STA234: Homework 3

Derin Gezgin

2025-02-27

Importing Required Libraries Before Start

```
library(ggplot2)
library(mosaicData)
library(mdsr)
library(ggmap)
```

Problem #1 [5 points]

During feature (column) selection using the following dataframe (named sample), “Column1” and “Column2” proved to be non-significant. Hence, we would not like to take these two features into our predictive model. Show in R how will you select all the rows from column 3 to column 6 for the below dataframe named table?

Sample

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6
Name 1	Alpha	12	24	54	0	Alpha
Name 2	Beta	16	32	51	1	Beta
Name 3	Alpha	52	104	32	0	Gamma
Name 4	Beta	36	72	84	1	Delta
Name 5	Beta	45	90	32	0	Phi
Name 6	Alpha	12	24	12	0	Zeta
Name 7	Beta	32	64	64	1	Sigma
Name 8	Alpha	42	84	54	0	Mu
Name 9	Alpha	56	112	31	1	Eta

We can start the solution by creating a dataframe

```
table = data.frame(
    Column_1 = c("Alpha", "Beta", "Alpha", "Beta", "Beta", "Alpha", "Beta",
"Alpha", "Alpha"),
    Column_2 = c(12, 16, 52, 36, 45, 12, 32, 42, 56),
    Column_3 = c(24, 32, 104, 72, 90, 24, 64, 84, 112),
    Column_4 = c(54, 51, 32, 84, 32, 12, 64, 54, 31),
    Column_5 = c(0, 1, 0, 1, 0, 0, 1, 0, 1),
    Column_6 = c("Alpha", "Beta", "Gamma", "Delta", "Phi", "Zeta", "Sigma",
"Mu", "Eta"))
```

```
rownames(table) = c("Name 1", "Name 2", "Name 3", "Name 4", "Name 5", "Name 6", "Name 7", "Name 8", "Name 9")

table

##        Column_1 Column_2 Column_3 Column_4 Column_5 Column_6
## Name 1    Alpha       12       24       54        0    Alpha
## Name 2     Beta       16       32       51        1     Beta
## Name 3    Alpha       52      104       32        0    Gamma
## Name 4     Beta       36       72       84        1    Delta
## Name 5     Beta       45       90       32        0      Phi
## Name 6    Alpha       12       24       12        0     Zeta
## Name 7     Beta       32       64       64        1    Sigma
## Name 8    Alpha       42       84       54        0       Mu
## Name 9    Alpha       56      112       31        1      Eta
```

Following this, we can simply use the column indexing to get columns 3 to 6.

```
table[, 3:6]

##        Column_3 Column_4 Column_5 Column_6
## Name 1       24       54        0    Alpha
## Name 2       32       51        1     Beta
## Name 3      104       32        0    Gamma
## Name 4       72       84        1    Delta
## Name 5       90       32        0      Phi
## Name 6       24       12        0     Zeta
## Name 7       64       64        1    Sigma
## Name 8       84       54        0       Mu
## Name 9      112       31        1      Eta

5/5
```

Problem #2 [30 points]

We will use the PIMA dataset which consists of a population of women who were at least 21 years old, of Pima Indian heritage and living near Phoenix, Arizona, was tested for diabetes according to World Health Organization criteria. There are nine variables, namely

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function

8. Age (years)
9. Class variable for diabetic or not according to WHO (0 or 1)

Part (A)

Import the data from Moodle or shared Google drive, it is called pima.csv. Change the name of the nine columns to preg_times, glucose_test, blood_press, tsk_thickness, serum, bm_index, pedigree_fun, age, class.

```r
pima = read.csv("../../DATA/pima.csv")

colnames(pima) = c("preg_times",
                   "glucose_test",
                   "blood_press",
                   "tsk_thickness",
                   "serum",
                   "bm_index",
                   "pedigree_fun",
                   "age",
                   "class")
```

Part (B)

All patients (768 Observations) in this dataset contains are females at least 21 years old of Pima Indian heritage. All zero values for the biological variables other than number of times pregnant should be treated as missing values. Count how many zeros are there in each variable (column). For any 0 in the data (except for class and preg_times) assign it as an NA.

```r
summary(pima == 0)

##   preg_times      glucose_test    blood_press     tsk_thickness
##   Mode :logical   Mode :logical   Mode :logical   Mode :logical
##   FALSE:657       FALSE:763       FALSE:733       FALSE:541
##   TRUE :111       TRUE :5         TRUE :35        TRUE :227
##     serum          bm_index       pedigree_fun       age
##   Mode :logical   Mode :logical   Mode :logical   Mode :logical
##   FALSE:394       FALSE:757       FALSE:768       FALSE:768
##   TRUE :374       TRUE :11
##     class
##   Mode :logical
##   FALSE:268
##   TRUE :500
```

The summary would show us the number of TRUEs which is 0s in each column.

```r
exclude.columns = c("class", "preg_times")
selected.cols = !(colnames(pima) %in% exclude.columns)
pima[, selected.cols][pima[, selected.cols] == 0] = NA
```

We can set the 0 values to NA while excluding specific columns.

```r
summary(pima == 0)
```

```
##  preg_times       glucose_test     blood_press      tsk_thickness
##  Mode :logical    Mode :logical    Mode :logical    Mode :logical
##  FALSE:657        FALSE:763        FALSE:733        FALSE:541
##  TRUE :111        NA's :5          NA's :35         NA's :227
##    serum           bm_index        pedigree_fun        age
##  Mode :logical    Mode :logical    Mode :logical    Mode :logical
##  FALSE:394        FALSE:757        FALSE:768        FALSE:768
##  NA's :374        NA's :11
##    class
##  Mode :logical
##  FALSE:268
##  TRUE :500
```

When we check again, we can see there are no more 0s in the targeted columns and they are all set to NA.

Part (C)

For class variable, check if it is a factor and if not, then make it a factor with levels 0 replaced with neg (for negative diabetic) and 1 replicated with pos (for positive diabetic),

```r
class(pima$class)
```

```
## [1] "integer"
```

From this, we can see that the class variable is not a factor.

```r
pima$class = factor(pima$class,
                    levels = c(0, 1),
                    labels = c("neg", "pos"))
class(pima$class)
```

```
## [1] "factor"
```

We can convert it to a factor and check it again to see it is converted to a factor.

Part (D)

Make data subsets for four age groups: 21-36, 37-51, 52-66 and 67-81.

```r
subset.21.36 = subset(pima, age %in% 21:36)
subset.37.51 = subset(pima, age %in% 37:51)
subset.52.66 = subset(pima, age %in% 52:66)
subset.67.81 = subset(pima, age %in% 67:81)
```

Part (E)

Create a new factor vector called age.factor, with age in pima data replaced with the age group.

```r
pima$age.factor[pima$age %in% 21:36] = "21-36"
pima$age.factor[pima$age %in% 37:51] = "37-51"
pima$age.factor[pima$age %in% 52:66] = "52-66"
pima$age.factor[pima$age %in% 67:81] = "67-81"

pima$age.factor = factor(pima$age.factor)
class(pima$age.factor)
```

```
## [1] "factor"
```

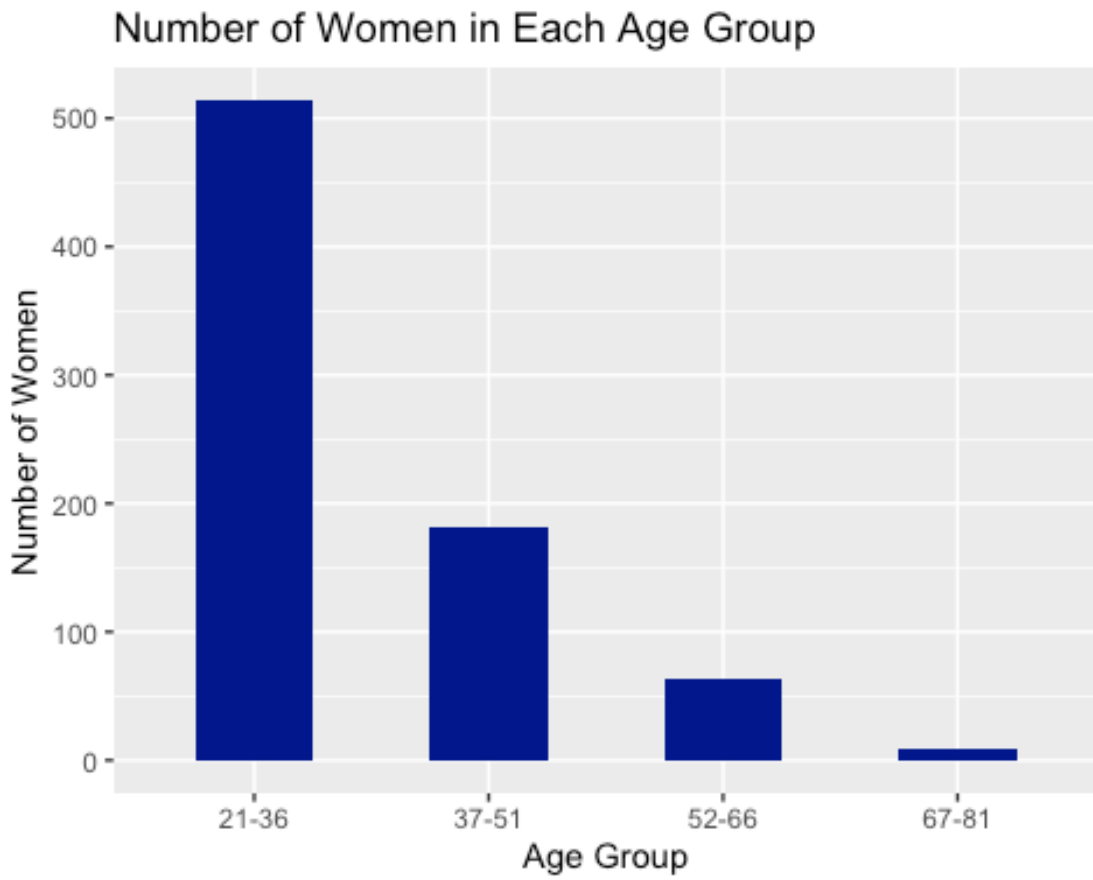
```r
summary(pima$age.factor)
```

```
## 21-36 37-51 52-66 67-81
##   514   181    64     9
```

Part (F) [5 points]

Using the age.factor variable in ggplot, make a barplot for four age groups: 21- 36, 37-51, 52-66 and 67-81 indicating the number of women in each age group.

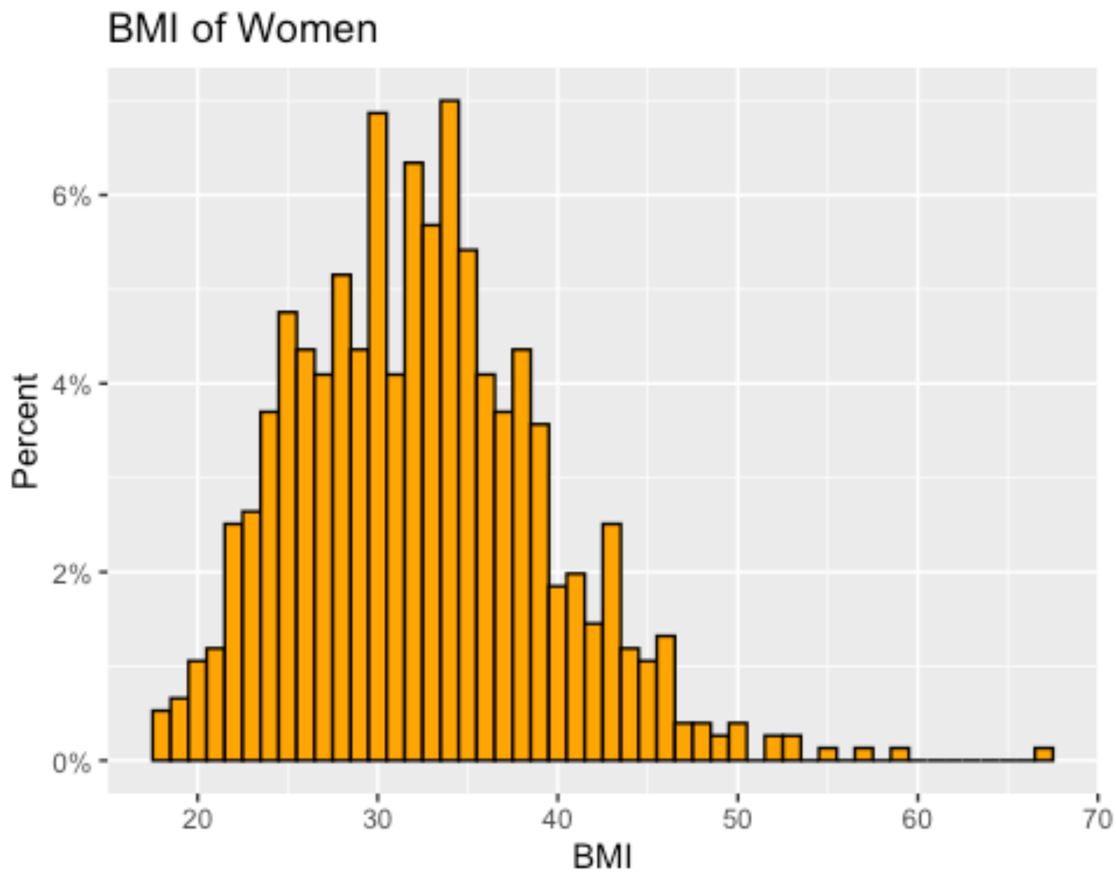
```r
age.plot = ggplot(data = pima,
                  aes(age.factor)) +
    geom_bar(fill = "darkblue",
             width = 0.5) +
    xlab("Age Group") +
    ylab("Number of Women") +
    ggtitle("Number of Women in Each Age Group")

age.plot
```



Part (G) [5 points]

Make a histogram of BMI for all women using ggplot function with percentage on the y-axis.

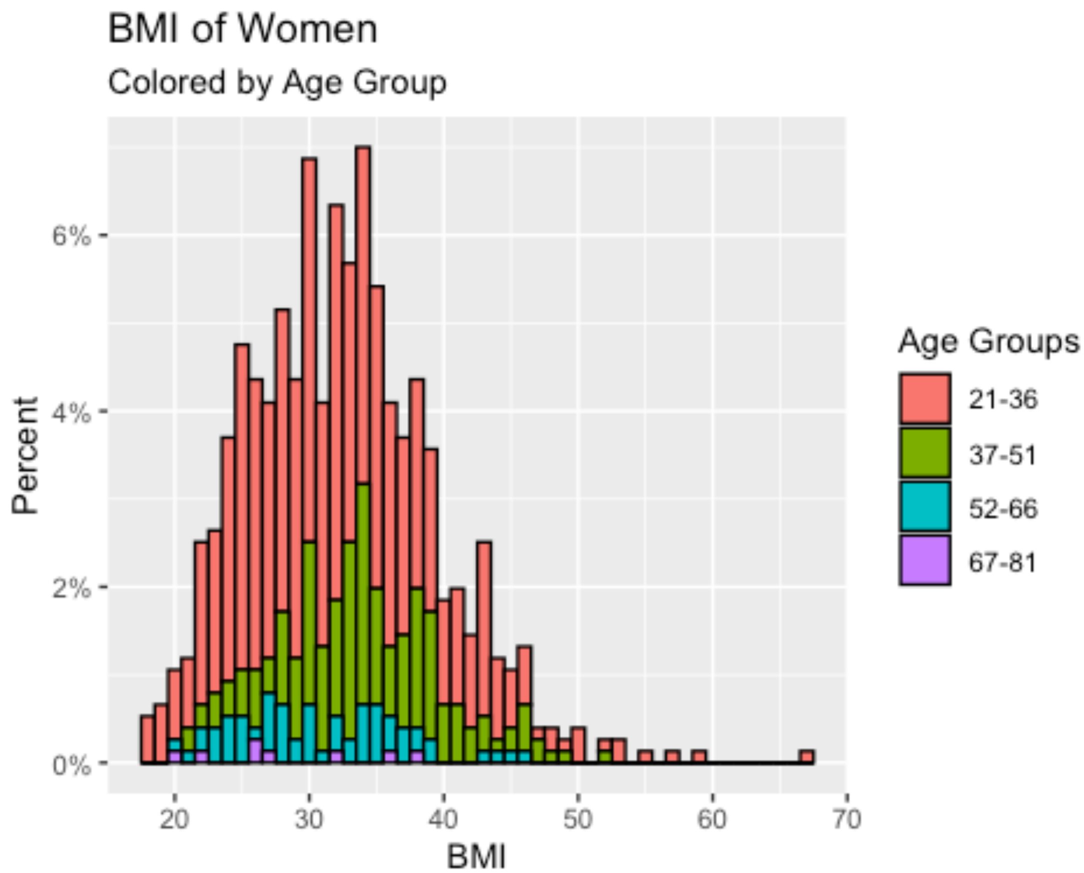
```
BMI.histogram = ggplot(data=pima,
                       aes(x = bm_index)) +
    geom_histogram(color="black",
                   fill="orange",
                   binwidth = 1,
                   aes(y =(..count..)/sum(..count..))) +
    scale_y_continuous(labels=scales::percent,"Percent") +
    xlab("BMI") +
    ggtitle("BMI of Women")

BMI.histogram
```



Part (H) [5 points]

Make a histogram for the BMI of women with different color for each age group with percentage on the y-axis.

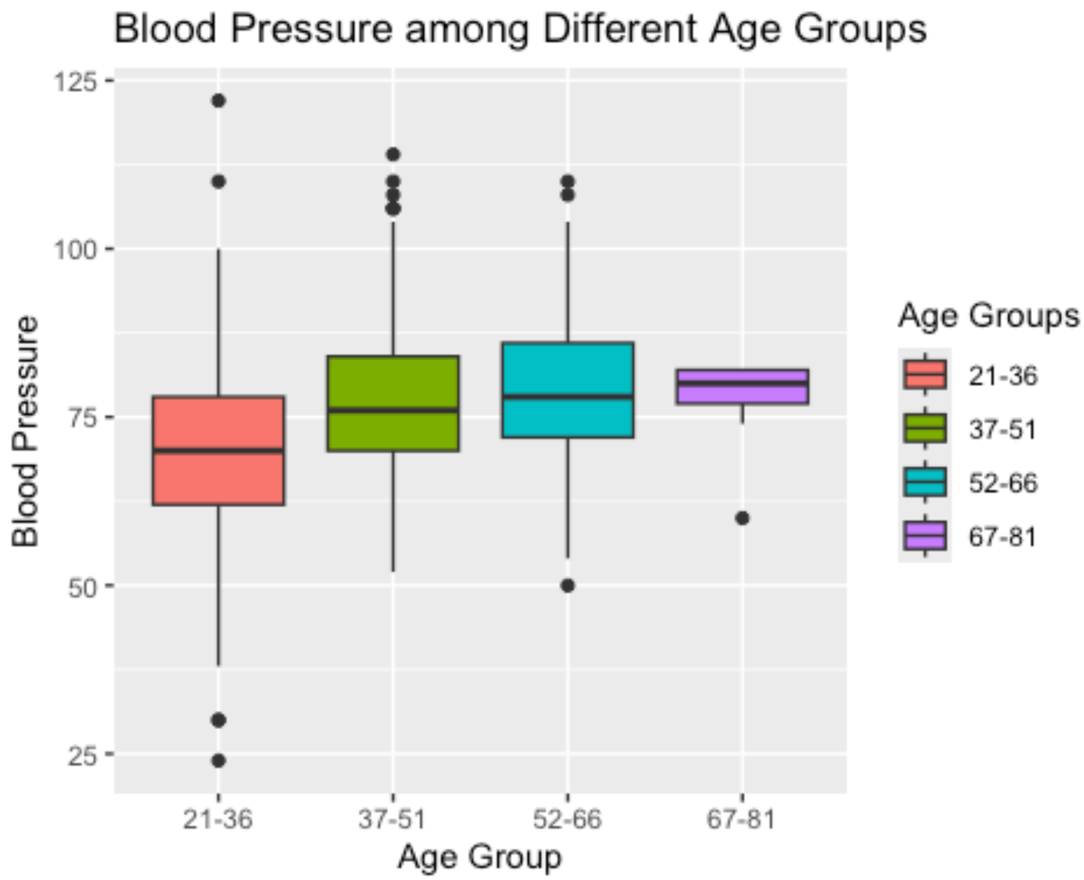
```
BMI.histogram.grouped = ggplot(data = pima,
                               aes(x = bm_index)) +
    geom_histogram(binwidth = 1,
                   color="black",
                   aes(fill = age.factor,
                       y =(..count..)/sum(..count..))) +
    scale_y_continuous(labels=scales::percent,"Percent") +
    xlab("BMI") +
    labs(title = "BMI of Women",
         subtitle = "Colored by Age Group",
         fill = "Age Groups")

BMI.histogram.grouped
```



Part (I) [5 points]

Make comparative boxplots for blood pressure of women in four age groups.

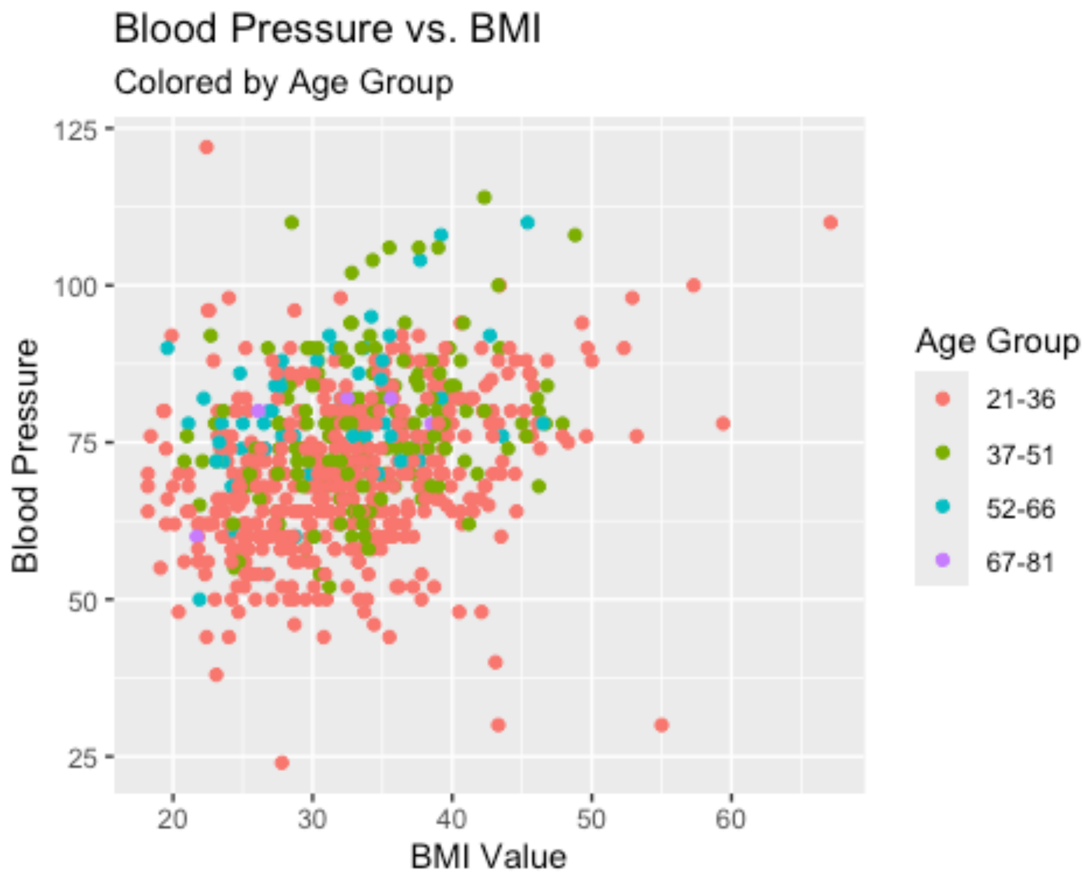
```
pressure.age.boxplot = ggplot(data = pima,
                              aes(x = age.factor,
                                  y = blood_press)) +
    geom_boxplot(aes(fill = age.factor)) +
    xlab("Age Group") +
    ylab("Blood Pressure") +
    ggtitle("Blood Pressure among Different Age Groups") +
    labs(fill = "Age Groups")

pressure.age.boxplot
```



Part (J) [5 points]

Make a scatterplot between blood pressure (y) and BMI (x) using separate colors for different age groups. Comment on the relation.

```
splot.pressure.bmi.grouped = ggplot(data = pima,
                                    aes(x = bm_index,
                                        y = blood_press,
                                        color = age.factor)) +
    geom_point() +
    xlab("BMI Value") +
    ylab("Blood Pressure") +
    labs(title = "Blood Pressure vs. BMI",
         subtitle = "Colored by Age Group",
         color = "Age Group")

splot.pressure.bmi.grouped
```

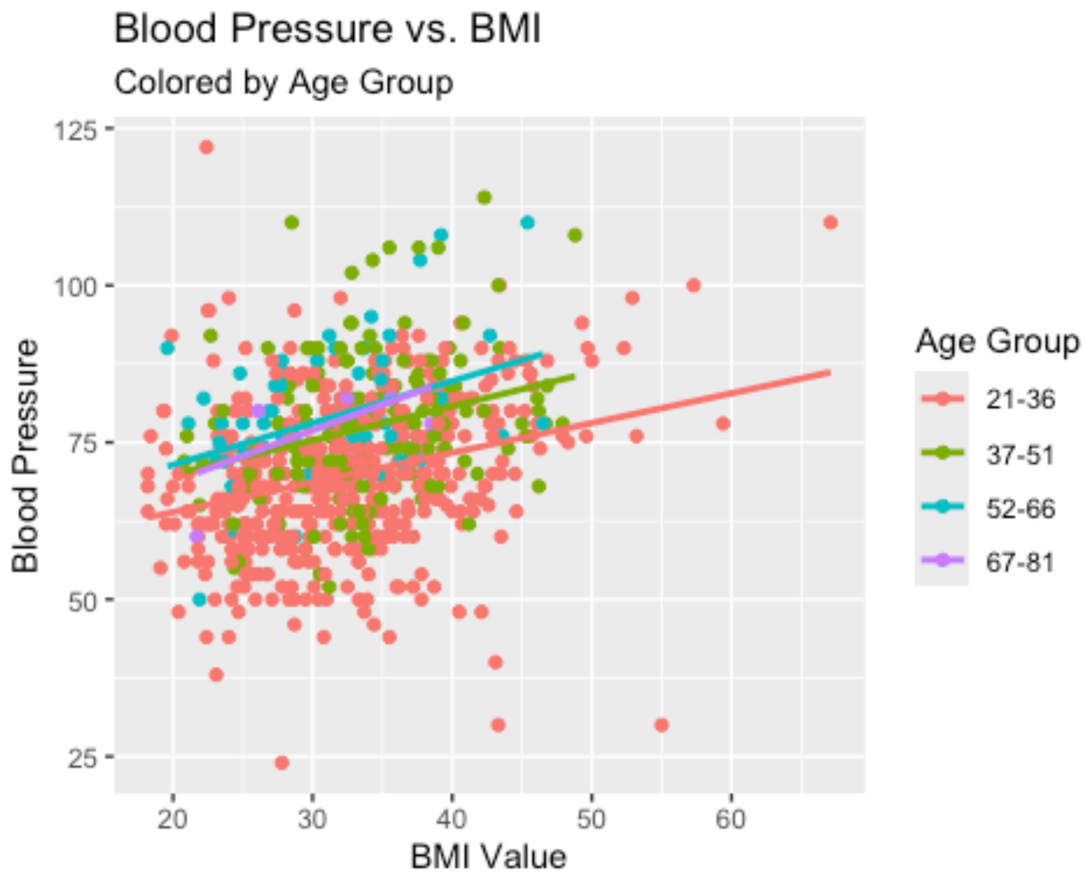


From this scatterplot, we can see that there are some significant outliers for age groups 21-36, 37-51, and 52-66. As we have not added the linear lines yet, it is hard to make a comment about the relationship between the variables for each age group. We can say that there is definitely a positive relationship in the age group 21-36 as we can see the upward trend. I also think that, any linear relationship we find would not be considered as strong as the points are very spread out.

Part (K) [5 points]

Make a layered scatterplot between blood pressure (y) and BMI (x) using separate colors for different age groups and add fitted regression least squares line. Comment of the relations.

```
splot.pressure.bmi.fitted = splot.pressure.bmi.grouped +
    geom_smooth(method = "lm", se = FALSE)

splot.pressure.bmi.fitted

## `geom_smooth()` using formula = 'y ~ x'
```



In this scatterplot, we can see the fitted lines, which support my case in the previous answer. There are a lot of points in the plot that have a high residual to the fit line, so I wouldn't say there is a strong relationship.

According to the fitted lines, in the same BMI value, same individual in group 21-36 have a lower blood-pressure compared to the other group of individuals.

30/30

Problem #3 [30 points]

Using the RailTrail dataset from mosaicData package. You need to install the package using install.packages("mosaicData") in your Rstudio Console, before you run the functions below:

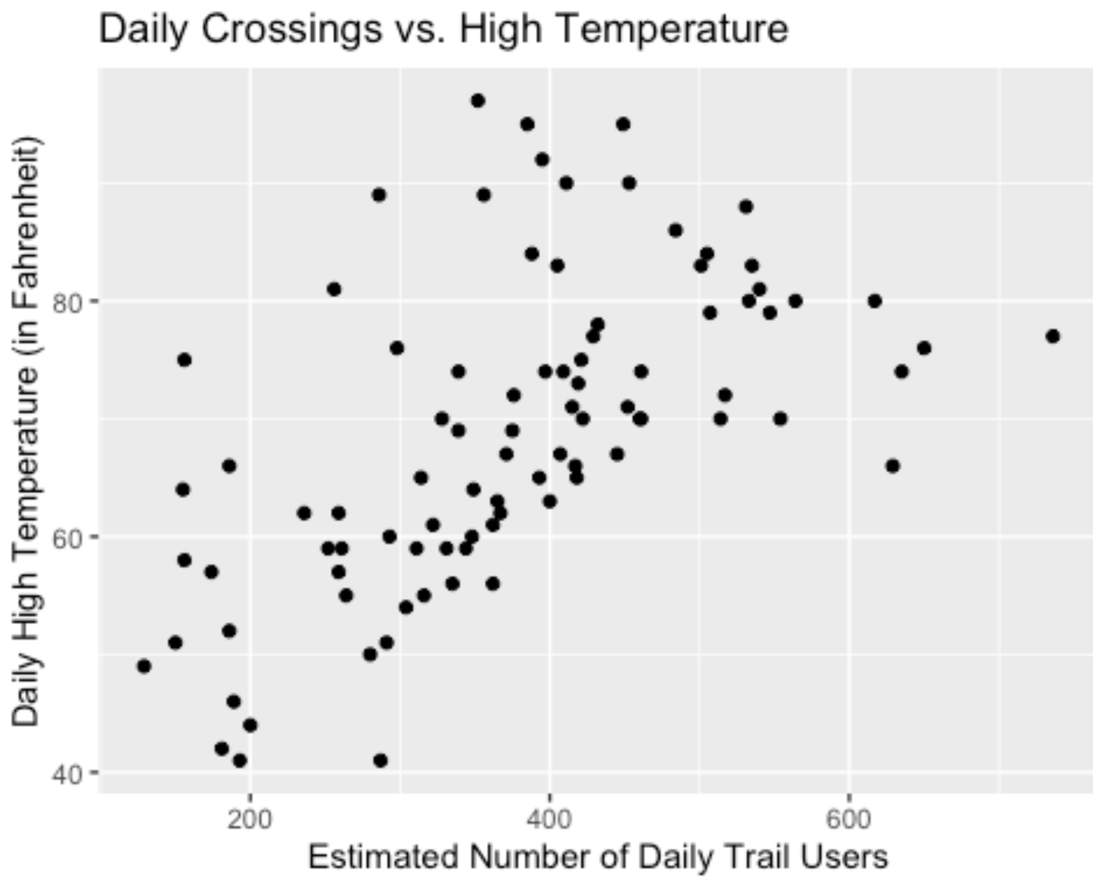
```
head(RailTrail)

##   hightemp lowtemp avgtemp spring summer fall cloudcover precip volume
weekday
## 1       83      50    66.5      0      1    0        7.6   0.00    501
TRUE
## 2       73      49    61.0      0      1    0        6.3   0.29    419
```

```
TRUE
## 3        74       52     63.0       1       0    0        7.5    0.32     397
TRUE
## 4        95       61     78.0       0       1    0        2.6    0.00     385
FALSE
## 5        44       52     48.0       1       0    0       10.0    0.14     200
TRUE
## 6        69       54     61.5       1       0    0        6.6    0.02     375
TRUE
##   dayType
## 1 weekday
## 2 weekday
## 3 weekday
## 4 weekend
## 5 weekday
## 6 weekday
```

Part (1)

Create a scatterplot of the number of crossings per day volume against the high temperature that day. Please note that you can use ?RailTrail to find out more about the dataset.

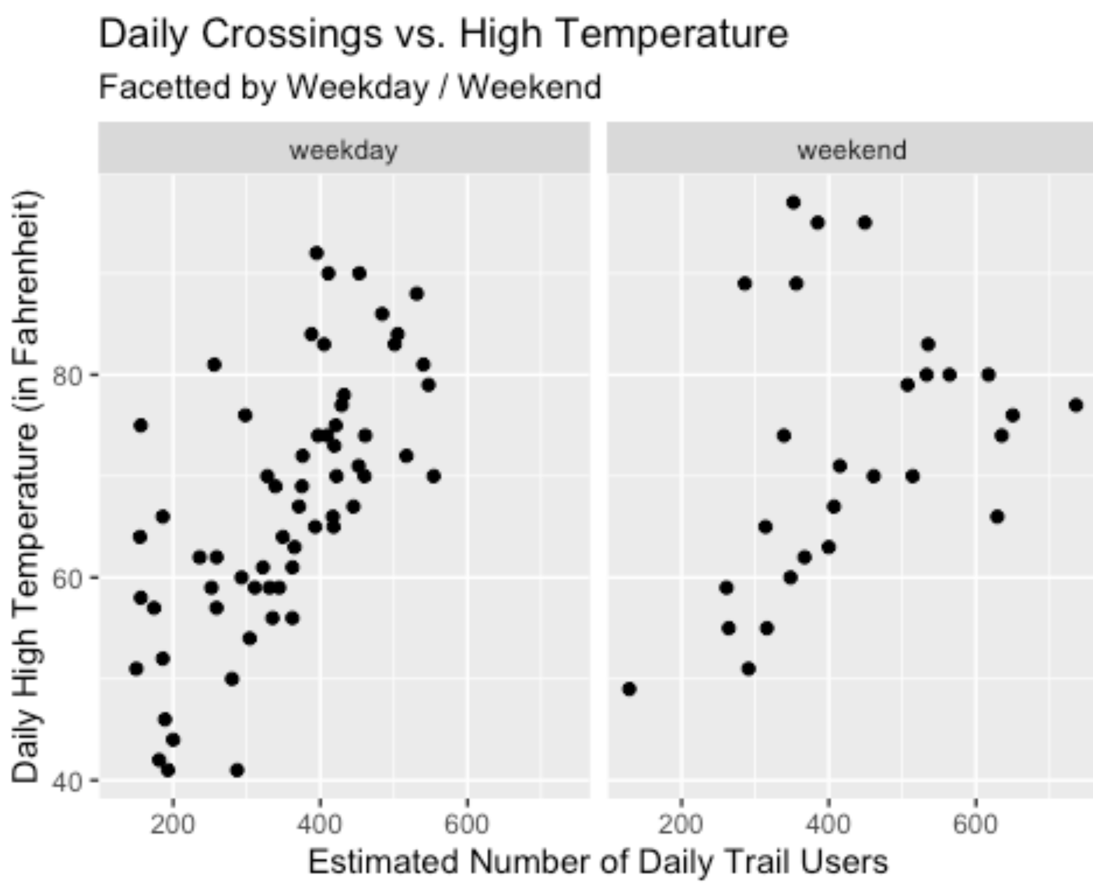
```
splot.crossings.temp = ggplot(data = RailTrail,
                              aes(x = volume,
                                  y = hightemp,)) +
    geom_point() +
    xlab("Estimated Number of Daily Trail Users") +
    ylab("Daily High Temperature (in Fahrenheit)") +
    ggtitle("Daily Crossings vs. High Temperature")

splot.crossings.temp
```



Part (2)

Separate the plot into facets by weekday.

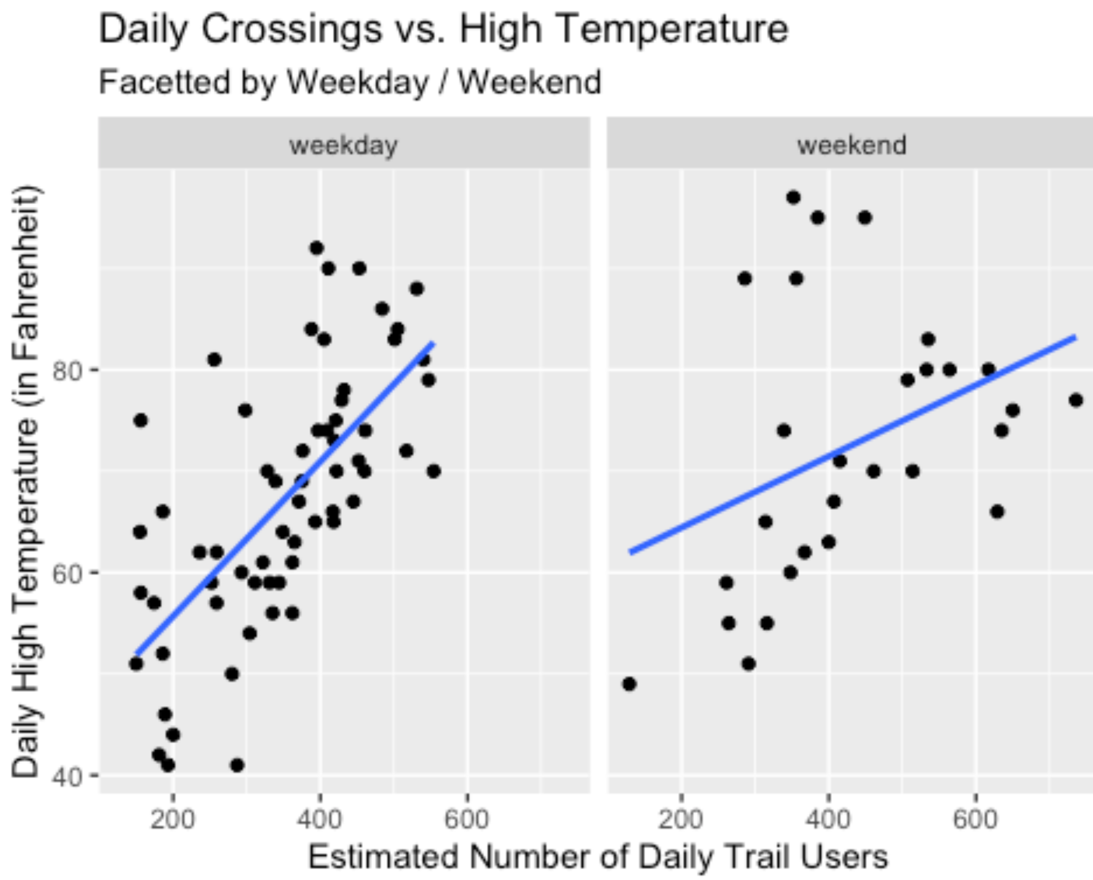
```
splot.crossings.temp.facets = splot.crossings.temp +
    labs(title = "Daily Crossings vs. High Temperature",
         subtitle = "Facetted by Weekday / Weekend") +
    facet_wrap(~dayType, nrow = 1) +
    theme(legend.position = "top")

splot.crossings.temp.facets
```



Part (3)

Add least square fitted regression lines to the two facets.

```
splot.crossings.temp.facets = splot.crossings.temp.facets +
    geom_smooth(method = "lm", se = FALSE)

splot.crossings.temp.facets

## `geom_smooth()` using formula = 'y ~ x'
```



Part (4)

Summarize the information that the data graphic from question 3 conveys.

Compared weekend, we can see that the weekdays have more data points. At the same time, the relationship between the number of trial users (crossings) and the highest temperature (in Fahrenheit) in weekdays is stronger than the weekends as the data points are closer to the fitted line. Both relationships seem positive and we can say that as the estimated number of trail users increase the high temperature (in Fahrenheit) also expected to increase.

30/30

Problem #4 [15 points]

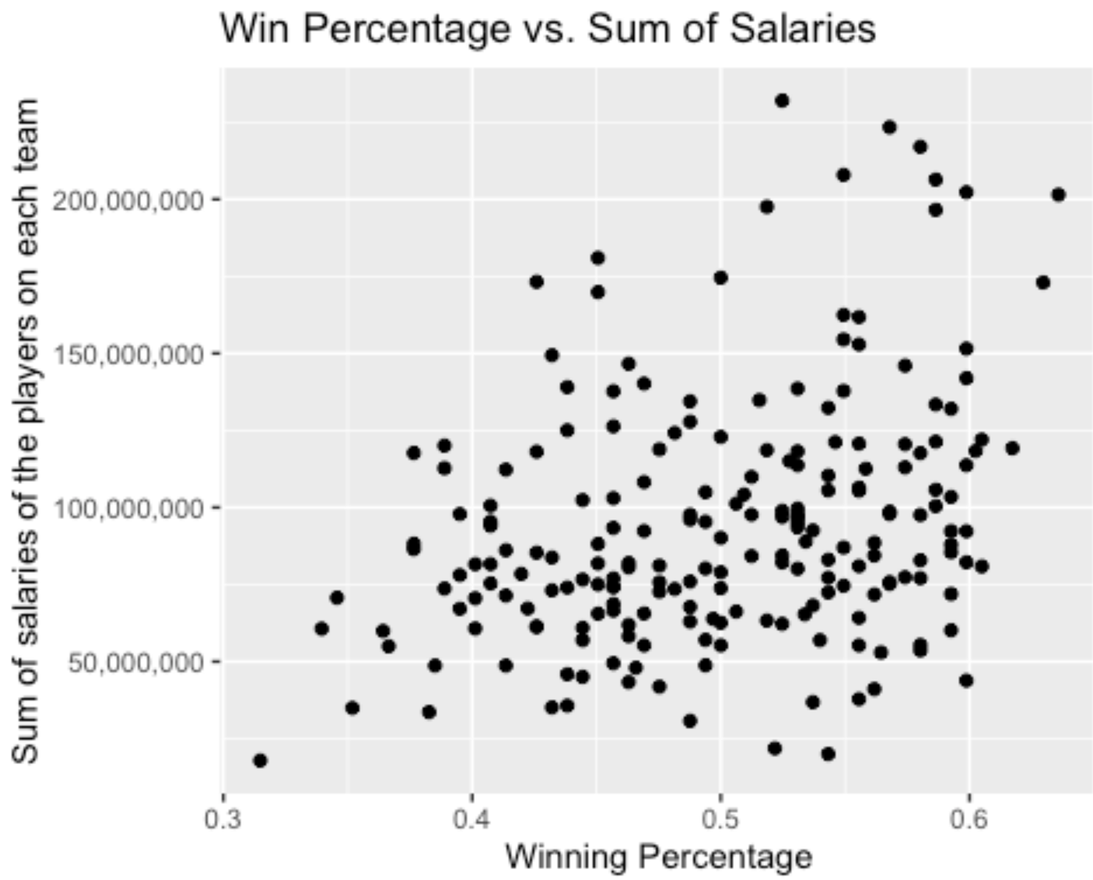
The MLB_teams dataset in the mdsr package contains information about Major League Baseball teams in the past four seasons. There are several quantitative and a few categorical variables present. You may need to install the package using install.packages("mdsr") in your Rstudio Console, before you run the functions below:(Please note that you can use ?MLB_teams to find out more about the dataset.)

```
head(MLB_teams,4)

## # A tibble: 4 × 11
##   yearID teamID lgID      W     L  WPct attendance normAttend   payroll
metroPop
##    <int> <chr>  <fct> <int> <int> <dbl>      <int>      <dbl>     <int>
<dbl>
## 1   2008 ARI    NL       82    80 0.506    2509924      0.584  66202712
4489109
## 2   2008 ATL    NL       72    90 0.444    2532834      0.589 102365683
5614323
## 3   2008 BAL    AL       68    93 0.422    1950075      0.454  67196246
2785874
## 4   2008 BOS    AL       95    67 0.586    3048250      0.709 133390035
4732161
## # ℹ 1 more variable: name <chr>

names(MLB_teams)

##  [1] "yearID"     "teamID"     "lgID"       "W"          "L"
##  [6] "WPct"       "attendance" "normAttend" "payroll"    "metroPop"
## [11] "name"
```

Part (1)

Make a scatterplot to illustrate the relationship between winning percentage and payroll in context.

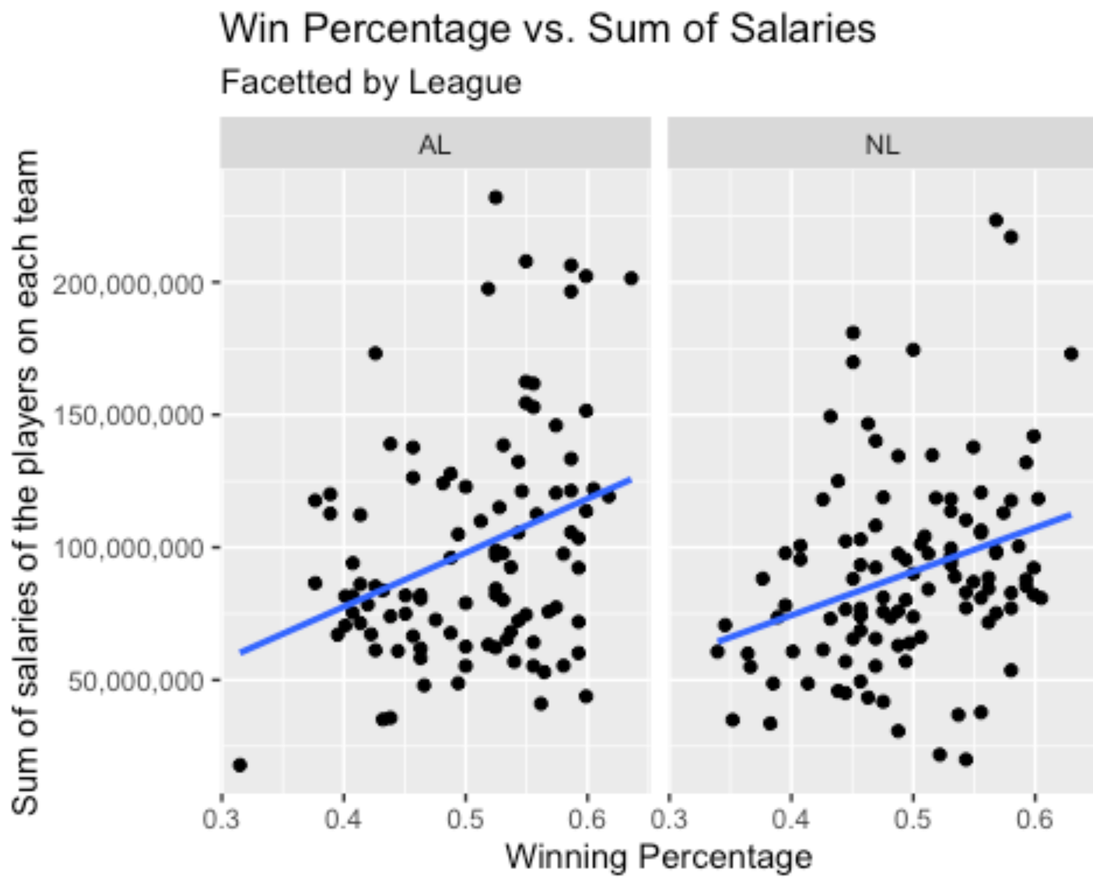
```
splot.win.payroll = ggplot(data = MLB_teams,
                           aes(x = WPct,
                               y = payroll)) +
    geom_point() +
    xlab("Winning Percentage") +
    ylab("Sum of salaries of the players on each team") +
    ggtitle("Win Percentage vs. Sum of Salaries") +
    scale_y_continuous(labels = scales::comma)

splot.win.payroll
```



Part (2)

Add the league in which team played to show more information to make layered or facets. Add smoothed regression line to show the trends.

```
splot.win.payroll.facet.fitted = splot.win.payroll +
    facet_wrap(~lgID, nrow = 1) +
    theme(legend.position = "top") +
    geom_smooth(method = "lm", se = FALSE) +
    labs(title = "Win Percentage vs. Sum of Salaries",
         subtitle = "Facetted by League")

splot.win.payroll.facet.fitted

## `geom_smooth()` using formula = 'y ~ x'
```



15/15

Problem #5 [20 points]

Using the mpg dataset in ggplot2 package answer the following:

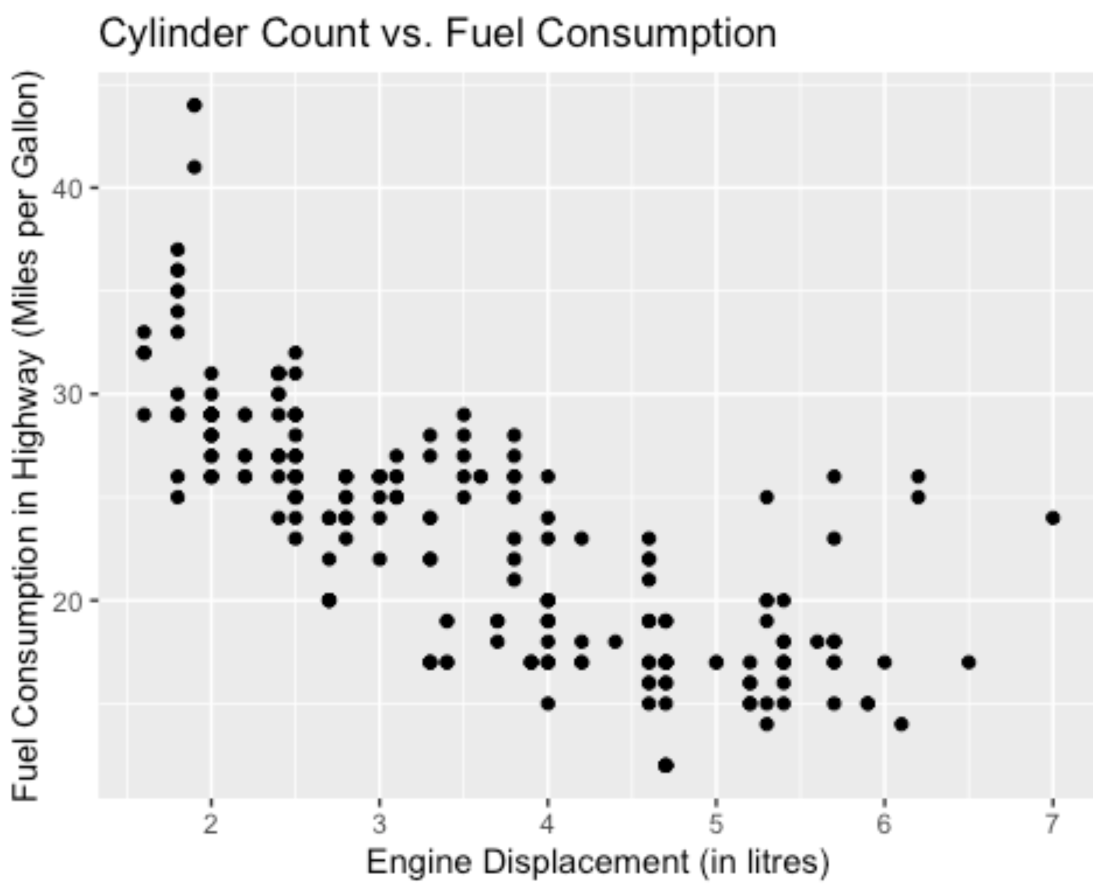
```
data(mpg)
```

Part (A)

Do cars with big engines use more fuel than cars with small engines? Create a scatterplot in ggplot to justify your answer.

```
splot.engine.fuel = ggplot(data = mpg,
                           aes(x = displ,
                               y = hwy)) +
    geom_point() +
    xlab("Engine Displacement (in litres)") +
    ylab("Fuel Consumption in Highway (Miles per Gallon)") +
    ggtitle("Cylinder Count vs. Fuel Consumption")
```

```
splot.engine.fuel
```



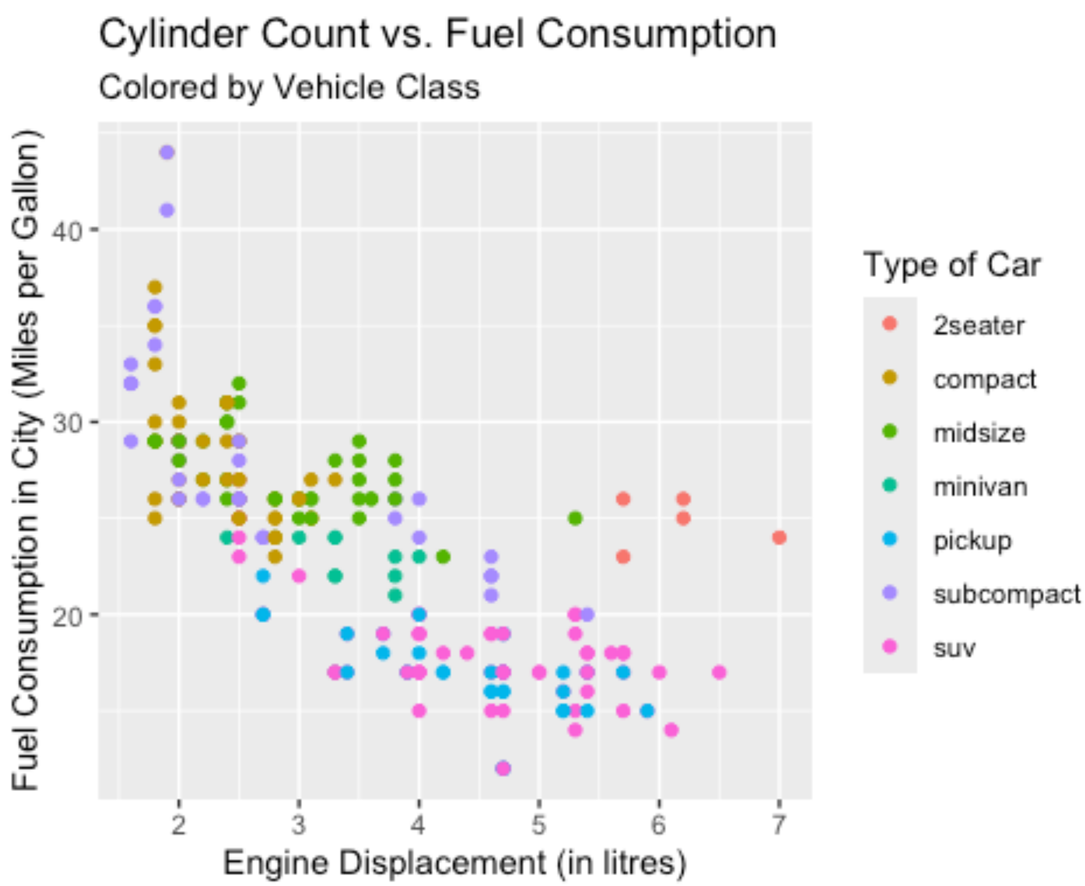
From the scatterplot, we can see that as the engine displacement increases, the fuel consumption in the highway increases as the vehicle can cover a shorter distance with the same amount of fuel. It seems like a quadratic relationship rather than a linear one.

Part (B)

To display the class of each car, use colors in the above scatterplot of displ versus hwy variables.

```
splot.engine.fuel.type = ggplot(data = mpg,
                                aes(x = displ,
                                    y = hwy,
                                    color = class)) +
    geom_point() +
    xlab("Engine Displacement (in litres)") +
    ylab("Fuel Consumption in City (Miles per Gallon)") +
    labs(title = "Cylinder Count vs. Fuel Consumption",
        subtitle = "Colored by Vehicle Class",
```

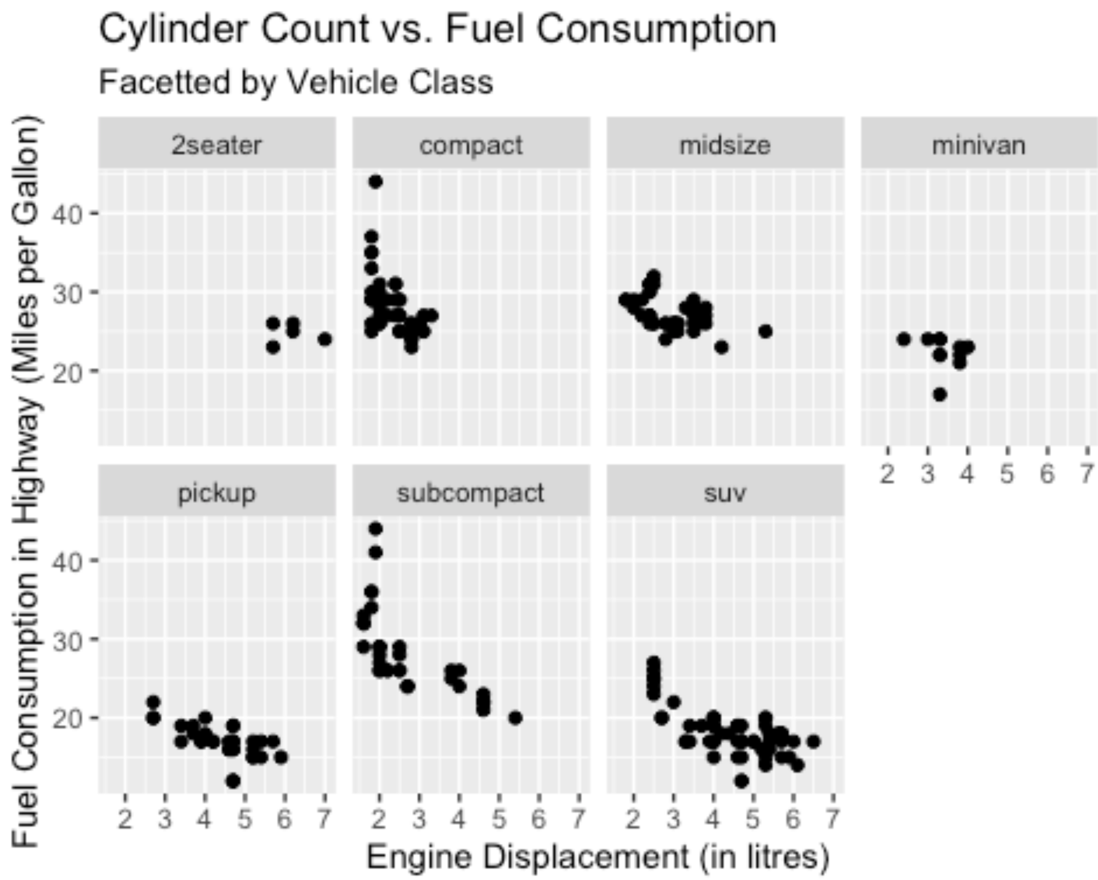
```
        color = "Type of Car")

splot.engine.fuel.type
```



Part (C)

Use facets to display the scatter plots for the class of each car.

```
splot.engine.fuel.facet = splot.engine.fuel +
    geom_point(size=0.5) +
    facet_wrap(~class, nrow = 2) +
    theme(legend.position = "top") +
    labs(title = "Cylinder Count vs. Fuel Consumption",
         subtitle = "Facetted by Vehicle Class")

splot.engine.fuel.facet
```



Part (D)

Using geom_smooth() to make scatter plot for displ vs hwy for each category in variable drv which describes a car's drivetrain. Use default method (do not specify method=lm) to get curved fits. Check class of drv variable and make sure it is a factor so R can make the right plot for all levels.

We can start by checking the class of `drv`

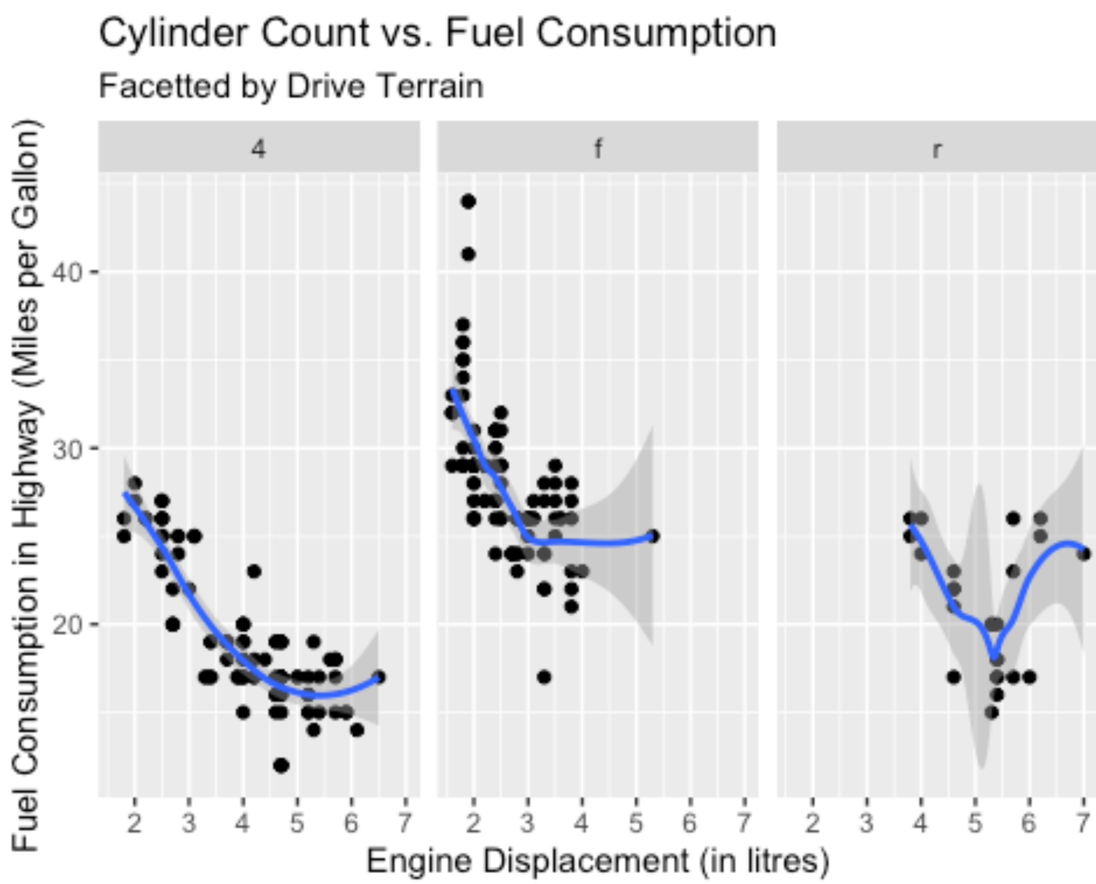
```
class(mpg$drv)

## [1] "character"
```

We can see that the type of the drive-train variable is character. We can use the built-in `factor` function to convert it to a factor and check its type again to confirm the conversion.

```
mpg$drv = factor(mpg$drv)
class(mpg$drv)

## [1] "factor"
```

Following this, we can create the scatterplot.

```
splot.engine.fuel.facet.drv = splot.engine.fuel +
    facet_wrap(~drv) +
    geom_smooth() +
    labs(title = "Cylinder Count vs. Fuel Consumption",
         subtitle = "Facetted by Drive Terrain")

splot.engine.fuel.facet.drv

## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



20/20

Project Problem [30 points]

Part 1: Introduction

Tell me what problem you are working on? Why is this problem interesting and important. State specific research questions your group will work on. Introduce recent research done in area related to your problem. You can pack all this together to motivate us. Do keep it short, to the point, and interesting.