Students Performance Project

Moaz M. Salaheldeen

2023-11-28

Introduction

Dataset

This dataset shows the Marks secured by students in different subjects, with other information about each student's behavior (Kind of lunch they have, test preparation course completion) in addition to information related to gender, race and parent's education level. Dataset is available for public in **Kaggle**

Why did I choose this Dataset?

One of the main reasons I chose this dataset is its size, it is relatively small. This is clear from the size of the source data file (one csv file -less than 10KB). And also proved to be true when seeing that the number of records is 1000 records only. While this is surely very limited number of records in real life, it perfectly fits our goals in a project with learning objectives. This size allows trying different functions and models that couldn't be executed on a personal laptop with a larger datasets -like MovieLens 10 Million Records dataset-.

Goals of the Project

The aim of this project is to explore Students Performance dataset. Our work will focus on trying to find out which and how of these factors affect the students results, and then will try to build a model that participate exam scores for a student given these factors.

To do so, we will do these steps:

- Data Importing and cleansing
- Data Exploring and Visualization
- Modeling
- Results and Conclusions

Step 1: Data Importing and Cleansing

```
studentsDF<- read.csv('StudentsPerformance.csv') #File must be under WD
```

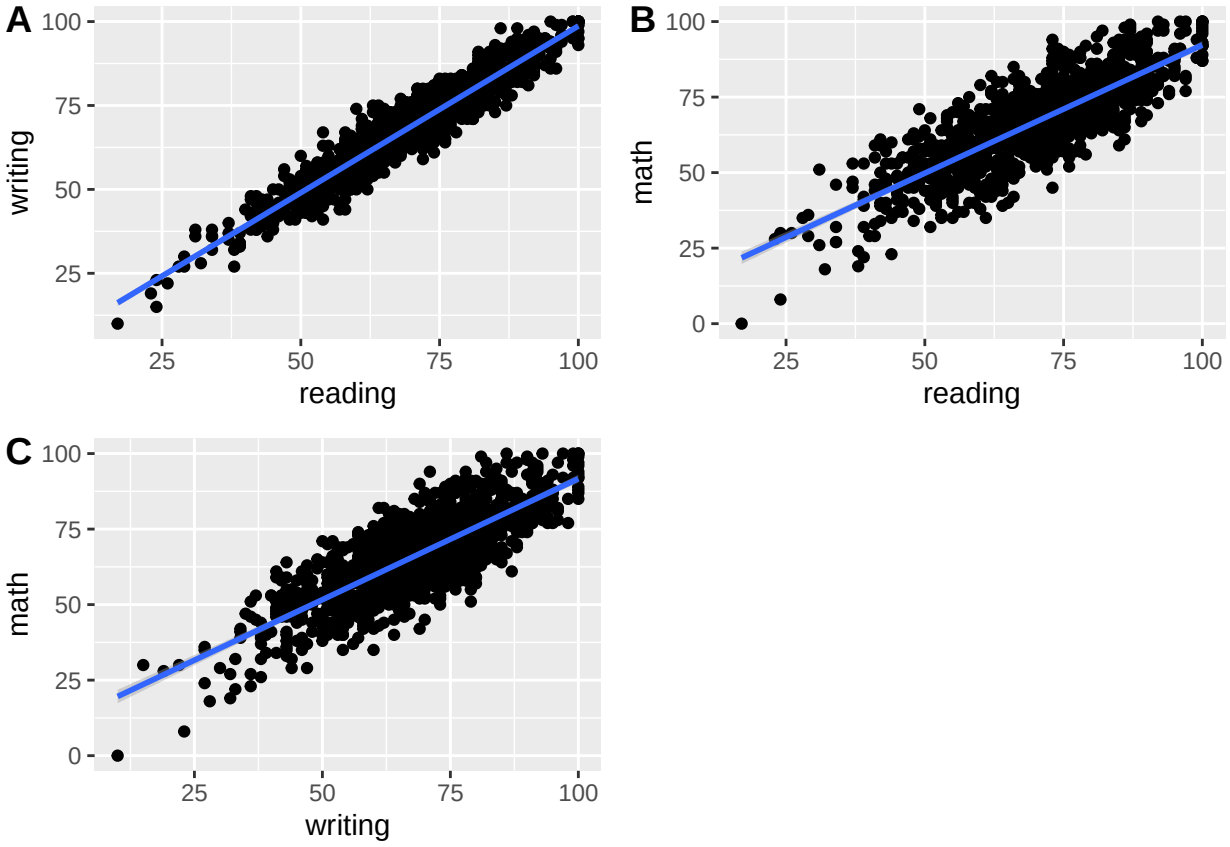
Dataset consists of 1000 record, 8 columns (5 related to properties associated with students and 3 test results:Math, reading and writing). The data is complete, no missing data in any column/value. All columns are of type 'Char'. We can see that the 5 properties are Categorical, while the scores are numeric values between 0 and 100.(Actually only in Math we have a Zero score, but it is logical to assume that 0 is possible exam mark)

Step 2: Data Visualization (& Insights)

Check relation between the three scores

First, We'll examine the three scores and check if there is any kind of relation between them (Note: for more readability I renamed the columns names to shorter names)

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




```
##
##  Pearson's product-moment correlation
##
## data:  studentsDF$reading and studentsDF$writing
## t = 101.23, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9487506 0.9597921
## sample estimates:
##       cor
## 0.9545981
```

```
##
##  Pearson's product-moment correlation
```

```
##
## data:  studentsDF$reading and studentsDF$math
## t = 44.855, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7959276 0.8371428
## sample estimates:
##       cor
## 0.8175797


##
##  Pearson's product-moment correlation
##
## data:  studentsDF$writing and studentsDF$math
## t = 42.511, df = 998, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7794321 0.8236517
## sample estimates:
##      cor
## 0.802642
```

As seen clearly in the diagrams there is a linear relation between each two subjects. This is also confirmed by the Pearson correlation factor,

$$r = \frac{N\sum XY - (\sum X \sum Y)}{\sqrt{[N\sum x^2 - (\sum x)^2][N\sum y^2 - (\sum y)^2]}}$$

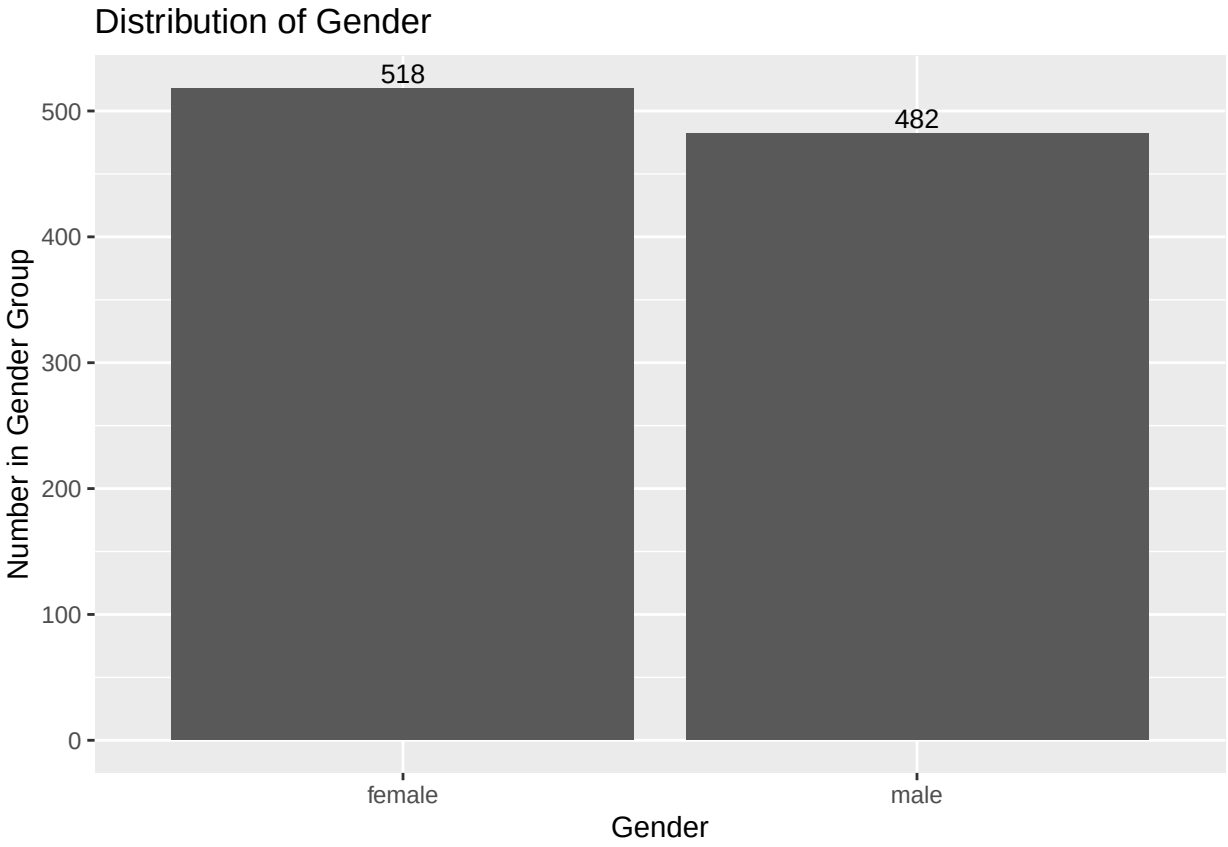
This indicates there is High correlation between all subjects, slightly -and as expected logically- higher between "reading" and "writing"; 0.9545981 between "reading" & "writing", compared to 0.8175797 and 0.802642 between "Math" and each of "reading" and "writing" respectively.

To simplify our work, we will benefit from this high correlation and create new variable at the dataset that represent the average of the three subjects

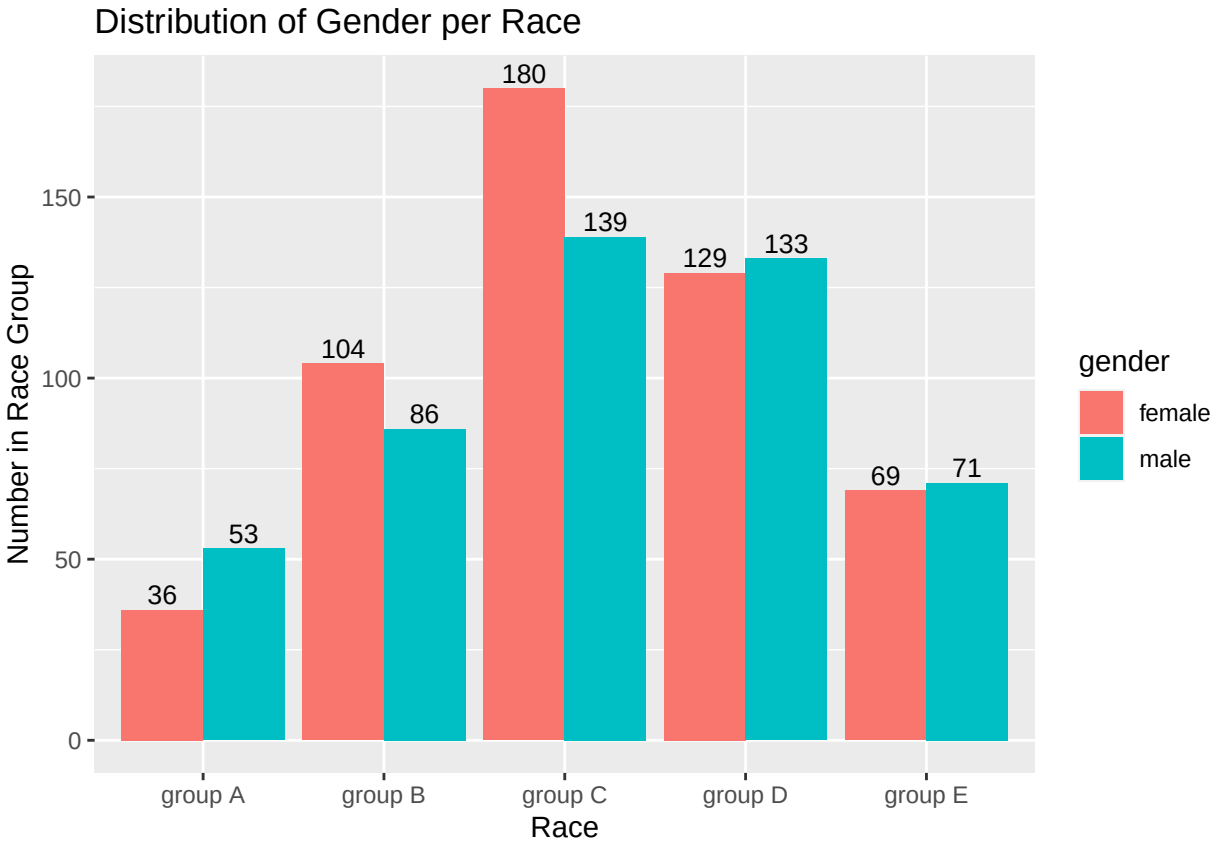
```
studentsDF<- studentsDF |> mutate(avg_score=(math+ reading + writing)/3)
```

Check distribution of Inputs

When we check "Gender" column of the data to see if there is a dominant value, we can see that




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

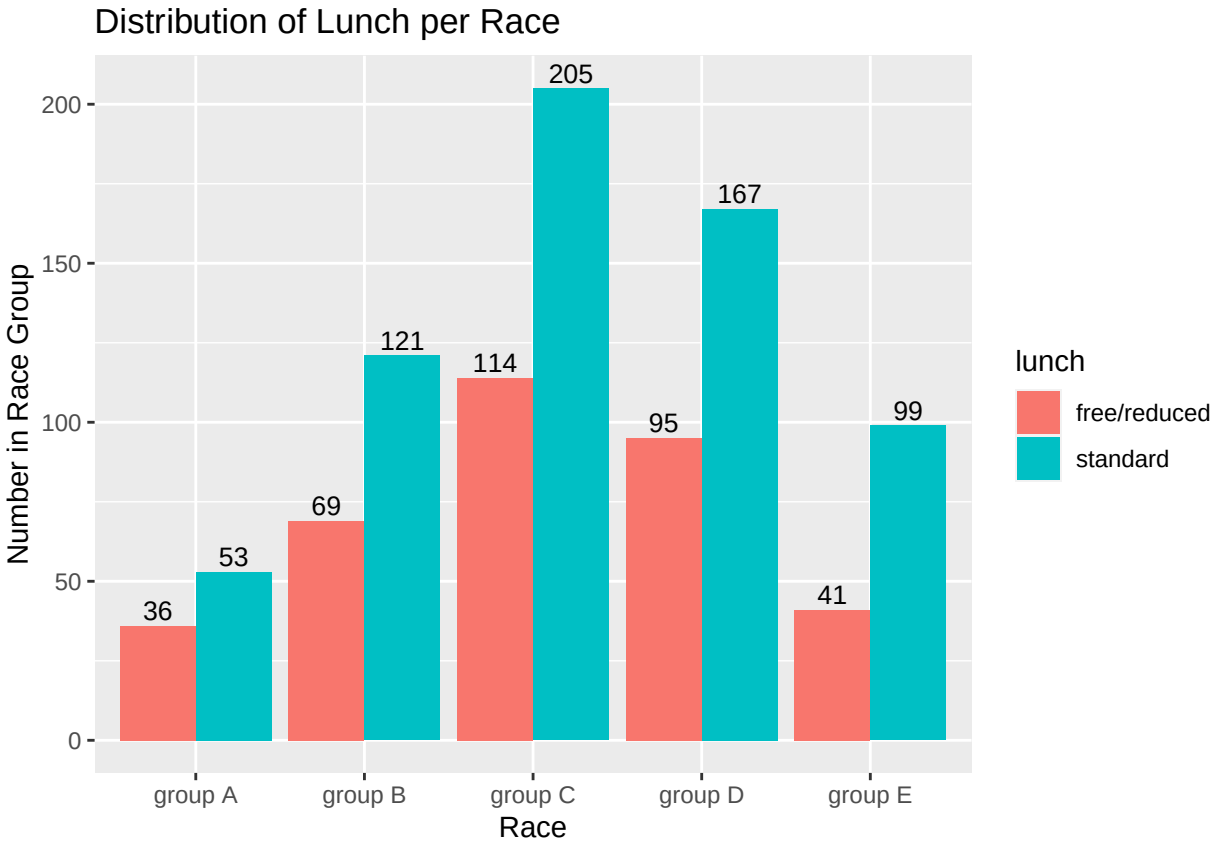


There are little bit "Females" than "Males", little bit more than the half (518), and the percentage of females to males in each "race" doesn't show dominant gender (Males was around 43.5% in their lowest percentage - in group C and females were 40.44% in their lowest percentage - in group A).

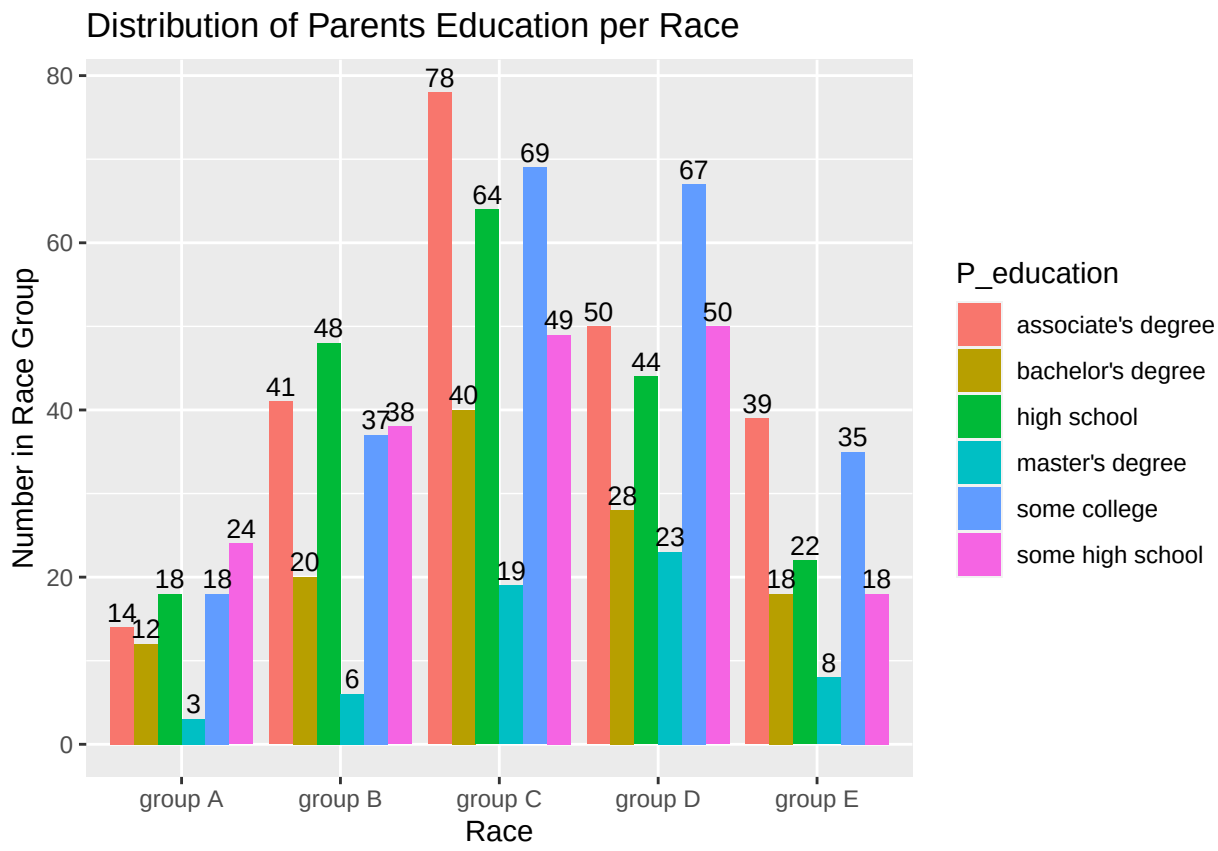
Check RACE effect

Now, it is worth checking if "race" affect other factors, for example if it affect lunch, parent education level or taking prep exam -due to possible economical differences between different group of races-

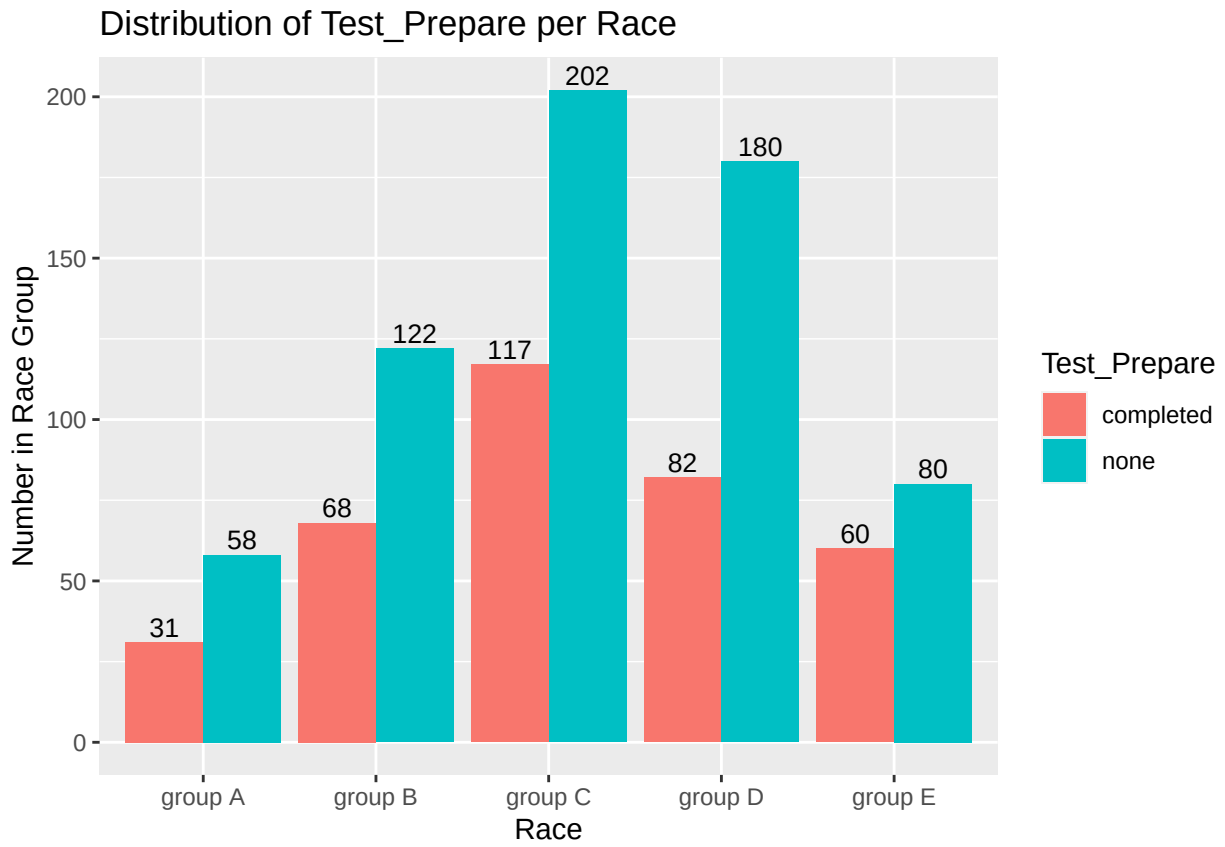
```
## `summarise()` has grouped output by 'race'. You can override using the
## `.groups` argument.
```




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




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



Looking at these diagrams, we can see that more Students in all races get 'standard' type of lunch than 'free/reduced' lunch, and the percentage doesn't differ much from a group to another (it ranges from 59.6% in group A as the lowest percentage to 70.7% in group E as the highest).

Regarding the Educational level, it seems also not highly affected by the race, but as it not very clear visually, let's group the Educational level into two Main groups: 'Has degree' and 'No degree'.

```
studentsDF |> group_by(race) %>% summarise(Number_in_Grp = n(),
                            has_degree = sum(str_count(P_education,"degree")>0, na.rm = TRUE),
                            degree_Perc =has_degree/Number_in_Grp )
```

```
## # A tibble: 5 x 4
##   race    Number_in_Grp has_degree degree_Perc
##   <chr>           <int>      <int>       <dbl>
## 1 group A            89         29       0.326
## 2 group B           190         67       0.353
## 3 group C           319        137       0.429
## 4 group D           262        101       0.385
## 5 group E           140         65       0.464
```

It confirms the previous statements as the number of students with parents holding a degree ranges from 32.6% in group A as the lowest to 46.4% in group E as the highest- **Again**.

Preparing Exam feature gave similar results. In all groups the number of students completed the exam ranges from 31.3% in group D -as the lowest percentage- to 42.9% in group E as the highest **Again**.

Another check to confirm the previous results, is to apply the χ^2 test

```r
chisq.test(as.factor(studentsDF$race), as.factor(studentsDF$lunch))
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.factor(studentsDF$race) and as.factor(studentsDF$lunch)
## X-squared = 3.4424, df = 4, p-value = 0.4867
```

```r
chisq.test(as.factor(studentsDF$race), as.factor(studentsDF$P_education))
```

```
##
##  Pearson's Chi-squared test
##
## data:  as.factor(studentsDF$race) and as.factor(studentsDF$P_education)
## X-squared = 29.459, df = 20, p-value = 0.07911
```

```r
chisq.test(as.factor(studentsDF$race), as.factor(studentsDF$Test_Prepare))
```

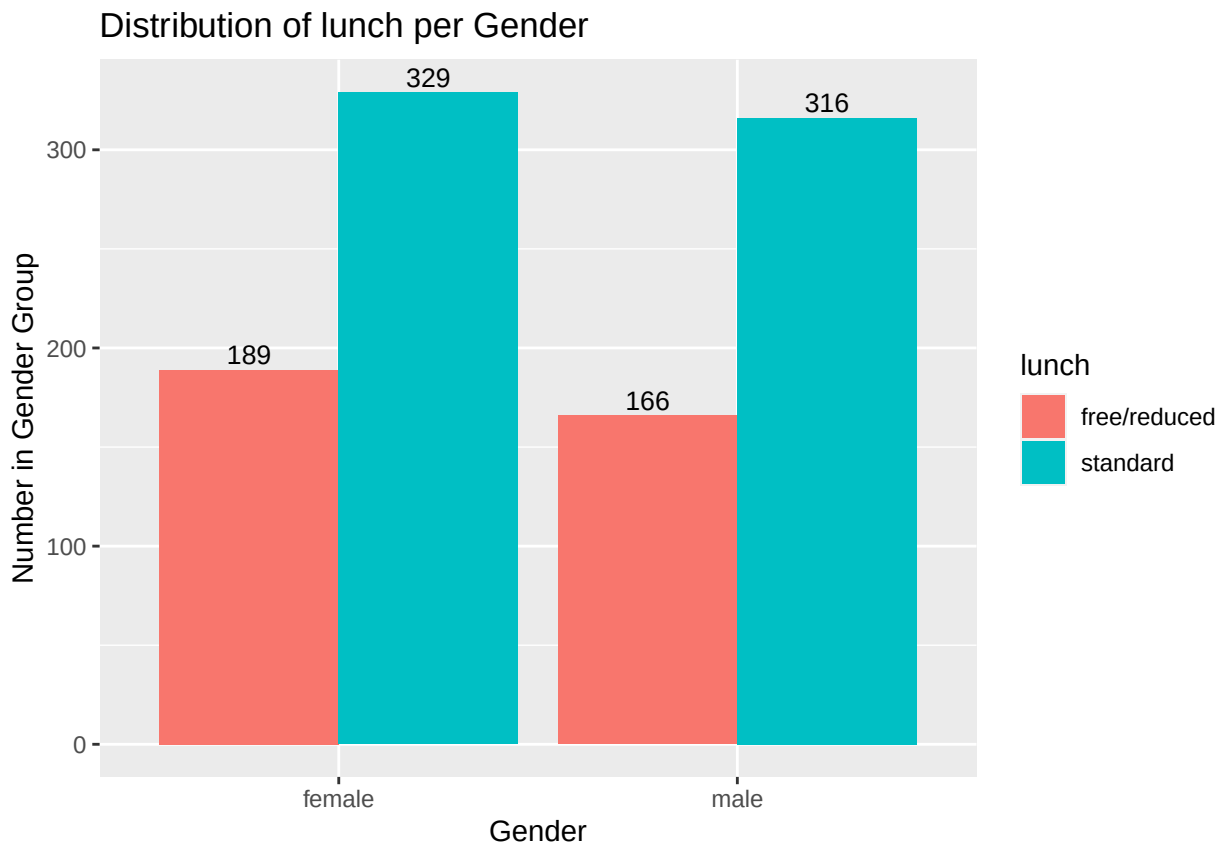
```
##
##  Pearson's Chi-squared test
##
## data:  as.factor(studentsDF$race) and as.factor(studentsDF$Test_Prepare)
## X-squared = 5.4875, df = 4, p-value = 0.2408
```

All p-values are larger than 5%, so we cant reject Null hypothesis and we cant say there is a direct relation. This match the visual results. Still, note that the smallest p-Value is for Educational level feature much less for Educational Levels.

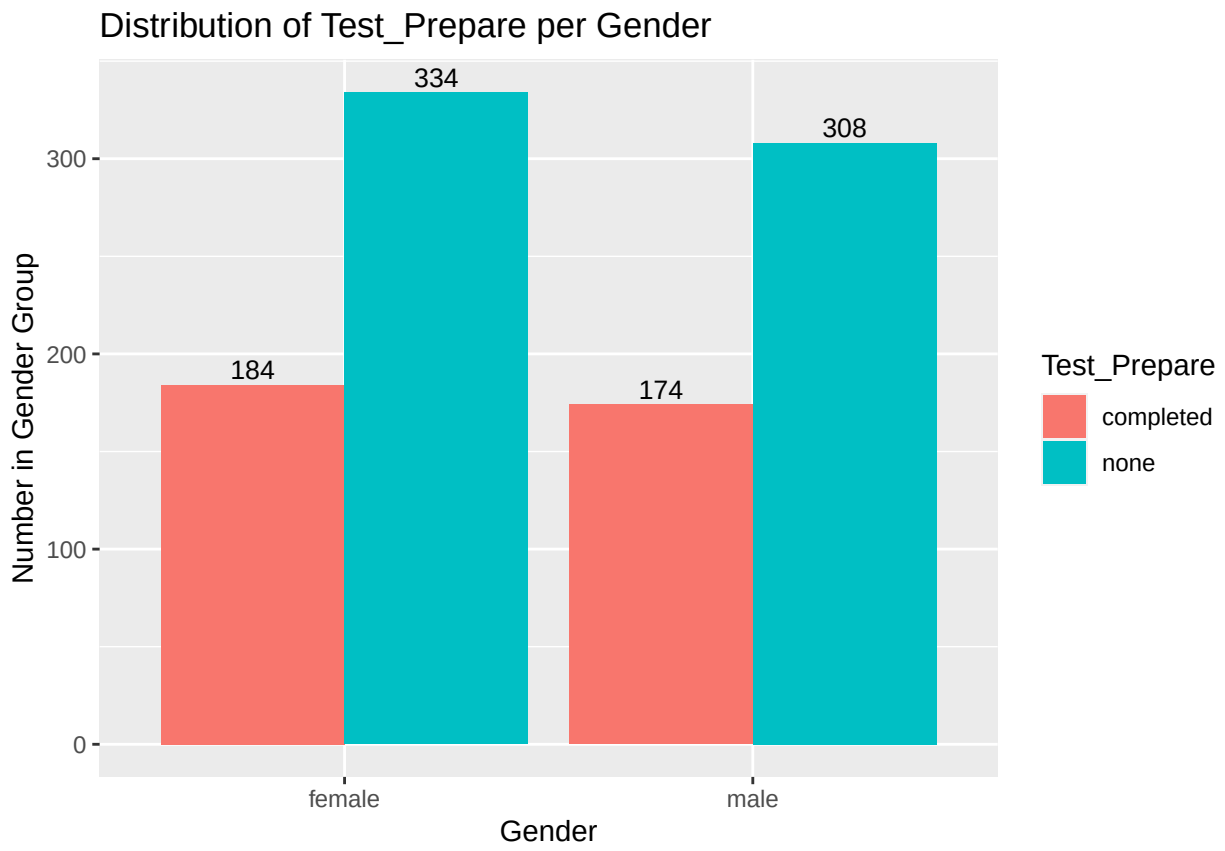
Check GENDER effect

Now, we check for a probability that a certain gender may be more committed to taking preparation exam or having 'standard lunch'.

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




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



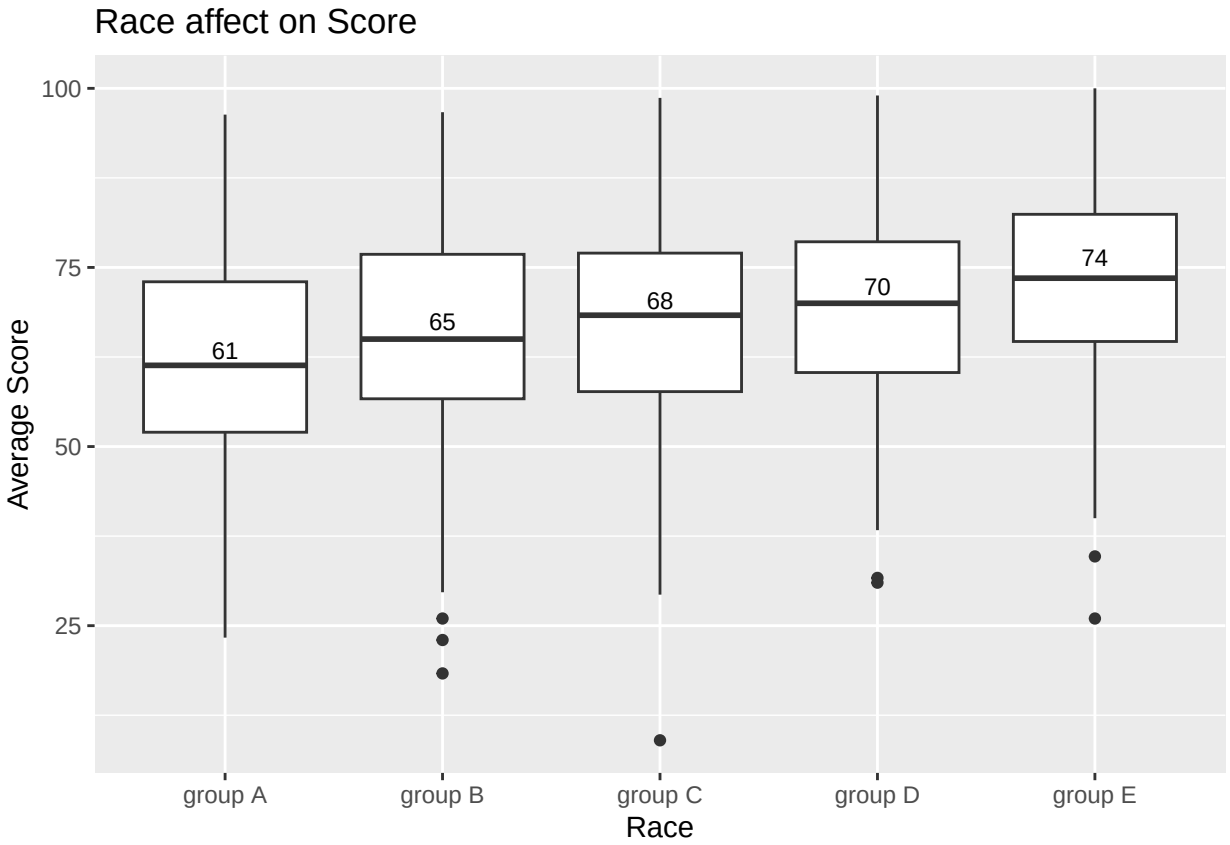
We see that the behavior and percentage is very similar, so we cant say that gender has relation with any of these two factors. The high p-Value resulted from running χ^2 test, confirms the same

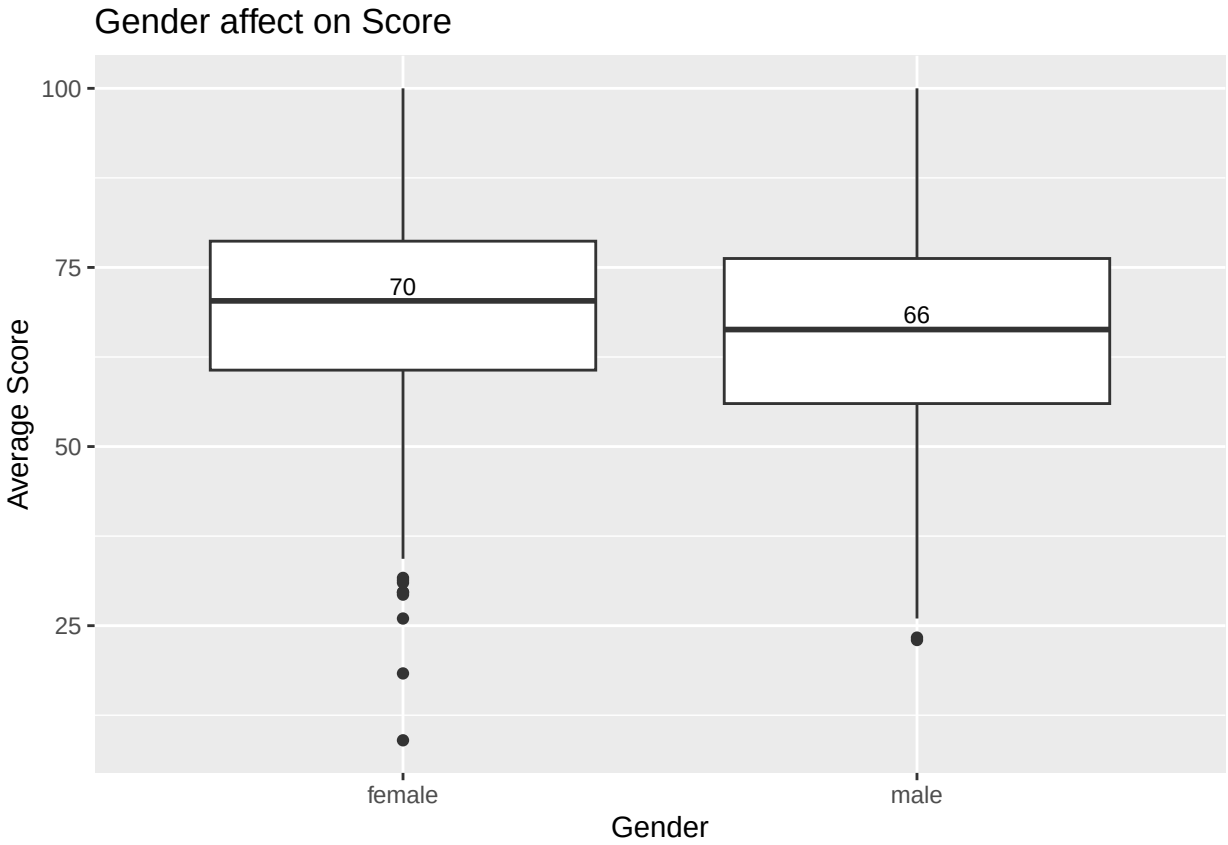
```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  as.factor(studentsDF$gender) and as.factor(studentsDF$lunch)
## X-squared = 0.37174, df = 1, p-value = 0.5421
```

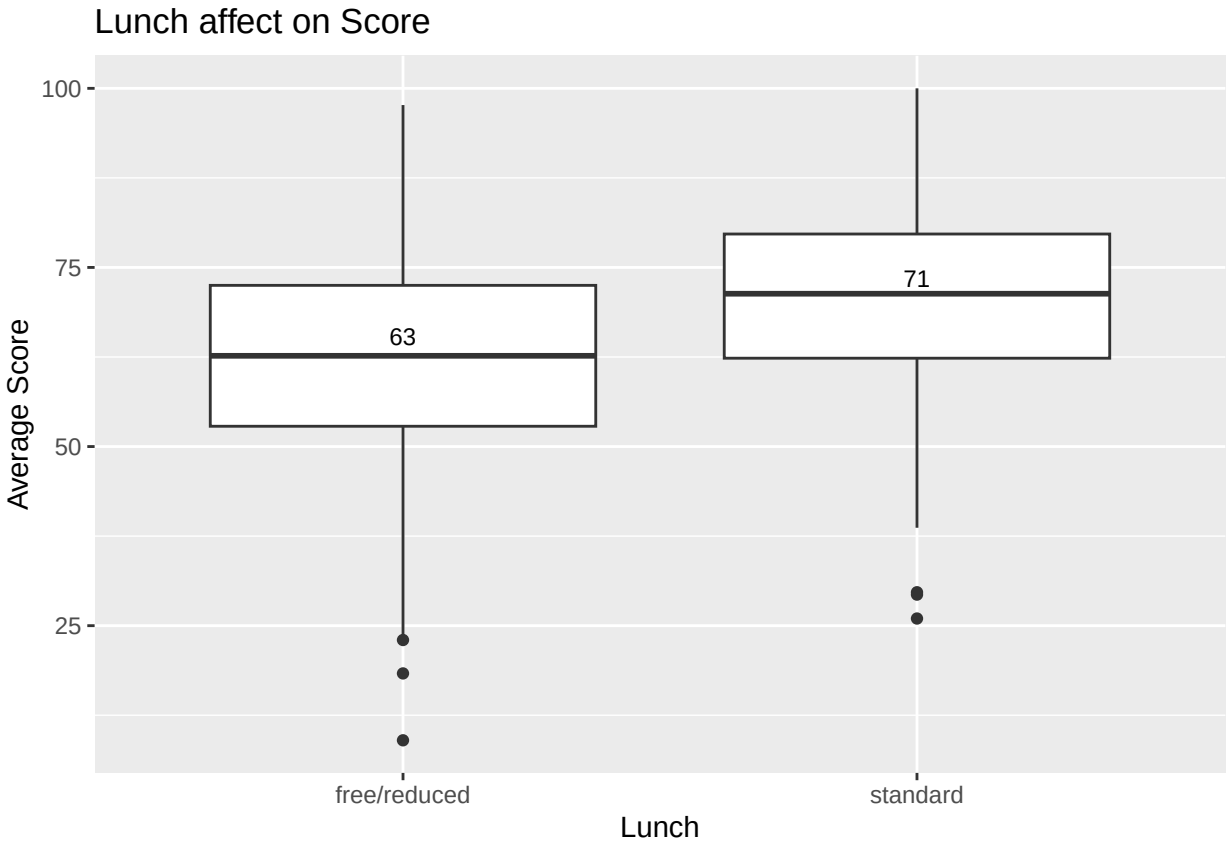
```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  as.factor(studentsDF$gender) and as.factor(studentsDF$Test_Prepare)
## X-squared = 0.015529, df = 1, p-value = 0.9008
```

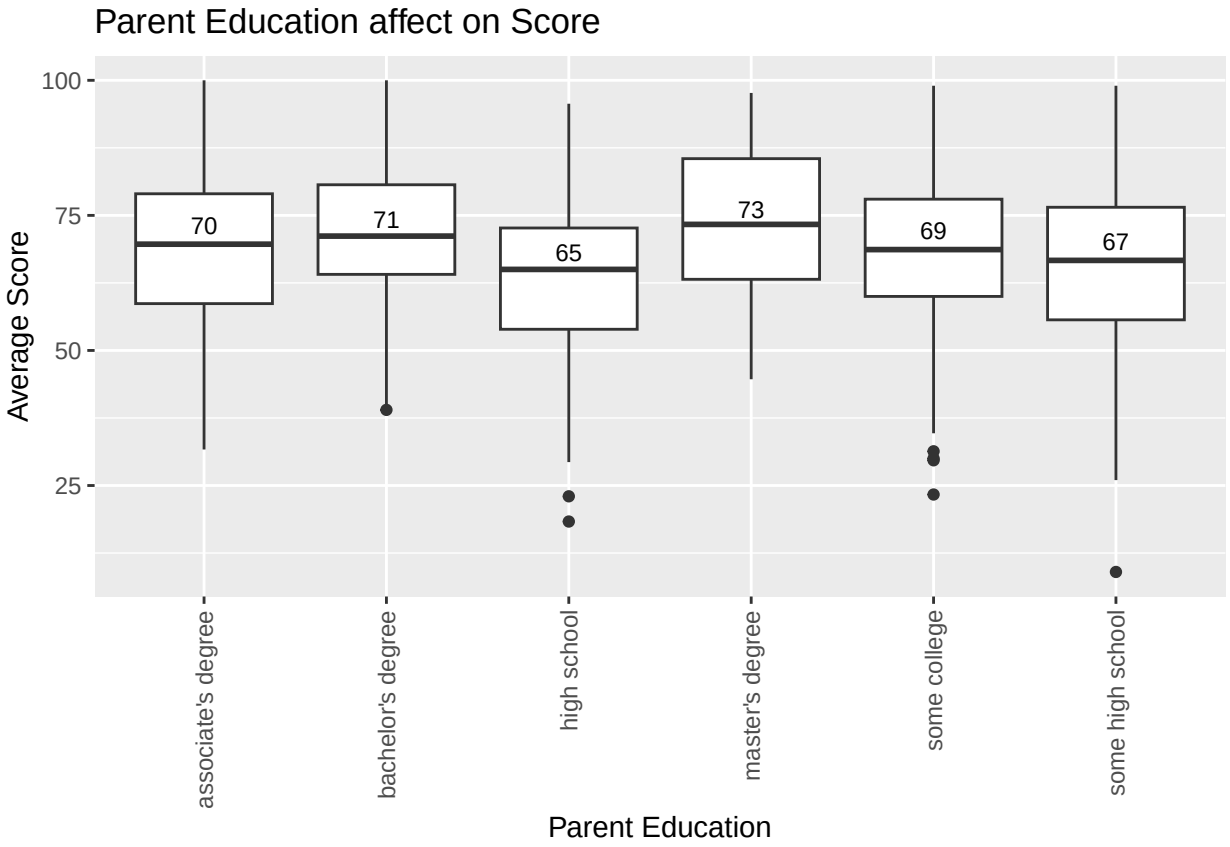
Check Factors effect on Average Score

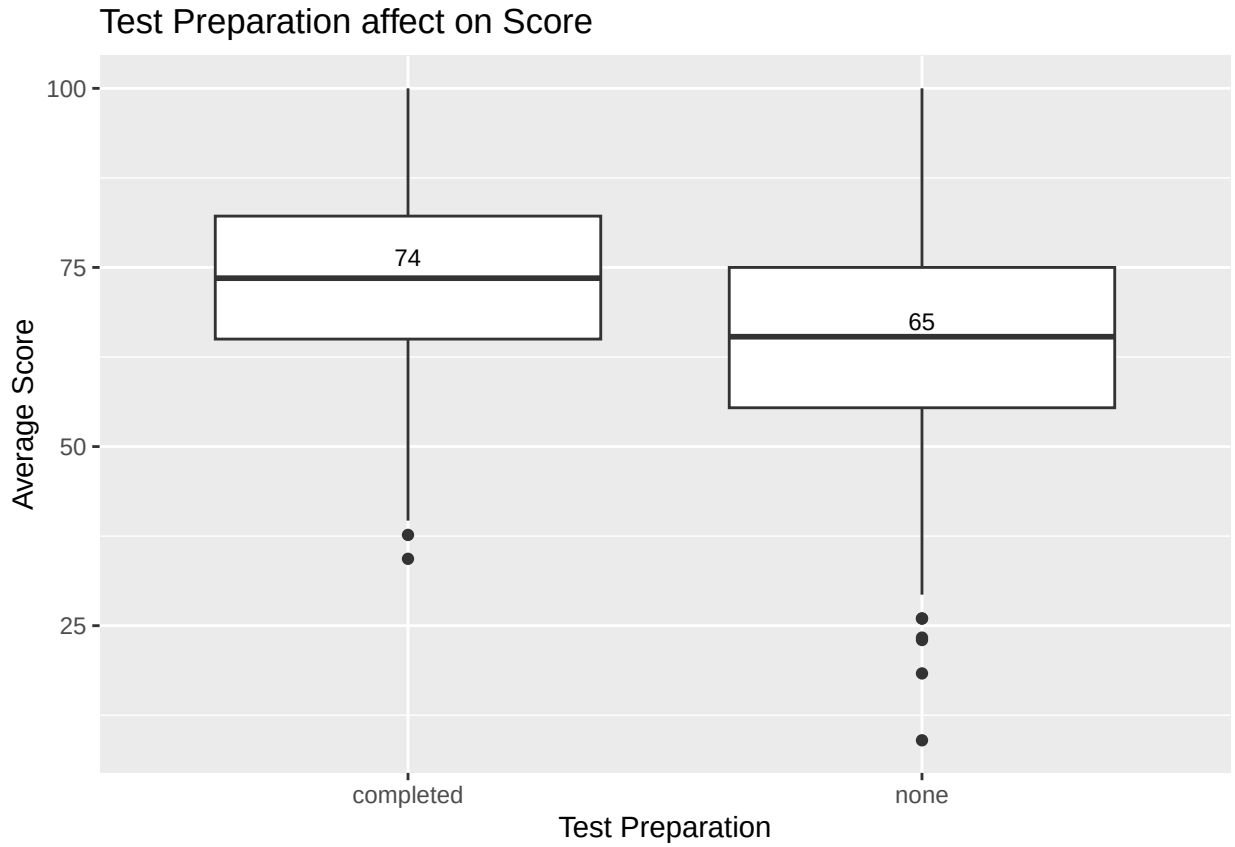
It is time now to check the relation between each feature and the Average score, that represents the three subjects' scores


Race affect on Score
Average Score
100
75
50
25
61
65
68
70
74
group A
group B
group C
group D
group E
Race


Gender affect on Score
100
75
50
25
Average Score
70
66
female
male
Gender


Lunch affect on Score
100
75
50
25
Average Score
63
71
free/reduced
standard
Lunch


Parent Education affect on Score
Average Score
100
75
50
25
70
71
65
73
69
67
associate's degree
bachelor's degree
high school
master's degree
some college
some high school
Parent Education



Race: Apparently, it has some affect on the results. Group A is the smallest median with only 61, while group E is the one with the highest median 74. This comes in compliance with the Analysis of input features, where this group was highest in almost all features.

Gender: Females students are doing little bit higher in Median than male students (70 compared to 66). Diagram shows also they are less deviated from the median

Lunch: Students having standard lunch do better in Exams. That can be seen easily from the diagram comparing the medians; 71 to 63. Diagram shows also they are less deviated from the median

Degree: Students with Parents holding degrees tend to do little bit higher than others, but this factor has the weakest impact on results. Their medians were 70, 71, 73 for associate's degree, bachelor's degree and master degree respectively. While others ranged from 65 to 69.

Test Preparation: Students completed Test Preparation do better in Exams. Their medians are 74 compared to 65 for those who didn't. This is the second strongest factor after Lunch.

Correlation equations shows the same results:

Factor	Correlation
Lunch	29%
Test Preparation	25%
race	18%
gender	13%
Parent Education	7%

Step 3: Data Preparation and Modeling

Data Preparation

20% of the dataset will be kept for the Evaluation of the final model. 20% of the remaining is considered to be the test set and the other 80% is the training set.

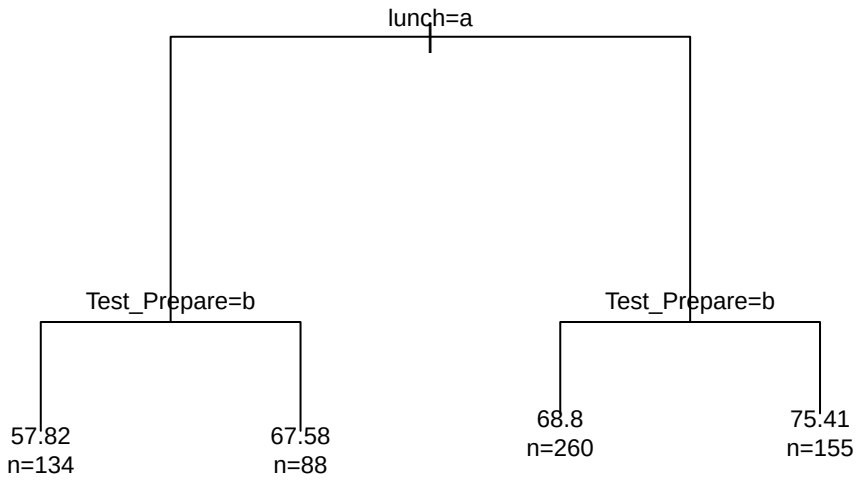
The Evaluation function used to compare models is the standard RMSE: $RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(d_i - f_i)^2}$

Modeling

As all factors are Categorical, we'll try Regression Tree model. I tried two the regression trees two times; One based on the most significant factors and the other is based on all factors

First: The model we build here is based on the most significant factors: lunch, Test_Prepare, and race

```
## n= 637
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
## 1) root 637 123585.40 67.93093
##   2) lunch=free/reduced 222  43763.97 61.69219
##     4) Test_Prepare=none 134  22742.38 57.82338 *
##     5) Test_Prepare=completed 88  15961.83 67.58333 *
##   3) lunch=standard 415  66558.58 71.26827
##     6) Test_Prepare=none 260  40422.00 68.79744 *
##     7) Test_Prepare=completed 155  21886.69 75.41290 *
```



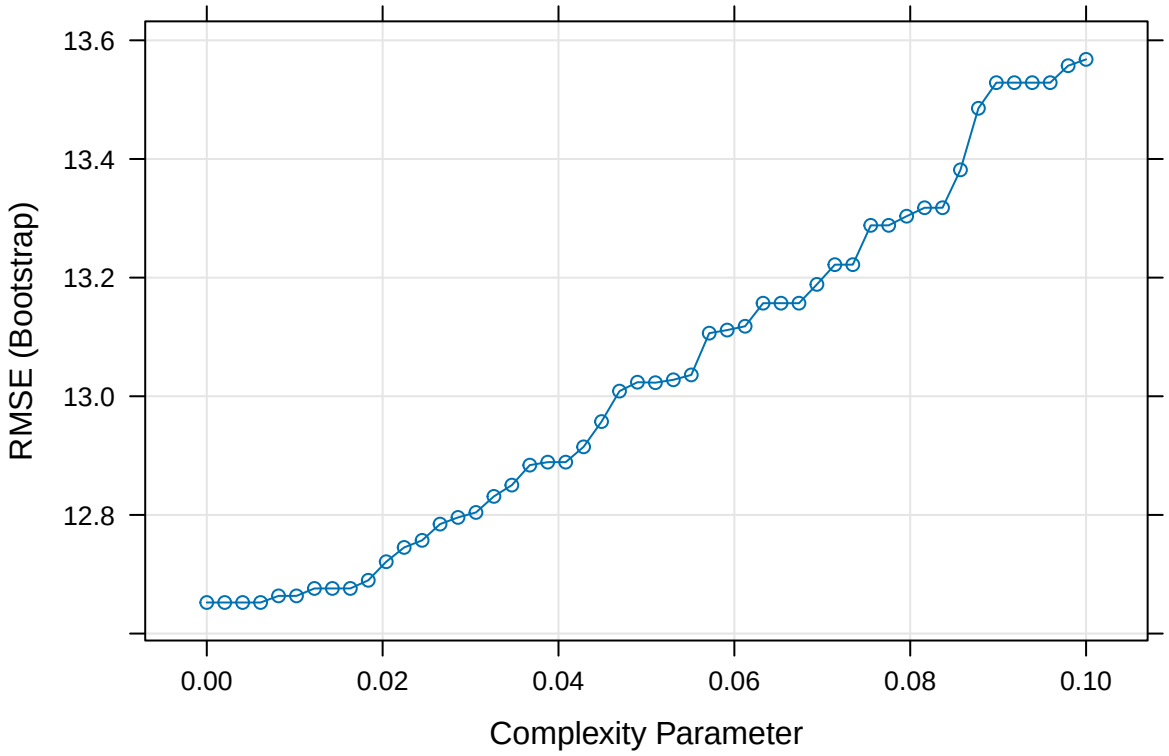
As can be seen from the tree, the effective factors are Lunch and Test_Preparation. Race had no place in the tree, indicating its effect is implicit inside the previous two.

I applied Cross Validation concept to reach the best parameters. I omitted Race to make it simpler

```
train_rpart_top <- train(avg_score ~ lunch+Test_Prepare,
                     method = "rpart",
                     tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 50)),
                     data = work_train_data)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

```
plot(train_rpart_top)
```




```
train_rpart_top$bestTune
```

```
##           cp
## 4 0.006122449
```

Now, applying this model on the test data, and adding to the table where we keep different models results, we can see very slight enhancement:

```
## # A tibble: 2 x 2
##   method                             RMSE
##   <chr>                             <dbl>
## 1 "Just the average"                 13.7
## 2 "Regression Tree- Top Factors: "   13.2
```

Second:

Now we build a module after adding the other two factors; Gender and Parent Education, and repeat the above steps

```
## # A tibble: 3 x 2
##   method                             RMSE
##   <chr>                             <dbl>
## 1 "Just the average"                 13.7
## 2 "Regression Tree- Top Factors: "   13.2
## 3 "Regression Tree- All Factors: "   13.2
```

These factors didntnt enhance the model.

In order to reach better results, I tried other models.

RandomForest

We tried the Random Forest technique, using all factors, and applied the results to the Evaluation Function.

```
## # A tibble: 4 x 2
##   method                             RMSE
##   <chr>                             <dbl>
## 1 "Just the average"                 13.7
## 2 "Regression Tree- Top Factors: "   13.2
## 3 "Regression Tree- All Factors: "   13.2
## 4 "Random Forest "                   12.6
```

Knn

To apply KNN model, we will first apply Cross Validation to get the best parameter and then apply the model

```
##     k
## 55 58
```

```
## # A tibble: 5 x 2
##   method                             RMSE
##   <chr>                             <dbl>
## 1 "Just the average"                 13.7
## 2 "Regression Tree- Top Factors: "   13.2
## 3 "Regression Tree- All Factors: "   13.2
## 4 "Random Forest "                   12.6
## 5 "KNN"                              12.6
```

Having built three models, will **Ensemble** the models together. In our case, ensemble can be done by just taking the average. I'll take the average between the best two models: KNN and Random Forest

```
y_hat_av<- ( y_hat_rf + y_hat_knn)/2
```

This gives us these final results

```
## # A tibble: 6 x 2
##   method                             RMSE
##   <chr>                             <dbl>
## 1 "Just the average"                 13.7
## 2 "Regression Tree- Top Factors: "   13.2
## 3 "Regression Tree- All Factors: "   13.2
## 4 "Random Forest "                   12.6
## 5 "KNN"                              12.6
## 6 "Average"                          12.6
```

Step 4: Results & Conclusion (The table)

For final Evaluation for the model, we applied the Ensemble model on the 20% of the dataset we split in the beginning, and calculated the RMSE

```
## [1] 14.38573
```

As expected, This gave higher RMSE (14.38).

Appendix: What if we removed OutLliars from the dataset before starting work

When we plotted the exam results for each subject (Box- plot), there were outliars in each one. We ignored this fact on the previous work, and worked on all data. But what if we try the same models but after removing these points from the dataset, to see if removing outliars has any effect on results.

Let's start by defining the outliars points in each diagram and removed the duplicate

```
outR<-boxplot.stats(studentsDF$reading)$out
outR_ind <- which(studentsDF$reading %in% c(outR))

outW<-boxplot.stats(studentsDF$writing)$out
outW_ind <- which(studentsDF$writing %in% c(outW))

outM<- boxplot.stats(studentsDF$math)$out
outM_ind <- which(studentsDF$math %in% c(outM))

Indexes_to_remove <- (outR_ind)
Indexes_to_remove <- Indexes_to_remove |> append (outW_ind)|> append (outM_ind)
Indexes_to_remove <-Indexes_to_remove[!duplicated(Indexes_to_remove)]
Indexes_to_remove
```

```
##  [1]  60  77 212 328 597 981  18 146 339 467 788 843
```

Then, we will eliminate these points (12 points) from the original dataset then do the same partitioning as before.

So, the dataset we work on become distributed as follows: Final Test, Working Test and Training

```
## [1] 199   9
```

```
## [1] 160   9
```

```
## [1] 629   9
```

Total of 988 points, which are the 1000 minus the 12 out-liars.

Applying the same three modeling techniques we used,

And the results are added to the table:

```
## # A tibble: 5 x 2
##   method                           RMSE
##   <chr>                           <dbl>
## 1 Just the average, No Out Liars  12.6
## 2 Regression tree, No Out Liars   11.7
## 3 Random Forest, No Out Liars     11.5
## 4 KNN, No Out Liars               11.6
## 5 Average-No OL                   13.9
```

And the result of the final testing data is :

```
final_rmse_NO_OL
```

```
## [1] 12.11704
```

As can be seen, results after removing outliars are better than results from data containing outliars

References

The dataset can be downloaded from this link: https://www.kaggle.com/datasets/spscientist/students-performance-in-exams