Assignment 4

Jordan Lian

3/1/2021

Question 1 (20 points)

From the New York collision dataset create the following parallel coordinate plot. The font type, font case, color, and theme in your visualization can differ. Use the code below to generate the parallel coordinate plot. Critique the visualization and include your improved solution.

Sample Code

```
ggparcoord(df, columns = , groupColumn = ,
           showPoints = TRUE,
           title = "Parallel Coordinate Plot for NY Collisions",
           scale = "globalminmax")
```

Original Plot

```
# Libraries
library(tidyverse)
library(ggplot2)
```

```
# Load dataset, get rid of NA values for BOROUGH
origin_df <- read_csv("ny_accidents.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   `CRASH TIME` = col_time(format = ""),
##   `ZIP CODE` = col_double(),
##   LATITUDE = col_double(),
##   LONGITUDE = col_double(),
##   `NUMBER OF PERSONS INJURED` = col_double(),
##   `NUMBER OF PERSONS KILLED` = col_double(),
##   `NUMBER OF PEDESTRIANS INJURED` = col_double(),
##   `NUMBER OF PEDESTRIANS KILLED` = col_double(),
##   `NUMBER OF CYCLIST INJURED` = col_double(),
##   `NUMBER OF CYCLIST KILLED` = col_double(),
##   `NUMBER OF MOTORIST INJURED` = col_double(),
##   `NUMBER OF MOTORIST KILLED` = col_double(),
```

```
##   COLLISION_ID = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

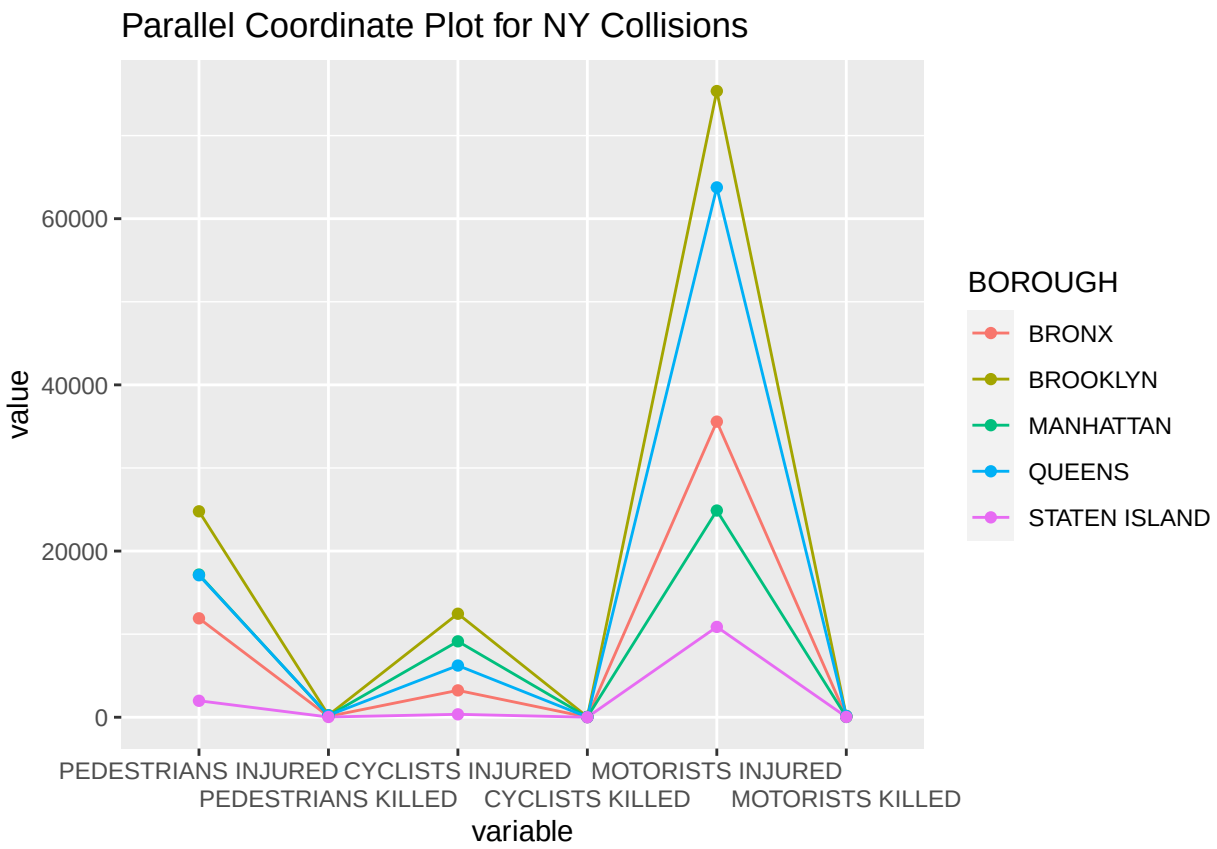
```
df <- origin_df
df <- df[!is.na(df$BOROUGH),]

# Rename columns, store as a vector
colnames(df)[13:18] <- c("PEDESTRIANS INJURED", "PEDESTRIANS KILLED",
                         "CYCLISTS INJURED", "CYCLISTS KILLED",
                         "MOTORISTS INJURED", "MOTORISTS KILLED")
names_vec <- colnames(df)[13:18]

# Aggregate data frame, rename new columns
par_cord <- aggregate(cbind(df$`PEDESTRIANS INJURED`,
                            df$`PEDESTRIANS KILLED`,
                            df$`CYCLISTS INJURED`,
                            df$`CYCLISTS KILLED`,
                            df$`MOTORISTS INJURED`,
                            df$`MOTORISTS KILLED`),
                      by=list(Category=df$BOROUGH), FUN=sum)
colnames(par_cord) <- c("BOROUGH", names_vec)
par_cord
```

```
##         BOROUGH PEDESTRIANS INJURED PEDESTRIANS KILLED CYCLISTS INJURED
## 1         BRONX               11907                106             3235
## 2      BROOKLYN               24784                250            12458
## 3     MANHATTAN               17171                182             9138
## 4        QUEENS               17088                230             6224
## 5 STATEN ISLAND                1984                 31              354
##   CYCLISTS KILLED MOTORISTS INJURED MOTORISTS KILLED
## 1              13             35576               64
## 2              52             75363              154
## 3              29             24879               37
## 4              27             63768              158
## 5               3             10876               39
```

```
# Plot
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggparcoord(par_cord, columns = 2:7, groupColumn = 1,
           showPoints = TRUE,
           title = "Parallel Coordinate Plot for NY Collisions",
           scale="globalminmax") + scale_x_discrete(guide = guide_axis(n.dodge=2))
```



This plot has too many lines, and the killings are so much smaller than the injuries. This makes viewing some of the different lines hard especially when some of them intersect at very similar points. If I used a bar graph, the same problem would occur, so I decided to generate two heat maps.

Improved Solution

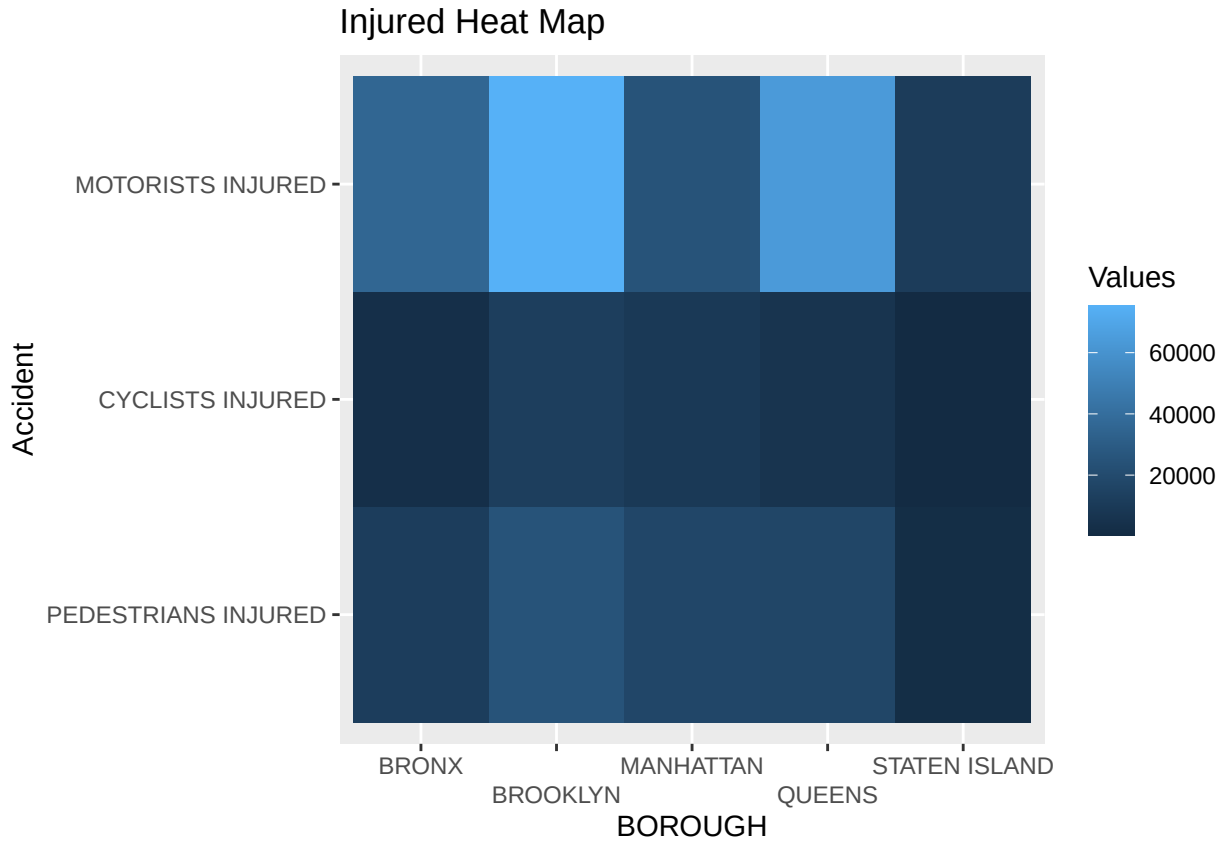
As mentioned above, I generated two heat maps, where one looks at injuries, and the other looks at killings. I stacked all of the columns in the parallel coordinate data frame apart from "BOROUGH", and then I used cbind() to add it back. THen I used ggplot() and geom_tile() to get the heat maps.

```
# Injured data frame
injured <- cbind(par_cord$BOROUGH,stack(par_cord[,c(2, 4, 6)]))
colnames(injured) <- c("BOROUGH", "Values", "Accident")
injured
```

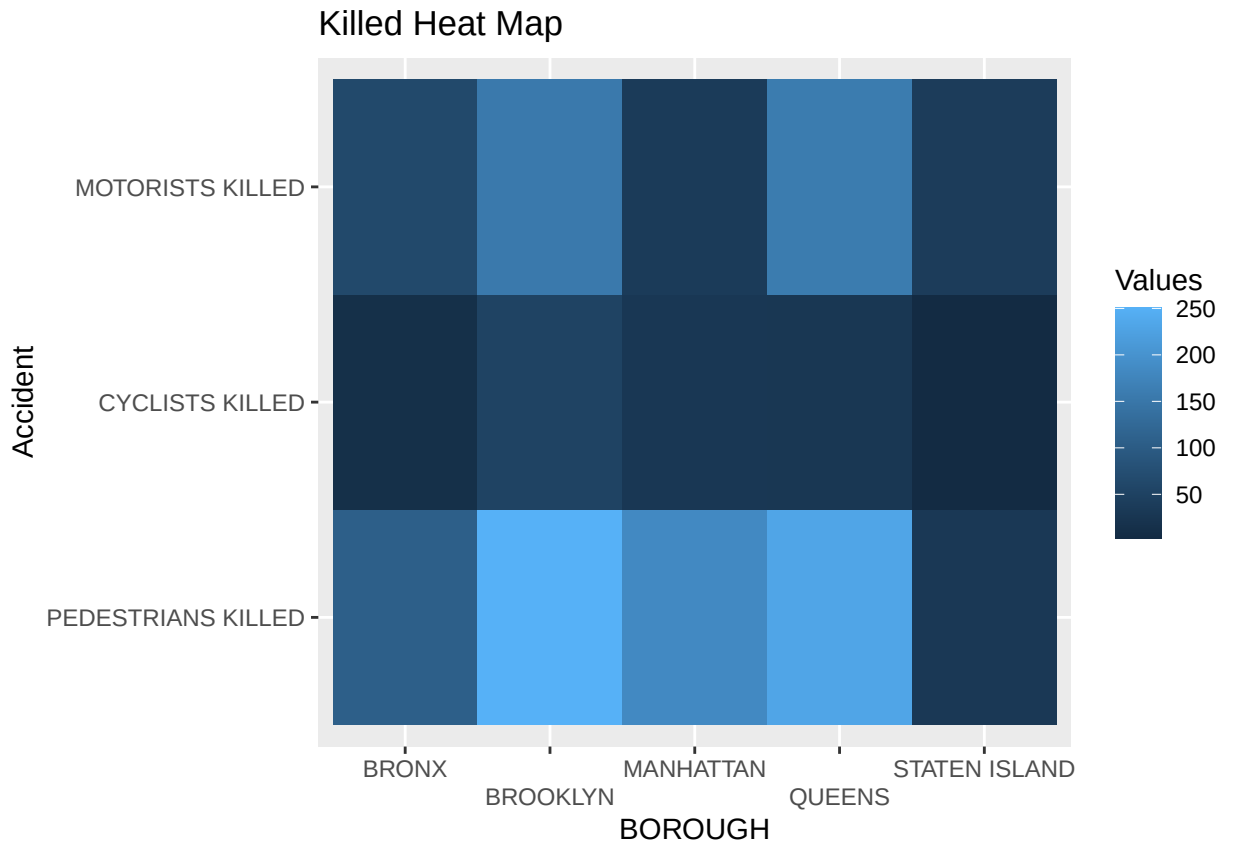
```
##          BOROUGH Values            Accident
## 1          BRONX  11907 PEDESTRIANS INJURED
## 2       BROOKLYN  24784 PEDESTRIANS INJURED
## 3      MANHATTAN  17171 PEDESTRIANS INJURED
## 4         QUEENS  17088 PEDESTRIANS INJURED
## 5  STATEN ISLAND   1984 PEDESTRIANS INJURED
## 6          BRONX   3235    CYCLISTS INJURED
## 7       BROOKLYN  12458    CYCLISTS INJURED
## 8      MANHATTAN   9138    CYCLISTS INJURED
## 9         QUEENS   6224    CYCLISTS INJURED
## 10 STATEN ISLAND    354    CYCLISTS INJURED
```

```
## 11         BRONX  35576   MOTORISTS INJURED
## 12      BROOKLYN  75363   MOTORISTS INJURED
## 13     MANHATTAN  24879   MOTORISTS INJURED
## 14        QUEENS  63768   MOTORISTS INJURED
## 15 STATEN ISLAND  10876   MOTORISTS INJURED
```

```r
# Killed data frame
killed <- cbind(par_cord$BOROUGH, stack(par_cord[,c(3, 5, 7)]))
colnames(killed) <- c("BOROUGH","Values", "Accident")

# Injured heat map
ggplot(injured, aes(fill=Values, y=Accident, x=BOROUGH)) +
  geom_tile() +
  ggtitle("Injured Heat Map") +
  scale_x_discrete(guide = guide_axis(n.dodge=2))
```




```r
# Killed heat map
ggplot(killed, aes(fill=Values, y=Accident, x=BOROUGH)) +
  geom_tile() +
  ggtitle("Killed Heat Map") +
  scale_x_discrete(guide = guide_axis(n.dodge=2))
```



For injuries, motorists seem to get the brunt of them, while pedestrians seemed to be the majority of people killed. This is not surprising given that this is a collision data set, and that pedestrians are the most vulnerable group when it comes to collisions while in traffic. Motorists are the most vulnerable on the road, but are far more likely to sustain bad injuries than die in an accident. With regards to location, it seems like most accidents take place in Brooklyn or Queens.

Question 2 (60 points)

From the link (http://profiles.doe.mass.edu/statereport/sat.aspx) download the average SAT scores for the year 2013-14 and create the following plots using this code. The font type, font case, color, and theme in your visualization can differ. Use the following codes to create the above three plots.

Sample Code

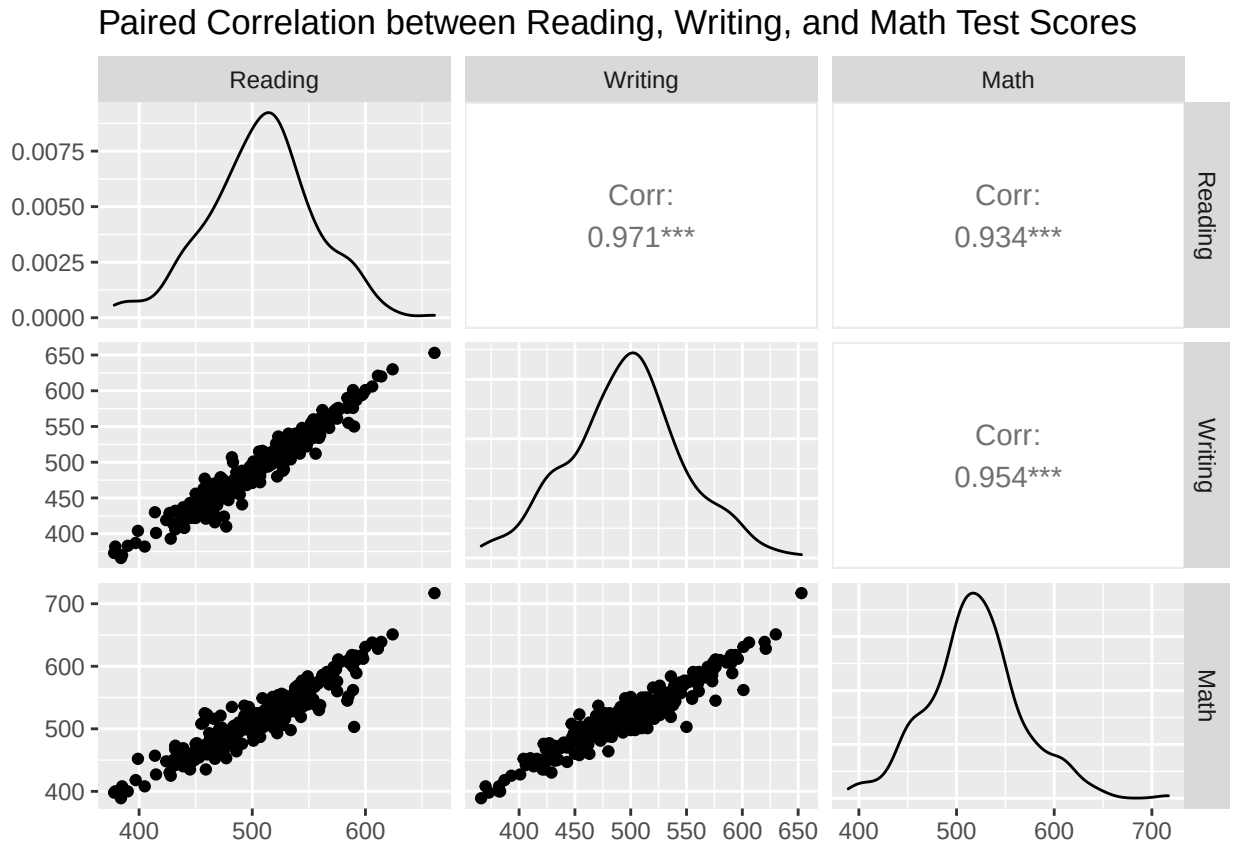
```
# For paired correlation
ggpairs(df)

# For boxplot
ggplot(df, aes(x=, y=)) + geom_boxplot()

# For density
ggplot(df, aes(x=, fill=)) + geom_density(alpha=0.5)
```

Actual Code

```
# Load dataset
library(readxl)
origin_data <- read_excel('sat_performance.xlsx')
data <- origin_data
head(data)
```

```
## # A tibble: 6 x 6
##   `District Name`            `District Code` `Tests Taken` Reading Writing  Math
##   <chr>                      <chr>           <chr>            <dbl>   <dbl> <dbl>
## 1 Abby Kelley Foster Charte~ 04450000        82                 483     460   476
## 2 Abington                   00010000        66                 488     477   496
## 3 Academy Of the Pacific Ri~ 04120000        30                 472     469   521
## 4 Acton-Boxborough           06000000        479                614     620   639
## 5 Adams-Cheshire             06030000        58                 492     485   508
## 6 Advanced Math and Science~ 04300000        97                 606     606   638
```

```
# For paired correlation
ggpairs(data[,4:6]) +
  ggtitle("Paired Correlation between Reading, Writing, and Math Test Scores")
```



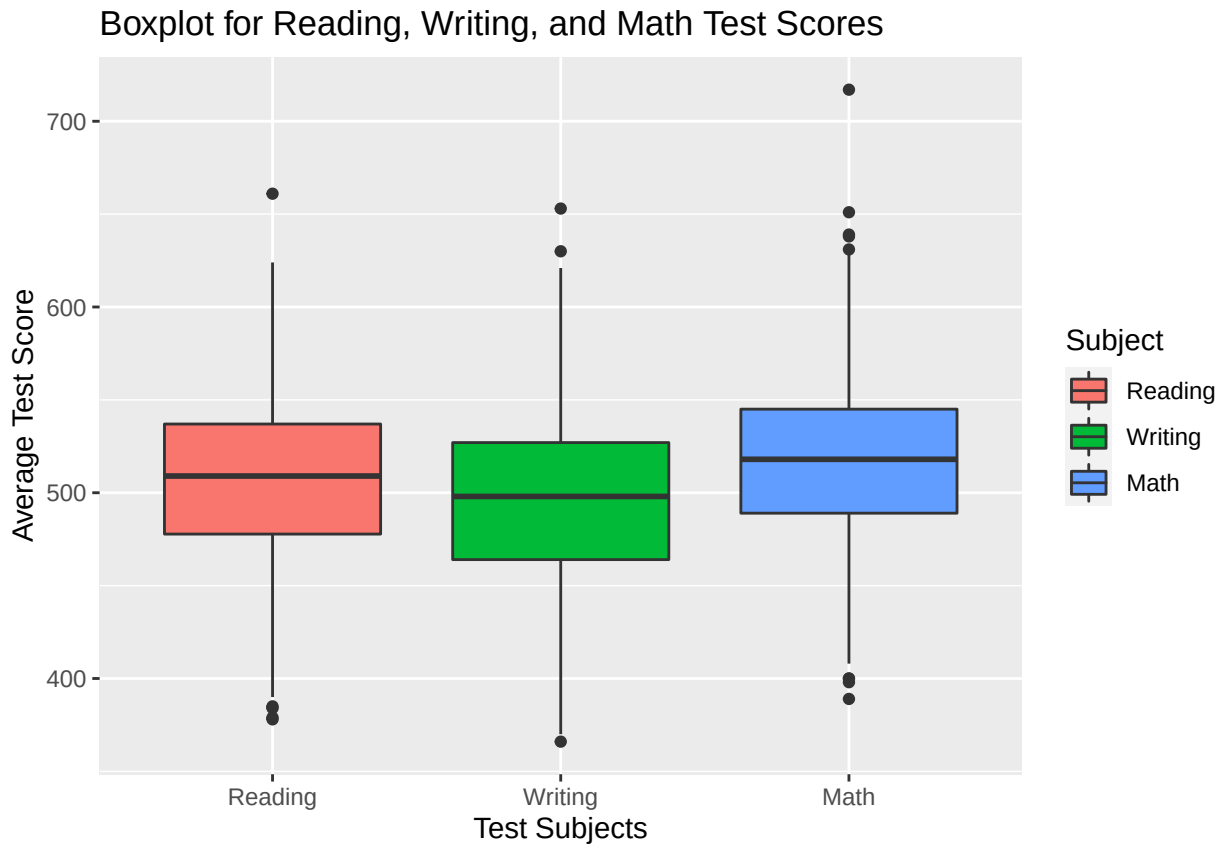
```
# For boxplot
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
ggplot(data = melt(data[,4:6]), aes(x=variable, y=value)) +
  geom_boxplot(aes(fill=variable)) +
  xlab("Test Subjects") +
  ylab("Average Test Score") +
  labs(fill="Subject") +
  ggtitle("Boxplot for Reading, Writing, and Math Test Scores")
```

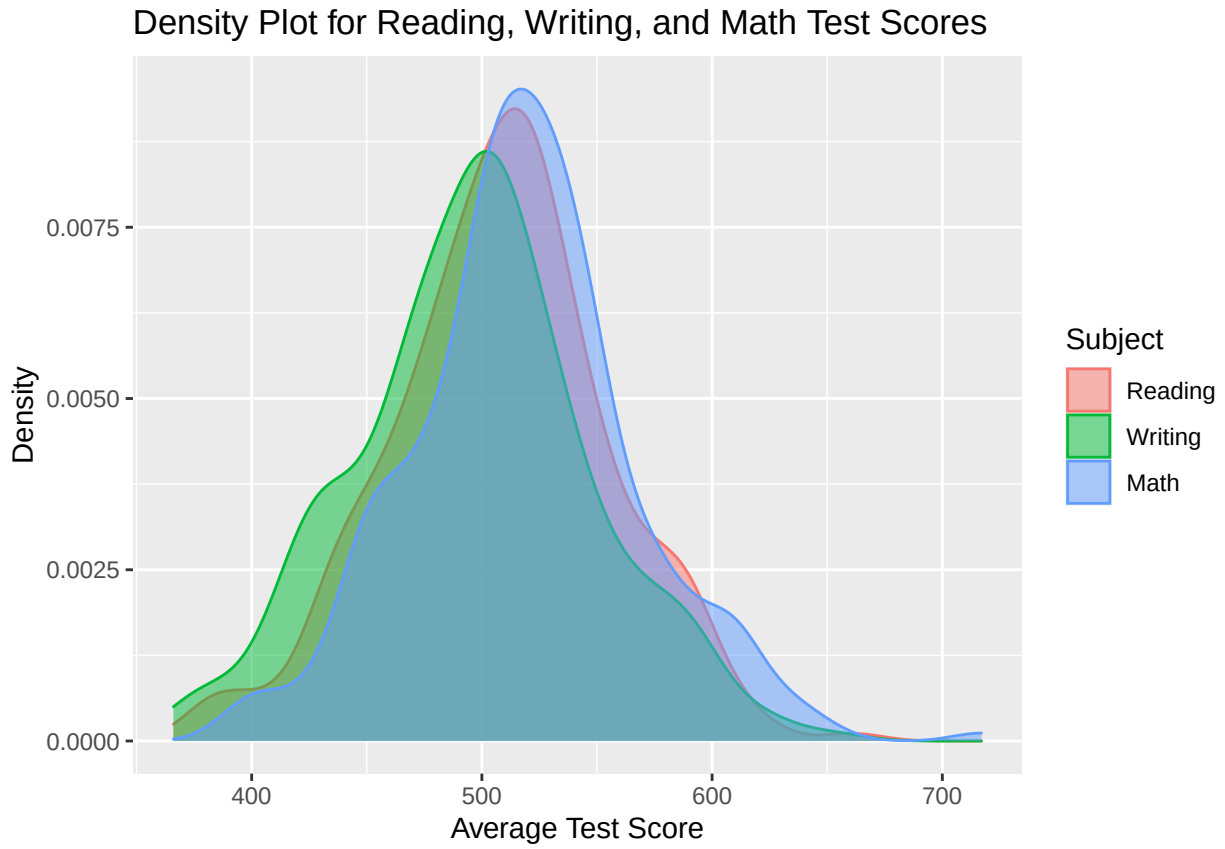
```
## No id variables; using all as measure variables
```




```
# Stack data by test subject for the density plot
mod_data <- stack(data[,4:6])
colnames(mod_data) <- c("Score", "Subject")
mod_data[sample(nrow(mod_data), 10), ]
```

```
##     Score Subject
## 767   560    Math
## 447   468 Writing
## 574   576 Writing
## 409   509 Writing
## 198   459 Reading
## 32    449 Reading
## 626   452    Math
## 608   504    Math
## 172   568 Reading
## 454   488 Writing
```

```r
# Density plot
ggplot(mod_data, aes(x=Score)) +
  geom_density(aes(group=Subject, colour=Subject, fill=Subject), alpha=0.5) +
  xlab("Average Test Score") +
  ylab("Density") +
  ggtitle("Density Plot for Reading, Writing, and Math Test Scores")
```



Question 3 (20 points)

Create a visualization that captures the relation between Avg. SAT scores (use data from question 2) and median household income in that school district. For the median household income of the school districts use http://www.usa.com/rank/massachusetts-state–median-household-income–school-district-rank.htm. Write your observations from the visualization.

So when I looked at the income dataset, the district names were not exact compared to the SAT dataset. The SAT dataset had the district codes, but the income dataset didn't, so I used Google to find all of the district codes for the income dataset. Once I got all of the codes, I put the matched incomes in the SAT dataset based off of the district codes using the match() function, where I matched the district codes from both datasets.

https://profiles.doe.mass.edu/search/search.aspx?leftNavId=11238

Go to the dropdown menu, select "Public School District", and click "Get Results".

Data Preparation

```
# Load dataset
income <- read_excel('median_income.xlsx')

# Change column names
colnames(income)[2] <- c("Income")
colnames(data)[2] <- c("District_Code")
head(income)
```

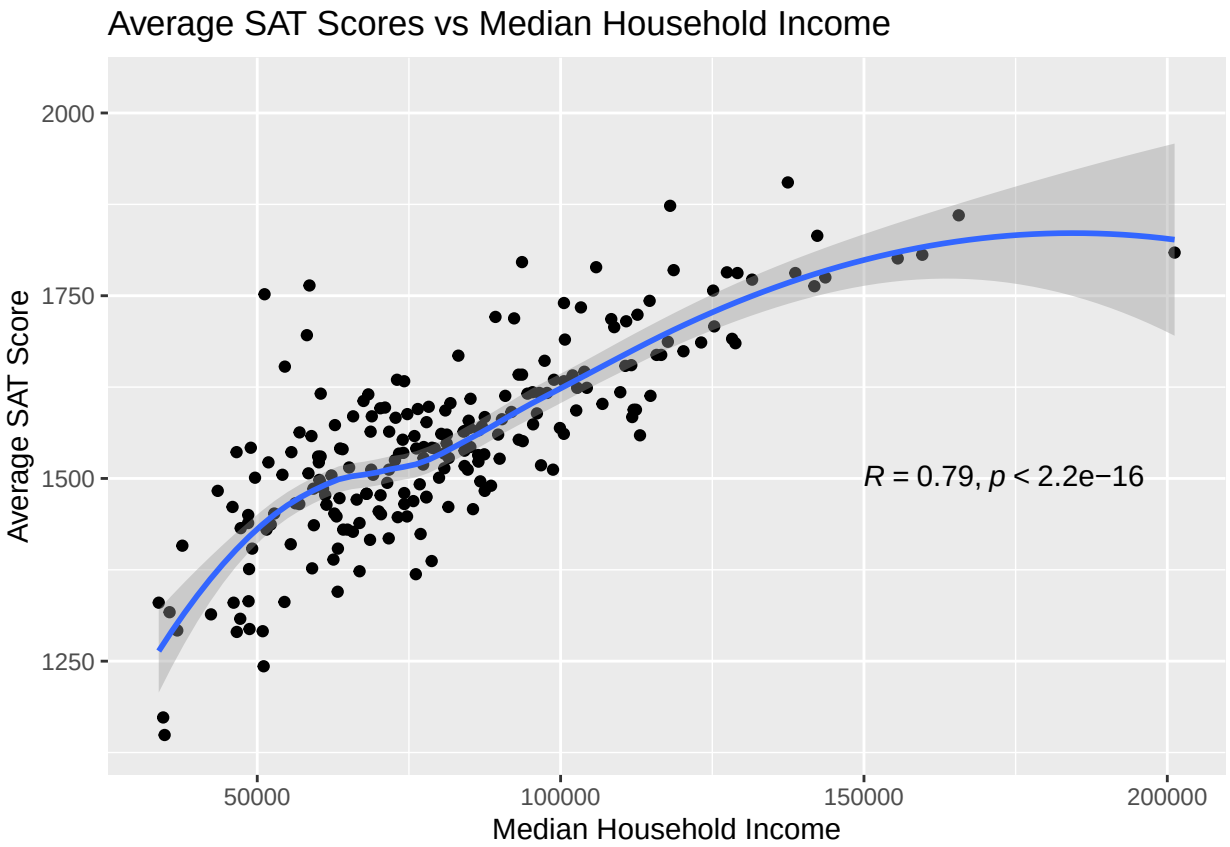
```
## # A tibble: 6 x 5
##    Rank Income `School District`                                Population Code
##   <dbl>  <dbl> <chr>                                                <dbl> <chr>
## 1   120  81500 Abington School District                             16081 000100~
## 2    28 120865 Acton School District                               22614 000200~
## 3    31 118054 Acton-Boxborough School District                    27716 060000~
## 4   187  69570 Acushnet School District                            10329 000300~
## 5   252  57222 Adams-Cheshire School District in Savoy (7-12~        703 <NA>
## 6   296  45081 Adams-Cheshire School District                      11593 <NA>
```

```
# Join the two tables and add an income column to the SAT dataset
data$Income <- income$Income[match(data$District_Code, income$Code)]

# Create a new overall score column
data$Overall <- data$Reading + data$Writing + data$Math
```

SAT Scores vs Income

```
# ggpubr library for stat_cor()
library(ggpubr)

# Plot
ggplot(data, aes(x=Income, y=Overall)) +
  geom_point() +
  xlab("Median Household Income") +
  ylab("Average SAT Score") +
  ggtitle("Average SAT Scores vs Median Household Income") +
  stat_cor(label.x = 150000, label.y = 1500) +
  geom_smooth(method='auto')
```

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



It looks like there is a relatively strong positive correlation, as shown by the R value, and by the scatter plot. However, it does start to flatten, which makes sense because the test scores have a maximum which is very hard to attain whereas income has no limit. So even as income continues to go up, the average scores will likely remain around a little under 2000 because an average around 2000 is very difficult to attain regardless of income for a school with a lot of students. The scores have a stronger correlation with IQ and intelligence rather than income.