.4159967            0.4188830
          New Jersey            Minnesota          Connecticut
           0.4286125            0.4492386            0.4621138
              Oregon             Michigan           California
           0.4692250            0.4713560            0.4863834
               Maine               Hawaii              Vermont
           0.4916084            0.4952652            0.5126263
              Nevada         South Dakota               Alaska
           0.5127907            0.5132450            0.5223022
             Montana             Colorado            Wisconsin
           0.5269231            0.5326633            0.5350330
          Washington             Delaware        New Hampshire
           0.5498288            0.5642023            0.6094050
```

```
#New Hampshire has the highest average number of drunk drivers per accident
```

Question 3

```
sort(tapply(dat$FATALS, dat$DAY_WEEK, sum, na.rm = T), decreasing = TRUE)
```

```
    7     1     6     5     4     2     3     9
72253 56985 56406 41802 38737 37115 36441     3
```

```
#It is observed that Sundays (Number - 7) have the highest number of fatalities

tapply(dat$FATALS, dat$DAY_WEEK, sum, na.rm = T) / tapply(dat$PERSONS, dat$DAY_WEEK, sum, na.rm = T)
```

```
        1         2         3         4         5         6         7
0.4219423 0.4440018 0.4486371 0.4509598 0.4512842 0.4319915 0.4289692
        9
1.0000000
```

```
#In my opinion, values obtained are pretty high as the ratio is almost close ~0.5 for some days of the
#It is observed that the 'Unknown' category demonstrates the highest average.
#However, it is the unknown category and therefore it is inconclusive.
#From the days in the week, it is observed that most of the days have a relatively close ratio.
#The highest value is on Thursday that is, 0.45
```

Question 4

```
tapply(dat$DRUNK_DR, dat$ALIGNMNT, mean, na.rm = T)
```

```
        1         2         9 
0.3143146 0.4729582 0.2764798 
```

```
#Don't need this, but this is another way of doing it. I jsut wanted to try it
#Straight roads
sum(dat$DRUNK_DR[dat$ALIGNMNT == 1]) / sum(dat$ALIGNMNT == 1)
```

```
[1] 0.3143146
```

```
#Curved roads
sum(dat$DRUNK_DR[dat$ALIGNMNT == 2]) / sum(dat$ALIGNMNT == 2)
```

```
[1] 0.4729582
```

```
#Unknown scenarios
sum(dat$DRUNK_DR[dat$ALIGNMNT == 9]) / sum(dat$ALIGNMNT == 9)
```

```
[1] 0.2764798
```

###Question 5

```
#Total number of fatalities in the respective breaks
tapply(dat$FATALS, cut(dat$HOUR, breaks = c(0,6,12,18,24,99), include.lowest = TRUE), sum)
```

```
  [0,6]  (6,12] (12,18] (18,24] (24,99] 
  93151   49764   96375   98715    1737 
```

```
#Average number of fatalities in the respective breaks
tapply(dat$FATALS, cut(dat$HOUR, breaks = c(0,6,12,18,24,99), include.lowest = TRUE),mean)
```

```
   [0,6]   (6,12]  (12,18]  (18,24]  (24,99] 
1.133293 1.123037 1.128671 1.140331 1.087664 
```

Submitting deliverables: project06.RMD, project06.R and project06.pdf

Pledge

By submitting this work I hereby pledge that this is my own, personal work. I've acknowledged in the designated place at the top of this file all sources that I used to complete said work, including but not limited to: online resources, books, and electronic communications. I've noted all collaboration with fellow students and/or TA's. I did not copy or plagiarize another's work.

> As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together - We are Purdue.