Student marks analysis

Jarun ~ Mohana ~ Aravind

Importing all the datasets

```
library(tidyverse) # metapackage with lots of helpful functions
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
#install.packages("reshape2")
library(reshape2)
```

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

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

Importing the dataset

```
data = read.csv("studentdata.csv")
str(data)
```

```
## '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's degree" "associate
##  $ 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 ...
```

```r
print(unique(data$gender))
```

```
## [1] "female" "male"
```

```r
print(unique(data$race.ethnicity))
```

```
## [1] "group B" "group C" "group A" "group D" "group E"
```

```r
print(unique(data$parental.level.of.education))
```

```
## [1] "bachelor's degree"  "some college"       "master's degree"
## [4] "associate's degree" "high school"        "some high school"
```

```r
print(unique(data$lunch))
```

```
## [1] "standard"     "free/reduced"
```

```r
print(unique(data$test.preparation.course))
```

```
## [1] "none"      "completed"
```

```r
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
```

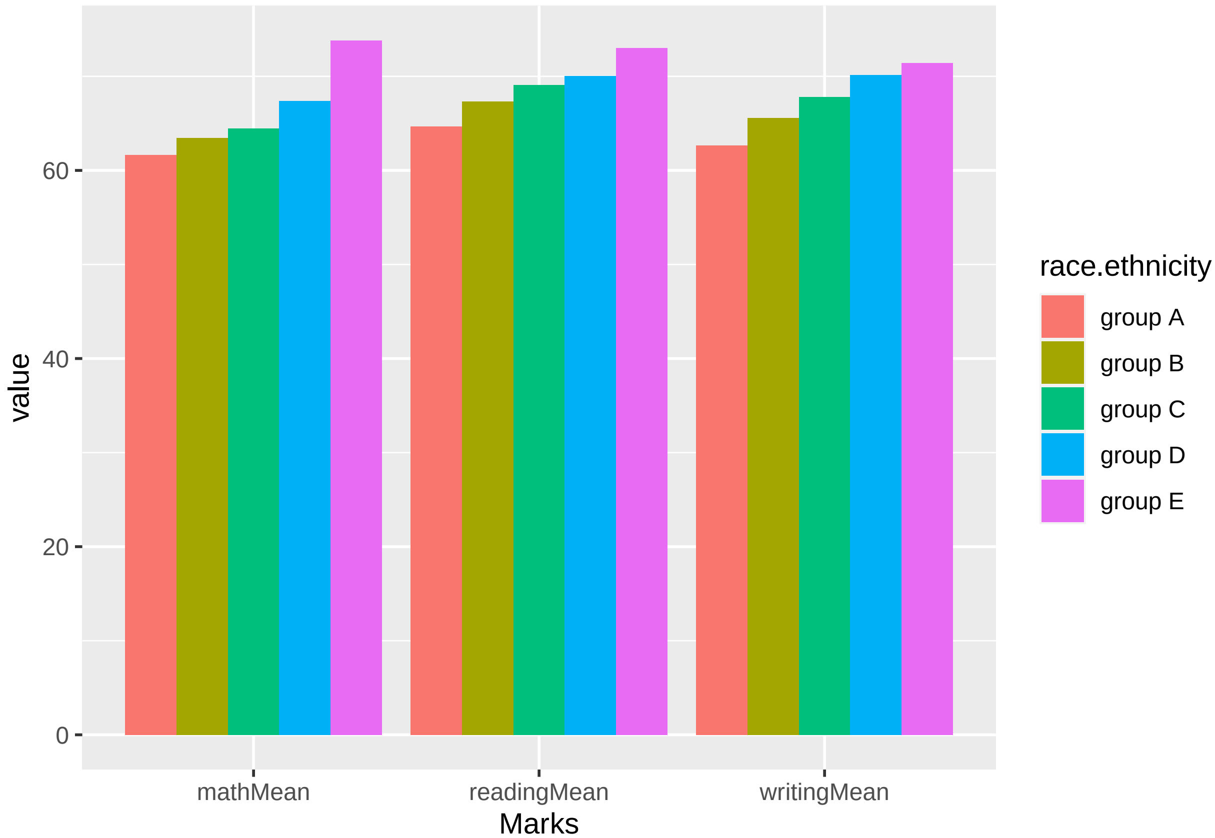
General Description

Affectors: 1) Gender 2) Race ethnicity 3) Parental level of education 4) Lunch 5) Test preparation course

Marks: 1) Math score 2) Reading score 3) Writing score

Analysis on Race ethinicity over students' marks

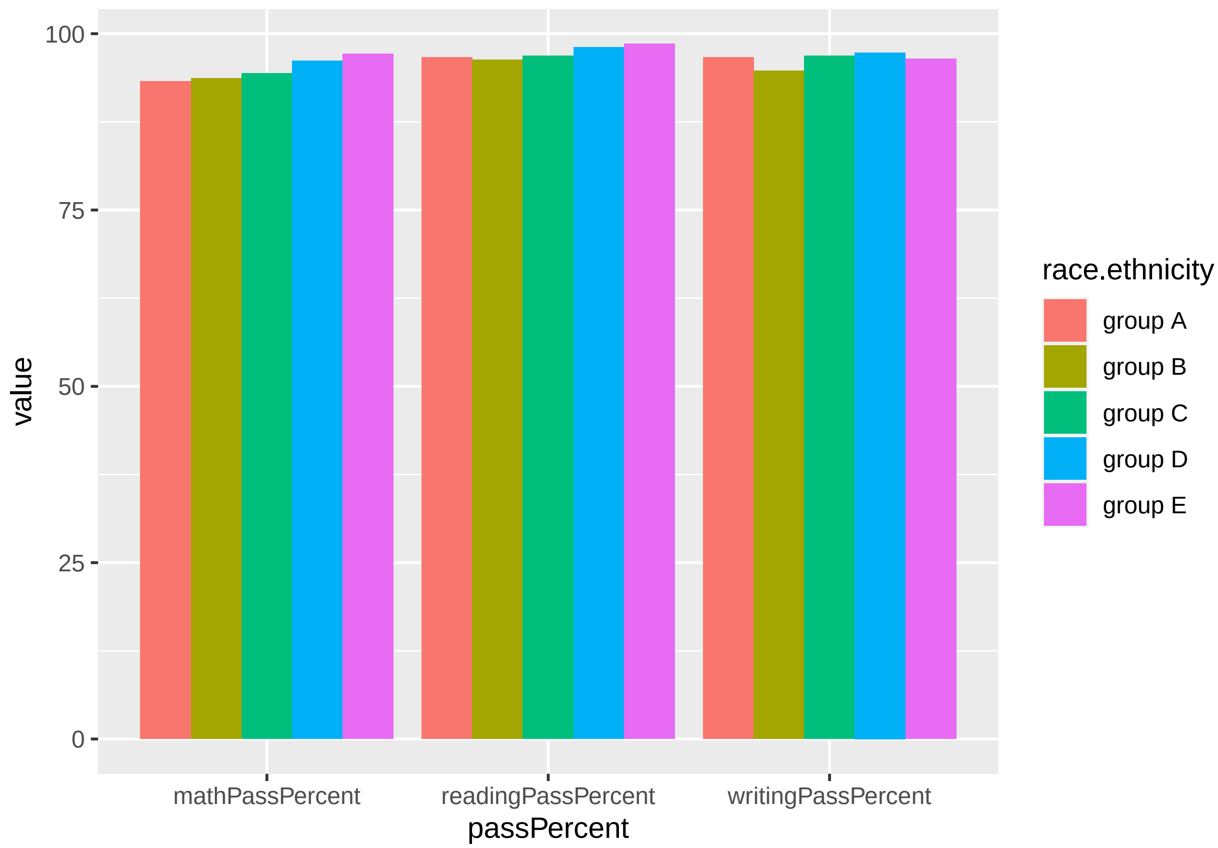
```
data %>%
    group_by(race.ethnicity) %>%
    summarise(mathMean = mean(math.score),readingMean =  mean(reading.score),writingMean = mean(writing
  melt(id.vars="race.ethnicity", variable.name = "Marks") %>%
  ggplot(aes(x=Marks, y=value, fill=race.ethnicity)) +
  geom_bar(stat="identity", position = position_dodge())
```



Result In general there are no large difference at all. Maybe group E is slighty better than others but generally, all the groups are fine in the total pool.

```
mathpass = data %>%
  group_by(race.ethnicity) %>%
  filter(math.score>40) %>%
  count(race.ethnicity)

writingpass = data %>%
  group_by(race.ethnicity) %>%
  filter(writing.score>40) %>%
  count(race.ethnicity)

readingpass = data %>%
  group_by(race.ethnicity) %>%
  filter(reading.score>40) %>%
  count(race.ethnicity)
```

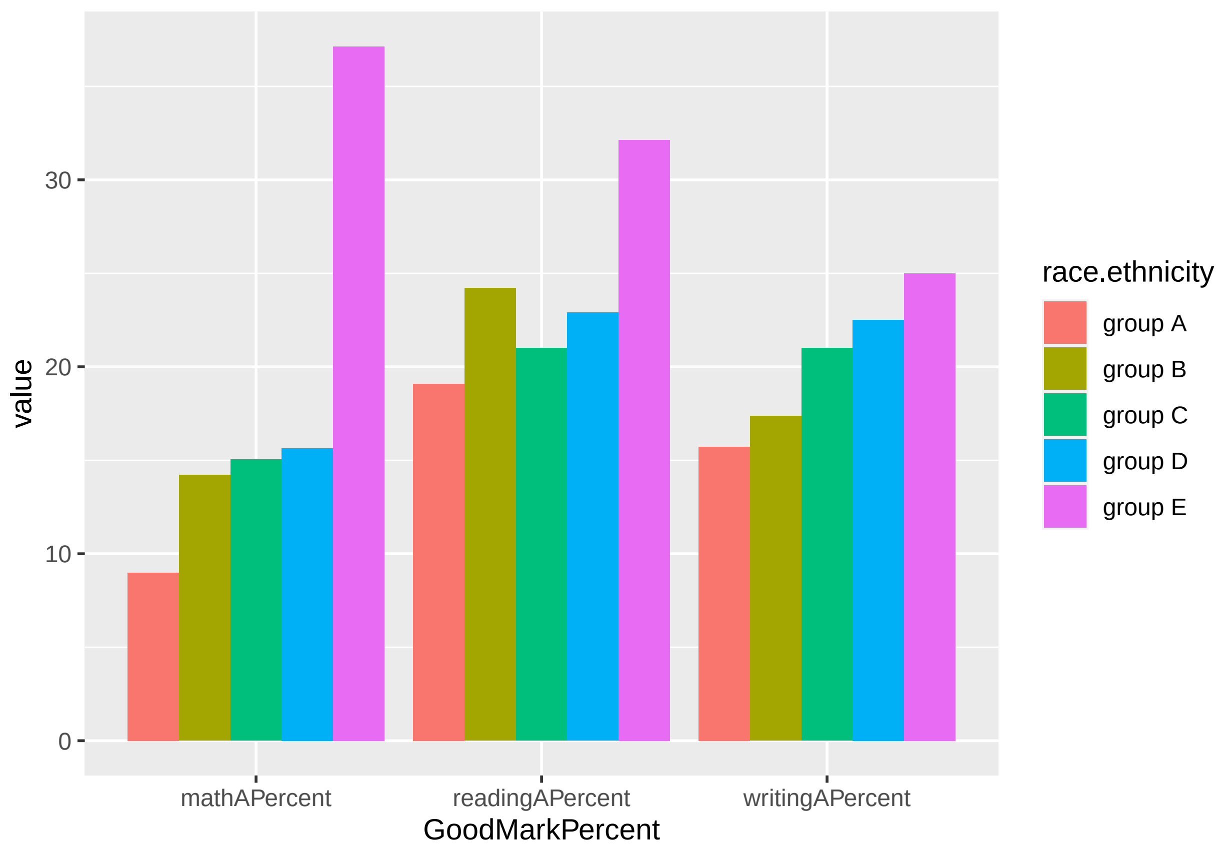
```r
total = data %>%
  group_by(race.ethnicity) %>%
  count(race.ethnicity)

total$mathPassPercent = mathpass$n / total$n * 100
total$readingPassPercent = readingpass$n / total$n * 100
total$writingPassPercent = writingpass$n / total$n * 100

total = subset(total, select = -c(n))

melt(total, id.vars="race.ethnicity", variable.name = "passPercent") %>%
  ggplot(aes(x=passPercent, y=value, fill=race.ethnicity)) +
  geom_bar(stat="identity", position = position_dodge())
```



Result In general there are no large difference at all. All the groups have good pass percentage.

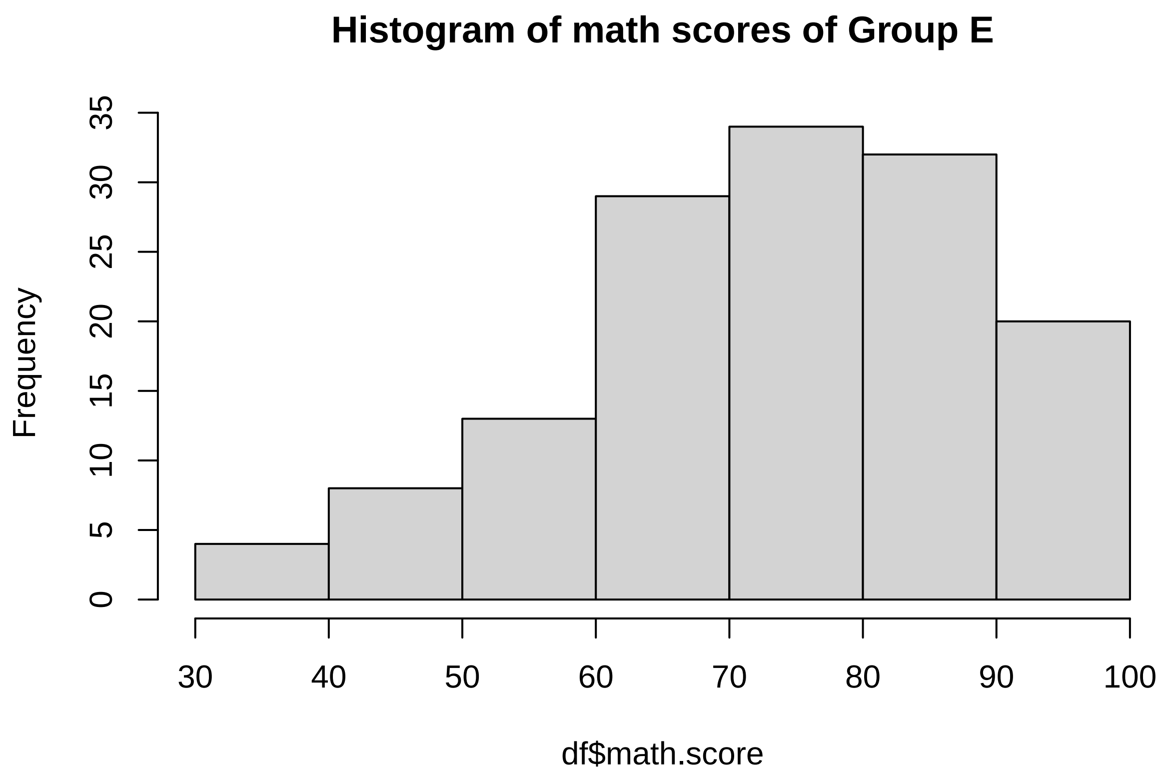
```r
math = data %>%
  group_by(race.ethnicity) %>%
  filter(math.score>80) %>%
  count(race.ethnicity)

writing = data %>%
  group_by(race.ethnicity) %>%
  filter(writing.score>80) %>%
```

```r
  count(race.ethnicity)

reading = data %>%
  group_by(race.ethnicity) %>%
  filter(reading.score>80) %>%
  count(race.ethnicity)

total = data %>%
  group_by(race.ethnicity) %>%
  count(race.ethnicity)

total$mathAPercent = math$n / total$n * 100
total$readingAPercent = reading$n / total$n * 100
total$writingAPercent = writing$n / total$n * 100

total = subset(total, select = -c(n))

melt(total, id.vars="race.ethnicity", variable.name = "GoodMarkPercent") %>%
  ggplot(aes(x=GoodMarkPercent, y=value, fill=race.ethnicity)) +
  geom_bar(stat="identity", position = position_dodge())
```

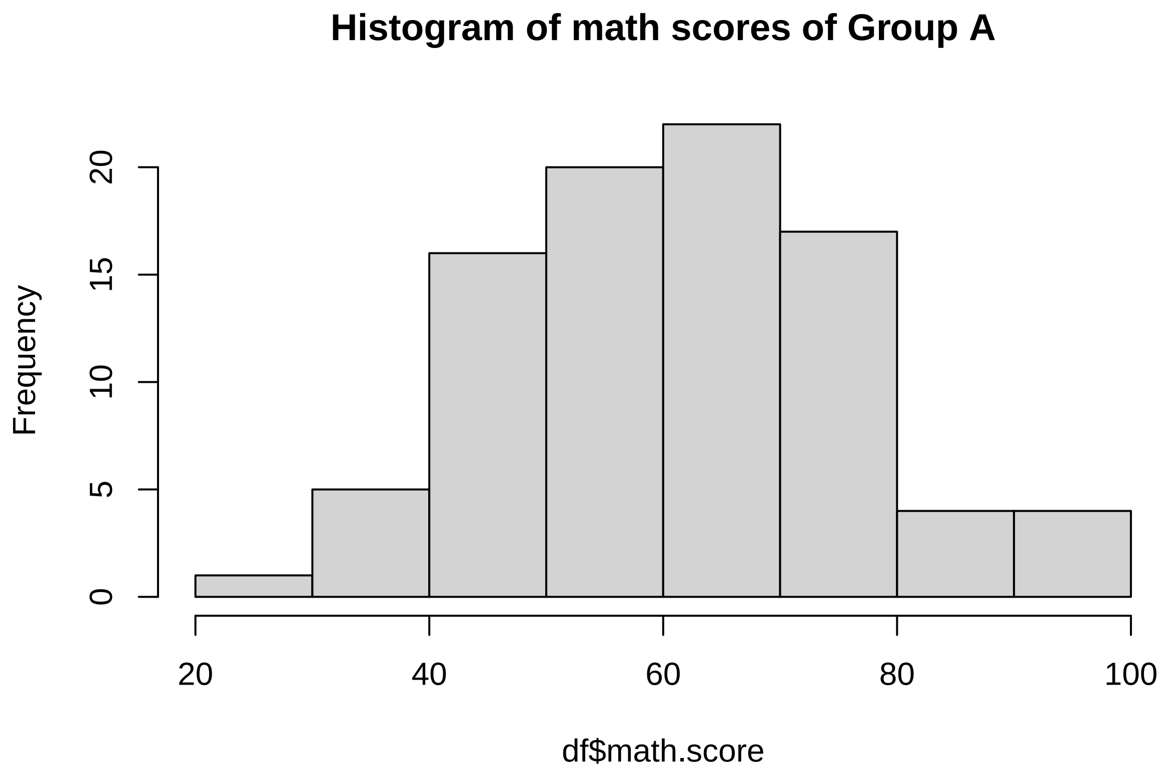


Result We see a huge difference here. A large chunk of students of Group E are talented in math and reading.

```r
df = data %>%
  filter(race.ethnicity == "group E") %>%
  select(race.ethnicity, math.score, writing.score, reading.score)

hist(x= df$math.score, main = "Histogram of math scores of Group E")
```



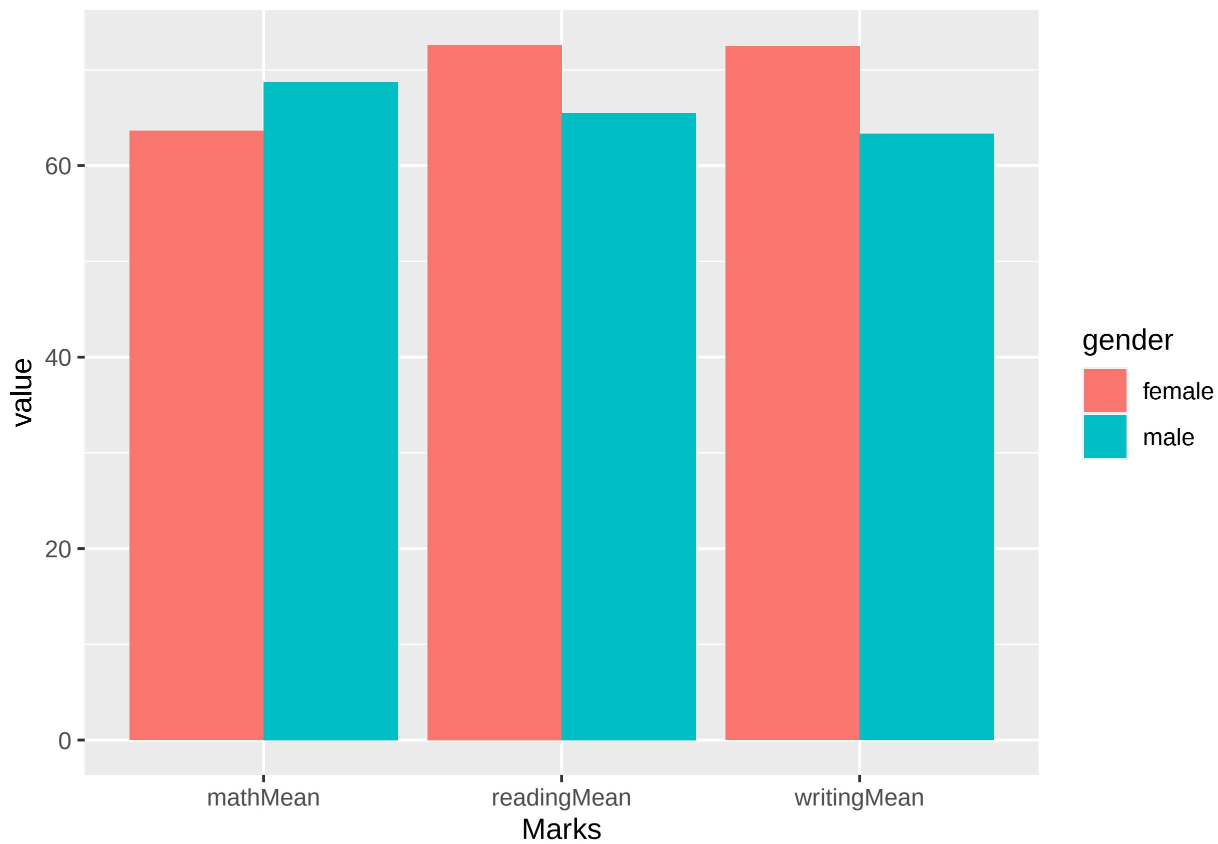

```r
df = data %>%
  filter(race.ethnicity == "group A") %>%
  select(race.ethnicity, math.score, writing.score, reading.score)

hist(x= df$math.score, main = "Histogram of math scores of Group A")
```



Result Group A is not doing good with math.

Gender

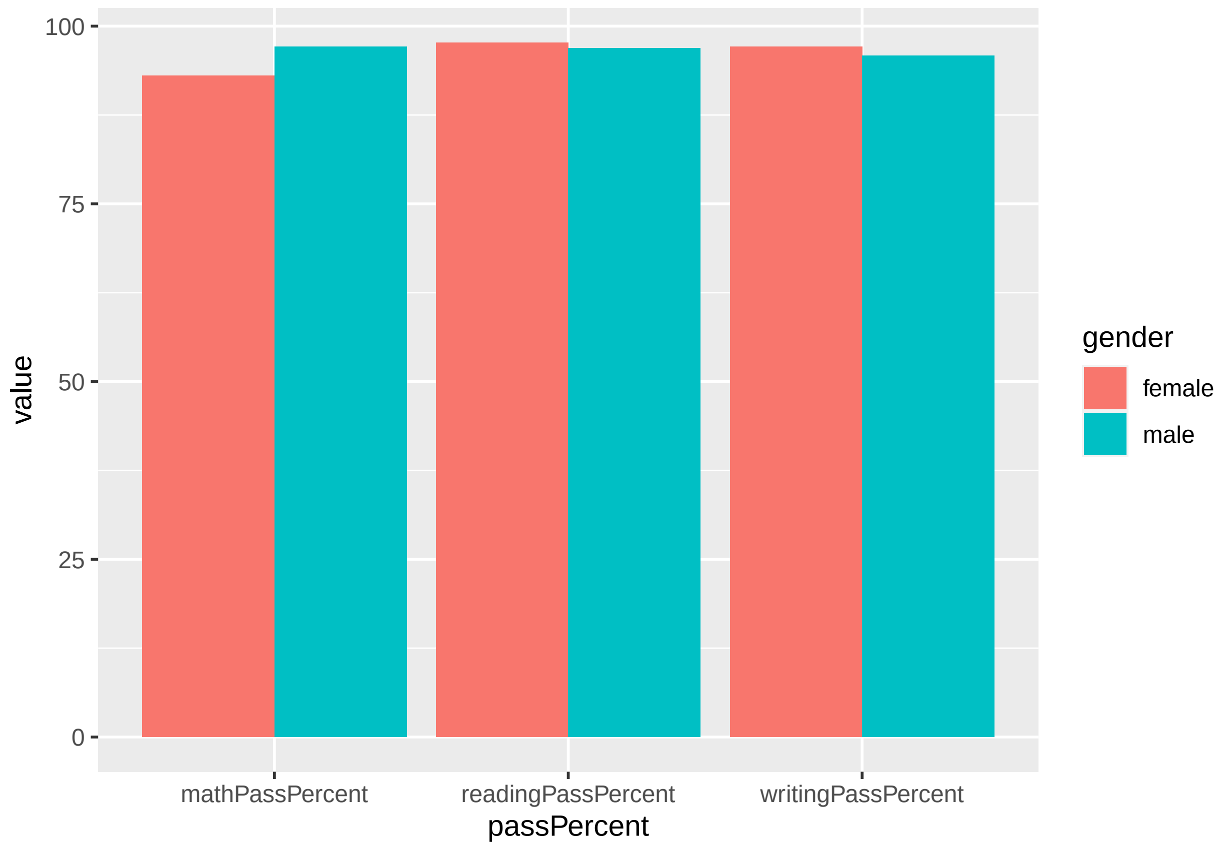
```
data %>%
    group_by(gender) %>%
    summarise(mathMean = mean(math.score),readingMean =  mean(reading.score),writingMean = mean(writing
  melt(id.vars="gender", variable.name = "Marks") %>%
  ggplot(aes(x=Marks, y=value, fill=gender)) +
  geom_bar(stat="identity", position = position_dodge())
```



Result In general there are no large difference found.

```
mathpass = data %>%
  group_by(gender) %>%
  filter(math.score>40) %>%
  count(gender)

writingpass = data %>%
  group_by(gender) %>%
  filter(writing.score>40) %>%
  count(gender)

readingpass = data %>%
  group_by(gender) %>%
  filter(reading.score>40) %>%
  count(gender)

total = data %>%
  group_by(gender) %>%
  count(gender)

total$mathPassPercent = mathpass$n / total$n * 100
total$readingPassPercent = readingpass$n / total$n * 100
total$writingPassPercent = writingpass$n / total$n * 100
```

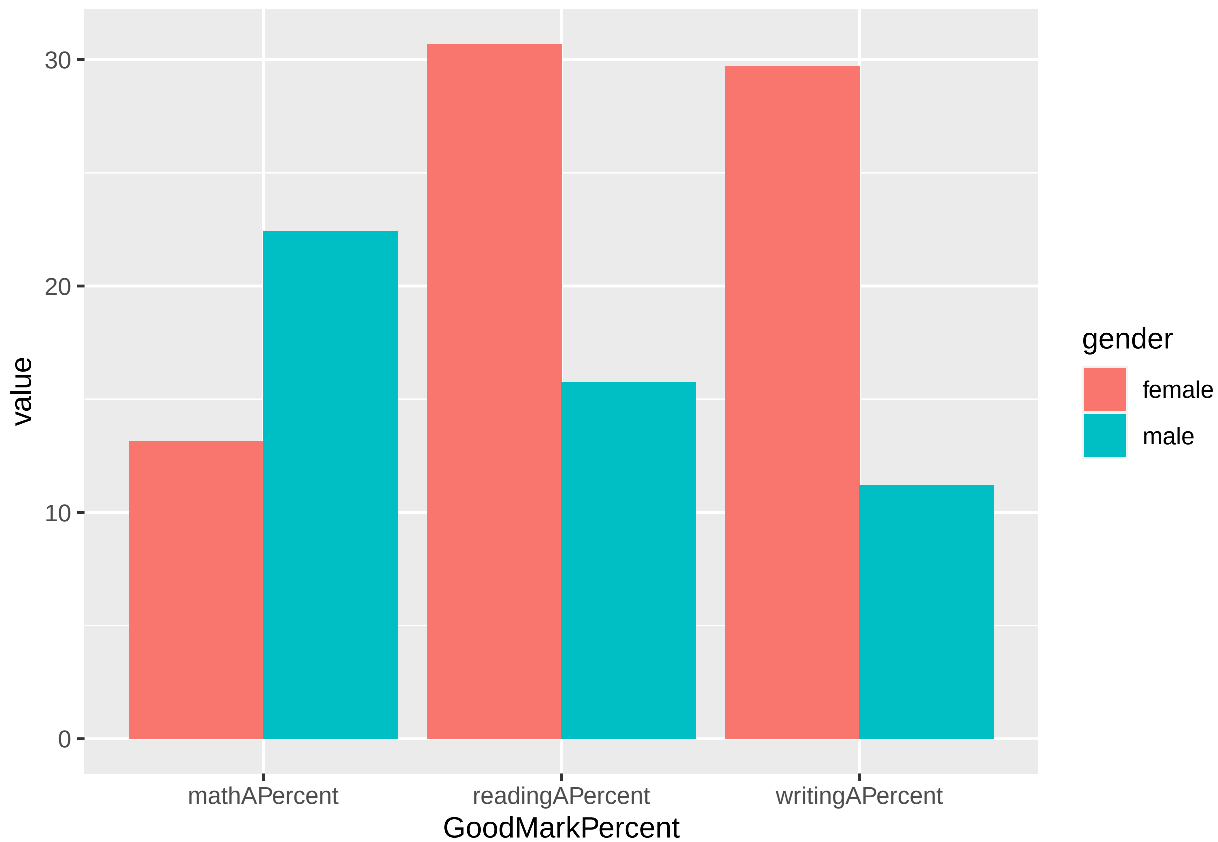
```r
total = subset(total, select = -c(n))

melt(total, id.vars="gender", variable.name = "passPercent") %>%
  ggplot(aes(x=passPercent, y=value, fill=gender)) +
  geom_bar(stat="identity", position = position_dodge())
```



Result Both boys and girls have good pass percentage over all the subjects. Boys have a little edge over math and girls have it over reading and writing.

```r
math = data %>%
  group_by(gender) %>%
  filter(math.score>80) %>%
  count(gender)

writing = data %>%
  group_by(gender) %>%
  filter(writing.score>80) %>%
  count(gender)

reading = data %>%
  group_by(gender) %>%
  filter(reading.score>80) %>%
  count(gender)
```

```
total = data %>%
  group_by(gender) %>%
  count(gender)

total$mathAPercent = math$n / total$n * 100
total$readingAPercent = reading$n / total$n * 100
total$writingAPercent = writing$n / total$n * 100

total = subset(total, select = -c(n))

melt(total, id.vars="gender", variable.name = "GoodMarkPercent") %>%
  ggplot(aes(x=GoodMarkPercent, y=value, fill=gender)) +
  geom_bar(stat="identity", position = position_dodge())
```



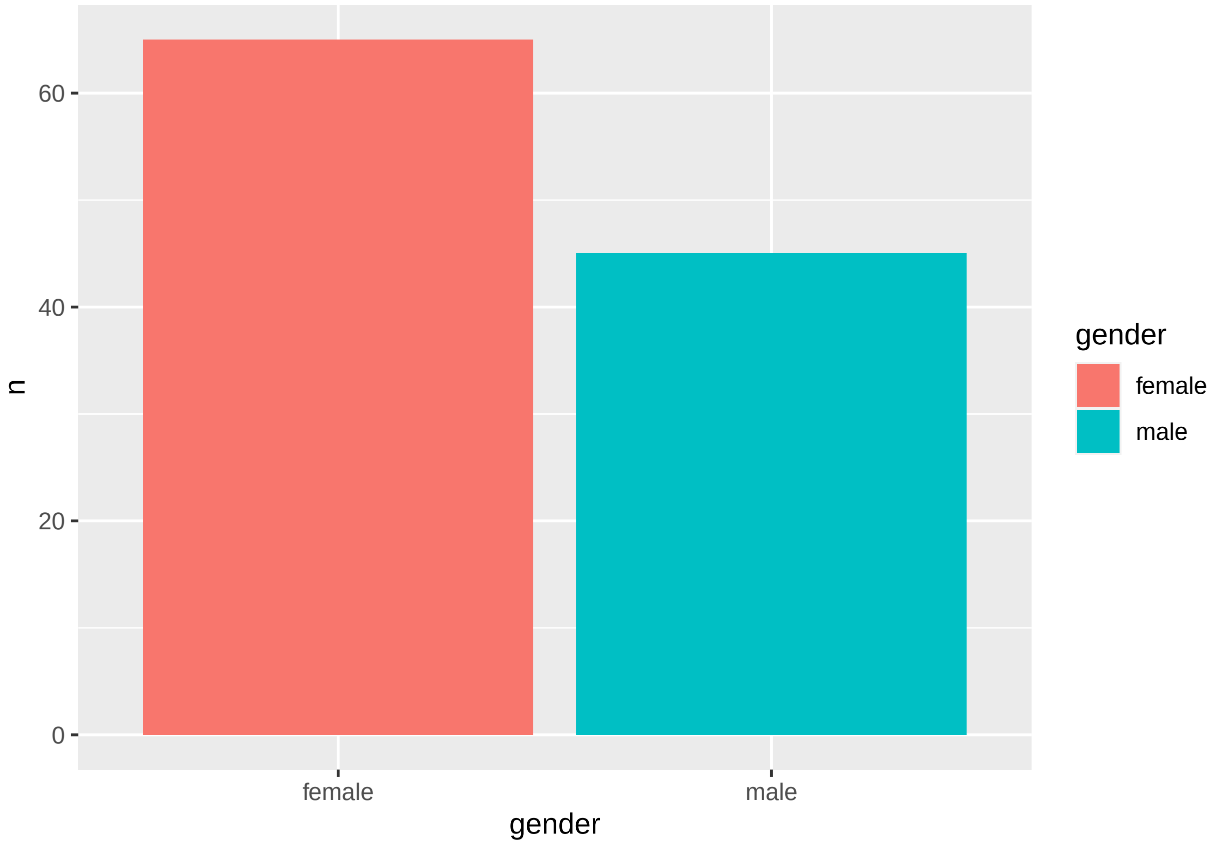
Result Adding to the inference we got in the last plot, we find boys perform well in math and girls perform well in reading and writing.

Number of students who has scored above 80 in all subjects grouped by gender.

```
data %>%
    group_by(gender) %>%
    select(math.score,reading.score,writing.score) %>%
    filter(math.score > 80, reading.score > 80, writing.score > 80) %>%
    count(gender) %>%
```

```r
    ggplot(data = ., aes(x = gender, y = n,
    fill = gender)) + geom_bar(stat = "identity")
```

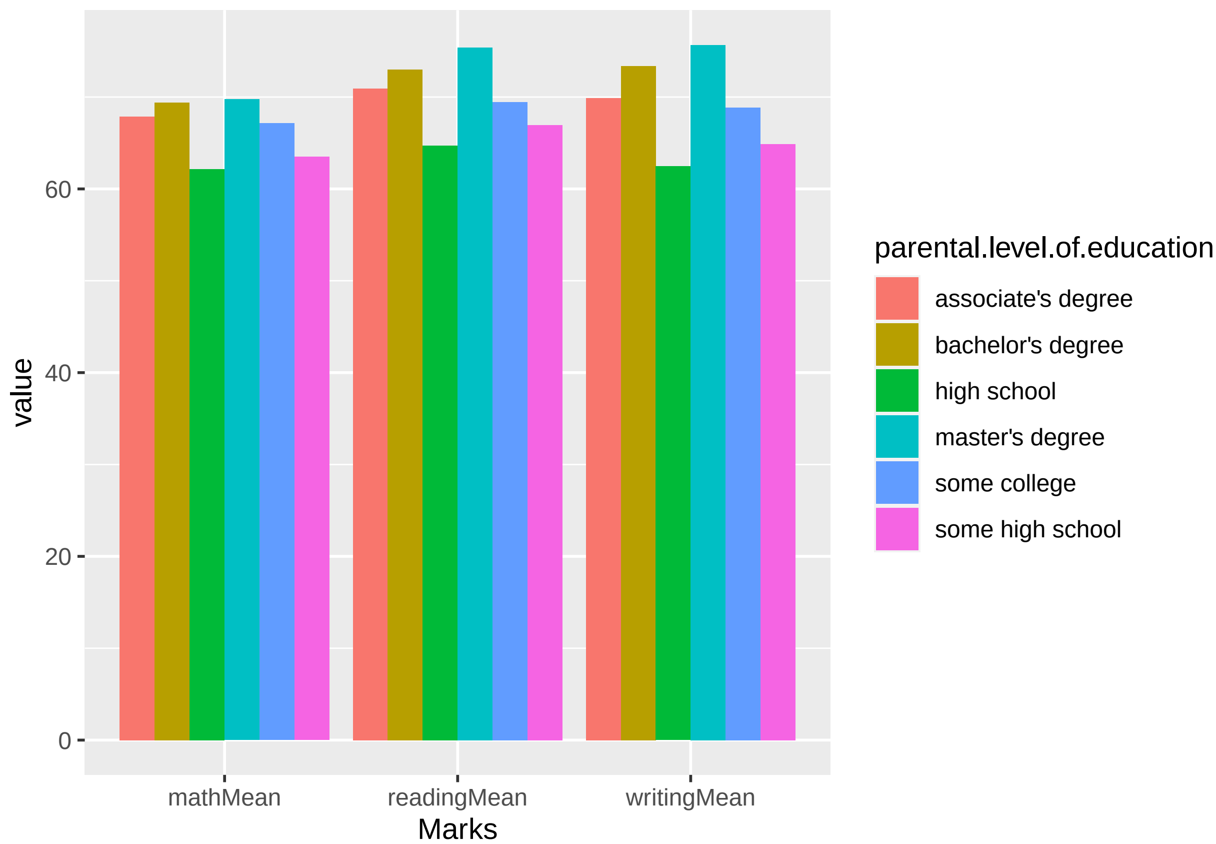
```
## Adding missing grouping variables: `gender`
```



Result Overall, Female students are more studious than male students.

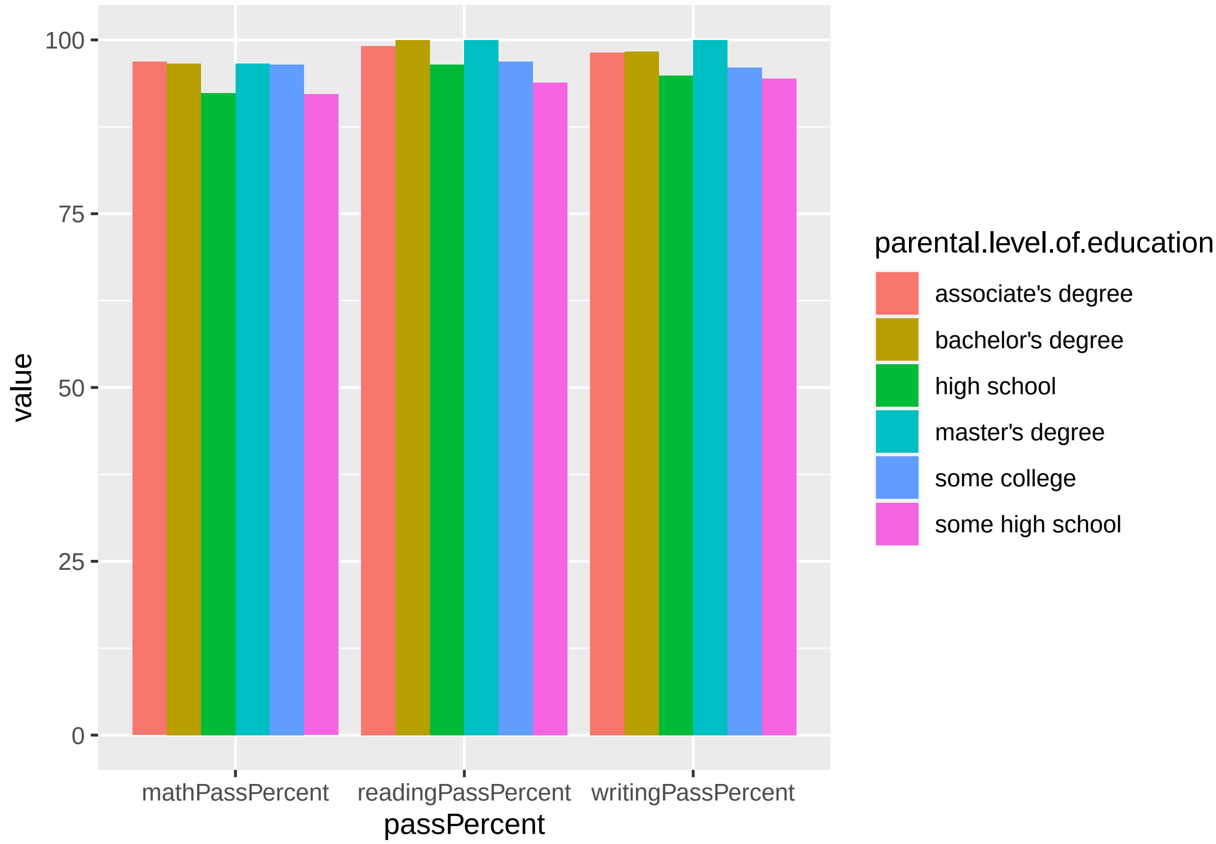
Level of Education Differences

```r
data %>%
    group_by(parental.level.of.education) %>%
    summarise(mathMean = mean(math.score),readingMean =  mean(reading.score),writingMean = mean(writing
  melt(id.vars="parental.level.of.education", variable.name = "Marks") %>%
  ggplot(aes(x=Marks, y=value, fill=parental.level.of.education)) +
  geom_bar(stat="identity", position = position_dodge())
```



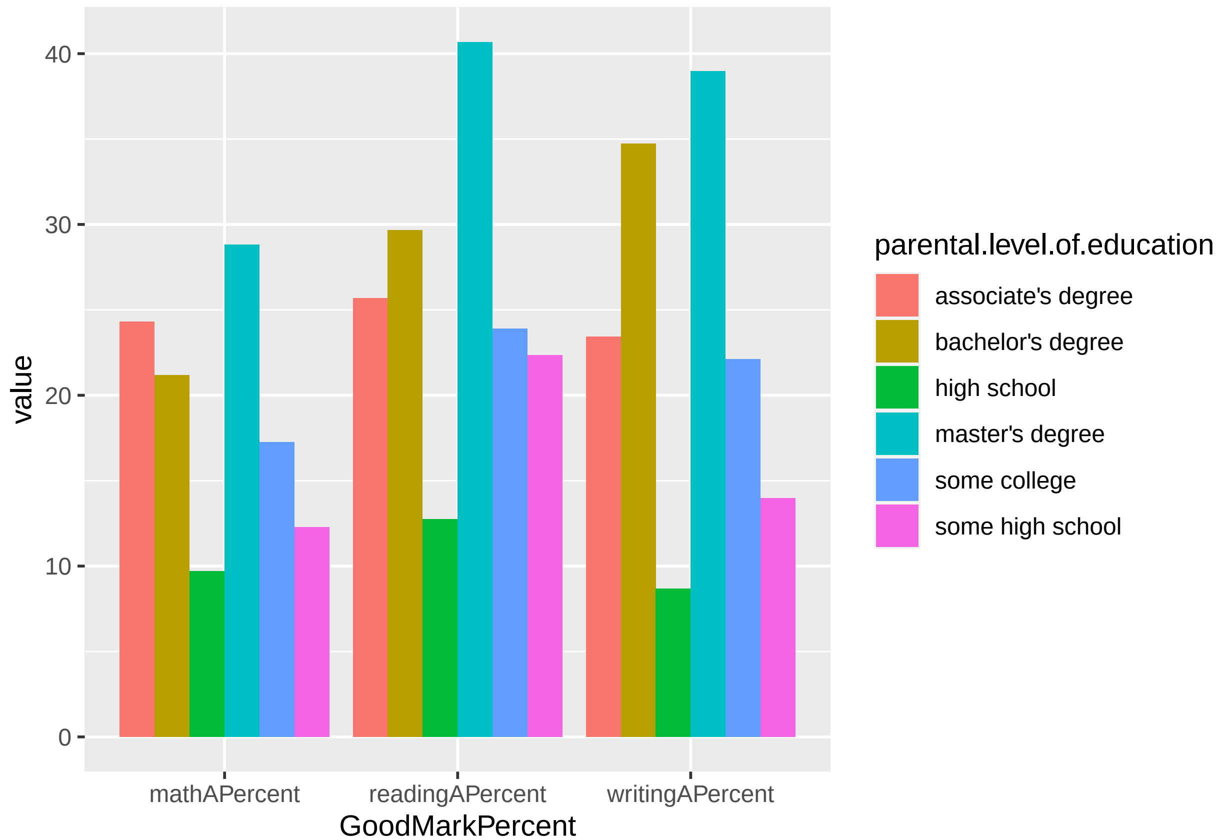

```
mathpass = data %>%
  group_by(parental.level.of.education) %>%
  filter(math.score>40) %>%
  count(parental.level.of.education)

writingpass = data %>%
  group_by(parental.level.of.education) %>%
  filter(writing.score>40) %>%
  count(parental.level.of.education)

readingpass = data %>%
  group_by(parental.level.of.education) %>%
  filter(reading.score>40) %>%
  count(parental.level.of.education)

total = data %>%
  group_by(parental.level.of.education) %>%
  count(parental.level.of.education)

total$mathPassPercent = mathpass$n / total$n * 100
total$readingPassPercent = readingpass$n / total$n * 100
total$writingPassPercent = writingpass$n / total$n * 100

total = subset(total, select = -c(n))

melt(total, id.vars="parental.level.of.education", variable.name = "passPercent") %>%
```

```
ggplot(aes(x=passPercent, y=value, fill=parental.level.of.education)) +
geom_bar(stat="identity", position = position_dodge())
```



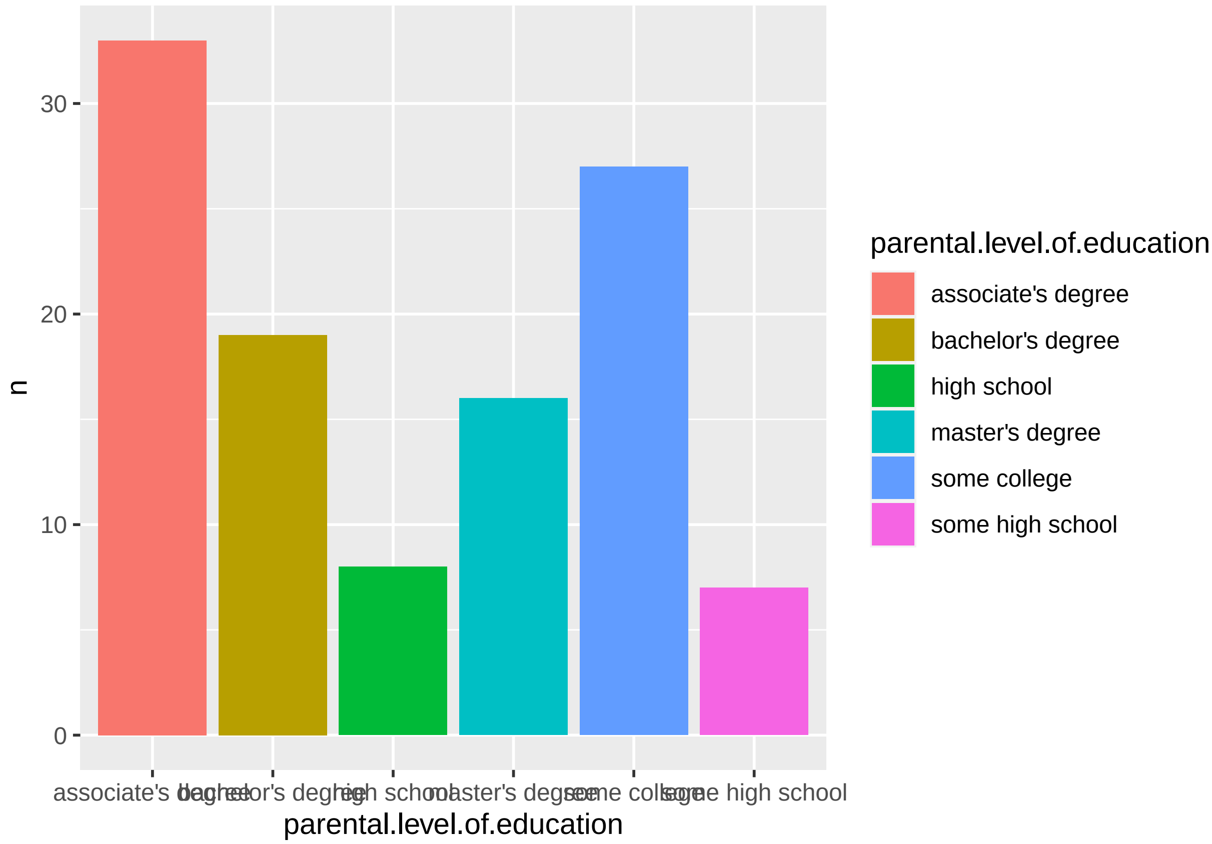

```
math = data %>%
  group_by(parental.level.of.education) %>%
  filter(math.score>80) %>%
  count(parental.level.of.education)

writing = data %>%
  group_by(parental.level.of.education) %>%
  filter(writing.score>80) %>%
  count(parental.level.of.education)

reading = data %>%
  group_by(parental.level.of.education) %>%
  filter(reading.score>80) %>%
  count(parental.level.of.education)

total = data %>%
  group_by(parental.level.of.education) %>%
  count(parental.level.of.education)

total$mathAPercent = math$n / total$n * 100
total$readingAPercent = reading$n / total$n * 100
total$writingAPercent = writing$n / total$n * 100
```

```
total = subset(total, select = -c(n))

melt(total, id.vars="parental.level.of.education", variable.name = "GoodMarkPercent") %>%
  ggplot(aes(x=GoodMarkPercent, y=value, fill=parental.level.of.education)) +
  geom_bar(stat="identity", position = position_dodge())
```




```
data %>%
    group_by(parental.level.of.education) %>%
    select(math.score,reading.score,writing.score) %>%
    filter(math.score > 80, reading.score > 80, writing.score > 80) %>%
    count(parental.level.of.education) %>%
    ggplot(data = ., aes(x = parental.level.of.education, y = n,
    fill = parental.level.of.education)) + geom_bar(stat = "identity")
```

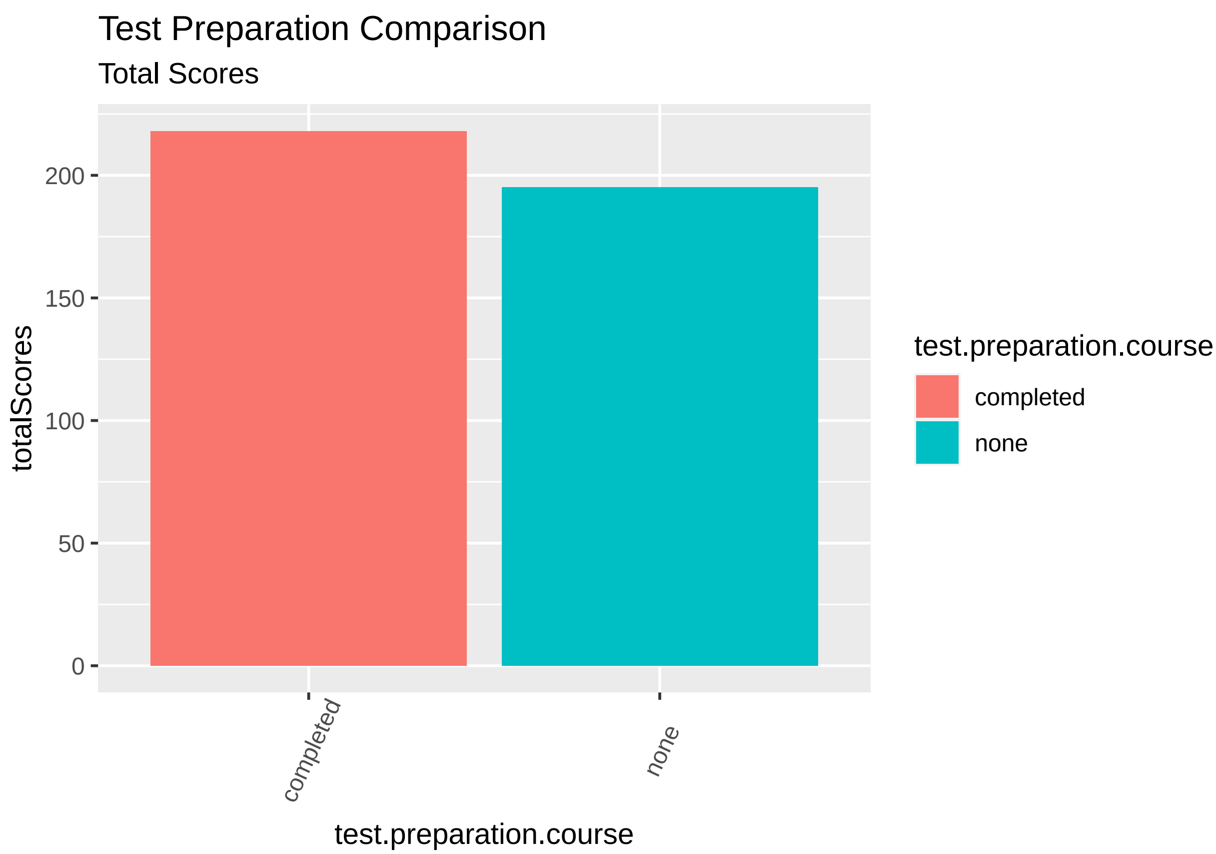
```
## Adding missing grouping variables: `parental.level.of.education`
```



Result Generally, the results are the same. Therefore, the scores does not changes with educational level.

Test Preperation Course Effect on Scores

```
data %>%
    group_by(test.preparation.course) %>%
    summarise(mathMean = mean(math.score),readingMean =  mean(reading.score),writingMean = mean(writing
    mutate(totalScores = mathMean + readingMean + writingMean) %>%
    ggplot(data = ., aes(x = test.preparation.course, y = totalScores,
    fill = test.preparation.course)) + geom_bar(stat = "identity") +
    labs(title="Test Preparation Comparison",
    subtitle="Total Scores") +
    theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



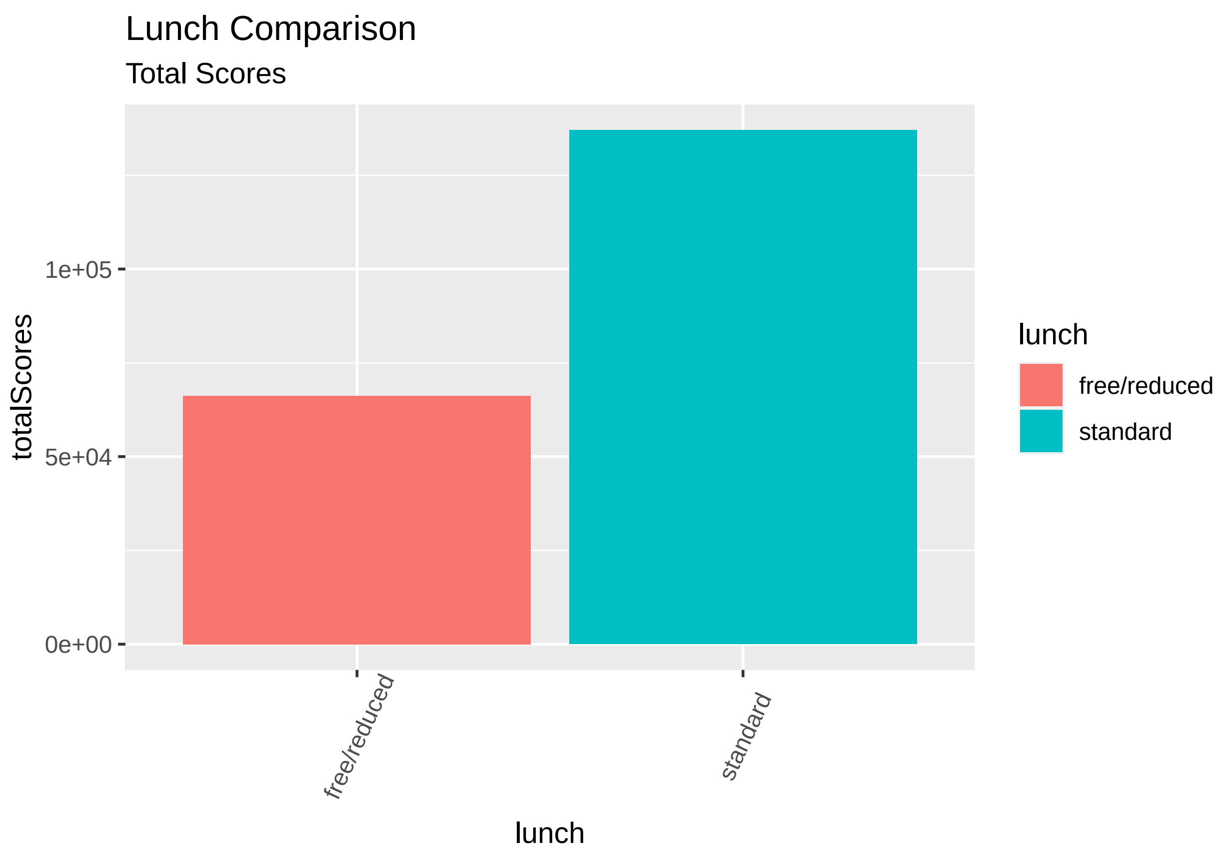
Result Total scores of competeled ones is higher than who had not taken the preparation course.

Analyzing Lunch effects on Scores

```
totalLunch <- table(data$lunch)
totalLunch
```

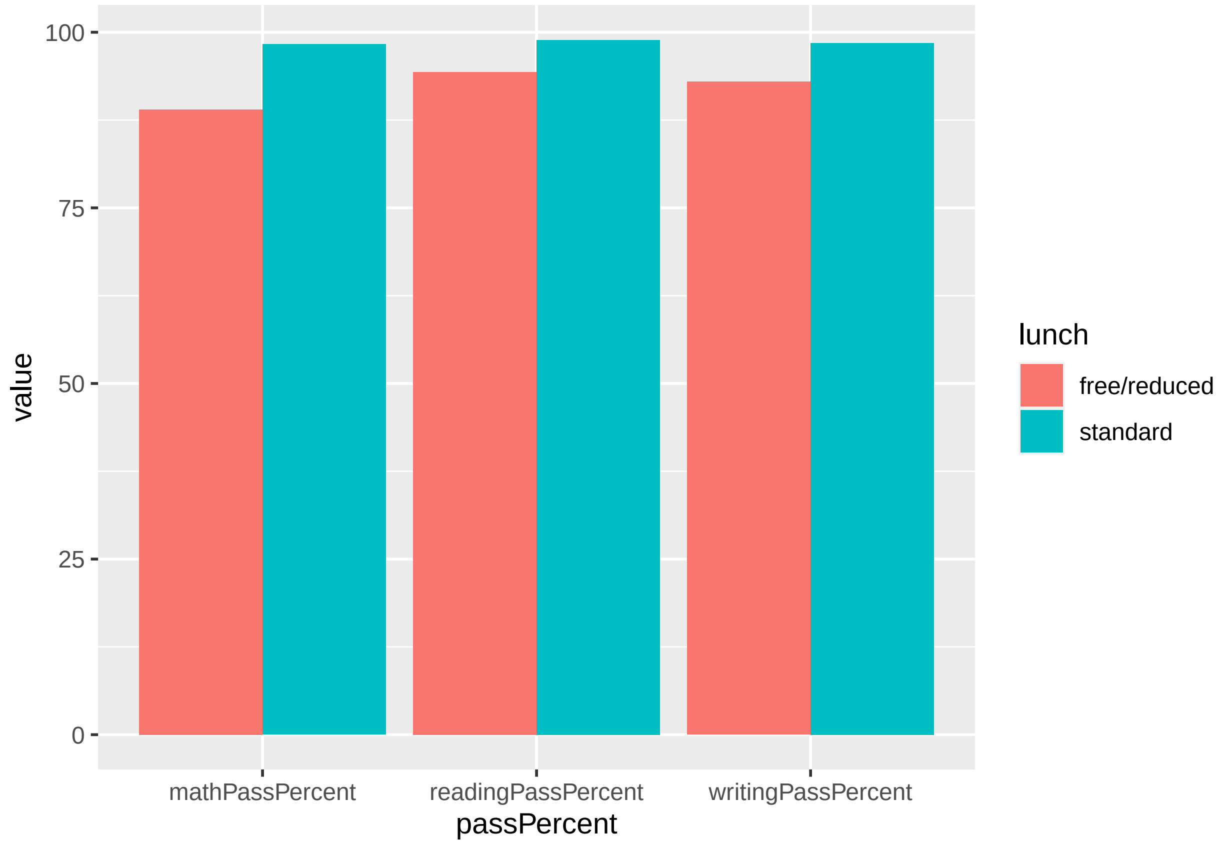
```
##
## free/reduced     standard
##          355          645
```

```
lunch <- data %>%
    group_by(lunch) %>%
    summarise(mathTotal = sum(math.score),readingTotal =  sum(reading.score),writingTotal = sum(writing
    mutate(totalScores = mathTotal + readingTotal + writingTotal)
lunch %>%
    ggplot(data = ., aes(x = lunch, y = totalScores,
    fill = lunch)) + geom_bar(stat = "identity") +
    labs(title="Lunch Comparison",
    subtitle="Total Scores") +
    theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



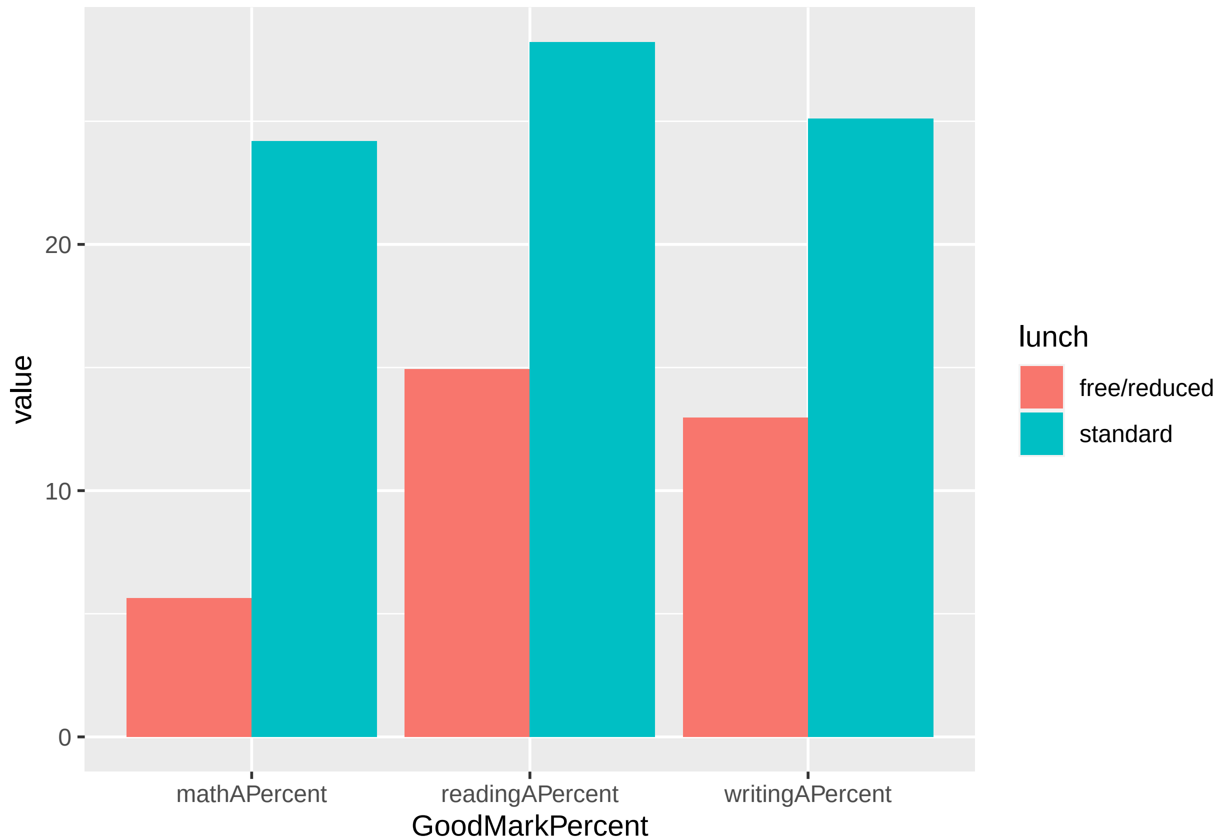

```
mathpass = data %>%
  group_by(lunch) %>%
  filter(math.score>40) %>%
  count(lunch)

writingpass = data %>%
  group_by(lunch) %>%
  filter(writing.score>40) %>%
  count(lunch)

readingpass = data %>%
  group_by(lunch) %>%
  filter(reading.score>40) %>%
  count(lunch)

total = data %>%
  group_by(lunch) %>%
  count(lunch)

total$mathPassPercent = mathpass$n / total$n * 100
total$readingPassPercent = readingpass$n / total$n * 100
total$writingPassPercent = writingpass$n / total$n * 100

total = subset(total, select = -c(n))

melt(total, id.vars="lunch", variable.name = "passPercent") %>%
```

```
  ggplot(aes(x=passPercent, y=value, fill=lunch)) +
  geom_bar(stat="identity", position = position_dodge())
```




```
math = data %>%
  group_by(lunch) %>%
  filter(math.score>80) %>%
  count(lunch)

writing = data %>%
  group_by(lunch) %>%
  filter(writing.score>80) %>%
  count(lunch)

reading = data %>%
  group_by(lunch) %>%
  filter(reading.score>80) %>%
  count(lunch)

total = data %>%
  group_by(lunch) %>%
  count(lunch)

total$mathAPercent = math$n / total$n * 100
total$readingAPercent = reading$n / total$n * 100
total$writingAPercent = writing$n / total$n * 100
```

```r
total = subset(total, select = -c(n))

melt(total, id.vars="lunch", variable.name = "GoodMarkPercent") %>%
  ggplot(aes(x=GoodMarkPercent, y=value, fill=lunch)) +
  geom_bar(stat="identity", position = position_dodge())
```



Result In general, standard lunches seen more effective on the students.

Correlation of Scores

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
    )
}
res2<-rcorr(as.matrix(data[,6:8]))
flattenCorrMatrix(res2$r, res2$P)
```

```
##             row        column       cor p
## 1    math.score reading.score 0.8175797 0
## 2    math.score writing.score 0.8026420 0
## 3 reading.score writing.score 0.9545981 0
```

Conclusion

There is highly correlation between scores. Thus, the students who takes low scores at one of area could take another low score and high scores take another high scores.

Being female, parents having a master degree and being a group E ethnicity is a advantage in education or maybe for carrier.

Machine learning

```
library(superml)
```

```
## Loading required package: R6
```

```
library(caTools)
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster

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

```
library(quantreg)
```

```
## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve

##
## Attaching package: 'quantreg'

## The following object is masked from 'package:Hmisc':
##
##     latex

## The following object is masked from 'package:survival':
##
##     untangle.specials
```

```
df = data
```

```
split = sample.split(df, SplitRatio = 0.9)
train = subset(df, split=="TRUE")
test = subset(df, split=="FALSE")
head(train)
```

```
##   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
```

```r
head(test)
```

```
##    gender race.ethnicity parental.level.of.education        lunch
## 8    male        group B                some college free/reduced
## 16 female        group C            some high school     standard
## 24 female        group C            some high school     standard
## 32 female        group B                some college     standard
## 40   male        group B          associate's degree free/reduced
## 48 female        group C                 high school     standard
##    test.preparation.course math.score reading.score writing.score
## 8                     none         40            43            39
## 16                    none         69            75            78
## 24                    none         69            73            73
## 32                    none         63            65            61
## 40                    none         57            56            57
## 48                    none         66            71            76
```

Train set size: 875 Test set size: 125

Label Encoding

```r
genderlabel = LabelEncoder$new()
train$gender = genderlabel$fit_transform(train$gender)

racelabel = LabelEncoder$new()
train$race.ethnicity = racelabel$fit_transform(train$race.ethnicity)

parentallabel <- LabelEncoder$new()
train$parental.level.of.education = parentallabel$fit_transform(train$parental.level.of.education)

lunchlabel <- LabelEncoder$new()
train$lunch = lunchlabel$fit_transform(train$lunch)

testlabel <- LabelEncoder$new()
train$test.preparation.course = testlabel$fit_transform(train$test.preparation.course)

head(train)
```

```
##   gender race.ethnicity parental.level.of.education lunch
## 1      0              0                           0     0
## 2      0              1                           1     0
## 3      0              0                           2     0
## 4      1              2                           3     1
## 5      1              1                           1     0
## 6      0              0                           3     0
##   test.preparation.course math.score reading.score writing.score
## 1                       0         72            72            74
## 2                       1         69            90            88
## 3                       0         90            95            93
## 4                       0         47            57            44
## 5                       0         76            78            75
## 6                       0         71            83            78
```

Train test split:

```r
train_x = subset(train, select = c(gender, race.ethnicity, parental.level.of.education, lunch, test.prep
train_y = subset(train, select = -c(gender, race.ethnicity, parental.level.of.education, lunch, test.pre


test_x = subset(test, select = c(gender, race.ethnicity, parental.level.of.education, lunch, test.prepa
test_y = subset(test, select = -c(gender, race.ethnicity, parental.level.of.education, lunch, test.prepa


head(train_x)
```

```
##   gender race.ethnicity parental.level.of.education lunch
## 1      0              0                           0     0
## 2      0              1                           1     0
## 3      0              0                           2     0
## 4      1              2                           3     1
## 5      1              1                           1     0
## 6      0              0                           3     0
##   test.preparation.course
## 1                       0
## 2                       1
## 3                       0
## 4                       0
## 5                       0
## 6                       0
```

```r
head(train_y)
```

```
##   math.score reading.score writing.score
## 1         72            72            74
## 2         69            90            88
## 3         90            95            93
## 4         47            57            44
## 5         76            78            75
## 6         71            83            78
```

```r
head(test_x)
```

```
##    gender race.ethnicity parental.level.of.education        lunch
## 8    male        group B                some college free/reduced
## 16 female        group C            some high school     standard
## 24 female        group C            some high school     standard
## 32 female        group B                some college     standard
## 40   male        group B          associate's degree free/reduced
## 48 female        group C                 high school     standard
##    test.preparation.course
## 8                     none
## 16                    none
## 24                    none
## 32                    none
## 40                    none
## 48                    none
```

```r
head(test_y)
```

```
##    math.score reading.score writing.score
## 8          40            43            39
## 16         69            75            78
## 24         69            73            73
## 32         63            65            61
## 40         57            56            57
## 48         66            71            76
```

```r
test_x$gender = genderlabel$transform(test_x$gender)
test_x$race.ethnicity = racelabel$transform(test_x$race.ethnicity)
test_x$parental.level.of.education = parentallabel$transform(test_x$parental.level.of.education)
test_x$lunch = lunchlabel$transform(test_x$lunch)
test_x$test.preparation.course = testlabel$transform(test_x$test.preparation.course)

head(test_x)
```

```
##    gender race.ethnicity parental.level.of.education lunch
## 8       1              0                           1     1
## 16      0              1                           5     0
## 24      0              1                           5     0
## 32      0              0                           1     0
## 40      1              0                           3     1
## 48      0              1                           4     0
##    test.preparation.course
## 8                        0
## 16                       0
## 24                       0
## 32                       0
## 40                       0
## 48                       0
```

Linear model between math score and all factors

```r
lm_model_math = lm(math.score ~ gender+race.ethnicity+parental.level.of.education+lunch+test.preparatio
summary(lm_model_math)
```

```
##
## Call:
## lm(formula = math.score ~ gender + race.ethnicity + parental.level.of.education +
##     lunch + test.preparation.course, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.861  -8.820  -0.369   9.647  31.134
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   65.1045     1.2446  52.312  < 2e-16 ***
```

```
## gender                          5.0521     0.9088   5.559 3.61e-08 ***
## race.ethnicity                  1.9775     0.3335   5.929 4.40e-09 ***
## parental.level.of.education    -1.2375     0.2689  -4.602 4.80e-06 ***
## lunch                         -11.0333     0.9466 -11.656  < 2e-16 ***
## test.preparation.course         5.5452     0.9434   5.878 5.93e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.4 on 869 degrees of freedom
## Multiple R-squared:  0.237,  Adjusted R-squared:  0.2327
## F-statistic:    54 on 5 and 869 DF,  p-value: < 2.2e-16
```

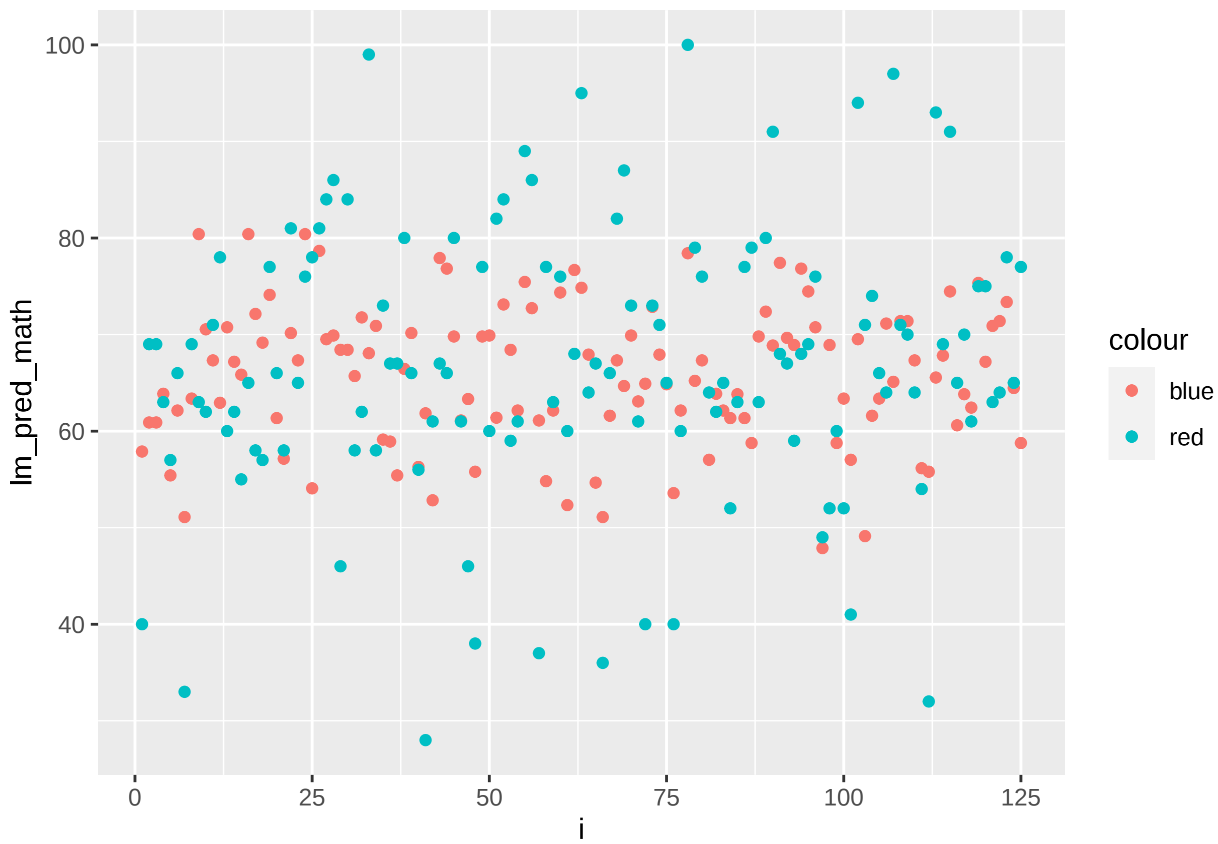
```
lm_pred_math = predict(lm_model_math, test_x)
lm_rmse_math  = sqrt(sum((lm_pred_math- test_y$math.score)^2)/125)
lm_rmse_math
```

```
## [1] 12.85681
```

```
i = seq(1:length(test_y$math.score))
lm_table = data.frame(i, lm_pred_math, test_y$math.score)
head(lm_table)
```

```
##    i lm_pred_math test_y.math.score
## 8  1     57.88585                40
## 16 2     60.89442                69
## 24 3     60.89442                69
## 32 4     63.86699                63
## 40 5     55.41079                57
## 48 6     62.13194                66
```

```
ggplot(data=lm_table) + geom_point(aes(x=i,y=lm_pred_math, color = "blue")) +
  geom_point(aes(x=i,y=test_y.math.score, color = "red"))
```



Linear model between writing score and all factors

```
lm_model_writing = lm(writing.score ~ gender+race.ethnicity+parental.level.of.education+lunch+test.prepa
summary(lm_model_writing)
```

```
##
## Call:
## lm(formula = writing.score ~ gender + race.ethnicity + parental.level.of.education +
##     lunch + test.preparation.course, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.492  -8.255   0.301   9.334  29.967
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  73.5678     1.1893  61.857  < 2e-16 ***
## gender                       -9.0851     0.8685 -10.461  < 2e-16 ***
## race.ethnicity                1.4371     0.3187   4.509 7.42e-06 ***
## parental.level.of.education  -1.6288     0.2569  -6.339 3.71e-10 ***
## lunch                        -8.3693     0.9045  -9.252  < 2e-16 ***
## test.preparation.course       9.8384     0.9016  10.913  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 12.8 on 869 degrees of freedom
## Multiple R-squared:  0.2999, Adjusted R-squared:  0.2959
## F-statistic: 74.47 on 5 and 869 DF,  p-value: < 2.2e-16
```

```
lm_pred_writing = predict(lm_model_writing, test_x)
lm_rmse_writing = sqrt(sum((lm_pred_writing- test_y$writing.score)^2)/125)
lm_rmse_writing
```

```
## [1] 11.92995
```

Linear model between reading score and all factors

```
lm_model_reading = lm(reading.score ~ gender+race.ethnicity+parental.level.of.education+lunch+test.prepa
summary(lm_model_reading)
```

```
##
## Call:
## lm(formula = reading.score ~ gender + race.ethnicity + parental.level.of.education +
##     lunch + test.preparation.course, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -44.59  -9.04   0.38   9.84  32.36
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  73.2475     1.2164  60.217  < 2e-16 ***
## gender                       -7.1774     0.8882  -8.081 2.15e-15 ***
## race.ethnicity                1.2477     0.3260   3.828 0.000139 ***
## parental.level.of.education  -1.1189     0.2628  -4.257 2.29e-05 ***
## lunch                        -7.3083     0.9251  -7.900 8.43e-15 ***
## test.preparation.course       7.2439     0.9221   7.856 1.17e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.09 on 869 degrees of freedom
## Multiple R-squared:  0.2021, Adjusted R-squared:  0.1976
## F-statistic: 44.04 on 5 and 869 DF,  p-value: < 2.2e-16
```

```
lm_pred_reading = predict(lm_model_reading, test_x)
lm_rmse_reading = sqrt(sum((lm_pred_reading- test_y$reading.score)^2)/125)
lm_rmse_reading
```

```
## [1] 12.74009
```

Anova

```r
aov_model_math = aov(math.score ~ gender+race.ethnicity+parental.level.of.education+lunch+test.preparati
aov_pred_math = predict(aov_model_math, test_x)
aov_rmse_math = sqrt(sum((aov_pred_math- test_y$math.score)^2)/125)
aov_rmse_math
```

```
## [1] 12.85681
```

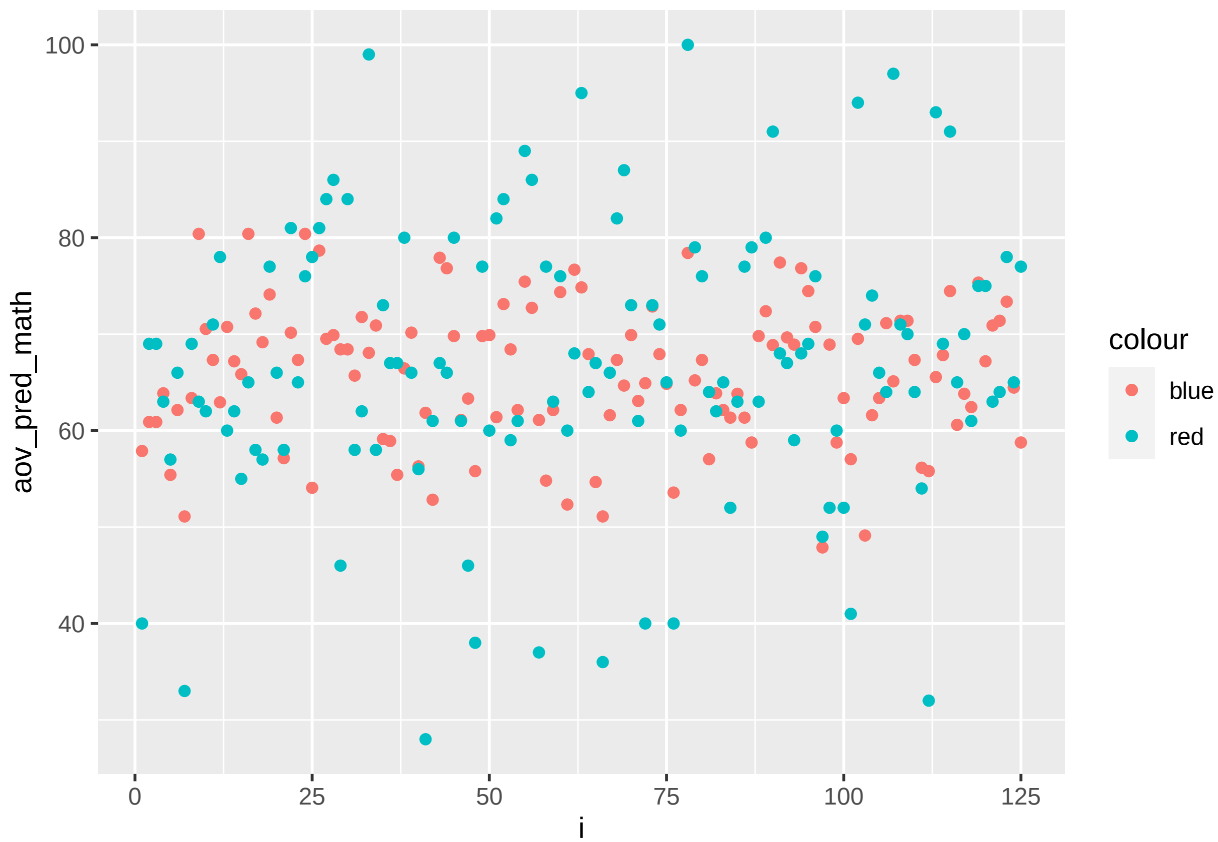
```r
aov_model_writing = aov(writing.score ~ gender+race.ethnicity+parental.level.of.education+lunch+test.pre
aov_pred_writing = predict(aov_model_writing, test_x)
aov_rmse_writing = sqrt(sum((aov_pred_writing- test_y$writing.score)^2)/125)
aov_rmse_writing
```

```
## [1] 11.92995
```

```r
aov_model_reading = aov(reading.score ~ gender+race.ethnicity+parental.level.of.education+lunch+test.pre
aov_pred_reading = predict(aov_model_reading, test_x)
aov_rmse_reading = sqrt(sum((aov_pred_reading- test_y$reading.score)^2)/125)
aov_rmse_reading
```

```
## [1] 12.74009
```

```r
i = seq(1:length(test_y$math.score))
aov_table = data.frame(i, aov_pred_math, test_y$math.score)

ggplot(data=aov_table) + geom_point(aes(x=i,y=aov_pred_math, color = "blue")) +
  geom_point(aes(x=i,y=test_y.math.score, color = "red"))
```



Random forest regression

```
library(caTools)
library(randomForest)
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

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

```r
library(Metrics)
```

```
##
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':
##
##     precision, recall
```

Random forest on Math score

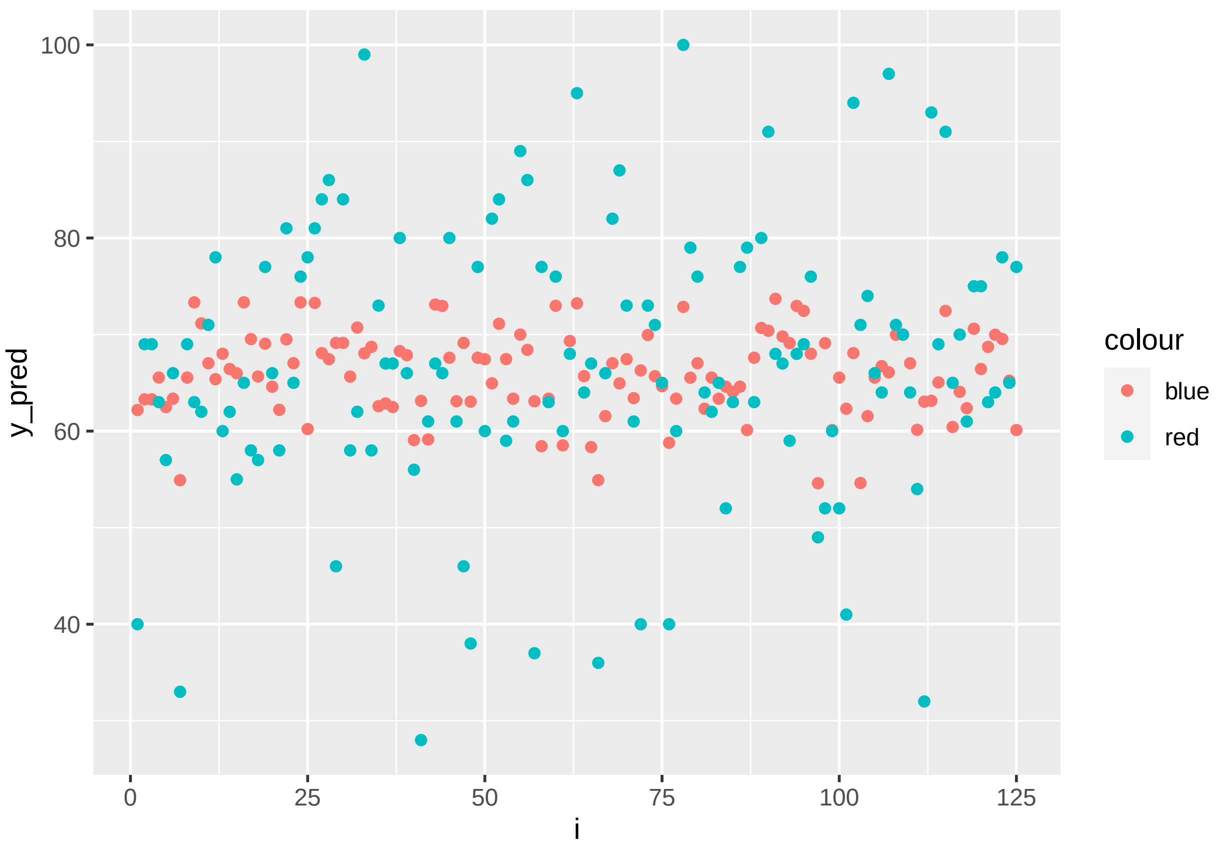
```r
set.seed(123)


regressor = randomForest(x = train_x,
                         y = train_y$math.score,
                         ntree = 500)

y_pred = predict(regressor, test_x)


i = seq(1:length(test_y$math.score))
regression_table = data.frame(i, y_pred, test_y$math.score)
head(regression_table)
```

```
##    i   y_pred test_y.math.score
## 8  1 62.19110                40
## 16 2 63.28899                69
## 24 3 63.28899                69
## 32 4 65.54655                63
## 40 5 62.48777                57
## 48 6 63.35773                66
```

```r
rf_rmse_math =  rmse(y_pred, test_y$math.score)
rf_rmse_math
```

```
## [1] 13.12167
```

```r
ggplot(data=regression_table) + geom_point(aes(x=i,y=y_pred, color = "blue")) +
  geom_point(aes(x=i,y=test_y.math.score, color = "red"))
```



Random forest on writing score

```
regressor = randomForest(x = train_x,
                         y = train_y$writing.score,
                         ntree = 500)

y_pred = predict(regressor, test_x)
```

Sum of squared error:

```
rf_rmse_writing = rmse(y_pred, test_y$writing.score)
rf_rmse_writing
```

```
## [1] 12.57931
```

Random forest on reading score

```
regressor = randomForest(x = train_x,
                         y = train_y$reading.score,
                         ntree = 500)

y_pred = predict(regressor, test_x)
```

```r
rf_rmse_reading = rmse(y_pred, test_y$reading.score)
rf_rmse_reading
```

```
## [1] 13.18457
```

Analysing the models

```r
name = c("Math RMSE", "Reading RMSE", "Writing RMSE")
reg_rmse = c(lm_rmse_math, lm_rmse_reading, lm_rmse_writing)
aov_rmse = c(aov_rmse_math, aov_rmse_reading, aov_rmse_writing)
rf_rmse = c(rf_rmse_math, rf_rmse_reading, rf_rmse_writing)

data.frame(name, reg_rmse, aov_rmse, rf_rmse)
```

```
##           name reg_rmse aov_rmse  rf_rmse
## 1    Math RMSE 12.85681 12.85681 13.12167
## 2 Reading RMSE 12.74009 12.74009 13.18457
## 3 Writing RMSE 11.92995 11.92995 12.57931
```