Student Performance Project

Yebin Kim

2024-12-07

1 Introduction

This report is for the second project submission of "HarvardX PH125.9x:Data Science: Capstone" course, 'Choose Your Own' project. For this project, one of the data from Kaggle, which is called 'Students Performance' dataset(https://www.kaggle.com/datasets/adithyabshetty100/student-performance?select=StudentsPerformance.csv), will be used.

In this introduction section, I'm going to describe the dataset and variables, and summarize the goal of the project and key steps to be performed.

1.1 Dataset Overview

First of all, I used my github as a data storage for this project so I just uploaded the Kaggle data there and created a raw link of the data from my github. Therefore, it can be easy to access the dataset by automatically downloading the data with the code as follows:

```
# Download kaggle data from github
url <- "https://raw.githubusercontent.com/yebinkim86/students_performance/refs/heads/main/StudentsPerfo
data <- read.csv(url)
```

Loading the data, we can see there are 8 columns: gender, race/ethnicity, parental level of education, lunch, test preparation course, math score, reading score, writing score.

```
head(data)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group B           bachelor's degree     standard
## 2 female        group C                some college     standard
## 3 female        group B             master's degree     standard
## 4   male        group A          associate's degree free/reduced
## 5   male        group C                some college     standard
## 6 female        group B          associate's degree     standard
##   test.preparation.course math.score reading.score writing.score
## 1                    none         72            72            74
## 2               completed         69            90            88
## 3                    none         90            95            93
## 4                    none         47            57            44
## 5                    none         76            78            75
## 6                    none         71            83            78
```

Columns	Description
gender	Gender of the student (Female / Male)
race/ethnicity	Race of the student as group A to E
parental level of education	What is the education Qualification of Students Parent
lunch	Whether the lunch is Standard type/Free lunch or some discounted lunch
test preparation course	Whether Student has Taken or not and completed
math score	Scores in math
reading score	Scores in reading
writing score	Scores in writing

According to each column, 'gender', 'race/ethnicity', 'parental level of education', 'lunch', and 'test preparation course' are categorical variables. On the other hand, 'math score', 'reading score', and 'writing score' are the continuous variables.

Variables	Columns	Data
Categorical	gender	female, male
-	race/ethnicity	group A, group B, group C, group D, group E
-	parental level of education	some college, associate's degree, high school, some high school, bachelor's degree
-	lunch	standard, free/reduced
-	test preparation course	none, completed
Continuous	math score	0 to 100
-	reading score	0 to 100
-	writing score	0 to 100

1.2 The Goal of the project

The aim of this project is to apply machine learning techniques to the real world in the education field with these students' performance data. Through this project, I hope to gain insight from this analysis and think about better ways of better education, especially for the efficiency of the 'test preparation course' part.

1.3 Key steps of the project

In this project, I used two different models for prediction. At first, linear regression models were made to predict the average score by using various variables such as gender, race/ethnicity, lunch, and test preparation. Next, random forest models were also made to develop the linear regression model to predict the average score. I hope that this random forest model could be more advanced than the former linear regression for prediction.

Models	Prediction(Y~X)
Linear Regression	Gender, Race/ethnicity, Parental level of education, Lunch, Test preparation course ~ Average score
Random Forest	Important variables among above ~ Average score#

2 Data Analysis

In this section, I'm going to explain the process and techniques used, including data preparation for prediction, data exploration and visualization, any insights gained, and my modeling approach.

2.1 Data Preparation for Prediction

Before starting the machine learning, we have to install some packages needed in this project, and the data should be reformed to be easily analyzed. So, I just installed packages and renamed the column names to avoid errors that could be caused by whitespace. Plus, I transformed the data of gender, lunch, and test preparation course column into 1 or 0, and the categorical variables into the factors. After that, I created the 'average score' column by using the math, reading, and writing score.

```
#Load packages that could be needed in this project
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(caret)
```

```
##          : lattice
##
##         : 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(randomForest)
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
```

```
##          : 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
#Renaming column names
colnames(data)[2] <- "race_ethnicity"
colnames(data)[3] <- "parental_level_of_education"
colnames(data)[5] <- "test_preparation_course"
colnames(data)[6] <- "math_score"
colnames(data)[7] <- "reading_score"
colnames(data)[8] <- "writing_score"

#Data transformation
data$gender <- ifelse(data$gender == "male", 1, 0)
data$gender <- as.factor(data$gender)
data$lunch <- ifelse(data$lunch == "standard", 1, 0)
data$lunch <- as.factor(data$lunch)
data$test_preparation_course <- ifelse(data$test_preparation_course == "completed", 1, 0)
data$test_preparation_course <- as.factor(data$test_preparation_course)
data$race_ethnicity <- as.factor(data$race_ethnicity)
data$parental_level_of_education <- as.factor(data$parental_level_of_education)

#Create average score in the data set
data <- data %>% mutate(average_score=(math_score+reading_score+writing_score)/3)
```

```
head(data)
```

```
##   gender race_ethnicity parental_level_of_education lunch
## 1      0        group B           bachelor's degree     1
## 2      0        group C                some college     1
## 3      0        group B             master's degree     1
## 4      1        group A          associate's degree     0
## 5      1        group C                some college     1
## 6      0        group B          associate's degree     1
##   test_preparation_course math_score reading_score writing_score average_score
## 1                       0         72            72            74      72.66667
## 2                       1         69            90            88      82.33333
## 3                       0         90            95            93      92.66667
## 4                       0         47            57            44      49.33333
## 5                       0         76            78            75      76.33333
## 6                       0         71            83            78      77.33333
```

2.2 Data Exploration & Visualization

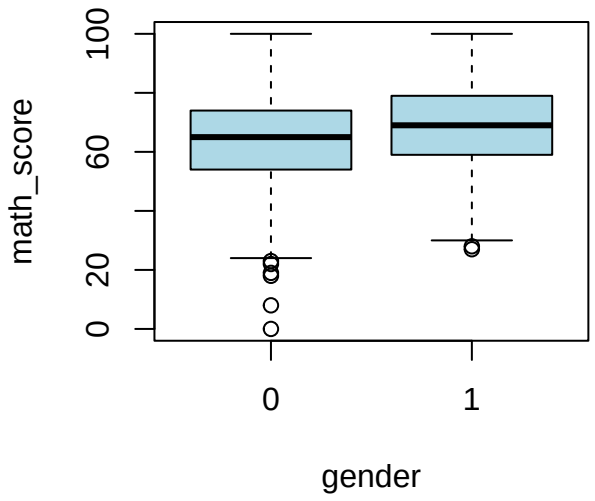
To know more about this data, I made several plots of each variable. There are two parts of data exploration: distribution of each score by other factors, and statistical relationship between each score. Let's start with

the first one. To know more about this data, I made several plots of each variable. There are two parts of data exploration: distribution of each score by other factors, and statistical relationship between each score. Let's start with the first one.

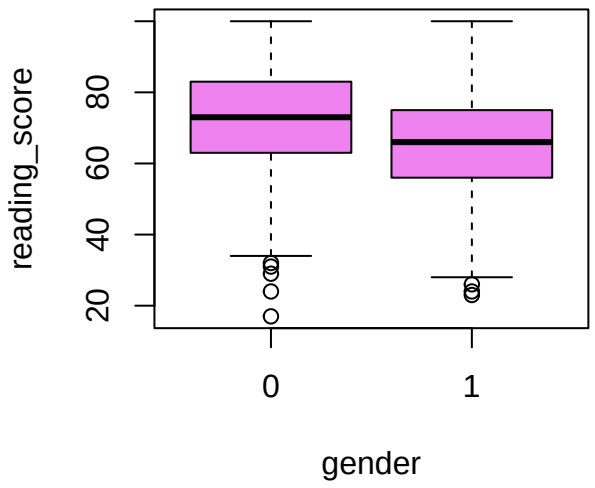
2.2.1 Distribution of Average score by each factor

(1) Distribution of each score by Gender In the box plot of this section, the number of the x-axis stands for the gender (0=female, 1=male) and y-axis means the score. As we see, as for the average score, it seems that females have a bit higher score than males. However, if we see the distribution of math scores, males are better than females, unlike in the other plots. In other words, males do better in math and females perform better in reading and writing relatively.

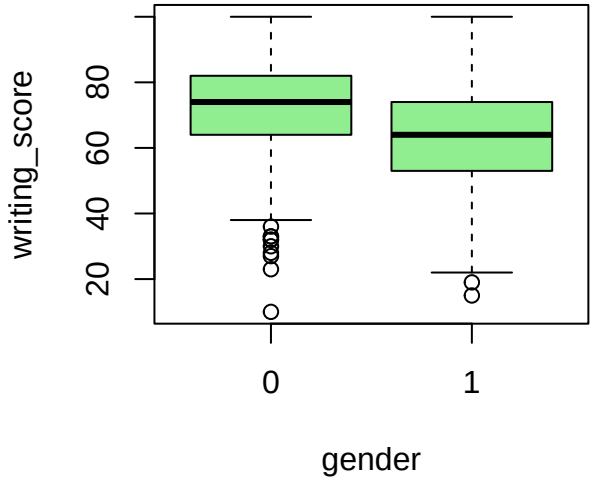
```
#Data summarize of each score
##by gender
boxplot(math_score ~ gender, data=data, col="lightblue")
```



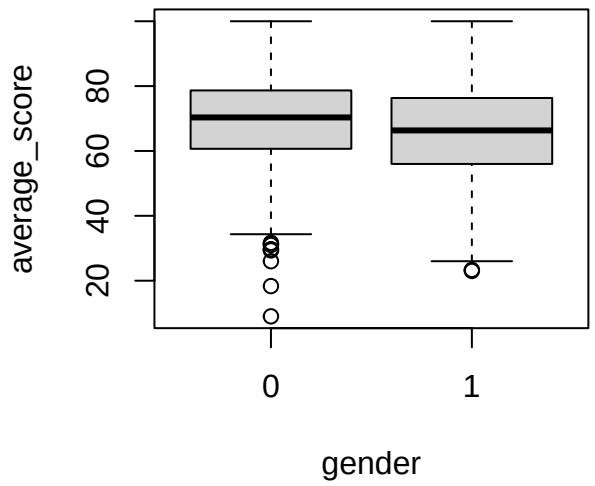

```
boxplot(reading_score ~ gender, data=data, col="violet")
```




```
boxplot(writing_score ~ gender, data=data, col="lightgreen")
```



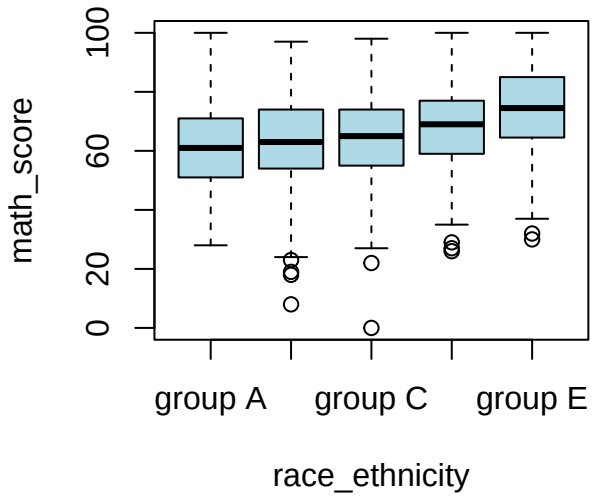

```
boxplot(average_score ~ gender, data=data)
```



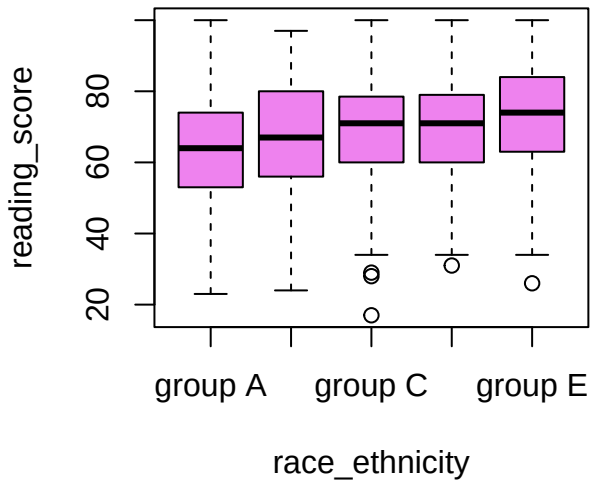

```
by_gender <- data %>% group_by(gender) %>% summarize(average_score=mean(average_score))
```

(2) Distribution of each score by Race/Ethnicity In these plots, we can see that the scores gradually increase from group A to group E. I can guess that the members of group E have higher scores than other groups in each part..

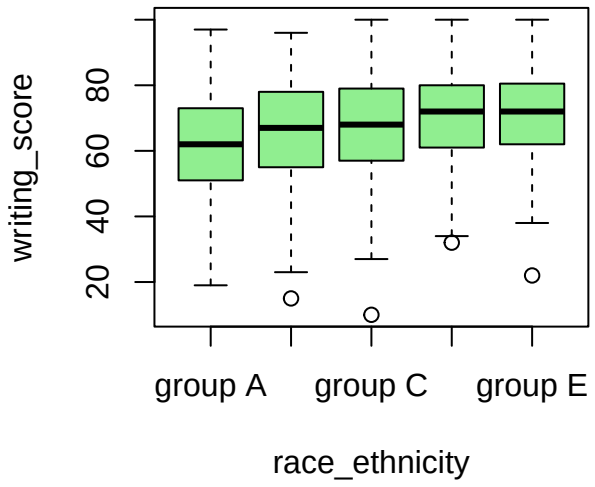
```
##by race/ethnicity
boxplot(math_score ~ race_ethnicity, data=data, col="lightblue")
```



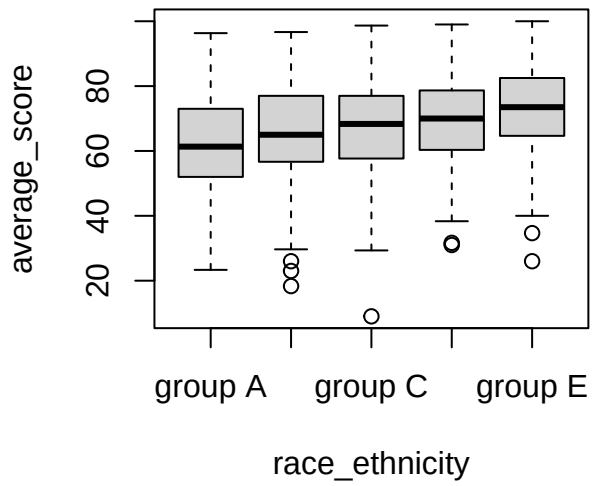
```r
boxplot(reading_score ~ race_ethnicity, data=data, col="violet")
```




```r
boxplot(writing_score ~ race_ethnicity, data=data, col="lightgreen")
```



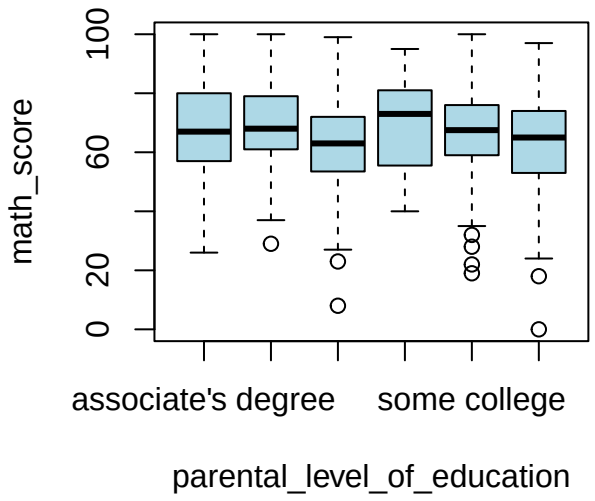

```r
boxplot(average_score ~ race_ethnicity, data=data)
```



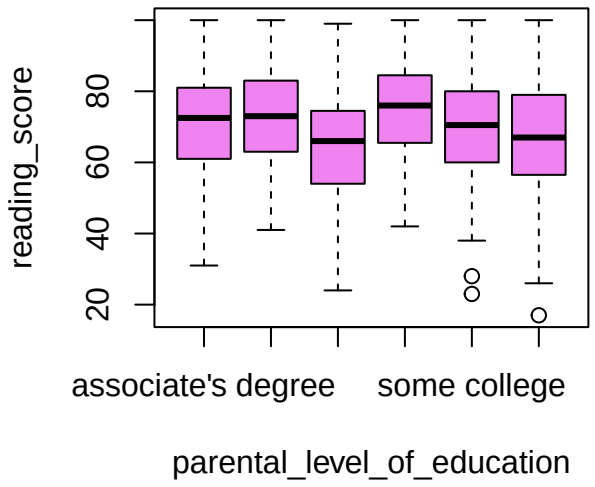

```
by_race_ethnicity <- data %>% group_by(race_ethnicity) %>% summarize(average_score=mean(average_score))
```

(3) Distribution of each score by Parental Level of Education According to these plots, I could see the apparent difference of scores between 'high school' and 'master's degree'. Even though we can never be sure to say that the higher parental level of education means the higher score they might get, this data tells us the one with the higher parental level of education mostly performs better than the lower one.

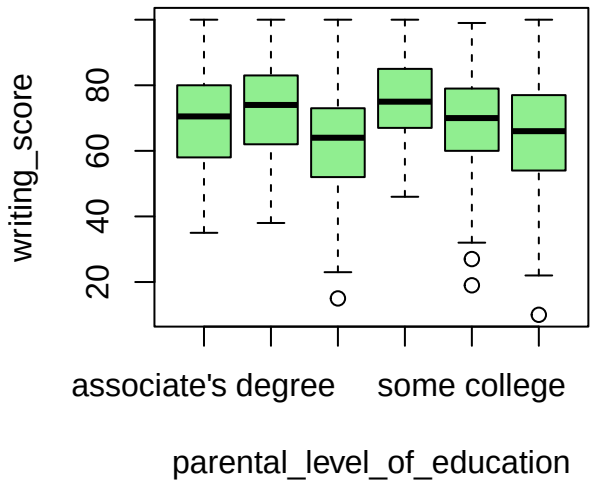
```
##by parental level of education
boxplot(math_score ~ parental_level_of_education, data=data, col="lightblue")
```



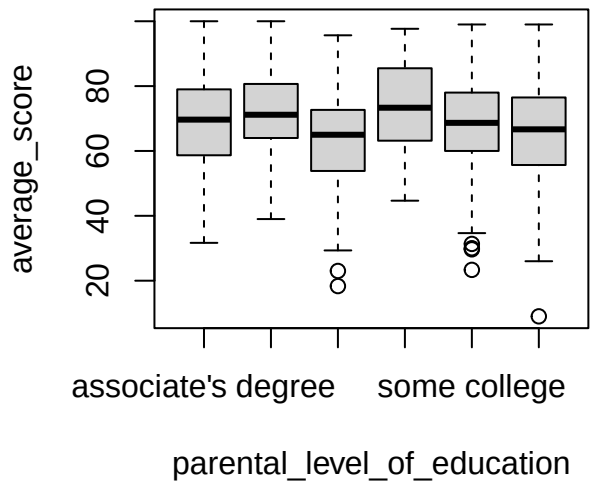
```
boxplot(reading_score ~ parental_level_of_education, data=data, col="violet")
```




```
boxplot(writing_score ~ parental_level_of_education, data=data, col="lightgreen")
```



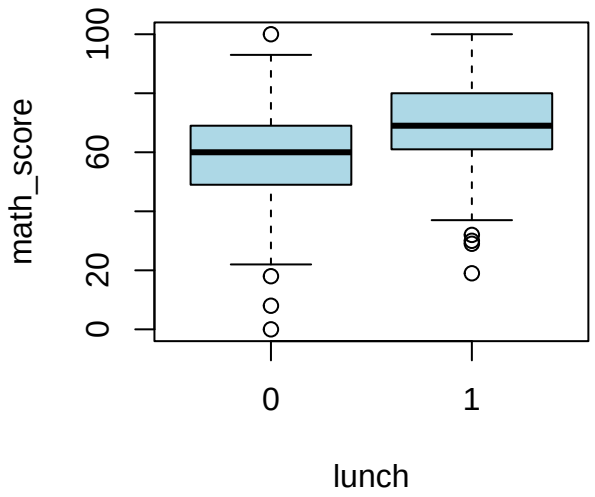

```
boxplot(average_score ~ parental_level_of_education, data=data)
```



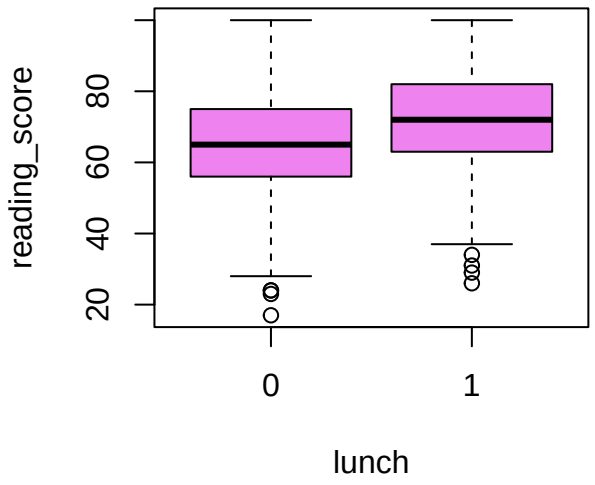

```
by_parental_level_of_education <- data %>% group_by(parental_level_of_education) %>% summarize(average_s
```

(4) Distribution of each score by Lunch These four plots on the right side and below definitely show the difference between the type of lunch (0=free/reduced, 1=standard). The students who didn't have standard lunch mostly have lower scores than the others. So, we can guess that there is a relationship between the type of lunch they eat and the score they have.

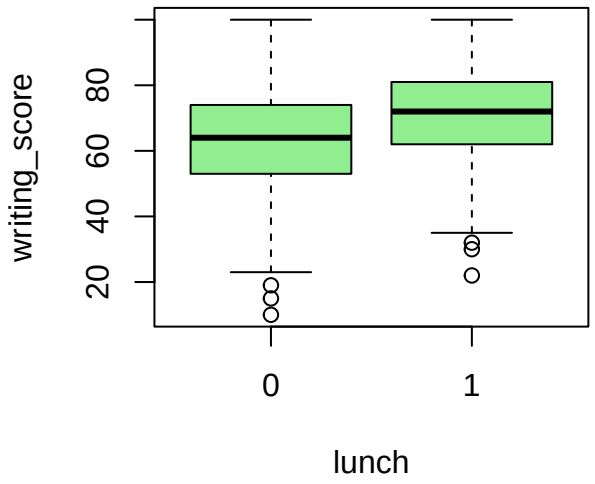
```
##by lunch
boxplot(math_score ~ lunch, data=data, col="lightblue")
```



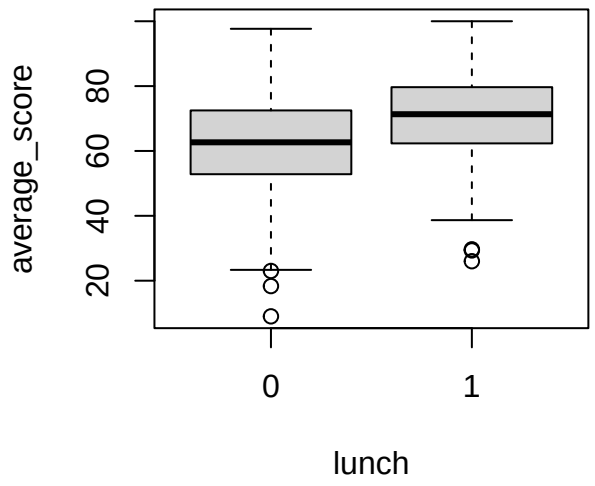
```
boxplot(reading_score ~ lunch, data=data, col="violet")
```




```
boxplot(writing_score ~ lunch, data=data, col="lightgreen")
```



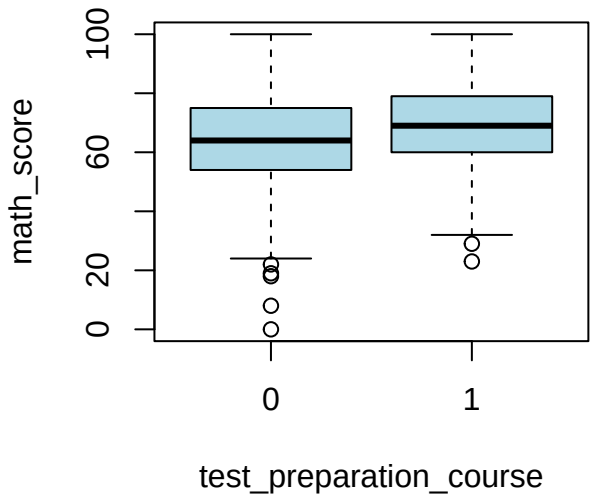

```
boxplot(average_score ~ lunch, data=data)
```



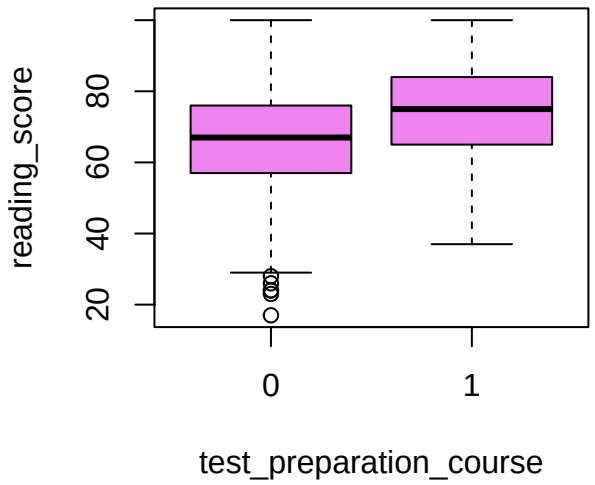

```
by_lunch <- data %>% group_by(lunch) %>% summarize(average_score=mean(average_score))
```

(5) Distribution of each score by Test Preparation Course Through these box plots, we can see the students who had the test preparation course perform better than the ones who didn't have the course. It seems that the course is more effectual for reading and writing. However, if we think about the meaning of 'test preparation course', this could be a kind of reasonable result, because students can have more opportunities to study and prepare for the test through the course.

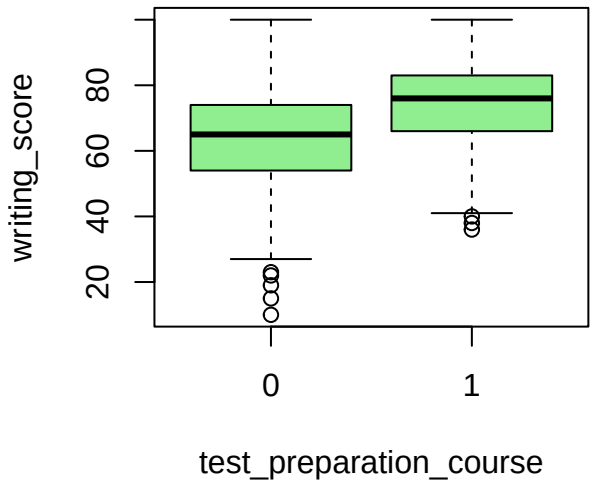
```
##by test preparation course
boxplot(math_score ~ test_preparation_course, data=data, col="lightblue")
```



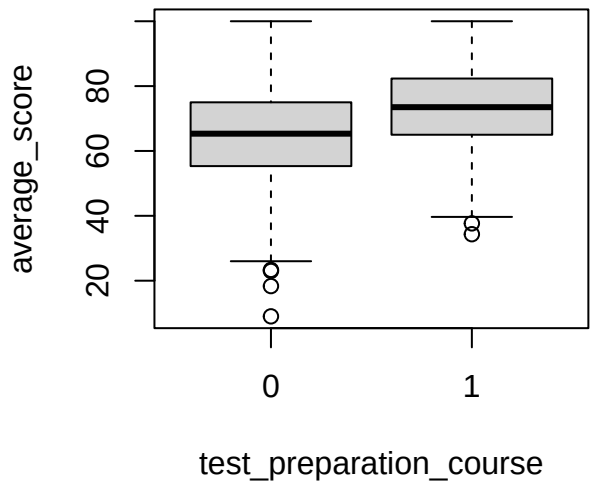
```
boxplot(reading_score ~ test_preparation_course, data=data, col="violet")
```




```
boxplot(writing_score ~ test_preparation_course, data=data, col="lightgreen")
```




```
boxplot(average_score ~ test_preparation_course, data=data)
```




```
by_test_preparation_course<- data %>% group_by(test_preparation_course) %>% summarize(average_score=mea
```

To sum up, as the variables of the x-axis change, there are some differences in the y-axis: average/math/reading/writing score. Then, among these variables, what are the significant ones that have an effect on the average score? So, I ran the code to calculate the difference between the max and min of average score for each section. The results are as follows:

Gender	Race/Ethnicity	Parental level of education	Lunch	Test preparation course
3.732015	9.759872	10.501931	8.638148	7.630519

It is clear that 'Race/Ethnicity', 'Parental level of education', and 'Lunch' are the TOP 3 significant factors that has effect on the

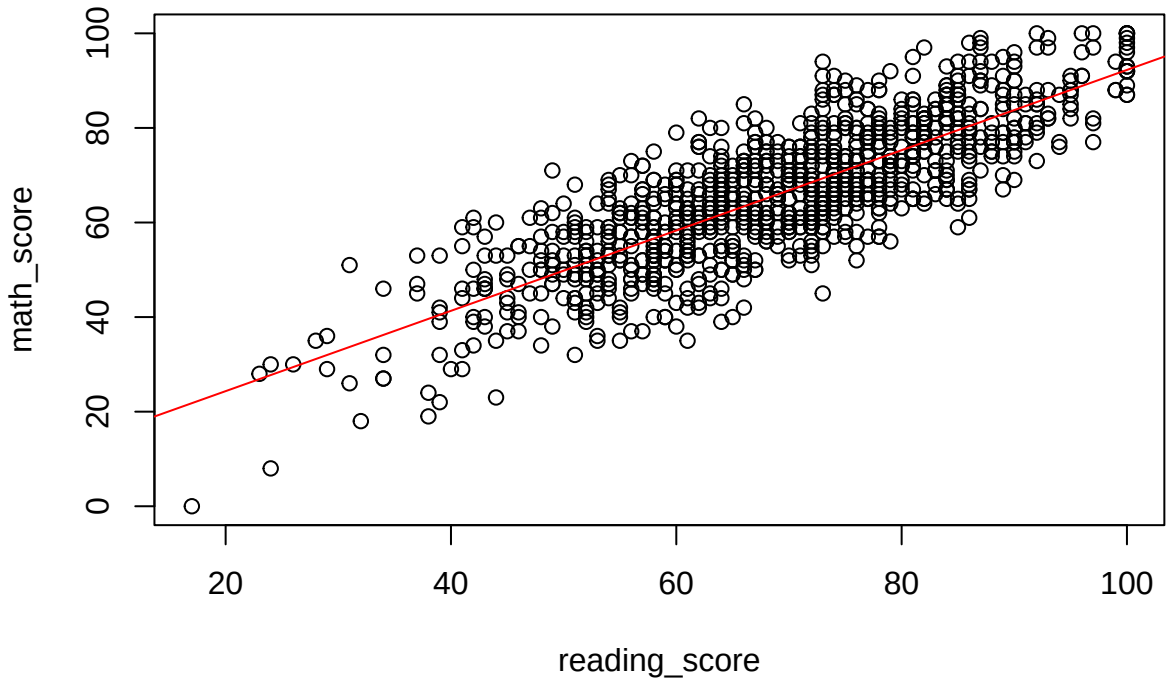
2.2.2 Statistical relationship between each score Plus, I was also curious about the statistical relationship between each score, such as math & reading score, math & writing score, and reading & writing score. This time I'll totally ignore other variables, and just focus on the linear model of each score. Let's see how it works.

(1) Math score & Reading score

```
  #Relationship between each score
##math & reading
plot(math_score~reading_score, data=data)
cor(data$math_score, data$reading_score)
```

```
## [1] 0.8175797
```

```
abline(lm(math_score~reading_score, data=data),col="red")
```

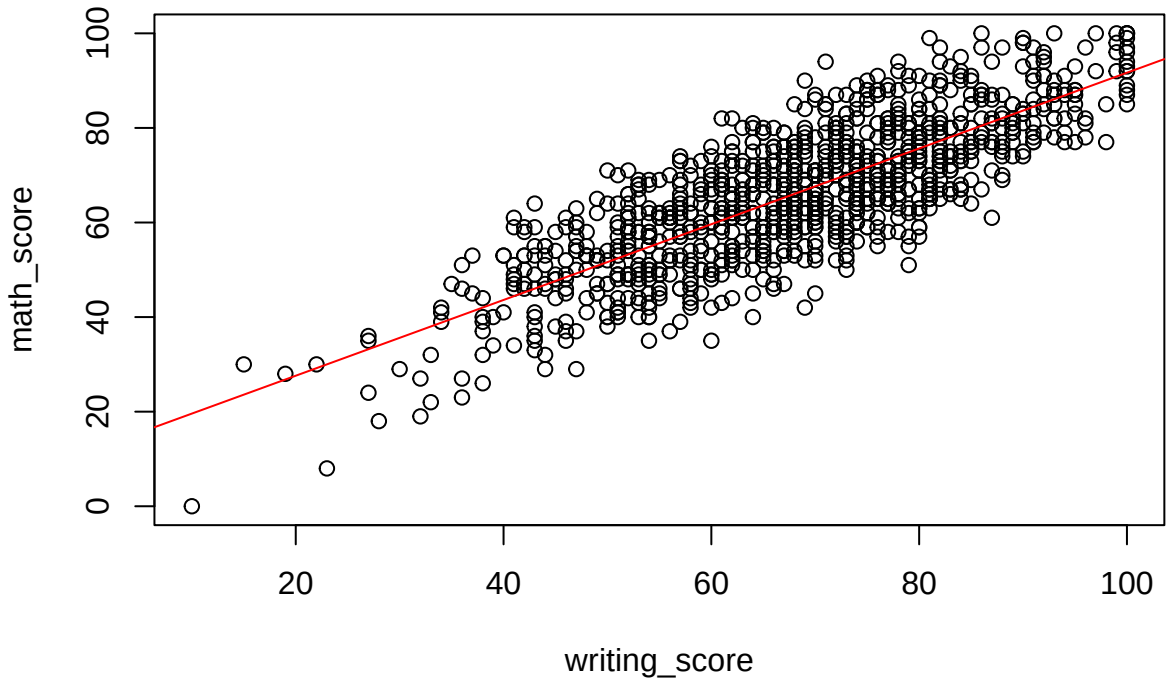


(2) Math score & Writing score

```
##math & writing
plot(math_score~writing_score, data=data)
cor(data$math_score, data$writing_score)
```

```
## [1] 0.802642
```

```
abline(lm(math_score~writing_score, data=data),col="red")
```

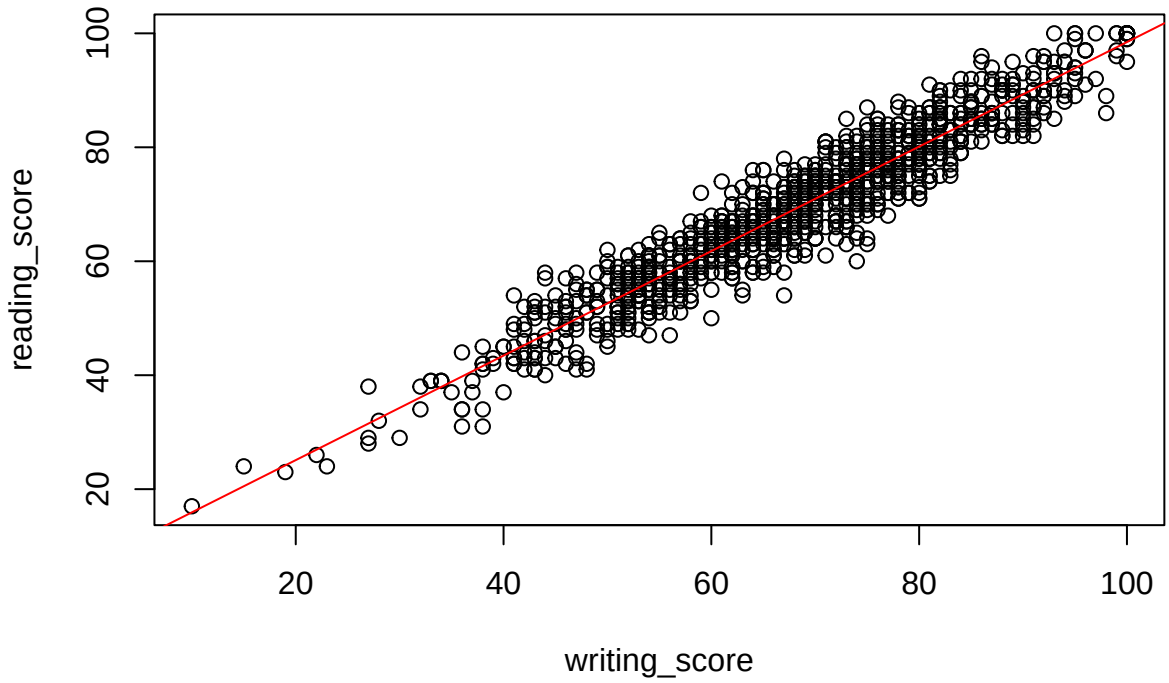


(3) Reading score & Writing score

```
##reading & writing
plot(reading_score~writing_score, data=data)
cor(data$reading_score, data$writing_score)
```

```
## [1] 0.9545981
```

```
abline(lm(reading_score~writing_score, data=data),col="red")
```



Through three scatter plots above, we can see the correlation coefficient between each score. Math & Reading, Math & Writing, and Reading & Writing all have quite meaningful values. Because all the absolute value of correlation coefficient is close to 1, the student who is good at one subject could be good at another subject. Especially for Reading & Writing, its correlation coefficient is shown as 0.9545981. So, the scores of literacy are more linked closely to each other than to math scores. These results are somewhat reasonable.

2.3 Prediction 1: Using Linear Regression Model Now, let's move on to the prediction. First, I'm going to focus on the prediction of average score so that I'll use the linear regression model to predict it by using categorical variables. Before starting the prediction, we have to split the data set into train and test set. This time 20% of data will be test set and the others will be training set.

Train set (80%)	Test set (20%)

Plus, we're going to use RMSE to find out the performance of the model. So, I just define the RMSE function to make the process of calculation simple.

```
#Split the data set into training and test set
##Create train set and test set
set.seed(1)
test_index<-createDataPartition(y=data$average_score, times=1, p=0.2, list=FALSE)
train_set<-data[-test_index,]
```

```r
test_set<-data[test_index,]

#The RMSE function that can be used in this project
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

And then, starting from the standard deviation of average score, I used each variable to make prediction model as follows:

```r
#Standard deviation of average score in test set
summary(test_set$average_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   58.33   68.33   67.91   77.33  100.00
```

```r
just_sd <- sd(test_set$average_score)
just_sd
```

```
## [1] 14.52332
```

```r
##Gender model
g_model <- lm(average_score~gender, data=train_set)
summary(g_model)
```

```
##
## Call:
## lm(formula = average_score ~ gender, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.871  -9.522   0.827   9.796  33.858
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  69.2040     0.6933   99.81  < 2e-16 ***
## gender1      -3.0622     1.0008   -3.06  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.12 on 796 degrees of freedom
## Multiple R-squared:  0.01162,    Adjusted R-squared:  0.01038
## F-statistic: 9.362 on 1 and 796 DF,  p-value: 0.00229
```

```r
predicted_score <- predict(g_model, newdata = test_set)
g_rmse <- RMSE(test_set$average_score, predicted_score)

##race model
r_model <- lm(average_score~race_ethnicity, data=train_set)
summary(r_model)
```

```
##
## Call:
## lm(formula = average_score ~ race_ethnicity, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.873  -9.491   0.430   9.918  32.585
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               63.749      1.689  37.747  < 2e-16 ***
## race_ethnicitygroup B      1.458      2.030   0.718 0.472984
## race_ethnicitygroup C      3.508      1.899   1.847 0.065085 .
## race_ethnicitygroup D      5.626      1.955   2.878 0.004114 **
## race_ethnicitygroup E      8.154      2.154   3.785 0.000165 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.03 on 793 degrees of freedom
## Multiple R-squared:  0.02865,    Adjusted R-squared:  0.02375
## F-statistic: 5.848 on 4 and 793 DF,  p-value: 0.0001219
```

```
predicted_score <- predict(r_model, newdata = test_set)
r_rmse <- RMSE(test_set$average_score, predicted_score)

##parental model
p_model <- lm(average_score~parental_level_of_education, data=train_set)
summary(p_model)
```

```
##
## Call:
## lm(formula = average_score ~ parental_level_of_education, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.936  -9.270   0.939  10.159  33.275
##
## Coefficients:
##                                             Estimate Std. Error t value
## (Intercept)                                68.400000   1.022174  66.916
## parental_level_of_educationbachelor's degree  4.441085   1.814514   2.448
## parental_level_of_educationhigh school       -5.130263   1.522012  -3.371
## parental_level_of_educationmaster's degree    4.763522   2.166085   2.199
## parental_level_of_educationsome college      -0.002996   1.459716  -0.002
## parental_level_of_educationsome high school  -2.675463   1.545047  -1.732
##                                             Pr(>|t|)
## (Intercept)                                  < 2e-16 ***
## parental_level_of_educationbachelor's degree 0.014600 *
## parental_level_of_educationhigh school       0.000786 ***
## parental_level_of_educationmaster's degree   0.028156 *
## parental_level_of_educationsome college      0.998363
## parental_level_of_educationsome high school  0.083727 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 13.9 on 792 degrees of freedom
## Multiple R-squared:  0.04716,    Adjusted R-squared:  0.04114
## F-statistic: 7.839 on 5 and 792 DF,  p-value: 3.239e-07
```

```
predicted_score <- predict(p_model, newdata = test_set)
p_rmse <- RMSE(test_set$average_score, predicted_score)

##Lunch model
l_model <- lm(average_score~lunch, data=train_set)
summary(l_model)
```

```
##
## Call:
## lm(formula = average_score ~ lunch, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.723  -8.847   0.610   9.277  35.112
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  62.5548     0.7988  78.311  < 2e-16 ***
## lunch1        8.1685     1.0032   8.143 1.49e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.65 on 796 degrees of freedom
## Multiple R-squared:  0.07689,    Adjusted R-squared:  0.07573
## F-statistic: 66.31 on 1 and 796 DF,  p-value: 1.487e-15
```

```
predicted_score <- predict(l_model, newdata = test_set)
l_rmse <- RMSE(test_set$average_score, predicted_score)

##test preparation model
t_model <- lm(average_score~test_preparation_course, data=train_set)
summary(t_model)
```

```
##
## Call:
## lm(formula = average_score ~ test_preparation_course, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.531  -8.864   0.802   9.544  35.136
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                64.8644     0.6066 106.929  < 2e-16 ***
## test_preparation_course1   7.9245     1.0080   7.862 1.23e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 13.69 on 796 degrees of freedom
## Multiple R-squared:  0.07205,    Adjusted R-squared:  0.07088
## F-statistic:  61.8 on 1 and 796 DF,  p-value: 1.232e-14
```

```
predicted_score <- predict(t_model, newdata = test_set)
t_rmse <- RMSE(test_set$average_score, predicted_score)

print(c(g_rmse, r_rmse, p_rmse, l_rmse, t_rmse))
```

```
## [1] 14.23034 14.07572 14.08020 13.66565 14.18057
```

Prediction Model	RMSE
Gender Model	14.2303438
Race Model	14.0757161
Parental Model	14.0802027
Lunch Model	13.6656539
Test preparation Model	14.1805737

All of the RMSEs are lower than the standard deviation of test set (14.5233198), but each of them has different values. So next, I combined all models to make a linear regression prediction model as follows:

```
##total model
model <- lm(average_score~gender+race_ethnicity+parental_level_of_education+lunch+test_preparation_cours
summary(model)
```

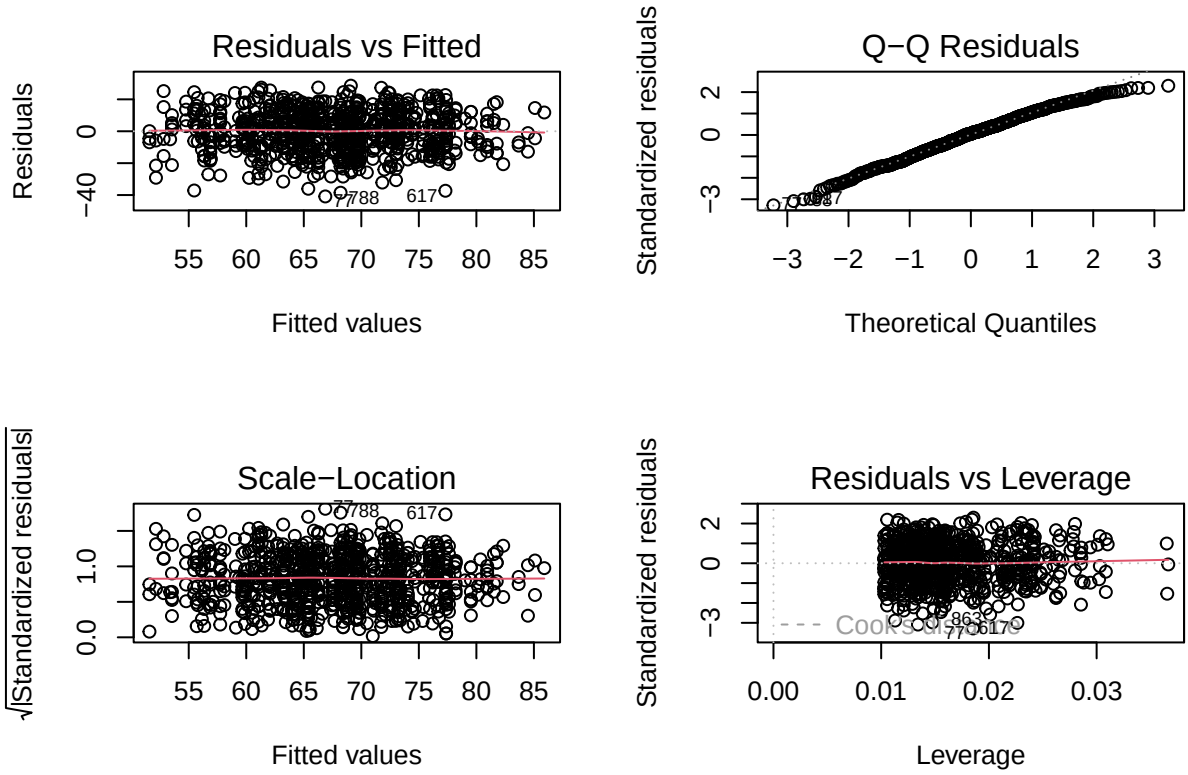
```
##
## Call:
## lm(formula = average_score ~ gender + race_ethnicity + parental_level_of_education +
##     lunch + test_preparation_course, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.870  -8.361   0.734   8.840  28.577
##
## Coefficients:
##                                              Estimate Std. Error t value
## (Intercept)                                 59.103622   1.943843  30.406
## gender1                                     -3.323717   0.897821  -3.702
## race_ethnicitygroup B                        0.565560   1.830149   0.309
## race_ethnicitygroup C                        1.955759   1.715615   1.140
## race_ethnicitygroup D                        4.106915   1.762341   2.330
## race_ethnicitygroup E                        5.525773   1.945979   2.840
## parental_level_of_educationbachelor's degree 4.155674   1.645183   2.526
## parental_level_of_educationhigh school      -4.193068   1.381318  -3.036
## parental_level_of_educationmaster's degree   4.961482   1.974485   2.513
## parental_level_of_educationsome college      0.001389   1.322412   0.001
## parental_level_of_educationsome high school -2.962083   1.406588  -2.106
## lunch1                                       8.526220   0.927591   9.192
## test_preparation_course1                     7.784307   0.934391   8.331
##                                             Pr(>|t|)
```

```
## (Intercept)                                    < 2e-16 ***
## gender1                                       0.000229 ***
## race_ethnicitygroup B                         0.757385
## race_ethnicitygroup C                         0.254644
## race_ethnicitygroup D                         0.020039 *
## race_ethnicitygroup E                         0.004634 **
## parental_level_of_educationbachelor's degree 0.011734 *
## parental_level_of_educationhigh school       0.002480 **
## parental_level_of_educationmaster's degree   0.012177 *
## parental_level_of_educationsome college      0.999162
## parental_level_of_educationsome high school  0.035533 *
## lunch1                                        < 2e-16 ***
## test_preparation_course1                     3.56e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.57 on 785 degrees of freedom
## Multiple R-squared:  0.2283, Adjusted R-squared:  0.2165
## F-statistic: 19.36 on 12 and 785 DF,  p-value: < 2.2e-16
```

```r
predicted_score <- predict(model, newdata = test_set)
rmse <- RMSE(test_set$average_score, predicted_score)
print(rmse)
```

```
## [1] 12.28199
```

```r
par(mfrow=c(2,2))
plot(model)
```




```
par(mfrow=c(1,1))
```

Using three variables (Parental level of education, Lunch, and Test preparation course), RMSE finally became much lower. And the four plots above also show that this model is made quite successfully.

2.4 Prediction 2: Using Random Forest

Though I made the prediction model in the former section, I was wondering and wanted to develop my model that has lower RMSE. So, I started to run the random forest to make it possible. I thought that just using random forest won't be that much helpful, so first I tried to find the best tuned mtry as follows:

```
#Prediction by Random forest
##Find the best tuned mtry of model
tuneGrid <- expand.grid(mtry = c(1, 2, 3, 4, 5, 6))
control <- trainControl(method = "cv", number = 5)
model_rf <- train(average_score ~ gender+race_ethnicity+parental_level_of_education+lunch+test_preparati
                  data = train_set,
                  method = "rf",
                  trControl = control,
                  tuneGrid = tuneGrid)

print(model_rf$bestTune)
```

```
##   mtry
## 2    2
```

I found the best tuned mtry, and applied this value to the random forest model. And I did several trials to make RMSE lower.

```
##modeling
model_rf <- randomForest(average_score ~ gender+race_ethnicity+parental_level_of_education+lunch+test_p
                         data = train_set,
                         ntree = 500,
                         mtry=2,
                         nodesize=5)
predicted_score_rf <- predict(model_rf, newdata = test_set)
rmse_rf <- RMSE(test_set$average_score, predicted_score_rf)
print(rmse_rf)
```

```
## [1] 12.98992
```

Eventually, the random forest model is made, and the RMSE of it is 12.9899208.

3 Results

Through this 'Students Performance Project', I made several prediction models, and these are the results below.

Prediction Model	RMSE
Gender Model	14.2303438
Race Model	14.0757161
Parental Model	14.0802027
Lunch Model	13.6656539
Test preparation Model	14.1805737
Gender + Race + Parental + Lunch + Test preparation Model (Linear Regression)	12.2819899
Gender + Race + Parental + Lunch + Test preparation Model (Random Forest)	12.9899208

As we see, the prediction model with multiple variables performs better than others with single variable.

4 Conclusion

In this conclusion section, I'm going to talk about the brief summary of the report, its potential impact, its limitations, and future work. Through this project, I found out the relationship between each categorical variable and the average score of students, and correlation of each score. And especially, I made the final prediction model which predicts the average score by using race/ethnicity, lunch, and test preparation course variables. The final RMSE is 12.2819899 for linear regression model, and 12.9899208 for random forest. And it's even lower than 'just_sd', `just_sd`. Therefore, even though the final RMSE is not that much precise for prediction model, it could perform quite nice when we compare this with the standard deviation. However, in order to make this prediction model perform better and do more exact prediction, it might be good to use other important variables or other better models. Or it can be helpful to set the range of errors in prediction when we use this model in real situation.

5 References

https://www.kaggle.com/datasets/adithyabshetty100/student-performance

https://github.com/yebinkim86/students_performance

Statistics and Data Science for Teachers by Anna Bargagliotti , Christine Franklin(2021)