Final Project: Student Performance

Miguel Garcia

December 4, 2020

Introduction

The data set “Students Performance” measures the math, reading, and writing exam scores of 1000 students. It contains 5 other variables which contain information about their gender, race/ethnicity, parental level of education, lunch status, and test preparation. The objective is to create a model that predicts whether the student will pass or fail their exams given their backgrounds. Another objective is to see how these different variables affect the student’s test scores. To achieve this goal, several tidyverse packages have been loaded in to help visualize, transform, read and model the data. The “partykit” and “modelr” packages are used for creating the model and predictions. Lastly, The “reshape2” package will be used to have transform the data so it can be used for creating multiple box plots.

```
library(tidyverse)
library(dplyr)
library(tidyr)
library("reshape2")
library("partykit")
library(modelr)
```

Data

The data set was posted as a csv file on Kaggle. The first step was importing the data set and looking for any irregularities. Fortunately, the data set is already tidy as each row corresponds to an observation, each column corresponds to a variable, and each entry only contains a single value.

```
student_performance<- read_csv("StudentsPerformance.csv")
```

```
## Parsed with column specification:
## cols(
##   gender = col_character(),
##   `race/ethnicity` = col_character(),
##   `parental level of education` = col_character(),
##   lunch = col_character(),
##   `test preparation course` = col_character(),
##   `math score` = col_double(),
##   `reading score` = col_double(),
##   `writing score` = col_double()
## )
```

```
student_performance
```

```
## # A tibble: 1,000 x 8
##    gender `race/ethnicity` `parental level… lunch `test preparati… `math score`
##    <chr>  <chr>            <chr>            <chr> <chr>                   <dbl>
##  1 female group B          bachelor's degr… stan… none                       72
##  2 female group C          some college     stan… completed                  69
##  3 female group B          master's degree  stan… none                       90
##  4 male   group A          associate's deg… free… none                       47
##  5 male   group C          some college     stan… none                       76
##  6 female group B          associate's deg… stan… none                       71
##  7 female group B          some college     stan… completed                  88
##  8 male   group B          some college     free… none                       40
##  9 male   group D          high school      free… completed                  64
## 10 female group B          high school      free… none                       38
## # … with 990 more rows, and 2 more variables: `reading score` <dbl>, `writing
## #   score` <dbl>
```

Although the data is tidy, it still needs to be transformed to complete the objectives. We need to rename the variables for easier data manipulation and add a surrogate key named id. We need to add a variable avgscore which takes the average of math, reading, and writing scores. We will use this new variable to mark whether the student passed or failed. The variable grade marks the student as pass if their average score is greater than 60, otherwise it marks it as fail. Hard_grade marks students as pass if all their scores are above 60, otherwise they are marked as fail. The variables grad and hgrad represent grade and hard_grade respectively in numeric form. 1 represents pass while 0 represents fail. The last step was ordering the levels of parent education.

```
educ_level<-c("some high school","high school","some college","associate's degree","b
achelor's degree","master's degree")
stu_perf_edited<-student_performance%>%
  rename(parenteduc="parental level of education", testprep="test preparation course"
, math_score="math score", reading_score="reading score",writing_score= "writing scor
e",race="race/ethnicity")%>%
  mutate(id = row_number())%>%
  select(id, everything())%>%
  mutate(avgscore=(math_score+writing_score+reading_score)/3)%>%
  mutate(grade= case_when(avgscore>=60 ~ "pass",avgscore<60 ~ "fail"))%>%
  mutate(hard_grade= case_when(math_score>=60 & reading_score>=60 & writing_score>=60
~ "pass",math_score<60 | reading_score<60 | writing_score<60 ~ "fail"))%>%
  mutate(grad= case_when(avgscore>=60 ~ 1,avgscore<60 ~ 0))%>%
  mutate(hgrad= case_when(math_score>=60 & reading_score>=60 & writing_score>=60 ~ 1,
math_score<60 | reading_score<60 | writing_score<60 ~ 0))%>%
   mutate(parenteduc=factor(parenteduc,level=educ_level))
```

71.5% percent of the students pass in this data using the lenient grading system. If we predict everyone passes, we will be right 71.5% of the time so this is the baseline solution. The baseline solution for the hard grading system is 60.3%. The goal is for the model to get a better percentage correct than the baseline solutions.

```
stu_perf_edited %>% summarize(pass_rate = mean(grad, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   pass_rate
##       <dbl>
## 1     0.715
```

```
stu_perf_edited %>% summarize(pass_rate = mean(hgrad, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   pass_rate
##       <dbl>
## 1     0.603
```

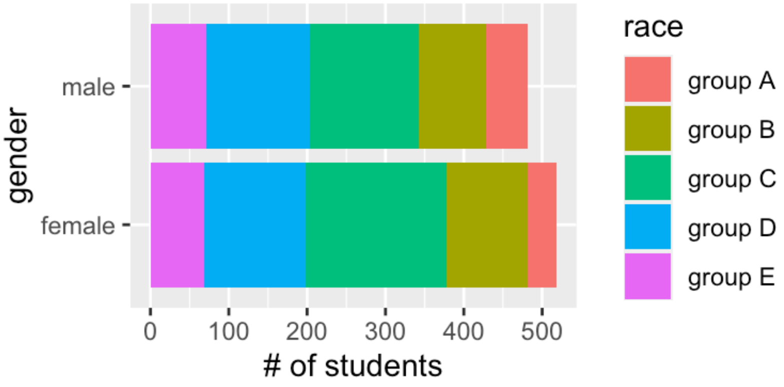
The model we will be using later on does not accept characters so we will change the variables to factors from characters.

```
stu_perf_fcts<-stu_perf_edited%>%
  mutate(hard_grade = factor(hard_grade))%>%
  mutate(grade = factor(grade))%>%
  mutate(lunch = factor(lunch))%>%
  mutate(testprep = factor(testprep))%>%
  mutate(race = factor(race))%>%
  mutate(gender = factor(gender))%>%
  select(hard_grade,race,parenteduc,lunch,testprep,gender,grade)
```

Demographic:

We want to get an understanding of the students and how they are represented by the different categories. Here we split up the bar chart by gender and noticed there are more females than males but the difference is not significant. Additionally, the bar chart and table with proportions shows the majority of students are either from the race D or C. Race B and E come after. Race A is the clear minority.

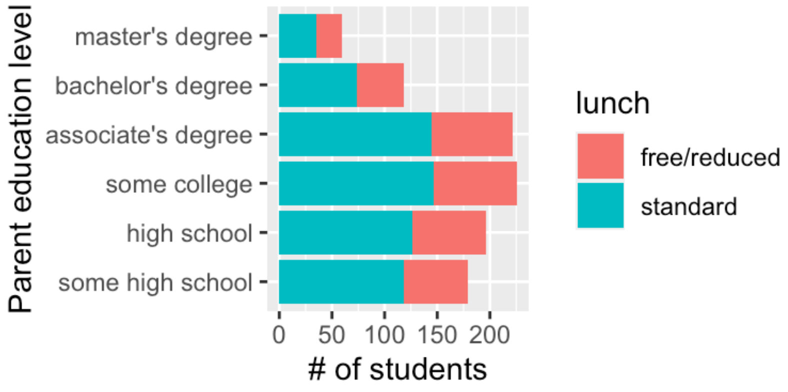
```
stu_perf_edited%>%
  ggplot(aes(fill=race))+
  geom_bar(aes(gender))+
  ylab("# of students")+
  coord_flip()
```



```
stu_perf_edited%>%
  group_by(race)%>%
  summarise(n=n(),prop=n/1000)%>%
  arrange(desc(prop))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
## # A tibble: 5 x 3
##   race        n  prop
##   <chr>   <int> <dbl>
## 1 group C   319 0.319
## 2 group D   262 0.262
## 3 group B   190 0.19 
## 4 group E   140 0.14 
## 5 group A    89 0.089
```

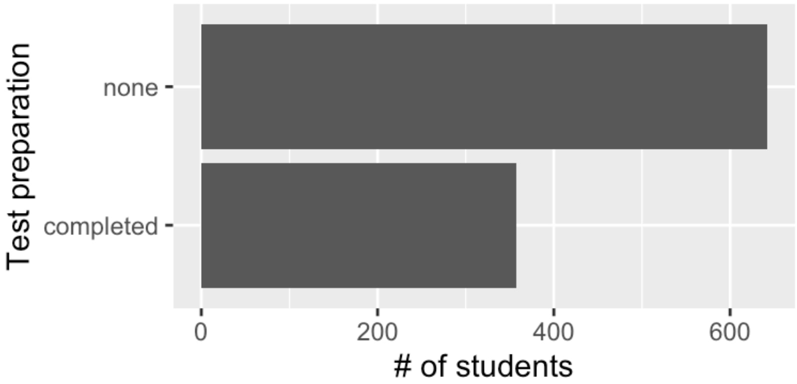
This bar chart is arranged in the order of education level with lowest at bottom. The majority of student's parents have an associate's degree, some college, highschool, highschool education level. The minority education level would be bachelor's degree and master's degree. Also, the majority of students are on standard lunch. It appears the student's parent education is not indicative of whether the student will be on standard or free/reduced lunch.



```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 6 x 3
##   parenteduc             n  prop
##   <fct>              <int> <dbl>
## 1 some college         226 0.226
## 2 associate's degree   222 0.222
## 3 high school          196 0.196
## 4 some high school     179 0.179
## 5 bachelor's degree    118 0.118
## 6 master's degree       59 0.059
```

The majority of students did not complete the test prep and the ratio for incomplete to complete is about 2:1.



```
## `summarise()` ungrouping output (override with `.groups` argument)
```

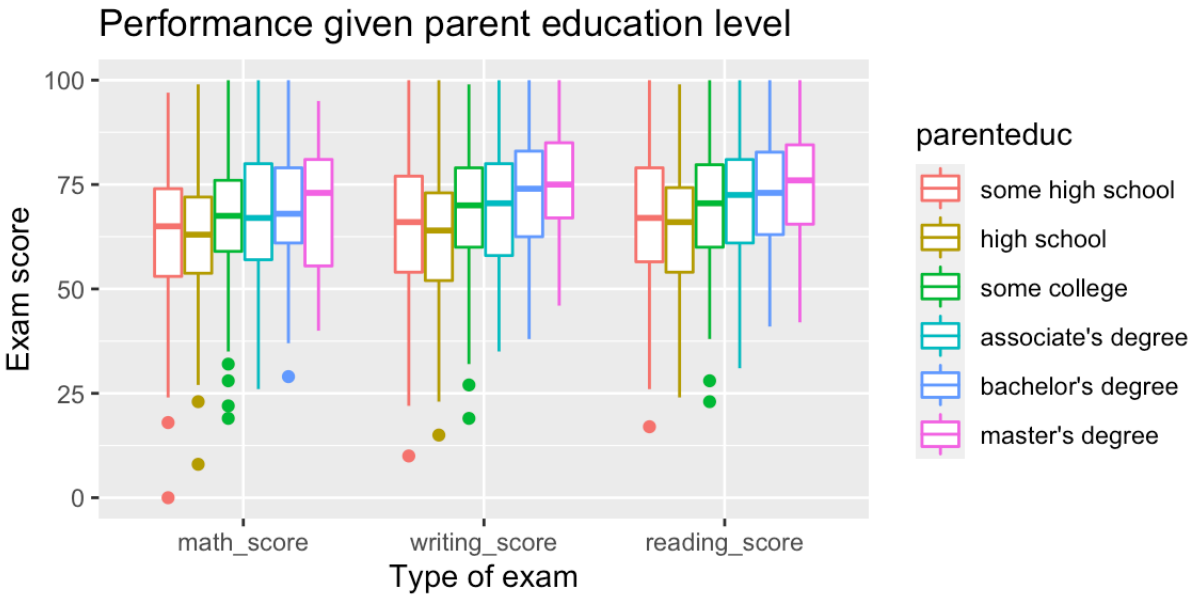
```
## # A tibble: 2 x 3
##   testprep      n  prop
##   <chr>     <int> <dbl>
## 1 none        642 0.642
## 2 completed   358 0.358
```

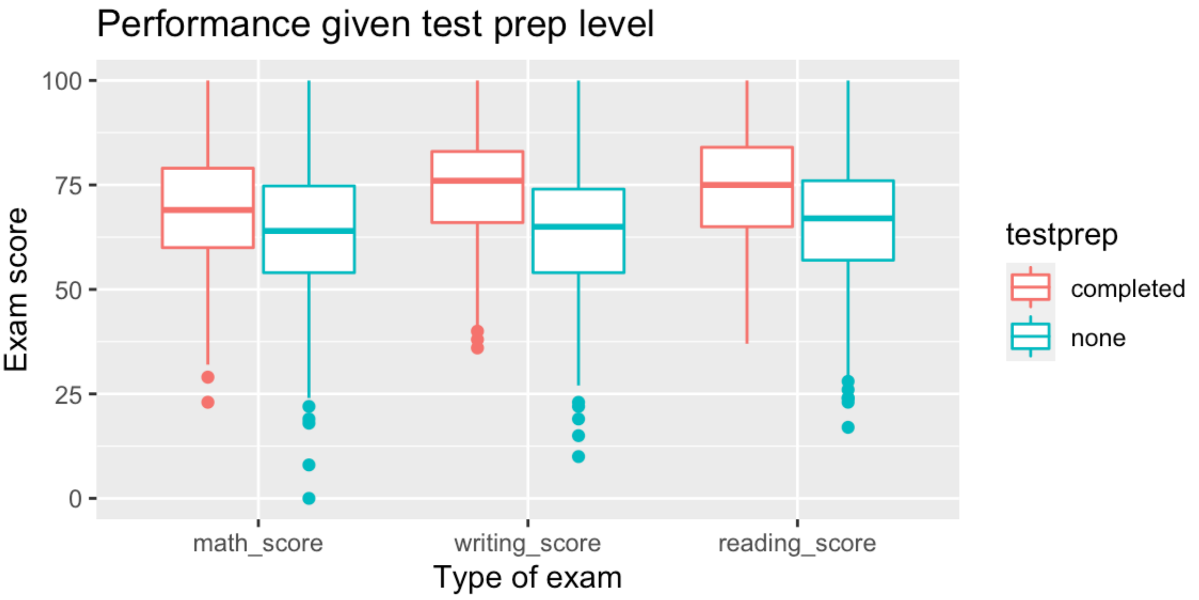
Data Visualizations:

Below we use a box plot to depict the variation of math, writing, and reading scores given the different descriptive variables. Also we calculate the average scores for the students by grouping them using the 5 descriptive variables about the student's background. From the average scores and the box plot one can see that students with certain backgrounds perform better than students with other backgrounds. For example, one would expect the best performing student's background to be one who has standard lunch, completes the test preparation, has parents who have a master's degree, is female, and is part of group E. One would expect the worst performing student's background to be one who has free/reduced lunch, has not completed the test preparation, has parent's who have high school education, is male, and is part of group A. The variable with the greatest difference between highest and lowest mean for exam scores is parent education. The variable with the smallest difference between highest and lowest mean for exam scores is gender.

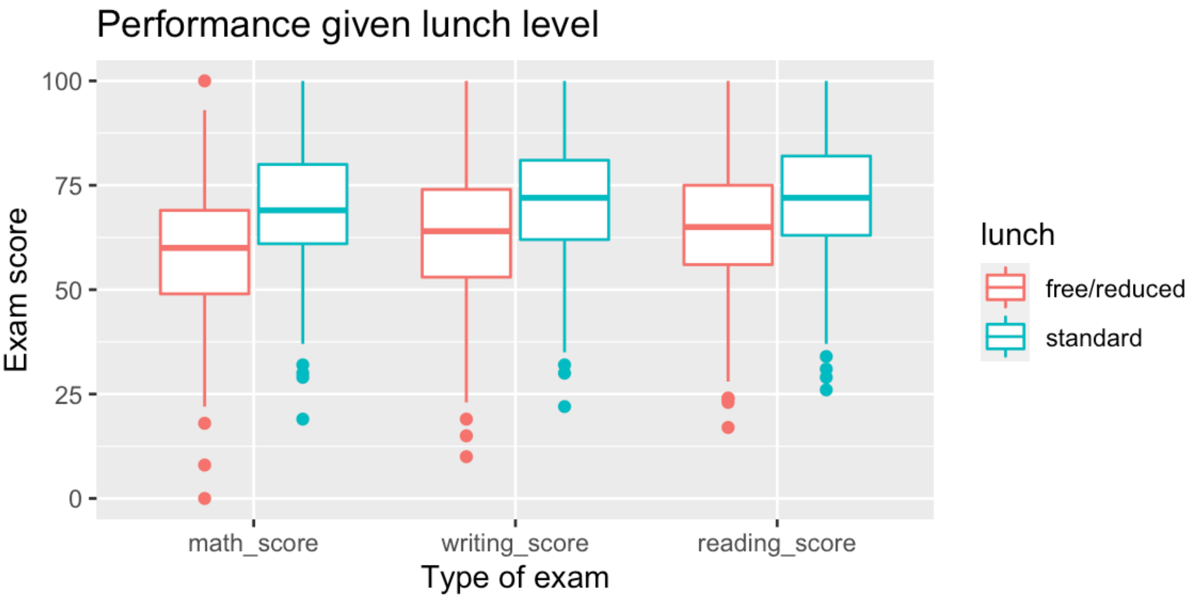
```
stu_perf_long_peduc<-stu_perf_edited%>%
  select(math_score,writing_score,reading_score,parenteduc)%>%
  melt(id = "parenteduc")

stu_perf_long_peduc %>% ggplot(aes(x=variable,y=value,color=parenteduc))+
  geom_boxplot()+
  xlab("Type of exam")+
  ylab("Exam score")+
  ggtitle("Performance given parent education level")
```

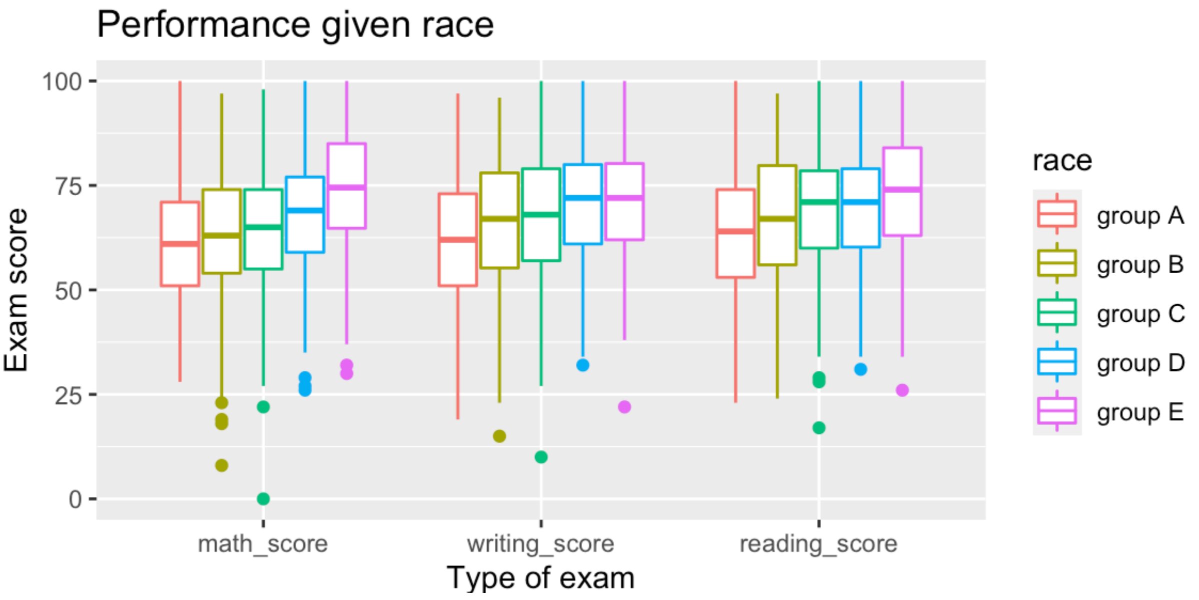

Performance given parent education level
Exam score
100
75
50
25
0
math_score
writing_score
reading_score
Type of exam
parenteduc
some high school
high school
some college
associate's degree
bachelor's degree
master's degree



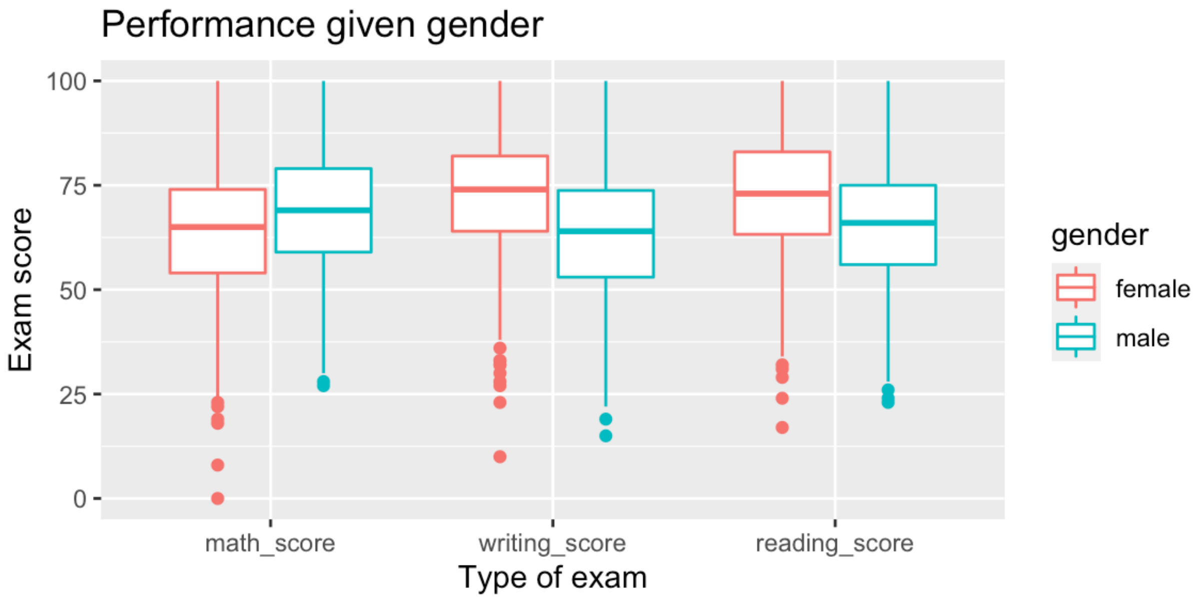
Performance given test prep level



Performance given lunch level



Performance given race



```
stu_perf_edited%>%
  group_by(lunch)%>%
  summarize(avg_math_score=mean(math_score),avg_writing_score=mean(writing_score),avg
_reading_score=mean(reading_score))%>%
  arrange(desc(avg_math_score,avg_writing_score,avg_reading_score))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   lunch        avg_math_score avg_writing_score avg_reading_score
##   <chr>                 <dbl>             <dbl>             <dbl>
## 1 standard               70.0              70.8              71.7
## 2 free/reduced           58.9              63.0              64.7
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   testprep  avg_math_score avg_writing_score avg_reading_score
##   <chr>              <dbl>             <dbl>             <dbl>
## 1 completed           69.7              74.4              73.9
## 2 none                64.1              64.5              66.5
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
## # A tibble: 6 x 4
##   parenteduc         avg_math_score avg_writing_score avg_reading_score
##   <fct>                       <dbl>             <dbl>             <dbl>
## 1 master's degree              69.7              75.7              75.4
## 2 bachelor's degree            69.4              73.4              73
## 3 associate's degree           67.9              69.9              70.9
## 4 some college                 67.1              68.8              69.5
## 5 some high school             63.5              64.9              66.9
## 6 high school                  62.1              62.4              64.7
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   gender avg_math_score avg_writing_score avg_reading_score
##   <chr>           <dbl>             <dbl>             <dbl>
## 1 male             68.7              63.3              65.5
## 2 female           63.6              72.5              72.6
```

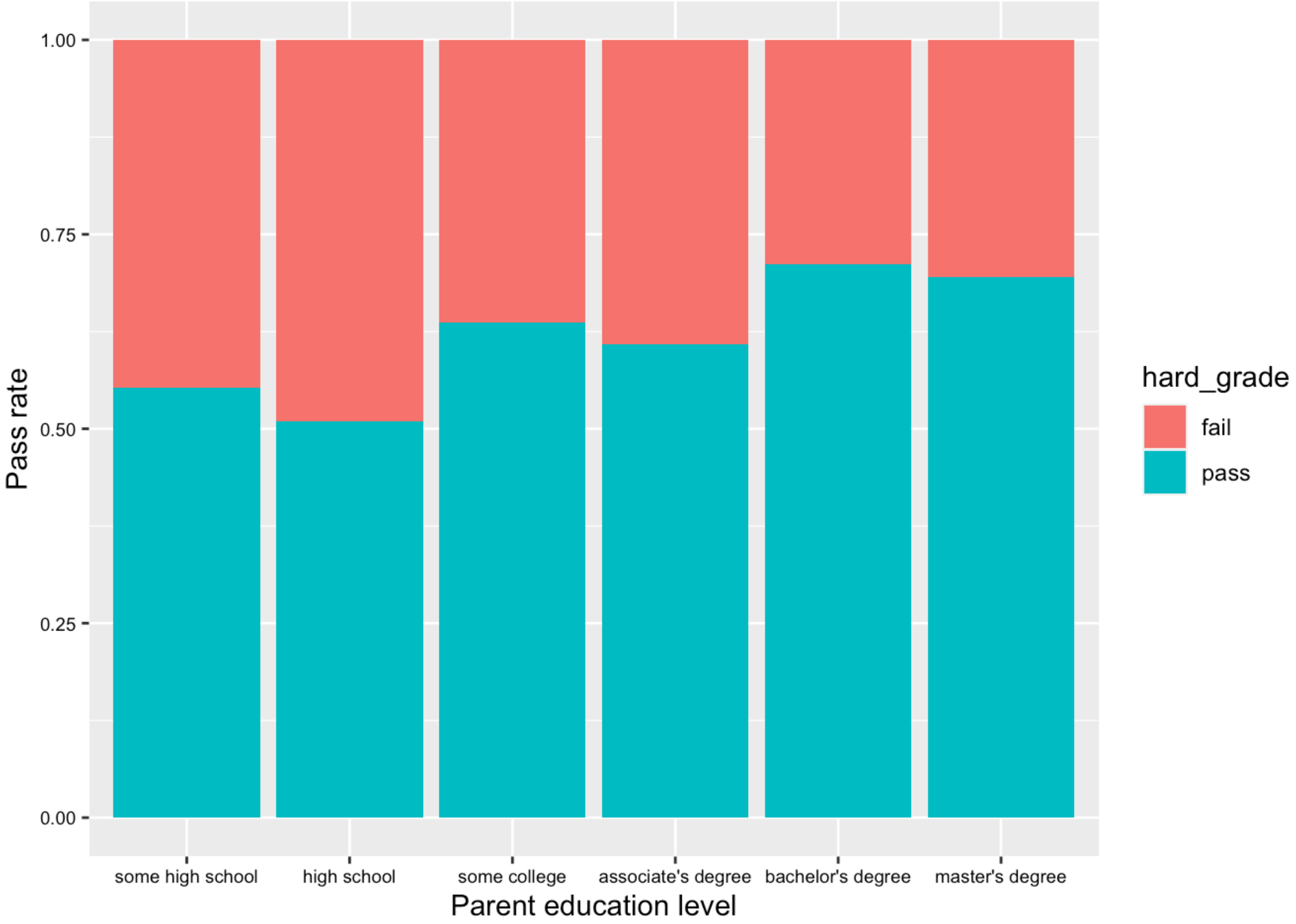
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

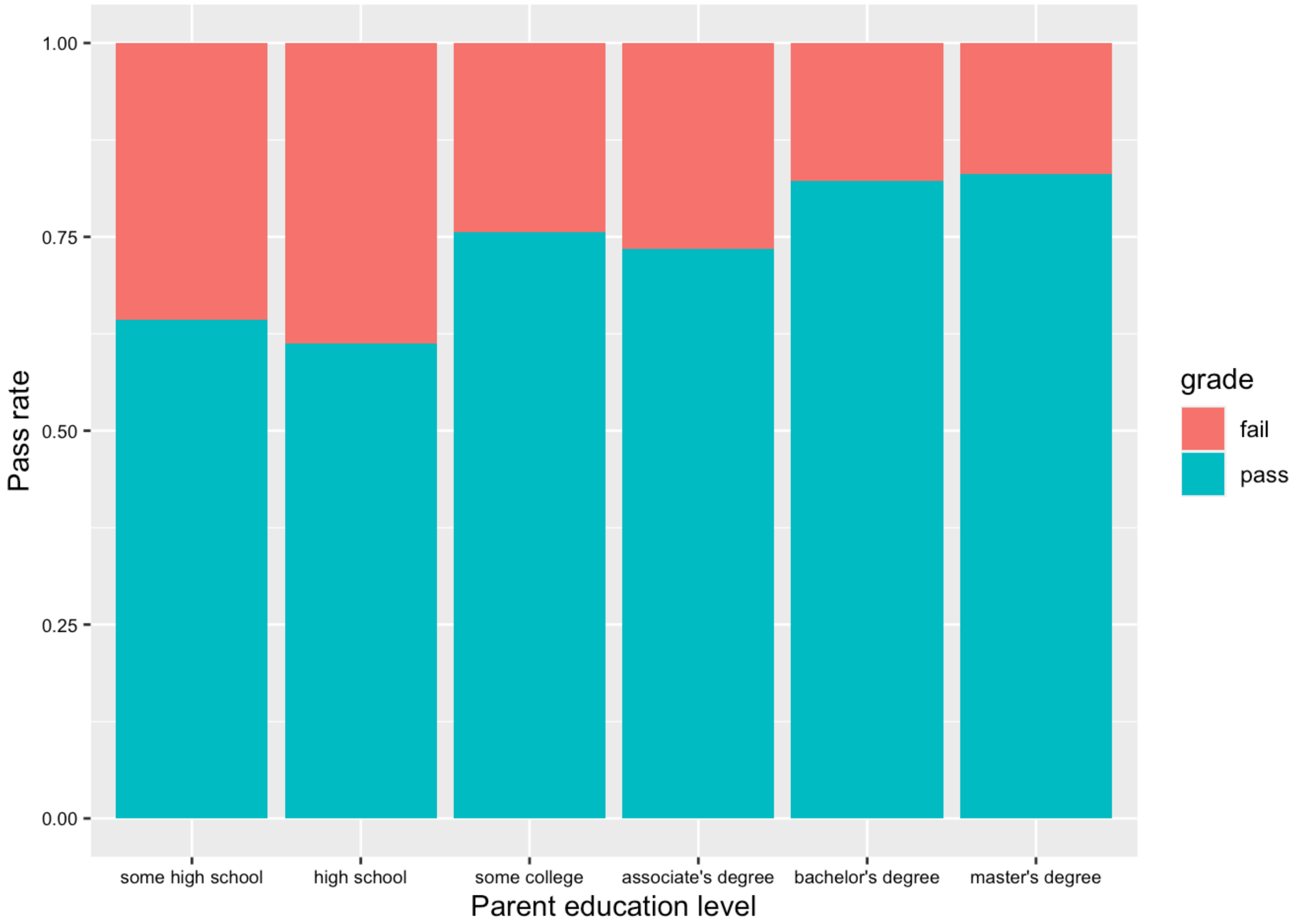
```
## # A tibble: 5 x 4
##   race    avg_math_score avg_writing_score avg_reading_score
##   <chr>            <dbl>             <dbl>             <dbl>
## 1 group E           73.8              71.4              73.0
## 2 group D           67.4              70.1              70.0
## 3 group C           64.5              67.8              69.1
## 4 group B           63.5              65.6              67.4
## 5 group A           61.6              62.7              64.7
```

Model for Pass Rate

In our previous examination of the different variables we noticed the greatest difference between highest and lowest mean for exam scores is parent education. In the bar charts we notice how the different levels in parent education affect the pass rate of the students in the lenient grading system and the hard system. The bar chart shows in the hard grading system students whose parents have a bachelor’s degree have the highest pass rate and students whose parents have a high school education have the lowest passing rate. In the lenient grading system students whose parents have a master’s degree have the highest pass rate and students whose parents have a high school education have the lowest passing rate. The pass rate in general is a lot lower in the hard system as it is less forgiving because it fails the student if they fail one class.

```
stu_perf_edited %>%
  ggplot(aes(x = parenteduc, fill = hard_grade))+
  geom_bar(position = "fill")+
  xlab("Parent education level")+
  ylab("Pass rate")+
  theme(axis.text = element_text(size=7,colour = "black"))
```





For the model we will be using a conditional inference tree. We will be using all 5 variables as predictors in this model in order to get the best result possible. We will be creating a model for both the lenient grading system and the hard grading system to see if we can predict whether a student passes or fails.

```
set.seed(123456)

cf_model1 <- cforest(grade ~ parenteduc+race+lunch+testprep+gender,  data = stu_perf_
fcts)

train1_pred <- stu_perf_fcts %>%
add_predictions(cf_model1)

#train1_pred %>% select(grade, pred)

train1_pred %>%
mutate(right = (grade == pred)) %>%
summarize(grade_pred_correct=mean(right))
```

```
## # A tibble: 1 x 1
##   grade_pred_correct
##                <dbl>
## 1              0.761
```

```
cf_model2 <- cforest(hard_grade ~ parenteduc+race+lunch+testprep+gender, data = stu_p
erf_fcts)

train2_pred <- stu_perf_fcts %>%
add_predictions(cf_model2)

#train2_pred %>% select(hard_grade, pred)

train2_pred %>%
mutate(right = (hard_grade == pred)) %>%
summarize(hard_grade_pred_correct=mean(right))
```

```
## # A tibble: 1 x 1
##   hard_grade_pred_correct
##                     <dbl>
## 1                   0.695
```

The model for the lenient grading system is right 76.1% of the time which is better than the base solution of 71.5%. The model for the hard grading system is right 69.5% of the time which is better than the base solution of 60.3%.

Conclusion

The first objective of this project was to create a model to predict whether the student will pass or fail their exam. With our conditional inference tree model we were able to generate results that were better than the baseline solution for both the lenient and hard grading system. One thing to keep in mind is we do not have a separate test data set so we will only be able to test it with the training set. This means our model will perform better than it would for a test data set. The other objective was to see how the different variable's affect the level student's test scores. We found out the level of the student's parent education has the greatest effect and gender has the smallest effect. We can see that a student's background is indicative of how a student will perform. Although the school can't change the a student's race, gender, parent education level or lunch level, the school can work on making sure everyone completes the test preparation as it did affect how well the student performed. It is important for the school to acknowledge that some students in the school will be at a disadvantage due to factors they can not control and the school should do its best to make a difference where it can.