Group 2: Project on dataset number 5

Authors:

Carelsen Quintin

Le Hoang Minh Trihn

Mayoraz Camille

Shamisheva Dina

Teacher:

Luca Mazzola

International IT Management

This project was submitted as part of the requirements for the Business Intelligence Module at the School of Information Technology, Lucerne University of Applied Sciences and Arts

January 2021

Table of contents

1. Loading of the data set and transformation of variables.

When starting to work in R, one normally must import a file of type comma-separated values (.csv) or tab separated value (.tsv) an example of this can be seen in excel spread sheets (Hester, n.d). It is important to open the file before importing it so that you can have a good overview as to what data is being imported and how the files are separating that data as well as whether it is categorical or numerical data. In line 17 of Figure 1, we run the function `read_delim` followed by the pathway to the .csv file we are importing. In line 18, we run the function `delim = ",",` to establish that we are working with data that is separated by commas.

```
17 Performance <- read_delim(file = "C:/Users/User/Desktop/rExercise/StudPerfs.csv",
18                           delim = ",",
19                           col_names = c("Gender","Ethnicity","ParentalEducation","Lunch",
20                                         "Preparation","Math","Reading","Writing"),
21                           skip = 2, #first two lines are not relevant
22                           col_types = cols(
23                             Gender = col_factor(levels = c("female","male")),
24                             Ethnicity = col_factor(levels = c("group A", "group B",
25                                                               "group C", "group D","group E")),
26                             ParentalEducation =
27                               col_factor(levels = c("high school","some high school", "some college",
28                                                     "associate's degree", "bachelor's degree", "master's degree")),
29                             Lunch = col_factor(levels = c("free/reduced", "standard")),
30                             Preparation = col_factor(levels = c("none", "completed")),
31                             Math = col_integer(),
32                             Reading = col_integer(),
33                             Writing = col_integer()
34                           ),
35                           na = c("", "NA", "TEST", "Test", "TESTING","?????"))
36
```

Figure 1. Importing and formatting the dataset

In line 19 and 20 we establish the column names with the function `col_names = c(…)`. In line 21 we `run skip = 2` to tell R to skip the first two rows because the first row is the column titles, and the second row is the internet source where this dataset can be found. In line 22 the function `col_types = cols()` is used to highlight the type of column for example the gender column is categorical and has two possible categories namely male and female, the ethnicity column has five possible categories namely group A, group B, group C, group D, and group E. This process is repeated for parental education, lunch, and preparation. Line 30 – 33 is also establishing the type of data, but this time it is set as numerical because a number is expected. Assigning the right column type helps the computation to work properly. If you assign numerical type to categorical variables, it does not function correctly. Leading to syntax errors or semantic errors.

Lastly, line 34 is to define which row is not applicable (na) this was necessarily needed as we had no wrongful entries in our data set, but we added it anyways.

2. Discussion of the dataset based on str() and summary()

The `str()` function is for displaying the internal structure of an R object (R documentation, 2019). This is useful to have an overview of the variables that will later be used. In our case (Figure 2), it shows factors with certain levels (number of categories) as well as the labeled categories (female, male) followed by the data 1, 1, 1, 2, 2, 1, 1, 2, 2, 1 … This process is repeated until completion of the preparation column. Thereafter, we have the numerical data sets. When looking at the math column we can see that an integer is expected for each of the 1000 data entries, and the data is respectively entered thereafter: 72, 69, 90, 47, 76, 71, 88,40, 64, 38 … This process is repeated until completion of the writing column. We can also see that the attributes are correctly displayed for all the columns from gender to writing, so this means that all the code that we previously entered was successfully interpreted by the system and correctly displayed.

```
> str(Performance)
tibble [1,000 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Gender           : Factor w/ 2 levels "female","male": 1 1 1 2 2 1 1 2 2 1 ...
 $ Ethnicity        : Factor w/ 5 levels "group A","group B",..: 2 3 2 1 3 2 2 2 4 2 ...
 $ ParentalEducation: Factor w/ 6 levels "high school",..: 5 3 6 4 3 4 3 3 1 1 ...
 $ Lunch            : Factor w/ 2 levels "free/reduced",..: 2 2 2 1 2 2 2 1 1 1 ...
 $ Preparation      : Factor w/ 2 levels "none","completed": 1 2 1 1 1 1 2 1 2 1 ...
 $ Math             : int [1:1000] 72 69 90 47 76 71 88 40 64 38 ...
 $ Reading          : int [1:1000] 72 90 95 57 78 83 95 43 64 60 ...
 $ Writing          : int [1:1000] 74 88 93 44 75 78 92 39 67 50 ...
 - attr(*, "spec")=
  .. cols(
  ..   Gender = col_factor(levels = c("female", "male"), ordered = FALSE, include_na = FALSE),
  ..   Ethnicity = col_factor(levels = c("group A", "group B", "group C", "group D", "group E"), ordered = FALSE, include_na = FALSE),
  ..   ParentalEducation = col_factor(levels = c("high school", "some high school", "some college", "associate's degree",
  ..     "bachelor's degree", "master's degree"), ordered = FALSE, include_na = FALSE),
  ..   Lunch = col_factor(levels = c("free/reduced", "standard"), ordered = FALSE, include_na = FALSE),
  ..   Preparation = col_factor(levels = c("none", "completed"), ordered = FALSE, include_na = FALSE),
  ..   Math = col_integer(),
  ..   Reading = col_integer(),
  ..   Writing = col_integer()
  .. )
```

Figure 2. str() of the dataset

The `summary()` function is used to produce result summaries of the various models and their functions (R documentation, n.d.-b).

```
> summary(Performance)
    Gender      Ethnicity          ParentalEducation         Lunch       Preparation       Math          Reading
 female:518   group A: 89   high school       :196   free/reduced:355   none     :642   Min.   :  0.00   Min.   : 17.00
 male  :482   group B:190   some high school  :179   standard    :645   completed:358   1st Qu.: 57.00   1st Qu.: 59.00
              group C:319   some college      :226                                     Median : 66.00   Median : 70.00
              group D:262   associate's degree:222                                     Mean   : 66.09   Mean   : 69.17
              group E:140   bachelor's degree :118                                     3rd Qu.: 77.00   3rd Qu.: 79.00
                            master's degree   : 59                                     Max.   :100.00   Max.   :100.00
    Writing
 Min.   : 10.00
 1st Qu.: 57.75
 Median : 69.00
 Mean   : 68.05
 3rd Qu.: 79.00
 Max.   :100.00
```

Figure 3. Dataset summary

For example, in Figure 3, under the gender column we see that the data set has 518 females and 482 males, and in ethnicity we see how many people belong to which certain group for example 89 in group A, 190 in group B, 319 in group C, 262 in group D, and 140 in group E, this process is repeated for each of the categorical sections. For the numerical columns, the

scores of each column are added together and then displayed in terms of minimum and maximum, 1st and 3rd quartile, as well as the mean and median. For example, the Math score column shows: minimum score = 0, 1st quartile = 57, median = 66 , mean = 66, 3rd quartile = 77 and maximum score = 100. This process is repeated for each of the numerical columns.

3. Prediction & discussion of the standardized test results and plot against variables

To analyze the relation that the categorical data has to the end scores that were achieved by the individuals, we created numerous boxplots for each of the categories and compared them to the numerical data sets. For example, gender versus math, gender versus writing, and gender versus reading. We repeated this step for all the categorical sets and compared them to all the numerical data sets (Figure 4).

```
#Visualizing each categorical variable to see if it is correlated
#Gender
ggboxplot(Performance, x = "Gender", y = "Math",
          ylab = "Math", xlab = "Gender")
ggboxplot(Performance, x = "Gender", y = "Reading",
          ylab = "Reading", xlab = "Gender")
ggboxplot(Performance, x = "Gender", y = "Writing",
          ylab = "Writing", xlab = "Gender")
#Ethnicity
ggboxplot(Performance, x = "Ethnicity", y = "Math",
          ylab = "Math", xlab = "Ethnicity")
ggboxplot(Performance, x = "Ethnicity", y = "Reading",
          ylab = "Reading", xlab = "Ethnicity")
ggboxplot(Performance, x = "Ethnicity", y = "Writing",
          ylab = "Writing", xlab = "Ethnicity")
#Parental Education
ggboxplot(Performance, x = "ParentalEducation", y = "Math",
          ylab = "Math", xlab = "Parental Education")
ggboxplot(Performance, x = "ParentalEducation", y = "Reading",
          ylab = "Reading", xlab = "Parental Education")
ggboxplot(Performance, x = "ParentalEducation", y = "Writing",
          ylab = "Writing", xlab = "Parental Education")
#Lunch
ggboxplot(Performance, x = "Lunch", y = "Math",
          ylab = "Math", xlab = "Lunch")
ggboxplot(Performance, x = "Lunch", y = "Reading",
          ylab = "Reading", xlab = "Lunch")
ggboxplot(Performance, x = "Lunch", y = "Writing",
          ylab = "Writing", xlab = "Lunch")
#Preparation
ggboxplot(Performance, x = "Preparation", y = "Math",
          ylab = "Math", xlab = "Preparation")
ggboxplot(Performance, x = "Preparation", y = "Reading",
          ylab = "Reading", xlab = "Preparation")
ggboxplot(Performance, x = "Preparation", y = "Writing",
          ylab = "Writing", xlab = "Preparation")
```

Figure 4. Use variables against test results variables

The boxplot graphs can be found in Appendix A. For simplicity we have created tables to summarize the information found on the graphs.

	Male	Female
Math	1st	2nd
Writing	2nd	1st
Reading	2nd	1st

Table 1: Gender versus score

Based on the boxplot, we can say that males are on average better at math, whereas on average females are better at reading and writing.

	Group A	Group B	Group C	Group D	Group E
Math	5th	4th	3rd	2nd	1st
Writing	5th	2nd	2nd	2nd	1st
Reading	5th	4th	3rd	1st	1st

Table 2: Ethnicity versus score

Based on the boxplot, we can say that group E, outperforms all other ethnicities in math and writing, however, they seem to tie with group D in relation to reading because their Median, max, and min scores are all similar. Regarding writing, group D, group C, and group B are tied because their Median, max, and min scores are all similar. Other than that, on average group A comes in 1st, group D comes in 2nd, group C comes in 3rd, group B comes in 4th, and group A comes in 5th.

	High school	Some high school	Some college	Associates degree	Bachelor's degree	Master's degree
Math	6th	5th	2nd	2nd	2nd	1st
Writing	6th	5th	4th	3rd	2nd	1st
Reading	6th	5th	3rd	3rd	2nd	1st

Table 3: Parental education versus score

Based on the boxplot, we can say that children who have parents with a master's degree score higher on average in math, writing and reading, regardless of the average low minimum score for math.

Overall, children with parents who have a bachelor's degree come in 2nd place in terms of reading and writing however, they tie for 2nd place regarding math with children who have

parents with associates degrees as well as children who have parents with a college degree this is because they all have a very similar median.

However, children of parents with an associates degree also tie in reading with children who have parents with some college degree, we call this a tie because the median is very similar and although children of associates education have a higher max points scored, children of parents with some college degree have a higher average minimum score. Having said that, on average we can still say that children with parents who have an associates degree take the overall 3rd place despite the math and reading ties.

Children with parents who have some college degree take the overall 4th place despite their reading and math ties. Overall, we can say that children of parents with some high school come in at the 5th position in terms of score. Lastly, comes the children with parents with high school, taking the 6th position with the lowest average scores in reading, math, and writing.

	Free/reduced	Standard
Math	2nd	1st
Writing	2nd	1st
Reading	2nd	1st

Table 4: Lunch versus score

Based on the boxplot, we can say that children who have standard lunch score highest in math reading and writing based on them having higher medians, min, and max values across the variables. Therefore, children with free/reduced lunch come in 2nd place.

	None	Completed
Math	2nd	1st
Writing	2nd	1st
Reading	2nd	1st

Table 5: Preparation versus scores

Based on the boxplot, we can clearly see that having completed the preparation directly influences the scores of reading writing and math. Therefore, having completed preparation comes in 1st place and not having completed preparation comes in 2nd place.

Finally, to separate data into training and testing set, steps were done (Figure 5). Line 45 shows the `set.seed()` function, this generates a random number but with repeatability, this ensures that you get the same result even if you repeat the process (ETHZ, n.d.-c). In Line 46 – 48 we are setting 95% of the data to be stored in a data set called `performance.train`. In line 49 and 50 we index the remaining data into a data set called `Performance.test`.

```r
#seperating the dataset. 5% test, 95% training
set.seed(578)
index_train <- sample(1:nrow(Performance),0.95*nrow(Performance))
Performance.train <- Performance[index_train,]
dim(Performance.train)
Performance.test <- Performance[-index_train,]
dim(Performance.test)
```

Figure 5. Dataset separation

4. Simple regression models with variables pairs & multiple regression models

This part aims to build the best regression models for Math, Reading, and Writing. In other words, Math, Reading, and Writing are dependent variables. Others are independent variables. Since the model's goal is to observe how numerical variables change based on categorical independent variables, it is essential to use `aov()` function instead of `lm()` (Bevans, 2020). The function `aov()` is a wrapper to `lm` object. The only differences to `lm` objects are in the way how `aov()` objects are `print()` and `summary()` (R documentation, n.d.-a).

To build the best regression model, we build simple regression models using one of the categorical vs. test results. It compares Gender to Math, Ethnicity to Math, Parental Education vs. Math, and so on (Figure 6, line 97, 99, 101, 103, 105).

```r
### MATH MODEL
###

#calculate simple linear regression for Gender vs Math
Math.Gender = aov(Math~Gender, data=Performance.train)
#calculate simple linear regression for Ethnicity vs Math
Math.Ethnicity = aov(Math~Ethnicity, data=Performance.train)
#calculate simple linear regression for Parental Education vs Math
Math.ParentalEducation = aov(Math~ParentalEducation, data=Performance.train)
#calculate simple linear regression for Lunch vs Math
Math.Lunch = aov(Math~Lunch, data=Performance.train)
#calculate simple linear regression for Preparation vs Math
Math.Preparation = aov(Math~Preparation, data=Performance.train)
#Calculate multiple regression model
Math.Multiple.1 = aov(Math~Gender*Ethnicity*ParentalEducation*Lunch*Preparation, data=Performance.train)
summary(Math.Multiple.1)
#Calculate multitple linear regression model with all significance variables
Math.Multiple.2 = aov(
  Math~
    Gender+Ethnicity+ParentalEducation+Lunch+Preparation+Gender:Lunch:Preparation+ParentalEducation:Lunch:Preparation,
  data=Performance.train)
summary(Math.Multiple.2)
#Calculate multiple linear regression model with all significance variables
Math.Multiple.3 = aov(Math~Gender+Ethnicity+ParentalEducation+Lunch+Preparation, data=Performance.train)
summary(Math.Multiple.3)
```

Figure 6. Different regression models for Math

However, upon inspection, the R squared results of those models are not high. Indeed, simple regression between a categorical variable and numerical can never have a high R squared value. This is because of the limitation of the number of categories from a categorical variable. For example, Gender is a categorical variable. It has male and female.

Using the `coef()` function, one can observe the slope and intercept of a regression model. In this case, we use `coef(Math.Gender)` to inspect the regression equation of the model (Figure 7). The equation of the regression model is y=5.36x+63.54 where y is the result

of math score. x = 1 if it is male, 0 if female. The predictions from this model are not reliable since the predicted test result can either be 63.54, or 68.9. That creates many errors between real values and predicted values, hence explaining the low R-squared value.

```
> coef(Math.Gender)
(Intercept)  Gendermale
  63.542857    5.357143
```

Figure 7. The slope and intercept of Gender vs. Math regression equation

Therefore, this paper decided to build multiple regression models on top of simple regression models get better predictions. It does so by examining all categorical variables and the relationship between them. The regression model tests which variables, and their relationships affect the dependent variables. With more combinations of categories, there are more outcomes compared to just two outcomes from the previous example.

The script then builds multiple regression models for the Math score. The first one is `Math.Multiple.1` (Figure 6, line 107). Again, its goal is to examine every combination of categorical variables possible to discover the best regression model.

After looking at the summary (Figure 8) it was shown that `Gender`, `Ethnicity`, `ParentalEducation`, `Lunch` and `Preparation` have low p-values. it is less than 0.001. Hence, they are important to be included in further models. In addition, the relationship between `ParentalEducation:Lunch:Preparation` seems to affect the Math score with p-value = 0.0437. `Gender:Lunch:Preparation` affect the score with lower level of significance with p value at 0.0517.

```
> summary(Math.Multiple.1)
                                                  Df Sum Sq Mean Sq F value   Pr(>F)
Gender                                             1   6809    6809  37.996 1.17e-09 ***
Ethnicity                                          4  12080    3020  16.852 3.13e-13 ***
ParentalEducation                                  5   6122    1224   6.833 3.04e-06 ***
Lunch                                              1  23483   23483 131.038  < 2e-16 ***
Preparation                                        1   6703    6703  37.405 1.56e-09 ***
Gender:Ethnicity                                   4    260      65   0.363   0.8351
Gender:ParentalEducation                           5   1124     225   1.255   0.2818
Ethnicity:ParentalEducation                       20   1460      73   0.407   0.9905
Gender:Lunch                                       1    254     254   1.420   0.2338
Ethnicity:Lunch                                    4    383      96   0.535   0.7101
ParentalEducation:Lunch                            5    704     141   0.785   0.5603
Gender:Preparation                                 1     17      17   0.096   0.7562
Ethnicity:Preparation                              4    363      91   0.506   0.7314
ParentalEducation:Preparation                      5    178      36   0.199   0.9629
Lunch:Preparation                                  1      0       0   0.000   0.9950
Gender:Ethnicity:ParentalEducation                20   3555     178   0.992   0.4698
Gender:Ethnicity:Lunch                             4    724     181   1.010   0.4016
Gender:ParentalEducation:Lunch                     5    646     129   0.721   0.6080
Ethnicity:ParentalEducation:Lunch                 20   5120     256   1.428   0.1009
Gender:Ethnicity:Preparation                       4    162      41   0.226   0.9237
Gender:ParentalEducation:Preparation               5   1080     216   1.205   0.3049
Ethnicity:ParentalEducation:Preparation           18   1861     103   0.577   0.9173
Gender:Lunch:Preparation                           1    681     681   3.799   0.0517 .
Ethnicity:Lunch:Preparation                        4    697     174   0.972   0.4221
ParentalEducation:Lunch:Preparation                5   2058     412   2.296   0.0437 *
Gender:Ethnicity:ParentalEducation:Lunch          16   2187     137   0.763   0.7290
Gender:Ethnicity:ParentalEducation:Preparation    15   2400     160   0.893   0.5725
Gender:Ethnicity:Lunch:Preparation                 3    496     165   0.923   0.4290
Gender:ParentalEducation:Lunch:Preparation         5    355      71   0.397   0.8512
Ethnicity:ParentalEducation:Lunch:Preparation     15   1432      95   0.533   0.9232
Gender:Ethnicity:ParentalEducation:Lunch:Preparation  8   2000     250   1.395   0.1950
Residuals                                        739 132434     179
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 8. Summary of Math.Multiple.1

The discovered significant variables and relationships created a new model called `Math.Mulitple.2` (Figure 6, line 110). With inspecting the `Math.Multiple.2` model, it was shown that only the categorical variables are important for predicting the Math variable (Figure 9).

```
> summary(Math.Multiple.2)
                                Df Sum Sq Mean Sq F value   Pr(>F)
Gender                           1   6809    6809  39.176 5.94e-10 ***
Ethnicity                        4  12080    3020  17.375 9.44e-14 ***
ParentalEducation                5   6122    1224   7.045 1.79e-06 ***
Lunch                            1  23483   23483 135.108  < 2e-16 ***
Preparation                      1   6703    6703  38.566 8.01e-10 ***
Gender:Lunch:Preparation         4    490     122   0.704    0.589
ParentalEducation:Lunch:Preparation 15   2582     172   0.991    0.463
Residuals                      918 159557     174
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 9. Summary of Math.Multiple.2

`Math.Multiple.3` model was created consisting of `Gender`, `Ethnicity`, `ParentalEducation`, `Lunch`, and `Preparation` (Figure 6, line 116).

The same procedure was done for `Reading` and `Writing`. To boil it down, the R script builds simple regression models. Subsequently, it builds multiple regression models that reduces the number of independent variables based on their p-values. The code of models' creation for `Reading` and `Writing` can be seen in Figure 10 and Figure 11.

```r
### READING MODEL
###

#calculate simple linear regression for Gender vs Reading
Reading.Gender = aov(Reading~Gender, data=Performance.train)
#calculate simple linear regression for Ethnicity vs Reading
Reading.Ethnicity = aov(Reading~Ethnicity, data=Performance.train)
#calculate simple linear regression for Parental Education vs Reading
Reading.ParentalEducation = aov(Reading~ParentalEducation, data=Performance.train)
#calculate simple linear regression for Lunch vs Reading
Reading.Lunch = aov(Reading~Lunch, data=Performance.train)
#calculate simple linear regression for Preparation vs Reading
Reading.Preparation = aov(Reading~Preparation, data=Performance.train)
#Calculate multiple regression model
Reading.Multiple.1 = aov(Reading~Gender*Ethnicity*ParentalEducation*Lunch*Preparation, data=Performance.train)
summary(Reading.Multiple.1)
#Calculate multitple linear regression model with all significance variables
Reading.Multiple.2 = aov(
  Reading~
    Gender+Ethnicity+ParentalEducation+Lunch+Preparation+Gender:Lunch:Preparation+ParentalEducation:Lunch:Preparation,
  data=Performance.train)
summary(Reading.Multiple.2)
#Calculate multiple linear regression model with all significance variables
Reading.Multiple.3 = aov(Reading~Gender+Ethnicity+ParentalEducation+Lunch+Preparation, data=Performance.train)
summary(Reading.Multiple.3)
```

Figure 10. Different regression models for Reading

```r
### WRITING MODEL
###

#calculate simple linear regression for Gender vs Writing
Writing.Gender = aov(Writing~Gender, data=Performance.train)
#calculate simple linear regression for Ethnicity vs Writing
Writing.Ethnicity = aov(Writing~Ethnicity, data=Performance.train)
#calculate simple linear regression for Parental Education vs Writing
Writing.ParentalEducation = aov(Writing~ParentalEducation, data=Performance.train)
#calculate simple linear regression for Lunch vs Writing
Writing.Lunch = aov(Writing~Lunch, data=Performance.train)
#calculate simple linear regression for Preparation vs Writing
Writing.Preparation = aov(Writing~Preparation, data=Performance.train)
#Calculate multiple regression model
Writing.Multiple.1 = aov(Writing~Gender*Ethnicity*ParentalEducation*Lunch*Preparation, data=Performance.train)
summary(Writing.Multiple.1)
#Calculate multitple linear regression model with all significance variables
Writing.Multiple.2 = aov(
  Writing~
    Gender+Ethnicity+ParentalEducation+Lunch+Preparation+Gender:Lunch:Preparation+ParentalEducation:Lunch:Preparation,
  data=Performance.train)
summary(Writing.Multiple.2)
#Calculate multiple linear regression model with all significance variables
Writing.Multiple.3 = aov(Writing~Gender+Ethnicity+ParentalEducation+Lunch+Preparation, data=Performance.train)
summary(Writing.Multiple.3)
```

Figure 11. Different regression models for Writing

At the end, apart from simple linear regression models for each dependent variable `Reading` and `Writing`, we have multiple regression models for them as well. They both have `Gender`, `Ethnicity`, `ParentalEducation`, `Lunch`, and `Preparation` as the predictors in the multiple regression models.

5. Best predictor for "math score", for "reading score" & "writing score"

based on the R-value

To calculate the best predictor, we analyze the lowest score on Akaike's An Information Criterion (AIC) and the lowest Bayesian information criteria (BIC), as well as the highest score in R2_adjusted. The idea of AIC is to punish the inclusion of additional variables and adds a penalty when adding including additional terms. BIC is the same as AIC only with a stronger penalty for including additional variables. The R-squared value is for the proportion of variation, this is to show the correlation between the actual values of the model and the predicted values in other words how efficiently the model was able to predict the values, and the R-squared adjusted simply adjusts for a model with too many variables.

```
> #Comparing models
> comparison <- compare_performance(
+   Math.Gender,Math.Ethnicity,Math.ParentalEducation, Math.Lunch, Math.Preparation,
+   Math.Multiple.1,Math.Multiple.2,Math.Multiple.3,
+   rank=TRUE)
> comparison
# Comparison of Model Performance Indices

Model                | Type |     AIC |     BIC |   R2 | R2_adjusted |  RMSE | Sigma |         BF | Performance_Score
-----------------------------------------------------------------------------------------------------------------------
Math.Multiple.3      |  aov | 7609.61 | 7677.60 | 0.25 |        0.24 | 13.08 | 13.17 |  BF > 1000 |            86.72%
Math.Multiple.2      |  aov | 7629.49 | 7789.75 | 0.27 |        0.24 | 12.96 | 13.18 |  BF > 1000 |            84.83%
Math.Multiple.1      |  aov | 7810.49 | 8840.06 | 0.39 |        0.22 | 11.81 | 13.39 | BF < 0.001 |            64.96%
Math.Lunch           |  aov | 7748.85 | 7763.42 | 0.12 |        0.11 | 14.24 | 14.26 |  BF > 1000 |            42.60%
Math.Ethnicity       |  aov | 7815.30 | 7844.44 | 0.06 |        0.05 | 14.70 | 14.74 | BF = 13.33 |            22.18%
Math.Preparation     |  aov | 7832.05 | 7846.62 | 0.03 |        0.03 | 14.88 | 14.90 |  BF = 4.49 |            15.94%
Math.Gender          |  aov | 7835.05 | 7849.62 | 0.03 |        0.03 | 14.90 | 14.92 |  BF = 1.00 |            14.95%
Math.ParentalEducation |  aov | 7841.10 | 7875.09 | 0.03 |        0.03 | 14.89 | 14.94 | BF < 0.001 |            14.01%

Model Math.Multiple.3 (of class aov) performed best with an overall performance score of 86.72%.
```

Figure 12. Comparison between models for Math

From AIC, BIC and R2_adjustd, the best model for Math is the regression model `Math.Multiple.3`.

```
> #Comparing models
> comparison <- compare_performance(
+   Reading.Gender,Reading.Ethnicity,Reading.ParentalEducation, Reading.Lunch, Reading.Preparation,
+   Reading.Multiple.1,Reading.Multiple.2,Reading.Multiple.3,
+   rank=TRUE)
> comparison
# Comparison of Model Performance Indices

Model                | Type |     AIC |     BIC |   R2 | R2_adjusted |  RMSE | Sigma |         BF | Performance_Score
-----------------------------------------------------------------------------------------------------------------------
Reading.Multiple.3   |  aov | 7584.29 | 7652.28 | 0.22 |        0.21 | 12.91 | 13.00 |  BF > 1000 |            85.45%
Reading.Multiple.2   |  aov | 7590.10 | 7677.52 | 0.22 |        0.21 | 12.90 | 13.01 |  BF > 1000 |            84.50%
Reading.Multiple.1   |  aov | 7783.96 | 8813.53 | 0.37 |        0.19 | 11.64 | 13.20 | BF < 0.001 |            62.30%
Reading.Preparation  |  aov | 7739.58 | 7754.15 | 0.06 |        0.06 | 14.17 | 14.19 | BF = 11.56 |            29.52%
Reading.Gender       |  aov | 7744.47 | 7759.04 | 0.06 |        0.06 | 14.21 | 14.22 |  BF = 1.00 |            27.77%
Reading.Lunch        |  aov | 7752.41 | 7766.98 | 0.05 |        0.05 | 14.27 | 14.28 | BF = 0.019 |            24.91%
Reading.ParentalEducation |  aov | 7763.71 | 7797.70 | 0.04 |        0.04 | 14.29 | 14.34 | BF < 0.001 |            21.96%
Reading.Ethnicity    |  aov | 7783.69 | 7812.83 | 0.02 |        0.02 | 14.46 | 14.50 | BF < 0.001 |            14.39%

Model Reading.Multiple.3 (of class aov) performed best with an overall performance score of 85.45%.
```

Figure 13. Comparison between models for Reading

From AIC, BIC and R2_adjustd, the best model for Reading is the regression model `Reading.Multiple.3`.

```
> #Comparing models
> comparison <- compare_performance(Writing.Gender,Writing.Ethnicity,Writing.ParentalEducation, Writing.Lunch, Writing.Preparation,
+                                   Writing.Multiple.1,Writing.Multiple.2,Writing.Multiple.3,
+                                   rank=TRUE)
> comparison
# Comparison of Model Performance Indices

Model                    | Type |     AIC |     BIC |   R2 | R2_adjusted |  RMSE | Sigma |         BF | Performance_Score
-------------------------------------------------------------------------------------------------------------------------------
Writing.Multiple.3       |  aov | 7515.19 | 7583.18 | 0.33 |        0.32 | 12.45 | 12.54 |  BF > 1000 |            90.20%
Writing.Multiple.2       |  aov | 7519.90 | 7607.32 | 0.33 |        0.32 | 12.43 | 12.54 |  BF > 1000 |            89.75%
Writing.Multiple.1       |  aov | 7731.56 | 8761.13 | 0.44 |        0.28 | 11.33 | 12.84 | BF < 0.001 |            68.55%
Writing.Preparation      |  aov | 7770.28 | 7784.85 | 0.10 |        0.10 | 14.40 | 14.42 |  BF > 1000 |            31.35%
Writing.Gender           |  aov | 7784.30 | 7798.87 | 0.09 |        0.08 | 14.51 | 14.53 |  BF = 1.00 |            27.95%
Writing.ParentalEducation |  aov | 7810.79 | 7844.78 | 0.07 |        0.06 | 14.65 | 14.70 | BF < 0.001 |            22.19%
Writing.Lunch            |  aov | 7815.04 | 7829.61 | 0.06 |        0.06 | 14.75 | 14.76 | BF < 0.001 |            20.38%
Writing.Ethnicity        |  aov | 7848.86 | 7877.99 | 0.03 |        0.02 | 14.96 | 15.01 | BF < 0.001 |            12.50%

Model Writing.Multiple.3 (of class aov) performed best with an overall performance score of 90.20%.
```

Figure 14. Comparison between models for Writing

From AIC, BIC and R2_adjustd, the best model for Writing is the regression model `Writing.Multiple.3`.

One can see that in each model, the best option was `Mulitple.3`. As explained in the previous section, simple regression models predict a low number of outcomes due to the limitation in the category. Thanks to multiple regression, one can better precise outcomes by increasing the combinations of categories. Take `Math.Mulitple.3` as an example. There are two categories in `Gender`, five in `Ethnicity`, six in `ParentalEducation`, two in `Lunch` and two in `Preparation`. Combining them together, there are 2*5*6*2*2 = 240 outcomes possible. Each with a different value.

6. Comparison and discussion of the straight-line equations of the models

According to Statistics Solution (2021), an equation of multiple regression model is written as follows:

$$y = b_1 x_1 + b_2 x_2 + \cdots + b_n x_n + c \quad (1)$$

Where:

y = Value of dependent variable

i = $1,2,\ldots,n$

b_i = Coefficients of the regression

x_i = Independent variables' values

Figure 15 presents the regression coefficients and intercept of the `Math.Fit` model.

```
> #result shows that Math.Multiple.3 has the lowest AIC, best performance score of 86.72%
> #Chose Math.Multiple.3
> Math.Fit= Math.Multiple.3
> coef(Math.Fit) #details of intercept b and slopes.
                      (Intercept)                        Gendermale                  Ethnicitygroup B
                        47.4186232                         5.1988305                         1.8432668
                Ethnicitygroup C                  Ethnicitygroup D                  Ethnicitygroup E
                        2.4874886                         5.3942858                        10.1295484
 ParentalEducationsome high school       ParentalEducationsome college ParentalEducationassociate's degree
                        0.5844529                         4.4530221                         4.7886778
ParentalEducationbachelor's degree    ParentalEducationmaster's degree                  Lunchstandard
                        6.5874167                         7.6576418                        10.5127163
            Preparationcompleted
                        5.5769308
```

Figure 15. Intercept and slopes of the best model for Math.Fit

Based on the values and equation (1), a table is created as a template to derive a math score from any given data (Table 6. Equation of Math modelTable 6). This template is useful because of its convenient and easy-to-read characteristic.

Table 6. Equation of Math model

Math =						
Gender	Male				Female	
	5.2				0	
+ Ethnicity	A	B	C	D	E	
	0	1.84	2.49	5.4	10.13	
+ Parental Education	High school	Some high schools	Some col-lege	Associate's degree	Bachelor's degree	Master's degree
	0	0.58	4.45	4.79	6.59	7.66
+ Lunch	Free/Reduced			Standard		
	0			10.51		
+ Prepara-tion	None			Completed		
	0			5.58		
+ Intercept	47.42					

Math equal to the sum of the intercept and the sum of each row. The sum of each row is calculated as each column times the value under it. For instance, if a student is a female, her Gender score would be:

$$GenderMale * 5.2 + GenderFemale * 0 = 0 * 5.2 + 1 * 0 = 0$$

In this case, she is a female, hence, $GenderMale = 0,\ GenderFemale = 1$. Finally, we can write the equation as follow:

$$Math = GenderMale * 5.2 + GenderFemale * 0$$
$$+EthnicityA * 0 + EthnicityB * 1.84 + EthinicityC * 2.49 + EthnicityD * 5.4$$
$$+ \cdots$$
$$+PreparationCompleted * 5.58 + 47.42$$

Based on Figure 16, and Figure 17, the same kind of table was done for Writing and Reading (Table 7, Table 8).

```
> #result shows that Reading.Multiple.3 has the lowest AIC, best performance score of 85.45%
> #Chose Reading.Multiple.3
> Reading.Fit= Reading.Multiple.3
> coef(Reading.Fit) #details of intercept b and slopes.
                      (Intercept)                         Gendermale                  Ethnicitygroup B
                        58.9969253                         -6.9239574                          1.1631935
                Ethnicitygroup C                  Ethnicitygroup D                  Ethnicitygroup E
                        2.3752236                          4.1211381                          5.5191945
  ParentalEducationsome high school       ParentalEducationsome college ParentalEducationassociate's degree
                        0.9568573                          3.7586727                          4.9753831
 ParentalEducationbachelor's degree    ParentalEducationmaster's degree                    Lunchstandard
                        6.8817899                          9.2625046                          6.8210018
              Preparationcompleted
                        7.4681510
```

Figure 16. Intercept and slopes of the best model for Reading

```
> #result shows that Writing.Multiple.3 has the lowest AIC, best performance score of 90.20%
> #Chose Writing.Multiple.3
> Writing.Fit= Writing.Multiple.3
> coef(Writing.Fit) #details of intercept b and slopes.
                      (Intercept)                         Gendermale                  Ethnicitygroup B
                        56.0745024                         -8.9618089                          1.1193331
                Ethnicitygroup C                  Ethnicitygroup D                  Ethnicitygroup E
                        2.5232008                          5.9250359                          5.1279071
  ParentalEducationsome high school       ParentalEducationsome college ParentalEducationassociate's degree
                        0.5513601                          5.1636617                          5.9114313
 ParentalEducationbachelor's degree    ParentalEducationmaster's degree                    Lunchstandard
                        9.1866152                         11.1210589                          7.7667479
              Preparationcompleted
                       10.0839380
```

Figure 17. Intercept and slopes of the best model for Writing

Table 7.Equation of Reading model

Reading =						
Gender	Male				Female	
	-6.92				0	
+ Ethnicity	A	B	C	D	E	
	0	1.16	2.38	4.12	5.52	
+ Parental Education	High school	Some high schools	Some college	Associate's degree	Bachelor's degree	Master's degree
	0	0.96	3.76	4.98	6.88	9.26
+ Lunch	Free/Reduced			Standard		
	0			6.82		
+ Preparation	None			Completed		
	0			7.47		
+ Intercept	59					

Table 8. Equation of Writing model

Writing =						
Gender	Male				Female	
	-8.96				0	
+ Ethnicity	A	B	C	D	E	
	0	1.12	2.52	5.93	5.13	
+ Parental Education	High school	Some high schools	Some college	Associate's degree	Bachelor's degree	Master's degree
	0	0.55	5.16	5.91	9.19	11.12
+ Lunch	Free/Reduced			Standard		
	0			7.77		
+ Preparation	None			Completed		
	0			10.08		
+ Intercept	56.07					

Analyzing the tables, especially Table 6, the smallest possible predicted Math score is a student who is female, belong to ethnicity group A. She has parents that finishes high school. She has free or reduced lunch and does not have prior preparation course. Her predicted Math score would be nothing but the intercept, which is 47.42.

The highest Math score would be one who is male, ethnicity group E, has parents with master's degree, standard lunch, and completed his preparation course. His predicted Math score would be:

$$5.2 + 10.13 + 7.66 + 10.51 + 5.58 + 47.42 = 86.5$$

Therefore, the predicted Math score would always be in between 47.42 to 86.5. This will affect the residual plots later.

Since categorical variables affect Math; Reading and Writing are no exception. They both are always in a range that is not from 0 to 100. Rather than that of Reading, it is between 52.08 and 88.07, and from 47.11 to 81.21 for Writing.

7. Residual plots for the models

To calculate the residual of each value, it is equal to the predicted value minus the actual value. We get the predicted value by calling function `predict()` (Figure 18, line 131).

The function returns a list of predicted values based on linear model object (ETHZ, n.d.-b). Since `aov` object is a wrapper of `lm` objects, the function extends to them as well. That is why we could predict values using our regression models. For Math score, we use the `Math.Fit` as our model, and predict the `Math` value based on `Performance.test` dataset.

The values of predicted Math scores are stored into `predicted.Math` object. A new data frame is then created for plotting and calculating residual values (line 135). `residual.Math`'s first column is the predicted values. The next column is the real values, following by a column n as an index. Finally, the last column is the residuals values which is the differences between predicted values and real values. The values are then rearranged from the lowest real score to the highest score.

```
130 #predict the values of Math based the regression model in test datasets
131 predicted.Math <- predict(Math.Fit, Performance.test)
132 predicted.Math
133 Performance.test
134 #Calculate the residuals
135 residuals.Math <- as.tibble(predicted.Math) %>%
136   mutate(real = Performance.test$Math, n=row_number()) %>%
137   mutate(error= value-real)
138 residuals.Math <- residuals.Math %>% arrange(real)  %>% mutate(n = row_number())
139 #Plotting the residuals
140 ggplot(data=residuals.Math) +
141   #errors
142   geom_point(aes(x=n,y=error),color="blue") +
143   geom_abline(slope = 0)
```

Figure 18. Predict Math scores and plot residuals

Line 140 of Figure 18 is used for graphing out the residual. Because errors are the differences between real values and predicted values, it can be negative or positive. The errors are plotted (Figure 19). For smaller real values, the model tends to predict them higher (positive errors), whereas, for larger values, models predicted them to be smaller (negative errors). Another observation is the errors seem to be smallest when real value is around the middle of the range. That error gets bigger as the real value increases to the upper limit (100) or decreases to the lower limit (0)

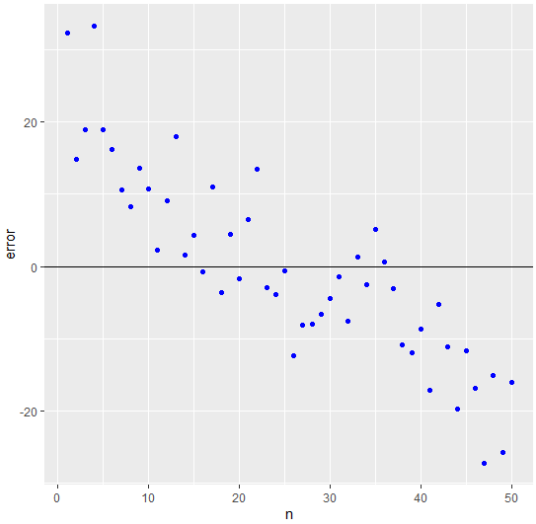

Figure 19. Residuals plot of Math

An explanation for this is explained from previous section. Math predicted values will always fall between 47.42 and 86.5, whereas the real score can be anywhere from 0 to 100.

The same lines of codes were done for Reading and Writing (Figure 25. Code of the residual plots for Reading, Figure 20, Figure 21). The results are plotted in Figure 22, and Figure 23. We can see that Reading and Writing suffer from the same condition as Math. Both models predict higher values for lower real scores and lower values for higher scores. The error tends to be low when the real value is in the middle of value range. It increases as the value progresses to either side of the limit range.

Again, this is the result of the constraints of prediction. Predicted reading and writing scores always fall on to a range from around 50 to around 80, while in real scenario, it could be anywhere between 0 and 100.

```
#predict the values of Reading based the regression model in test datasets
predicted.Reading <- predict(Reading.Fit, Performance.test)
predicted.Reading
Performance.test
#Calculate the residuals
residuals.Reading <- as.tibble(predicted.Reading) %>%
  mutate(real = Performance.test$Reading, n=row_number()) %>%
  mutate(error= value-real, ratio=error/real)
residuals.Reading <- residuals.Reading %>% arrange(real)  %>% mutate(n = row_number())
#Plotting the residuals
ggplot(data=residuals.Reading) +
  #errors
  geom_point(aes(x=n,y=error),color="blue") +
  geom_abline(slope = 0)
```

Figure 20. Predict Reading scores and plot residuals

```
#predict the values of Writing based the regression model in test datasets
predicted.Writing <- predict(Writing.Fit, Performance.test)
predicted.Writing
Performance.test
#Calculate the residuals
residuals.Writing <- as.tibble(predicted.Writing) %>%
  mutate(real = Performance.test$Writing, n=row_number()) %>%
  mutate(error= value-real, ratio=error/real)
residuals.Writing <- residuals.Writing %>% arrange(real)  %>% mutate(n = row_number())
#Plotting the residuals
ggplot(data=residuals.Writing) +
  #errors
  geom_point(aes(x=n,y=error),color="blue") +
  geom_abline(slope = 0)
```

Figure 21. Predict Writing scores and plot residuals

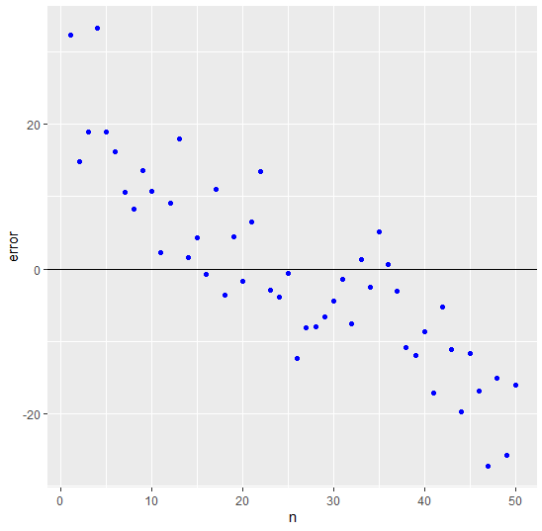

Figure 22. Residuals plot of Reading

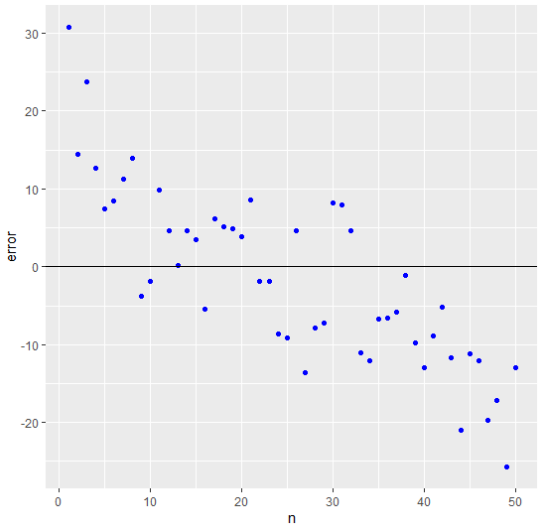

Figure 23. Residuals plot of Writing

8. Discussion and plots of the test values predicted for each instance

To plot real values and predicted values, we used residuals.Math data frame from the previous section. This plot is a scatter one that has y-axis representing real and predicted value. X-axis represents the index of the point on the data frame. Blue points are real value from test dataset. Green points are on the other hand predicted values. Between each instance of real

value, there is a segment connecting the real value and predicted value. If the segment is red, it means the predicted value is smaller than the actual value. In contrast, if the segment is turquois, the predicted value is higher than the actual value. In addition, there is another black line that would run through the middle point of each section. The changes on this line would represent if the model produces in general good predictions or not (Figure 24, Figure 25, Figure 26).

```
#Plotting the real value in order and its corresponding predicted value
ggplot(data=residuals.Math) +
  #real data from test set
  geom_point(aes(x=n,y=real),color="blue") +
  #predicted data
  geom_point(aes(x=n,y=value),color="green", alpha=0.25) +
  geom_segment(
    aes(x=n,xend=n,y=real,yend=value, color=factor(sign(value-real),levels=c(-1,1))),
    size=1, alpha=0.5)+
  geom_line(aes(x=n,y=(value+real)/2),color="black") +
  theme(legend.position = "none")
```

Figure 24. Code of the residual plot for Math

```
  #real data from test set
  geom_point(aes(x=n,y=real),color="blue") +
  #predicted data
  geom_point(aes(x=n,y=value),color="green", alpha=0.25) +
  geom_segment(
    aes(x=n,xend=n,y=real,yend=value, color=factor(sign(value-real),levels=c(-1,1))),
    size=1, alpha=0.5)+
  geom_line(aes(x=n,y=(value+real)/2),color="black") +
  theme(legend.position = "none")
```

Figure 25. Code of the residual plots for Reading

```
#Plotting the real value in order and its corresponding predicted value
ggplot(data=residuals.Writing) +
  #real data from test set
  geom_point(aes(x=n,y=real),color="blue") +
  #predicted data
  geom_point(aes(x=n,y=value),color="green", alpha=0.25) +
  geom_segment(
    aes(x=n,xend=n,y=real,yend=value, color=factor(sign(value-real),levels=c(-1,1))),
    size=1, alpha=0.5)+
  geom_line(aes(x=n,y=(value+real)/2),color="black") +
  theme(legend.position = "none")
```

Figure 26. Code of the residual plots for Writing

The result can be seen in Figures: Figure 27, Figure 28, and Figure 29. As suspected in the previous section, the predicted values tend to be higher as the real values decrease. Vice versa, the predicted values seem to be smaller as the real values increase.

We can observe that generally, when real values increase, the black lines also go up. Therefore, it proves that the regression models definitely have a certain relationship with the real values.

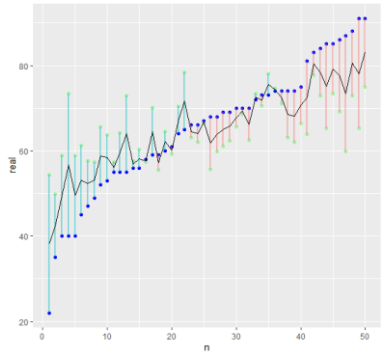

Figure 27. Ordered residual plot for Math

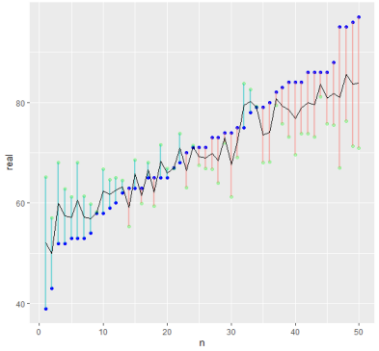

Figure 28. Ordered residual plot for Reading

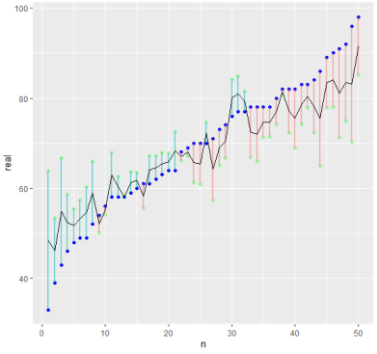

Figure 29. Ordered residual plot for Writing

9. Business insights gained from the data set

Looking at the data description, there is no implication on what standard lunch means (Royce Kimmons, 2012). The assumption is standard lunch is served from the school's cafeteria or brought from home.

From the equation tables (1,2,3), one could advertise that, students have better scores when their lunch is standard. The model predicts ten marks higher for Math, seven for Reading, and eight for Writing. Therefore, school's administration can encourage students to buy lunch or at least bring one from home. Based on that, they could also improve their menu to make the lunch-buying option more appealing, hence, resulting in more income.

Another business relevant insight is how students with preparation courses performed better. Ethically, the purpose of training and teaching is not to make money, therefore, the administration could make preparation courses be accessible for every student. Students could voluntarily registered if they want to take part in the courses.

10. Bibliography

Bevans, R. (2020, March 6). ANOVA in R | A Complete Step-by-Step Guide with Examples. Retrieved January 2, 2021, from https://www.scribbr.com/statistics/anova-in-r/

Data Novia. (2018). ANOVA in R: The Ultimate Guide. Retrieved from https://www.data-novia.com/en/lessons/anova-in-r/

ETHZ. (n.d.-a). R: Fit an Analysis of Variance Model. Retrieved January 2, 2021, from https://stat.ethz.ch/R-manual/R-patched/library/stats/html/aov.html

ETHZ. (n.d.-b). R: Predict method for Linear Model Fits. Retrieved January 3, 2021, from https://stat.ethz.ch/R-manual/R-devel/library/stats/html/predict.lm.html

ETHZ. (n.d.-c). R: Random number generation. Retrieved from stat.ethz.ch website: https://stat.ethz.ch/R-manual/R-patched/library/base/html/Random.html

Hester, J. (n.d.). Read_delim function | r documentation. Retrieved December 14, 2020, from https://www.rdocumentation.org/packages/readr/versions/1.3.1/topics/read_delim

Long, J., & Teetor, P. (2019). 11 Linear Regression and ANOVA | R Cookbook, 2nd Edition. In *rc2e.com*. Retrieved from https://rc2e.com/linearregressionandanova#recipe-id218

Prabhakaran, S. (2019). Linear Regression With R. Retrieved from http://r-statistics.co/Linear-Regression.html

R documentation. (2019). Str function | R documentation. Retrieved from https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/str

R documentation. (n.d.-a). aov function | R Documentation. Retrieved January 3, 2021, from https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/aov

R documentation. (n.d.-b). Summary function | r documentation. Retrieved December 23, 2020, from https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/summary

Royce Kimmons. (2012). Exam Scores. Retrieved from http://roycekimmons.com/tools/generated_data/exams

Statistics Solution. (2021). Multiple Regression. Retrieved January 2, 2021, from https://www.statisticssolutions.com/regression-analysis-multiple-regression/#:~:text=Multiple%20regression%20requires%20two%20or

STHDA. (n.d.). One-Way ANOVA Test in R - Easy Guides - Wiki - STHDA. Retrieved January 2, 2021, from http://www.sthda.com/english/wiki/one-way-anova-test-in-r#what-is-one-way-anova-test

UCLA. (2018a). Choosing the Correct Statistical Test in SAS, Stata, SPSS and R. Retrieved January 2, 2021, from https://stats.idre.ucla.edu/other/mult-pkg/whatstat/

UCLA. (2018b). What statistical analysis should I use? Statistical analyses using R. Retrieved January 2, 2021, from https://stats.idre.ucla.edu/r/whatstat/what-statistical-analysis-should-i-usestatistical-analyses-using-r/#1repanovA

Yeager, K. (2020, November 10). LibGuides: SPSS Tutorials: One-Way ANOVA. Retrieved January 2, 2021, from https://libguides.library.kent.edu/spss/one-wayanova#:~:text=Data%20Set%2DUp

11. Appendix A. Boxplots of all independent vs. dependent variables

Gender

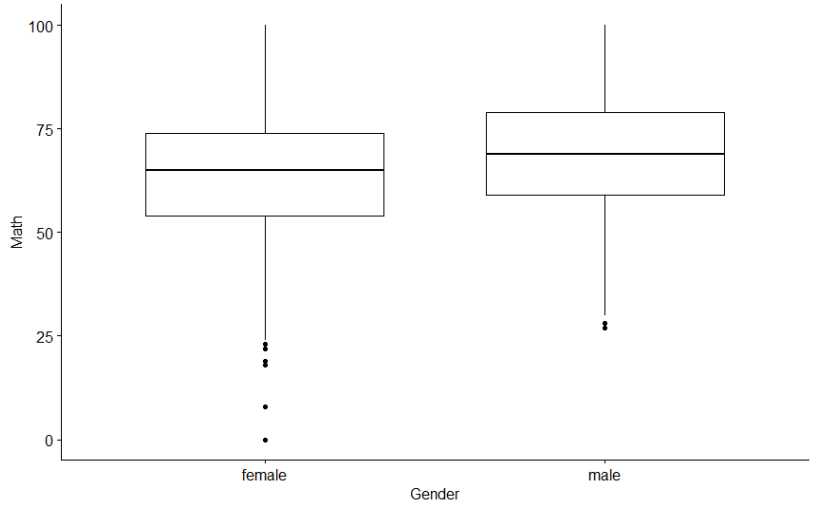

Figure 1: Boxplot Gender vs. Math

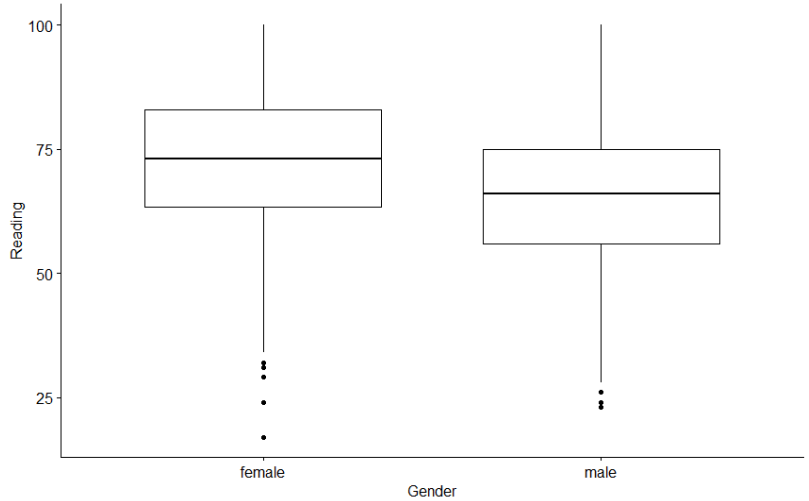

Figure 2: Boxplot Gender vs. Reading

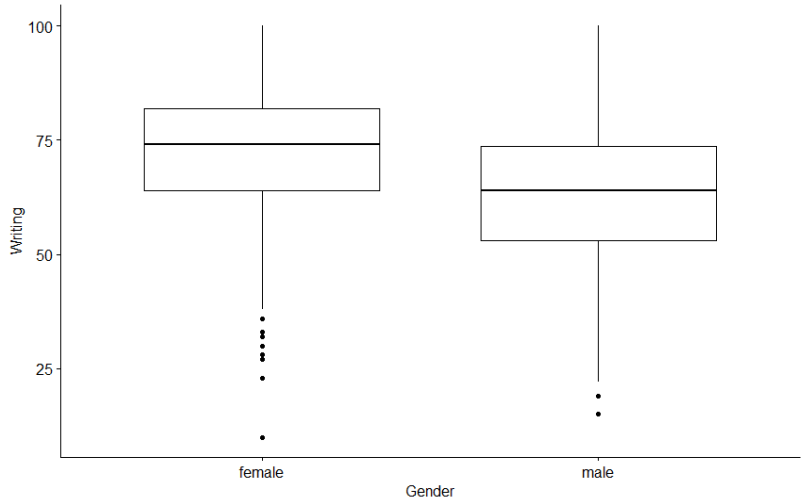

Figure 3: Boxplot Gender vs. Writing

Ethnicity

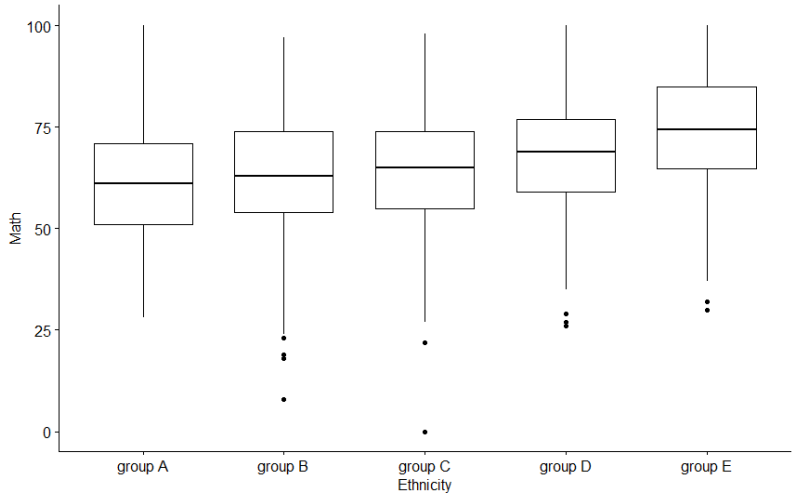

Figure 4: Boxplot Ethnicity vs. Math

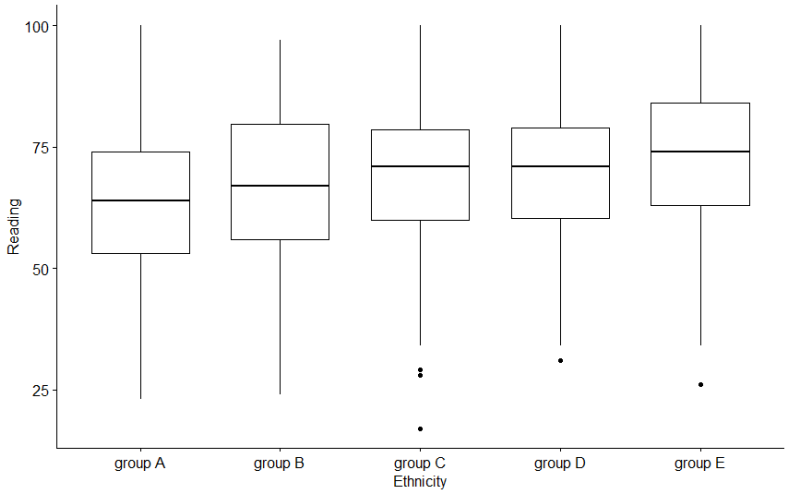

Figure 5: Boxplot Ethnicity vs. Reading

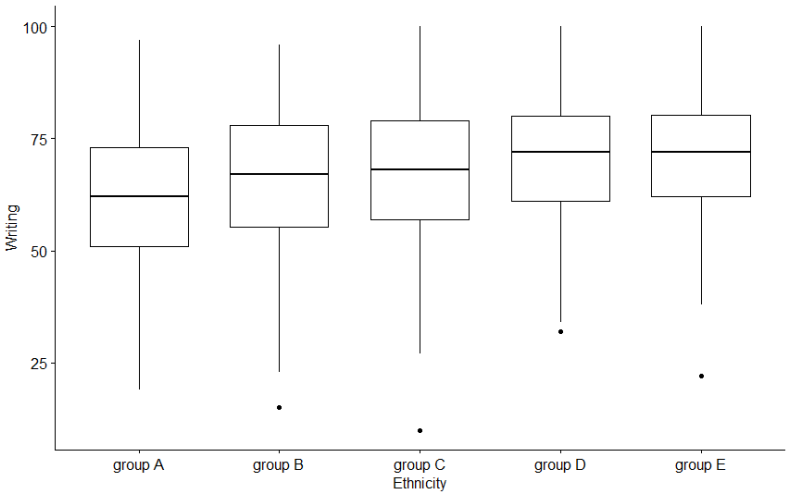

Figure 6: Boxplot Ethnicity vs. Writing

Parental Education

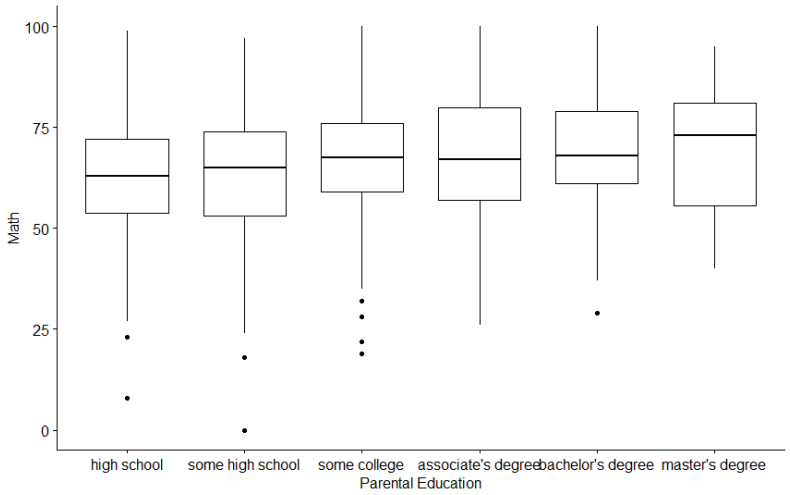

Figure 7: Boxplot Parental Education vs. Math

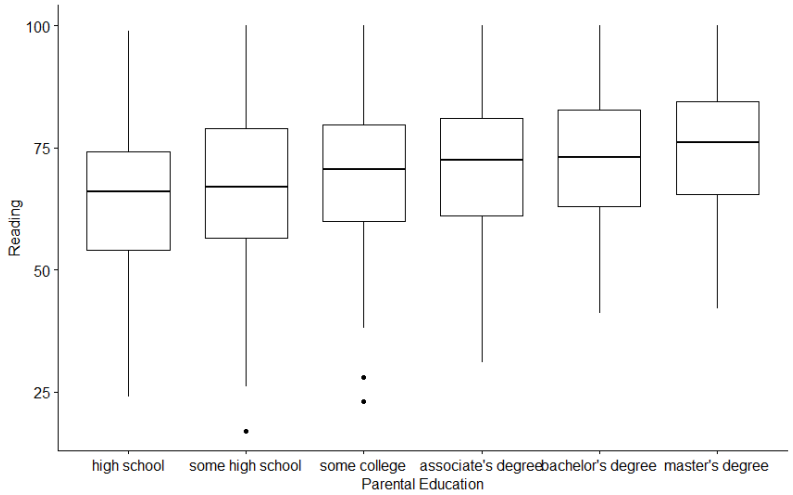

Figure 8: Boxplot Parental Education vs. Reading

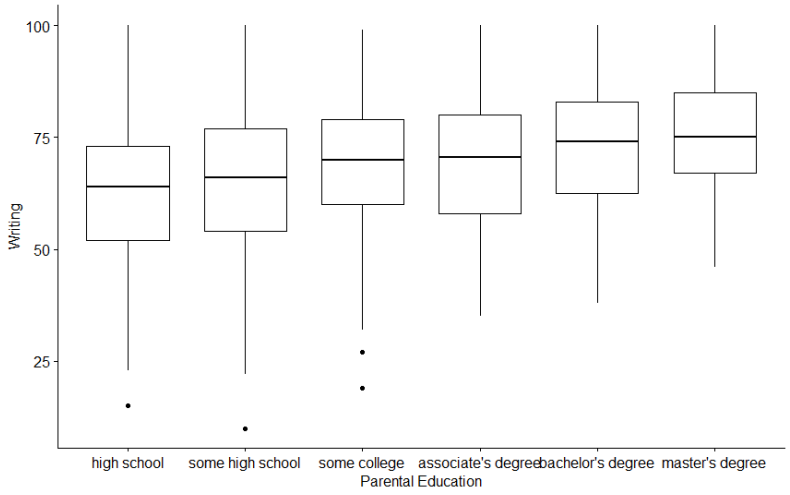

Figure 9: Boxplot Parental Education vs. Writing

Lunch

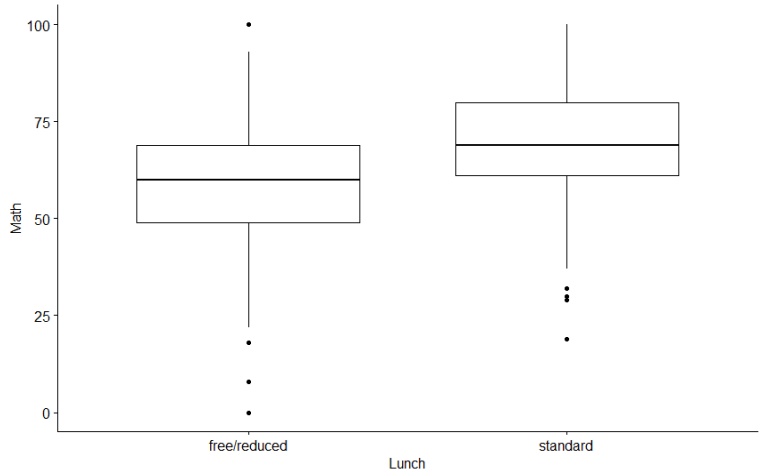

Figure 10: Boxplot Lunch vs. Math

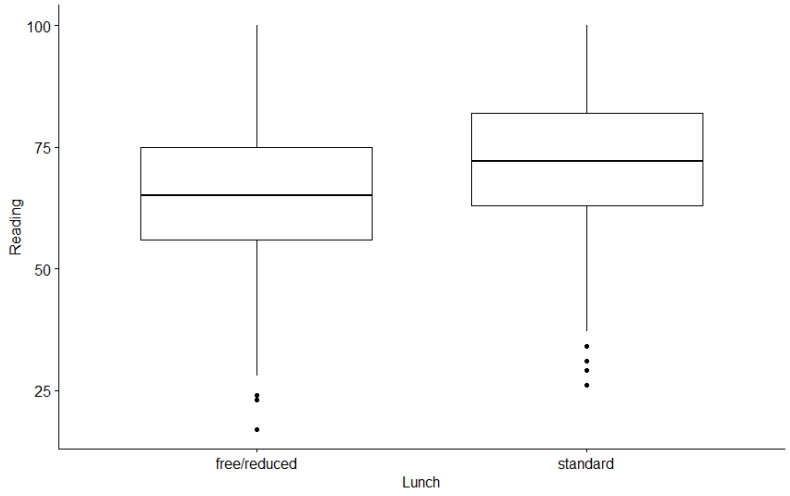

Figure 11: Boxplot Lunch vs. Reading

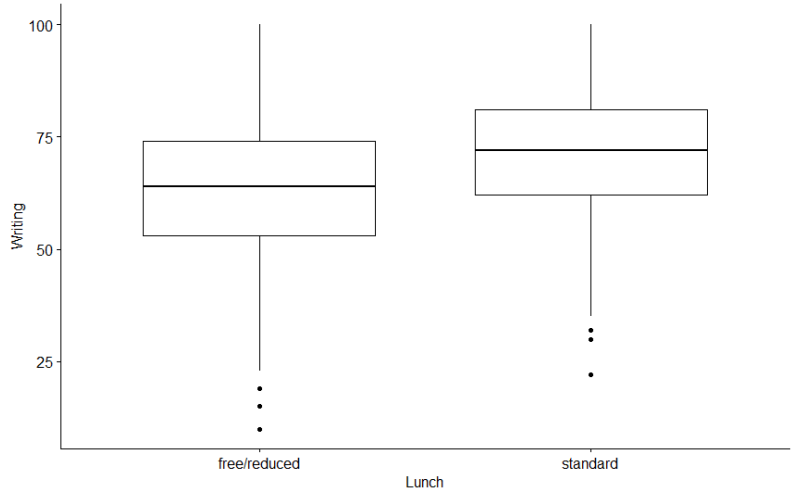

Figure 12: Boxplot Lunch vs. Writing

Preparation

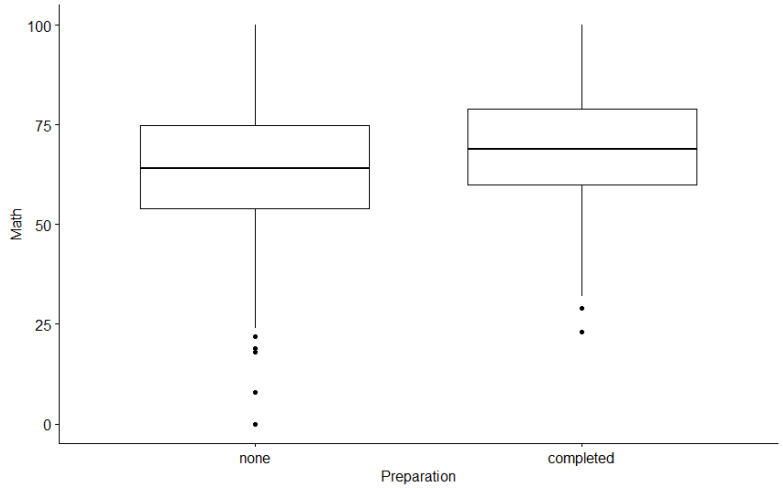

Figure 13: Boxplot Preparation vs. Math

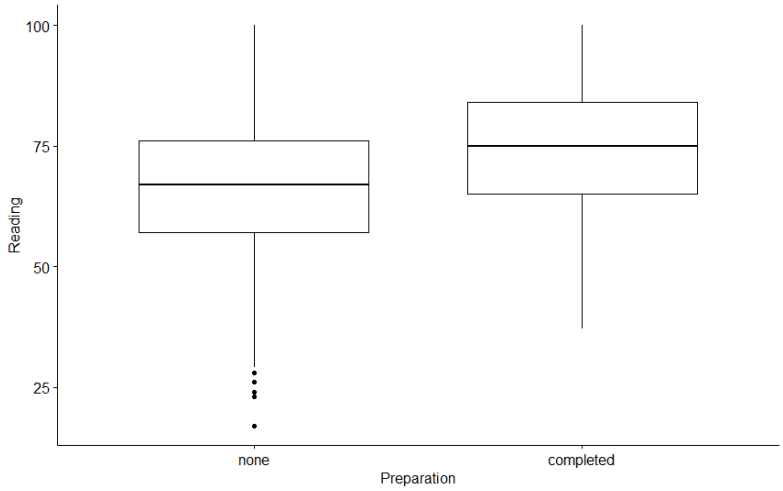

Figure 14: Boxplot Preparation vs. Reading

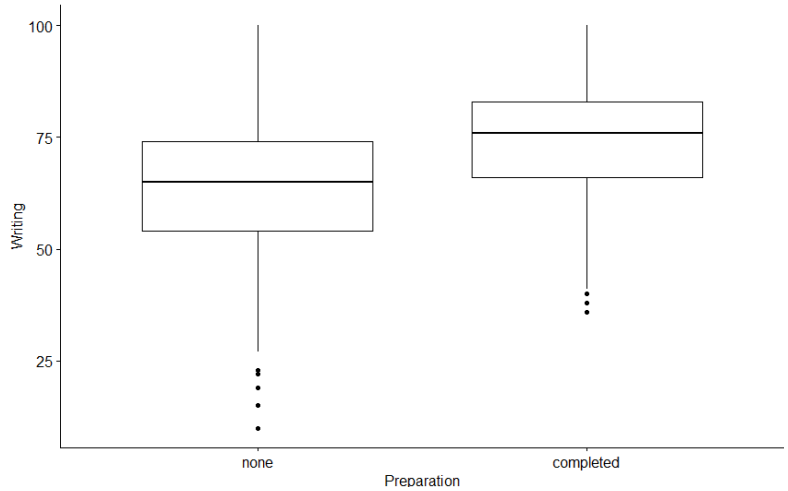

Figure 15: Boxplot Preparation vs. Writing