Exploratory Data Analysis, Linear Regression, and Prediction Modeling of Student Performance

Travis Gubbe September 25, 2022

In this R Markdown session, I will use various methods of exploratory data analysis to examine the characteristics of the “Students Performance in Exams” dataset. In addition, linear regression to view relationships within the data.

About the Dataset

The dataset used can be found on the Kaggle website, an online platform for data scientists containing free datasets and code collaboration. Below is the link for the dataset: https://www.kaggle.com/datasets/spscientist/students-performance-in-exams.

The data contains eight variables used to explore the effect of various factors on test scores. The variables are:

- Gender - male or female
- Race/Ethnicity - split into 5 Groups (A, B, C, D, E)
- Parent Education Level - a student’s parent’s highest level of education
- Lunch - whether a student is in the standard or free/reduced lunch program
- Test Prep Course - whether or not a student completed the test preparation course
- Math Score
- Reading Score
- Writing Score

Note: this is a fictional dataset used strictly for to demonstrate beginner data analysis skills. The results are not official and should not be used to conclude actual relationships between the variables listed and education

First, the dataset is loaded into R and saved as “student”.

```
student <- read.csv("~/R datasets/StudentsPerformance.csv")
```

Viewing and Cleaning the Data

Now that the dataset is loaded into R, the next step is to view the data and see if it's clean for analysis.

```
#Inspect the data frame.
head(student)
```

```
##   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
```

```
#View the column names.
colnames(student)
```

```
## [1] "gender"                      "race.ethnicity"
## [3] "parental.level.of.education" "lunch"
## [5] "test.preparation.course"     "math.score"
## [7] "reading.score"               "writing.score"
```

```
#View summary of the data frame.
summary(student)
```

```
##     gender          race.ethnicity     parental.level.of.education
```

```
##  Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##     lunch           test.preparation.course   math.score     reading.score
##  Length:1000        Length:1000              Min.   :  0.00   Min.   : 17.00
##  Class :character   Class :character         1st Qu.: 57.00   1st Qu.: 59.00
##  Mode  :character   Mode  :character         Median : 66.00   Median : 70.00
##                                              Mean   : 66.09   Mean   : 69.17
##                                              3rd Qu.: 77.00   3rd Qu.: 79.00
##                                              Max.   :100.00   Max.   :100.00
##  writing.score
##  Min.   : 10.00
##  1st Qu.: 57.75
##  Median : 69.00
##  Mean   : 68.05
##  3rd Qu.: 79.00
##  Max.   :100.00
```

```
#View data types in the data frame.
str(student)
```

```
## 'data.frame':    1000 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
##  $ lunch                      : chr  "standard" "standard" "standard" "free/red
##  $ 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 ...
```

I don't like the "." in the names, electing to change the "." to a "_" for easier reading. Also, the name of the "race_ethnicity" variable is shortened to "ethnicity".

```
#Rename the columns in the student data frame.
student <- student%>%rename(parental_education = parental.level.of.education, math
#View the updated column names in the student data frame.
colnames(student)
```

```
## [1] "gender"             "ethnicity"          "parental_education"
## [4] "lunch"              "test_prep_course"   "math_score"
## [7] "reading_score"      "writing_score"
```

Next, I want to check if there are any missing values in the dataset.

```
#Print the total number of missing values in the data frame.
sum(is.na(student))
```

```
## [1] 0
```

Now that I know there aren't any missing values, next I check to see if there are any duplicates in the dataset.

```
#Create a variable storing the amount of duplicates in the data frame.
duplicates <- student%>%duplicated()
#Displays how many duplicates are present in a table. If a value is not a duplicat
duplicates_count <- duplicates%>%table()
duplicates_count
```

```
## .
## FALSE
##  1000
```

So far, there are no missing values or duplicates in the data. Next, I want to view the distribution of the data for each variable.

First, a count and frequency table is created to see the distribution of the data.

```
#The total number of males and females in the data frame.
count_gender <- student%>%count(gender)
count_gender
```

```
##   gender   n
## 1 female 518
## 2   male 482
```

```r
#Create a frequency table to show the percentage of each gender in the data frame.
freq_gender <- table(student$gender)/length(student$gender)
freq_gender
```

```
##
## female   male
##  0.518  0.482
```

```r
#The total number of each ethnicity group in the data frame.
count_ethnicity <- student%>%count(ethnicity)
count_ethnicity
```

```
##   ethnicity   n
## 1   group A  89
## 2   group B 190
## 3   group C 319
## 4   group D 262
## 5   group E 140
```

```r
#Creates a frequency table to show the percentage of each ethnicity group in the d
freq_ethnicity <- table(student$ethnicity)/length(student$ethnicity)
freq_ethnicity
```

```
##
## group A group B group C group D group E
##   0.089   0.190   0.319   0.262   0.140
```

```r
#The total number of each parental highest education group in the data frame.
count_parental_education <- student%>%count(parental_education)
count_parental_education
```

```
##   parental_education   n
## 1 associate's degree 222
## 2  bachelor's degree 118
## 3        high school 196
## 4    master's degree  59
## 5       some college 226
```

```
## 6   some high school 179
```

```
#Creates a frequency table to show the percentage of each parental highest educati
freq_parental <- table(student$parental_education)/length(student$parental_educati
freq_parental
```

```
##
## associate's degree  bachelor's degree        high school    master's degree
##              0.222              0.118              0.196              0.059
##       some college   some high school
##              0.226              0.179
```

```
#The total number of students who took the test prep course.
count_test_prep <- student%>%count(test_prep_course)
count_test_prep
```

```
##   test_prep_course   n
## 1        completed 358
## 2             none 642
```

```
#Creates a frequency table to show the percentage of each parental highest educati
freq_test <- table(student$test_prep_course)/length(student$test_prep_course)
freq_test
```

```
##
## completed      none
##     0.358     0.642
```

```
#The total number of students in each lunch group.
count_lunch <- student%>%count(lunch)
count_lunch
```

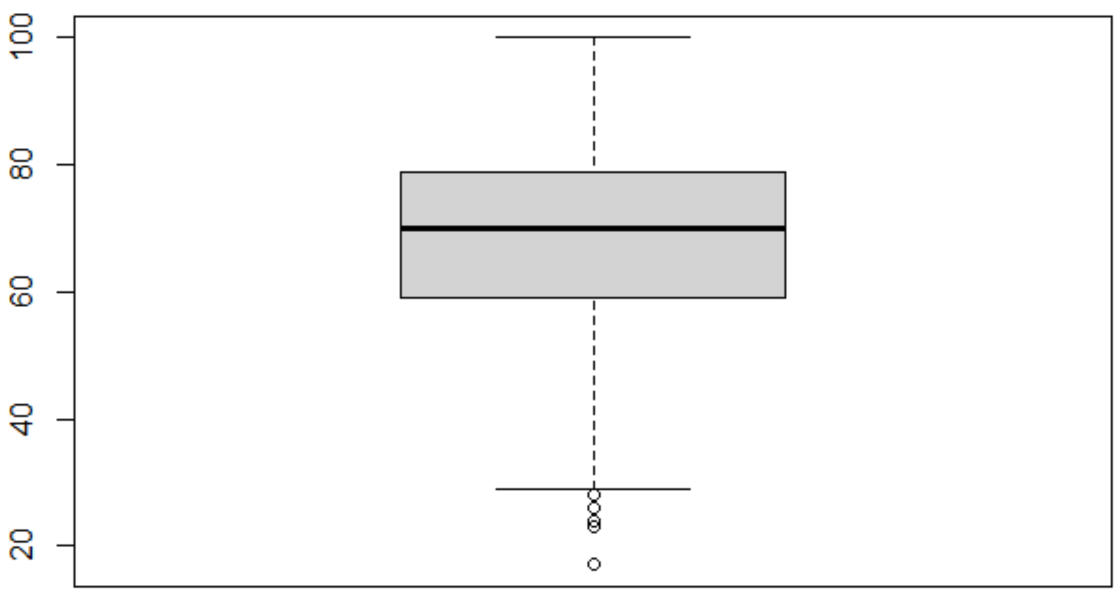
```
##          lunch   n
## 1 free/reduced 355
## 2     standard 645
```

```
#Creates a frequenct table to show the percentage of each lunch group.
freq_lunch <- table(student$lunch)/length(student$lunch)
freq_lunch
```

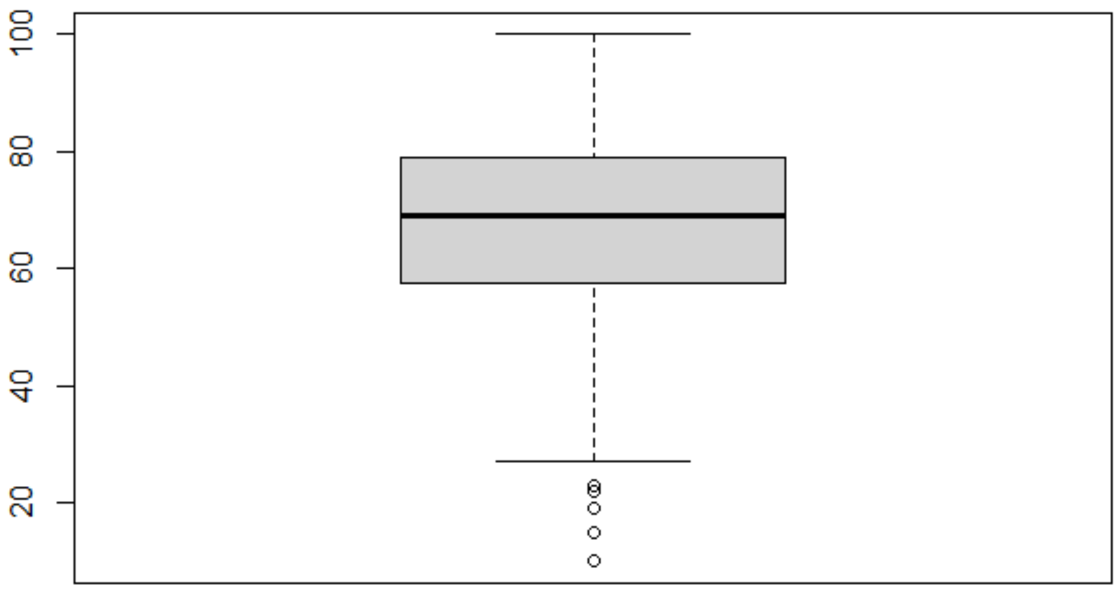
```
##
## free/reduced     standard
##        0.355        0.645
```

Boxplots can also be used to view the distribution of the data in each test score. The boxplot will show the interquartile range (IQR), mean, and outliers in the data.

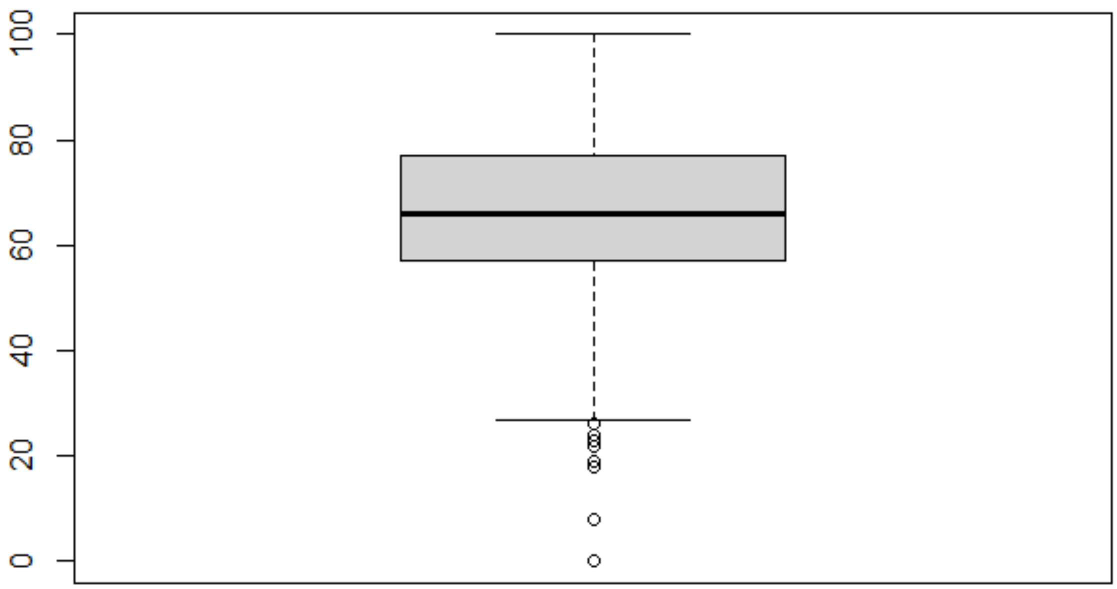
```
boxplot(student$reading_score)
```




```
boxplot(student$writing_score)
```




```
boxplot(student$math_score)
```



From the boxplots, there appear to be multiple outliers in each test score variable. The one I'm most concerned with is the "0" in the math test section.

I want to find out which student scored a 0 on the math exam.

```
which.min(student$math_score)
```

```
## [1] 60
```

Row 60 (student #60) contains the student who scored a "0" on the math test. In addition, I want to view the values of student #60 to see if all of their exam results and to determine if they are an outlier in the data frame:

```
#View student 60 to see their overall test results and determine if they're an out
print(student[60, ])
```

```
##    gender ethnicity parental_education        lunch test_prep_course math_score
## 60 female   group C   some high school free/reduced             none          0
```

```
##    reading_score writing_score
## 60            17            10
```

Student #60 has very low scores in each test. I can check the range of each test to see if student #60 has the lowest scores in each test:

```
range(student$reading_score)
```

```
## [1]  17 100
```

```
range(student$writing_score)
```

```
## [1]  10 100
```

```
range(student$math_score)
```

```
## [1]   0 100
```

Student #60 has the lowest test score in each section. To me, student #60 is an outlier and can be removed from the data.

```
#Remove the 0 test score from the data set. Will also remove the student completel
student<- subset(student, math_score != 0)
#Counts the amount of rows in the data to see if the outlier was properly removed.
count(student)
```

```
##     n
## 1 999
```

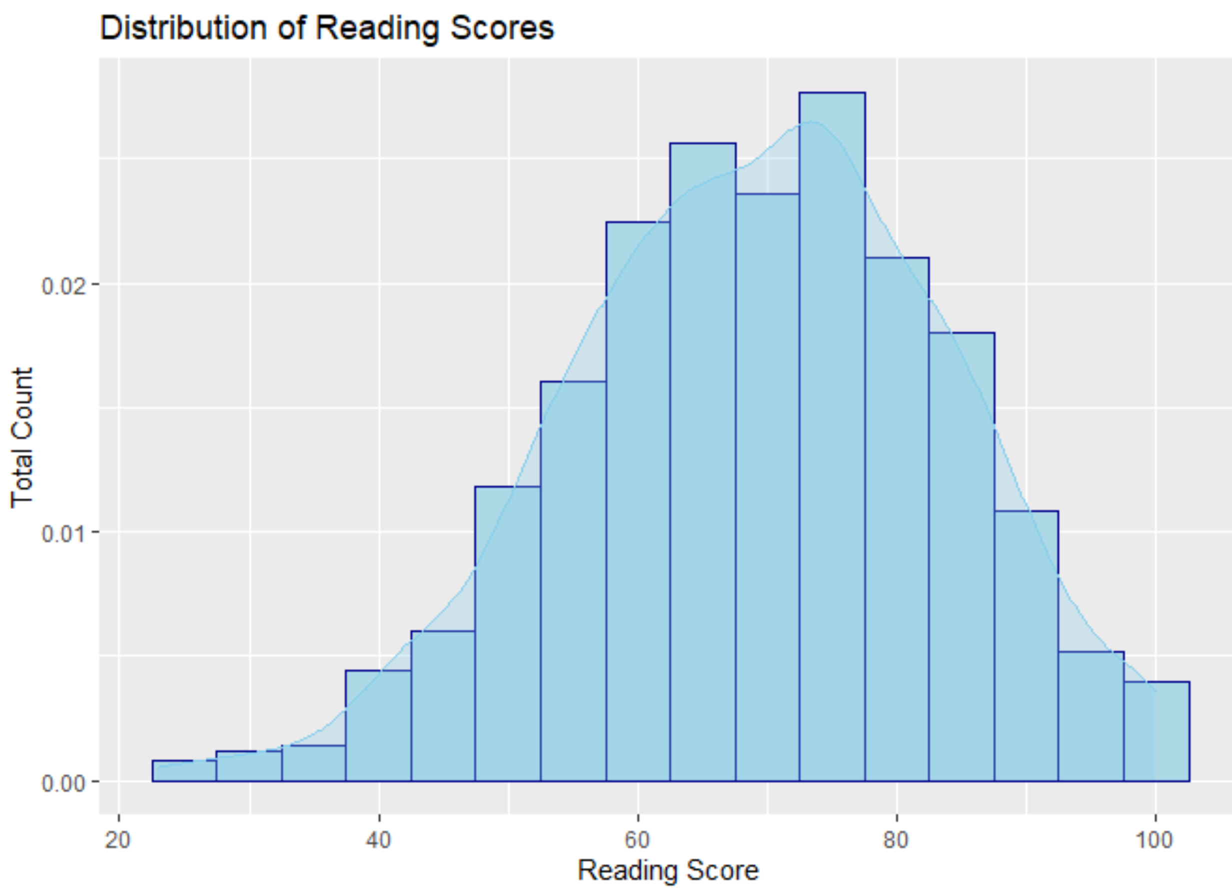
```
#Count the amount in each gender. Compared to the original count, the female count
count_gender <- student%>%count(gender)
count_gender
```

```
##   gender   n
## 1 female 517
## 2   male 482
```

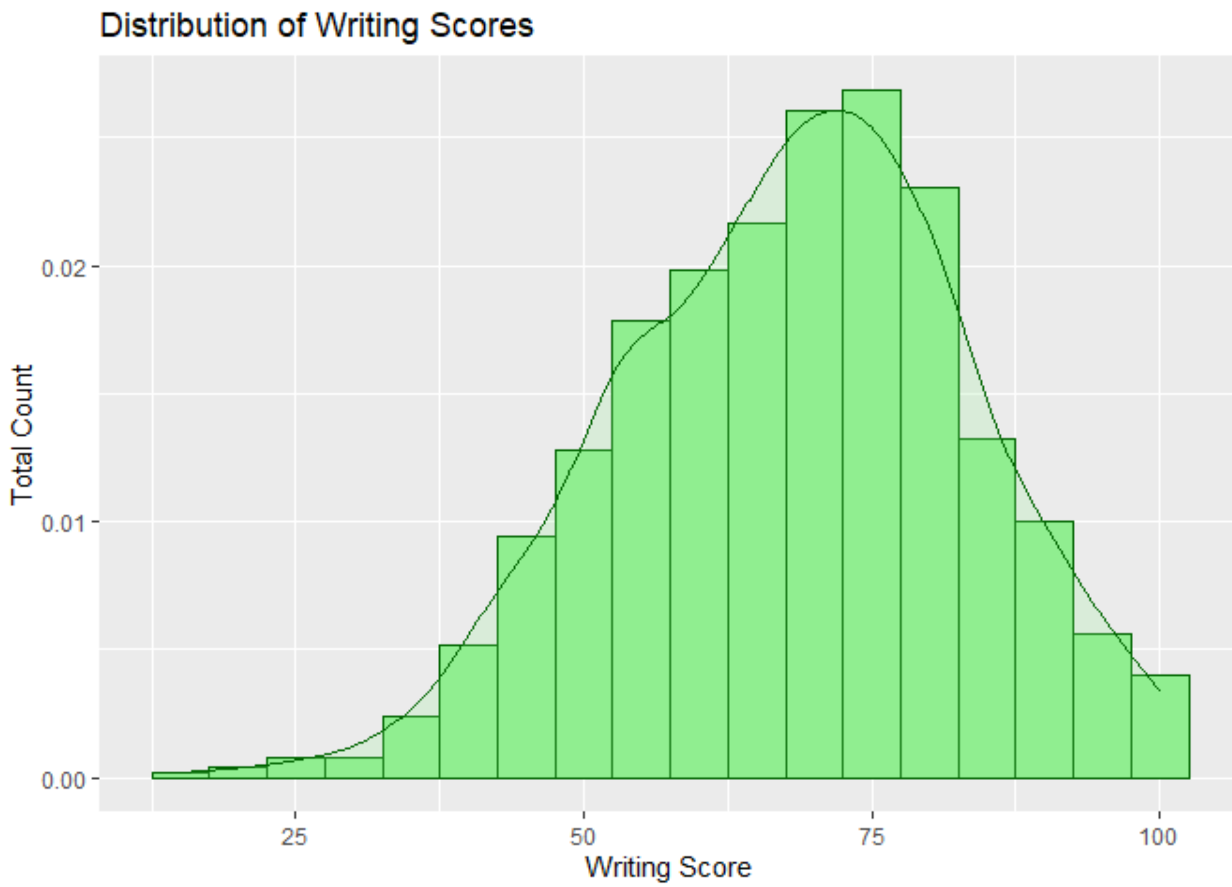
Viewing the Distribution of the Data Among Scores

With the data inspected and cleaned, histograms are created to see the distribution of the variables. First, the distribution of the test scores are viewed using ggplot2.

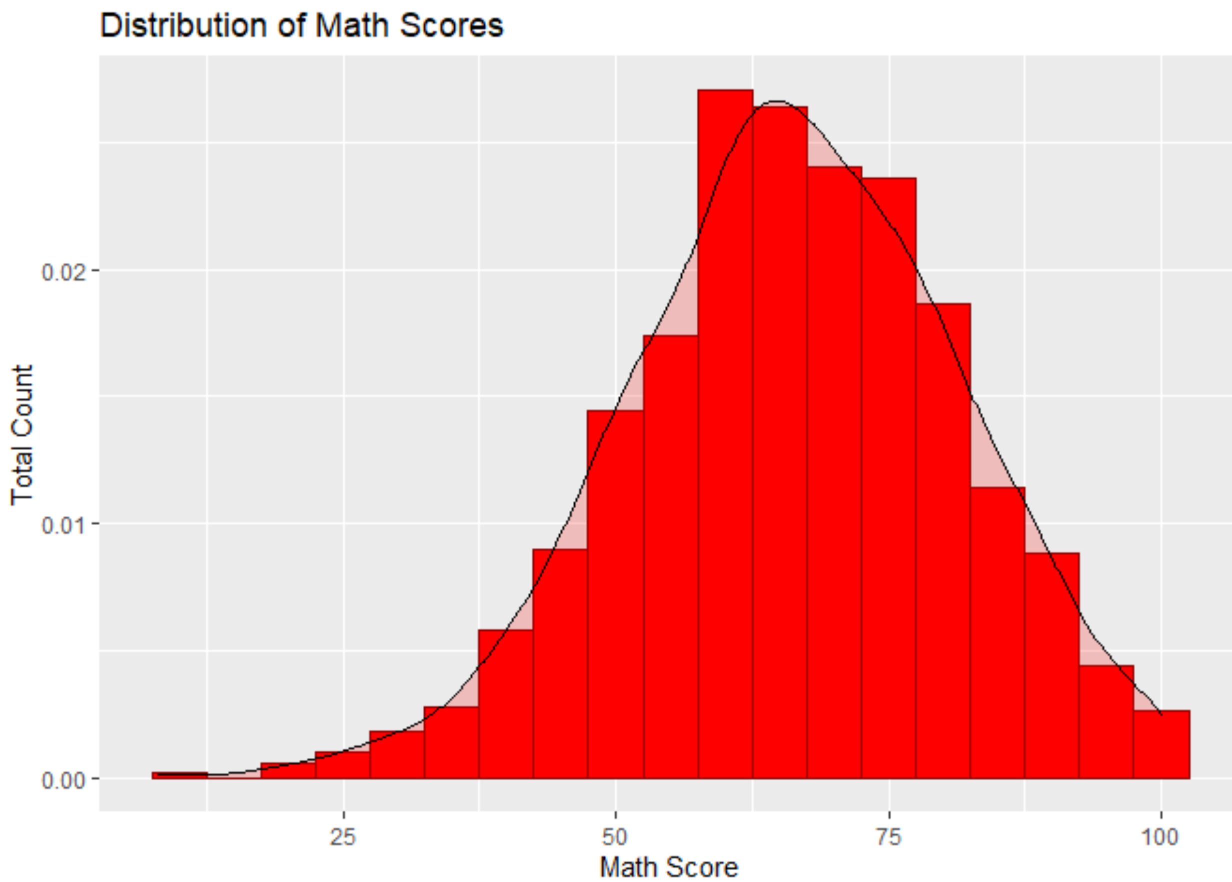
```
#Histogram displaying the distribution of the Reading Scores for the data frame
ggplot(data = student, aes(reading_score))+geom_histogram(aes(y = ..density..), co
```




```
#Histogram displaying the distribution of the Writing Scores for the data frame
ggplot(data = student, aes(writing_score))+geom_histogram(aes(y = ..density..), co
```

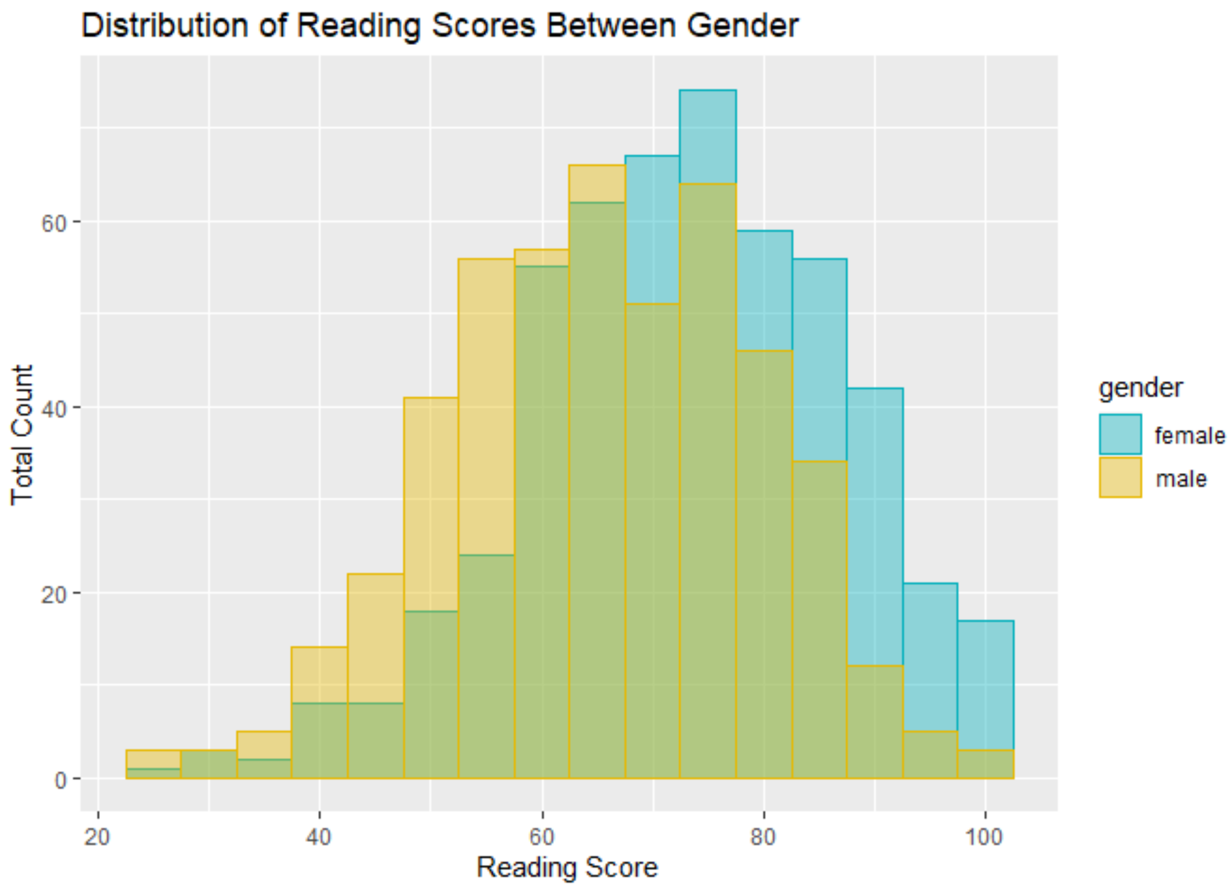



```
#Histogram displaying the distribution of the Math Scores for the data frame
ggplot(data = student, aes(math_score))+geom_histogram(aes(y = ..density..), color
```

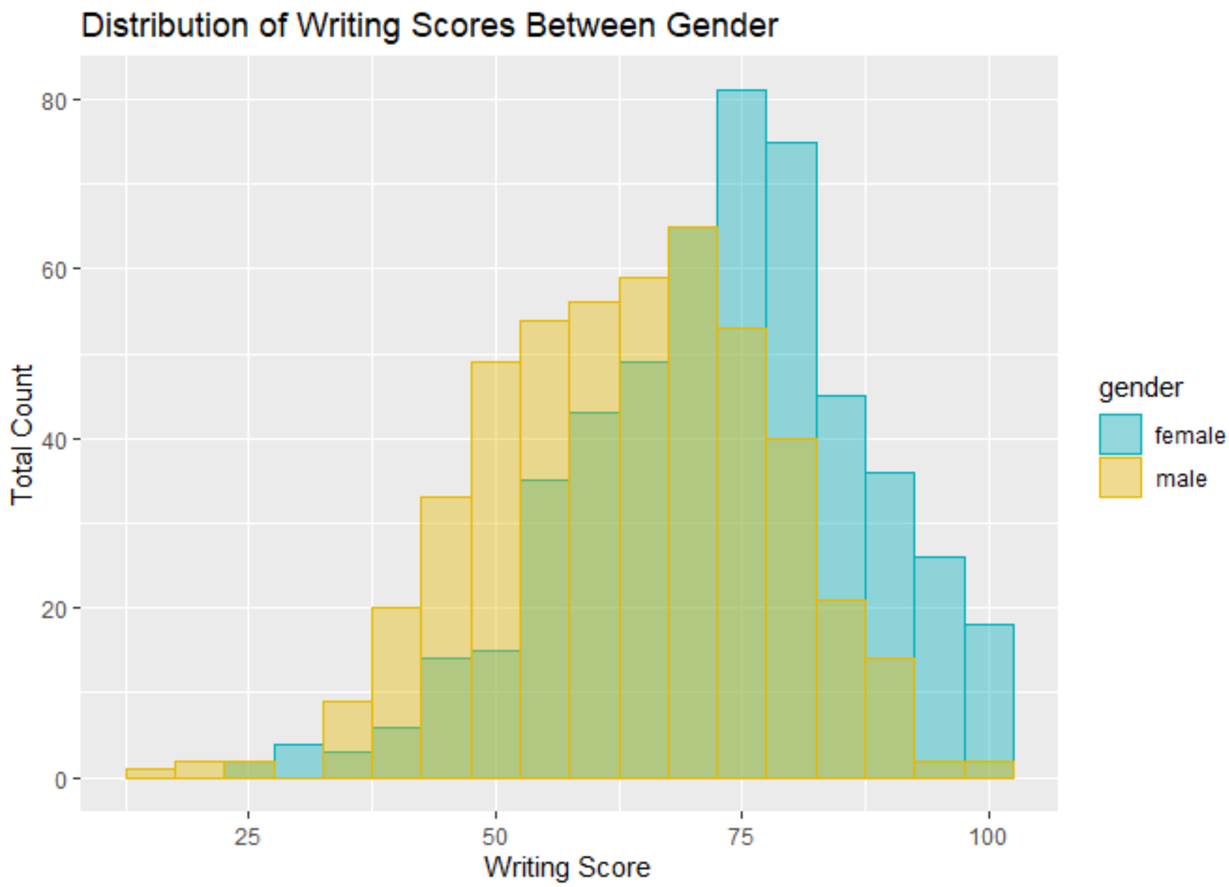


Next, the distribution of gender in each test is viewed using ggplot2.

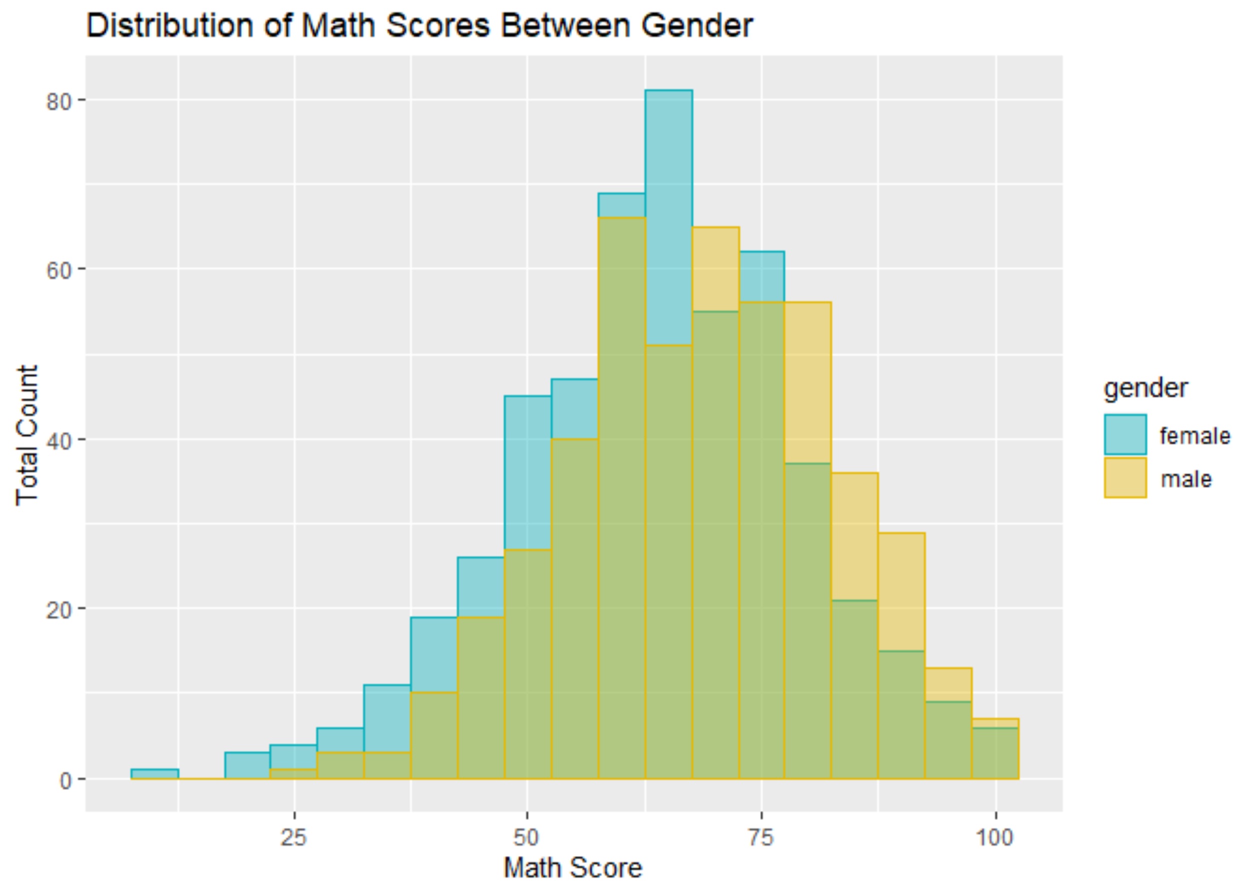
```
#Histogram displaying distribution of reading scores between gender
ggplot(student, aes(x = reading_score))+geom_histogram(aes(color = gender, fill =
```




```
#Histogram displaying distribution of writing scores between gender
ggplot(student, aes(x = writing_score))+geom_histogram(aes(color = gender, fill =
```




```
#Histogram displaying distribution of math scores between gender
ggplot(student, aes(x = math_score))+geom_histogram(aes(color = gender, fill = gen
```

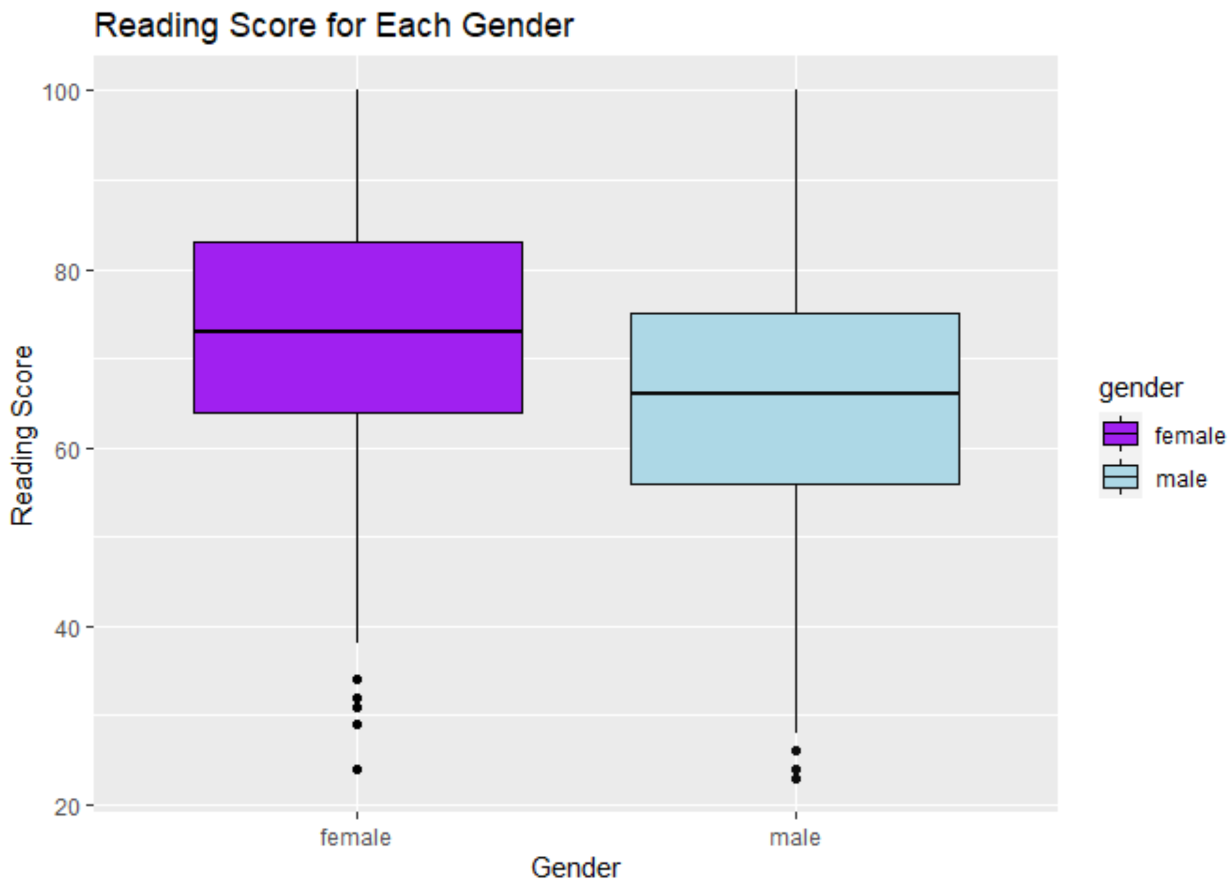


Overall, the data appears to be normally distributed.

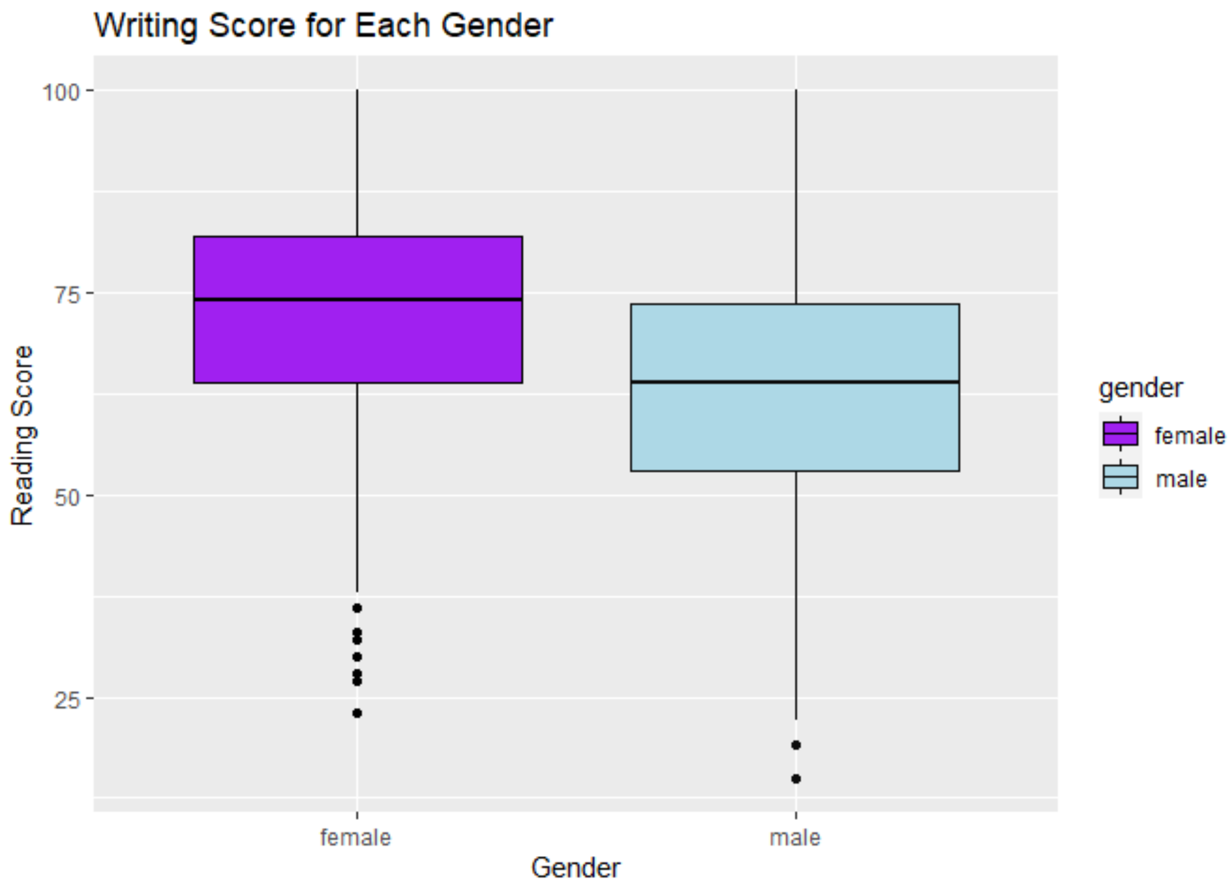
Boxplots of Each Variable

Next, boxplots are created to compare the test scores to the categorical variables. A boxplot comparison is done for gender, race/ethnicity, parental education level, lunch program, and test preparation.

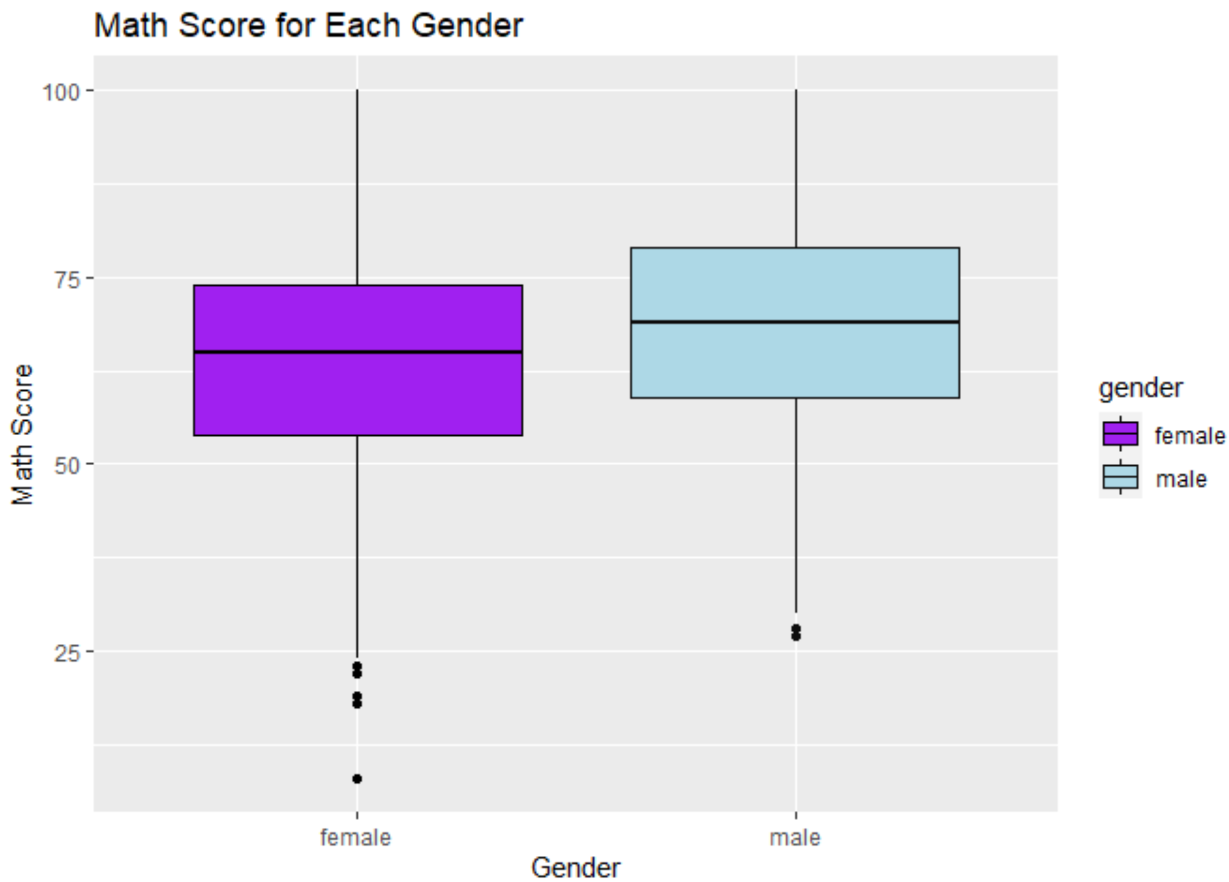
```
ggplot(student, aes(gender, reading_score, fill = gender, color = gender))+geom_bo
```



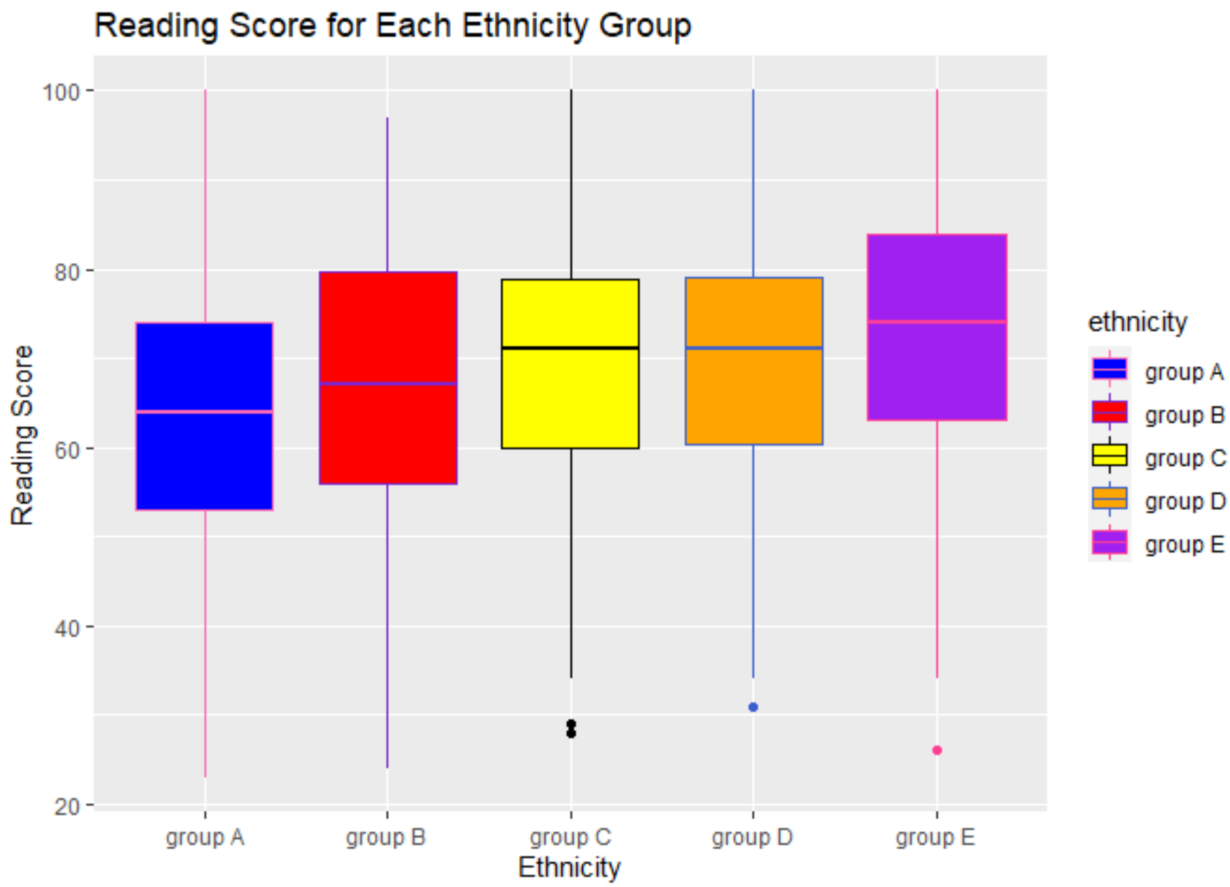

```
ggplot(student, aes(gender, writing_score, fill = gender, color = gender))+geom_bo
```



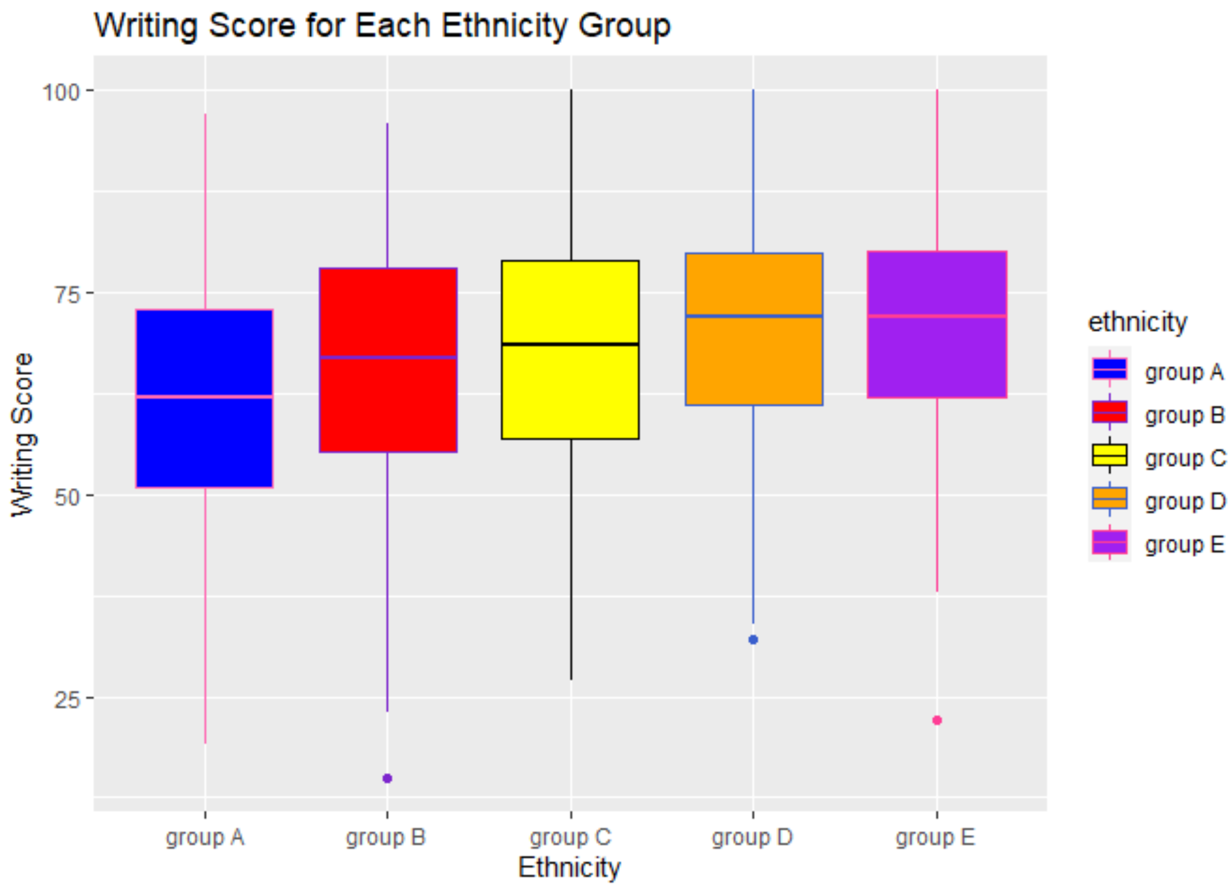

```
ggplot(student, aes(gender, math_score, fill = gender, color = gender))+geom_boxpl
```



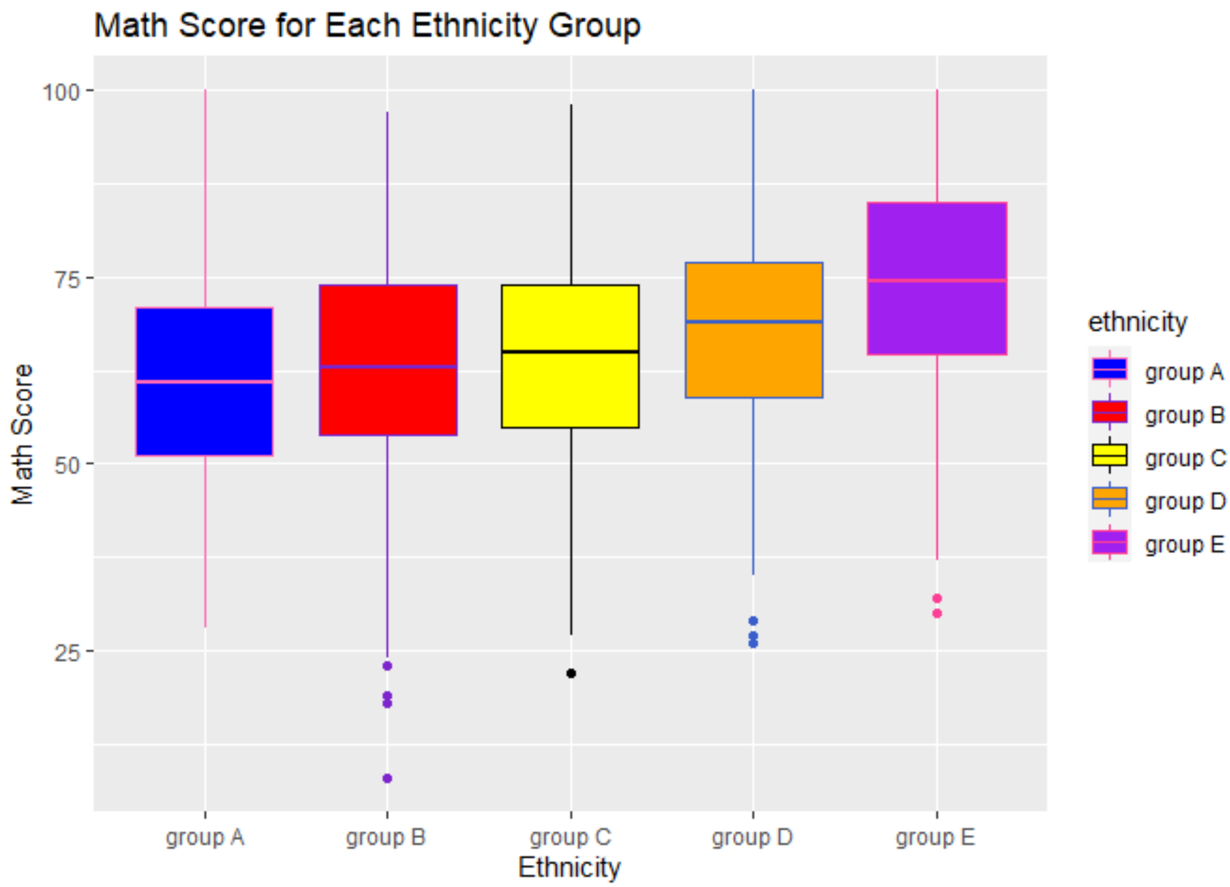

```
ggplot(student, aes(ethnicity, reading_score, fill = ethnicity, color = ethnicity)
```



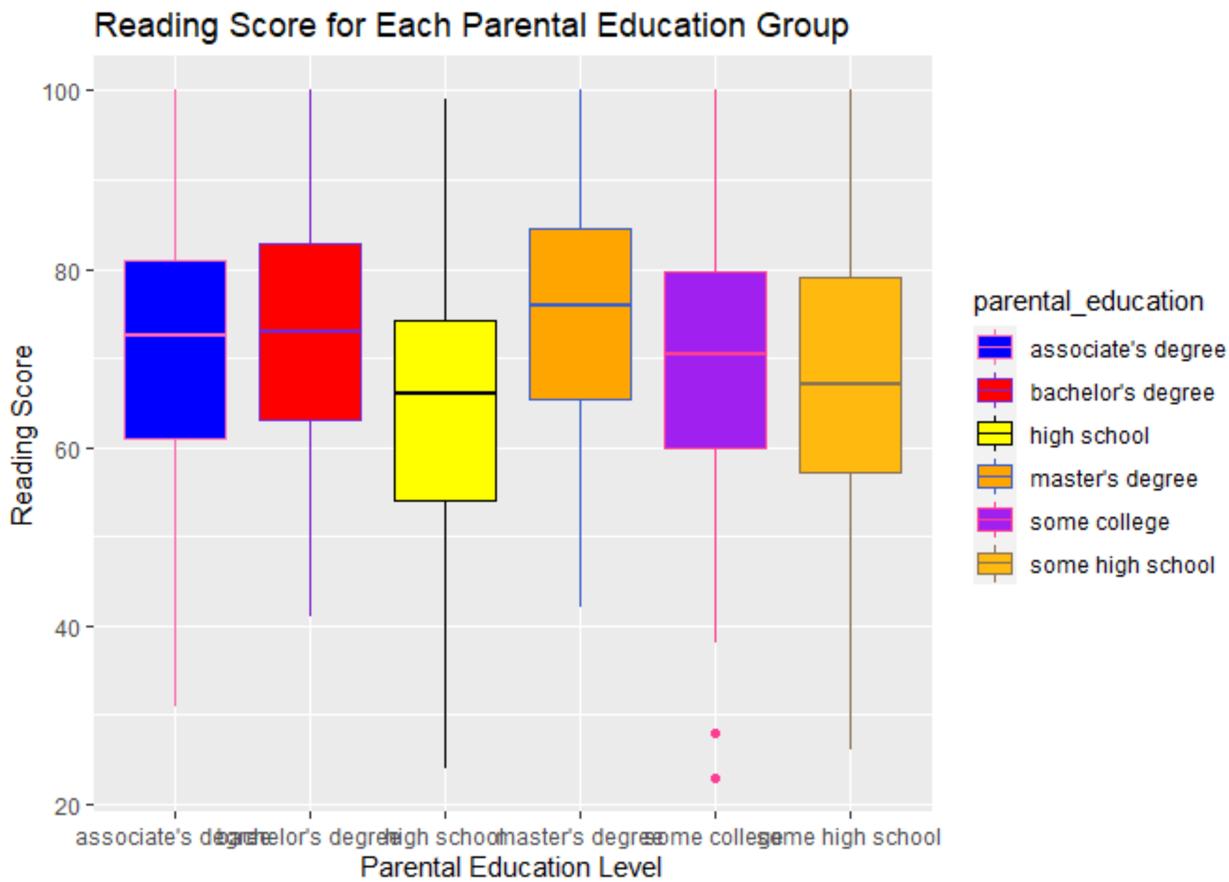

```
ggplot(student, aes(ethnicity, writing_score, fill = ethnicity, color = ethnicity)
```



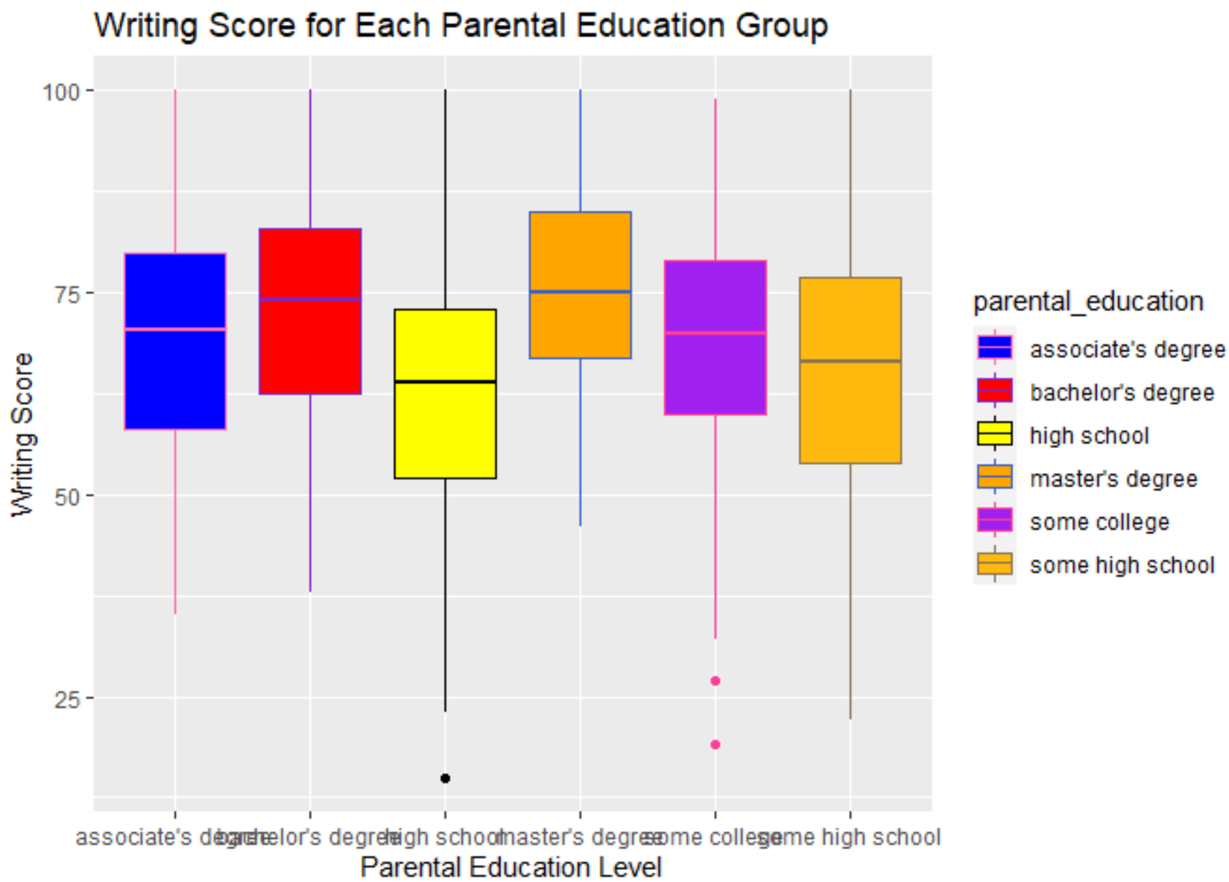

```
ggplot(student, aes(ethnicity, math_score, fill = ethnicity, color = ethnicity))+g
```



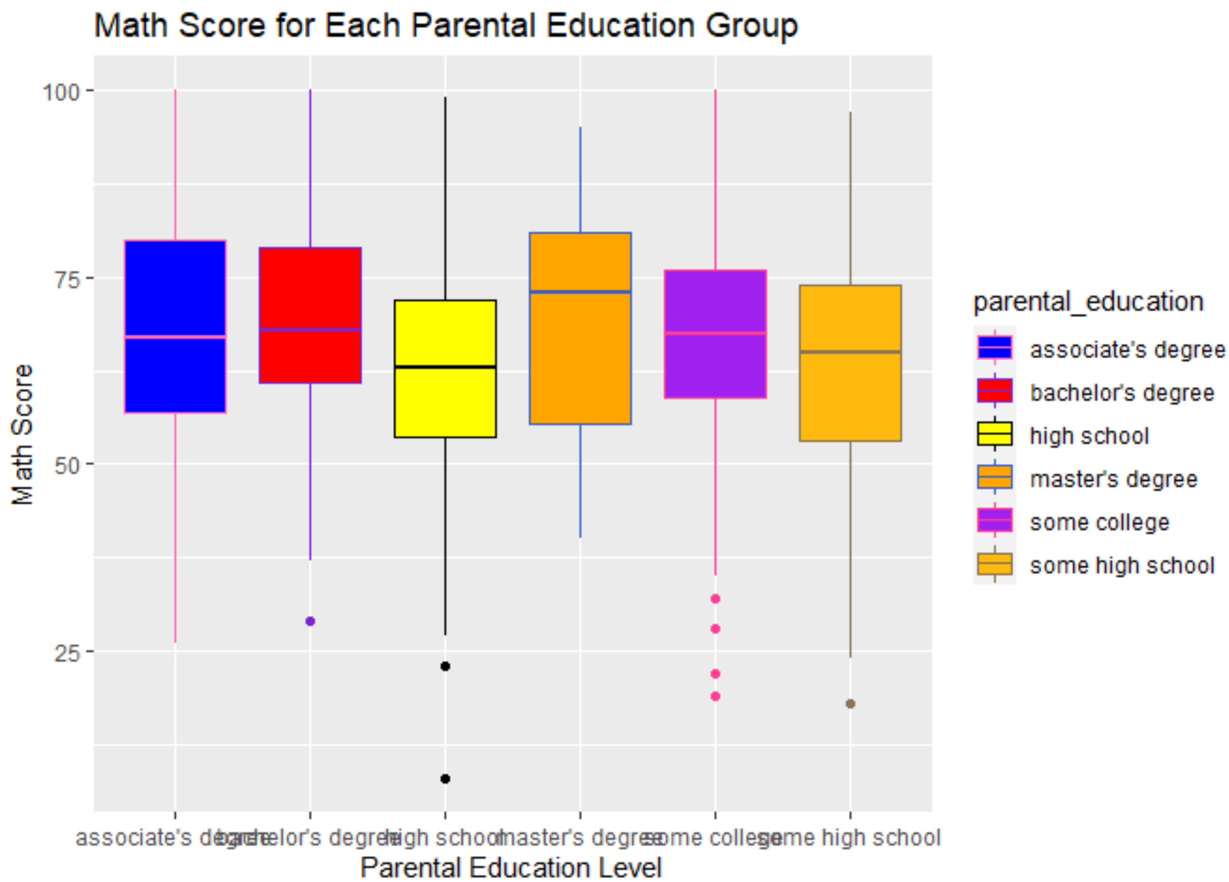

```
ggplot(student, aes(parental_education, reading_score, fill = parental_education,
```



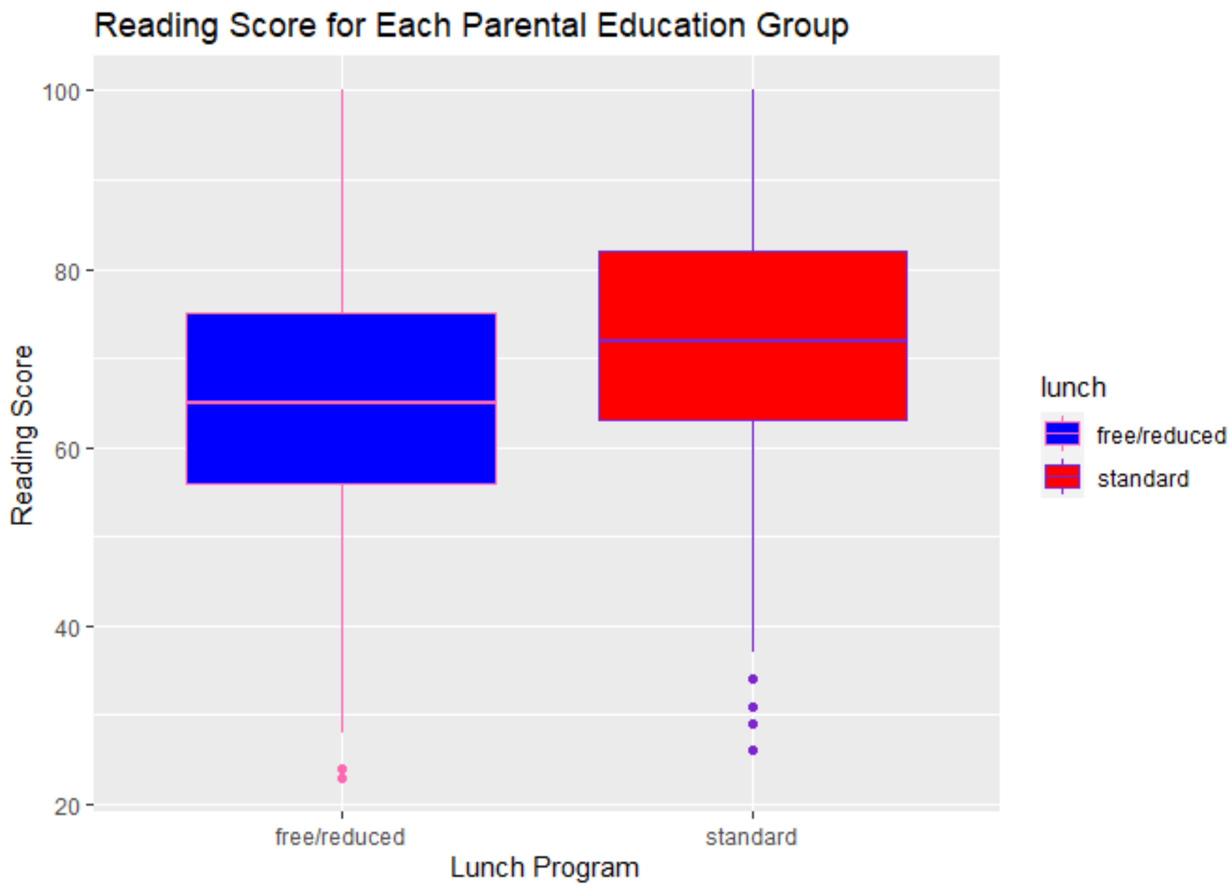

```
ggplot(student, aes(parental_education, writing_score, fill = parental_education,
```



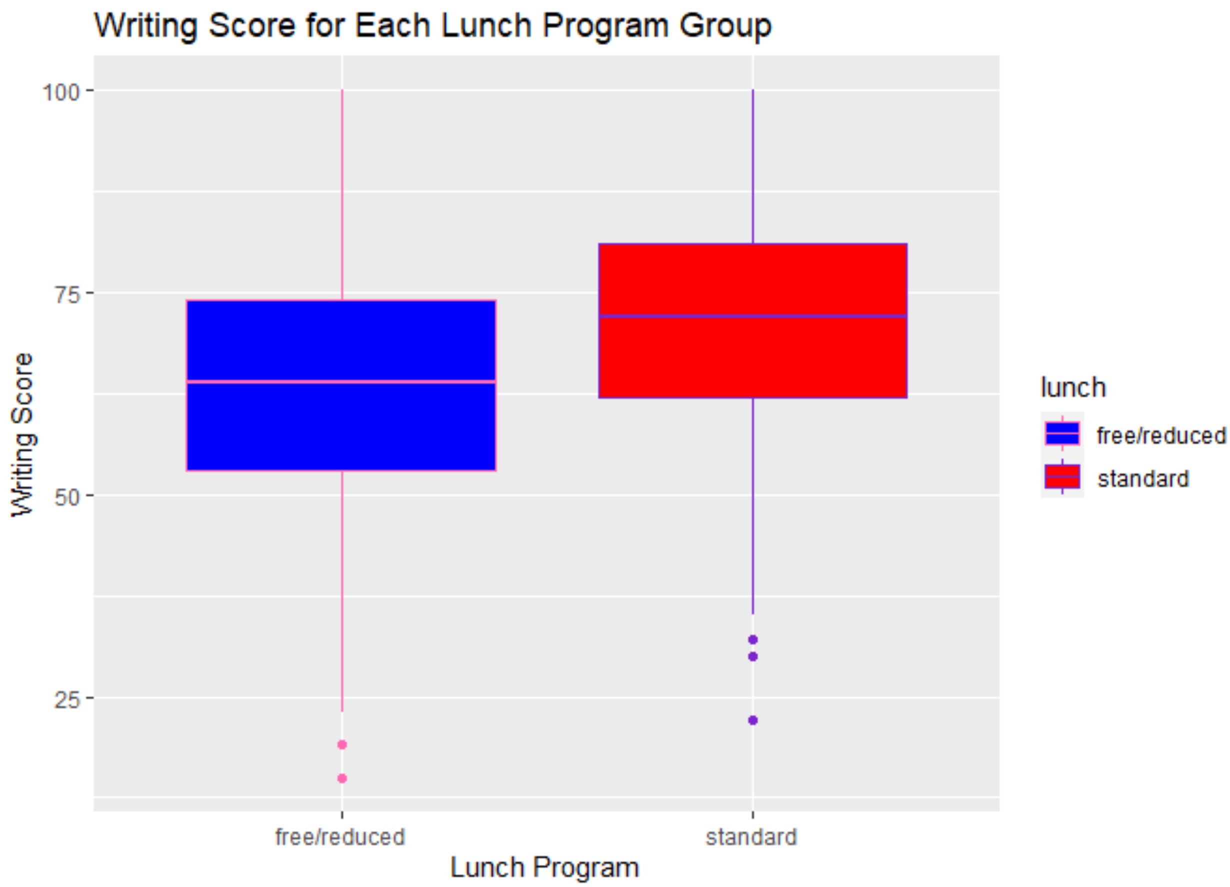

```
ggplot(student, aes(parental_education, math_score, fill = parental_education, col
```



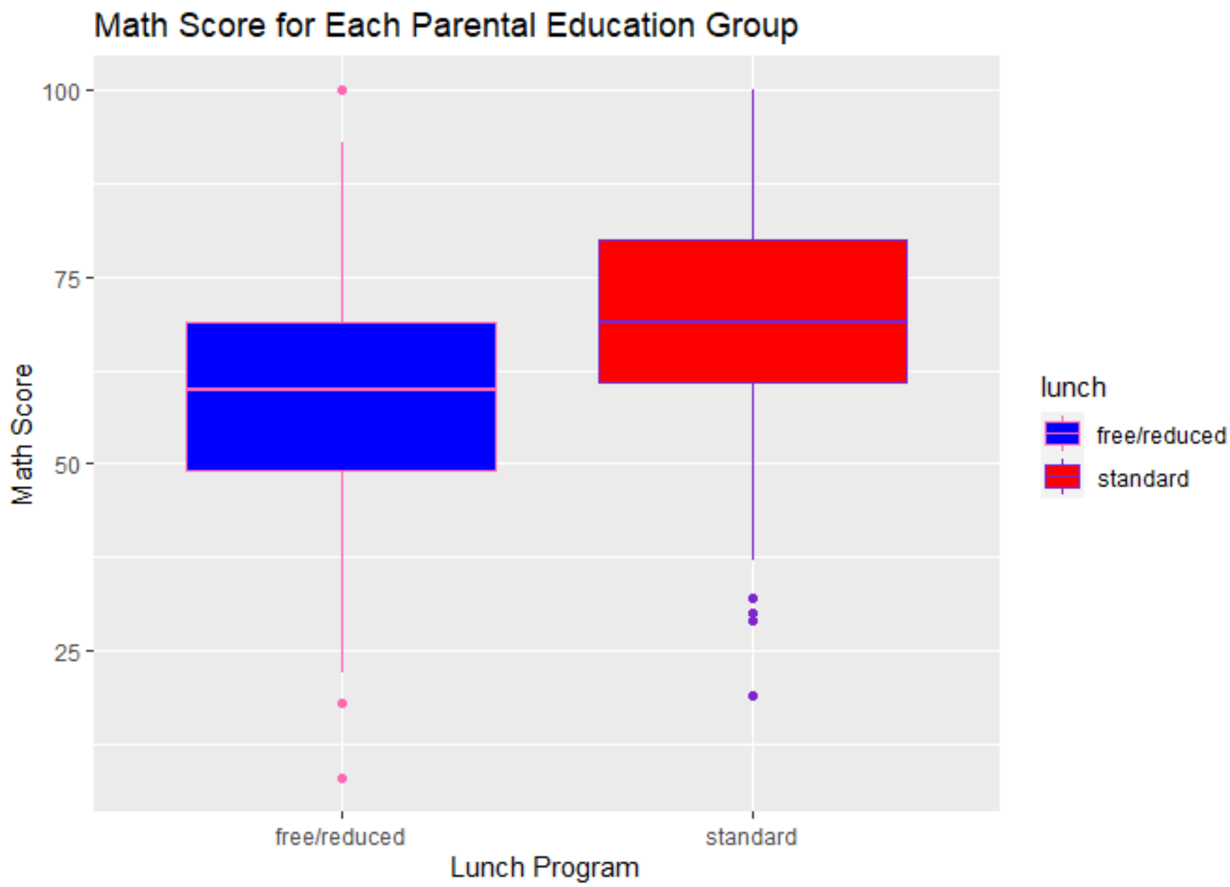

```
ggplot(student, aes(lunch, reading_score, fill = lunch, color = lunch))+geom_boxpl
```



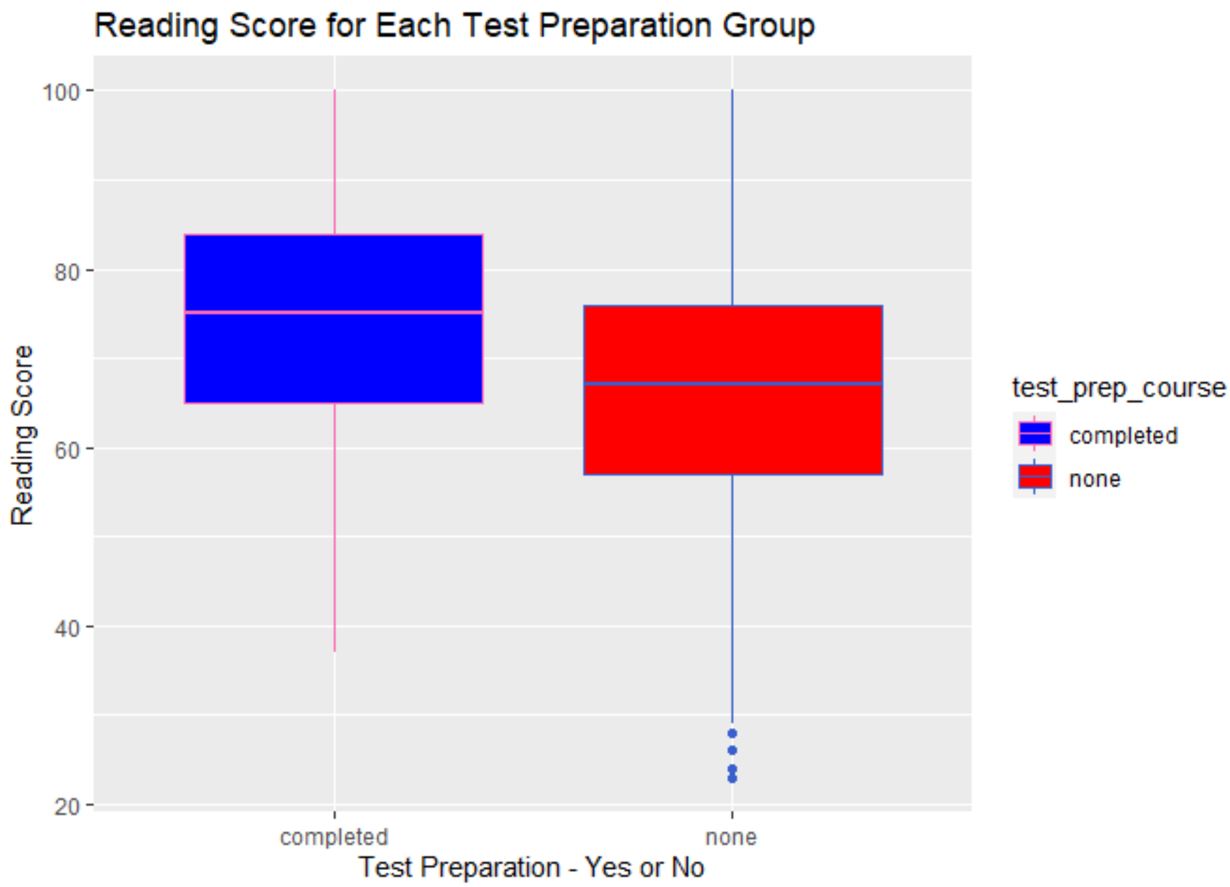

```
ggplot(student, aes(lunch, writing_score, fill = lunch, color = lunch))+geom_boxpl
```



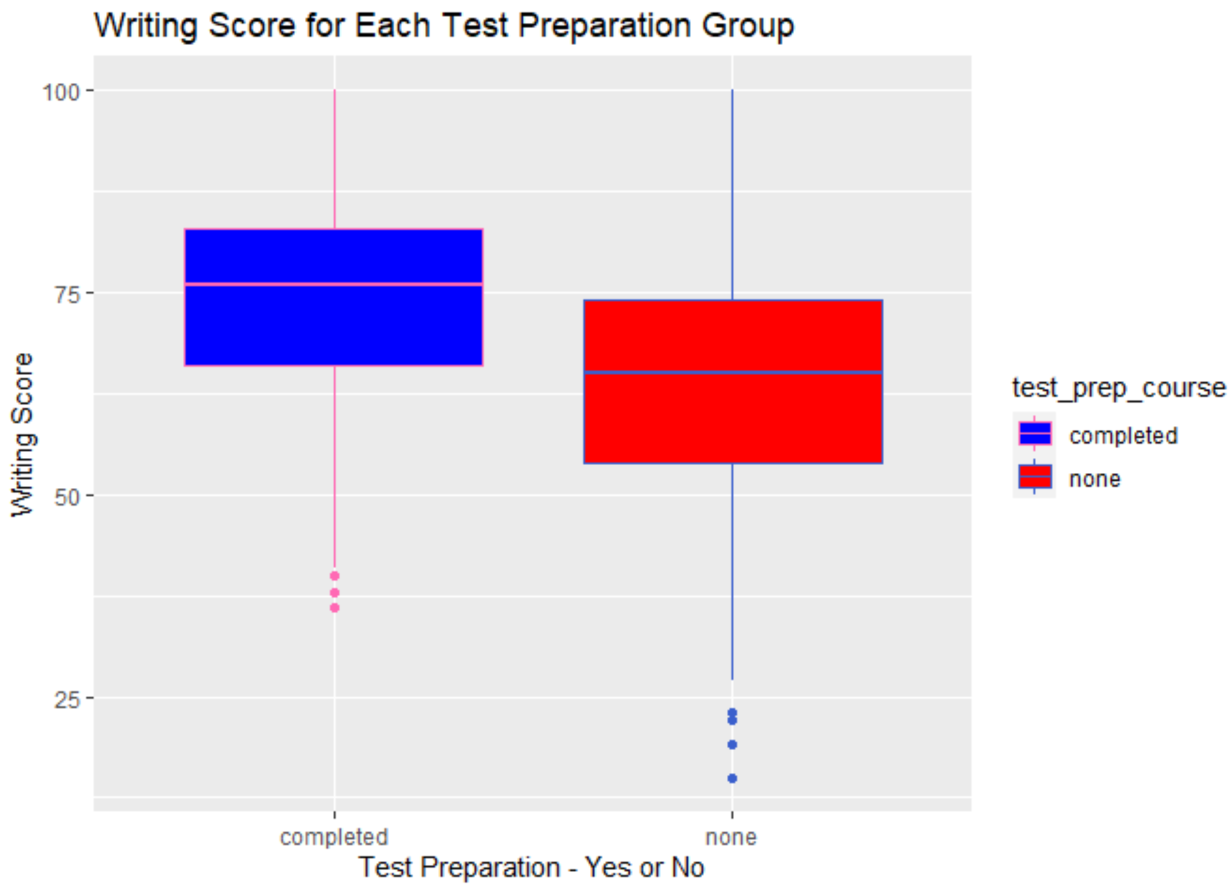

```
ggplot(student, aes(lunch, math_score, fill = lunch, color = lunch))+geom_boxplot(
```



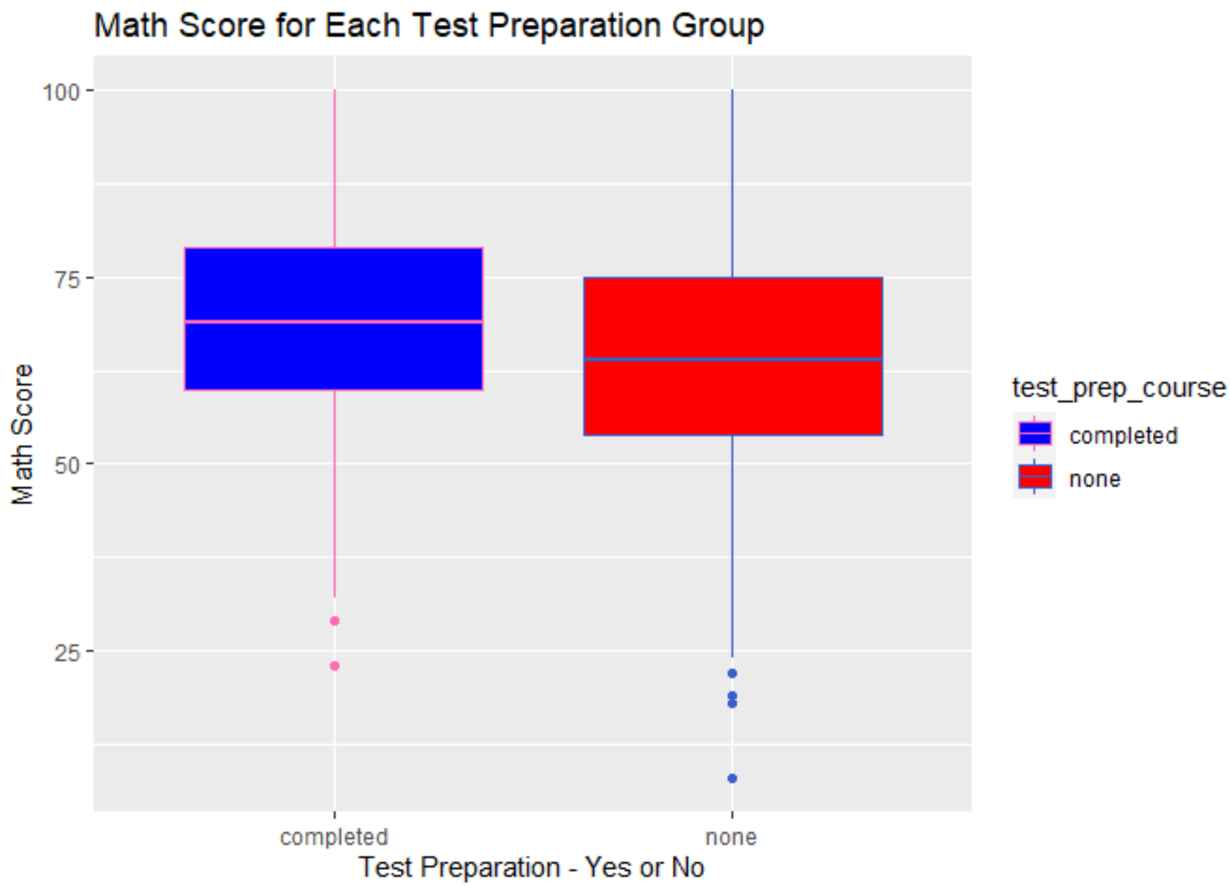

```
ggplot(student, aes(test_prep_course, reading_score, fill = test_prep_course, colo
```




```
ggplot(student, aes(test_prep_course, writing_score, fill = test_prep_course, colo
```




```
ggplot(student, aes(test_prep_course, math_score, fill = test_prep_course, color =
```



In each boxplot, there appears to be a difference in means between each group as related to reading, writing, and math test scores.

Mean and Standard Deviation Table for Each Variable

A table is also created displaying the means and standard deviation for each group's reading, writing, and math test scores.

```
gender_table <- student%>%group_by(gender)%>%summarize(reading_mean = mean(reading
```

```
gender_table
```

```
## # A tibble: 2 × 7
##   gender reading_mean writing_mean math_mean reading_sd writing_sd sd_math
##   <chr>         <dbl>        <dbl>     <dbl>      <dbl>      <dbl>   <dbl>
## 1 female         72.7         72.6      63.8       14.2       14.6    15.3
## 2 male           65.5         63.3      68.7       13.9       14.1    14.4
```

```
#Find the mean and standard deviation of the reading, writing, and math scores for
ethnic_table <- student%>%group_by(ethnicity)%>%summarize(reading_mean = mean(read

ethnic_table
```

```
## # A tibble: 5 × 7
##   ethnicity reading_mean writing_mean math_mean reading_sd writing_sd sd_math
##   <chr>            <dbl>        <dbl>     <dbl>      <dbl>      <dbl>   <dbl>
## 1 group A           64.7         62.7      61.6       15.5       15.5    14.5
## 2 group B           67.4         65.6      63.5       15.2       15.6    15.5
## 3 group C           69.3         68.0      64.7       13.7       14.7    14.4
## 4 group D           70.0         70.1      67.4       13.9       14.4    13.8
## 5 group E           73.0         71.4      73.8       14.9       15.1    15.5
```

```
education_table <- student%>%group_by(parental_education)%>%summarize(reading_mean

education_table
```

```
## # A tibble: 6 × 7
##   parental_education reading_mean writing_mean math_mean readi…¹ writi…² sd_mat
##   <chr>                     <dbl>        <dbl>     <dbl>   <dbl>   <dbl>   <dbl
## 1 associate's degree         70.9         69.9      67.9    13.9    14.3    15.
## 2 bachelor's degree          73           73.4      69.4    14.3    14.7    14.
## 3 high school                64.7         62.4      62.1    14.1    14.1    14.
## 4 master's degree            75.4         75.7      69.7    13.8    13.7    15.
## 5 some college               69.5         68.8      67.1    14.1    15.0    14.
## 6 some high school           67.2         65.2      63.9    15.1    15.2    15.
## # … with abbreviated variable names ¹reading_sd, ²writing_sd
```

```
lunch_table <- student%>%group_by(lunch)%>%summarize(reading_mean = mean(reading_s

lunch_table
```

```
## # A tibble: 2 × 7
##   lunch        reading_mean writing_mean math_mean reading_sd writing_sd sd_mat
##   <chr>               <dbl>        <dbl>     <dbl>      <dbl>      <dbl>   <dbl
## 1 free/reduced         64.8         63.2      59.1       14.7       15.2    14.
## 2 standard             71.7         70.8      70.0       13.8       14.3    13.
```

```
test_prep_table <- student%>%group_by(test_prep_course)%>%summarize(reading_mean =
```

```
test_prep_table
```

```
## # A tibble: 2 × 7
##   test_prep_course reading_mean writing_mean math_mean reading…¹ writi…² sd_mat
##   <chr>                   <dbl>        <dbl>     <dbl>     <dbl>   <dbl>   <dbl
## 1 completed                73.9         74.4      69.7      13.6    13.4    14.
## 2 none                     66.6         64.6      64.2      14.3    14.9    15.
## # … with abbreviated variable names ¹reading_sd, ²writing_sd
```

Sometimes we want to see the results of those who fit in multiple groups. Another table is created to show the mean and standard deviation of each gender, further grouped by each gender's race/ethnicity.

```
ethnic_gender_table <- student%>%group_by(ethnicity, gender)%>%summarize(reading_m
```

```
## `summarise()` has grouped output by 'ethnicity'. You can override using the
## `.groups` argument.
```

```
ethnic_gender_table
```

```
## # A tibble: 10 × 8
## # Groups:   ethnicity [5]
##    ethnicity gender reading_mean writing_mean math_mean readin…¹ writi…² sd_mat
##    <chr>     <chr>         <dbl>        <dbl>     <dbl>    <dbl>   <dbl>   <dbl
##  1 group A   female         69           67.9      58.5     14.8    14.7    14.
##  2 group A   male           61.7         59.2      63.7     15.5    15.1    14.
##  3 group B   female         71.1         70.0      61.4     14.6    14.9    16.
##  4 group B   male           62.8         60.2      65.9     14.7    14.9    14.
##  5 group C   female         72.3         72.1      62.4     13.3    14.1    14.
##  6 group C   male           65.4         62.7      67.6     13.3    13.6    14.
##  7 group D   female         74.0         75.0      65.2     13.9    13.9    14.
##  8 group D   male           66.1         65.4      69.4     12.8    13.3    13.
##  9 group E   female         75.8         75.5      70.8     15.3    15.7    16.
## 10 group E   male           70.3         67.4      76.7     14.0    13.4    14.
## # … with abbreviated variable names ¹reading_sd, ²writing_sd
```

The same classification can be done to see the mean and standard deviation of the test results for each combination of test preparation and lunch program groupings.

```
test_lunch_table <- student%>%group_by(test_prep_course, lunch)%>%summarize(readin
```

```
## `summarise()` has grouped output by 'test_prep_course'. You can override using
## the `.groups` argument.
```

```
test_lunch_table
```

```
## # A tibble: 4 × 8
## # Groups:   test_prep_course [2]
##   test_prep_course lunch        readin…¹ writi…² math_…³ readi…⁴ writi…⁵ sd_mat
##   <chr>            <chr>           <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl
## 1 completed        free/reduced     69.9    70.4    63.0    14.2    13.9    14.
## 2 completed        standard         76.2    76.8    73.5    12.8    12.5    13.
## 3 none             free/reduced     61.8    59.0    56.8    14.2    14.4    14.
## 4 none             standard         69.2    67.6    68.1    13.8    14.2    13.
## # … with abbreviated variable names ¹reading_mean, ²writing_mean, ³math_mean,
## #   ⁴reading_sd, ⁵writing_sd
```

From the tables printed, the means seem to differ in each group. The large standard deviations in the tables means the test scores are very spread out, hinting at large variance in the data. The similar standard deviations in each table means we can use t-tests and ANOVA to see if there is a statistical difference in the means.

T-Testing

T-testing is effective when comparing the means of two groups. Multiple t-tests can be run to see if there are differences in the testing score means between males and females. To keep the session short, only one t-test is done to demonstrate how a t-test is interpreted.

In the code below, a t-test is done to see if there is a difference in average reading scores between male and female students.

```
t.test(reading_score ~ gender, data = student)
```

```
##
##  Welch Two Sample t-test
##
## data:  reading_score by gender
## t = 8.1397, df = 994.27, p-value = 1.177e-15
## alternative hypothesis: true difference in means between group female and group
## 95 percent confidence interval:
##  5.496561 8.988715
## sample estimates:
## mean in group female   mean in group male
##             72.71567             65.47303
```

From the results, we want to see the p-value. The general idea is that a p-value less than 0.05 means the result is statistically significant, and we can conclude there is a difference in average reading scores between male and female students. From above, we see there is a p-value of 1.49e-08. Since this is less than 0.05, we can conclude there is a difference in the average reading scores of male and female students.

ANOVA Testing

ANOVA testing is used to check if there is a difference in means among more than two groups. For example, from the boxplot it appears there is a difference in means between the groups in the highest parent education variable. To compare the average reading score in this variable, an ANOVA test is run:

```
parent_aov <- aov(reading_score ~ parental_education, data = student)

summary(parent_aov)
```

```
##                     Df Sum Sq Mean Sq F value   Pr(>F)
## parental_education   5   9290  1858.0   9.182 1.49e-08 ***
## Residuals          993 200938   202.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the summary, the p-value is 1.49e-08. Since this is less than 0.05, we can conclude there is a difference in the average reading scores between the parent education groups.

Simple Linear Regression

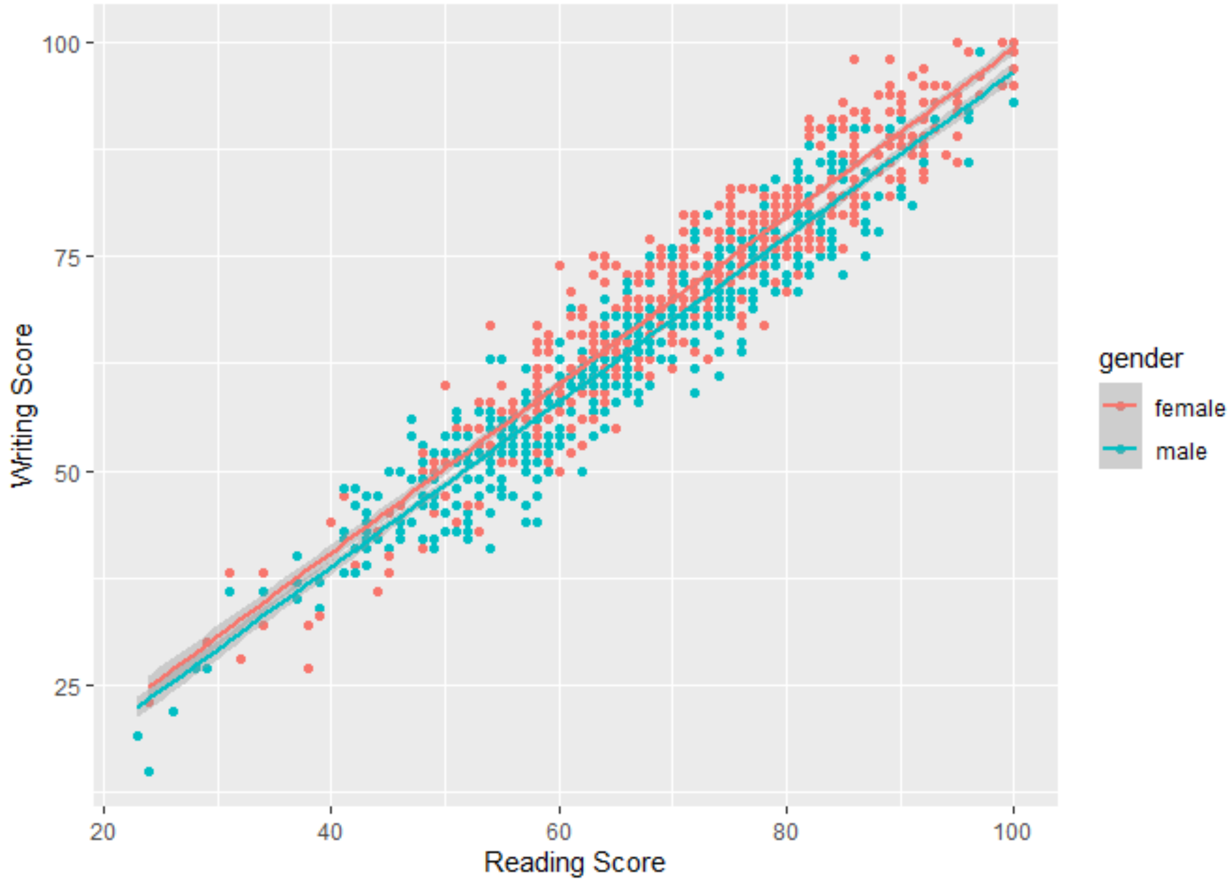
Linear regression is used to see if there is a relationship between the variables. Normally, linear regression has a dependent variable (y) that can be predicted based on the explanatory variable (x). Linear regression is simple to implement and on the easier statistical methods to interpret, and will be used in this session.

In this session, simple linear regression is used to see if there is a relationship between two variables. The formula for simple linear regression is $y' = a + bx$, where 'a' is the intercept value (value of y when x = 0) and 'b' is the slope of the line. A positive value for 'b' shows a positive means there's a positive relationship between the variables. In other words, when 'x' increases, so does 'y'. Also, if 'x' decreases, 'y' decreases as well. A negative value for 'b' means there's a negative relationship between the variables. As 'x' increases, 'y' decreases. When 'x' decreases, 'y' will increase.

The simple linear regression I will use will see if there is a relationship between reading score and writing score for each gender.

```
ggplot(student, aes(x = reading_score, y = writing_score, color = gender)) + geom_
```

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




```r
male_fit_read_write <- lm(writing_score ~ reading_score, subset = gender == "male"

summary(male_fit_read_write)
```

```
##
## Call:
## lm(formula = writing_score ~ reading_score, data = student, subset = gender ==
##     "male")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1143  -3.1530  -0.0046   2.9987  11.3661
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.25772    0.96052   0.268    0.789
## reading_score  0.96305    0.01435  67.112   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.385 on 480 degrees of freedom
## Multiple R-squared:  0.9037, Adjusted R-squared:  0.9035
```

```
## F-statistic:  4504 on 1 and 480 DF,  p-value: < 2.2e-16
```

```
female_fit_read_write <- lm(writing_score ~ reading_score, subset = gender == "fem

summary(female_fit_read_write)
```

```
##
## Call:
## lm(formula = writing_score ~ reading_score, data = student, subset = gender ==
##     "female")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5112  -2.8671   0.0409   2.9673  13.8937
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.21042    1.01447   1.193    0.233
## reading_score  0.98160    0.01369  71.683   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.411 on 515 degrees of freedom
## Multiple R-squared:  0.9089, Adjusted R-squared:  0.9087
## F-statistic:  5138 on 1 and 515 DF,  p-value: < 2.2e-16
```

A best fit line, or regression line, is drawn for each gender to show the slope of the formula for each gender. The best fit line also shows the point where the distance between an observed point and the line is minimized. Linear regression is best used for continuous data, or data with numbers that dare measured (time, distance, weight). Exam scores can only be from 0 to 100 and would be considered discrete data (countable data). This will mean we will have “non-integer” number when looking at the linear regression equation. Linear regression can still be used for exam scores, but the values in the formula will not be whole numbers.

For example, from the summary it is seen the formula for female students is writing score = 1.21042 + (0.98160) * reading score. So, a female student with a reading score of 50 is predicted to have a writing score of 50.29. A student can’t actually score a 50.29 on the exam, it can be interpreted a student with a 50 on the reading score will score around a 50 on the writing score.

From the summary, we can also get the R^2 value. R^2 is known as the coefficient of determination, which is used to describe proportion of variance the outcome Y is explained by the regression model. Another way to describe R^2 is it **measures how well the linear regression model fits the data**. The linear regression model for females has an adjusted R^2 value = 0.9087, which means 90.87% of the variance can be explained by the model.

The p-value also tells us if the relationship highlighted is statistically significant. The linear regression models above gives a p-value less than 2.2e-16 for both genders, which is significantly less than 0.05. It can be concluded there is a relationship between reading scores and writing scores for both genders.

After creating a simple linear regression model to see the relationship between the writing score and reading score for each gender, another model is created to see the relationship between reading score and math score for each gender:

```
ggplot(student, aes(x = reading_score, y = math_score, color = gender))+geom_point
```

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


```
male_fit_math_read <- lm(math_score ~ reading_score, subset = gender == "male", da

summary(male_fit_math_read)
```

```
##
## Call:
## lm(formula = math_score ~ reading_score, data = student, subset = gender ==
##     "male")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.0844  -4.7250  -0.2276   3.9660  18.4037
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.98630    1.46288   6.143  1.7e-09 ***
## reading_score  0.91247    0.02185  41.751  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.678 on 480 degrees of freedom
## Multiple R-squared:  0.7841, Adjusted R-squared:  0.7836
```

```
## F-statistic:  1743 on 1 and 480 DF,  p-value: < 2.2e-16
```

```
female_math_read <- lm(math_score ~ reading_score, subset = gender == "female", da
summary(female_math_read)
```

```
##
## Call:
## lm(formula = math_score ~ reading_score, data = student, subset = gender ==
##     "female")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.0334  -3.9071  -0.0082   4.2445  17.8402
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.12226    1.48272  -4.804 2.05e-06 ***
## reading_score  0.97474    0.02001  48.702  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.448 on 515 degrees of freedom
## Multiple R-squared:  0.8216, Adjusted R-squared:  0.8213
## F-statistic:  2372 on 1 and 515 DF,  p-value: < 2.2e-16
```

Looking at the female students linear regression model, the formula is math score = 8.9863 + (0.91247) * reading score. The R^2 value is 0.8213, so 82.13% of the variance can be explained by the model. The p-value is also less than 2.2e-16 which is less than 0.05, so it can be concluded there is a relationship between math scores and reading scores for female students.

Another model is created to see the relationship between math score and writing score for each gender:

```
ggplot(student, aes(x = writing_score, y = math_score, color = gender))+geom_point
```

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


```
male_fit_math_write <- lm(math_score ~ writing_score, subset = gender == "male", d

summary(male_fit_math_write)
```

```
##
## Call:
## lm(formula = math_score ~ writing_score, data = student, subset = gender ==
##     "male")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.3729  -4.1146  -0.1411   4.5408  18.2958
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.28627    1.36142    8.29 1.14e-15 ***
## writing_score  0.90730    0.02099   43.23  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.497 on 480 degrees of freedom
## Multiple R-squared:  0.7956, Adjusted R-squared:  0.7952
```

```
## F-statistic:  1869 on 1 and 480 DF,  p-value: < 2.2e-16
```

```
female_math_write <- lm(math_score ~ writing_score, subset = gender == "female", d

summary(female_math_write)
```

```
##
## Call:
## lm(formula = math_score ~ writing_score, data = student, subset = gender ==
##     "female")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.9040  -3.8215   0.2304   4.0243  18.5495
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -5.83959    1.35143  -4.321 1.86e-05 ***
## writing_score  0.95878    0.01825  52.528  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.054 on 515 degrees of freedom
## Multiple R-squared:  0.8427, Adjusted R-squared:  0.8424
## F-statistic:  2759 on 1 and 515 DF,  p-value: < 2.2e-16
```

Looking at the female students linear regression model, the formula is math score = -5.83959 + (0.95878) * writing score. The R^2 value is 0.8424, so 84.24% of the variance can be explained by the model. The p-value is also less than 2e-16 which is less than 0.05, so it can be concluded there is a relationship between math scores and reading scores for female students.

Predict Exam Scores

Since it's been concluded there is a relationship between the test scores for each gender, a prediction model is created to see if a student's test score can be predicted based on another test score and another variable.

Factorization

Before setting up the prediction model, the categorical variables must be set up as factors. This way, categorical variables are converted into numeric variables and can be used in the prediction model.

```
student_factor <- as.factor(student)
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
print(is.factor(student_factor))
```

```
## [1] TRUE
```

Now the prediction models can be created in R.

Exam Score Prediction Models

One prediction model that can be used is the predict function in R.

```
#Multiple linear regression model with writing score as the dependent variable and
test_fit_writing_read <- lm(writing_score ~ gender + reading_score + gender + test_
#Prints the summary of the linear regression model.
summary(test_fit_writing_read)
```

```
##
## Call:
## lm(formula = writing_score ~ gender + reading_score + gender +
##     test_prep_course, data = student)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3174  -3.0413  -0.1605   2.7416  11.5685
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.568402   0.795523   7.000 4.71e-12 ***
## gendermale           -2.427810   0.274119  -8.857  < 2e-16 ***
## reading_score         0.947719   0.009728  97.419  < 2e-16 ***
## test_prep_coursenone -2.941145   0.285000 -10.320  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.182 on 995 degrees of freedom
## Multiple R-squared:  0.9234, Adjusted R-squared:  0.9232
## F-statistic:  4000 on 3 and 995 DF,  p-value: < 2.2e-16
```

```
#Prediction model using the linear regression model.
predict_test_writing_read <- predict(test_fit_writing_read, interval = "prediction
```

```
## Warning in predict.lm(test_fit_writing_read, interval = "prediction"): predicti
```

```
#Data frame comparing the prediction model's writing score to the actual writing s
actual_pred_test_1 <- data.frame(cbind(prediction = predict_test_writing_read, act
#Lists the first 10 prediction's fit prediction and confidence interval (lwr and u
head(actual_pred_test_1, 10)
```

```
##         fit      lwr       upr actual
## 1  70.86300 62.64554  79.08047     74
## 2  90.86309 82.63706  99.08911     88
## 3  92.66054 84.42943 100.89164     93
## 4  54.21941 46.00064  62.43819     44
## 5  74.12151 65.89838  82.34463     75
## 6  81.28791 73.06684  89.50898     78
## 7  95.60168 87.37232 103.83104     92
## 8  40.95135 32.72461  49.17809     39
## 9  63.79459 55.57077  72.01841     67
## 10 59.49038 51.27073  67.71003     50
```

```
#Views the correlation between the prediction model and the actual model.
cor(actual_pred_test_1)
```

```
##              fit       lwr       upr    actual
## fit    1.0000000 0.9999999 0.9999999 0.9609532
## lwr    0.9999999 1.0000000 0.9999997 0.9609581
## upr    0.9999999 0.9999997 1.0000000 0.9609482
## actual 0.9609532 0.9609581 0.9609482 1.0000000
```

From the results, the prediction model had relatively close values compared to the actual model’s writing score values. The high correlation values (all over 0.95) means the variables in the model have a strong relationship.

Another prediction that can be used is creating a training data set and a testing data set. A training data set is a subset of examples used to train the model, while the testing data set is a subset used to test the training model.

```
#Initializes number generator.
set.seed(123)
#New sample created for the training and testing data sets. The data is split with
sample <- sample(c(TRUE, FALSE), nrow(student), replace = TRUE, prob = c(0.75, 0.2
train <- student[sample, ]
test <- student[!sample, ]
```

Next, the same linear regression model is set up using the training and testing data sets.

```
#Multiple linear regression model with writing score as the dependent variable and
lm_practice <- lm(formula = writing_score ~ gender + reading_score + test_prep_cou
#Prediction model using the linear regression model.
lm_predict <- predict(lm_practice, newdata = test)
#Data frame comparing the prediction model's writing score to the actual writing s
actual_pred_1 <- data.frame(cbind(prediction = lm_predict, actual = test$writing_s
#Lists 10 predictions of the fit prediction model and compares the values to the a
head(actual_pred_1, 10)
```

```
##    prediction actual
## 2    90.52266     88
## 4    54.29783     44
## 5    74.03495     75
## 8    41.13975     39
## 11   51.47824     52
## 16   73.70415     78
## 20   57.72648     61
## 21   65.57618     63
## 24   71.82442     73
## 31   72.76429     74
```

```
#Views the correlation between the prediction model and the actual model.
cor(actual_pred_1)
```

```
##            prediction   actual
## prediction   1.000000 0.966545
## actual       0.966545 1.000000
```

From the results, the prediction model had relatively close values compared to the actual model’s writing score values. The high correlation value (over 0.95) means the variables in the model have a strong relationship.

Conclusion

In this session, exploratory data analysis techniques, linear regression and prediction modeling were demonstrated using the fictional “Students Performance in Exams” dataset from Kaggle.

First, the dataset was inspected and checked for missing and duplicated values. A frequency table was created to view the sample size of each variable. The data was also checked for outliers, with one outlier chosen to be removed from the dataset. The distribution was viewed and boxplots were created to see the IQR, mean, and outliers in the data.

Next, a mean and standard deviation table were created to compare the testing score means and standard deviations of each group. Since there appears to be differences in the means for each group, a t-test and ANOVA test are performed to confirm. From the t-test and ANOVA test run, it is concluded there are differences in the reading score means for gender and parent highest education. More t-tests and ANOVA tests can be run to see further differences in means, but only these two were completed since this session is used for demonstration purposes.

A linear regression model was completed to demonstrate the relationship between the various testing scores. A linear regression model was done for both male and female students to see the linear regression formula and variance for each gender.

A prediction model was created in two different ways to see if test scores can be predicted based off of select variables. From both methods, it appears an accurate prediction model was created for predicting writing test scores. Based on this, prediction models can be created for reading and math test scores if desired.

This analysis was meant to demonstrate basic exploratory data analysis techniques as well as linear regression and prediction modeling. For a real world data set, I would delve deeper into the data using more complex statistical methods. Such statistical methods can include logistic regression, random forest modeling, and clustering.

Thank you to the reader for viewing my work.

END