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.

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.

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.

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.

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),

class(pima$class)

## [1] "integer"

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

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.

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.

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"

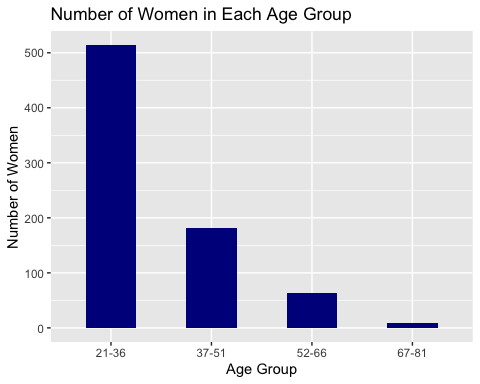
summary(pima$age.factor)

## 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.

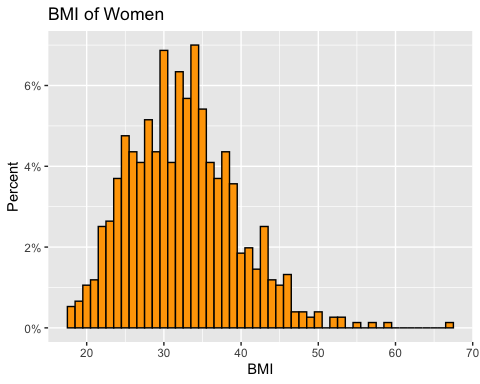
age.plot = ggplot(data = pima,  
 aes(age.factor)) +   
 geom\_bar(fill = "darkblue",  
 width = 0.5) +   
 xlab("Age Group") +   
 ylab("Number of Women") +   
 ggtitle("Number of Women in Each Age Group")  
  
age.plot



## Part (G) [5 points]

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

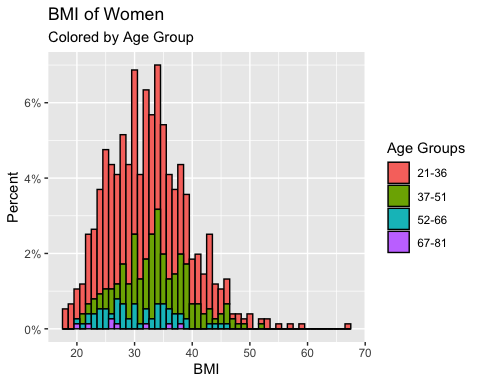
BMI.histogram = ggplot(data=pima,  
 aes(x = bm\_index)) +  
 geom\_histogram(color="black",  
 fill="orange",  
 binwidth = 1,  
 aes(y =(..count..)/sum(..count..))) +  
 scale\_y\_continuous(labels=scales::percent,"Percent") +   
 xlab("BMI") +   
 ggtitle("BMI of Women")  
  
BMI.histogram



## 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.

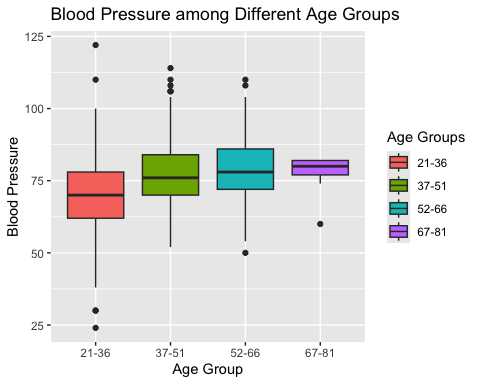
BMI.histogram.grouped = ggplot(data = pima,  
 aes(x = bm\_index)) +  
 geom\_histogram(binwidth = 1,  
 color="black",  
 aes(fill = age.factor,  
 y =(..count..)/sum(..count..))) +  
 scale\_y\_continuous(labels=scales::percent,"Percent") +  
 xlab("BMI") +   
 labs(title = "BMI of Women",  
 subtitle = "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.

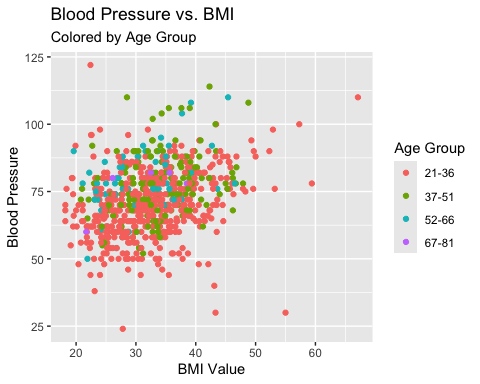
pressure.age.boxplot = ggplot(data = pima,  
 aes(x = age.factor,  
 y = blood\_press)) +   
 geom\_boxplot(aes(fill = age.factor)) +  
 xlab("Age Group") +   
 ylab("Blood Pressure") +   
 ggtitle("Blood Pressure among Different Age Groups") +   
 labs(fill = "Age Groups")  
  
pressure.age.boxplot



## 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



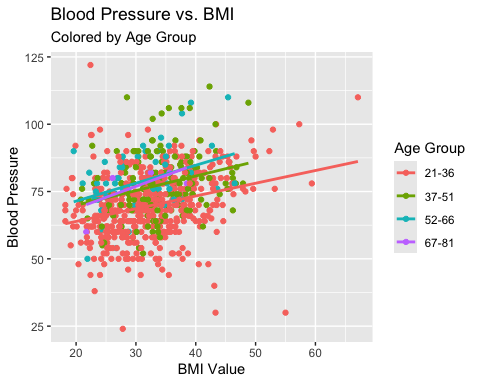
From this scatterplot, we can see that there are some significant outliers for age groups 21-36, 37-51, and 52-66. As we have not added the linear lines yet, it is hard to make a comment about the relationship between the variables for each age group. We can say that there is definitely a positive relationship 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:

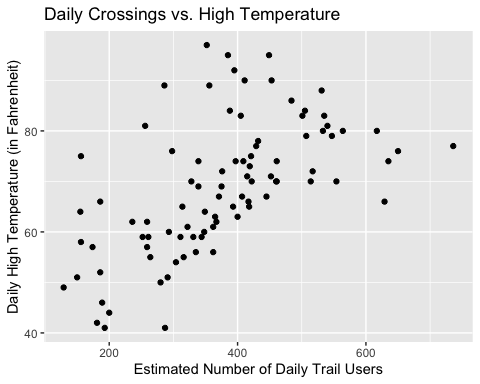
head(RailTrail)

## 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.

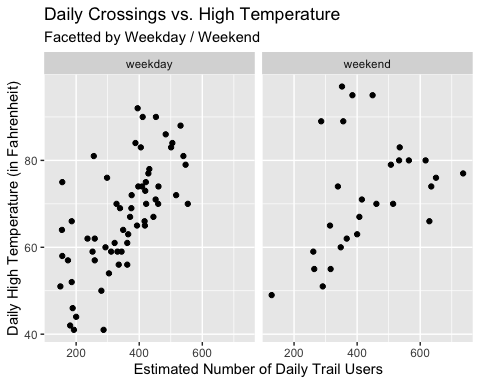
splot.crossings.temp = ggplot(data = RailTrail,  
 aes(x = volume,  
 y = hightemp,)) +  
 geom\_point() +  
 xlab("Estimated Number of Daily Trail Users") +  
 ylab("Daily High Temperature (in Fahrenheit)") +  
 ggtitle("Daily Crossings vs. High Temperature")  
  
splot.crossings.temp



## 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

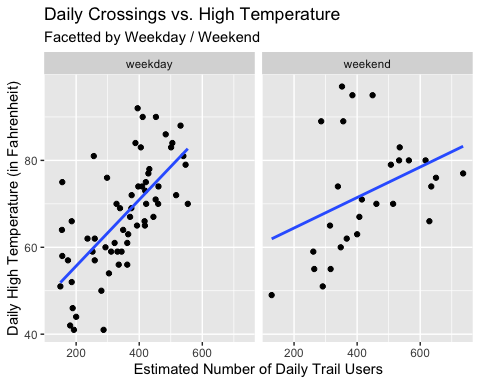


## Part (3)

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>

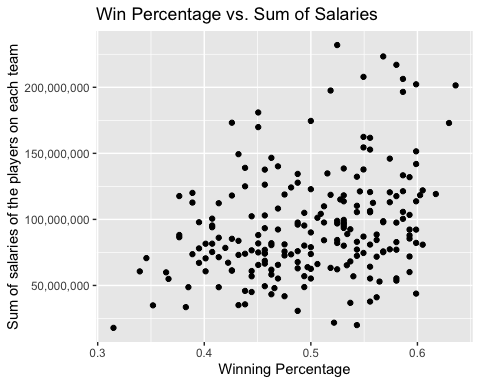
names(MLB\_teams)

## [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

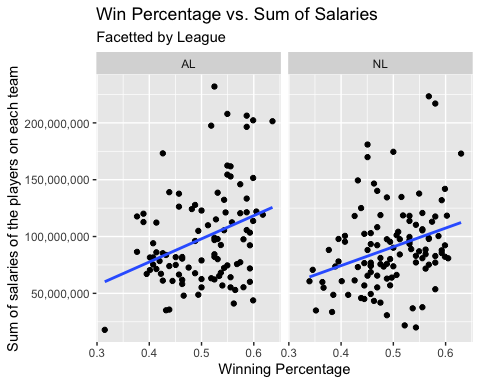


## Part (2)

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]

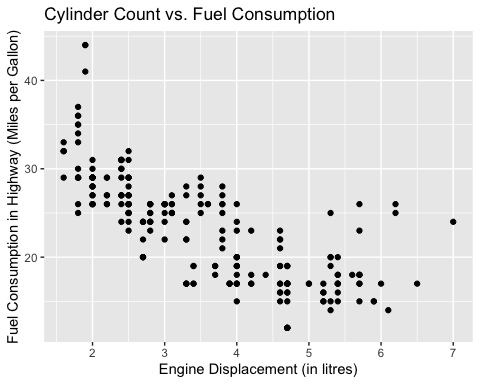
Using the mpg dataset in ggplot2 package answer the following:

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

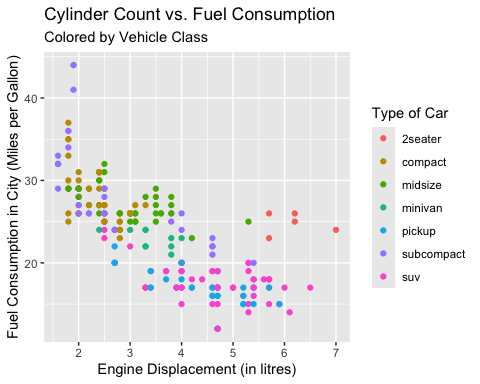


From the scatterplot, we can see that as the engine displacement increases, the fuel consumption in the highway increases as the vehicle can cover a shorter distance with the same amount of fuel. It seems like a quadratic relationship rather than a linear one.

## Part (B)

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

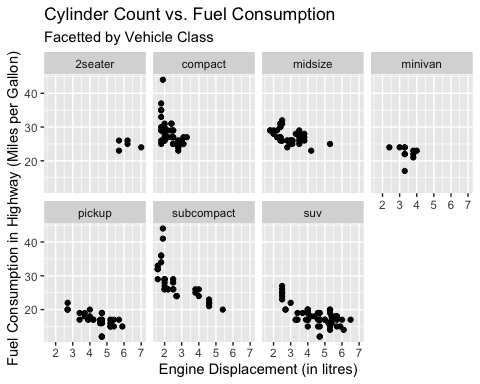
splot.engine.fuel.type = ggplot(data = mpg,  
 aes(x = displ,  
 y = hwy,  
 color = class)) +   
 geom\_point() +   
 xlab("Engine Displacement (in litres)") +   
 ylab("Fuel Consumption in City (Miles per Gallon)") +   
 labs(title = "Cylinder Count vs. Fuel Consumption",  
 subtitle = "Colored by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

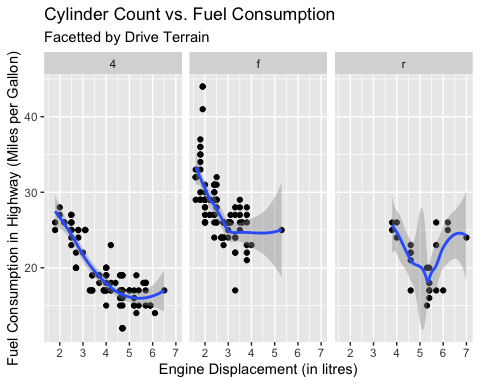
mpg$drv = factor(mpg$drv)  
class(mpg$drv)

## [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.**

Traffic stops are a regular part of our lives, in fact, more than 20 million Americans are stopped each year in the traffic (Pierson et al., 2020). Traffic stops are one of the most common ways of public-police interaction. As police officers conduct these stops, the decision-making process comes down to human judgment, which certainly comes with a certain type of bias. There have been many research projects that focuses on possible bias factors in traffic stops. Most of these studies found that the race of the driver is an important factor influencing the likelihood of being stopped and the outcome of the stop.

According to Pierson et al. (2020) Black and Hispanic drivers are stopped and searched more often than White drivers. However, Black drivers are less likely to be stopped after sunset -compared to the rate of being stopped during the day- when the face of the driver is less visible. It is also pointed out that, the bar to search Black and Hispanic drivers is generally lower than White drivers. Lastly, the study also concludes that the success rates of searches is lower for Hispanic drivers compared to White and Black drivers who has comparable hit rates. Similarly Xu et al. (2024) points out that Black drivers are stopped at higher rates compared to their proportion in the traffic.

In my data-analysis project, I am planning to focus on the demographical analysis of the traffic stops conducted in San Francisco between 2007 and 2016. I already found a research project conducted by *San Francisco Bay Area Planning and Urban Research Association (SPUR)* on San Francisco traffic stop data which only covers the 2019 data. San Francisco Bay Area Planning and Urban Research Association (2023) also has similar findings of Pierson et al. (2020), as it shows Black and Hispanic drivers are stopped more than their share in the population while Black drivers have significantly lower citation rate compared to White and Hispanic drivers. In fact, according to the SPUR study, more than half of the Black drivers who are stopped do not end up with citation at all. At the same time, different than the previous studies I was able to find, I am planning to focus on the locational aspects of the traffic stops and analyze the relationship between the time, outcome, demographics, and the location of the stop.

I can list my possible research questions for my traffic stops data of San Francisco between 2007 and 2016 as the following:

1. What is the relationship between being stopped and the general demographics?
   * How does this relationship change when we split the data into separate times of the day?
2. How does the outcome of the traffic stop (warning, citation, search, arrest) relate to the racial demographics of the driver?
3. How does the amount of drivers stopped vary by time of the day and day of the year?
4. Are certain parts of SF have higher traffic-stop rates?
   * How does this relationship look like if we take race into account as well.
   * How does the outcome of the stop relates to the location

## Part 2: Data

**Tell me about the data resource and explain dimensions of the data, variables in the data, and how does this data relate to your research questions.**

In my research project I am using the San Francisco police stop data which is a part of the Stanford Open Policing Project. The Open Policing Project is an ongoing project, that collects and organizes law enforcement stop data from different counties across the USA. San Francisco data includes details of 905,070 vehicle stops from January 1st, 2007 to June 30th, 2016. The dataset has 22 different variables on different detail of the stop such as date/time/location of the stop, age/race/sex of the driver that is stopped, and the outcome of the stop.

It is important to note that, I will use data from January 1st, 2007 to December 31st, 2015, inclusive. This is crucial as having 2016 until June can skew my data and lead me to incorrect conclusions. At the same time, some of the variables like the location/district/open address, raw data fields won’t be used anywhere in my research project. The final dataset I will use (without the extra variables and half of 2016) will have 864,722 stops and 17 variables.

The variables in this dataset are helpful for my research questions as they provide important details about the demographic information of each traffic stop. At the same time, I am able to access information on what happened during the stop and how did the stop resulted. In the EDA and further parts of my project, these variables will provide me with much flexibility and potential to explore more.

## Part 3: EDA

**Use your dataset to make data visualizations that explain the variables of interest and how information through the graphics provides easy solution for your research questions. Explain your steps on how these visualizations help with your project.**

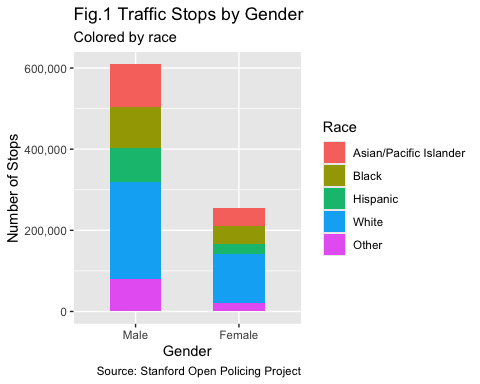
Before starting, I can read my data, convert specific columns into Date objects and also factors.

traffic.data = read.csv("../../DATA/ca\_sf\_vehicle\_2007\_2016.csv")  
  
traffic.data$subject\_sex = factor(traffic.data$subject\_sex,  
 levels = c("male", "female"),  
 labels = c("Male", "Female"))  
  
traffic.data$subject\_race = factor(traffic.data$subject\_race,  
 levels = c("asian/pacific islander",  
 "black",  
 "hispanic",  
 "white",  
 "other"),  
 labels = c("Asian/Pacific Islander",  
 "Black",  
 "Hispanic",  
 "White",  
 "Other"))  
  
traffic.data$date = as.Date(traffic.data$date)  
traffic.data$time = strptime(traffic.data$time, format="%H:%M:%S")  
  
# Outcome is the most severe action taken among (warning, citation arrest).  
# If no action is taken, it is NA. I made this NA values No Action so that  
# I can make it a factor  
traffic.data$outcome[is.na(traffic.data$outcome)] = "No Action"  
  
traffic.data$outcome = factor(traffic.data$outcome,  
 levels = c("citation",  
 "warning",  
 "arrest",  
 "No Action"),  
 labels = c("Citation",  
 "Warning",  
 "Arrest",  
 "No Action"))

As the prompt suggests, in the initial EDA, I worked on getting straightforward answers from my dataset without anything advanced.

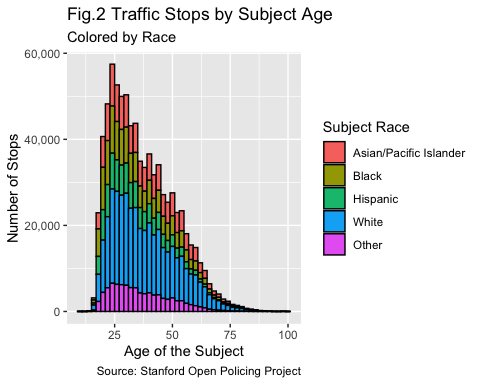
First of all, I wanted to look into the total number of traffic stops by gender and race throughout the time period.

group.race.sex.total = aggregate(traffic.data$raw\_row\_number,  
 by = list(traffic.data$subject\_race,  
 traffic.data$subject\_sex),  
 FUN = length)  
  
colnames(group.race.sex.total)= c("Race", "Gender", "Count")  
  
g = ggplot(data = group.race.sex.total, aes(x = Gender,  
 y = Count,  
 fill = Race)) +  
 geom\_bar(stat = "identity", width = 0.5) +  
 xlab("Gender") +  
 ylab("Number of Stops") +  
 labs(title = "Fig.1 Traffic Stops by Gender",  
 subtitle = "Colored by race",  
 caption = "Source: Stanford Open Policing Project") +  
   
 scale\_y\_continuous(labels = scales::comma)  
  
g



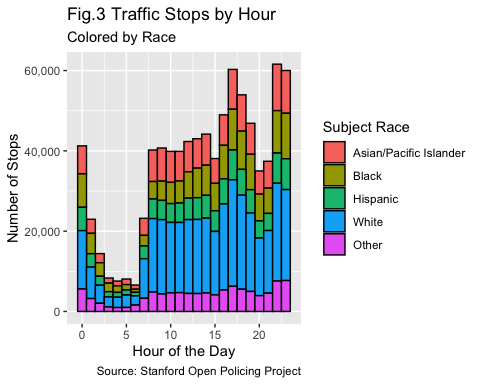
As another demographical data exploration, I checked the distribution of the subject age

g = ggplot(traffic.data, aes(x = subject\_age,  
 fill = subject\_race)) +  
 geom\_histogram(binwidth = 2,  
 color = "black") +  
 xlab("Age of the Subject") +  
 ylab("Number of Stops") +  
 labs(title = "Fig.2 Traffic Stops by Subject Age",  
 subtitle = "Colored by Race",  
 caption="Source: Stanford Open Policing Project") +  
 scale\_y\_continuous(labels = scales::comma) +   
 labs(fill="Subject Race")  
  
g



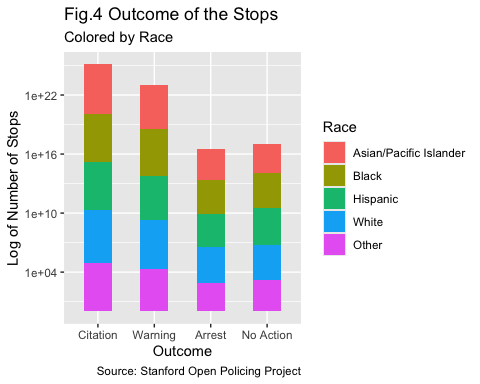
Following this initial exploration on gender, race and age; I focused on the distribution of time of the stops throughout the time-frame.

g = ggplot(traffic.data, aes(x = as.numeric(format(time, "%H")),  
 fill = subject\_race)) +  
 geom\_histogram(binwidth = 1, color = "black") +  
 xlab("Hour of the Day") +  
 ylab("Number of Stops") +  
 scale\_y\_continuous(labels = scales::comma) +  
 labs(title = "Fig.3 Traffic Stops by Hour",  
 subtitle = "Colored by Race",  
 caption = "Source: Stanford Open Policing Project",  
 fill = "Subject Race")  
  
g



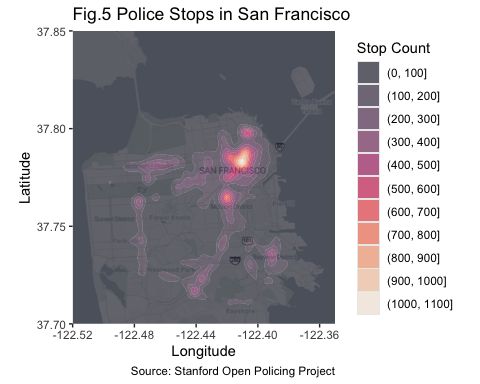
One of the other important parts of the traffic stops is the outcome of the stop. In this part, I plotted the outcome of the stops divided by the race. As the initial graph was hard to read, I applied log transformation to Y-Axis.

stop.race.total = aggregate(traffic.data$raw\_row\_number,  
 by = list(traffic.data$subject\_race,  
 traffic.data$outcome),  
 FUN = length)  
  
colnames(stop.race.total)= c("Race", "Outcome", "Count")  
  
g = ggplot(data = stop.race.total, aes(x = Outcome,  
 y = Count,  
 fill = Race)) +  
 geom\_bar(stat = "identity", width = 0.5) +  
 xlab("Outcome") +  
 ylab("Log of Number of Stops") +  
 labs(title = "Fig.4 Outcome of the Stops",  
 subtitle = "Colored by Race",  
 caption = "Source: Stanford Open Policing Project") +  
 scale\_y\_continuous(trans = "log10")  
  
g



As a bonus, I created a heat-map of the stops in order to see which parts o the city have higher traffic-stop rates.

sf\_map = get\_stadiamap(bbox = c(left = -122.52,  
 bottom = 37.70,  
 right = -122.35,  
 top = 37.85),  
 zoom = 12,  
 maptype = "alidade\_smooth")  
  
ggmap(sf\_map) +  
 geom\_density2d\_filled(data = traffic.data,  
 aes(x = lng, y = lat),  
 alpha = 0.6) +  
 scale\_fill\_viridis\_d(option = "rocket") +  
 labs(title = "Fig.5 Police Stops in San Francisco",  
 fill = "Stop Count",  
 caption = "Source: Stanford Open Policing Project") +  
 xlab("Longitude") +  
 ylab("Latitude")



From these simple EDA steps, I was able to have a general grasp of the dataset. One of my main and strongest finding was, according to Figure 1, male drivers had been stopped significantly more than the female drivers throughout the time period we work on. At the same time, shockingly, in Figure 2, the age-distribution of the stopped drivers does not follow a normal distribution but is significantly right-skewed. On the other hand, the distribution of the stops during the day had peaks at 17.00/22.00/23.00 PM and it significantly drops after 12AM which can be seen in Figure 3. When we check the outcome of the stops in Figure 4, we can see that it seems like pretty balanced among the races for all possible outcomes.

One of the main points I will work on in the upcoming parts of this project is to proportionate my data. Despite having the stop counts for specific races for all the graphs, it is hard to comment on the relationship between the races as we do not have information about their share in the larger population. In the upcoming parts of the project, I can access the census data for San Francisco and compare the subject race counts with the share of that specific race in the population. This can give us a more valuable insight.

Lastly, the density map of the traffic stops (Figure 5) shows us that the traffic stops really accumulate in the center area of the city while following through some major roads in the map. This plot is directly helpful to answer my 3rd research question.

In general, these simple EDA steps are helpful for me to answer some of my research questions I presented in the previous part. With few additions to the dataset and modifications on how I present the data, it is possible to have further information and visualizations from this dataset. As I mentioned, one of the crucial part of my data presentation would be normalizing the data by race distribution in order to have a better comparison.

## Resources

* How to extract time from the HH:MM. [Source](https://stackoverflow.com/questions/10705328/extract-hours-and-seconds-from-posixct-for-plotting-purposes-in-r)
* How to create a heat-map of the San Francisco police stops: [ggmap](https://github.com/dkahle/ggmap) / [Stadia Documentation](https://docs.stadiamaps.com/tutorials/getting-started-in-r-with-ggmap/) / [Density Map](https://ggplot2.tidyverse.org/reference/geom_density_2d.html)

## References

Pierson, E., Simoiu, C., Overgoor, J., Corbett-Davies, S., Jenson, D., Shoemaker, A., Ramachandran, V., Barghouty, P., Phillips, C., Shroff, R., & Goel, S. (2020). A large-scale analysis of racial disparities in police stops across the United States. *Nature Human Behaviour*, *4*(7), 736–745. <https://doi.org/10.1038/s41562-020-0858-1>

San Francisco Bay Area Planning and Urban Research Association. (2023). *Putting an end to biased traffic stops in san francisco*. <https://www.spur.org/news/2023-02-21/putting-end-biased-traffic-stops-san-francisco>

Xu, W., Smart, M., Tilahun, N., Askari, S., Dennis, Z., Li, H., & Levinson, D. (2024). The racial composition of road users, traffic citations, and police stops. *Proceedings of the National Academy of Sciences*, *121*(24). <https://doi.org/10.1073/pnas.2402547121>