Analysis of Student Data for BTRY 6020 Final Project

Samantha Davies

2025-05-15

Contents

1 Introduction

Students have varying levels of performance in school, and it is known that performance depends on a number of factors. Student demographics, such as race and ethnicity or family income are likely to have an influence on student performance in school and on exams. A model that can show this would improve equality within school systems because the level of additional help students who are disadvantaged at a baseline level could be identified and school systems worldwide could improve average scores on exams while also providing a better education to historically disadvantaged groups of students.

In this analysis, I used provided demographic variables as predictors for the student's math test score, which was also provided.

2 Setup

```r
# Install all necessary libraries for the analysis
library(ggplot2)
library(dplyr)
library(tidyr)
library(leaps)
library(car)
library(lmtest)

# Read in data, downloaded from Kaggle (see references for link)
data = read.csv("StudentsPerformance.csv")
```

3 Exploratory Data Analysis

3.1 Summary statistics of variables

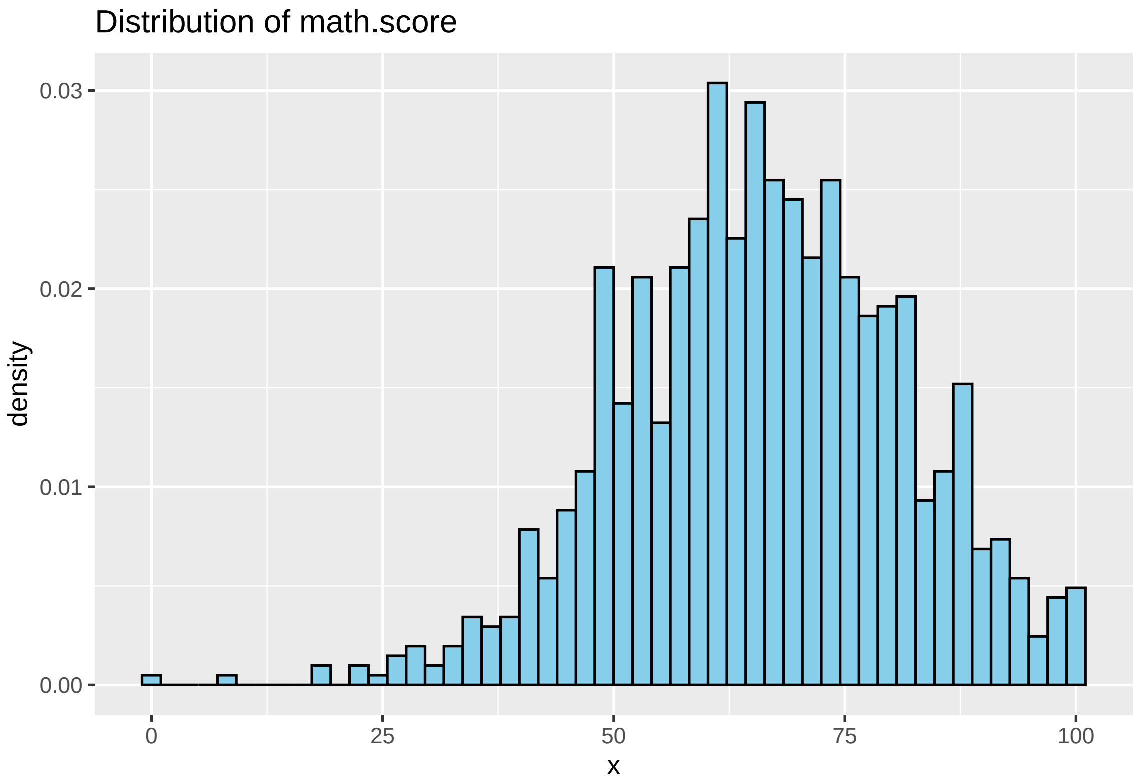
```r
summary = summary(data)

# Table of summary statistics
summary_table = data.frame(
  Min = summary[1,],
  `1st Qu.` = summary[2,],
  Median = summary[3,],
  Mean = summary[4,],
  `3rd Qu.` = summary[5,],
  Max = summary[6,]
)
# These are the variables in the dataset I am using as well as the summary
# statistics for each column:
summary_table
```

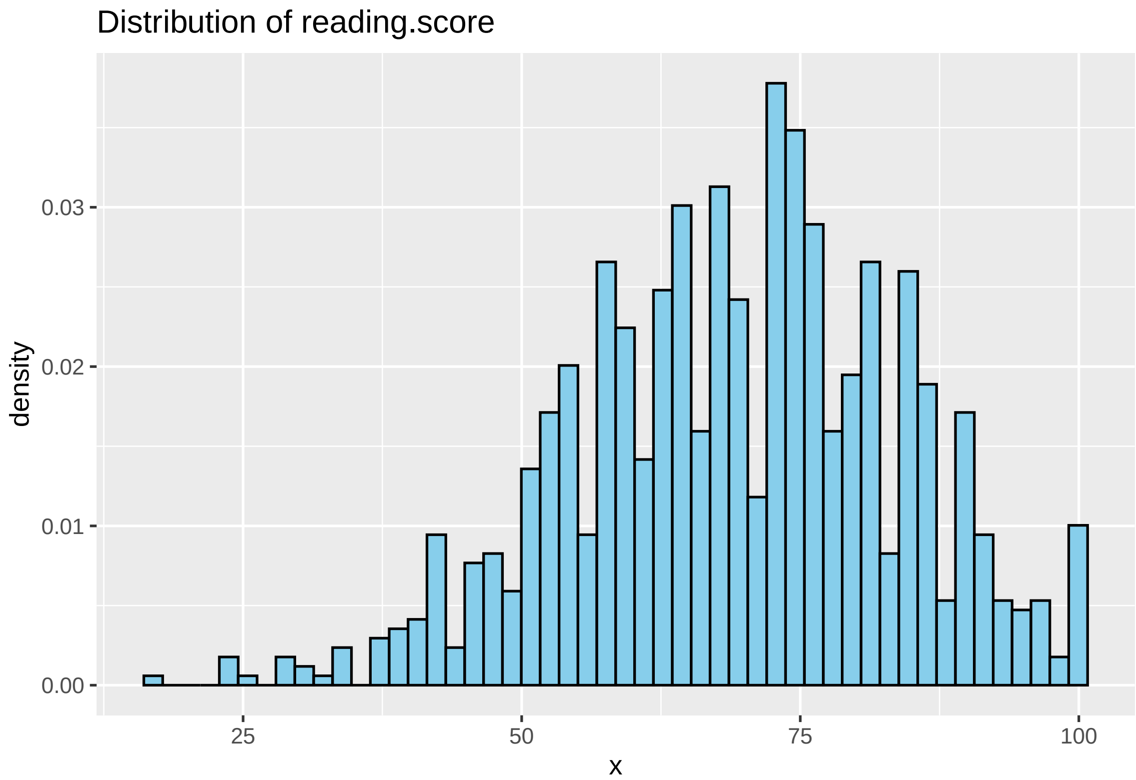
```
##                                           Min            X1st.Qu.
##    gender                    Length:1000        Class :character
## race.ethnicity               Length:1000        Class :character
## parental.level.of.education Length:1000        Class :character
##    lunch                     Length:1000        Class :character
## test.preparation.course      Length:1000        Class :character
##   math.score                   Min.   :  0.00     1st Qu.: 57.00
## reading.score                  Min.   : 17.00     1st Qu.: 59.00
## writing.score                  Min.   : 10.00     1st Qu.: 57.75
##                                        Median                   Mean
##    gender                    Mode  :character                   <NA>
## race.ethnicity               Mode  :character                   <NA>
## parental.level.of.education Mode  :character                   <NA>
##    lunch                     Mode  :character                   <NA>
## test.preparation.course      Mode  :character                   <NA>
##   math.score                   Median : 66.00   Mean   : 66.09
## reading.score                  Median : 70.00   Mean   : 69.17
## writing.score                  Median : 69.00   Mean   : 68.05
##                                    X3rd.Qu.                   Max
##    gender                                <NA>                <NA>
## race.ethnicity                           <NA>                <NA>
## parental.level.of.education             <NA>                <NA>
##    lunch                                 <NA>                <NA>
## test.preparation.course                  <NA>                <NA>
##   math.score                 3rd Qu.: 77.00   Max.   :100.00
## reading.score                3rd Qu.: 79.00   Max.   :100.00
## writing.score                3rd Qu.: 79.00   Max.   :100.00
```

4 Visualizations of distributions and relationships

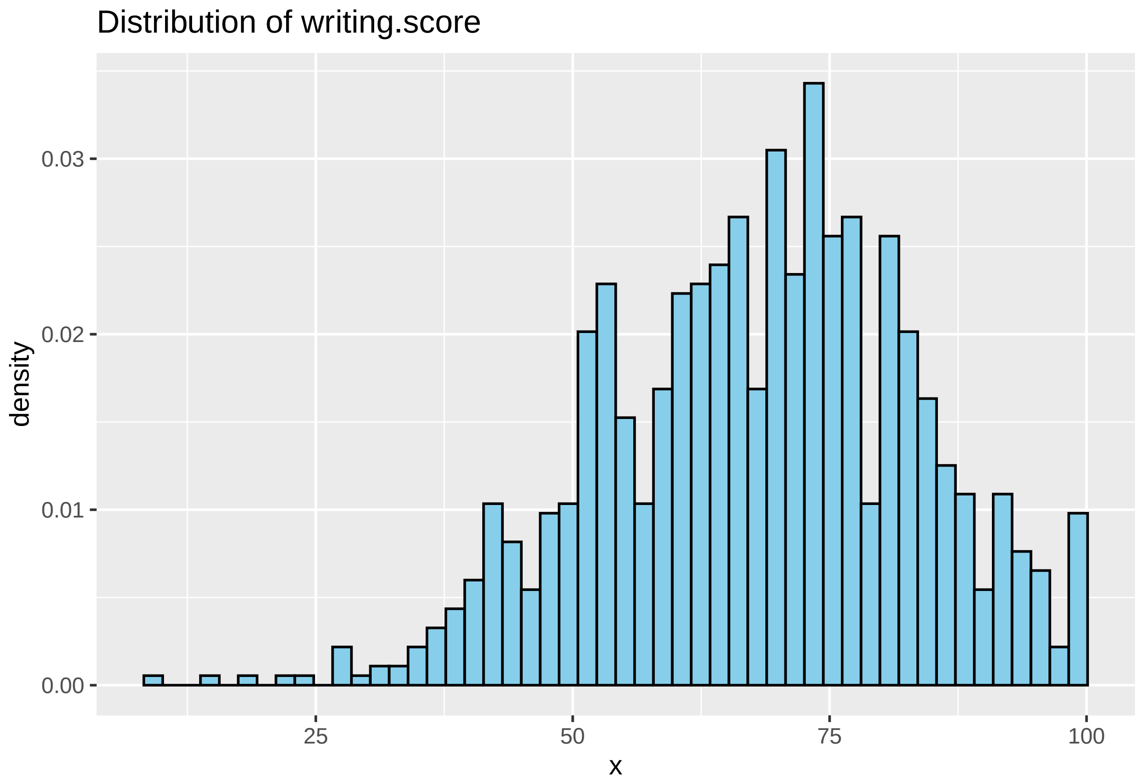
```
# Histogram and density plots for each column with continuous variables
  math_density = ggplot(data = data, aes_string(x = data$math.score)) +
    geom_histogram(aes(y = after_stat(density)), bins = 50,
                   fill = "skyblue", color = "black") +
    ggtitle(paste("Distribution of math.score"))
  print(math_density)
```




```
  reading_density = ggplot(data = data, aes_string(x = data$reading.score)) +
    geom_histogram(aes(y = after_stat(density)), bins = 50,
                   fill = "skyblue", color = "black") +
    ggtitle(paste("Distribution of reading.score"))
  print(reading_density)
```

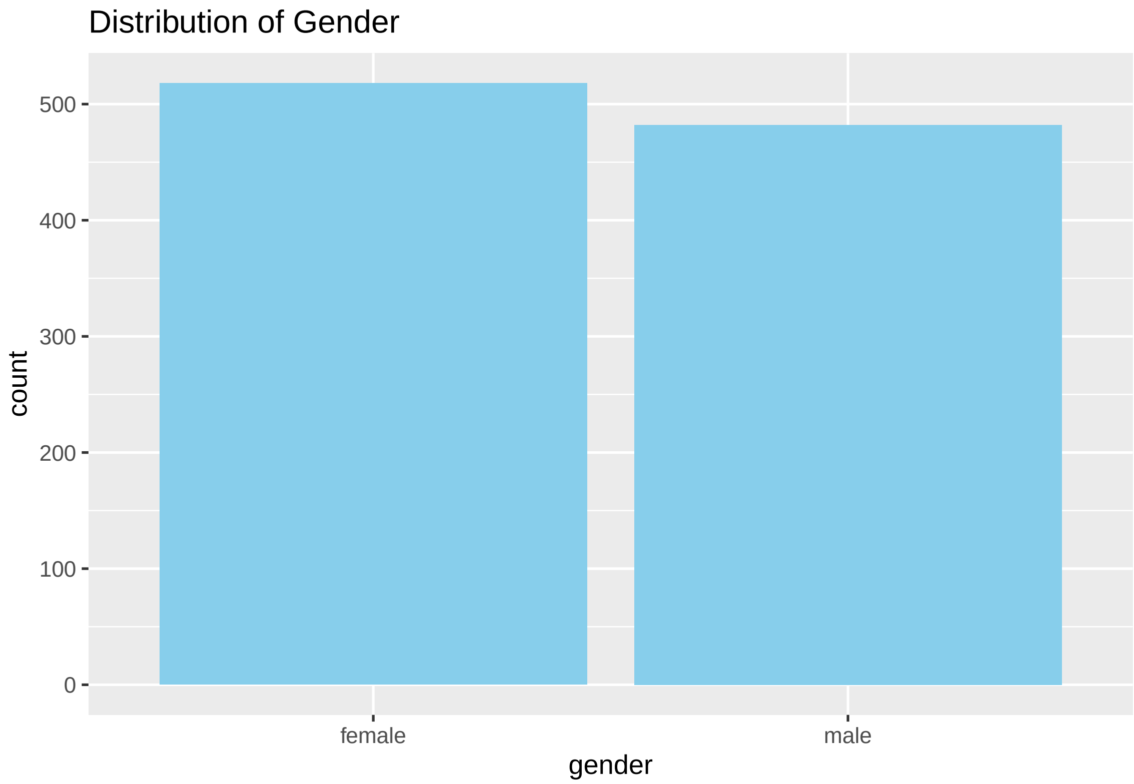



```r
writing_density = ggplot(data = data, aes_string(x = data$writing.score)) +
  geom_histogram(aes(y = after_stat(density)), bins = 50,
                 fill = "skyblue", color = "black") +
  ggtitle(paste("Distribution of writing.score"))
print(writing_density)
```

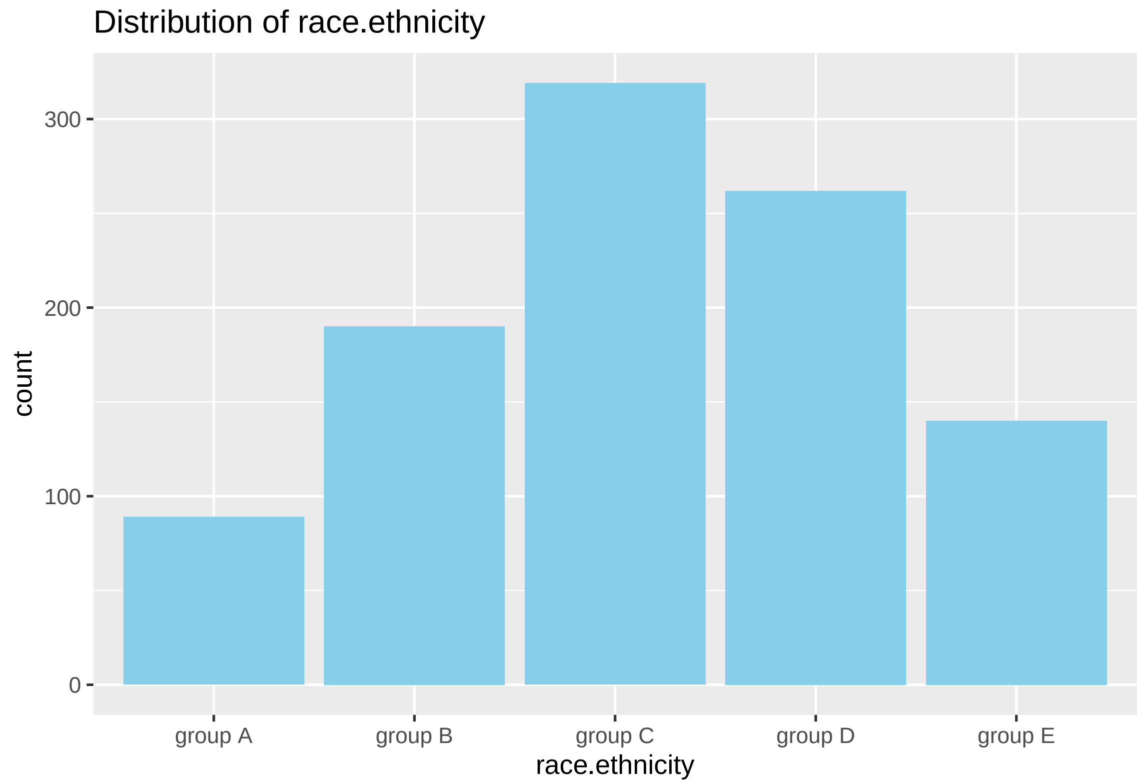


4.1 Visualize the categorical variables

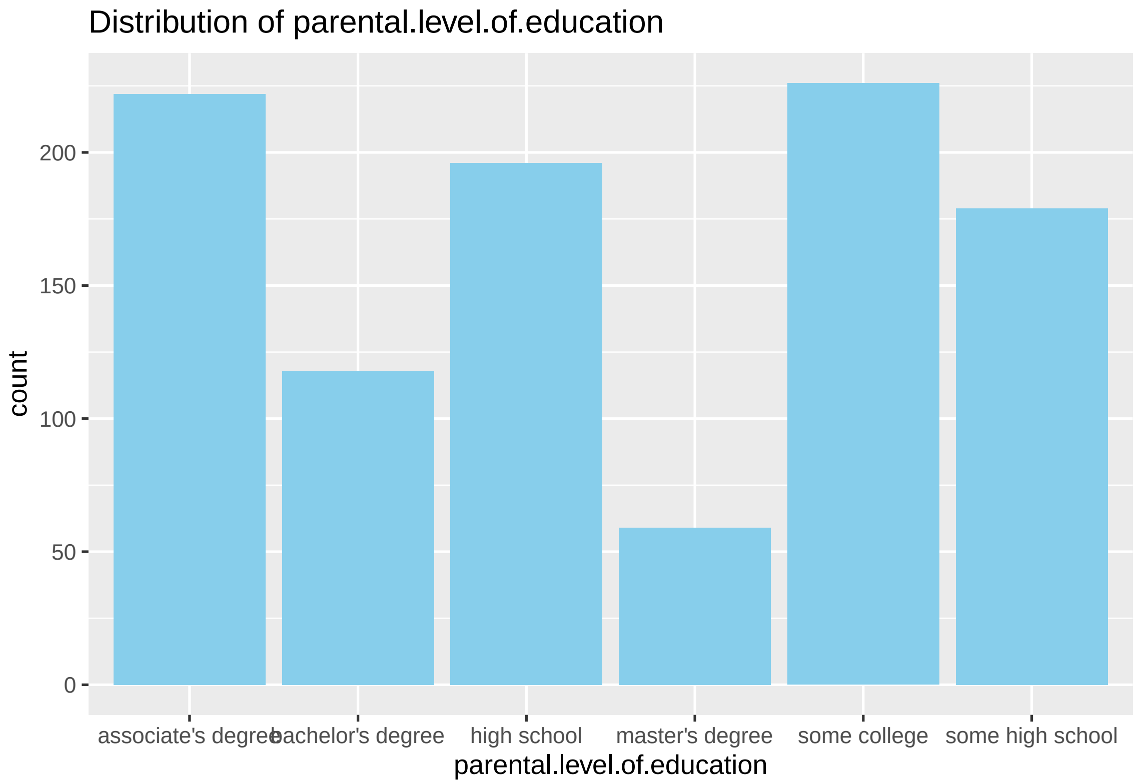
```
ggplot(data, aes(x = gender)) +
geom_bar(fill = "skyblue") +
ggtitle("Distribution of Gender")
```



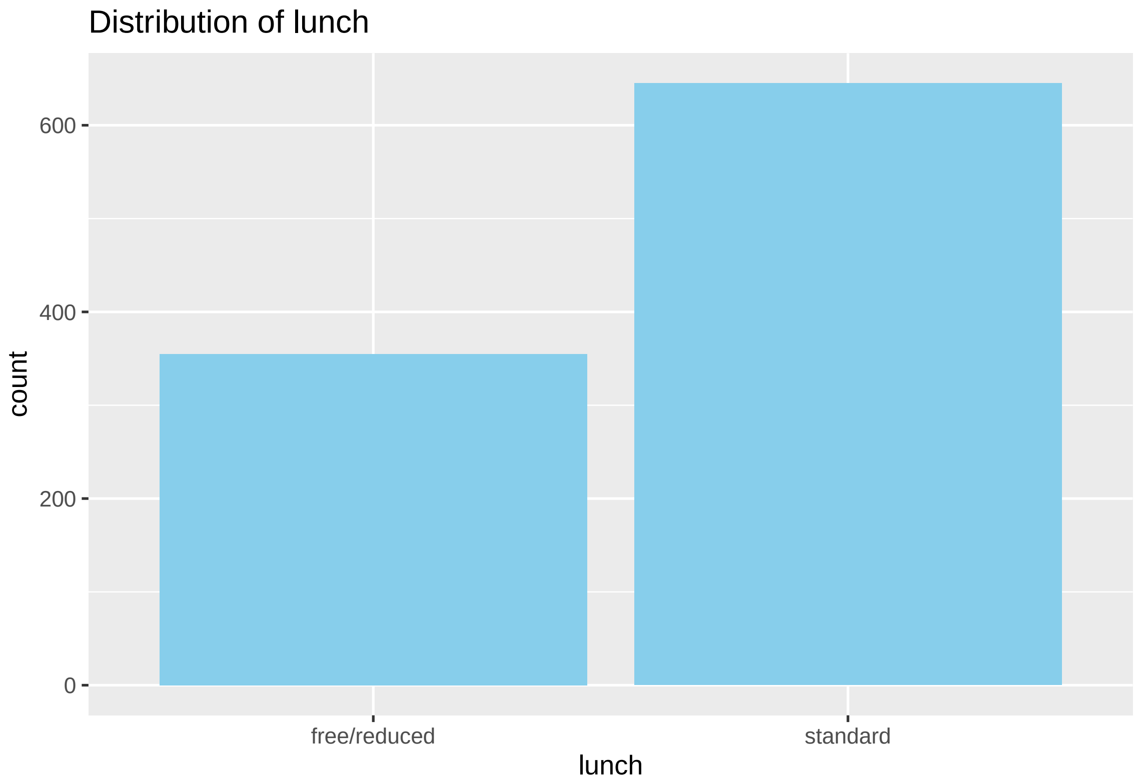

```r
ggplot(data, aes(x = race.ethnicity)) +
geom_bar(fill = "skyblue") +
ggtitle("Distribution of race.ethnicity")
```



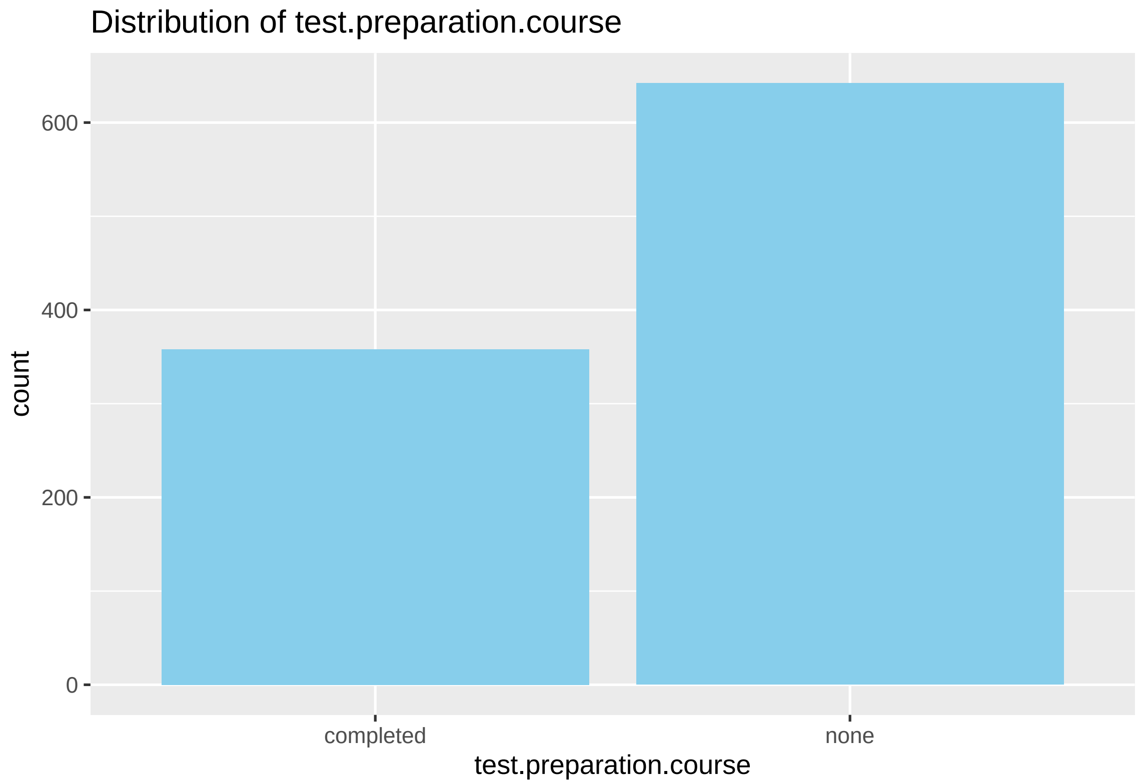

```r
ggplot(data, aes(x = parental.level.of.education)) +
geom_bar(fill = "skyblue") +
ggtitle("Distribution of parental.level.of.education")
```




```r
ggplot(data, aes(x = lunch)) +
geom_bar(fill = "skyblue") +
ggtitle("Distribution of lunch")
```

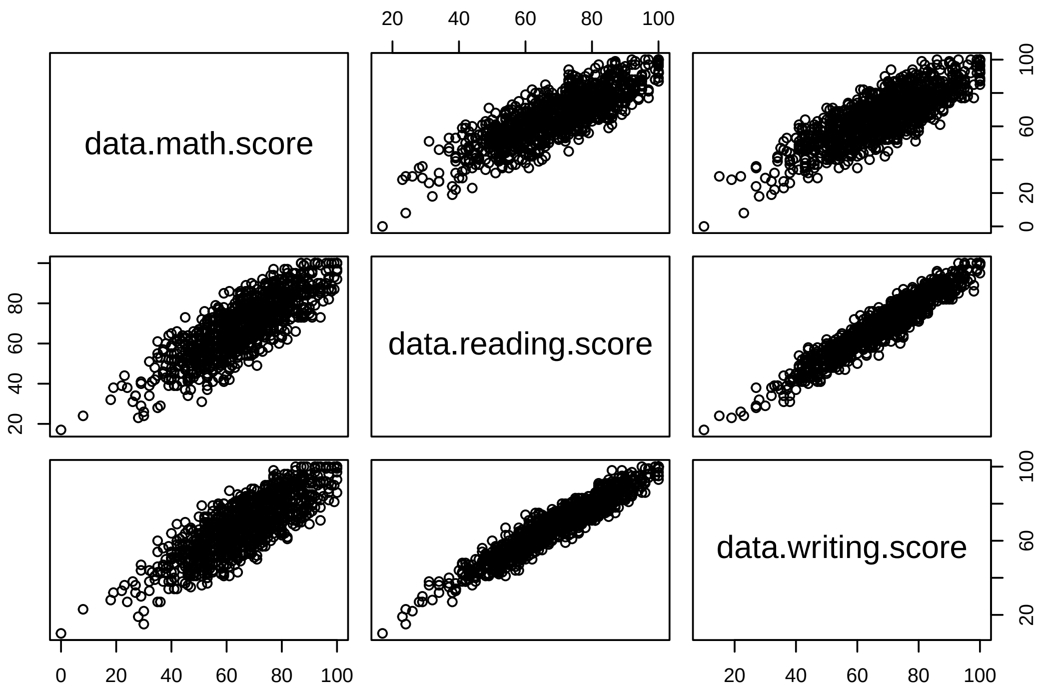



```r
ggplot(data, aes(x = test.preparation.course)) +
geom_bar(fill = "skyblue") +
ggtitle("Distribution of test.preparation.course")
```



4.2 Pairs plot to visualize relationships between numerical variables

```
pairs(data.frame(data$math.score, data$reading.score,
                 data$writing.score))
```



4.2.1 Identification of missing values and outliers

4.3 For math.score

```
# Find Q1, Q3, and IQR
Q1 = quantile(data$math.score, 0.25)
Q3 = quantile(data$math.score, 0.75)
IQR = Q3 - Q1

# Find lower and upper bounds
lower_math = Q1 - 1.5 * IQR
upper_math = Q3 + 1.5 * IQR

# Filter outliers
math_score_outliers = data %>%
  filter(math.score < lower_math | math.score > upper_math)
```

This table shows the information for all students whose math scores were outliers compared to the whole set of math scores:

```r
print(math_score_outliers)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group B            some high school free/reduced
## 2 female        group C            some high school free/reduced
## 3 female        group C                some college free/reduced
## 4 female        group B            some high school free/reduced
## 5 female        group D          associate's degree free/reduced
## 6 female        group B                some college     standard
## 7 female        group B                 high school free/reduced
## 8 female        group B                 high school free/reduced
##   test.preparation.course math.score reading.score writing.score
## 1                    none         18            32            28
## 2                    none          0            17            10
## 3                    none         22            39            33
## 4                    none         24            38            27
## 5                    none         26            31            38
## 6                    none         19            38            32
## 7               completed         23            44            36
## 8                    none          8            24            23
```

4.4 For writing.score

```r
# Find Q1, Q3, and IQR
Q1 = quantile(data$writing.score, 0.25)
Q3 = quantile(data$writing.score, 0.75)
IQR = Q3 - Q1

# Find lower and upper bounds
lower_writing = Q1 - 1.5 * IQR
upper_writing = Q3 + 1.5 * IQR

# Filter outliers
writing_score_outliers = data %>%
  filter(writing.score < lower_writing | writing.score > upper_writing)
```

This table shows the information for all students whose writing scores were outliers compared to the whole set of math scores:

```r
print(writing_score_outliers)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group C            some high school free/reduced
## 2   male        group E            some high school     standard
```

```
## 3   male        group A               some college free/reduced
## 4   male        group B                high school free/reduced
## 5 female        group B                high school free/reduced
##   test.preparation.course math.score reading.score writing.score
## 1                    none          0            17            10
## 2                    none         30            26            22
## 3                    none         28            23            19
## 4                    none         30            24            15
## 5                    none          8            24            23
```

4.5 For reading.score

```
# Find Q1, Q3, and IQR
Q1 = quantile(data$reading.score, 0.25)
Q3 = quantile(data$reading.score, 0.75)
IQR = Q3 - Q1

# Find lower and upper bounds
lower_reading = Q1 - 1.5 * IQR
upper_reading = Q3 + 1.5 * IQR

# Filter outliers
reading_score_outliers = data %>%
  filter(reading.score < lower_reading | reading.score > upper_reading)
```

This table shows the information for all students whose reading scores were outliers compared to the whole set of math scores

```
print(reading_score_outliers)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group C            some high school free/reduced
## 2   male        group E            some high school     standard
## 3   male        group C               some college free/reduced
## 4   male        group A               some college free/reduced
## 5   male        group B                high school free/reduced
## 6 female        group B                high school free/reduced
##   test.preparation.course math.score reading.score writing.score
## 1                    none          0            17            10
## 2                    none         30            26            22
## 3                    none         35            28            27
## 4                    none         28            23            19
## 5                    none         30            24            15
## 6                    none          8            24            23
```

4.6 Determine if there are any missing values

```
print("Are there any NA vlaues in the dataset?")
```

```
## [1] "Are there any NA vlaues in the dataset?"
```

```
anyNA(data)
```

```
## [1] FALSE
```

This dataset has no missing values.

5 Data cleaning and preprocessing steps

Remove outliers, as determined from "Identification of outliers" section above

```
data_clean = data %>%
  filter(
    math.score >= lower_math & math.score <= upper_math,
    reading.score >= lower_reading & reading.score <= upper_reading,
    writing.score >= lower_writing & writing.score <= upper_writing
  )

summary = summary(data_clean)

summary_table_clean = data.frame(
  Min = summary[1,],
  `1st Qu.` = summary[2,],
  Median = summary[3,],
  Mean = summary[4,],
  `3rd Qu.` = summary[5,],
  Max = summary[6,]
)
```

After removing outliers, the minimum, 1st quartile, mean, and 3rd quartile values have changed for some or all of the reading, writing, and math test scores. The median and maximum values did not change for any continuous variables.

```
print(summary_table_clean)
```

```
##                                            Min              X1st.Qu.
##     gender                      Length:988         Class :character
## race.ethnicity                  Length:988         Class :character
```

```
## parental.level.of.education Length:988         Class :character
##    lunch                     Length:988         Class :character
## test.preparation.course     Length:988         Class :character
##   math.score                  Min.   : 27.00     1st Qu.: 57.00
## reading.score                 Min.   : 29.00     1st Qu.: 60.00
## writing.score                 Min.   : 27.00     1st Qu.: 58.00
##                                         Median              Mean
##    gender                    Mode  :character               <NA>
## race.ethnicity               Mode  :character               <NA>
## parental.level.of.education Mode  :character               <NA>
##    lunch                     Mode  :character               <NA>
## test.preparation.course     Mode  :character               <NA>
##   math.score                  Median : 66.00   Mean   : 66.63
## reading.score                 Median : 70.00   Mean   : 69.64
## writing.score                 Median : 69.00   Mean   : 68.57
##                                     X3rd.Qu.               Max
##    gender                                <NA>              <NA>
## race.ethnicity                           <NA>              <NA>
## parental.level.of.education             <NA>              <NA>
##    lunch                                 <NA>              <NA>
## test.preparation.course                 <NA>              <NA>
##   math.score                 3rd Qu.: 77.00   Max.   :100.00
## reading.score                3rd Qu.: 80.00   Max.   :100.00
## writing.score                3rd Qu.: 79.00   Max.   :100.00
```

In case there are any duplicates

```
data_clean = data_clean[!duplicated(data_clean), ]
```

Convert categorical variables to factors and continuous variables to numeric

```
#See data type for each column
str(data_clean)
```

```
## 'data.frame':    988 obs. of  8 variables:
##  $ gender                     : chr  "female" "female" "female" "male" ...
##  $ race.ethnicity             : chr  "group B" "group C" "group B" "group A" ...
##  $ parental.level.of.education: chr  "bachelor's degree" "some college" "master's degree" "asso
##  $ lunch                      : chr  "standard" "standard" "standard" "free/reduced" ...
##  $ test.preparation.course    : chr  "none" "completed" "none" "none" ...
##  $ math.score                 : int  72 69 90 47 76 71 88 40 64 38 ...
##  $ reading.score              : int  72 90 95 57 78 83 95 43 64 60 ...
##  $ writing.score              : int  74 88 93 44 75 78 92 39 67 50 ...
```

```
# Convert categorical variables to factors
data_clean$gender = as.factor(data_clean$gender)
```

```r
data_clean$race.ethnicity = as.factor(data_clean$race.ethnicity)
data_clean$parental.level.of.education =
  as.factor(data_clean$parental.level.of.education)
data_clean$lunch = as.factor(data_clean$lunch)
data_clean$test.preparation.course =
  as.factor(data_clean$test.preparation.course)

# Convert continuous variables to numeric
data_clean$math.score = as.numeric(data_clean$math.score)
data_clean$reading.score = as.numeric(data_clean$reading.score)
data_clean$writing.score = as.numeric(data_clean$writing.score)


# Assign completed/not completed prep course to 1/0 for true or false
data_clean$test.preparation.course =
  ifelse(data_clean$test.preparation.course == "completed", 1, 0)

# Assign standard or reduced lunch to 1/0 for true or false
data_clean$lunch = ifelse(data_clean$lunch == "standard", 1, 0)

# Assign male or female to 1/0 for male or female
data_clean$gender = ifelse(data_clean$gender == "male", 1, 0)
```

6 Variable Selection & Hypothesis Testing

6.1 Implement at least two different variable selection techniques

Set up model selection by creating empty and full models

```r
#empty model as baseline
empty = lm(math.score ~ 1, data = data_clean)

#Full model
model = lm(math.score ~ gender + race.ethnicity + parental.level.of.education
           + lunch + test.preparation.course,
           data = data_clean)
```

Conduct forward selection using AIC

```r
forward_model = step(empty,
                     scope = list(lower = empty, upper = model),
                     direction = "forward")
```

```
## Start:  AIC=5272.73
## math.score ~ 1
```

```
##
##                                Df Sum of Sq    RSS    AIC
## + lunch                        1   23209.3 181722 5156.0
## + race.ethnicity               4   11445.0 193486 5224.0
## + test.preparation.course      1    5726.9 199205 5246.7
## + gender                       1    5425.4 199506 5248.2
## + parental.level.of.education  5    5888.3 199043 5253.9
## <none>                                     204931 5272.7
##
## Step:  AIC=5155.98
## math.score ~ lunch
##
##                                Df Sum of Sq    RSS    AIC
## + race.ethnicity               4   10018.8 171703 5107.9
## + test.preparation.course      1    6287.4 175435 5123.2
## + gender                       1    5060.1 176662 5130.1
## + parental.level.of.education  5    6388.5 175334 5130.6
## <none>                                     181722 5156.0
##
## Step:  AIC=5107.95
## math.score ~ lunch + race.ethnicity
##
##                                Df Sum of Sq    RSS    AIC
## + test.preparation.course      1    5756.5 165947 5076.3
## + gender                       1    4935.8 166768 5081.1
## + parental.level.of.education  5    5188.9 166514 5087.6
## <none>                                     171703 5107.9
##
## Step:  AIC=5076.26
## math.score ~ lunch + race.ethnicity + test.preparation.course
##
##                                Df Sum of Sq    RSS    AIC
## + gender                       1    4857.1 161090 5048.9
## + parental.level.of.education  5    4964.1 160983 5056.2
## <none>                                     165947 5076.3
##
## Step:  AIC=5048.91
## math.score ~ lunch + race.ethnicity + test.preparation.course +
##     gender
##
##                                Df Sum of Sq    RSS    AIC
## + parental.level.of.education  5    5486.7 155603 5024.7
## <none>                                     161090 5048.9
##
## Step:  AIC=5024.67
## math.score ~ lunch + race.ethnicity + test.preparation.course +
##     gender + parental.level.of.education
```

```r
#View variables included in forward-selected model
print("The forward selectoion produces the model:")
```

```
## [1] "The forward selectoion produces the model:"
```

```r
formula(forward_model)
```

```
## math.score ~ lunch + race.ethnicity + test.preparation.course +
##     gender + parental.level.of.education
```

This model retains all possible variables.

Conduct branch and bound selection

```r
bnb_model = regsubsets(math.score ~ gender + race.ethnicity +
                         parental.level.of.education +
                         lunch + test.preparation.course,
                           data = data_clean)

# Get the summary of the subset model
bnb_summary = summary(bnb_model)


# Extract all possible combinations' BIC
#The possible BIC values from models analyzed using branch and bound are:
bnb_summary$bic
```

```
## [1] -104.9631 -141.7081 -166.4965 -188.2966 -199.0165 -206.6015 -209.5545
## [8] -204.6269
```

```r
# View variables included in ideal model
# The variables included in the ideal model chosen with the branch and bound
# model are:
bnb_summary$which[which.min(bnb_summary$bic), ]
```

```
##                                  (Intercept)
##                                         TRUE
##                                       gender
##                                         TRUE
##                         race.ethnicitygroup B
##                                        FALSE
##                         race.ethnicitygroup C
##                                        FALSE
##                         race.ethnicitygroup D
##                                         TRUE
```

```
##                         race.ethnicitygroup E
##                                          TRUE
## parental.level.of.educationbachelor's degree
##                                         FALSE
##       parental.level.of.educationhigh school
##                                          TRUE
##   parental.level.of.educationmaster's degree
##                                         FALSE
##      parental.level.of.educationsome college
##                                         FALSE
##  parental.level.of.educationsome high school
##                                          TRUE
##                                         lunch
##                                          TRUE
##                       test.preparation.course
##                                          TRUE
```

This model retains all possible variables, and is therefore identical to the forward selection model.

Therefore, the model I will use is math.score ~ lunch + race.ethnicity + test.preparation.course + gender + parental.level.of.education.

6.2 Validate model using an appropriate cross-validation technique and assess model performance with rmse and R2

I am using train/test split (80/20) to validate my model's performance because there is a lot of data (1000 rows), and I am not concerned with the exact precision of this model, but rather with its ability to obtain a pretty good value for math score.

```
set.seed(123)
train_index = sample(1:nrow(data_clean), 0.8 * nrow(data_clean))
test = data_clean[-train_index, ]

predictions = predict(model, newdata = test)

# Evaluate performance
actual = test$math.score

rmse = sqrt(mean((actual - predictions)^2))
rmse
```

```
## [1] 12.77249
```

```
r2 = 1 - sum((actual - predictions)^2) / sum((actual - mean(actual))^2)
r2
```

```
## [1] 0.2350381
```

The model's prediction is, on average, 12.8 points off of the value it should have predicted. This seems promising to be able to pinpoint students who will need the most help!

The R2 of this model (both in cross-validation and entire model) is low (0.24, when 1 is ideal). This means the model is likely underfitting, and there is a need for more predictors. Unfortunately, given the goals of this project, additional available data do not exist to include in the model; we have included all available predictors (besides other test scores, which would not contribute to achieving the goals of this project).

6.3 Perform hypothesis tests on coefficients

```
summary = summary(model)
alpha = 0.05


   #Coefficient lunch
#H0: coefficient == 0
#HA: coefficient != 0
#a = 0.05

p_value = summary$coefficients["lunch", "Pr(>|t|)"]
p_value
```

```
## [1] 1.655201e-30
```

Reject H0: The coefficient for lunch is statistically significant.

```
   #Coefficient race.ethnicity
#H0: coefficient == 0
#HA: coefficient != 0
#a = 0.05

p_value = summary$coefficients["race.ethnicitygroup D", "Pr(>|t|)"]
p_value
```

```
## [1] 0.0009893341
```

Reject H0: At least one of the coefficients for race.ethnicity is statistically significant.

```
   #Coefficient test.preparation.course
#H0: coefficient == 0
#HA: coefficient != 0
#a = 0.05

p_value = summary$coefficients["test.preparation.course", "Pr(>|t|)"]
p_value
```

```
## [1] 7.430834e-09
```

Reject H0: The coefficient for test.preparation.course is statistically significant.

```
   #Coefficient gender
#H0: coefficient == 0
#HA: coefficient != 0
#a = 0.05

p_value = summary$coefficients["gender", "Pr(>|t|)"]
p_value
```

```
## [1] 8.652198e-09
```

Reject H0: The coefficient for gender is statistically significant.

```
   #Coefficient parental.level.of.education
#H0: coefficient == 0
#HA: coefficient != 0
#a = 0.05

p_value = summary$coefficients["parental.level.of.educationhigh school",
                                "Pr(>|t|)"]
p_value
```

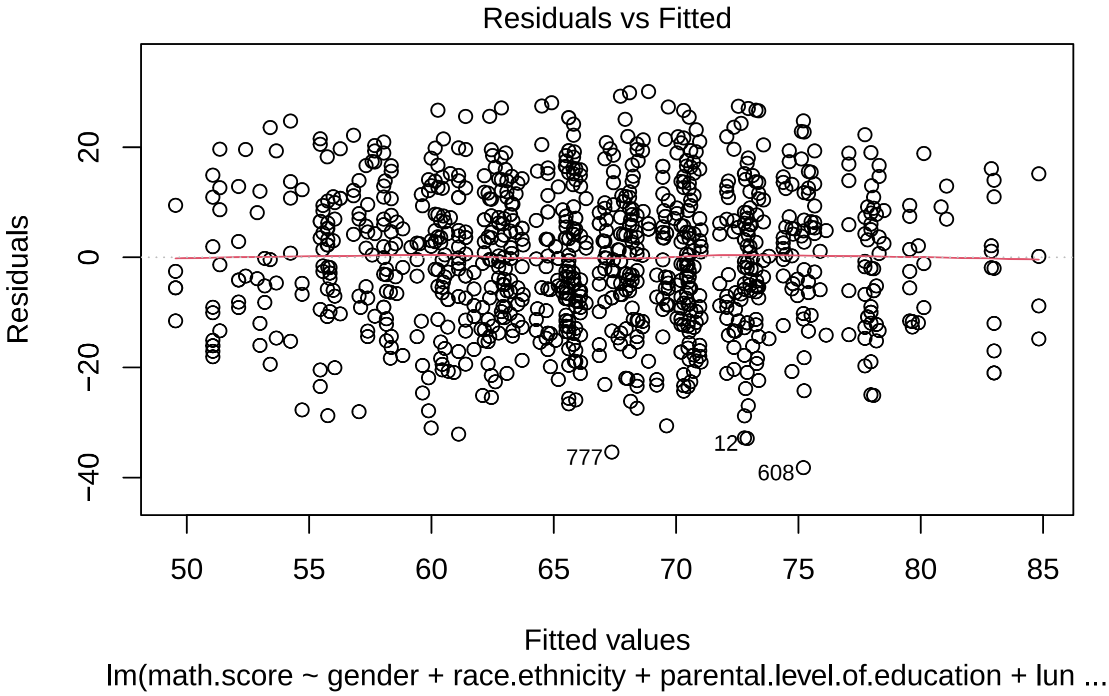
```
## [1] 0.0003340506
```

Reject H0: At least of the coefficients for parental.level.of.education is statistically significant.

7 Regression Assumptions Verification

7.1 Linearity and homoscedasticity (constant variance of residuals) assessment and independence of observations

```
plot(model, which = 1)
```



The plot of fitted values vs residuals looks very random. There is no apparent pattern within these points. Therefore, the assumption of linearity holds for this model.

This plot looks good in regards to variance of the residuals. points appear evenly scattered above and below the '0' line across all fitted values. Therefore, the assumption of homoscedasticity holds for this model.

Also per the residuals vs fitted plot, it appears that the independence of observations assumption holds because there is no trend in residuals as fitted values increase. There was no specified sampling order, and this data is not time-dependent, so there is no reason to believe independence of observation does not hold. It would be important to know what sampling method was used, though, to obtain data.

We can also check independence with the durbin-watson test

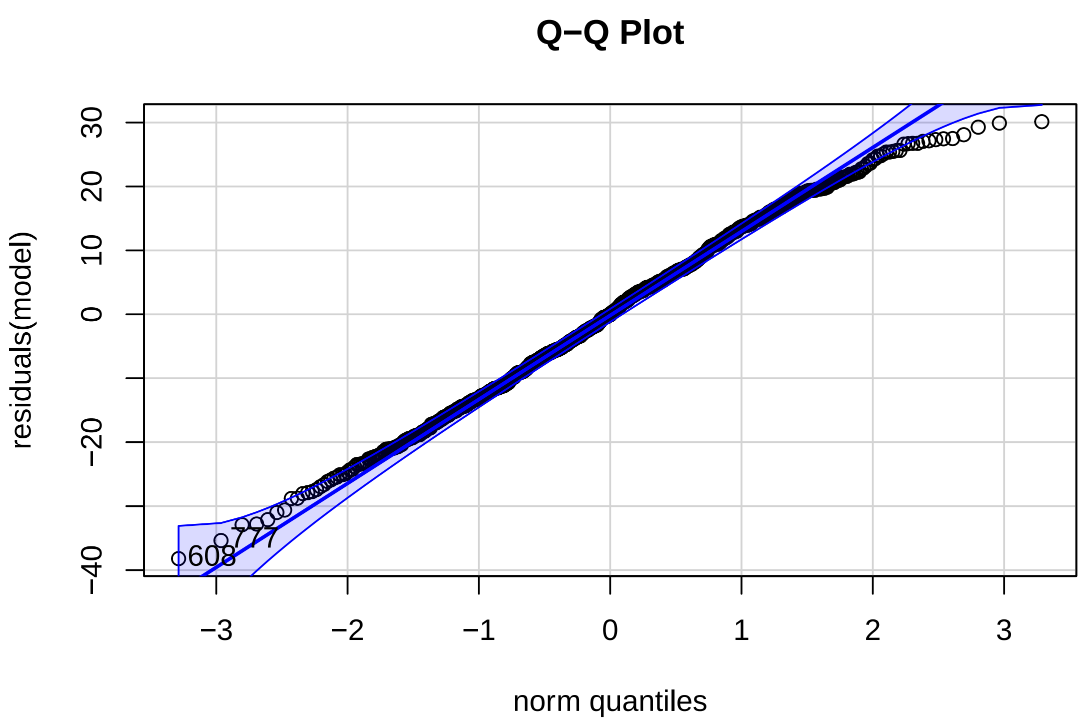
```
dwtest(model)
```

```
##
##  Durbin-Watson test
##
## data:  model
## DW = 2.0115, p-value = 0.5731
## alternative hypothesis: true autocorrelation is greater than 0
```

The p-value is » alpha (0.05), so we do not reject the null hypothesis, and therefore will assume that residuals are not autocorrelated.

7.2 Normality of residuals

```
qqPlot(residuals(model), main = "Q-Q Plot")
```




```
## [1] 608 777
```

Residuals appear to be slightly skewed, but probably not too much to worry about. I also tried transforming math.score in case that could further improve the normality of residuals, but all attempted transformations worsened the skewness. I tried log(), exp(), and squaring math.score.")

log() y did not improve residuals

squaring y did not improve residuals

exp(y) did not improve residuals

7.3 Multicollinearity assessment

We can use the variance inflation factor (VIF) to test for multicolinearity. VIF of 1-2 is ideal.

```
vif(model)
```

```
##                                GVIF Df GVIF^(1/(2*Df))
## gender                      1.012541  1        1.006251
## race.ethnicity              1.049353  4        1.006040
## parental.level.of.education 1.047941  5        1.004694
## lunch                       1.005614  1        1.002803
## test.preparation.course     1.017855  1        1.008888
```

All VIF vlaues are very close to 1, so we can conclude that the multicolinearity assumption is met for this model.

8 Feature Impact Analysis

8.1 Quantify and interpret the impact of each feature on the target and Provide confidence intervals for significant coefficients

```
coeffs = summary$coefficients
coeffs
```

```
##                                              Estimate Std. Error     t value
## (Intercept)                                52.9892571  1.7482488 30.30990536
## gender                                      4.6985860  0.8092680  5.80597037
## race.ethnicitygroup B                       2.8628682  1.6447934  1.74056404
## race.ethnicitygroup C                       2.5759208  1.5354382  1.67764536
## race.ethnicitygroup D                       5.1713760  1.5653877  3.30357520
## race.ethnicitygroup E                      10.3397493  1.7373267  5.95152852
## parental.level.of.educationbachelor's degree  1.8285172  1.4420371  1.26800983
## parental.level.of.educationhigh school     -4.5013893  1.2503330 -3.60015219
## parental.level.of.educationmaster's degree   2.7578299  1.8607187  1.48213153
## parental.level.of.educationsome college    -0.1103487  1.2032570 -0.09170831
## parental.level.of.educationsome high school -3.4481003  1.2893709 -2.67425016
## lunch                                      10.0462159  0.8454457 11.88274489
## test.preparation.course                     4.9228744  0.8440717  5.83229373
##                                                Pr(>|t|)
## (Intercept)                                1.036462e-142
## gender                                      8.652198e-09
## race.ethnicitygroup B                       8.207550e-02
## race.ethnicitygroup C                       9.373685e-02
## race.ethnicitygroup D                       9.893341e-04
```

```
## race.ethnicitygroup E                           3.701565e-09
## parental.level.of.educationbachelor's degree  2.050971e-01
## parental.level.of.educationhigh school         3.340506e-04
## parental.level.of.educationmaster's degree    1.386284e-01
## parental.level.of.educationsome college       9.269486e-01
## parental.level.of.educationsome high school   7.614958e-03
## lunch                                          1.655201e-30
## test.preparation.course                        7.430834e-09
```

The impact of gender on math.score is 4.7. This means that, holding all else constant, a male student would be associated with an average of 4.7(+- 0.8) more points of their math score than a female student.

The impact of lunch on math.score is 10.0. This means that, holding all else constant, a student who receives standard lunch prices would be associated with an average of 10.0(+- 0.8) more points of their math score than a student who receives reduced lunch prices.

The impact of prep courses on math.score is 4.9. This means that, holding all else constant, a student who completed a prep course would be associated with an average of 4.9(+- 0.8) more points of their math score than a student who did not complete a prep course.

The impact of a student being in race/ethnicity group D or E on math.score is 5.2 and 10.3, respectively. This means that, holding all else constant, a student who was in race/ethnicity groups D or E would be associated with an average of 5.2(+-1.6) and 10.3(+-1.7) more points of their math score than a student in race/ethnicity group A.

The impact of a student's parental level of education being 'high school' or 'some high school' on math.score is -4.5 and -3.4, respectively. This means that, holding all else constant, a student whose parent completed high school or some high school would be associated with an average of 4.5(+-1.3) and 3.4(+-1.3) LESS points of their math score than a student in the reference category, whose parent completed an associates degree.

9 Conclusions

From this analysis, I have demonstrated that gender, race, parental education, lunch price, and test prep courses all significantly factor into determining a student's math test score. The overall R2 of 0.23 means that there are other factors that need to be considered to accurately predict a students's test score, though. These factors could include parents age, number of siblings the student has, talkativeness of the student. In a larger survey, maybe a short math pre-test could be administered as well to assess a student's capabilities.

10 References

Data was sources from Kaggle.com. The dataset was called 'students-performance in exams', and was uploaded by Jakki Seshapanpu.

The online dataset can be found at the following link:

https://www.kaggle.com/datasets/spscientist/students-performance-in-exams