
# Programming for Data Analysis

| | | |
|---|---|---|
| **Module Code** | **:** | CT127-3-2-PDFA |
| **Intake Code** | **:** | APD2F2211IT(MBT) |
| **Lecturer Name** | **:** | Ms. Farhana Illiani binti Hassan |
| **Hand out Date** | **:** | Week 8 |
| **Hand in Date** | **:** | Week 13 |
| **Lab No.** | **:** | Lab 30 |

| **Student ID** | **Student Name** |
|---|---|
| *TP059963* | *YIP ZI XIAN* |

# Table of Contents

# 1.0 Introduction

This documentation aims to perform a comprehensive analysis of a given dataset that pertains to the placement of students within an organization. By leveraging the powerful analytical tools of RStudio with R programming languages, we can generate a variety of data charts and graphs to gain deeper insights into the underlying factors that impact students' placement. Our goal is to identify the root cause of any issues that may be hindering students' progress within the organization and provide meaningful and informative recommendations for improvement. Through this analysis, we can uncover patterns and trends in the data, identify potential areas for optimization, and ultimately help drive better outcomes for students and the organization.

## 1.1 Assumption

When analyzing a dataset of students' placements, it is crucial to ensure that the results are accurate and reliable. However, it is often impossible to account for every possible scenario or special case that may arise in the dataset. In such cases, assumptions must be made to fill in the gaps and ensure the analysis remains valid:

- All students are graduated from university and have a degree certification along with a Master of Business Association (MBA).
- All students' age is taken before they entered university.
- Line graph won't be appliable for this documentation as the dataset given lack of period or evolution.

## 1.2 Data Analysis Techniques

### 1.2.1 Data Import

Before analyzing data, a Comma-Separated Values or CSV file is needed and imported into RStudio to read the data available with the code below:

```
# Import the .csv file (Data Import)
placementData <- read.table(
  "C:/Users/User/Documents - Local/Degree/Sem 1/PFDA/Assignment/Assignment RScripts/placement_data.csv",
  header = TRUE,
  sep = ","
)
```

The code above creates a variable called as **placementData**. By using **read.table** function, three arguments or parameters, the local absolute file directory, header, and separator, are parsed to satisfy the function. The **header** parameter is to indicate whether the .csv file contains the column title while the **sep** parameter is to set a value to separate all data in the file. Since the file is a .csv file, thus **comma** is used.

### 1.2.2 Data Cleaning

To obtain accurate and unbiased conclusions, it is important to perform data cleaning to address any invalid values such as **NULL or NA** within the imported dataset. By performing data cleaning, the dataset will be more reliable and trustworthy, free from any errors or inconsistencies that could alter the results.

There are several ways to perform data cleaning:

a. **Omitting rows containing NULL or NA values.** However, it is not advisable because the result might be skewed due to the number of dropped rows. **This method is used as an example and not implemented in this documentation.**

```
# Remove data without salary (Data Cleaning)
newPlacementData <- na.omit(placementData)
```

b. **Removing duplicated rows.** It is a must to perform this method because it will reduce the biasedness and redundancy of the result. **This method is used in this documentation.**

```
# Check for duplicated rows (Data Cleaning)
duplicated_rows <- duplicated(placementData)

# Print the number of duplicated rows (Data Cleaning)
cat("Number of duplicated rows:", sum(duplicated_rows), "\n")

# Remove duplicated rows (Data Cleaning)
noRedData <- placementData[!duplicated_rows, ]
```

c. **Altering NULL or NA values to empty string or 0.** This is most suitable method to clean a dataset because it won't remove any rows within the dataset and the result also won't be altered. **This method is used in this documentation.**

```
# Change NA values to 0 (Data Cleaning)
placementData[is.na(placementData)] <- 0
```

### 1.2.3 Data Pre-processing / Transformation

Data pre-processing and transformation are not just encouraged, but essential steps in the data analysis process, especially when dealing with large and complex datasets. It is common to encounter raw data with weird and illogical naming or headers which can lead to human error thus affecting the accuracy of the analysis. By performing data pre-processing and transformation, quality of the data can be improved and more manageable and understandable.

```
# Change the name of headers (Data Pre-processing / Transformation)
alteredHeaderNames <- c(
  "UID", "Gender", "Age", "Address", "Mother_Education", "Father_Education",
  "Mother_Current_Job", "Father_Current_Job", "Family_Support", "Paid Classes",
  "Curricular_Activities", "Internet_Usage", "Secondary_Grade_Percentage",
  "Secondary_Education_Board", "Higher_Secondary_Grade_Percentage",
  "Higher_Secondary_Education_Board", "Higher_Secondary_Specialism",
  "Degree_Grade_Percentage", "Degree_Specialism", "Working_Experience",
  "Employment_Test", "Working_Specialism", "Master_Grade_Percentage",
  "Placement_Status", "Salary"
)

names(placementData) <- alteredHeaderNames
names(newPlacementData) <- alteredHeaderNames
```

### 1.2.4 Data Visualization

Raw data can be visualized to see the patterns or frequencies to make a general conclusion or have a better view of the whole imported dataset. By using GGPlot2 library, graphs and plot can be created for an easier verdict to be made.

```
# Use line graph and count plot to generate some random raw analysis about the dataset
# - Age (Count Plot)
df_age <- data.frame(
  studentAge = as.vector(placementData$Age)
)

ggplot(df_age, aes(x = studentAge, fill = factor(studentAge))) +
  geom_bar() +
  scale_fill_manual(values = c("#7400B8", "#5E60CE", "#4EA8DE", "#56CFE1", "#72EFDD", "#52B788")) +
  labs(
    x = "Student Age",
    y = "Count",
    title = "Count Plot of Student Age",
    fill = "Age Category"
  )
```

### 1.2.5 Data Manipulation

Data manipulation is another crucial step in data analysis process that involves modifying, restructuring, and transforming the raw data to provide a better understanding of the underlying patterns and trends within the dataset. Categorical number values can be changed into informative keywords to extract more meaningful insights.

```
# Replace R to Rural, U to Urban (Data Manipulation)
placementData$address <- gsub("R", "Rural", placementData$address)
placementData$address <- gsub("U", "Urban", placementData$address)

placementData$gender <- gsub("F", "Female", placementData$gender)
placementData$gender <- gsub("M", "Male", placementData$gender)

# Replace numbering to education levels
for (col in c("Medu", "Fedu")) {
  placementData[[col]][placementData[[col]] == 0] <- "No Education"
  placementData[[col]][placementData[[col]] == 1] <- "Primary Education"
  placementData[[col]][placementData[[col]] == 2] <- "Secondary Education"
  placementData[[col]][placementData[[col]] == 3] <- "Degree Level"
  placementData[[col]][placementData[[col]] == 4] <- "Post Graduate"
}
```

### 1.2.6 Data Exploration

Data exploration is a fundamental step in the data analysis process that plays a crucial role in uncovering valuable insights or patterns or both from raw data provided, enabling a deeper understanding of the data and make informed decisions based on the findings.

```
# Get summary of the filtered .csv file (Data exploration)
summary(placementData)
summary(newPlacementData)

names(placementData)
names(newPlacementData)

nrow(placementData)
nrow(newPlacementData)

ncol(placementData)
ncol(newPlacementData)

str(placementData)
str(newPlacementData)

head(placementData)
head(newPlacementData)

tail(placementData)
tail(newPlacementData)
```

The code above is some of the built-in functions in R programming:

a. **summary() –** It will return a summary of the whole imported dataset, for instance the minimum and maximum value, first and third quartile, median, mean, mode and length of the imported dataset.

```
      UID            Gender
 Min.   :    1   Length:17007
 1st Qu.: 4252   Class :character
 Median : 8504   Mode  :character
 Mean   : 8504
 3rd Qu.:12756
 Max.   :17007
```

b. **names() –** It will return the names of the headers.

```
 [1] "UID"                          "Gender"                       "Age"                                "Address"
 [5] "Mother_Education"             "Father_Education"             "Mother_Current_Job"                 "Father_Current_Job"
 [9] "Family_Support"               "Paid Classes"                 "Curricular_Activities"              "Internet_Usage"
[13] "Secondary_Grade_Percentage"   "Secondary_Education_Board"    "Higher_Secondary_Grade_Percentage"  "Higher_Secondary_Education_Board"
[17] "Higher_Secondary_Specialism"  "Degree_Grade_Percentage"      "Degree_Specialism"                  "Working_Experience"
[21] "Employment_Test"              "Working_Specialism"           "Master_Grade_Percentage"            "Placement_Status"
[25] "Salary"
```

c. **nrow() & ncol() –** It will display the total rows and columns of the imported dataset.

```
> nrow(placementData)
[1] 17007
> nrow(newPlacementData)
[1] 8742
>
> ncol(placementData)
[1] 25
> ncol(newPlacementData)
[1] 25
```

d. **str() –** It will display the data structure or data type of the imported dataset.

```
'data.frame':   17007 obs. of  25 variables:
 $ UID                                : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Gender                             : chr  "Male" "Male" "Male" "Male" ...
 $ Age                                : int  23 19 19 21 22 19 19 18 19 21 ...
 $ Address                            : chr  "Urban" "Urban" "Urban" "Urban" ...
 $ Mother_Education                   : int  4 1 1 4 3 4 2 4 3 3 ...
 $ Father_Education                   : int  4 1 1 2 3 3 2 4 2 4 ...
 $ Mother_Current_Job                 : chr  "at_home" "at_home" "at_home" "health" ...
 $ Father_Current_Job                 : chr  "teacher" "other" "other" "services" ...
 $ Family_Support                     : chr  "no" "yes" "no" "yes" ...
 $ Paid Classes                       : chr  "no" "no" "yes" "yes" ...
 $ Curricular_Activities              : chr  "no" "no" "no" "yes" ...
 $ Internet_Usage                     : chr  "no" "yes" "yes" "yes" ...
 $ Secondary_Grade_Percentage         : num  67 79.3 65 56 85.8 ...
 $ Secondary_Education_Board          : chr  "State" "State" "Private" "Central" ...
 $ Higher_Secondary_Grade_Percentage: num  91 78.3 68 52 73.6 ...
 $ Higher_Secondary_Education_Board : chr  "State" "Central" "Private" "State" ...
 $ Higher_Secondary_Specialism        : chr  "Commerce" "Science" "Arts" "Science" ...
 $ Degree_Grade_Percentage            : num  58 77.5 64 52 73.3 ...
 $ Degree_Specialism                  : chr  "Sci&Tech" "Sci&Tech" "Comm&Mgmt" "Sci&Tech" ...
 $ Working_Experience                 : chr  "No" "Yes" "No" "No" ...
 $ Employment_Test                    : num  55 86.5 75 66 96.8 ...
 $ Working_Specialism                 : chr  "Mkt&HR" "Mkt&Fin" "Mkt&Fin" "Mkt&HR" ...
 $ Master_Grade_Percentage            : int  78 80 77 50 86 63 59 83 51 67 ...
 $ Placement_Status                   : chr  "Placed" "Placed" "Placed" "Not Placed" ...
 $ Salary                             : num  350000 200000 350000 0 250000 0 0 300000 350000 0 ...
```

e. **head() & tail() –** It will display the first six and last six row of the imported dataset, the number of rows displayed can be manually change to suit one's needs.

```
> head(placementData)
  UID Gender Age Address
1   1   Male  23   Urban
2   2   Male  19   Urban
3   3   Male  19   Urban
4   4   Male  21   Urban
5   5   Male  22   Urban
6   6   Male  19   Urban
```

```
> tail(placementData)
          UID Gender Age
17002 17002   Male  20
17003 17003 Female  18
17004 17004   Male  19
17005 17005 Female  19
17006 17006 Female  20
17007 17007 Female  21
```

### 1.2.7 Environment Cleaning

Besides cleaning the dataset, the RStudio environments need to be cleaned to ensure that the unused variables declared in another file will not tamper or have conflicts with the current variables. By doing so, the accuracy and reliability of the analysis can be maintained.

```
# Clean environment to ensure data are updated
rm(list = ls())
```

### 1.2.8 Plot Cleaning

Since data graphs will be generated, it is important to remove previously generated graph to avoid additional memory usage from RStudio so that the device will not have lagging or performance issues. It is also to avoid misinterpretation from previous generated graphs with current generated graphs.

```
# Clean plot graphs
if(!is.null(dev.list())) dev.off()
```

### 1.2.9 Library Installation and Import

Relying only on RStudio built-in functions will not be enough to provide a more detailed analysis. External libraries can be installed and imported to RStudio to include more additional functions and features to the data analysis process.

```
# Install relevant libraries
install.packages(c("crayon", "dplyr", "ggplot2", "reshape2", "stringr", "Hmisc"))
library(crayon)
library(dplyr)
library(ggplot2)
library(reshape2)
library(stringr)
library(Hmisc)
```

Here are some libraries implemented in this documentation:

a. Crayon – To style console output with color and formatting options so that results and error messages can be seen clearly.
b. Dplyr – To filter, summarise, arrange, join data together for a better data manipulation process.

c. GGPlot2 – To provide a flexible and powerful framwork for creating more data visualizations like generating graphs and plots.
d. Reshape2 – To transform and reshape data between different formats, providing functions to melt and cast data frames and pivot and unpivot data.
e. Stringr – To provide string manipulation and formatting.
f. Hmisc – To include functions for data cleaning and imputation, regression modeling etc.

## 1.3 Raw Data Exploration and Visualization

Some raw data are explored to have a better insight of the imported dataset before actual analysis to check whether the dataset is more inclined towards any factors like age, address, gender etc.

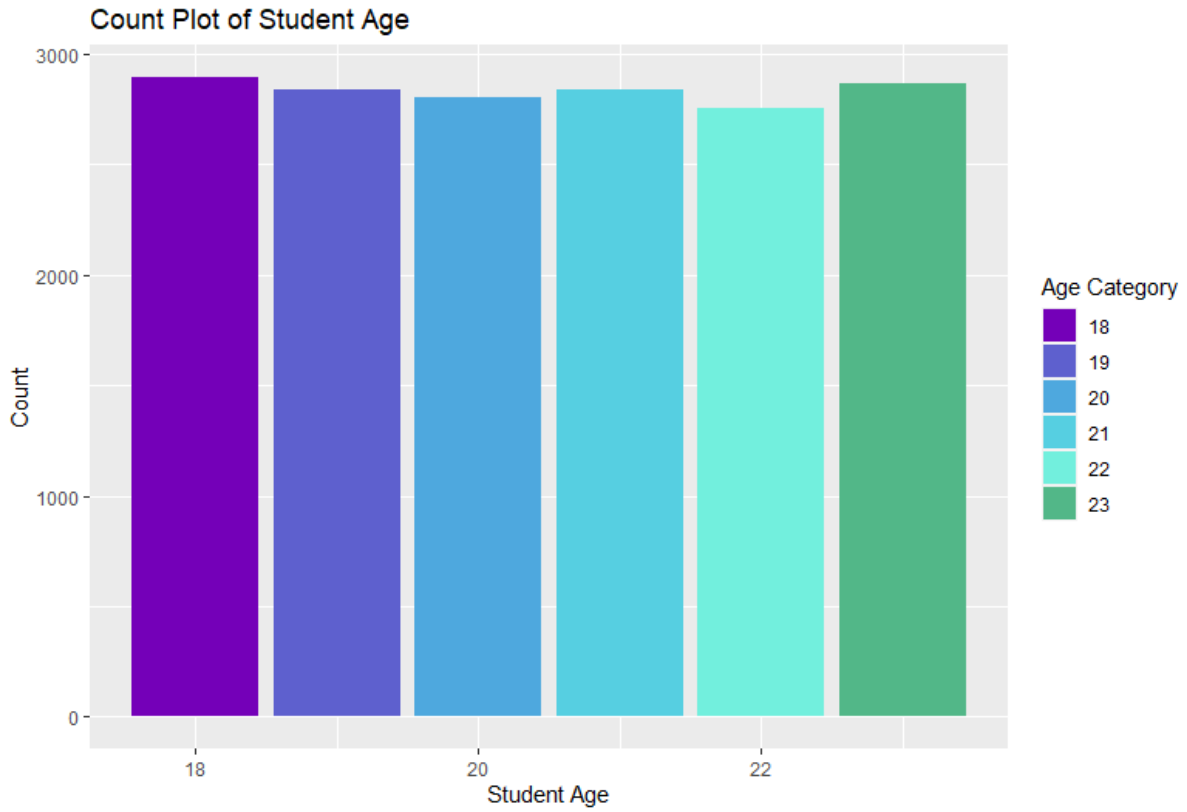
### 1.3.1 Age

```
# - Age (Count Plot)
df_age <- data.frame(
  studentAge = as.vector(placementData$Age)
)

ggplot(df_age, aes(x = studentAge, fill = factor(studentAge))) +
  geom_bar() +
  scale_fill_manual(values = c(
    "#7400B8", "#5E60CE", "#4EA8DE", "#56CFE1", "#72EFDD", "#52B788"
  )) +
  labs(
    x = "Student Age",
    y = "Count",
    title = "Count Plot of Student Age",
    fill = "Age Category"
  )
```

The code above demonstrates on how to generate a **count plot of students' age** using GGPlot2 library. A data frame, **df_age,** is first created to store a vector consisting of **students' age**. Then, **ggplot()** function is used to generate the count plot with the declared data frames and **aes()** function. The **aes()** function means "aesthetics" which is to provide visual properties like x-axis, y-axis, color, size, shape and etc. into a graph. For example, x-axis will be **student age** and different color will fill up the count plot to differentiate each age group.

**geom_bar()** function is to create bar plots. **scale_fill_manual()** function is to customize the color to be used in the graph to distinguish each age group. **labs()** function is to provide meaningful titles and labels to have a better overall readability of the graph.

The output of the code is as below:



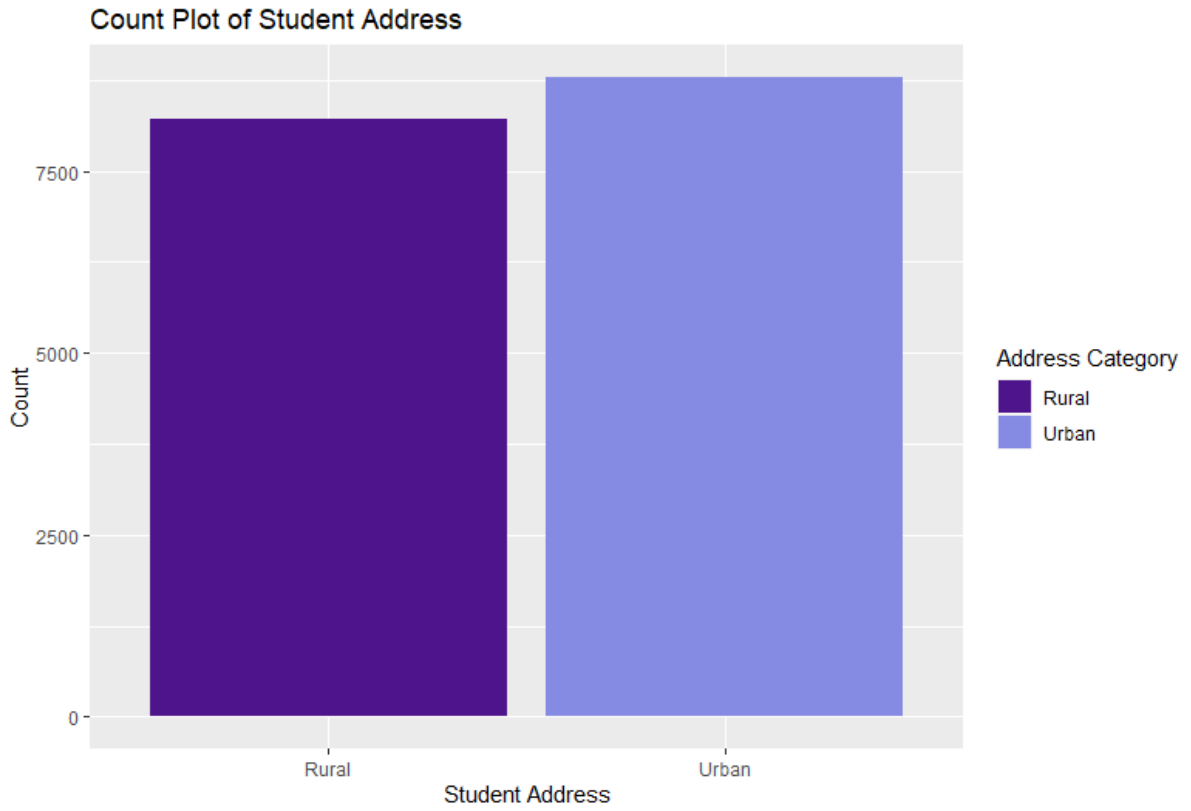

From this count plot, students with age of 22 are the least while students with age of 18 and 23 are the most within the imported dataset.

### 1.3.2 Address

```
# - Address (Count Plot)
df_address <- data.frame(
  studentAddress = as.vector(placementData$Address)
)

ggplot(df_address, aes(x = studentAddress, fill = factor(studentAddress))) +
  geom_bar() +
  scale_fill_manual(values = c("#4E148C", "#858AE3")) +
  labs(
    x = "Student Address",
    y = "Count",
    title = "Count Plot of Student Address",
    fill = "Address Category"
  )
```

The code above demonstrates on how to generate a **count plot of students' address** using GGPlot2 library.

The output of the code is as below:



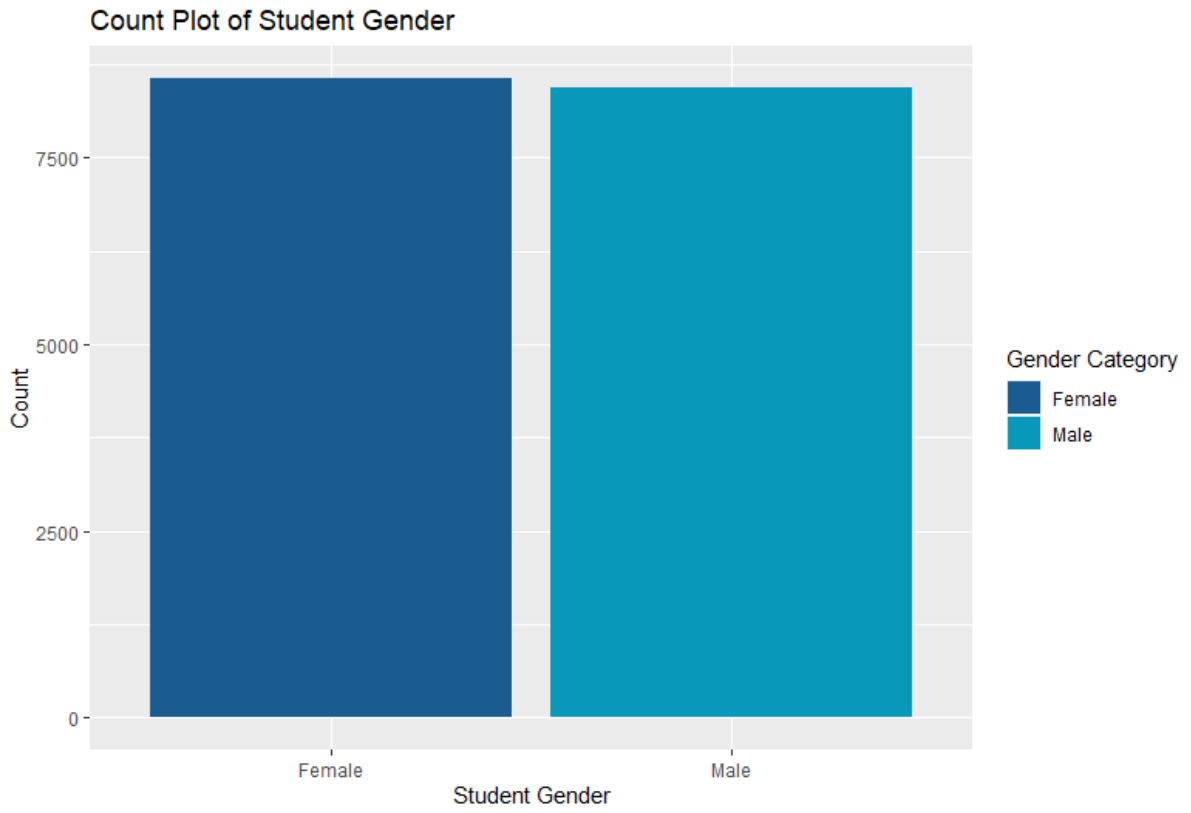

From this count plot, most students lived in an urban environment rather than a rural environment.

### 1.3.3 Gender

```
# - Gender (Count Plot)
df_gender <- data.frame(
  studentGender = as.vector(placementData$Gender)
)

ggplot(df_gender, aes(x = studentGender, fill = factor(studentGender))) +
  geom_bar() +
  scale_fill_manual(values = c("#1A5B92", "#0899BA")) +
  labs(
    x = "Student Gender",
    y = "Count",
    title = "Count Plot of Student Gender",
    fill = "Gender Category"
  )
```

The code above demonstrates on how to generate a **count plot of students' gender** using GGPlot2 library.

The output of the code is as below:



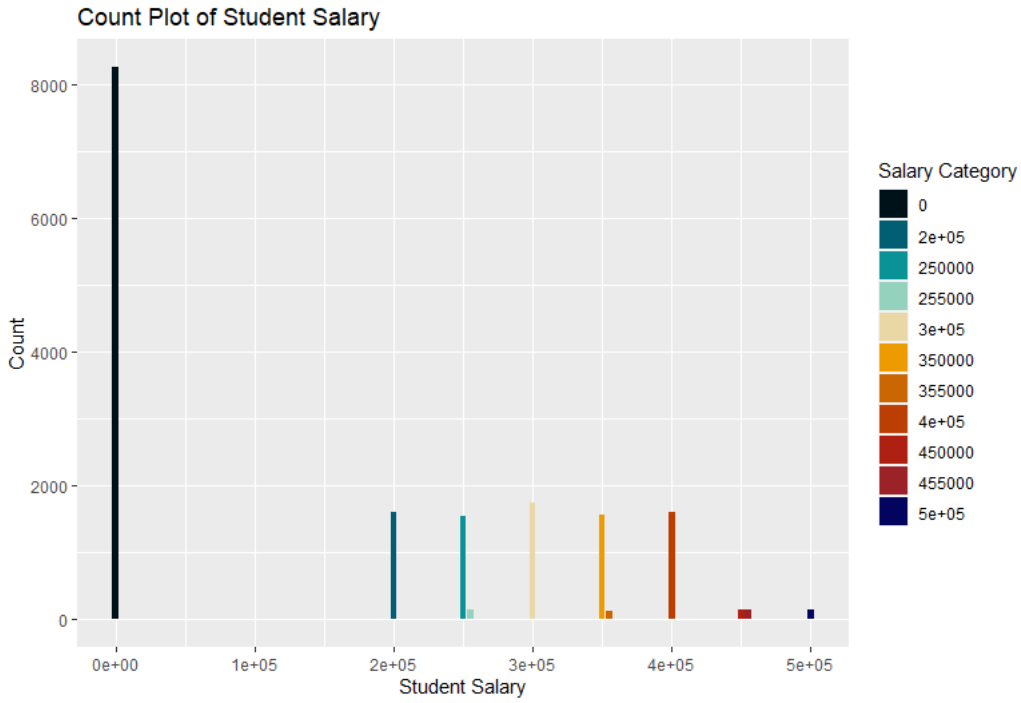

From this count plot, female students are slightly more than male students but with such large datasets, the result analyzed won't be gender biased.

### 1.3.4 Salary

```
# - Salary (Count Plot)
df_salary <- data.frame(
  studentSalary = as.vector(placementData$Salary)
)

ggplot(df_salary, aes(x = studentSalary, fill = factor(studentSalary))) +
  geom_bar() +
  scale_fill_manual(
    values = c(
      "#001219", "#005F73",
      "#0A9396", "#94D2BD",
      "#E9D8A6", "#EE9B00",
      "#CA6702", "#BB3E03",
      "#AE2012", "#9B2226",
      "#03045E", "#0077B6"
    )
  ) +
  labs(
    x = "Student Salary",
    y = "Count",
    title = "Count Plot of Student Salary",
    fill = "Salary Category"
  )
```

The code above demonstrates on how to generate a **count plot of students' salary** using GGPlot2 library.

The output of the code is as below:



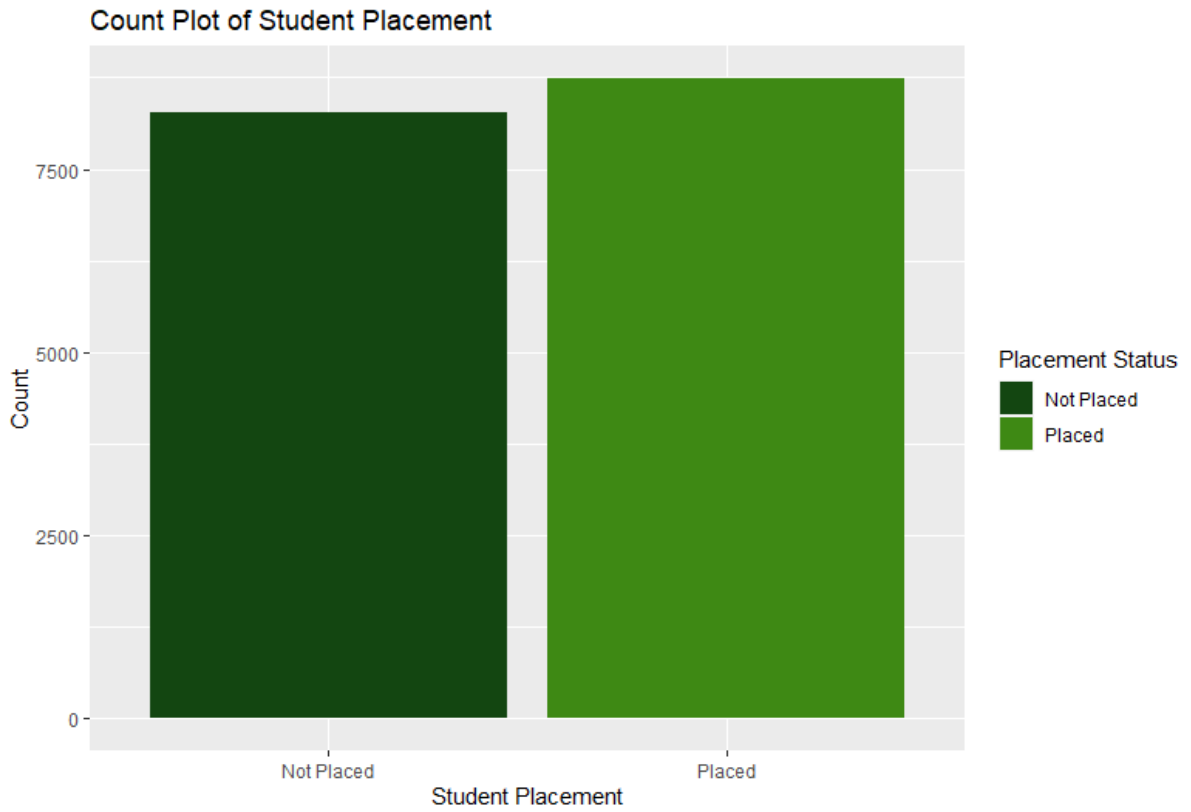

From this count plot, more than 8000 students do not have a salary and only a tiny group of students are earning RM 500K salary while other students earn around RM 200K to RM455K.

### 1.3.5 Placement

```
# - Placement (Count Plot)
df_placement <- data.frame(
  studentPlacement = as.vector(placementData$Placement_Status)
)

ggplot(df_placement, aes(x = studentPlacement, fill = factor(studentPlacement))) +
  geom_bar() +
  scale_fill_manual(values = c("#134611", "#3E8914")) +
  labs(
    x = "Student Placement",
    y = "Count",
    title = "Count Plot of Student Placement",
    fill = "Placement Status"
  )
```

The code above demonstrates on how to generate a **count plot of students' placement** using GGPlot2 library.

The output of the code is as below:




From this count plot, despite the analysis about students' salary in the previous count plot, most students already have a placement within the organization.

## 2.0 Question 1 - What will affect students to get a placement?

Multiple factors from the imported dataset are used to investigate what will affect students' placement within the organization by creating **bar graph** as data visualization.

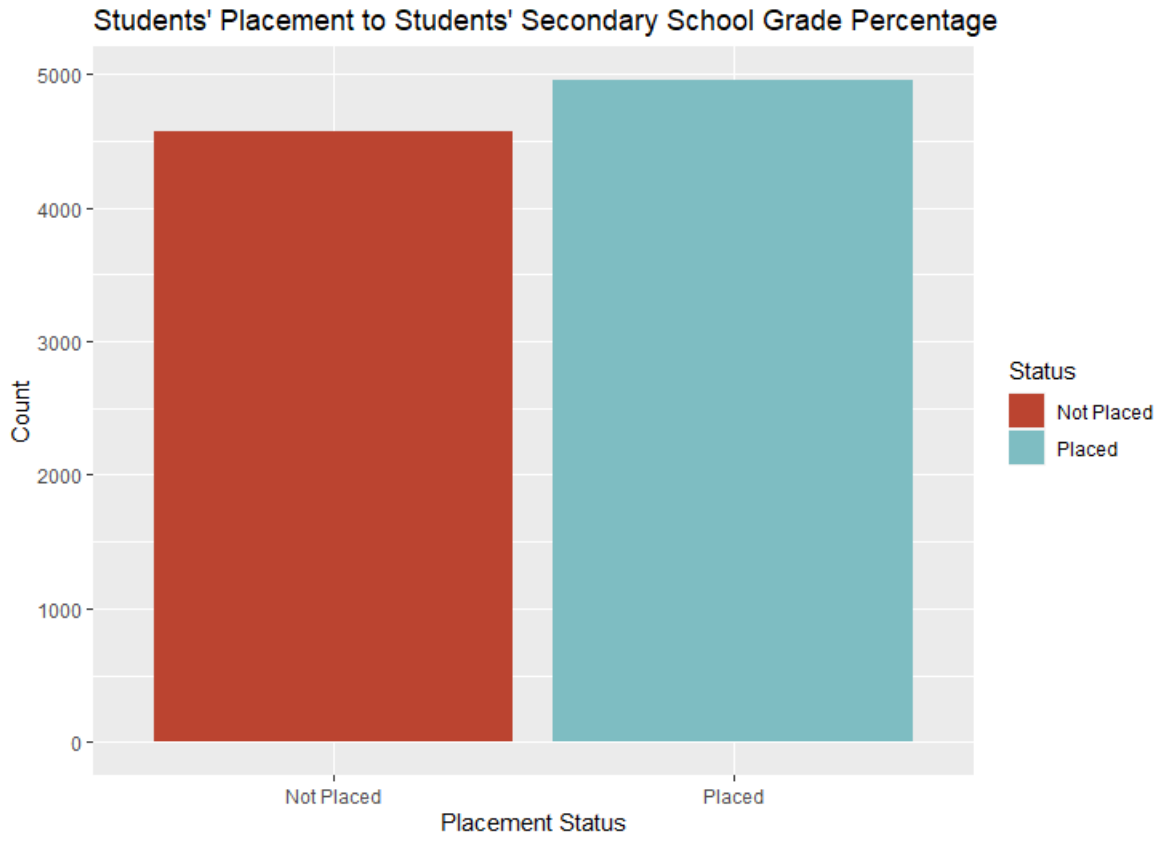
### 2.1 Analysis 1.1 – Secondary School Grade higher than 70 marks

```
# - Secondary school grade percentage (If higher than 70 marks)
ssc_p_placed <- sum(
  placementData$Placement_Status == "Placed" &
  placementData$Secondary_Grade_Percentage >= 70
)
ssc_p_not_placed <- sum(
  placementData$Placement_Status == "Not Placed" &
  placementData$Secondary_Grade_Percentage >= 70
)

ssc_data_for_placement_status <- data.frame(
  Status = c("Placed", "Not Placed"),
  Count = c(ssc_p_placed, ssc_p_not_placed)
)

ggplot(ssc_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Placement Status") +
  ylab("Count") +
  ggtitle("Students' Placement to Students' Secondary School Grade Percentage") +
  scale_fill_manual(values = c("#BB4430", "#7EBDC2"))
```

The code above demonstrates on how to create a bar chart of **students' placement to students' secondary school grade percentage** using GGPlot2 library. The number of students with **secondary school grade percentage higher than 70 marks** along with their **placement status** are being calculated. Then, the values parsed into a vector, **Count**, within a data frame, **ssc_data_for_placement_status.**

Since it is a bar chart, the y-axis will also be defined in the **aes()** function unlike count plot which is not needed. Within the **geom_bar()** function, **stats = identity** is used to specify the height of the bars correspond to the values of y-axis defined while **position = dodge** is to allow the bars to be grouped side-by-side by a categorical variable, in this case, is the values of x-axis. **xlab(), ylab(), and ggtitle()** is to set the label for x-axis, y-axis and graph title respectively.

The output of the code is as below:



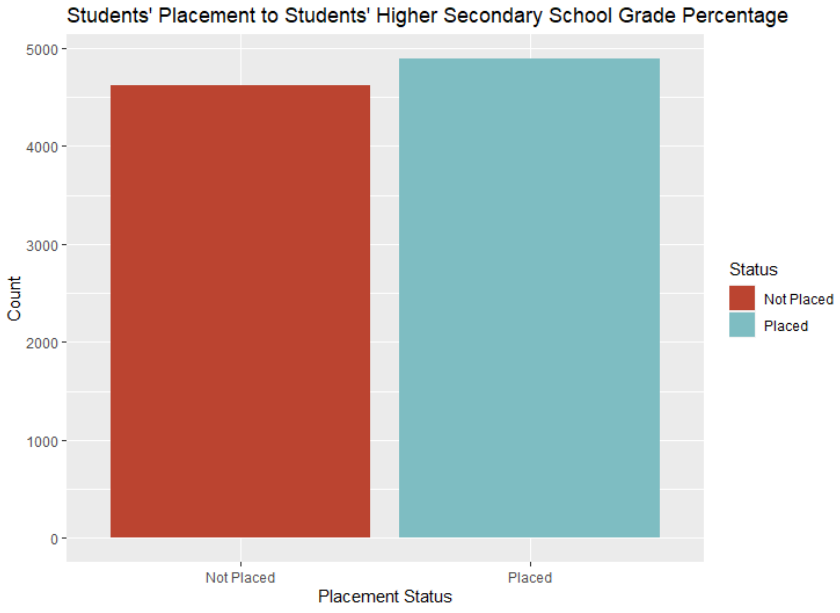

From this bar chart, it is clear that students with secondary school grade percentage higher than 70 marks are preferable and has a higher chance to be placed in the organization.

## 2.2 Analysis 1.2 – Higher Secondary School Grade higher than 70 marks

```
# - Higher secondary school grade percentage (If higher than 70 marks)
hsc_p_placed <- sum(
  placementData$Placement_Status == "Placed" &
  placementData$Higher_Secondary_Grade_Percentage >= 70
)
hsc_p_not_placed <- sum(
  placementData$Placement_Status == "Not Placed" &
  placementData$Higher_Secondary_Grade_Percentage >= 70
)

hsc_data_for_placement_status <- data.frame(
  Status = c("Placed", "Not Placed"),
  Count = c(hsc_p_placed, hsc_p_not_placed)
)

ggplot(hsc_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Placement Status") +
  ylab("Count") +
  ggtitle("Students' Placement to Students' Higher Secondary School Grade Percentage") +
  scale_fill_manual(values = c("#BB4430", "#7EBDC2"))
```

The code above demonstrates on how to create a bar chart of **students' placement to student's higher secondary school grade percentage** using GGPlot2 library.

The output of the code is as below:



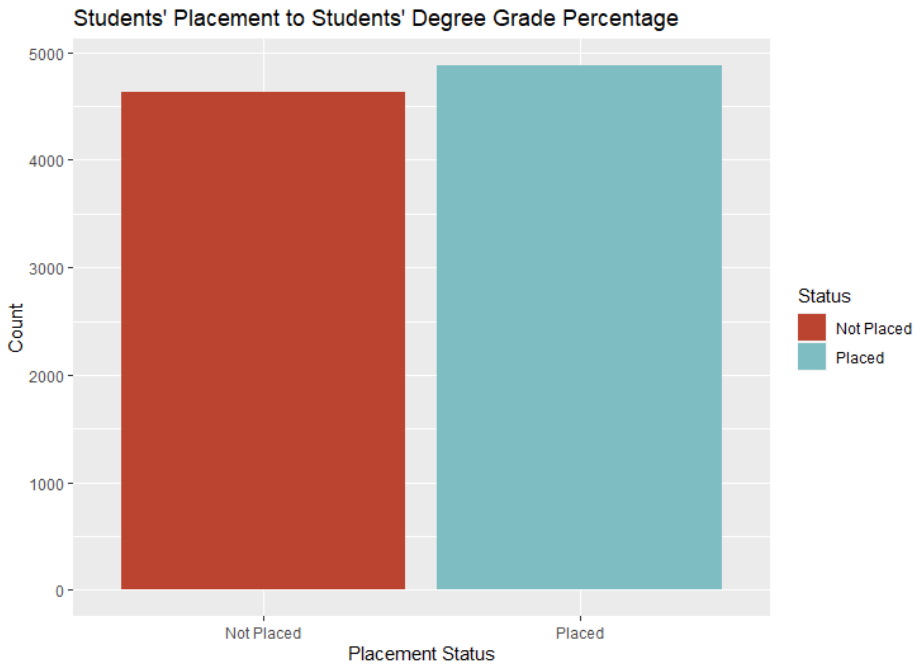

From this bar chart, it is clear that students with higher secondary school grade percentage higher than 70 marks are preferable and has a higher chance to be placed in the organization.

## 2.3 Analysis 1.3 – Degree Grade higher than 70 marks

```
# - Degree grade (If higher than 70 marks)
degree_p_placed <- sum(
  placementData$Placement_Status == "Placed" &
  placementData$Degree_Grade_Percentage >= 70
)
degree_p_not_placed <- sum(
  placementData$Placement_Status == "Not Placed" &
  placementData$Degree_Grade_Percentage >= 70
)

degree_data_for_placement_status <- data.frame(
  Status = c("Placed", "Not Placed"),
  Count = c(degree_p_placed, degree_p_not_placed)
)

ggplot(degree_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Placement Status") +
  ylab("Count") +
  ggtitle("Students' Placement to Students' Degree Grade Percentage") +
  scale_fill_manual(values = c("#BB4430", "#7EBDC2"))
```

The code above demonstrates on how to create a bar chart of **students' placement to student's degree grade percentage** using GGPlot2 library.

The output of the code is as below:



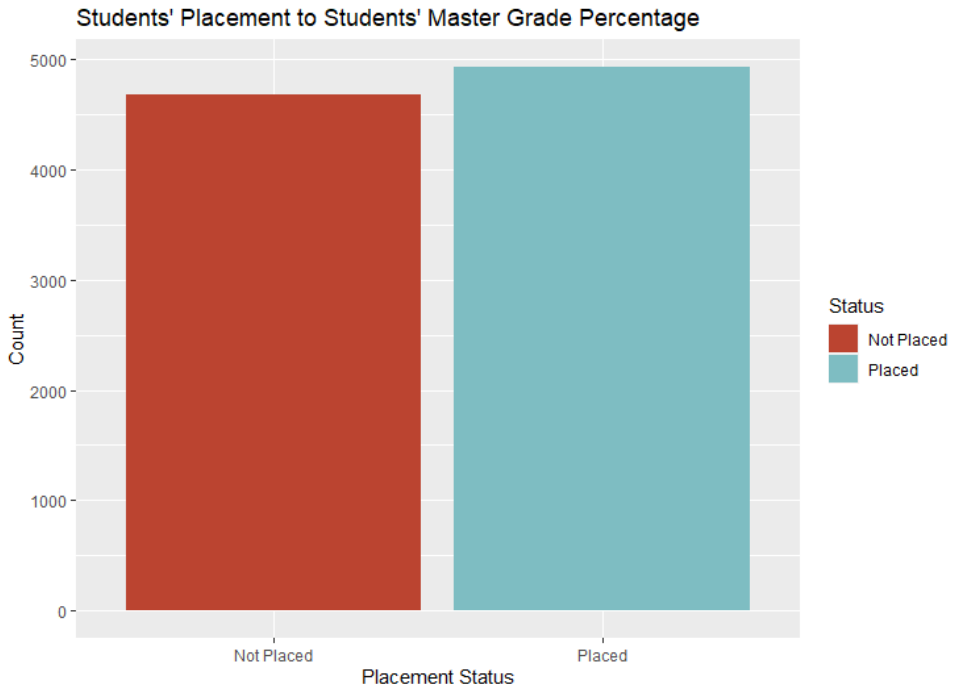

From this bar chart, it is clear that students with degree grade percentage higher than 70 marks are preferable and has a higher chance to be placed in the organization.

## 2.4 Analysis 1.4 – Master Grade higher than 70 marks

```
# - Master grade (If higher than 70 marks)
mba_p_placed <- sum(
  placementData$Placement_Status == "Placed" &
  placementData$Master_Grade_Percentage >= 70
)
mba_p_not_placed <- sum(
  placementData$Placement_Status == "Not Placed" &
  placementData$Master_Grade_Percentage >= 70
)

mba_data_for_placement_status <- data.frame(
  Status = c("Placed", "Not Placed"),
  Count = c(mba_p_placed, mba_p_not_placed)
)

ggplot(mba_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Placement Status") +
  ylab("Count") +
  ggtitle("Students' Placement to Students' Master Grade Percentage") +
  scale_fill_manual(values = c("#BB4430", "#7EBDC2"))
```

The code above demonstrates on how to create a bar chart of **students' placement to student's master grade percentage** using GGPlot2 library.

The output of the code is as below:



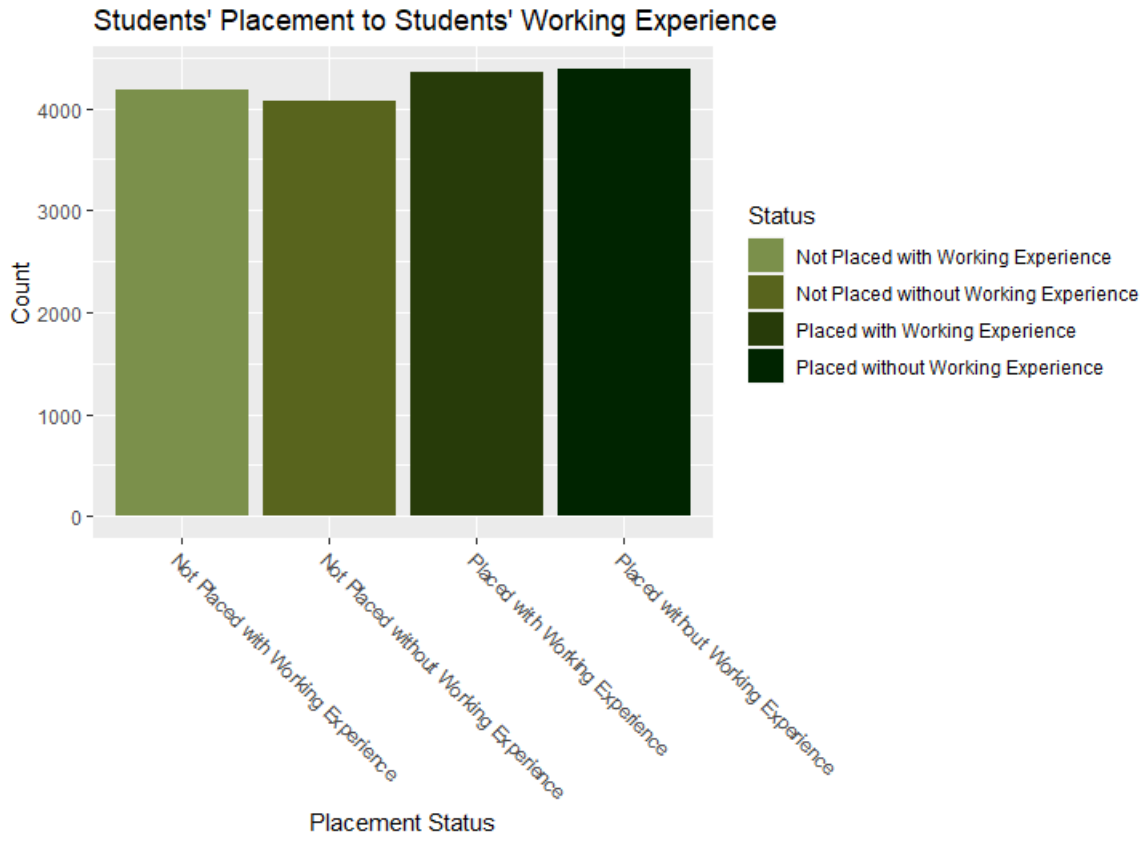

From this bar chart, it is clear that students with master grade percentage higher than 70 marks are preferable and has a higher chance to be placed in the organization.

## 2.5 Analysis 1.5 – Working Experience

```r
# - Working experience
workex_placed_yes <- sum(placementData$Placement_Status == "Placed" & placementData$Working_Experience == "Yes")
workex_not_placed_yes <- sum(placementData$Placement_Status == "Not Placed" & placementData$Working_Experience >= "Yes")
workex_placed_no <- sum(placementData$Placement_Status == "Placed" & placementData$Working_Experience == "No")
workex_not_placed_no <- sum(placementData$Placement_Status == "Not Placed" & placementData$Working_Experience == "No")

workex_data_for_placement_status <- data.frame(
  Status = c(
    "Placed with Working Experience",
    "Not Placed with Working Experience",
    "Placed without Working Experience",
    "Not Placed without Working Experience"
  ),
  Count = c(
    workex_placed_yes,
    workex_not_placed_yes,
    workex_placed_no,
    workex_not_placed_no)
  )

ggplot(workex_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Placement Status") +
  ylab("Count") +
  ggtitle("Students' Placement to Students' Working Experience") +
  scale_fill_manual(values = c("#7B904B", "#58641D", "#273B09", "#002400")) +
  # Rotate x-axis label, align to the right
  theme(axis.text.x = element_text(angle = -45, hjust = 0))
```

The code above demonstrates on how to create a bar chart of **students' placement to student's working experience** using GGPlot2 library. The **theme(axis.text.x = element_text())** function here is to make the x-axis label to rotate 45 degree and align to the right so that the label will not overlap each other and improve the readability.

The output of the code is as below:



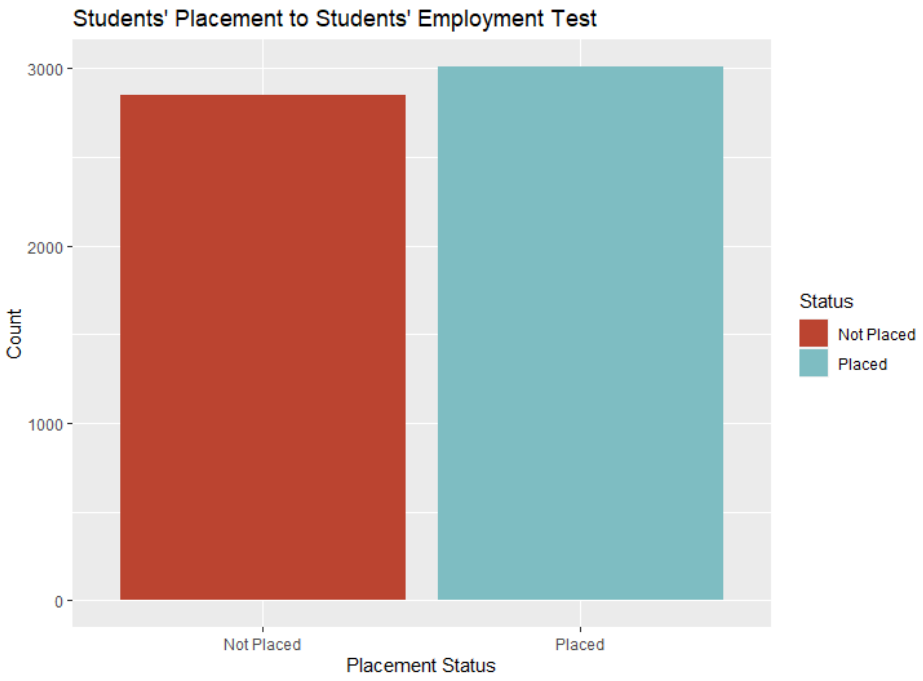

From this bar chart, it is clear that students' working experience doesn't affect the student's placement in the organization because it shows that students either have work experience or no work experience also have a high chance to have a placement in the organization. Hypothetically, the organization might be a grooming organization which is suitable for fresh graduate to apply for their first job.

## 2.6 Analysis 1.6 – Employment Test

```
# - Employment test (If higher than 80 marks)
etest_p_placed <- sum(
  placementData$Placement_Status == "Placed" &
  placementData$Employment_Test >= 80
)
etest_p_not_placed <- sum(
  placementData$Placement_Status == "Not Placed" &
  placementData$Employment_Test >= 80
)

etest_data_for_placement_status <- data.frame(
  Status = c("Placed", "Not Placed"),
  Count = c(etest_p_placed, etest_p_not_placed)
)

ggplot(etest_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Placement Status") +
  ylab("Count") +
  ggtitle("Students' Placement to Students' Employment Test") +
  scale_fill_manual(values = c("#BB4430", "#7EBDC2"))
```

The code above demonstrates on how to create a bar chart of **students' placement to student's employment test** using GGPlot2 library.

The output of the code is as below:




From this bar chart, it is clear that students with employment test higher than 70 marks are preferable and has a higher chance to be placed in the organization.

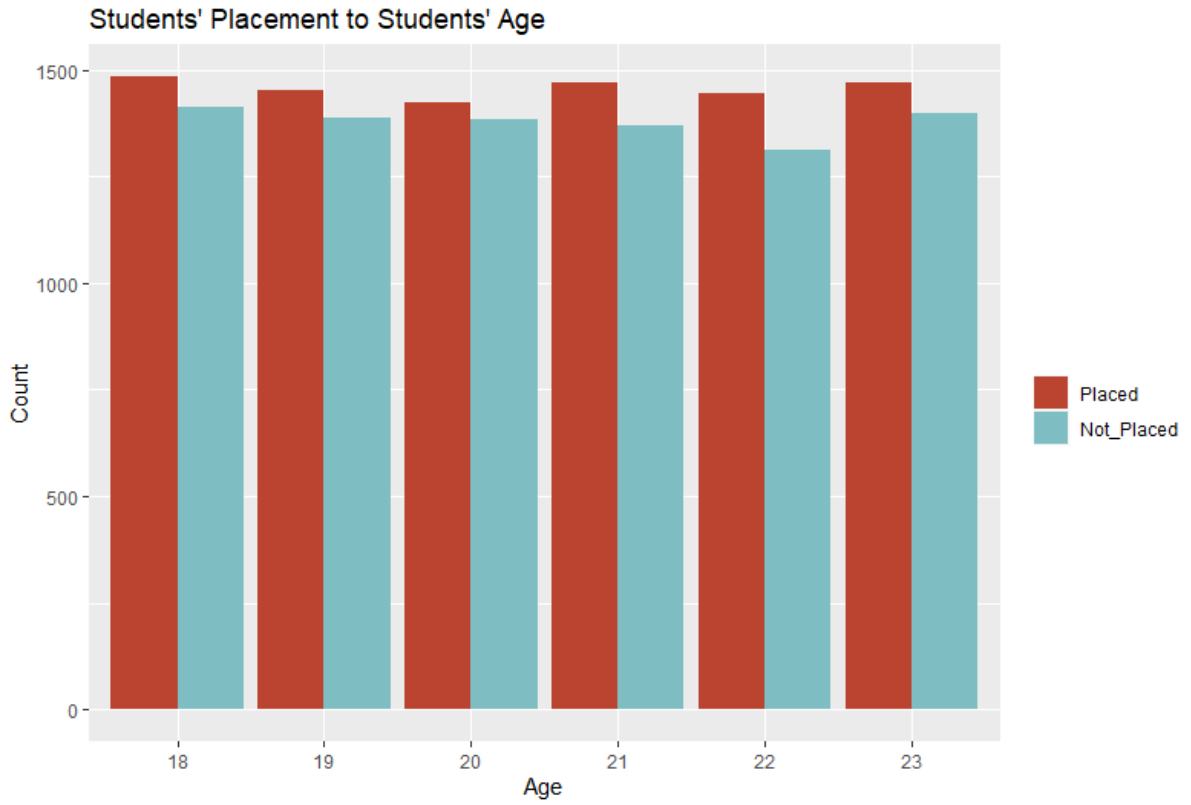
## 2.7 Analysis 1.7 – Age

```
# - Age
ages <- table(placementData$Age)
# Create empty vector
age_placed <- vector("numeric", length(ages))
age_not_placed <- vector("numeric", length(ages))
# Extract the age as vectors and numeric
age_vec <- as.vector(names(ages))
age_num <- as.numeric(names(ages))

for(i in 1:length(age_num)) {
  age <- age_num[i]

  count_age_placed <- sum(placementData$Age == age & placementData$Placement_Status == "Placed")
  count_age_not_placed <- sum(placementData$Age == age & placementData$Placement_Status == "Not Placed")

  age_placed[i] <- count_age_placed
  age_not_placed[i] <-count_age_not_placed
}

age_data_for_placement_status <- data.frame(
  Age_Group = age_vec,
  Placed = age_placed,
  Not_Placed = age_not_placed
)

age_data_for_placement_status_long <- melt(
  age_data_for_placement_status,
  id.vars = "Age_Group",
  variable.name = "Placement_Status",
  value.name = "Count"
  )

ggplot(
  age_data_for_placement_status_long,
  aes(x = factor(Age_Group), y = Count, fill = Placement_Status)) +
  geom_bar(position = "dodge", stat = "identity") +
  labs(x = "Age", y = "Count", fill = "") +
  ggtitle("Students' Placement to Students' Age") +
  scale_fill_manual(values = c("#BB4430", "#7EBDC2"))
```

The code above demonstrates on how to create a bar chart of **students' placement to student's age** using GGPlot2 library. The **melt()** function from **reshape2** library is used to convert a data frame **from wide format to a long format to be plotted and analysed easier.**

The functions here accepts four arguments or parameter, the data frame **(age_data_for_placement_status)**, the specified column in the data frame kept as identifier variables, **(id.vars = "Age_Group")**, in this case is **Age_Group**, the name of the column that will be created in the output data frame to store the **column names** from the original data frame, **(variable.name = "Placement_Status")**, in this case is **"Placement_Status",** and the name of the column that will be create in the output data frame to store the **values** from the original data frame, **(value.name = "Count"),** in this case is **"Count".**

The output of the code is as below:



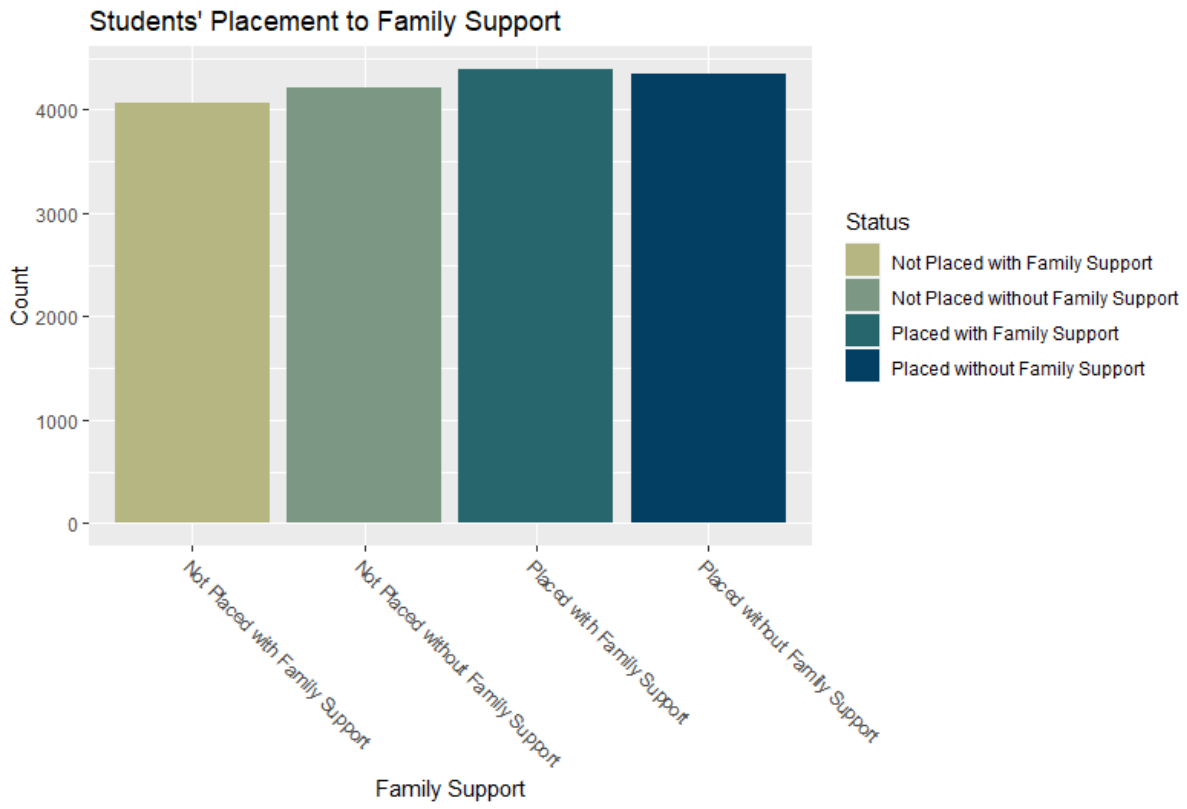

From this bar chart, it is clear that students' age doesn't affect the student's placement in the organization because it shows that regardless the students' age, all age group available in the imported database also have a high chance to have a placement in the organization.

## 2.8 Analysis 1.8 – Family Support

```
# - Family Support
famsup_data_for_placement_status <- placementData %>%
  # Group data frame by status and family support
  group_by(Placement_Status, Family_Support) %>%
  # Summarizes grouped data count
  summarise(Count = n()) %>%
  # Removes grouping info from data frame
  ungroup() %>%
  # Add a new column to the data frame called Status based on values of Status and Family Support
  # If yes, then Placed or Not Placed with Family Support
  # If no, then Placed or Not Placed without Family Support
  mutate(Status = ifelse(
    Family_Support == "yes",
    paste0(Placement_Status, " with Family Support"),
    paste0(Placement_Status, " without Family Support")
  )) %>%
  # Select both columns from data frame
  select(Status, Count)

ggplot(famsup_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Family Support") +
  ylab("Count") +
  ggtitle("Students' Placement to Family Support") +
  scale_fill_manual(values = c("#B5B682", "#7C9885", "#28666E", "#033F63")) +
  # Rotate x-axis label, align to the right
  theme(axis.text.x = element_text(angle = -45, hjust = 0))
```

The code above demonstrates on how to create a bar chart of **students' placement to family support** using GGPlot2 library. The **dplyr** library is used to perform some data manipulation with functions like **group_by(), summarise(), ungroup(), mutate(),** and **select().** The **%>%** expression is to chain multiple different functions together as a single line.

The output of the code is as below:



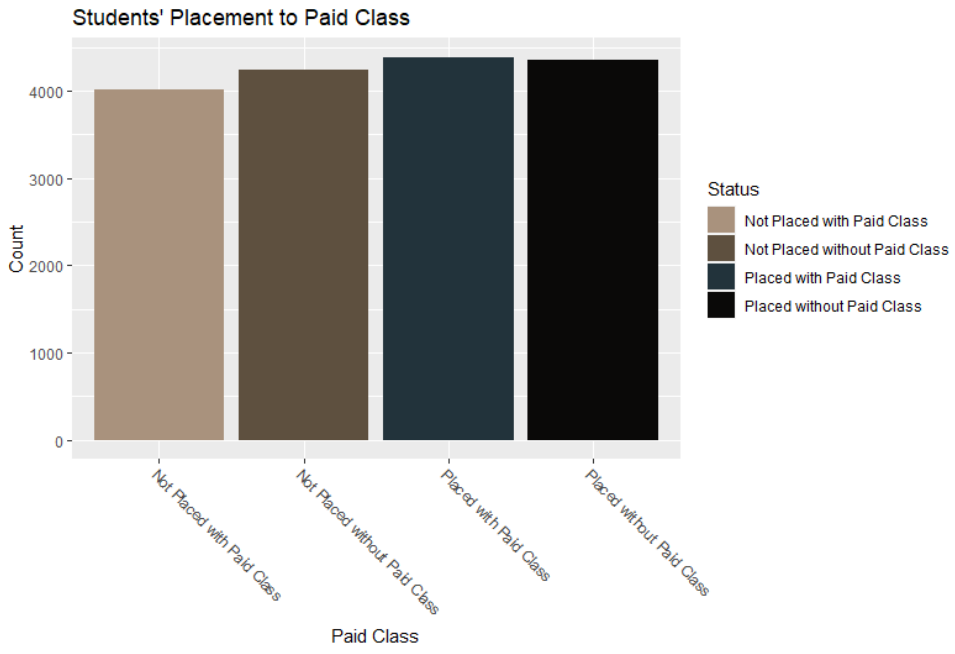

From this bar chart, it is clear that family support doesn't affect the student's placement in the organization because it shows that regardless have or not having family support, the chances of having a placement in the organization is still more or less the same.

## 2.9 Analysis 1.9 – Paid Class

```r
# - Paid class
paid_data_for_placement_status <- placementData %>%
  group_by(Placement_Status, Paid_Classes) %>%
  summarise(Count = n()) %>%
  ungroup() %>%
  mutate(Status = ifelse(
    Paid_Classes == "yes",
    paste0(Placement_Status, " with Paid Class"),
    paste0(Placement_Status, " without Paid Class")
  )) %>%
  select(Status, Count)

ggplot(paid_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Paid Class") +
  ylab("Count") +
  ggtitle("Students' Placement to Paid Class") +
  scale_fill_manual(values = c("#A9927D", "#5E503F", "#22333B", "#0A0908")) +
  # Rotate x-axis label, align to the right
  theme(axis.text.x = element_text(angle = -45, hjust = 0))
```

The code above demonstrates on how to create a bar chart of **students' placement to paid classes** using GGPlot2 library.

The output of the code is as below:



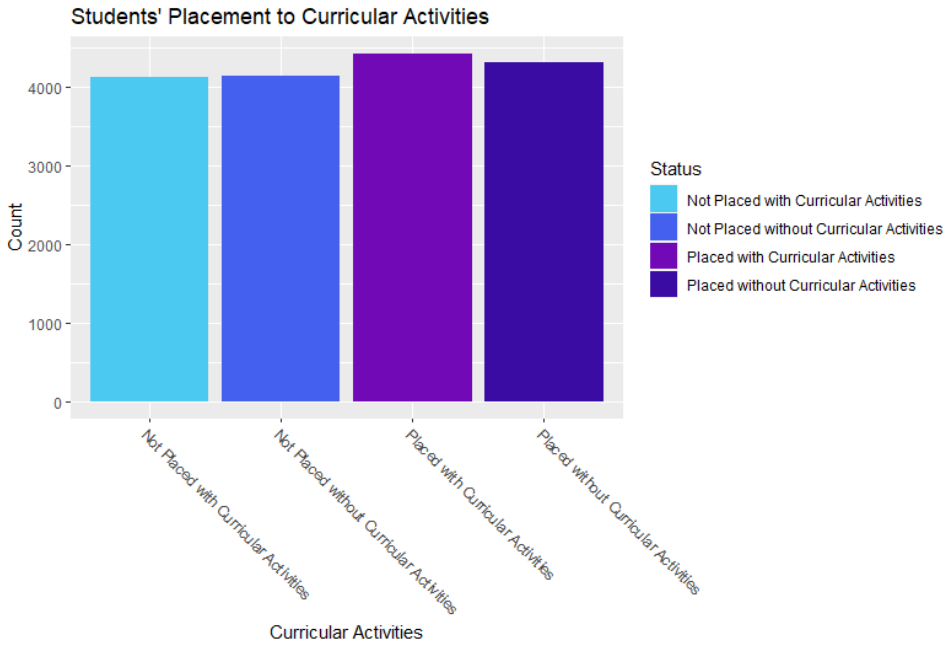

From this bar chart, it is clear that paid classes doesn't affect the student's placement in the organization because it shows that regardless have or not having paid classes, the chances of having a placement in the organization is still more or less the same.

## 2.10 Analysis 1.10 – Curricular Activities

```
# - Curricular activities
activities_data_for_placement_status <- placementData %>%
  group_by(Placement_Status, Curricular_Activities) %>%
  summarise(Count = n()) %>%
  ungroup() %>%
  mutate(Status = ifelse(
    Curricular_Activities == "yes",
    paste0(Placement_Status, " with Curricular Activities"),
    paste0(Placement_Status, " without Curricular Activities")
  )) %>%
  select(Status, Count)

ggplot(activities_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Curricular Activities") +
  ylab("Count") +
  ggtitle("Students' Placement to Curricular Activities") +
  scale_fill_manual(values = c("#4CC9F0", "#4361EE", "#7209B7", "#3A0CA3")) +
  # Rotate x-axis label, align to the right
  theme(axis.text.x = element_text(angle = -45, hjust = 0))
```

The code above demonstrates on how to create a bar chart of **students' placement to curricular activities** using GGPlot2 library.

The output of the code is as below:



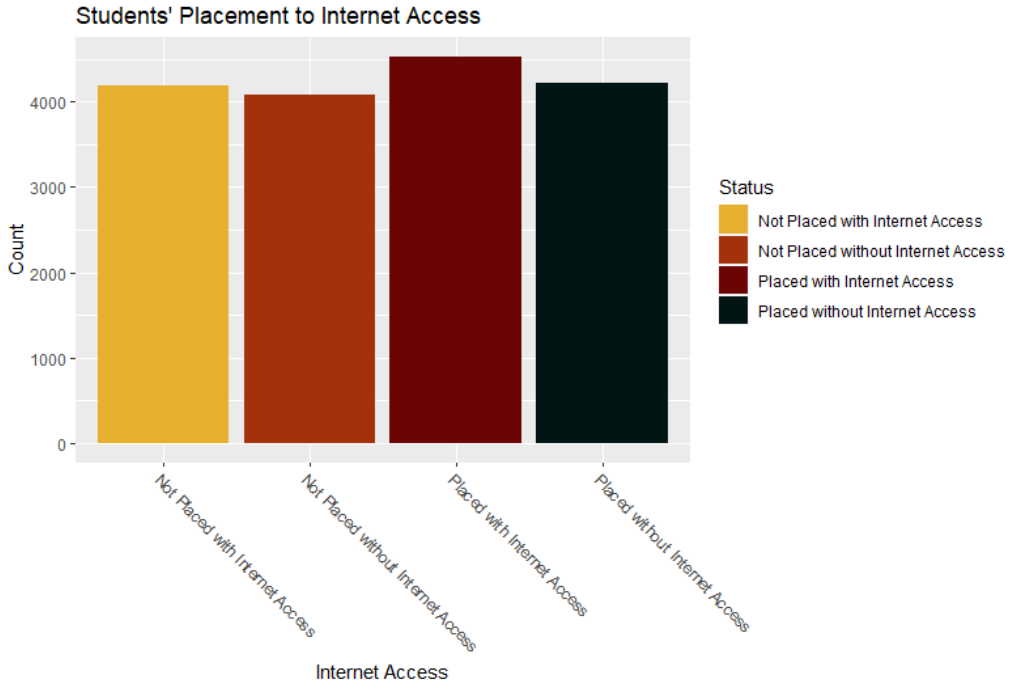

From this bar chart, it is clear that students with curricular activities have a slight preferability to be given a placement in the organization compared to students without curricular activities. Hypothetically, the organization might be a young company where they prefer students with curricular activities because they may be more proactive, participative, and engaging.

## 2.11 Analysis 1.11 – Internet Access

```
# - Internet Access
internet_data_for_placement_status <- placementData %>%
  group_by(Placement_Status, Internet_Usage) %>%
  summarise(Count = n()) %>%
  ungroup() %>%
  mutate(Status = ifelse(
    Internet_Usage == "yes",
    paste0(Placement_Status, " with Internet Access"),
    paste0(Placement_Status, " without Internet Access")
  )) %>%
  select(Status, Count)

ggplot(internet_data_for_placement_status, aes(x = Status, y = Count, fill = Status)) +
  # Height of bars based on data values
  # Separate each bars for better reading
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Internet Access") +
  ylab("Count") +
  ggtitle("Students' Placement to Internet Access") +
  scale_fill_manual(values = c("#E6AF2E", "#A3320B", "#6B0504", "#001514")) +
  # Rotate x-axis label, align to the right
  theme(axis.text.x = element_text(angle = -45, hjust = 0))
```

The code above demonstrates on how to create a bar chart of **students' placement to internet access** using GGPlot2 library.

The output of the code is as below:




From this bar chart, it is clear that students with internet access are much more preferable to the organization because they can access more information and knowledge from the Internet to increase their productivity and work efficiency and can also work remotely.

## 2.12 Analysis Conclusion

Based on all bar chart analysis, there are several factors that are governing students' placement within the organization like **secondary and higher secondary school grade, degree and master grade, employment test**. This is more likely to be logical because most companies will preferably hire employees who excel in their academics and has a basic understanding of what they are going to be doing and what they will be doing within the organization.

Besides that, curricular activities and internet access are another two factors that have influences to students' placement. The organization might be a young company where they prefer their employees to be more proactive, participative, and engaging. Moreover, having internet access means that they can access more information and knowledge from the Internet to increase their productivity and work efficiency and can also work remotely.

Other factors like working experience, age, family support, and paid classes doesn't affect students to get a placement within the organization.

# 3.0 Question 2 – What will affect students to get grades higher than 70 marks in secondary school and higher secondary school?

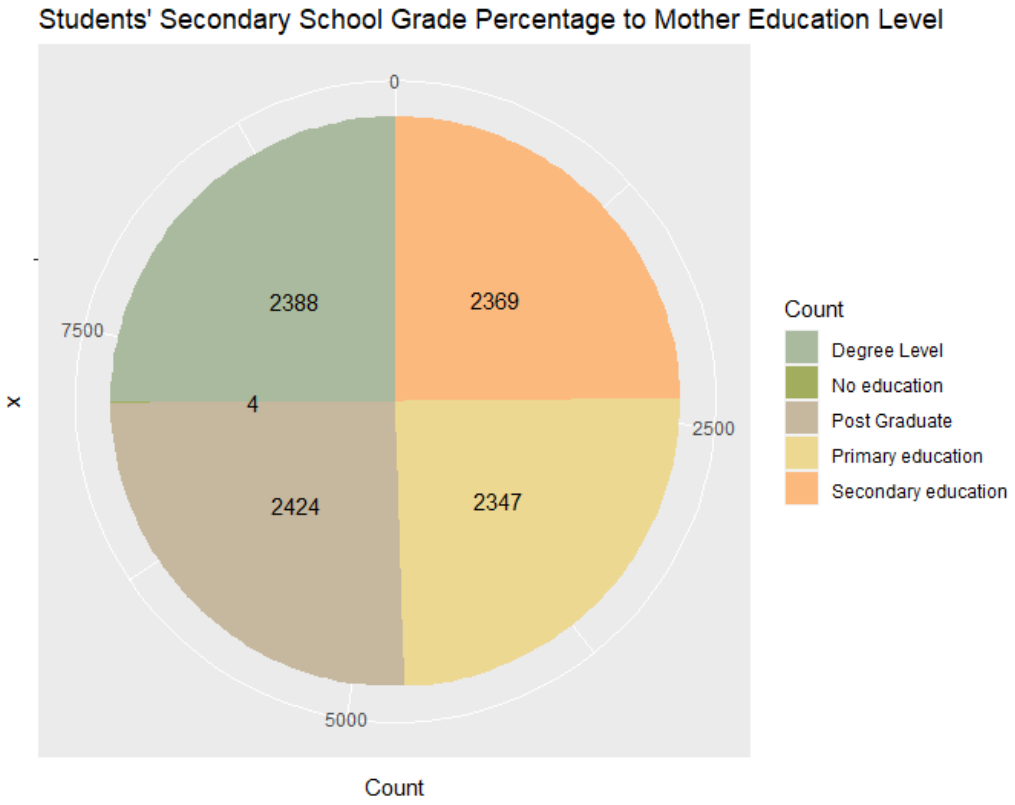
Multiple factors from the imported dataset are used to investigate what will affect students' secondary and higher secondary school grade by creating **pie chart** as data visualization.

## 3.1 Analysis 2.1 – Mother Education

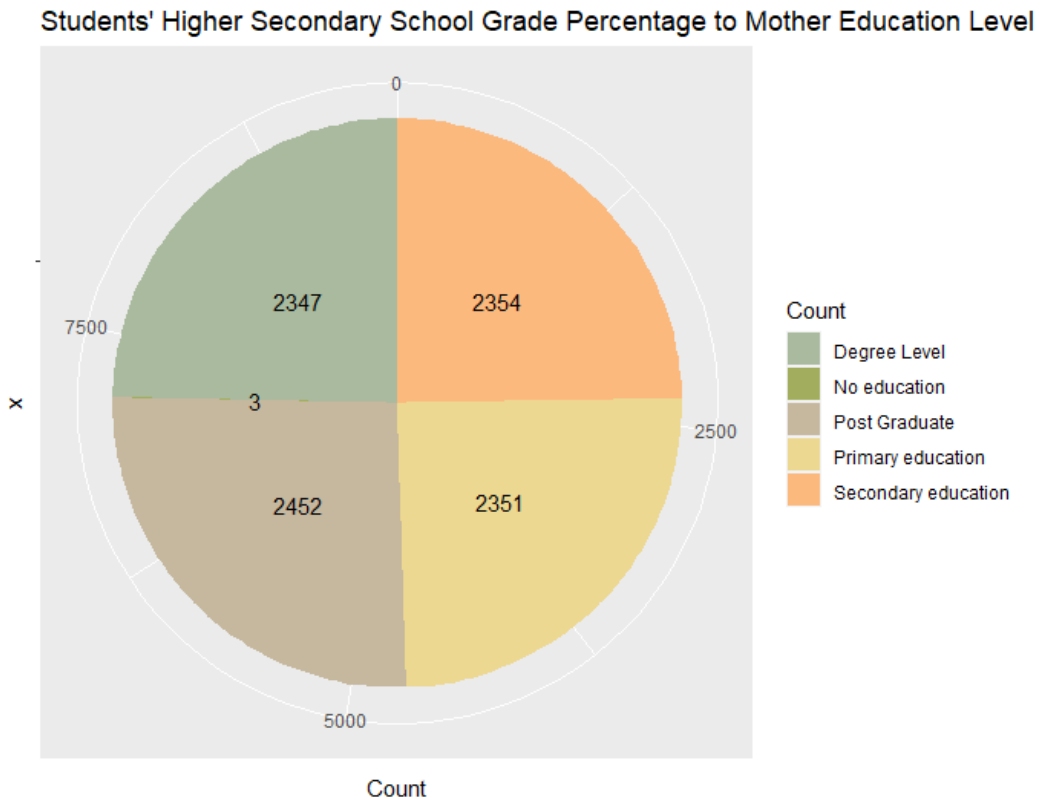
```
# - Mother education
momEducation <- table(placementData$Medu)

momEduLevel <- as.vector(names(momEducation))
# Rename table entry with education level
names(momEduLevel) <- educationLevel
# Create empty vector
mom_pass70ssc <- vector("numeric", length(educationLevel))
mom_pass70hsc <- vector("numeric", length(educationLevel))

for(i in 1:length(educationLevel)) {
  mom_count70ssc <- sum(placementData$Medu == i-1 & placementData$ssc_p >= 70)
  mom_count70hsc <- sum(placementData$Medu == i-1 & placementData$hsc_p >= 70)

  mom_pass70ssc[i] <- mom_count70ssc
  mom_pass70hsc[i] <- mom_count70hsc
}

mom_pass70ssc_data <- data.frame(
  Mother_Education <- educationLevel,
  Count <- mom_pass70ssc
)

ggplot(mom_pass70ssc_data, aes(x = "", y = Count, fill = Mother_Education)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Mother Education Level") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#AABA9E", "#A2AE5E", "#C6B89E", "#EDD892", "#FCB97D"))

mom_pass70hsc_data <- data.frame(
  Mother_Education <- educationLevel,
  Count <- mom_pass70hsc
)

ggplot(mom_pass70hsc_data, aes(x = "", y = Count, fill = Mother_Education)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Mother Education Level") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#AABA9E", "#A2AE5E", "#C6B89E", "#EDD892", "#FCB97D"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Higher Secondary School Grade Percentage to Mother's Education** using the GGPlot2 library. The **geom_col(width = 1)** function is to create columns within the chart and the width is set to 1 to create a polar chart. **coord_polar()** function is to create the polar coordinate system where y argument specifies that the chart is polar in the y-direction and **"start"** argument specifies the starting angle of the chart. **geom_text()** function is to add label into the chart and the position of the label can be set with **position_stack()** function.

The output of the code is as below:




From this pie chart, it is clear that students' mother with no education can be omitted as it is insufficient in terms of data. While students' mother with primary, secondary, degree, and post graduate education have sufficient, yet it does not affect the students to achieve marks higher than 70 in their secondary school.



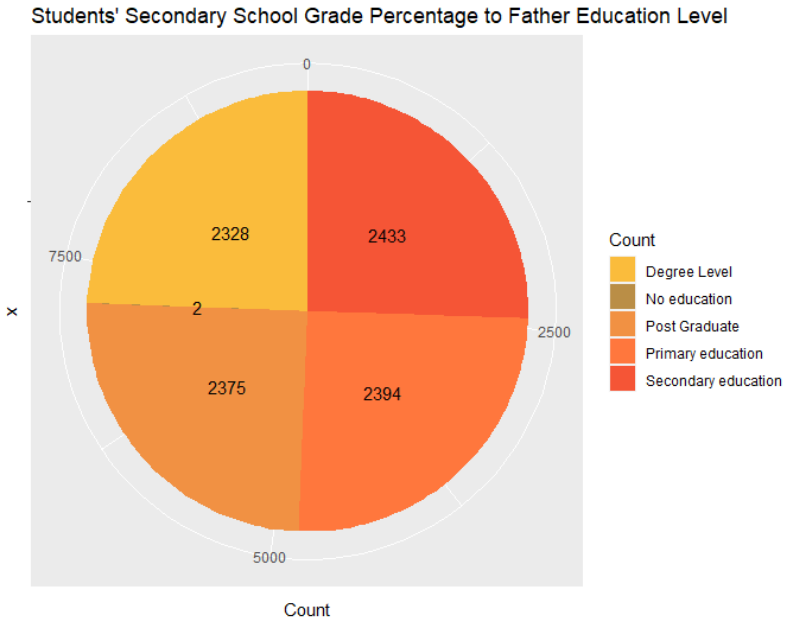

From this pie chart, it is clear that students' mother with no education can be omitted as it is insufficient in terms of data. While students' mother with primary, secondary, degree, and post graduate education have sufficient, yet it does not affect the students to achieve marks higher than 70 in their higher secondary school.

## 3.2 Analysis 2.2 – Father Education

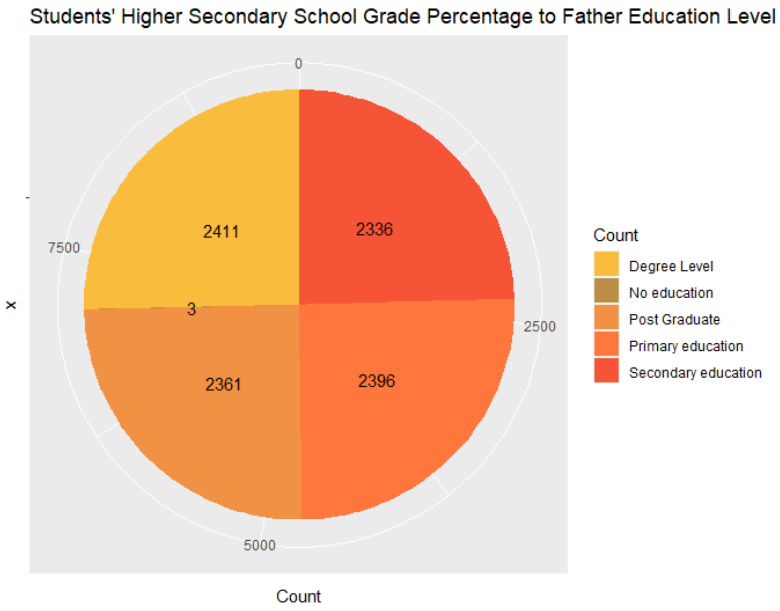
```
# - Father education
dadEducation <- table(placementData$Fedu)

dadEduLevel <- as.vector(names(dadEducation))
# Rename table entry with education level
names(dadEduLevel) <- educationLevel
# Create empty vector
dad_pass70ssc <- vector("numeric", length(educationLevel))
dad_pass70hsc <- vector("numeric", length(educationLevel))

for(i in 1:length(educationLevel)) {
  dad_count70ssc <- sum(placementData$Fedu == i-1 & placementData$ssc_p >= 70)
  dad_count70hsc <- sum(placementData$Fedu == i-1 & placementData$hsc_p >= 70)

  dad_pass70ssc[i] <- dad_count70ssc
  dad_pass70hsc[i] <- dad_count70hsc
}

dad_pass70ssc_data <- data.frame(
  Father_Education <- educationLevel,
  Count <- dad_pass70ssc
)

ggplot(dad_pass70ssc_data, aes(x = "", y = Count, fill = Father_Education)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Father Education Level") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#FABC3C", "#BA8E46", "#F19143", "#FF773D", "#F55536"))

dad_pass70hsc_data <- data.frame(
  Father_Education <- educationLevel,
  Count <- dad_pass70hsc
)

ggplot(dad_pass70hsc_data, aes(x = "", y = Count, fill = Father_Education)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Father Education Level") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#FABC3C", "#BA8E46", "#F19143", "#FF773D", "#F55536"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Higher Secondary School Grade Percentage to Father's Education** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students' father with no education can be omitted as it is insufficient in terms of data. While students' father with primary, secondary, degree, and post graduate education have sufficient, yet it does not affect the students to achieve marks higher than 70 in their secondary school.



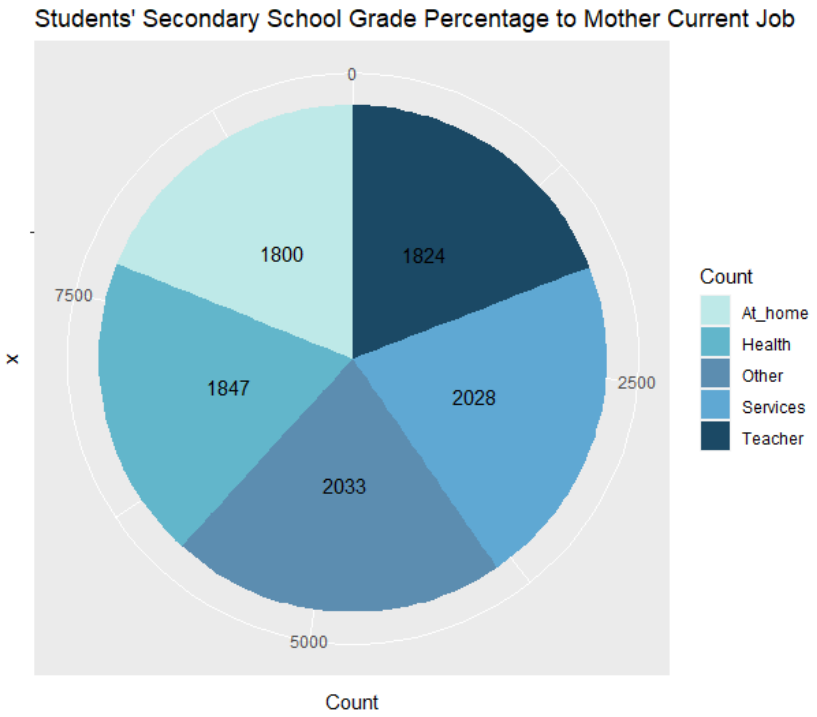

From this pie chart, it is clear that students' father with no education can be omitted as it is insufficient in terms of data. While students' father with primary, secondary, degree, and post graduate education have sufficient, yet it does not affect the students to achieve marks higher than 70 in their higher secondary school.

## 3.3 Analysis 2.3 – Mother Current Job

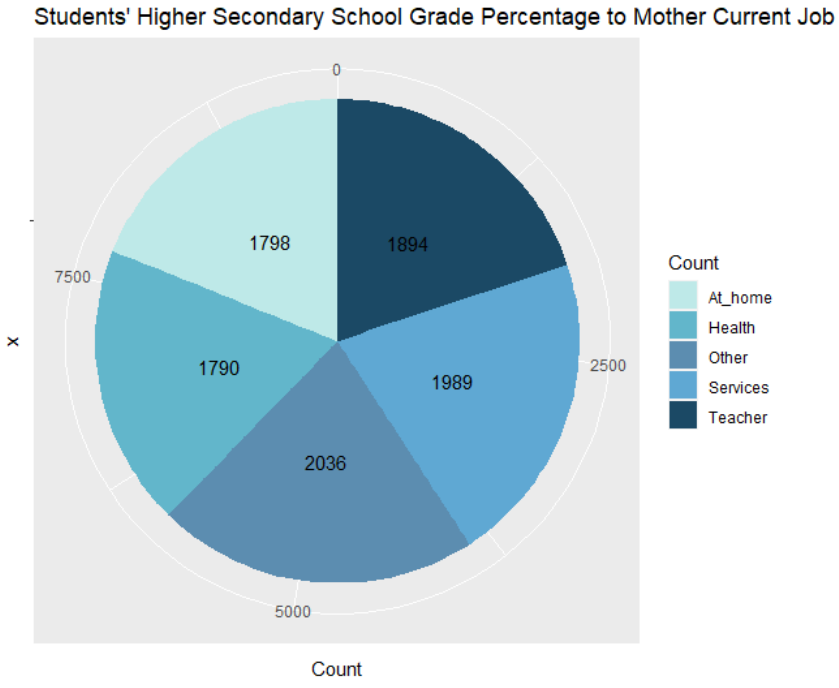
```r
# - Mother current job
momJob <- table(placementData$Mjob)

momJobName <- as.vector(names(momJob))

# Create empty vector
momjob_pass70ssc <- vector("numeric", length(momJobName))
momjob_pass70hsc <- vector("numeric", length(momJobName))

for(i in 1:length(momJobName)) {
  momjob_count70ssc <- sum(placementData$Mjob == momJobName[i] & placementData$ssc_p >= 70)
  momjob_count70hsc <- sum(placementData$Mjob == momJobName[i] & placementData$hsc_p >= 70)

  momjob_pass70ssc[i] <- momjob_count70ssc
  momjob_pass70hsc[i] <- momjob_count70hsc
}

momjob_pass70ssc_data <- data.frame(
  Mother_Current_Job <- str_to_title(momJobName),
  Count <- momjob_pass70ssc
)

ggplot(momjob_pass70ssc_data, aes(x = "", y = Count, fill = Mother_Current_Job)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Mother Current Job") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#BEE9E8", "#62B6CB", "#5C8DB0", "#5FA8D3", "#1B4965"))

momjob_pass70hsc_data <- data.frame(
  Mother_Current_Job <- str_to_title(momJobName),
  Count <- momjob_pass70hsc
)

ggplot(momjob_pass70hsc_data, aes(x = "", y = Count, fill = Mother_Current_Job)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Mother Current Job") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#BEE9E8", "#62B6CB", "#5C8DB0", "#5FA8D3", "#1B4965"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Higher Secondary School Grade Percentage to Mother's Current Job** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students' mother who is currently working in either services or other industries is more likely to affect students to achieve marks higher than 70 in their secondary school.



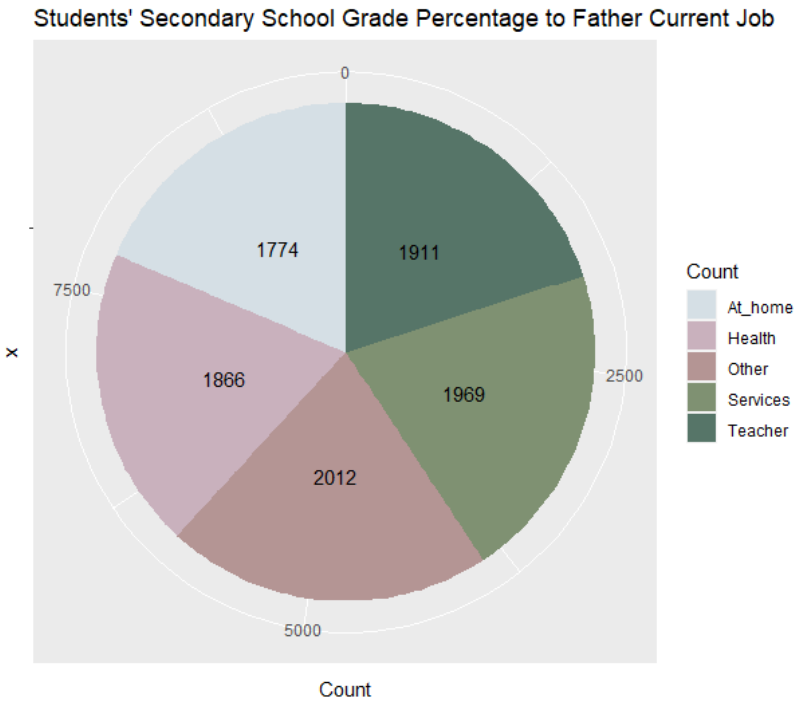

From this pie chart, it is clear that students' mother's current job does not affect students to achieve marks higher than 70 in their higher secondary school.

## 3.4 Analysis 2.4 – Father Current Job

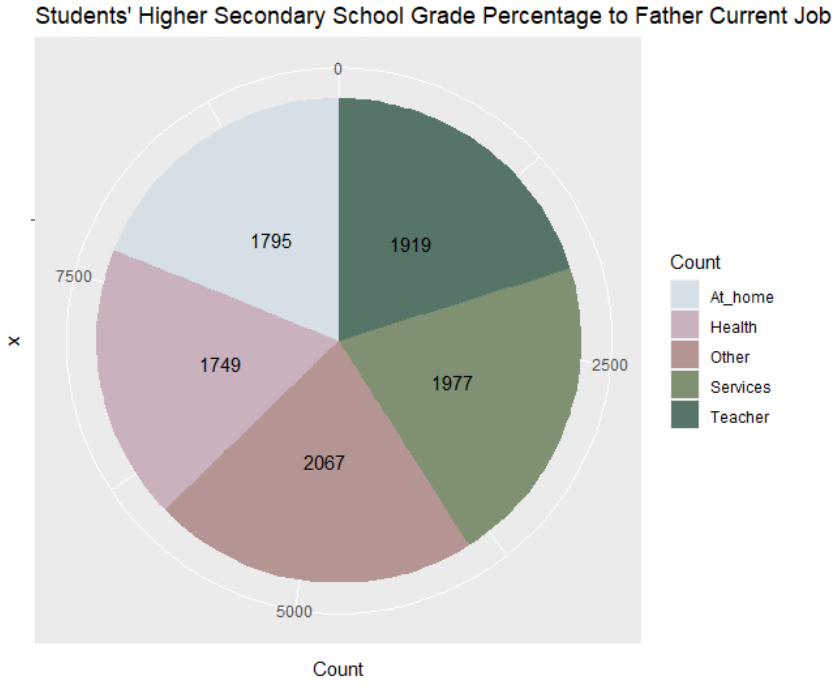
```
# - Father current job
dadJob <- table(placementData$Fjob)

dadJobName <- as.vector(names(dadJob))

# Create empty vector
dadjob_pass70ssc <- vector("numeric", length(dadJobName))
dadjob_pass70hsc <- vector("numeric", length(dadJobName))

for(i in 1:length(dadJobName)) {
  dadjob_count70ssc <- sum(placementData$Fjob == dadJobName[i] & placementData$ssc_p >= 70)
  dadjob_count70hsc <- sum(placementData$Fjob == dadJobName[i] & placementData$hsc_p >= 70)

  dadjob_pass70ssc[i] <- dadjob_count70ssc
  dadjob_pass70hsc[i] <- dadjob_count70hsc
}

dadjob_pass70ssc_data <- data.frame(
  Father_Current_Job <- str_to_title(dadJobName),
  Count <- dadjob_pass70ssc
)

ggplot(dadjob_pass70ssc_data, aes(x = "", y = Count, fill = Father_Current_Job)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Father Current Job") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#D5DFE5", "#C9B1BD", "#B49594", "#7F9172", "#567568"))

dadjob_pass70hsc_data <- data.frame(
  Father_Current_Job <- str_to_title(dadJobName),
  Count <- dadjob_pass70hsc
)

ggplot(momjob_pass70hsc_data, aes(x = "", y = Count, fill = Father_Current_Job)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Father Current Job") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#D5DFE5", "#C9B1BD", "#B49594", "#7F9172", "#567568"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Father's Current Job** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students' father who is currently as a househusband has the least influences towards students to achieve marks higher than 70 in their secondary school.



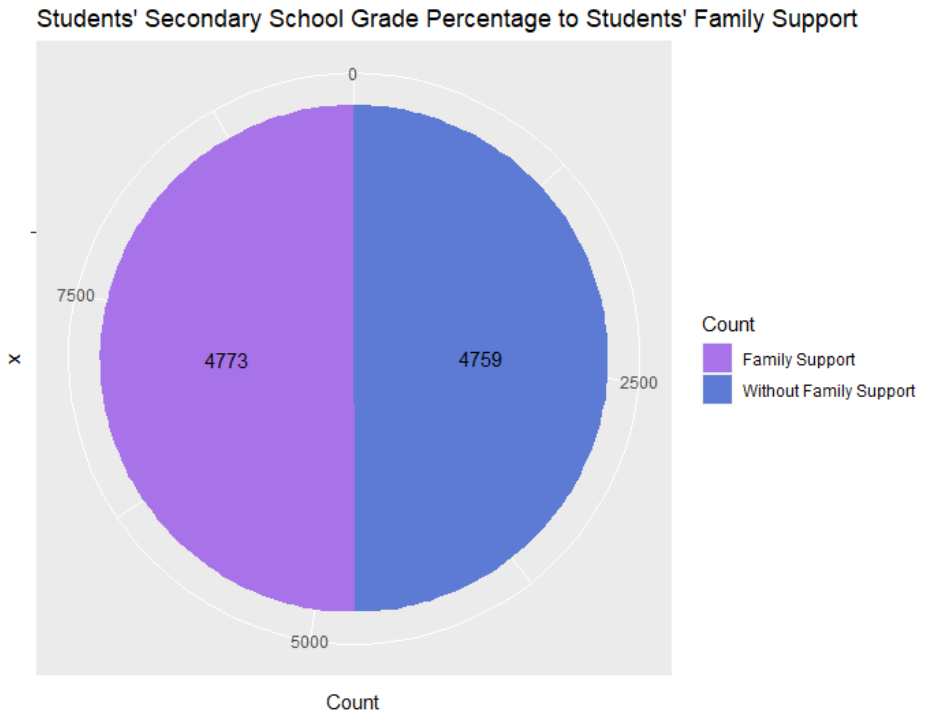

From this pie chart, it is clear that students' father who is currently working either in health or other industries has the least influences towards students to achieve marks higher than 70 in their higher secondary school.

## 3.5 Analysis 2.5 – Family Support

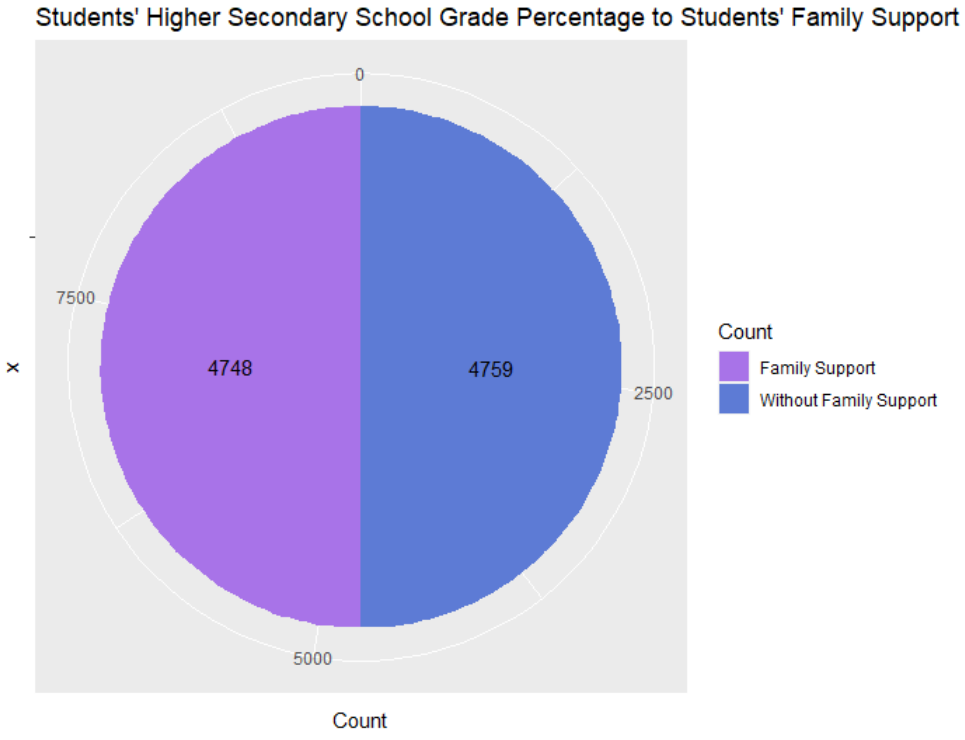
```
# - Family Support
famsup_yes_pass70ssc <- sum(placementData$famsup == "yes" & placementData$ssc_p >= 70)
famsup_no_pass70ssc <- sum(placementData$famsup == "no" & placementData$ssc_p >= 70)

famsup_yes_pass70hsc <- sum(placementData$famsup == "yes" & placementData$hsc_p >= 70)
famsup_no_pass70hsc <- sum(placementData$famsup == "no" & placementData$hsc_p >= 70)

famsup_pass70ssc <- data.frame(
  Status <- c("Family Support", "Without Family Support"),
  Count <- c(famsup_yes_pass70ssc, famsup_no_pass70ssc)
)

ggplot(famsup_pass70ssc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Students' Family Support") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#A873E8", "#5D7BD5"))

famsup_pass70hsc <- data.frame(
  Status <- c("Family Support", "Without Family Support"),
  Count <- c(famsup_yes_pass70hsc, famsup_no_pass70hsc)
)

ggplot(famsup_pass70hsc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Students' Family Support") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#A873E8", "#5D7BD5"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Family Support** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that family support does not affect students to achieve marks higher than 70 in their secondary school.



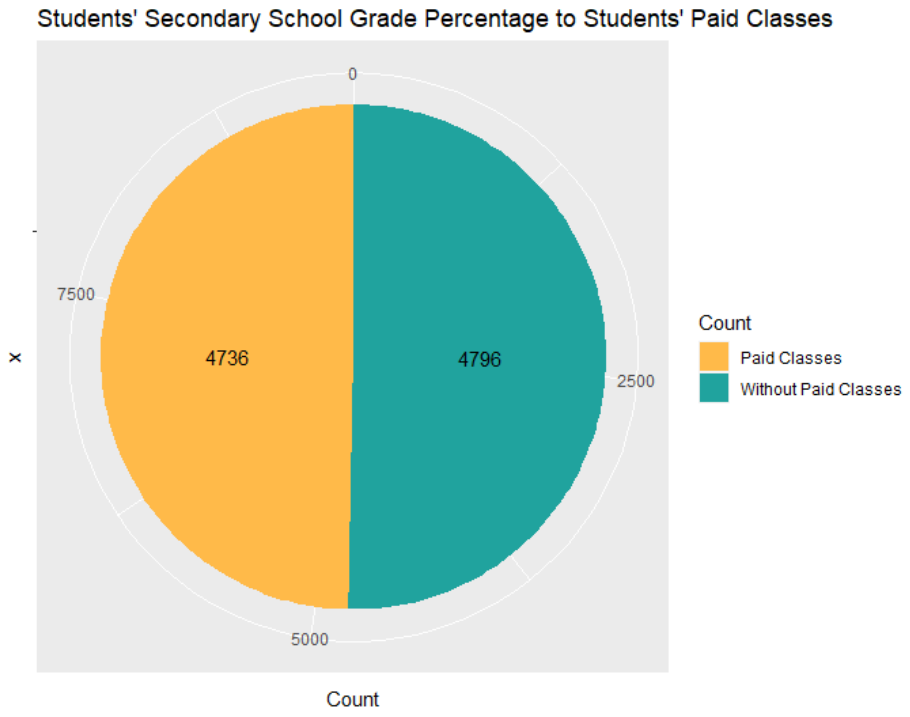

From this pie chart, it is clear that family support does not affect students to achieve marks higher than 70 in their higher secondary school.

## 3.6 Analysis 2.6 – Paid Classes

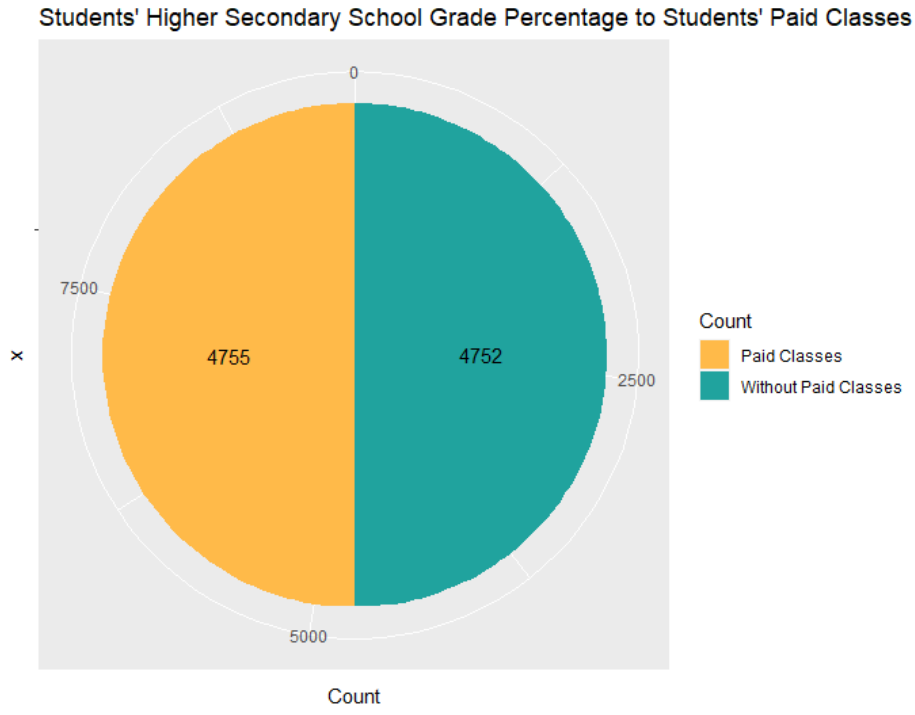
```
# - Paid class
paid_yes_pass70ssc <- sum(placementData$paid == "yes" & placementData$ssc_p >= 70)
paid_no_pass70ssc <- sum(placementData$paid == "no" & placementData$ssc_p >= 70)

paid_yes_pass70hsc <- sum(placementData$paid == "yes" & placementData$hsc_p >= 70)
paid_no_pass70hsc <- sum(placementData$paid == "no" & placementData$hsc_p >= 70)

paid_pass70ssc <- data.frame(
  Status <- c("Paid Classes", "Without Paid Classes"),
  Count <- c(paid_yes_pass70ssc, paid_no_pass70ssc)
)

ggplot(paid_pass70ssc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Students' Paid Classes") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#FFBA49", "#20A39E"))

paid_pass70hsc <- data.frame(
  Status <- c("Paid Classes", "Without Paid Classes"),
  Count <- c(paid_yes_pass70hsc, paid_no_pass70hsc)
)

ggplot(paid_pass70hsc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Students' Paid Classes") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#FFBA49", "#20A39E"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Paid Classes** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students have or not have paid classes does not affect them to achieve marks higher than 70 in their secondary school.



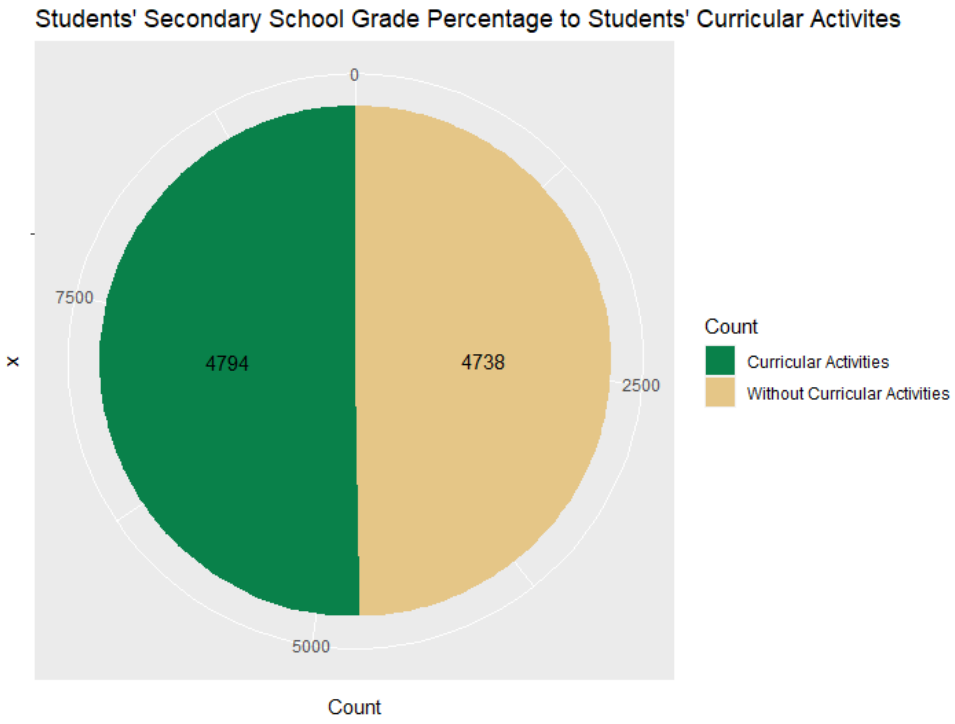

From this pie chart, it is clear that students have or not have paid classes does not affect them to achieve marks higher than 70 in their higher secondary school.

## 3.7 Analysis 2.7 – Curricular Activities

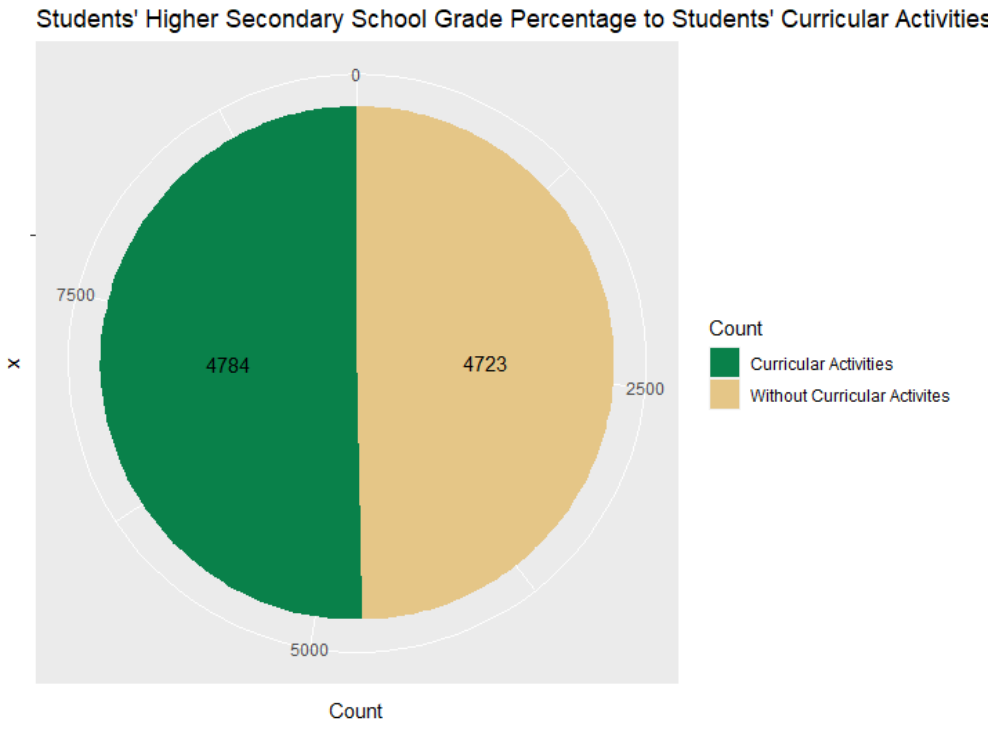
```
# - Curricular activities
activities_yes_pass70ssc <- sum(placementData$activities == "yes" & placementData$ssc_p >= 70)
activities_no_pass70ssc <- sum(placementData$activities == "no" & placementData$ssc_p >= 70)

activities_yes_pass70hsc <- sum(placementData$activities == "yes" & placementData$hsc_p >= 70)
activities_no_pass70hsc <- sum(placementData$activities == "no" & placementData$hsc_p >= 70)

activities_pass70ssc <- data.frame(
  Status <- c("Curricular Activities", "Without Curricular Activities"),
  Count <- c(activities_yes_pass70ssc, activities_no_pass70ssc)
)

ggplot(activities_pass70ssc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Students' Curricular Activites") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#09814A", "#E5C687"))

activities_pass70hsc <- data.frame(
  Status <- c("Curricular Activities", "Without Curricular Activites"),
  Count <- c(activities_yes_pass70hsc, activities_no_pass70hsc)
)

ggplot(activities_pass70hsc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Students' Curricular Activities") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#09814A", "#E5C687"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Curricular Activities** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students have or not have curricular activities does not affect them to achieve marks higher than 70 in their secondary school.



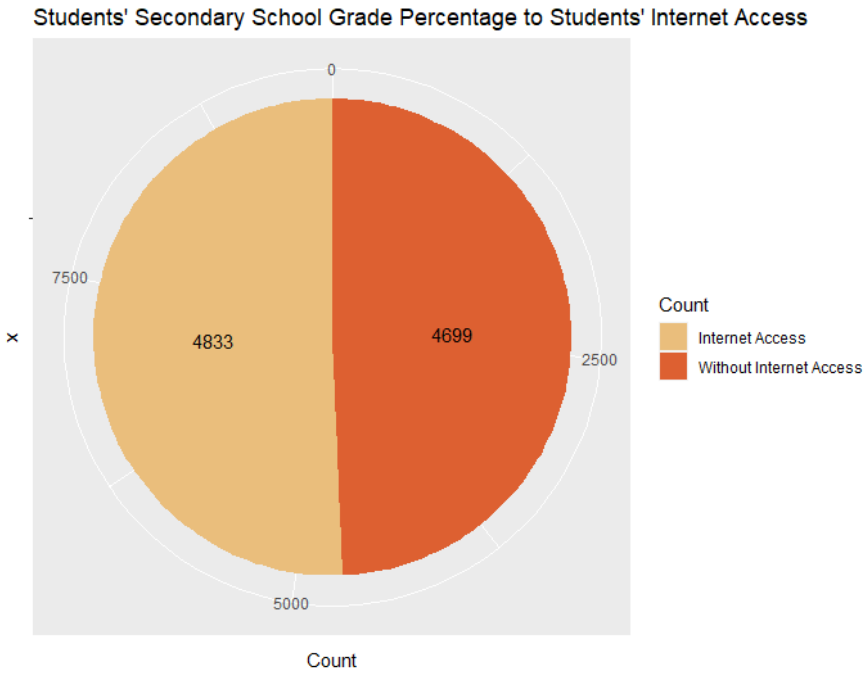

From this pie chart, it is clear that students have or not have curricular activities does not affect them to achieve marks higher than 70 in their higher secondary school.

## 3.8 Analysis 2.8 – Internet Access

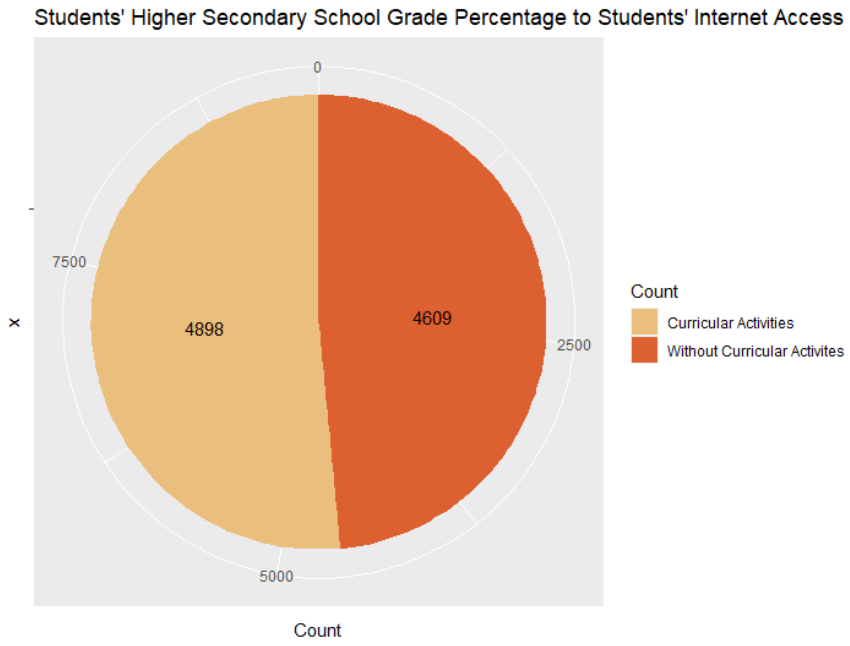
```
# - Internet Access
internet_yes_pass70ssc <- sum(placementData$internet == "yes" & placementData$ssc_p >= 70)
internet_no_pass70ssc <- sum(placementData$internet == "no" & placementData$ssc_p >= 70)

internet_yes_pass70hsc <- sum(placementData$internet == "yes" & placementData$hsc_p >= 70)
internet_no_pass70hsc <- sum(placementData$internet == "no" & placementData$hsc_p >= 70)

internet_pass70ssc <- data.frame(
  Status <- c("Internet Access", "Without Internet Access"),
  Count <- c(internet_yes_pass70ssc, internet_no_pass70ssc)
)

ggplot(internet_pass70ssc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Students' Internet Access") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#EABE7C", "#DD6031"))

internet_pass70hsc <- data.frame(
  Status <- c("Curricular Activities", "Without Curricular Activites"),
  Count <- c(internet_yes_pass70hsc, internet_no_pass70hsc)
)

ggplot(internet_pass70hsc, aes(x = "", y = Count, fill = Status)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Students' Internet Access") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#EABE7C", "#DD6031"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Internet Access** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students who have internet access are more prone to achieve marks higher than 70 compared to those who don't have internet access in their secondary school.



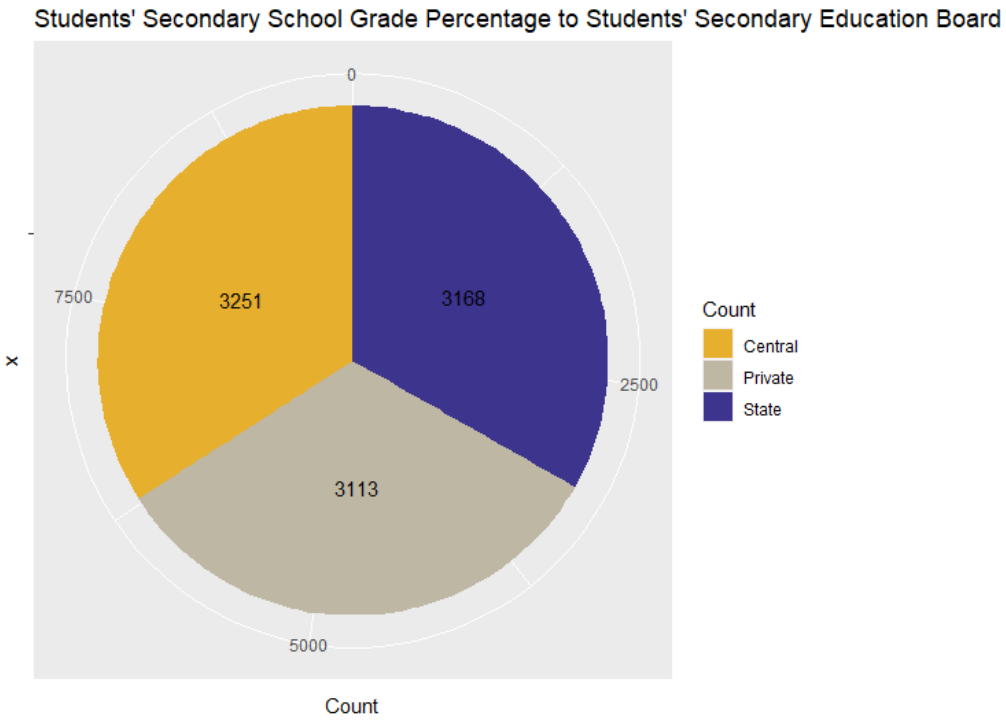

From this pie chart, it is clear that students who have internet access are more prone to achieve marks higher than 70 compared to those who don't have internet access in their higher secondary school.

## 3.9 Analysis 2.9 – Secondary Education Board

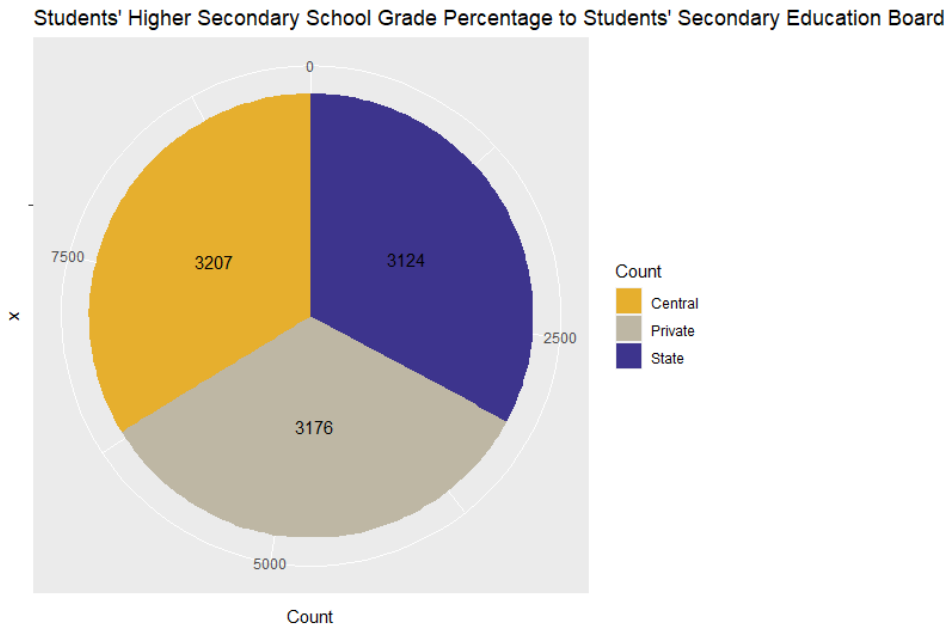
```
# - Secondary education board
secondEduBoard <- table(placementData$ssc_b)

secondEduBoardName <- as.vector(names(secondEduBoard))

# Create empty vector
secondEduBoard_pass70ssc <- vector("numeric", length(secondEduBoardName))
secondEduBoard_pass70hsc <- vector("numeric", length(secondEduBoardName))

for(i in 1:length(secondEduBoardName)) {
  secondEduBoard_count70ssc <- sum(placementData$ssc_b == secondEduBoardName[i] & placementData$ssc_p >= 70)
  secondEduBoard_count70hsc <- sum(placementData$ssc_b == secondEduBoardName[i] & placementData$hsc_p >= 70)

  secondEduBoard_pass70ssc[i] <- secondEduBoard_count70ssc
  secondEduBoard_pass70hsc[i] <- secondEduBoard_count70hsc
}

secondEduBoard_pass70ssc_data <- data.frame(
  Secondary_Education_Board <- secondEduBoardName,
  Count <- secondEduBoard_pass70ssc
)

ggplot(secondEduBoard_pass70ssc_data, aes(x = "", y = Count, fill = Secondary_Education_Board)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Students' Secondary Education Board") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#E6AF2E", "#BEB7A4", "#3D348D"))

secondEduBoard_pass70hsc_data <- data.frame(
  Secondary_Education_Board <- secondEduBoardName,
  Count <- secondEduBoard_pass70hsc
)

ggplot(secondEduBoard_pass70hsc_data, aes(x = "", y = Count, fill = Secondary_Education_Board)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Students' Secondary Education Board") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#E6AF2E", "#BEB7A4", "#3D348D"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Secondary Education Board** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students' secondary education board does not affect students to achieve marks higher than 70 marks in their secondary school.



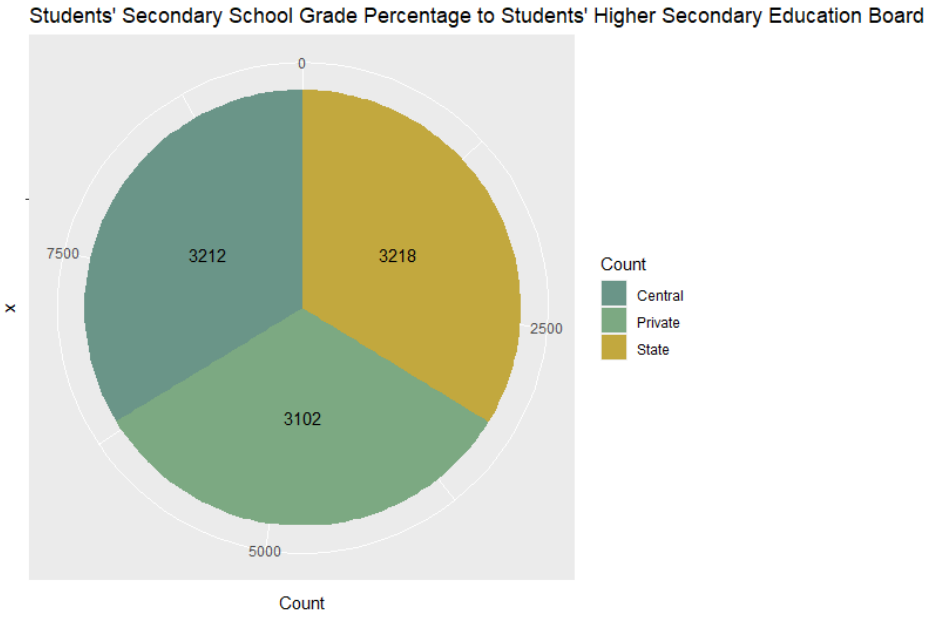

From this pie chart, it is clear that students' secondary education board does not affect students to achieve marks higher than 70 marks in their higher secondary school.

## 3.10 Analysis 2.10 – Higher Secondary Education Board

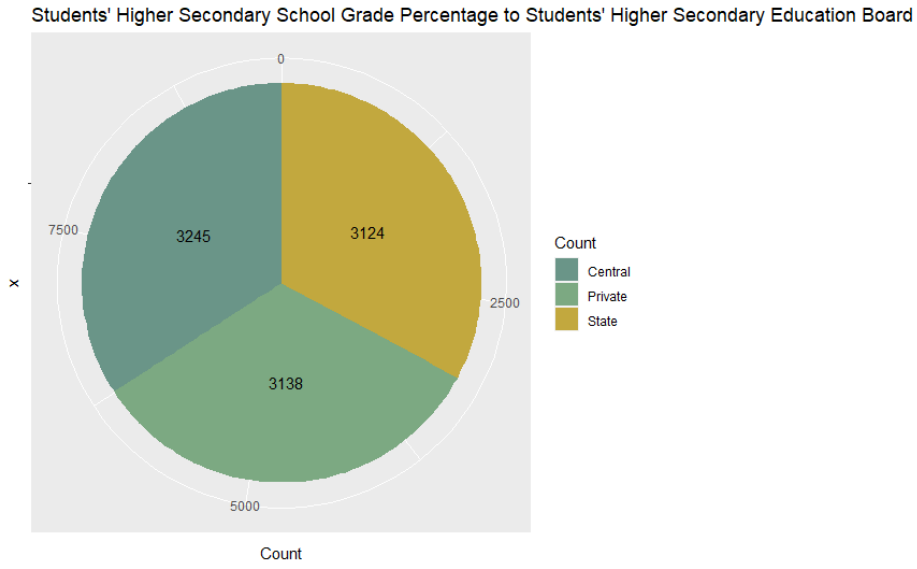
```
# - Higher secondary education board
higherSecondEduBoard <- table(placementData$hsc_b)

higherSecondEduBoardName <- as.vector(names(higherSecondEduBoard))

# Create empty vector
higherSecondEduBoard_pass70ssc <- vector("numeric", length(higherSecondEduBoardName))
higherSecondEduBoard_pass70hsc <- vector("numeric", length(higherSecondEduBoardName))

for(i in 1:length(higherSecondEduBoardName)) {
  higherSecondEduBoard_count70ssc <- sum(placementData$hsc_b == secondEduBoardName[i] & placementData$ssc_p >= 70)
  higherSecondEduBoard_count70hsc <- sum(placementData$hsc_b == secondEduBoardName[i] & placementData$hsc_p >= 70)

  higherSecondEduBoard_pass70ssc[i] <- higherSecondEduBoard_count70ssc
  higherSecondEduBoard_pass70hsc[i] <- higherSecondEduBoard_count70hsc
}

higherSecondEduBoard_pass70ssc_data <- data.frame(
  Higher_Secondary_Education_Board <- higherSecondEduBoardName,
  Count <- higherSecondEduBoard_pass70ssc
)

ggplot(higherSecondEduBoard_pass70ssc_data, aes(x = "", y = Count, fill = Higher_Secondary_Education_Board)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Secondary School Grade Percentage to Students' Higher Secondary Education Board") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#6A9588", "#7CA982", "#C2A83E"))

higherSecondEduBoard_pass70hsc_data <- data.frame(
  Higher_Secondary_Education_Board <- higherSecondEduBoardName,
  Count <- higherSecondEduBoard_pass70hsc
)

ggplot(higherSecondEduBoard_pass70hsc_data, aes(x = "", y = Count, fill = Higher_Secondary_Education_Board)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  labs(fill = "Count", title = "Students' Higher Secondary School Grade Percentage to Students' Higher Secondary Education Board") +
  geom_text(aes(label = Count), position = position_stack(vjust = 0.5)) +
  scale_fill_manual(values = c("#6A9588", "#7CA982", "#C2A83E"))
```

The code above demonstrates on how to create two pie charts of **Students' Secondary School Grade Percentage and Secondary School Grade Percentage to Higher Secondary Education Board** using the GGPlot2 library.

The output of the code is as below:




From this pie chart, it is clear that students who studied in private higher secondary education board school have the least influences towards themselves to achieve marks higher than 70 in their secondary school. Probably this is due to the fact that private education board schools tend to have more advanced teaching methods and harder subjects compared to state and central education board schools.




From this pie chart, it is clear that students who studied in central higher secondary education board school have the most influences towards themselves to achieve marks higher than 70 in their higher secondary school. Probably this is due to the fact that central education board schools have easier subjects compared to state and private education board schools.

## 3.11 Analysis Conclusion

Based on all pie chart analysis, there are several factors that influenced students to achieve 70 marks or above in secondary and higher secondary school.

During the analysis for secondary school grade, factors like **parent's job, internet access and higher secondary school education board** will affect students' grade. Students' mother who currently works in services or other industries tends to encourage students to achieve higher marks and students' father who currently a househusband tends to discourage students to achieve higher marks. This can be concluded as parents' working environment or family background will affect students' secondary school grade. Besides that, internet access will also affect students' secondary school grade as they can gain more knowledge and conduct more researches. However, private higher secondary school education board has the least influences mostly because of private education board schools tend to have more advanced teaching methods and harder subjects.

During the analysis for higher secondary school grade, factors like **father's job, internet access and higher secondary school education board** will affect students' grade. Students' father who currently works in health and other industries tends to discourage students to achieve higher marks. Like secondary school grade, internet access also affects students to score a better mark. On the other hand, central higher secondary school education board will allow students to achieve better grades probably because of easier subjects compared to state and private education board schools.

# 4.0 Question 3 – What will affect students' degree percentage?

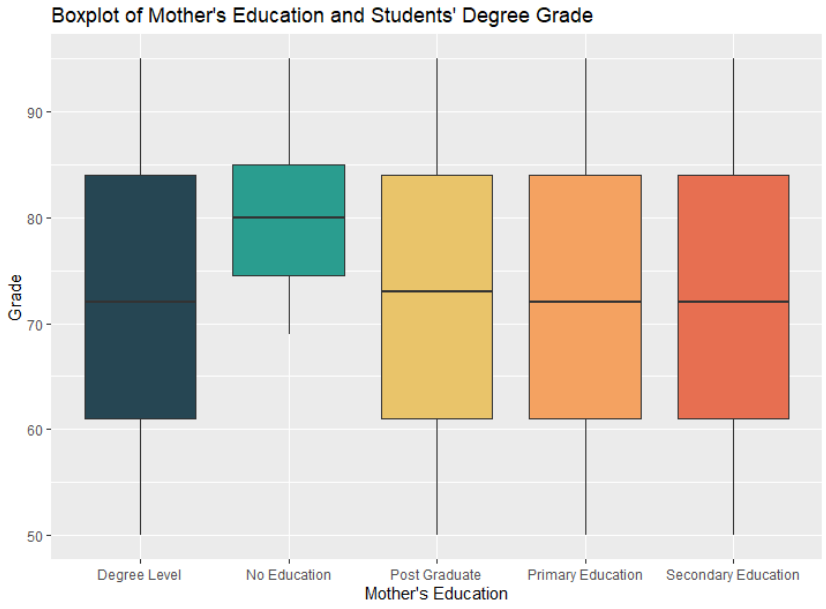
Multiple factors from the imported dataset are used to investigate what will affect students' degree grade by creating **boxplot** as data visualization.

## 4.1 Mother Education

```
# - Mother education
df_mom_edu <- data.frame(
  mom_edu_x = as.vector(placementData$Mother_Education),
  mom_edu_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_mom_edu, aes(x = factor(mom_edu_x), y = mom_edu_y)) +
  geom_boxplot(fill = c("#264653", "#2A9D8F", "#E9C46A", "#F4A261", "#E76F51")) +
  labs(
    x = "Mother's Education",
    y = "Grade",
    title = "Boxplot of Mother's Education and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Mother's Education to Students' Degree Grade** using the GGPlot2 library. The **geom_boxplot()** function is to create a boxplot graph while the **fill()** function inside it is to specify custom colours to each boxes to represent different categories.

The output of the code is as below:



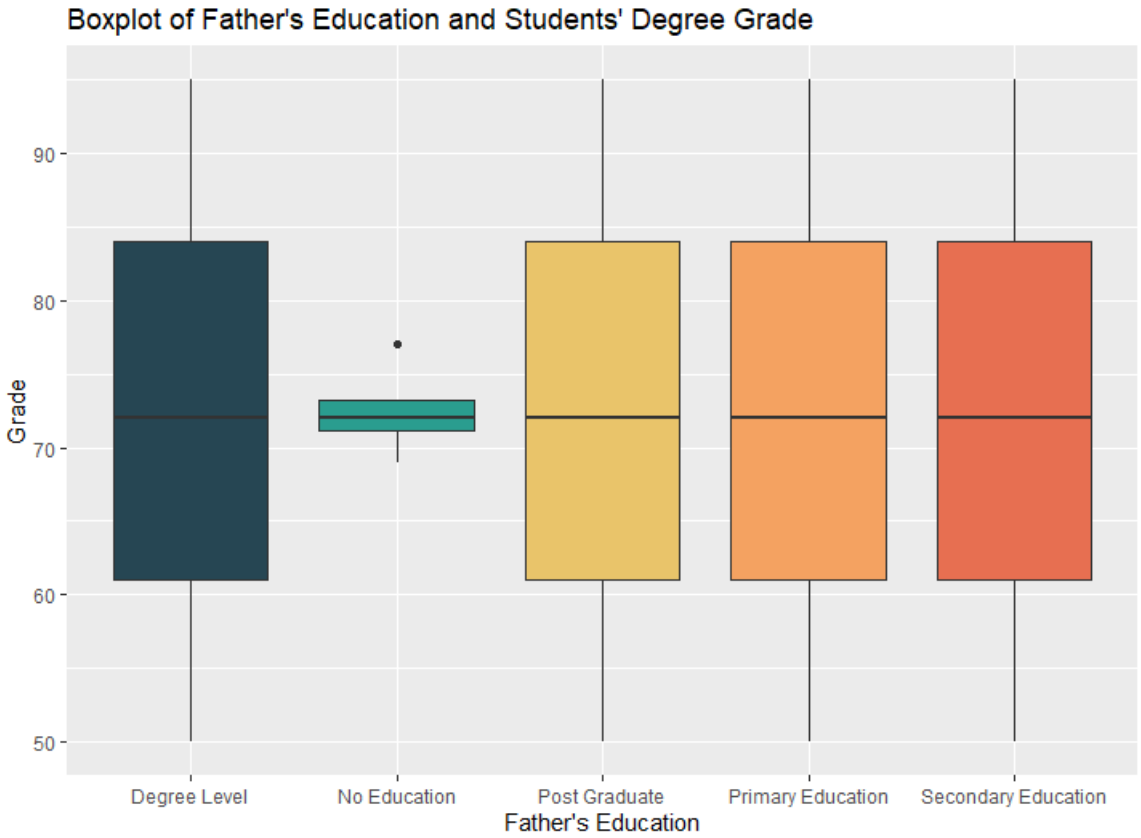

From this boxplot, it is clear that students' mother who does not have any education will affect students to have a higher degree percentage, ranging from around 70 to 95 marks and with a median of 80 marks.

## 4.2 Father Education

```
# - Father education
df_dad_edu <- data.frame(
  dad_edu_x = as.vector(placementData$Father_Education),
  dad_edu_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_dad_edu, aes(x = factor(dad_edu_x), y = dad_edu_y)) +
  geom_boxplot(fill = c("#264653", "#2A9D8F", "#E9C46A", "#F4A261", "#E76F51")) +
  labs(
    x = "Father's Education",
    y = "Grade",
    title = "Boxplot of Father's Education and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Father's Education to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



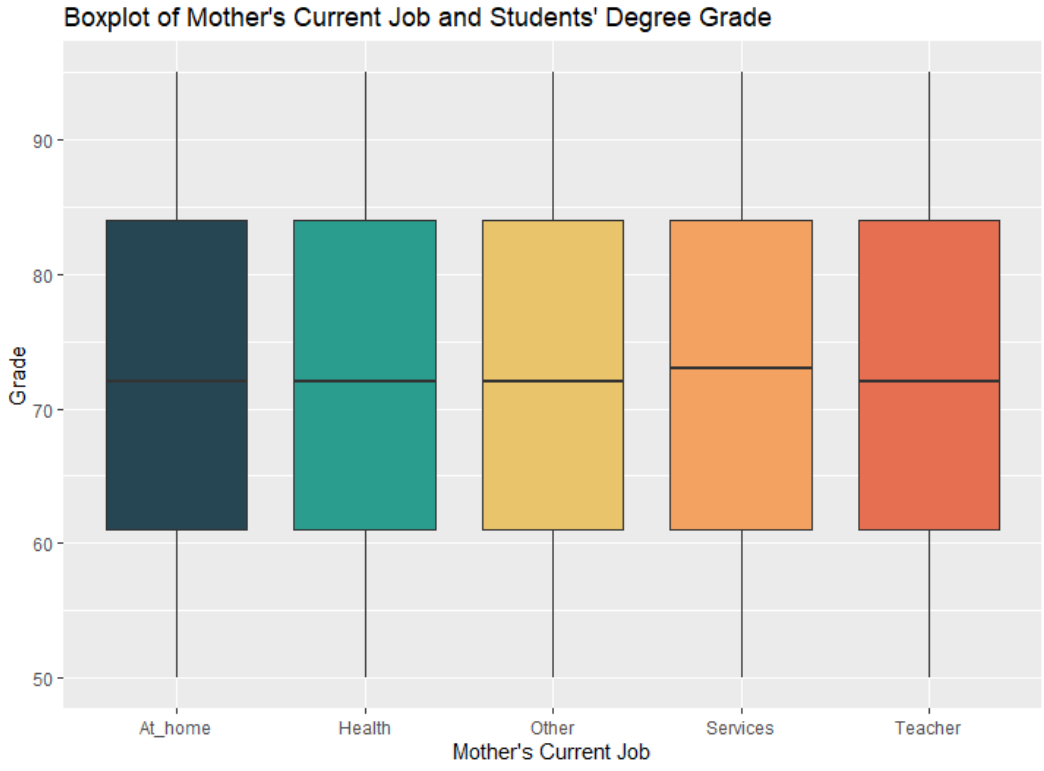

From this boxplot, it is clear that father's education does not affect the students' degree grade because all of the boxes have the same median of around 72.

## 4.3 Mother Current Job

```
# - Mother current job
df_mom_job <- data.frame(
  mom_job_x = as.vector(placementData$Mother_Current_Job),
  mom_job_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_mom_job, aes(x = stringr::str_to_title(factor(mom_job_x)), y = mom_job_y)) +
  geom_boxplot(fill = c("#264653", "#2A9D8F", "#E9C46A", "#F4A261", "#E76F51")) +
  labs(
    x = "Mother's Current Job",
    y = "Grade",
    title = "Boxplot of Mother's Current Job and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Mother's Current Job to Students' Degree Grade** using the GGPlot2 library. The **stringr::str_to_title()** function is to format the string text to a proper case string, for instance, from rstudio to Rstudio.

The output of the code is as below:



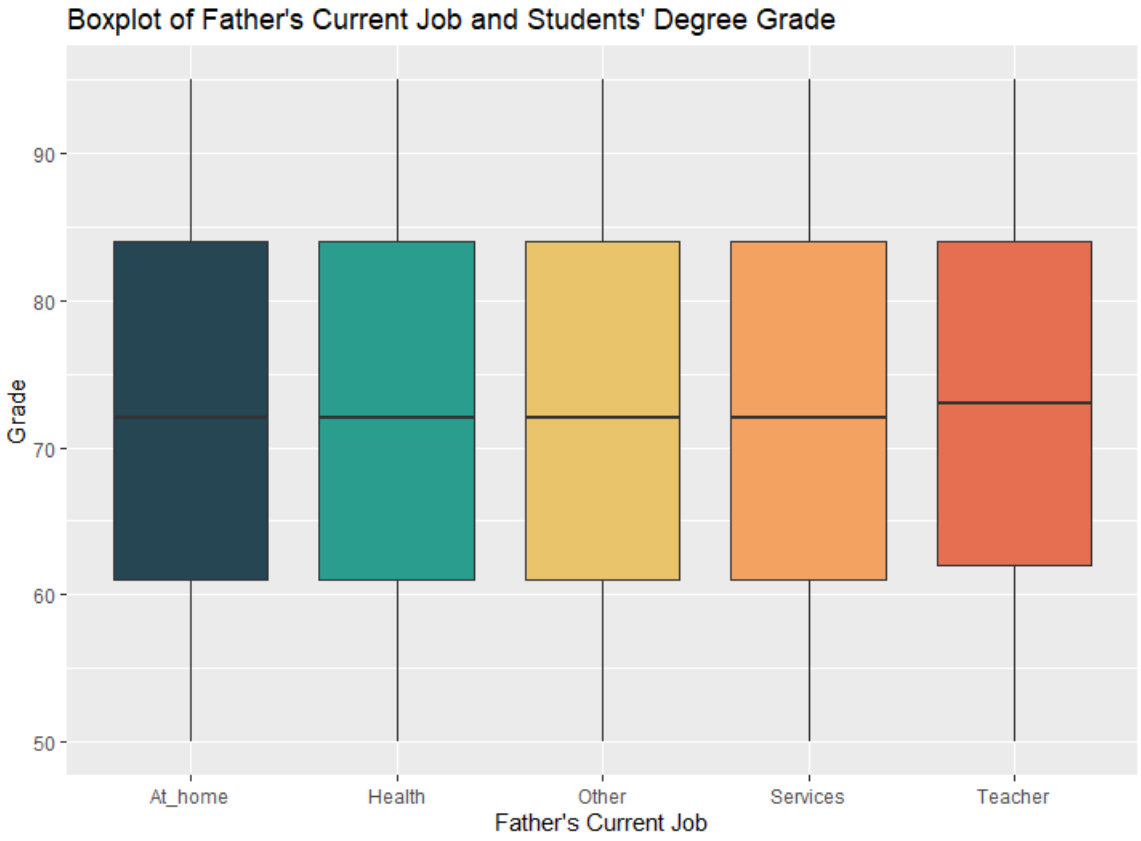

From this boxplot, it is clear that even though there is a slight positive increase of median for students' mother who is currently working in services industry, but comparing other boxes, the median and range of data are more or less the same. Thus, it has no effect on students' degree grade.

## 4.4 Father Current Job

```
# - Father current job
df_dad_job <- data.frame(
  dad_job_x = as.vector(placementData$Father_Current_Job),
  dad_job_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_dad_job, aes(x = stringr::str_to_title(factor(dad_job_x)), y = dad_job_y)) +
  geom_boxplot(fill = c("#264653", "#2A9D8F", "#E9C46A", "#F4A261", "#E76F51")) +
  labs(
    x = "Father's Current Job",
    y = "Grade",
    title = "Boxplot of Father's Current Job and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Father's Current Job to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



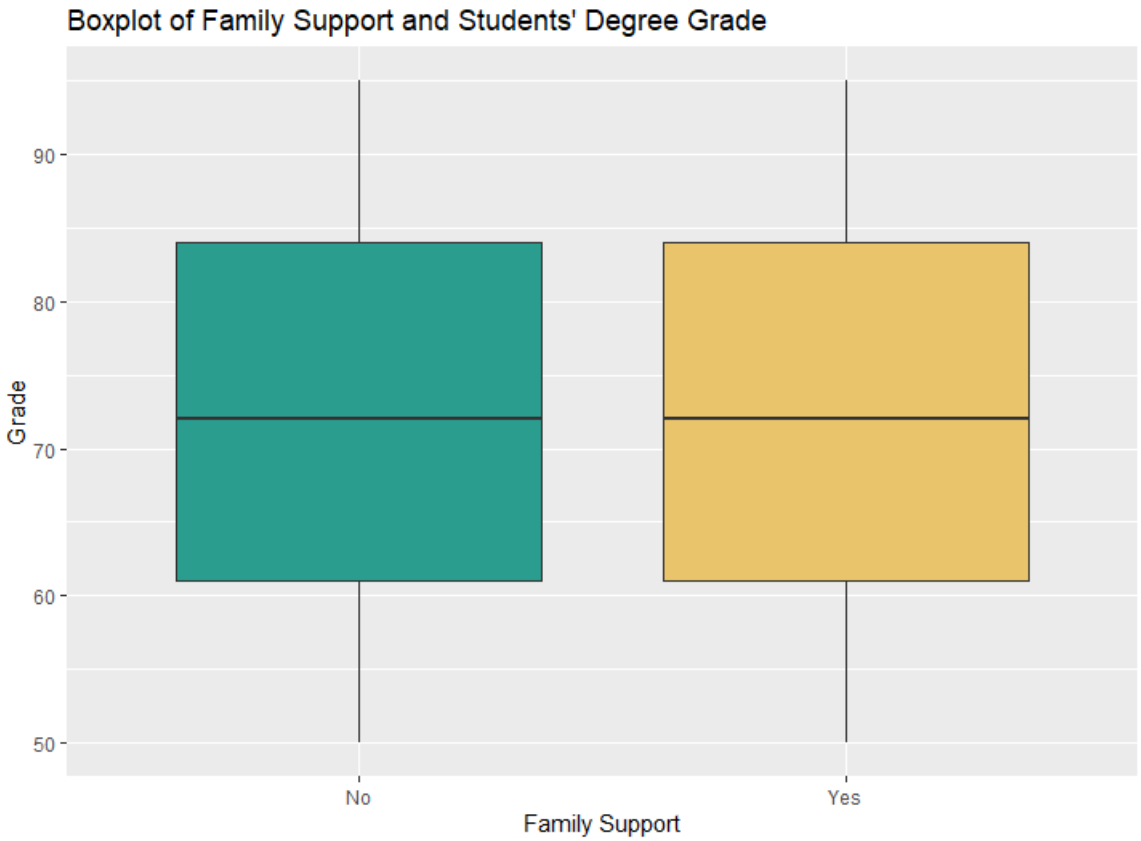

From this boxplot, it is clear that even though there is a slight positive increase of median for students' father who is currently working as a teacher, but comparing other boxes, the median and range of data are more or less the same. Thus, it has no effect on students' degree grade.

## 4.5 Family Support

```
# - Family Support
df_famsup <- data.frame(
  famsup_x = as.vector(placementData$Family_Support),
  famsup_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_famsup, aes(x = stringr::str_to_title(factor(famsup_x)), y = famsup_y)) +
  geom_boxplot(fill = c("#2A9D8F", "#E9C46A")) +
  labs(
    x = "Family Support",
    y = "Grade",
    title = "Boxplot of Family Support and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Family Support to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



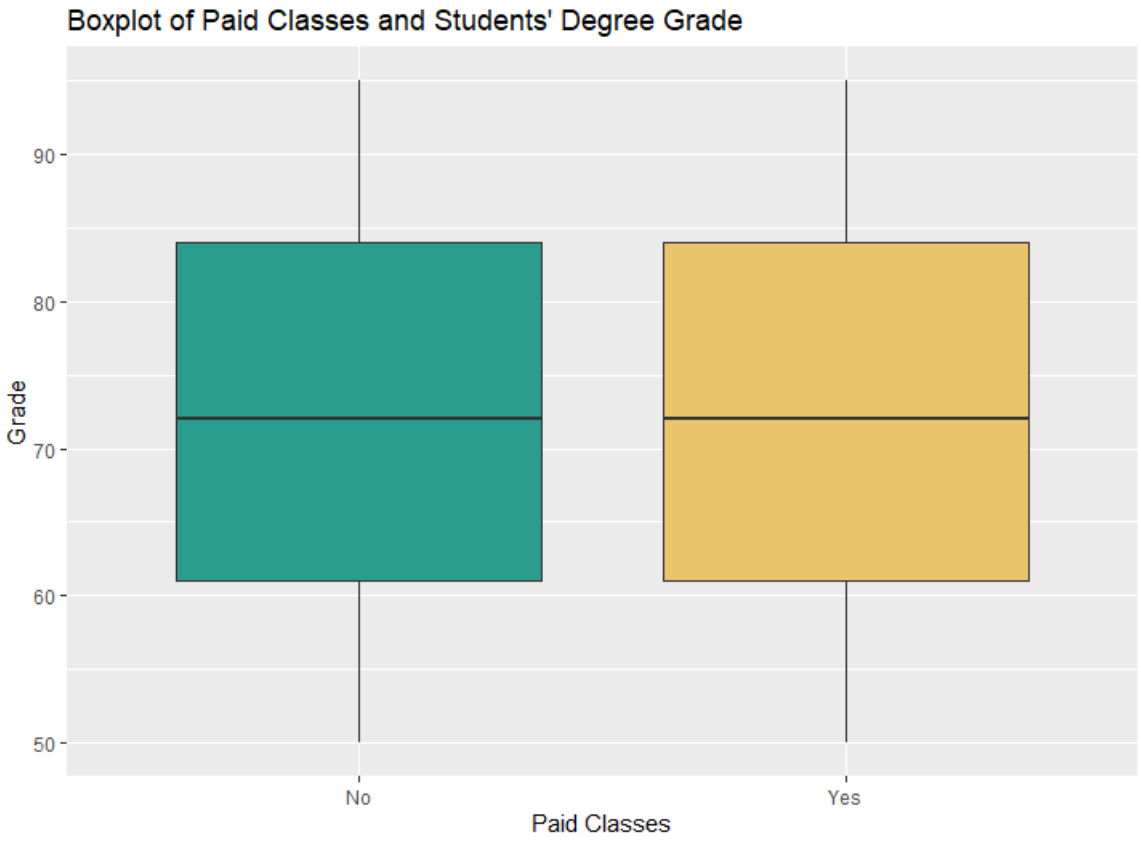

From this boxplot, it is clear that family support does not influence students to achieve degree marks higher than 70.

### 4.6 Paid Class

```
# - Paid class
df_paid <- data.frame(
  paid_x = as.vector(placementData$Paid_Classes),
  paid_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_paid, aes(x = stringr::str_to_title(factor(paid_x)), y = paid_y)) +
  geom_boxplot(fill = c("#2A9D8F", "#E9C46A")) +
  labs(
    x = "Paid Classes",
    y = "Grade",
    title = "Boxplot of Paid Classes and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Paid Class to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



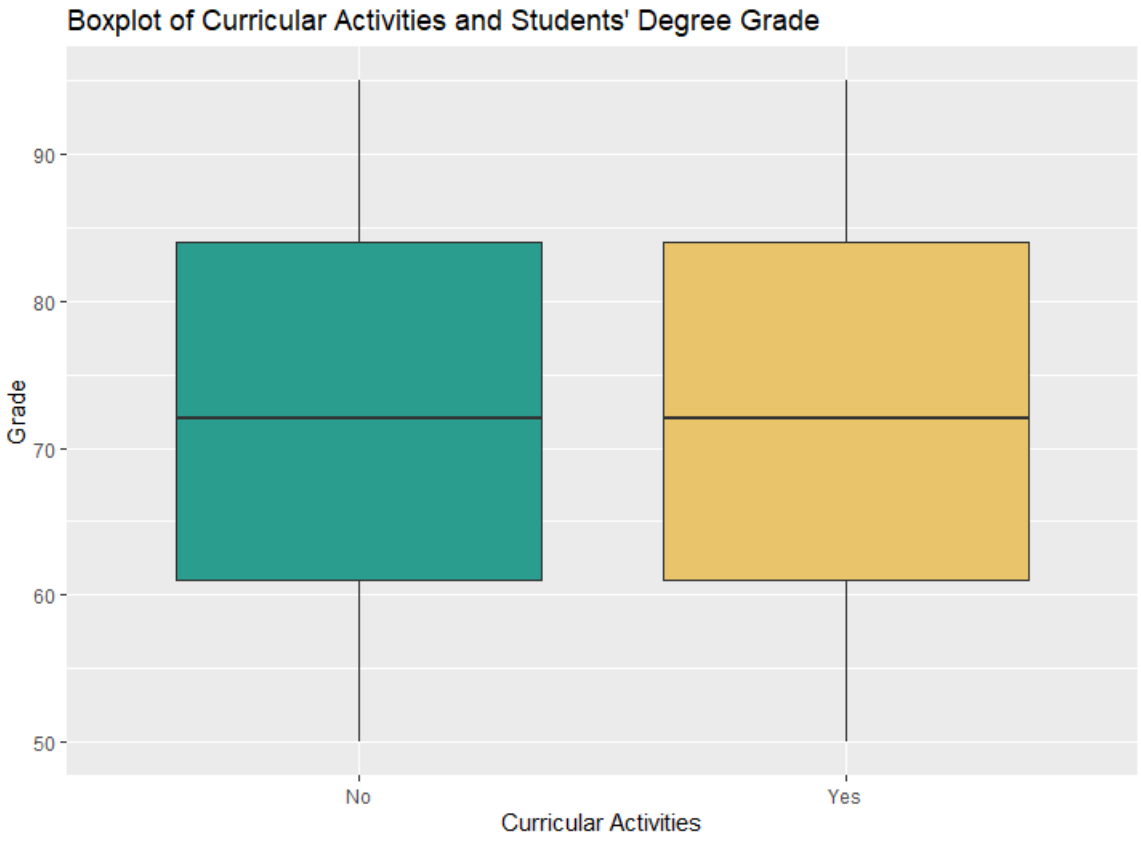

From this boxplot, it is clear that have or not have paid classes does not influence students to achieve degree marks higher than 70.

## 4.7 Curricular Activities

```
# - Curricular activities
df_activities <- data.frame(
  activities_x = as.vector(placementData$Curricular_Activities),
  activities_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_activities, aes(x = stringr::str_to_title(factor(activities_x)), y = activities_y)) +
  geom_boxplot(fill = c("#2A9D8F", "#E9C46A")) +
  labs(
    x = "Curricular Activities",
    y = "Grade",
    title = "Boxplot of Curricular Activities and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Curricular Activities to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



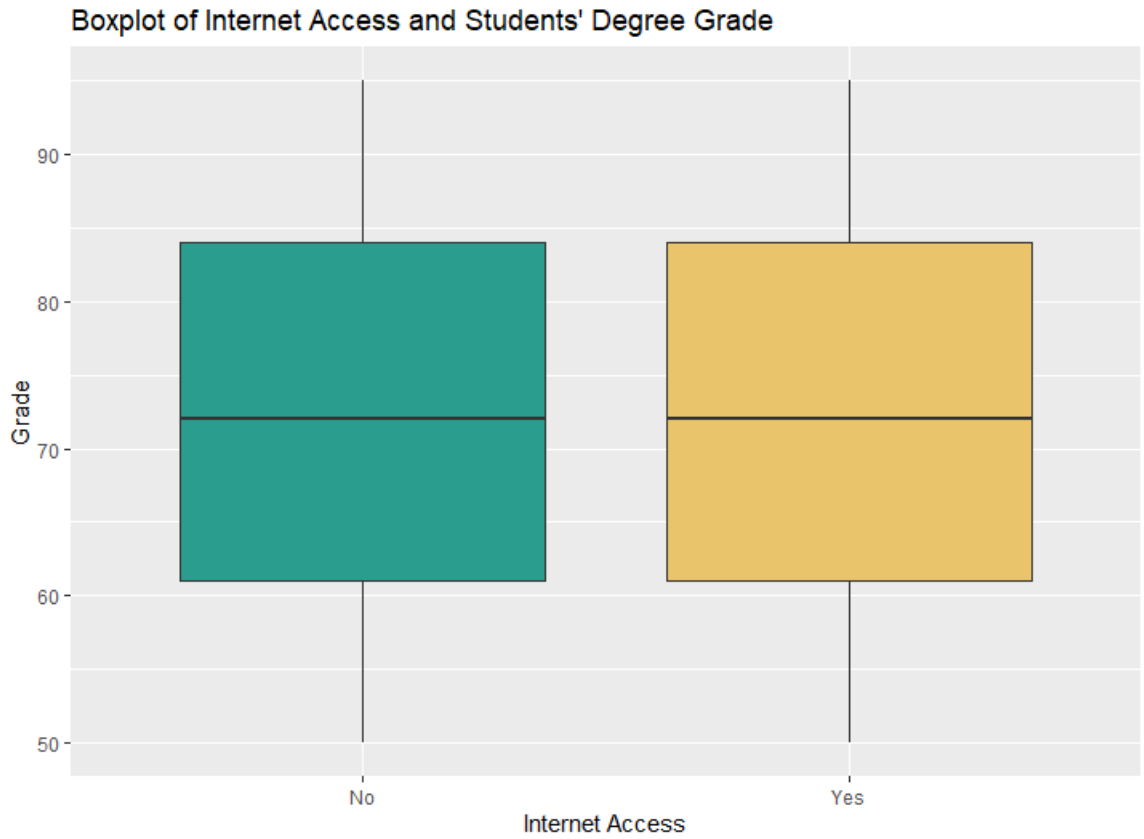

From this boxplot, it is clear that have or not have curricular activities does not influence students to achieve degree marks higher than 70.

## 4.8 Internet Access

```
# - Internet Access
df_internet <- data.frame(
  internet_x = as.vector(placementData$Internet_Usage),
  internet_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_internet, aes(x = stringr::str_to_title(factor(internet_x)), y = internet_y)) +
  geom_boxplot(fill = c("#2A9D8F", "#E9C46A")) +
  labs(
    x = "Internet Access",
    y = "Grade",
    title = "Boxplot of Internet Access and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Internet Access to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



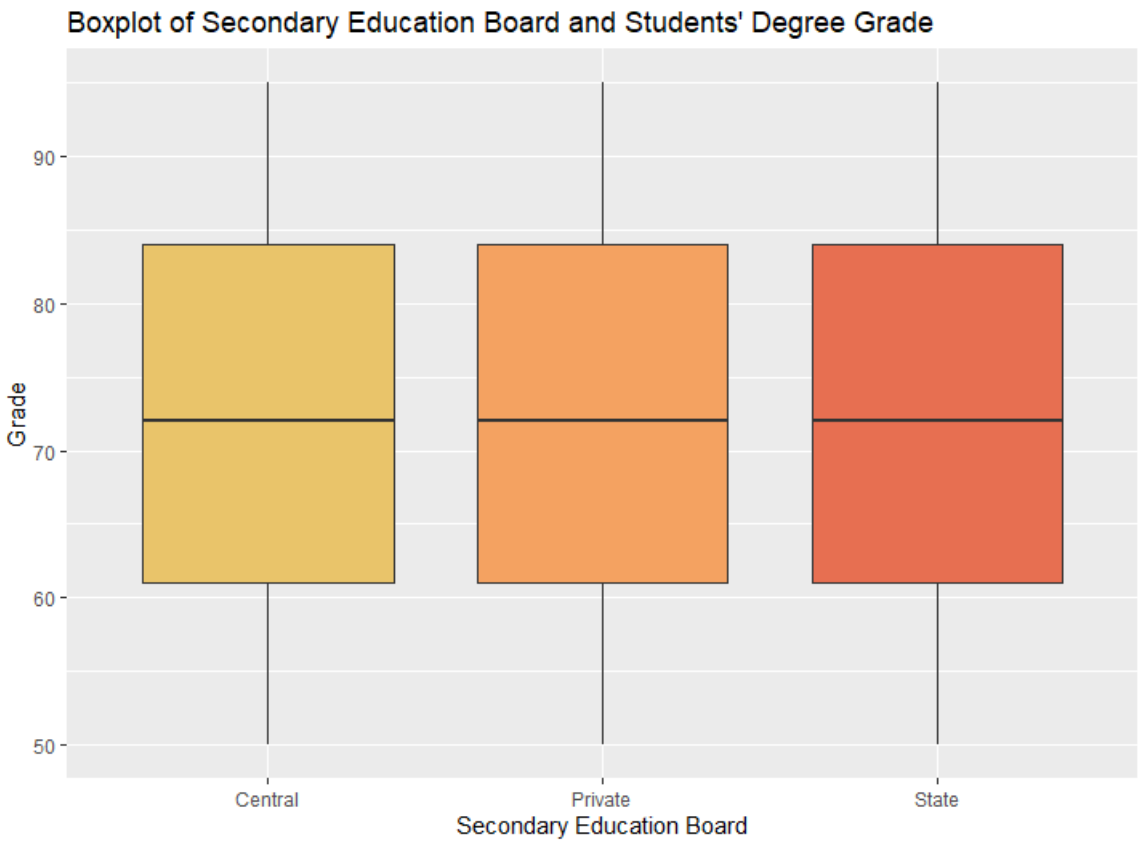

From this boxplot, it is clear that have or not have internet access does not influence students to achieve degree marks higher than 70.

## 4.9 Secondary Education Board

```
# - Secondary Education Board
df_secondary_board <- data.frame(
  secondary_board_x = as.vector(placementData$Secondary_Education_Board),
  secondary_board_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(df_secondary_board, aes(x = stringr::str_to_title(factor(secondary_board_x)), y = secondary_board_y)) +
  geom_boxplot(fill = c("#E9C46A", "#F4A261", "#E76F51")) +
  labs(
    x = "Secondary Education Board",
    y = "Grade",
    title = "Boxplot of Secondary Education Board and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Secondary Education Board to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:



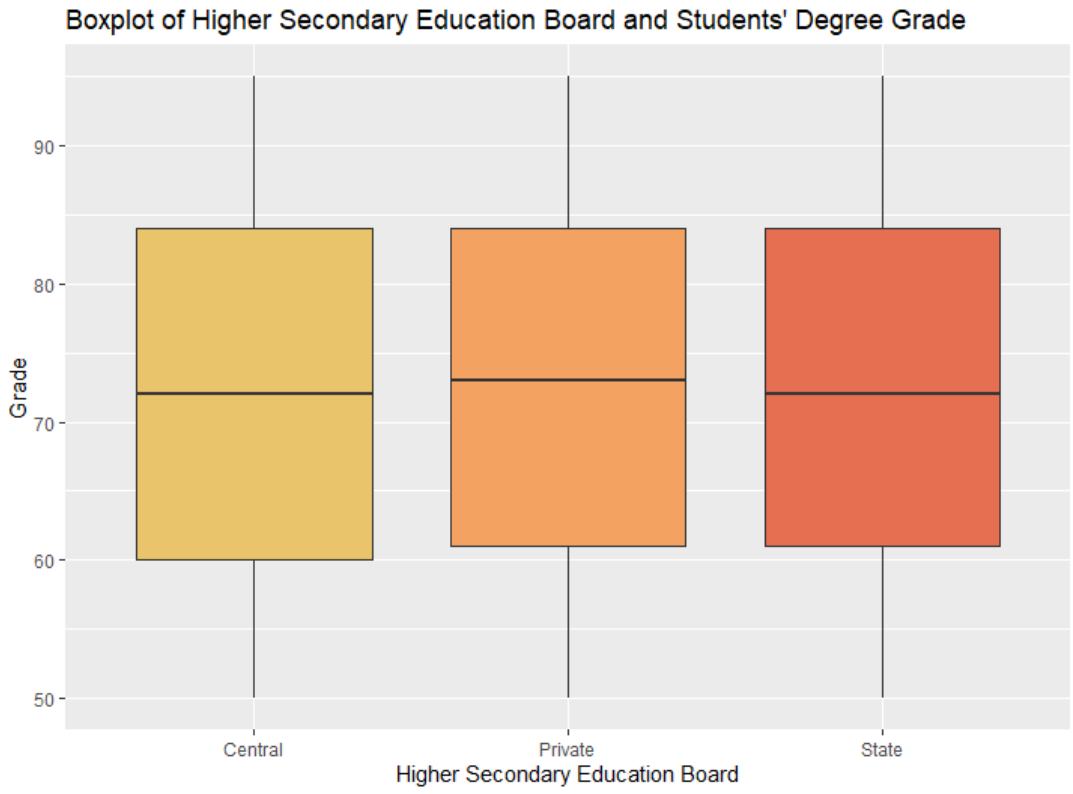

From this boxplot, it is clear that secondary education board does not influence students to achieve degree marks higher than 70.

## 4.10 Higher secondary Education Board

```
# - Higher secondary Education Board
df_higher_secondary_board <- data.frame(
  higher_secondary_board_x = as.vector(placementData$Higher_Secondary_Education_Board),
  higher_secondary_board_y = as.vector(placementData$Degree_Grade_Percentage)
)

ggplot(
    df_higher_secondary_board,
    aes(x = stringr::str_to_title(factor(higher_secondary_board_x)), y = higher_secondary_board_y)
  ) +
  geom_boxplot(fill = c("#E9C46A", "#F4A261", "#E76F51")) +
  labs(
    x = "Higher Secondary Education Board",
    y = "Grade",
    title = "Boxplot of Higher Secondary Education Board and Students' Degree Grade"
  )
```

The code above demonstrates on how to create a boxplot of **Higher Secondary Education Board to Students' Degree Grade** using the GGPlot2 library.

The output of the code is as below:




From this boxplot, it is clear that higher secondary education board does not influence students to achieve degree marks higher than 70, even though that is a slight increase in median of private education board.

## 4.11 Analysis Conclusion

According to all box plot analysis, it is obvious that **none of the factors above** has impact on students' degree grade. It is accurate and logical because students who are studying at degree level are mostly self-independent thus external factors like parent's education level, parent's job, family support, paid classes, curricular activities, internet access, secondary and higher secondary education board will not affect students' degree grade. As long as students are independent and hard-working, they will still achieve a distinction.

# 5.0 Question 4 – What will affect students' master?

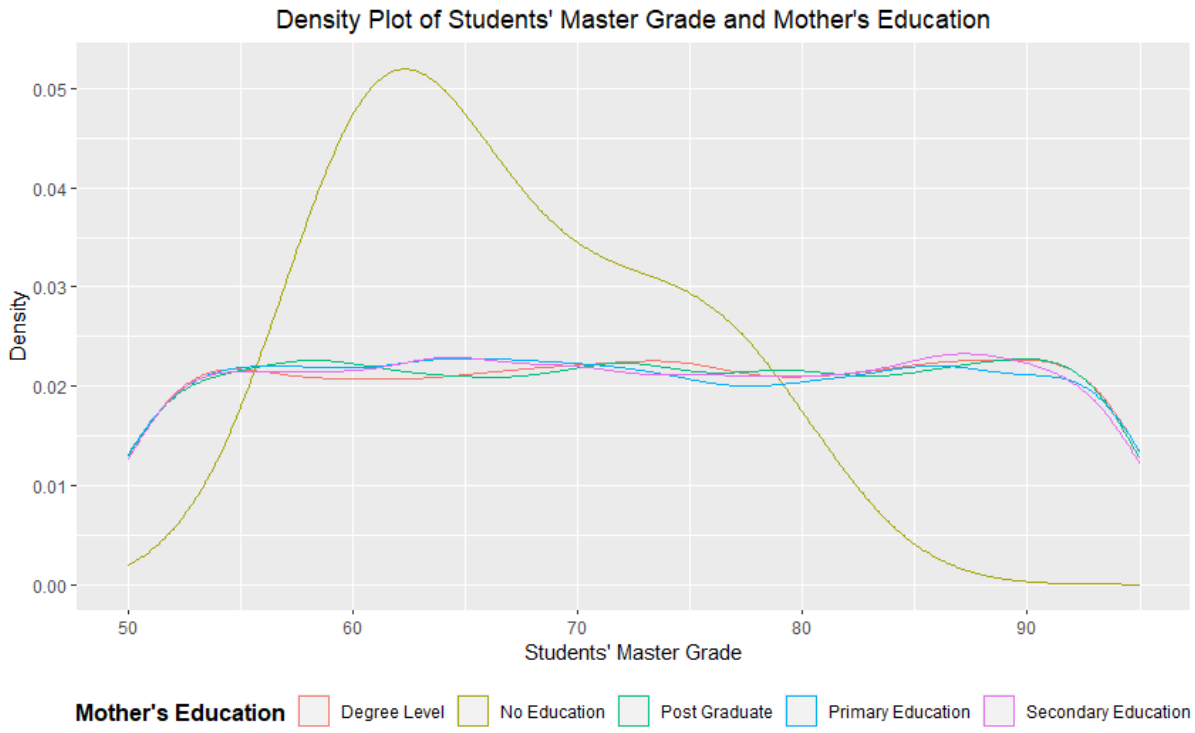
Multiple factors from the imported dataset are used to investigate what will affect students' master grade by creating **density plot** as data visualization.

## 5.1 Mother Education

```
# - Mother education
df_mom_edu <- data.frame(
  student_mom_edu = as.vector(placementData$Mother_Education),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(df_mom_edu, aes(x = student_master, y = after_stat(density), color = factor(student_mom_edu))) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Mother's Education"
  ) +
  scale_color_discrete(name = "Mother's Education") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Mother's Education** using the GGPlot2 library. The **geom_density()** function is to add a density layer to the graph while the **alpha()** function inside is to specify the transparency of the density lines. The **after_stat(density)** function is to standardize the **student_master** variable for computation. The **scale_color_discrete()** function is to set the legend title. Inside the **theme()** function, **plot.title** is to specify the horizontal justification of the title, **legend.title** is to specify the font size and weight, and **legend.position** is to specify the position of the legend in the graph.

The output of the code is as below:



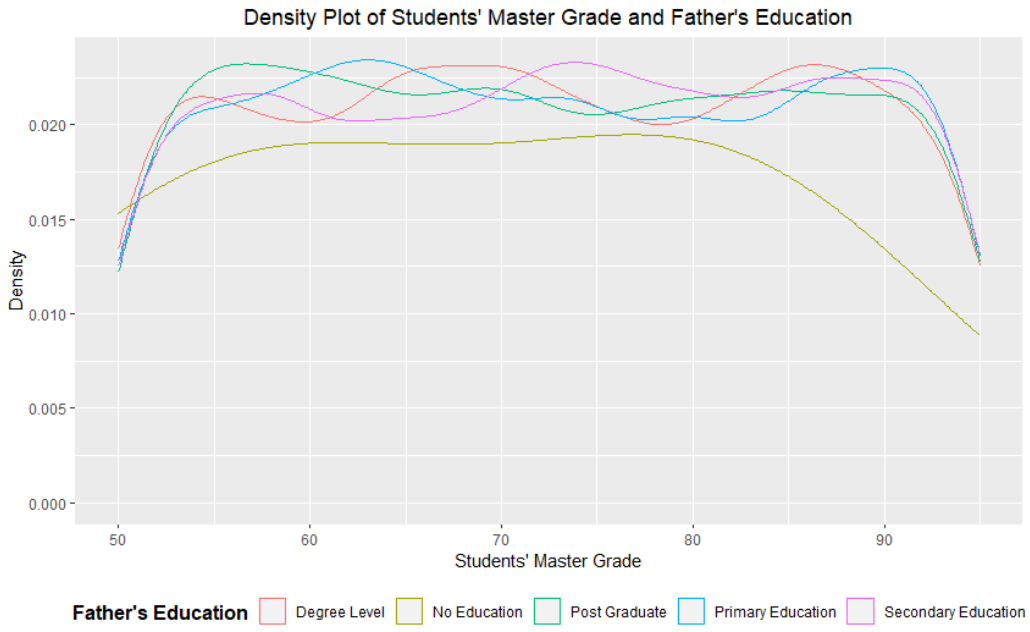

From this density graph, it is clear that students' mother that has no education will have a significant effect on students' master grade as the peak suggests that students only achieve marks around 55 to below 80 in their master's degree.

## 5.2 Father Education

```r
# - Father education
df_dad_edu <- data.frame(
  student_dad_edu = as.vector(placementData$Father_Education),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(df_dad_edu, aes(x = student_master, y = after_stat(density), color = factor(student_dad_edu))) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Father's Education"
  ) +
  scale_color_discrete(name = "Father's Education") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Father's Education** using the GGPlot2 library.

The output of the code is as below:



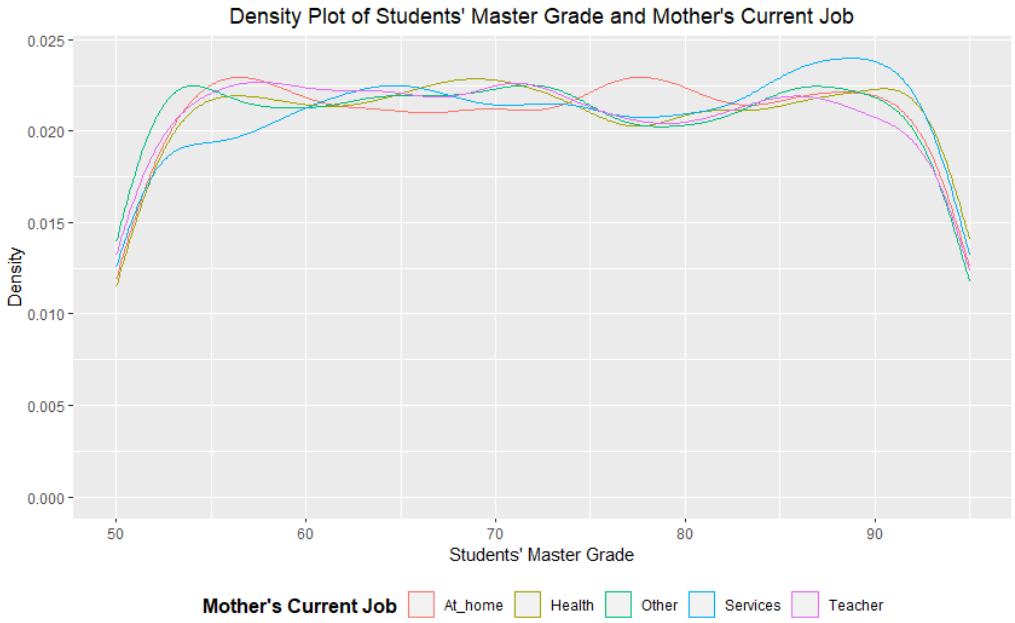

From this density graph, it is clear that students' father that has no education will have least effect on students' master grade as the density line of the category suggests that other factors like socio-economic status, access to educational resources, or parental involvement might be in play because of the distribution is lower than students' father who has at least primary education to post graduate education. Comparing the density lines of students' father who has at least primary education to post graduate education, they are more likely to affect students' master grade.

## 5.3 Mother Current Job

```
# - Mother current job
df_mom_job <- data.frame(
  student_mom_job = as.vector(placementData$Mother_Current_Job),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
  df_mom_job,
  aes(
    x = student_master,
    y = after_stat(density),
    color = stringr::str_to_title(factor(student_mom_job))
  )) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Mother's Current Job"
  ) +
  scale_color_discrete(name = "Mother's Current Job") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Mother's Current Job** using the GGPlot2 library.

The output of the code is as below:



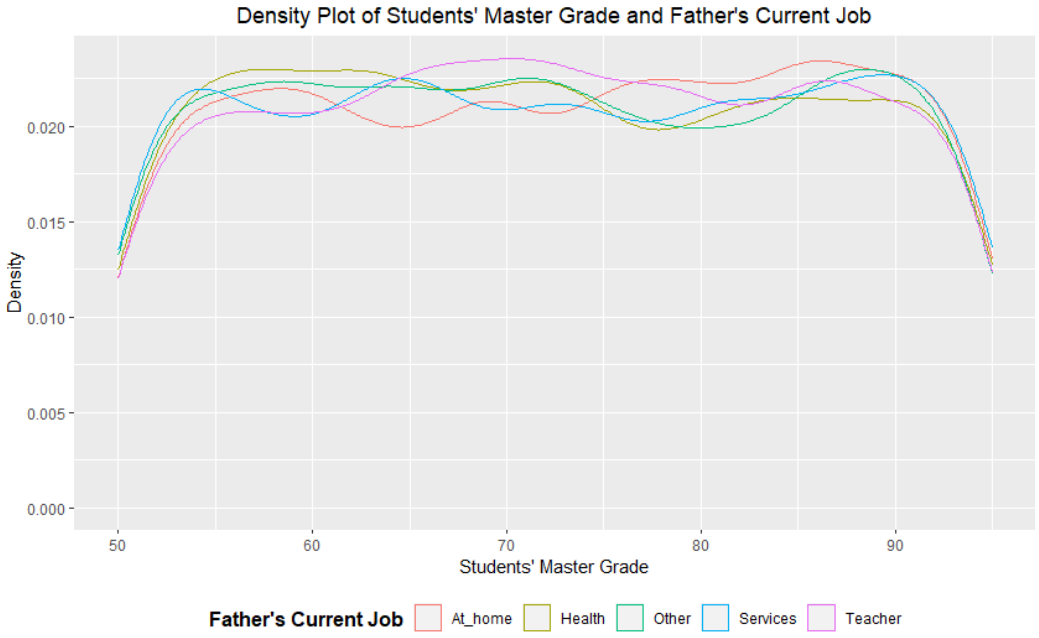

From this density graph, it is clear that students' mother's current job has no effect on students' master grade.

## 5.4 Father Current Job

```
# - Father current job
df_dad_job <- data.frame(
  student_dad_job = as.vector(placementData$Father_Current_Job),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
  df_dad_job,
  aes(
    x = student_master, y = after_stat(density),
    color = stringr::str_to_title(factor(student_dad_job))
  )) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Father's Current Job"
  ) +
  scale_color_discrete(name = "Father's Current Job") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Father's Current Job** using the GGPlot2 library.

The output of the code is as below:



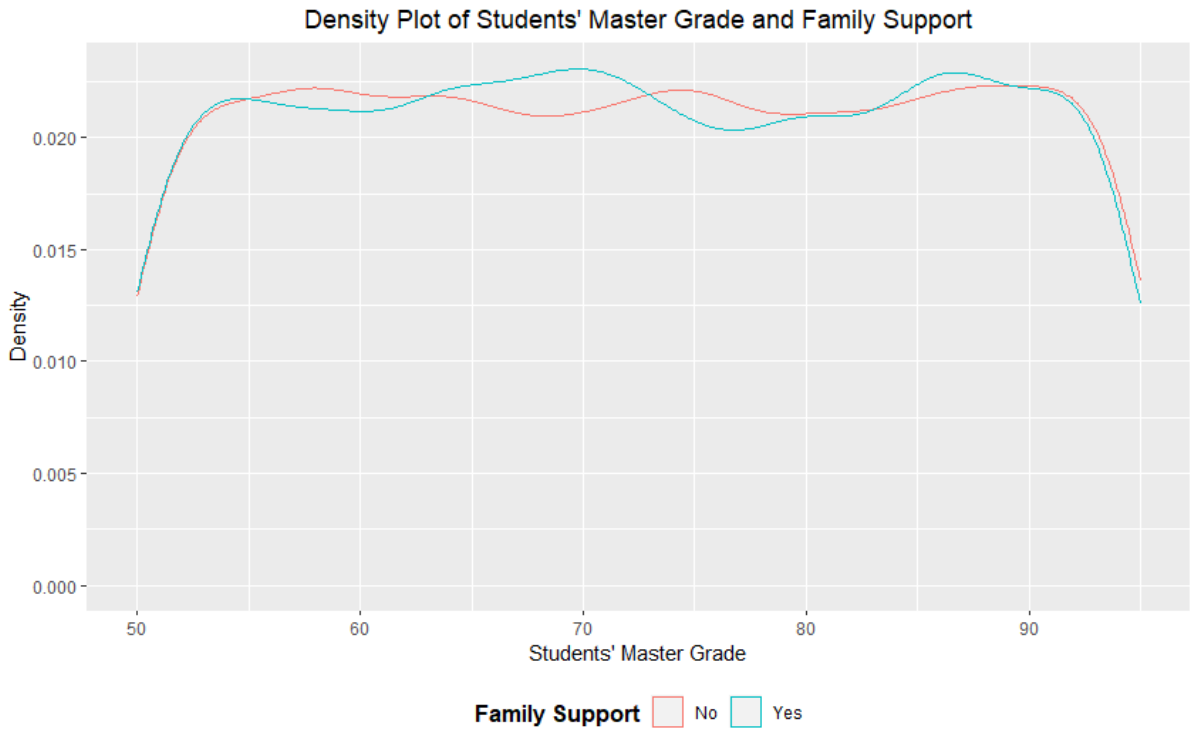

From this density graph, it is clear that students' father's current job has no effect on students' master grade.

## 5.5 Family Support

```
# - Family Support
df_famsup <- data.frame(
  student_famsup = as.vector(placementData$Family_Support),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
    df_famsup,
    aes(x = student_master, y = after_stat(density), color = stringr::str_to_title(factor(student_famsup)))
  ) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Family Support"
  ) +
  scale_color_discrete(name = "Family Support") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Family Support** using the GGPlot2 library.

The output of the code is as below:



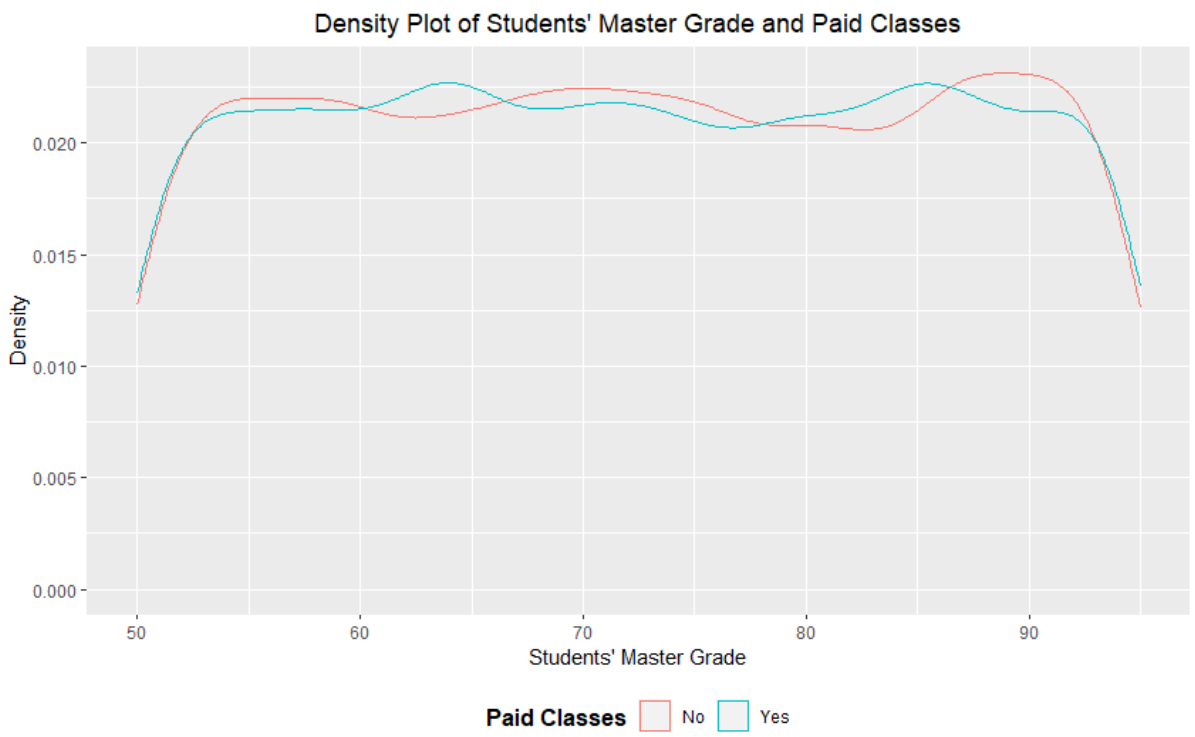

From this density graph, it is clear that family support has no effect on students' master grade.

## 5.6 Paid Class

```
# - Paid class
df_paid <- data.frame(
  student_paid = as.vector(placementData$Paid_Classes),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
    df_paid,
    aes(x = student_master, y = after_stat(density), color = stringr::str_to_title(factor(student_paid)))
  ) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Paid Classes"
  ) +
  scale_color_discrete(name = "Paid Classes") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Paid Class** using the GGPlot2 library.

The output of the code is as below:



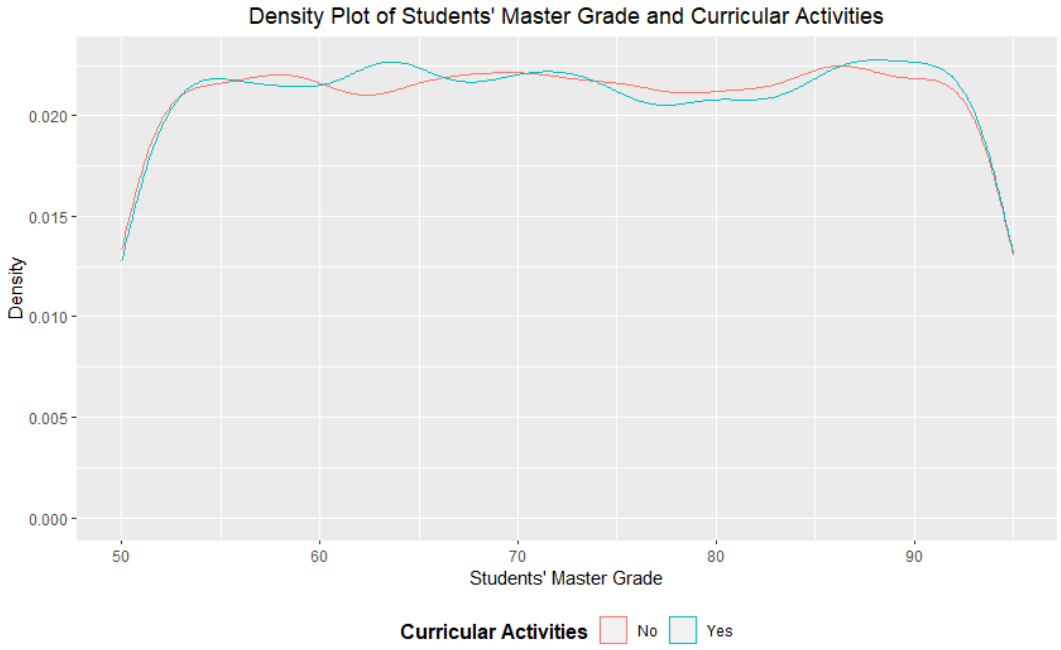

From this density graph, it is clear that students who have or not have paid classes has no effect on students' master grade.

## 5.7 Curricular Activities

```
# - Curricular activities
df_activities <- data.frame(
  student_activities = as.vector(placementData$Curricular_Activities),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
  df_activities,
  aes(
    x = student_master, y = after_stat(density),
    color = stringr::str_to_title(factor(student_activities))
  )) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Curricular Activities"
  ) +
  scale_color_discrete(name = "Curricular Activities") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Curricular Activities** using the GGPlot2 library.

The output of the code is as below:



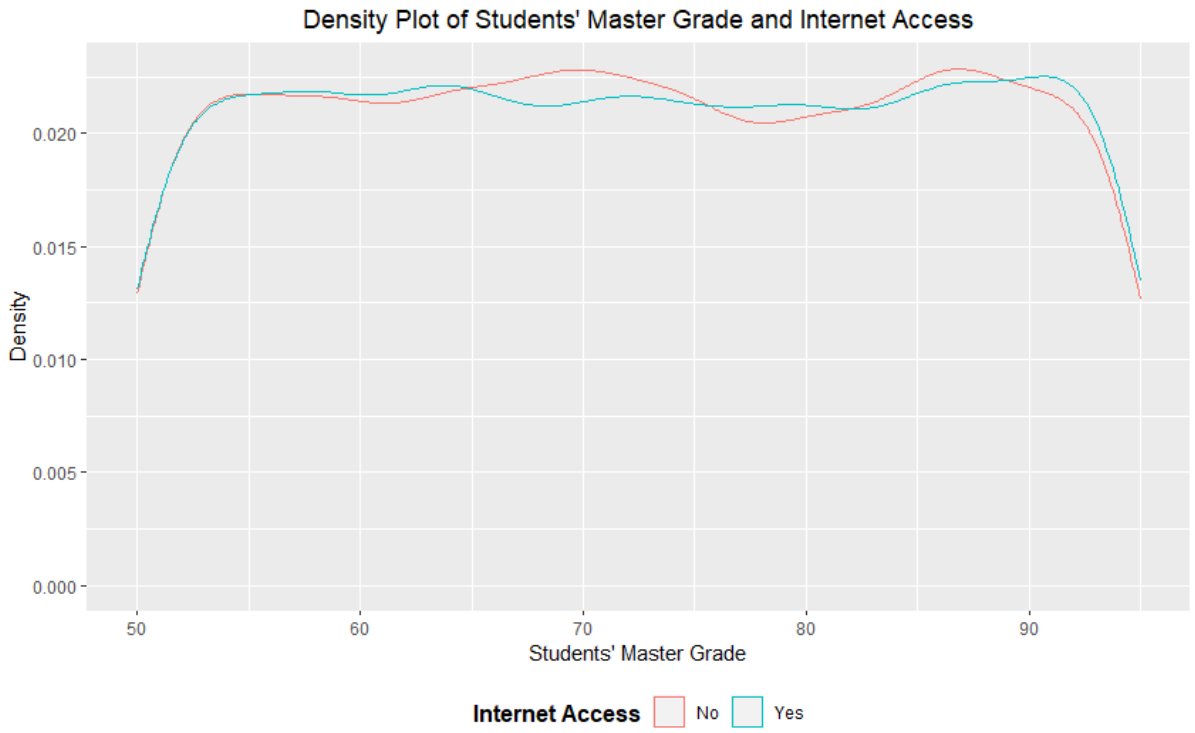

From this density graph, it is clear that students who have or not have curricular activities has no effect on students' master grade.

## 5.8 Internet Access

```
# - Internet Access
df_internet <- data.frame(
  student_internet = as.vector(placementData$Internet_Usage),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
    df_internet,
    aes(x = student_master, y = after_stat(density), color = stringr::str_to_title(factor(student_internet)))
  ) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Internet Access"
  ) +
  scale_color_discrete(name = "Internet Access") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Internet Access** using the GGPlot2 library.

The output of the code is as below:



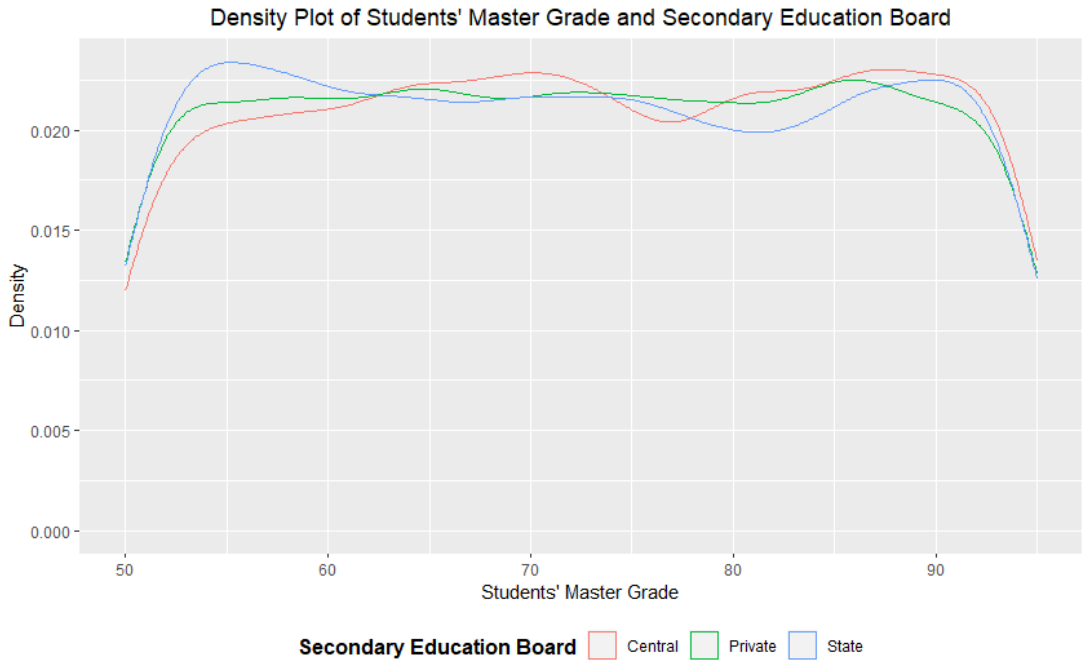

From this density graph, it is clear that students who have or not have internet access has no effect on students' master grade.

## 5.9 Secondary Education Board

```
# - Secondary Education Board
df_secondary_board <- data.frame(
  student_secondary_board = as.vector(placementData$Secondary_Education_Board),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
  df_secondary_board,
  aes(
    x = student_master, y = after_stat(density),
    color = stringr::str_to_title(factor(student_secondary_board))
  )) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Secondary Education Board"
  ) +
  scale_color_discrete(name = "Secondary Education Board") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Secondary Education Board** using the GGPlot2 library.

The output of the code is as below:



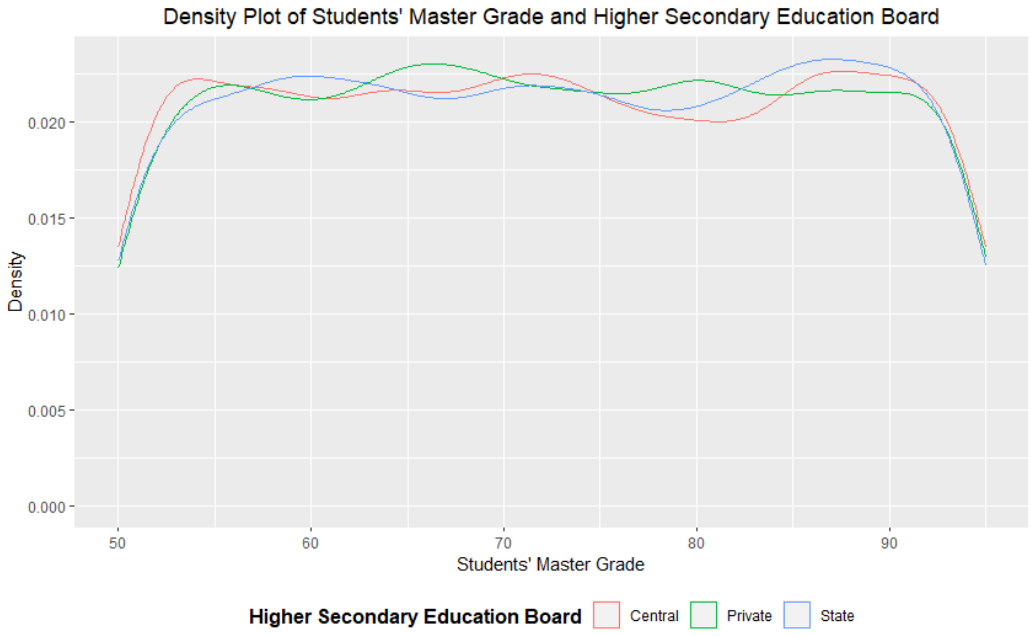

From this density graph, it is clear that students' secondary education board has no effect on students' master grade.

## 5.10 Higher Secondary Education Board

```
# - Higher secondary Education Board
df_higher_secondary_board <- data.frame(
  student_higher_secondary_board = as.vector(placementData$Higher_Secondary_Education_Board),
  student_master = as.vector(placementData$Master_Grade_Percentage)
)

ggplot(
  df_higher_secondary_board,
  aes(
    x = student_master, y = after_stat(density),
    color = stringr::str_to_title(factor(student_higher_secondary_board))
  )) +
  geom_density(alpha = 0.6) +
  labs(
    x = "Students' Master Grade", y = "Density",
    title = "Density Plot of Students' Master Grade and Higher Secondary Education Board"
  ) +
  scale_color_discrete(name = "Higher Secondary Education Board") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Students' Master Grade to Higher Secondary Education Board** using the GGPlot2 library.

The output of the code is as below:




From this density graph, it is clear that students' higher secondary education board has no effect on students' master grade.

## 5.11 Analysis Conclusion

According to all density plot, **only parent education** level will affect students' master's grade where students' mother that has no education will have a significant effect on students' master grade as the peak suggests that students only achieve marks around 55 to below 80 in their master's degree and students' father who has no education will have least effect on students' master grade. Comparing the density lines of students' father who has at least primary education to post graduate education, they are more likely to affect students' master grade.

This probably suggests that students most likely to look upon their fathers as their role model and the father's education level when they reach the age of an adult as they are taking master's degree.

Other factors like parent's job, family support, paid classes, curricular activities, internet access and secondary and higher secondary education board doesn't affect students' master's grade.

# 6.0 Question 5 – What will affect students' employment test?

Multiple factors from the imported dataset are used to investigate what will affect students' employment test by creating **violin plot** as data visualization.

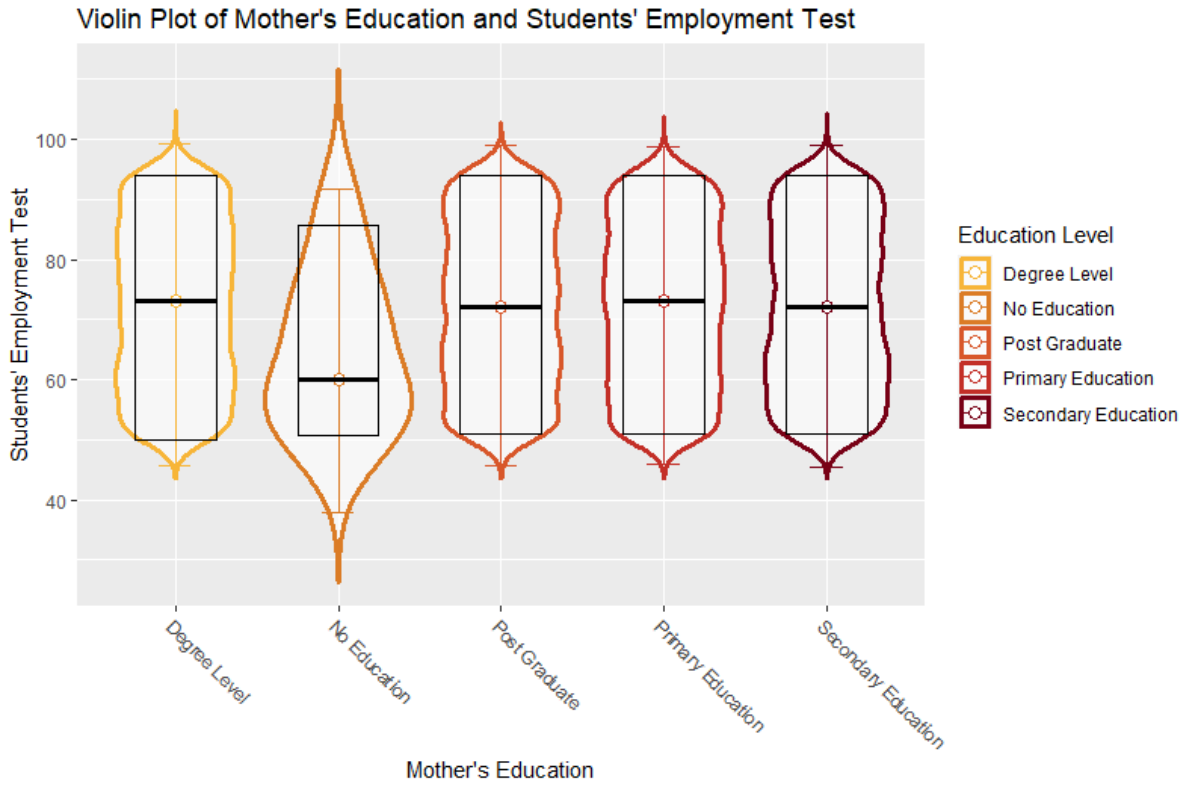
## 6.1 Mother Education

```
# - Mother education
df_mom_edu <- data.frame(
  student_mom_edu = as.vector(placementData$Mother_Education),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(df_mom_edu, aes(x = factor(student_mom_edu), y = student_employ_test, color = factor(student_mom_edu))) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#F7B538", "#DB7C26", "#D8572A", "#C32F27", "#780116")) +
  labs(
    x = "Mother's Education",
    y = "Students' Employment Test",
    title = "Violin Plot of Mother's Education and Students' Employment Test",
    color = "Education Level"
  ) +
  theme(axis.text.x = element_text(angle = -45, hjust = 0)) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Mother's Education** using the GGPlot2 library. The **geom_violin()** function is used to create a violin plot and **trim = FALSE** to ensure the full range of data is display while **alpha** and **linewidth** is to control the transparency and line width of the violins respectively.

The first **stat_summary()** function is used to add summary statistics to the violin plot. **fun.data = mean_sdl** is to calculate the mean and standard deviation of the data and **geom = "errorbar"** is to display the error bars on the plot. **fun = "median"** is to calculate the median of the data and **geom = "point"** is to display the median as a point in the plot. **fun.data = median_hilow** is to display the median and the 95% confidence interval as a crossbar.

The output of the code is as below:



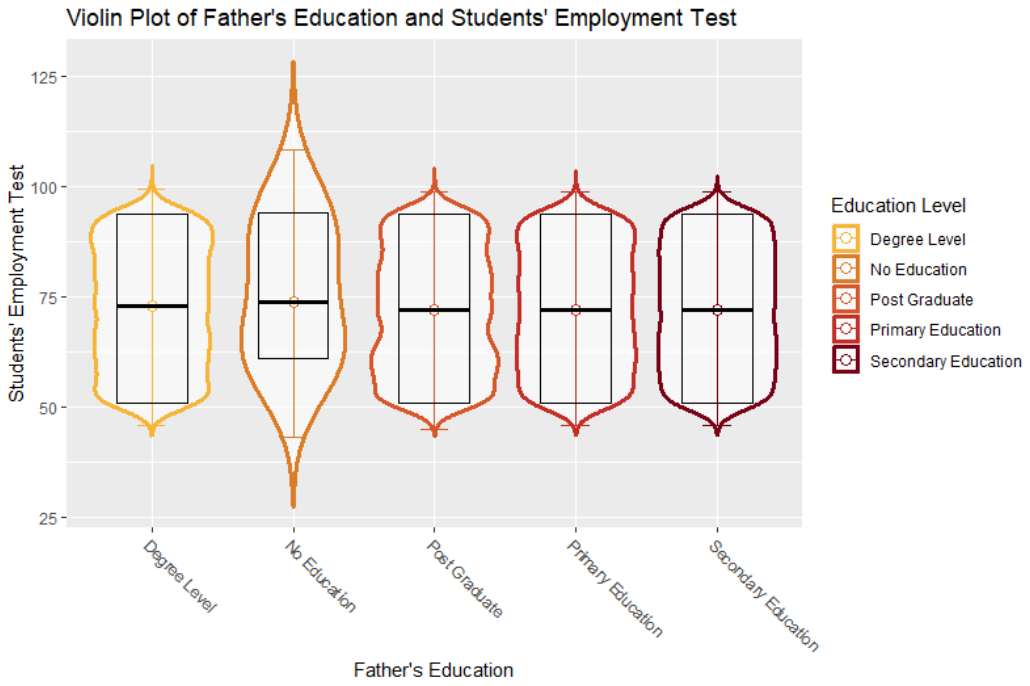

From this violin plot, it is clear that students' mother without education will significantly affect students to have a lower employment test grade due to a lower median compared to other violins. However, this hypothesis can be countered due to the distance of Q1 to median is closer than the distance of Q3 to median which means there are a lot of high values between Q3 to median and the data is not as sufficient as the other violins.

## 6.2 Father Education

```
# - Father education
df_dad_edu <- data.frame(
  student_dad_edu = as.vector(placementData$Father_Education),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(df_dad_edu, aes(x = factor(student_dad_edu), y = student_employ_test, color = factor(student_dad_edu))) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#F7B538", "#DB7C26", "#D8572A", "#C32F27", "#780116")) +
  labs(
    x = "Father's Education",
    y = "Students' Employment Test",
    title = "Violin Plot of Father's Education and Students' Employment Test",
    color = "Education Level"
  ) +
  theme(axis.text.x = element_text(angle = -45, hjust = 0)) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Father's Education** using the GGPlot2 library.

The output of code is as below:



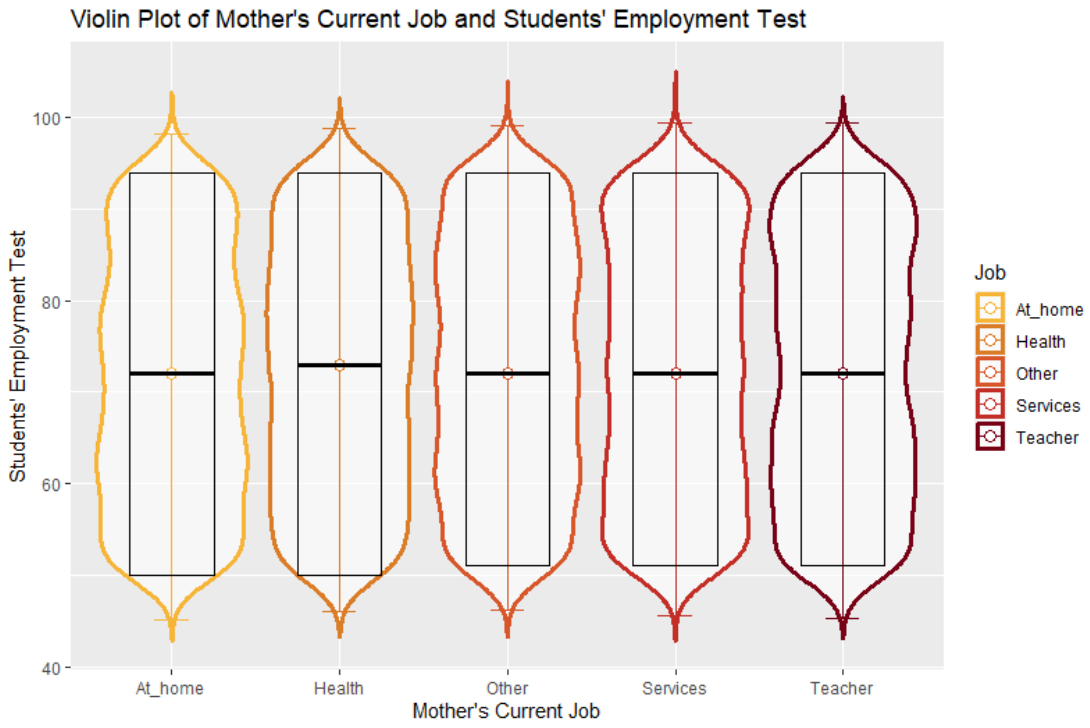

From this violin plot, it is clear that students' father without education will significantly affect students to have a higher employment test grade due to the distance of Q1 to median is closer than the distance of Q3 to median which means there are a lot of high values between Q3 to median. However, this hypothesis can be countered due to the data is not as sufficient as the other violins.

## 6.3 Mother Current Job

```
# - Mother current job
df_mom_job <- data.frame(
  student_mom_job = as.vector(placementData$Mother_Current_Job),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_mom_job,
  aes(
    x = stringr::str_to_title(factor(student_mom_job)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_mom_job))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#F7B538", "#DB7C26", "#D8572A", "#C32F27", "#780116")) +
  labs(
    x = "Mother's Current Job",
    y = "Students' Employment Test",
    title = "Violin Plot of Mother's Current Job and Students' Employment Test",
    color = "Job"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Mother's Current Job** using the GGPlot2 library.

The output of code is as below:



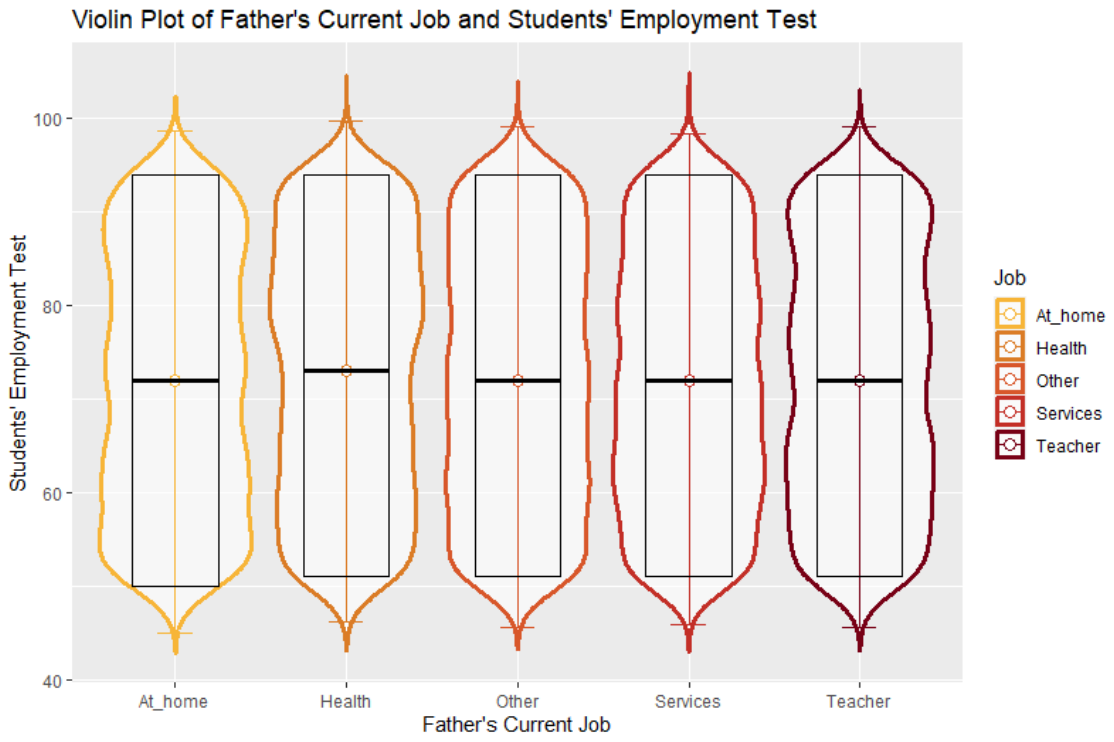

From this violin plot, it is clear that students' mother's current job does not affect students' employment test grade.

## 6.4 Father Current Job

```
# - Father current job
df_dad_job <- data.frame(
  student_dad_job = as.vector(placementData$Father_Current_Job),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_dad_job,
  aes(
    x = stringr::str_to_title(factor(student_dad_job)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_dad_job))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#F7B538", "#DB7C26", "#D8572A", "#C32F27", "#780116")) +
  labs(
    x = "Father's Current Job",
    y = "Students' Employment Test",
    title = "Violin Plot of Father's Current Job and Students' Employment Test",
    color = "Job"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Father's Current Job** using the GGPlot2 library.

The output of code is as below:



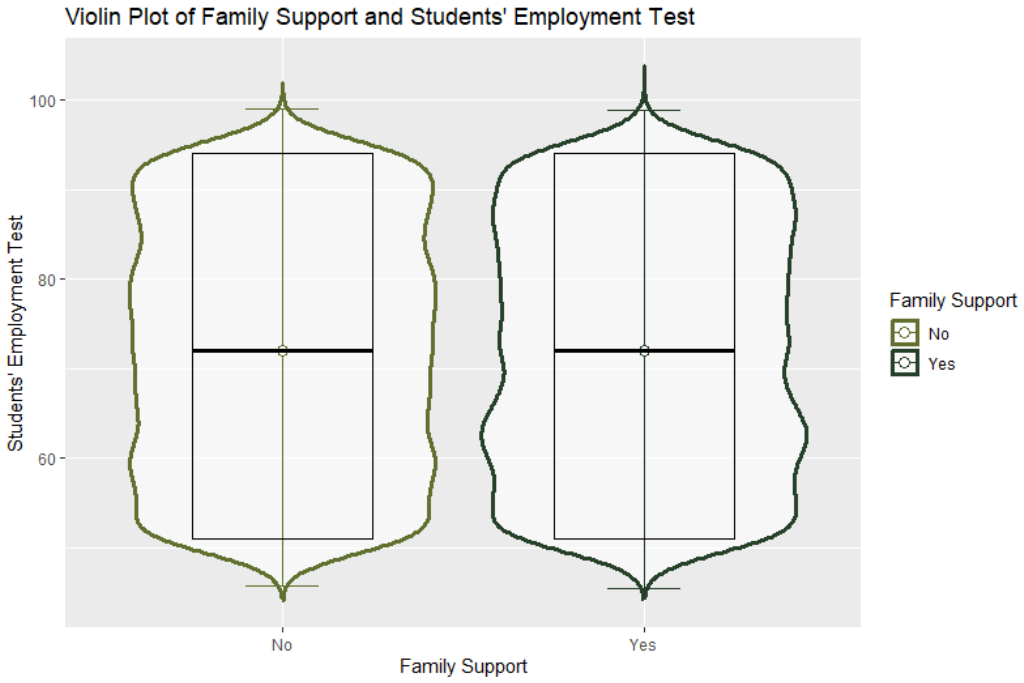

From this violin plot, it is clear that students' father's current job does not affect students' employment test grade.

## 6.5 Family Support

```
# - Family Support
df_famsup <- data.frame(
  student_famsup = as.vector(placementData$Family_Support),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_famsup,
  aes(
    x = stringr::str_to_title(factor(student_famsup)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_famsup))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#60712F", "#274029")) +
  labs(
    x = "Family Support",
    y = "Students' Employment Test",
    title = "Violin Plot of Family Support and Students' Employment Test",
    color = "Family Support"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Family Support** using the GGPlot2 library.

The output of code is as below:



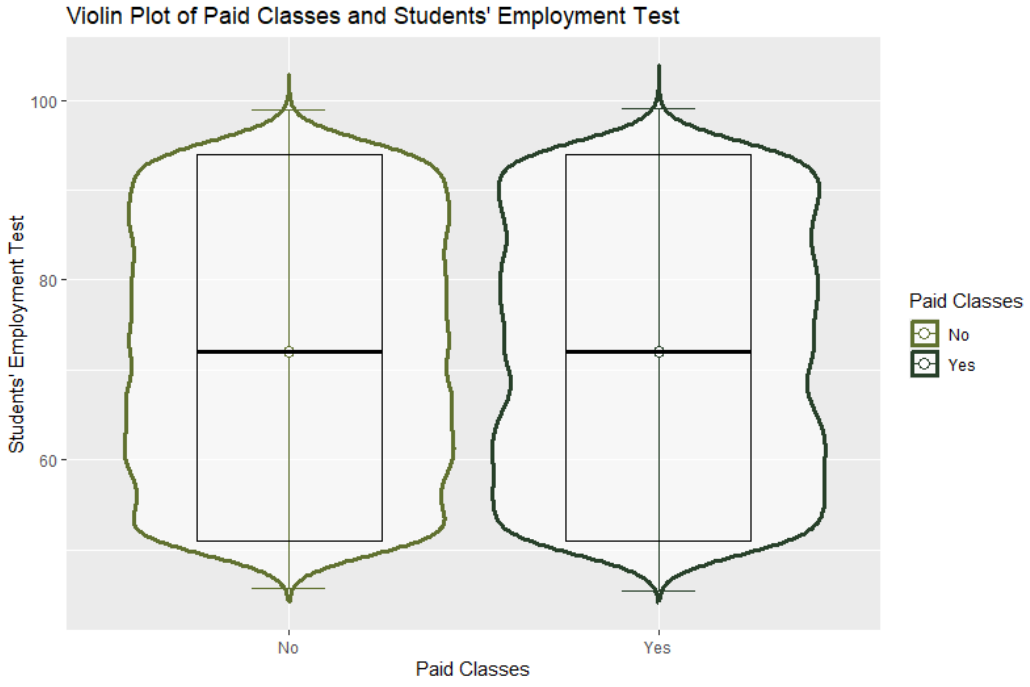

From this violin plot, it is clear that family support does not affect students' employment test grade.

## 6.6 Paid Class

```
# - Paid class
df_paid <- data.frame(
  student_paid = as.vector(placementData$Paid_Classes),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_paid,
  aes(
    x = stringr::str_to_title(factor(student_paid)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_paid))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#60712F", "#274029")) +
  labs(
    x = "Paid Classes",
    y = "Students' Employment Test",
    title = "Violin Plot of Paid Classes and Students' Employment Test",
    color = "Paid Classes"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Paid Class** using the GGPlot2 library.

The output of code is as below:



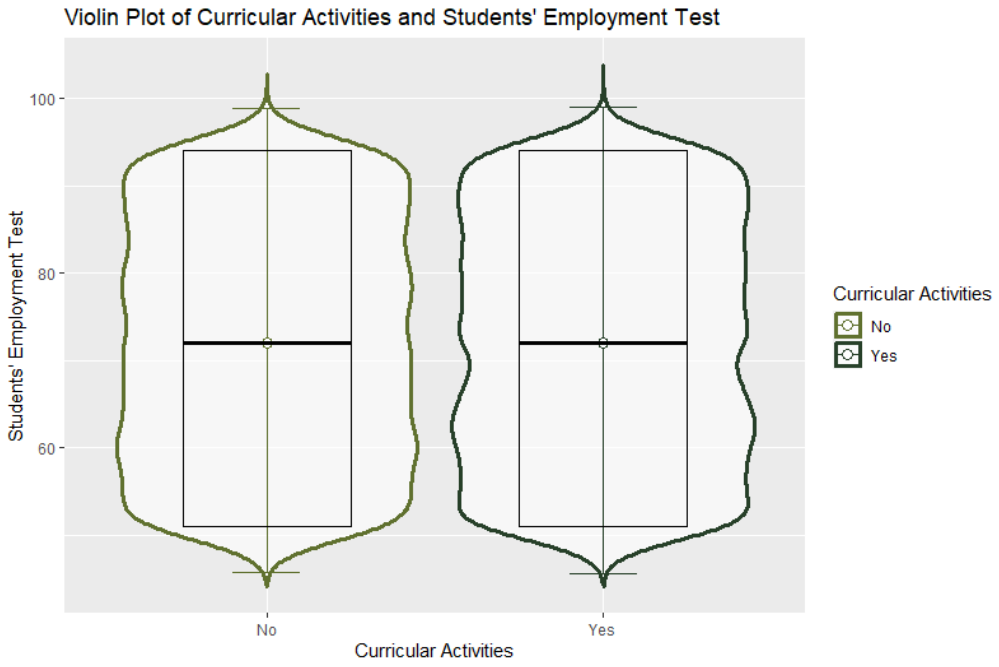

From this violin plot, it is clear that students who have or not have paid classes does not affect students' employment test grade.

## 6.7 Curricular Activities

```
# - Curricular activities
df_activities <- data.frame(
  student_activities = as.vector(placementData$Curricular_Activities),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_activities,
  aes(
    x = stringr::str_to_title(factor(student_activities)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_activities))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#60712F", "#274029")) +
  labs(
    x = "Curricular Activities",
    y = "Students' Employment Test",
    title = "Violin Plot of Curricular Activities and Students' Employment Test",
    color = "Curricular Activities"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Curricular Activities** using the GGPlot2 library.

The output of code is as below:



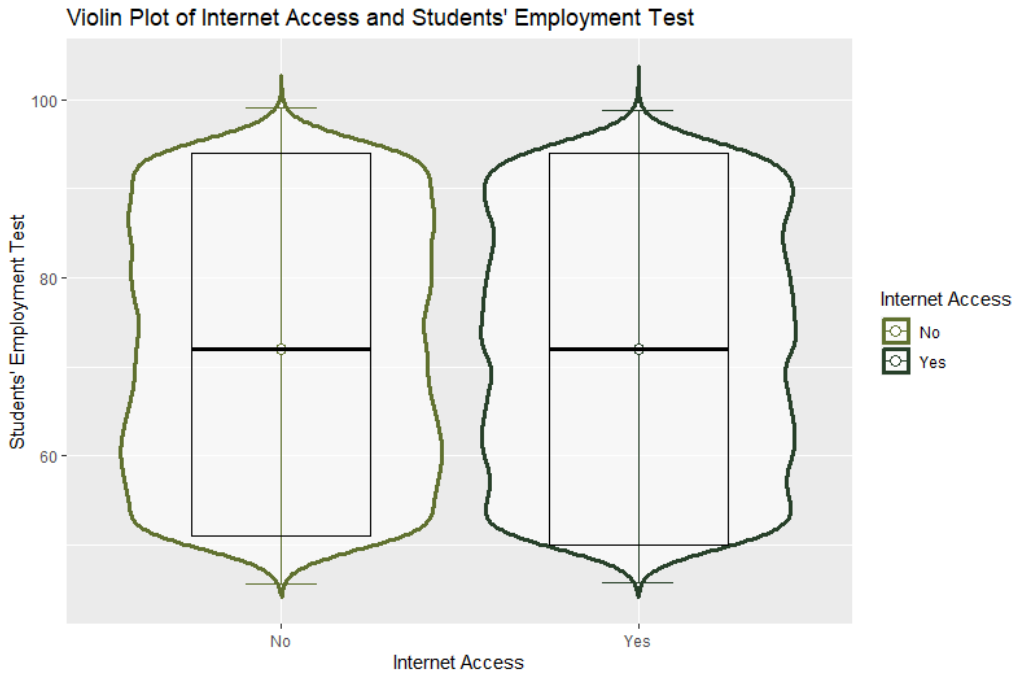

From this violin plot, it is clear that students who have or not have curricular activities does not affect students' employment test grade.

## 6.8 Internet Access

```
# - Internet Access
df_internet <- data.frame(
  student_internet = as.vector(placementData$Internet_Usage),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_internet,
  aes(
    x = stringr::str_to_title(factor(student_internet)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_internet))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#60712F", "#274029")) +
  labs(
    x = "Internet Access",
    y = "Students' Employment Test",
    title = "Violin Plot of Internet Access and Students' Employment Test",
    color = "Internet Access"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Internet Access** using the GGPlot2 library.

The output of code is as below:



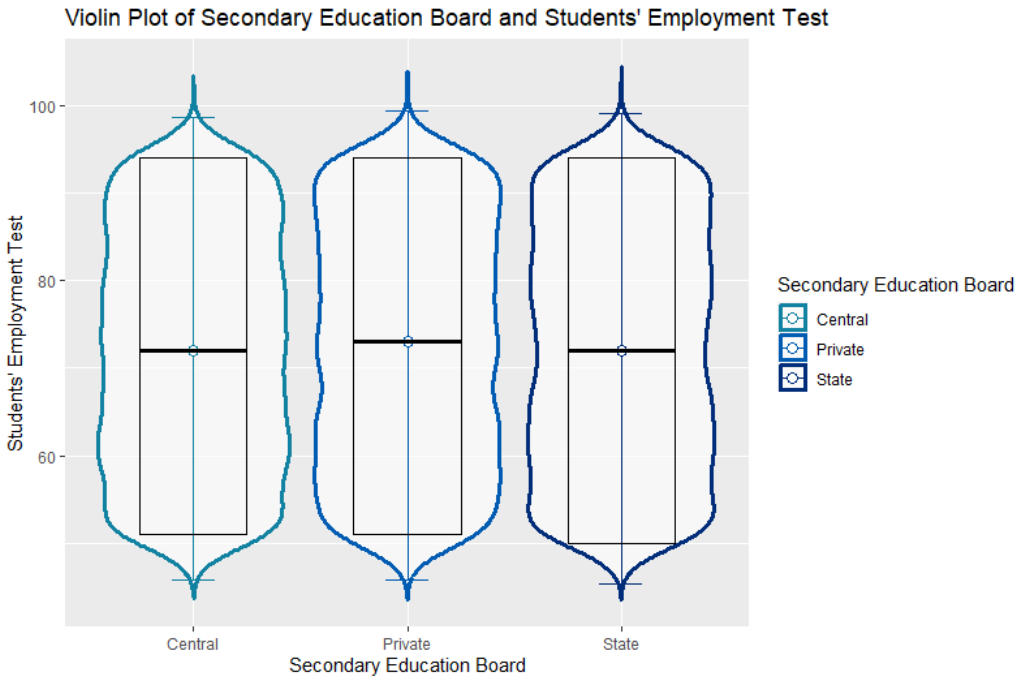

From this violin plot, it is clear that students who have or not have internet access does not affect students' employment test grade.

## 6.9 Secondary Education Board

```
# - Secondary Education Board
df_secondary_board <- data.frame(
  student_secondary_board = as.vector(placementData$Secondary_Education_Board),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_secondary_board,
  aes(
    x = stringr::str_to_title(factor(student_secondary_board)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_secondary_board))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#1282A2", "#005CB3", "#002D7A")) +
  labs(
    x = "Secondary Education Board",
    y = "Students' Employment Test",
    title = "Violin Plot of Secondary Education Board and Students' Employment Test",
    color = "Secondary Education Board"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Secondary Education Board** using the GGPlot2 library.

The output of code is as below:



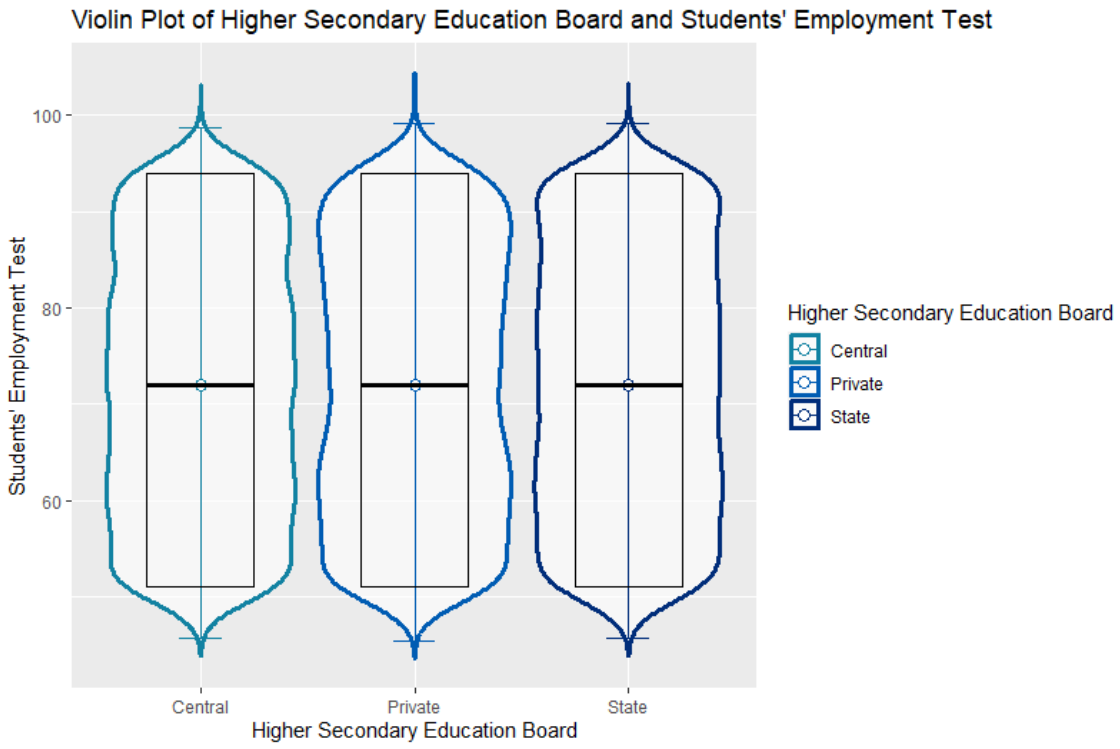

From this violin plot, it is clear that secondary education board does not affect students' employment test grade.

## 6.10 Higher Secondary Education Board

```
# - Higher secondary Education Board
df_higher_secondary_board <- data.frame(
  student_higher_secondary_board = as.vector(placementData$Higher_Secondary_Education_Board),
  student_employ_test = as.vector(placementData$Employment_Test)
)

ggplot(
  df_higher_secondary_board,
  aes(
    x = stringr::str_to_title(factor(student_higher_secondary_board)), y = student_employ_test,
    color = stringr::str_to_title(factor(student_higher_secondary_board))
  )) +
  geom_violin(trim = FALSE, alpha = 0.6, linewidth = 1.2) +
  scale_color_manual(values = c("#1282A2", "#005CB3", "#002D7A")) +
  labs(
    x = "Higher Secondary Education Board",
    y = "Students' Employment Test",
    title = "Violin Plot of Higher Secondary Education Board and Students' Employment Test",
    color = "Higher Secondary Education Board"
  ) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar", width = 0.2) +
  stat_summary(fun = "median", geom = "point", shape = 21, size = 3, fill = "white") +
  stat_summary(fun.data = median_hilow, geom = "crossbar", width = 0.5, fill = NA, color = "black")
```

The code above demonstrates on how to create a violin plot of **Students' Employment Test Grade to Higher Secondary Education Board** using the GGPlot2 library.

The output of code is as below:




From this violin plot, it is clear that higher secondary education board does not affect students' employment test grade.

## 6.11 Analysis Conclusion

Referring the all violin plots, **students' mother tends to not care** about students' employment test whereas **students' father tends to encourage** students to perform well in society while working. This is probably due to father plays an important role in motivating students to work well and perform better to have an easier life in the future.

Other factors like parent job, family support, paid classes, curricular activities, internet access, secondary and higher secondary education board doesn't affect students' employment test because there are not related to the possible question or topics asking in employment test.

# 7.0 Question 6 – What will affect students to have paid classes?

Multiple factors from the imported dataset are used to investigate what will affect students' to have paid classes by creating **frequency polygon and histogram** as data visualization.

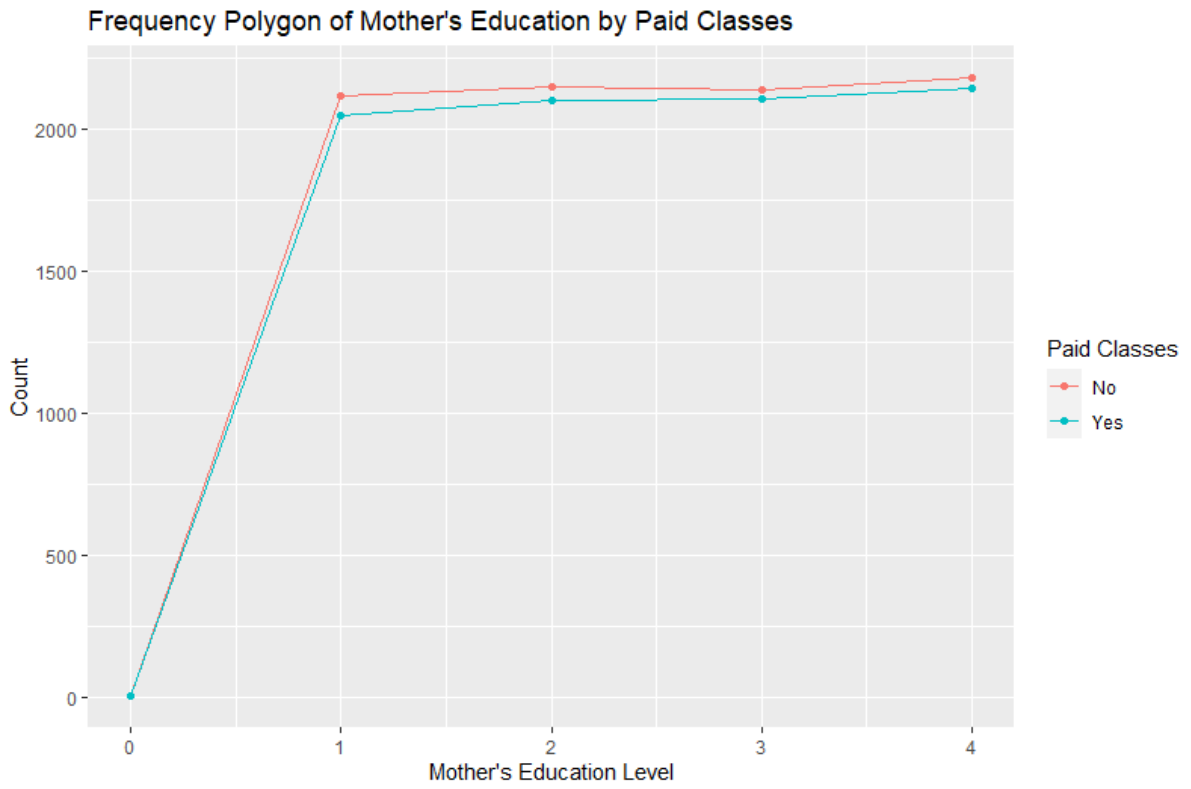
## 7.1 Mother Education

```
# - Mother education (F.Polygon)
df_mom_education <- data.frame(
  mom_education = as.vector(placementData$Mother_Education),
  student_paid = as.vector(placementData$Paid_Classes)
)

freq_table <- df_mom_education %>%
  group_by(mom_education, student_paid) %>%
  summarise(count = n())

ggplot(freq_table, aes(x = mom_education, y = count, color = stringr::str_to_title(student_paid))) +
  geom_line(stat = "identity") +
  geom_point() +
  labs(
    title = "Frequency Polygon of Mother's Education by Paid Classes",
    x = "Mother's Education Level",
    y = "Count",
    color = "Paid Classes"
  )
```

The code above demonstrates on how to create a frequency polygon of **Mother's Education by Paid Classes** using the GGPlot2 library. The **group_by()** function is to group two variables together, in this case, **mom_education** and **student_paid** while the **summarise()** function is to calculate the count of observations in each group and saves it as a new variable, in this case, **count.**

The **geom_line()** function is to add lines to the chart while **stat = "identity"** is to tell that the y-values in the data frames should be used as is. The **geom_point()** is to create points to the graph for each category.

The output of the code is as below:




Before interpreting the chart, do note that:

**0 – No Education**

**1 – Primary Education**

**2 – Secondary Education**

**3 – Degree Education**

**4 – Post Graduate Education**

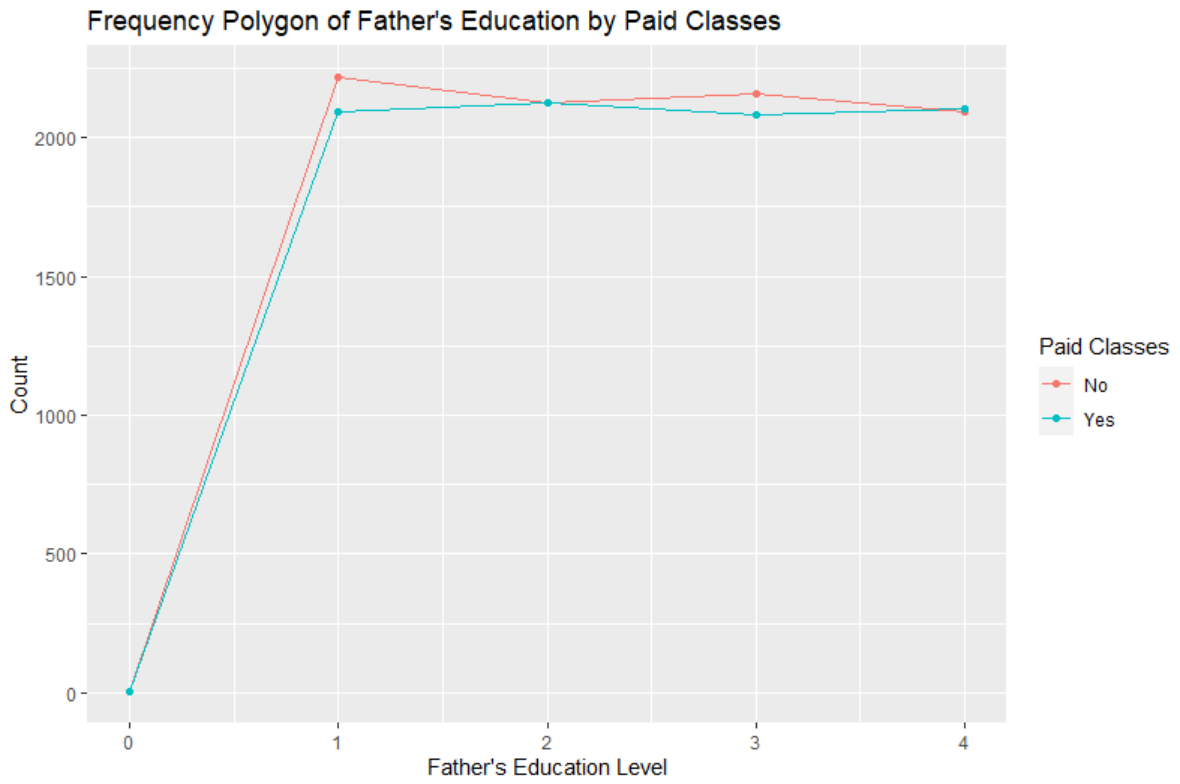
From this frequency polygon, it is clear that students' mother who has at least primary education to post-graduate education are most likely to not let students to have paid classes. It may be due to several factors like insufficient time, expensive fees, poor quality of the paid classes etc.

## 7.2 Father Education

```
# - Father education (F.Polygon)
df_dad_education <- data.frame(
  dad_education = as.vector(placementData$Father_Education),
  student_paid = as.vector(placementData$Paid_Classes)
)

freq_table <- df_dad_education %>%
  group_by(dad_education, student_paid) %>%
  summarise(count = n())

ggplot(freq_table, aes(x = dad_education, y = count, color = stringr::str_to_title(student_paid))) +
  geom_line(stat = "identity") +
  geom_point() +
  labs(
    title = "Frequency Polygon of Father's Education by Paid Classes",
    x = "Father's Education Level",
    y = "Count",
    color = "Paid Classes"
  )
```

The code above demonstrates on how to create a frequency polygon of **Father's Education by Paid Classes** using the GGPlot2 library.

The output of the code is as below:




Before interpreting the chart, do note that:

**0 – No Education**

**1 – Primary Education**

**2 – Secondary Education**

**3 – Degree Education**

**4 – Post Graduate Education**

From this frequency polygon, it is clear that students' father who has primary education and degree education are most likely to not let students to have paid classes. It may be due to several factors like insufficient time, expensive fees, poor quality of the paid classes etc.

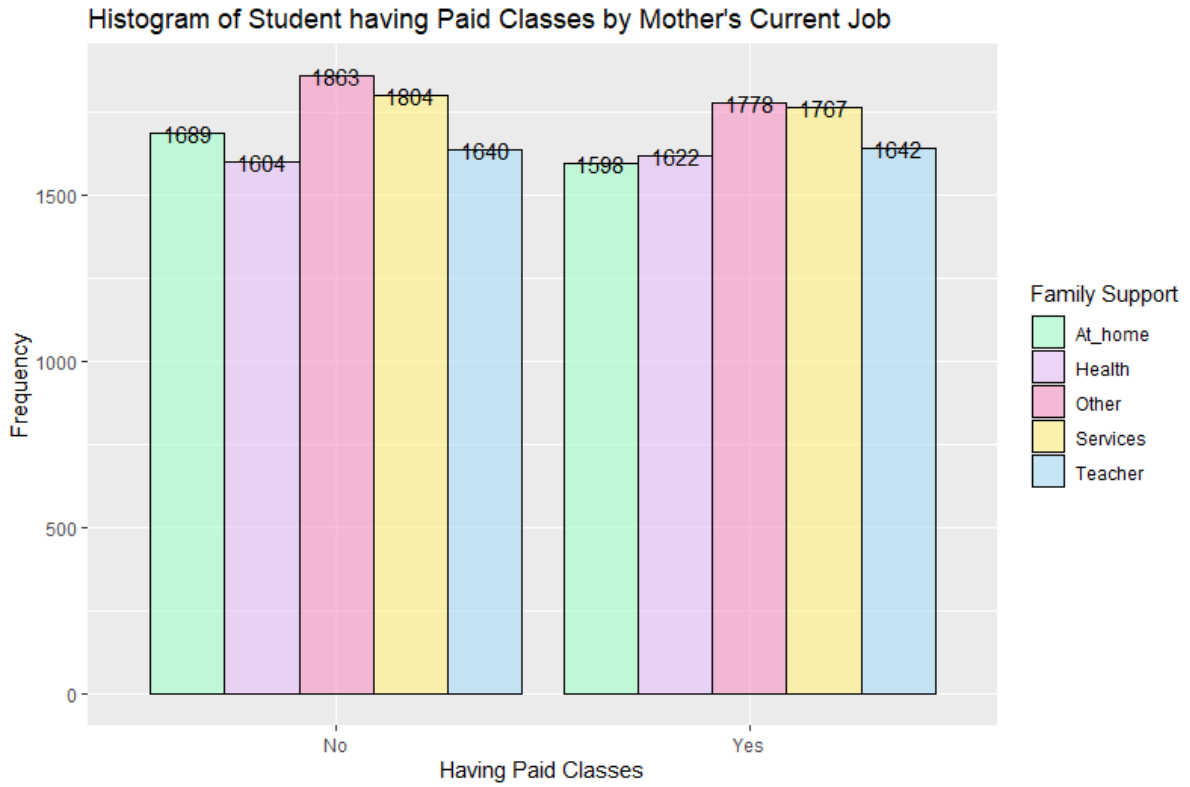
## 7.3 Mother Current Job

```
# - Mother current job (Histogram)
df_mom_job <- data.frame(
  mom_job = as.vector(placementData$Mother_Current_Job),
  student_paid = as.vector(placementData$Paid_Classes)
)

ggplot(df_mom_job, aes(x = stringr::str_to_title(student_paid), fill = stringr::str_to_title(mom_job))) +
  geom_histogram(
    alpha = 0.6,
    position = "dodge",
    stat = "count",
    color = "black"
  ) +
  scale_fill_manual(values = c("#A4FFC9", "#E4C1F9", "#F694C1", "#FFF27E", "#A9DEF9")) +
  labs(
    x = "Having Paid Classes",
    y = "Frequency",
    title = "Histogram of Student having Paid Classes by Mother's Current Job",
    fill = "Family Support"
  ) +
  geom_text(
    aes(label = after_stat(count)),
    stat = "count",
    position = position_dodge(width = 0.9),
    vjust = 0.5,
    size = 4
  )
```

The code above demonstrates on how to create a histogram of **Mother's Current Job to Paid Classes** using GGPlot2 library. The **geom_histogram()** function is used to generate a histogram. Inside this function, **alpha** is to set the transparency level of the bars, **position = "dodge"** is to make the bar side-by-side, **stat = "count"** means the height of the bars will represent the count of observations in each bin and the **color** is to make the border of the bars black.

The **geom_text()** function is to create text labels in the plot. Inside this function, **label = after_stat(count)** means that the label for each bin will be the frequency in that bin, **stat = "count"** means the labels will correspond to the frequency in each bin, **position_dodge()** function is to offset the text labels and **vjust()** is to set the vertical justification of the text labels.

The output of the code is as below:



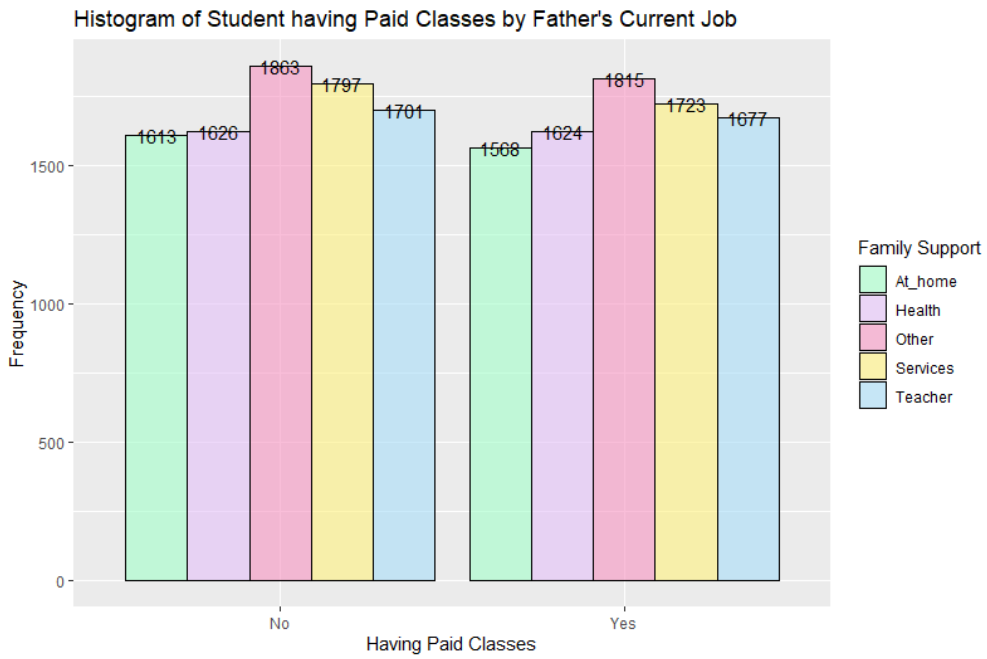

From this histogram, it is clear that students' mother who is currently a housewife or working in services and other industries are most likely to not let students to have paid classes. Probably it is due to lack of transportation, unsuitable working hours to fetch student to paid classes etc.

## 7.4 Father Current Job

```
# - Father current job (Histogram)
df_dad_job <- data.frame(
  dad_job = as.vector(placementData$Father_Current_Job),
  student_paid = as.vector(placementData$Paid_Classes)
)

ggplot(df_dad_job, aes(x = stringr::str_to_title(student_paid), fill = stringr::str_to_title(dad_job))) +
  geom_histogram(
    alpha = 0.6,
    position = "dodge",
    stat = "count",
    color = "black"
  ) +
  scale_fill_manual(values = c("#A4FFC9", "#E4C1F9", "#F694C1", "#FFF27E", "#A9DEF9")) +
  labs(
    x = "Having Paid Classes",
    y = "Frequency",
    title = "Histogram of Student having Paid Classes by Father's Current Job",
    fill = "Family Support"
  ) +
  geom_text(
    aes(label = after_stat(count)),
    stat = "count",
    position = position_dodge(width = 0.9),
    vjust = 0.5,
    size = 4
  )
```

The code above demonstrates on how to create a histogram of **Father's Current Job to Paid Classes** using GGPlot2 library.

The output of the code is as below:



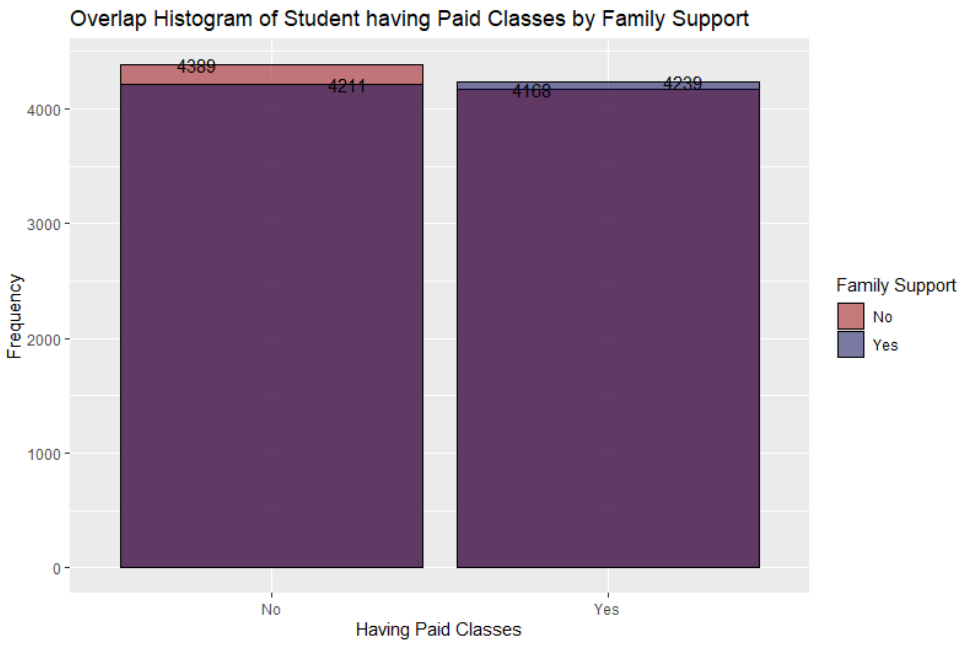

From this histogram, it is clear that students' father who is currently working in services and other industries are most likely to not let students to have paid classes. Probably it is due to lack of transportation, unsuitable working hours to fetch student to paid classes etc.

## 7.5 Family Support

```
# - Family support (Histogram)
df_famsup <- data.frame(
  student_famsup = as.vector(placementData$Family_Support),
  student_paid = as.vector(placementData$Paid_Classes)
)

ggplot(df_famsup, aes(x = stringr::str_to_title(student_paid), fill = stringr::str_to_title(student_famsup))) +
  geom_histogram(
    alpha = 0.5,
    position = "identity",
    stat = "count",
    color = "black"
  ) +
  scale_fill_manual(values = c("#970005", "#000052")) +
  labs(
    x = "Having Paid Classes",
    y = "Frequency",
    title = "Overlap Histogram of Student having Paid Classes by Family Support",
    fill = "Family Support"
  ) +
  geom_text(
    aes(label = after_stat(count)),
    stat = "count",
    position = position_dodge(width = 0.9),
    vjust = 0.5,
    size = 4
  )
```

The code above demonstrates on how to create an overlapped histogram of **Family Support to Paid Classes** using GGPlot2 library. Instead of setting the position to **dodge** in the **geom_histogram()** function, it can be set to **identity** to create an overlapped histogram.

The output of the code is as below:



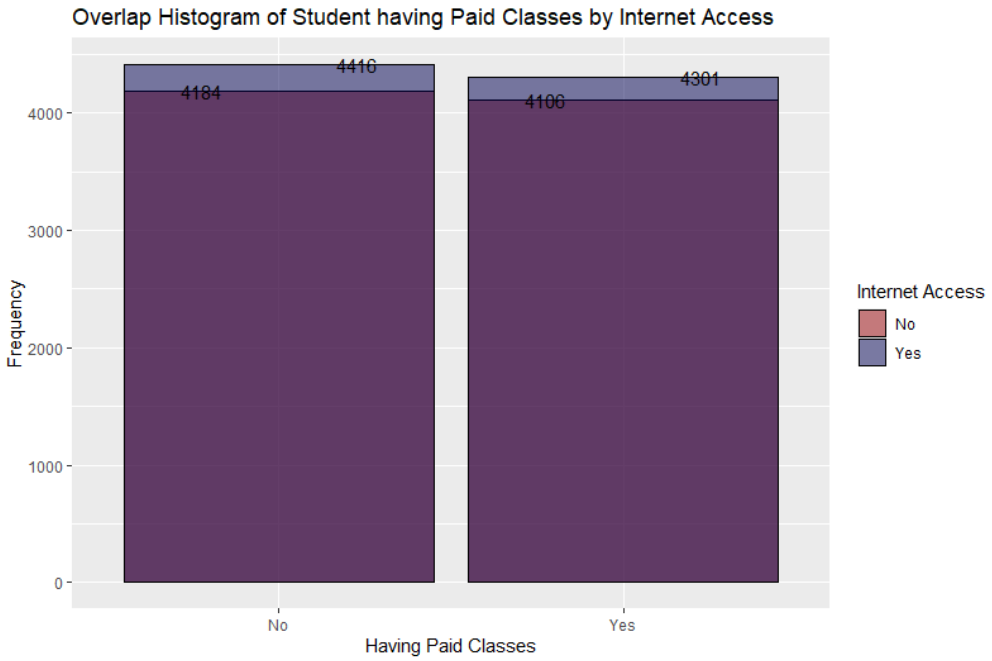

From this overlapped histogram, it is clear that students who don't have family support doesn't go for paid classes. This is logical because it can be financial issues which disallow students to go for paid classes.

## 7.6 Internet Access

```
# - Internet access (Histogram)
df_internet <- data.frame(
  student_internet = as.vector(placementData$Internet_Usage),
  student_paid = as.vector(placementData$Paid_Classes)
)

ggplot(df_internet, aes(x = stringr::str_to_title(student_paid), fill = stringr::str_to_title(student_internet))) +
  geom_histogram(alpha = 0.5, position = "identity", stat = "count", color = "black") +
  scale_fill_manual(values = c("#970005", "#000052")) +
  labs(
    x = "Having Paid Classes",
    y = "Frequency",
    title = "Overlap Histogram of Student having Paid Classes by Internet Access",
    fill = "Internet Access"
  ) +
  geom_text(
    aes(label = after_stat(count)),
    stat = "count",
    position = position_dodge(width = 0.9),
    vjust = 0.5,
    size = 4
  )
```

The code above demonstrates on how to create an overlapped histogram of **Internet Access to Paid Classes** using GGPlot2 library.

The output of the code is as below:



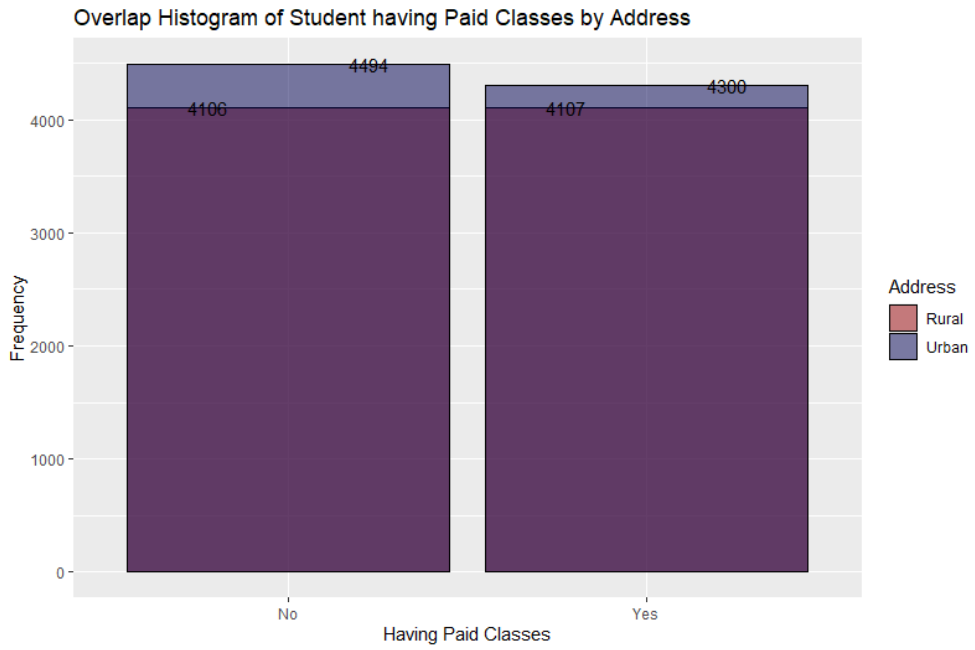

From this overlapped histogram, it is clear that students who doesn't have internet access doesn't affect them having paid classes. However, there are more students who have internet access don't go for paid classes compared to those who have internet access and go for paid classes. This is probably due to students have access to the Internet to search for information they need.

## 7.7 Address

```r
# - Address (Histogram)
df_address <- data.frame(
  student_address = as.vector(placementData$Address),
  student_paid = as.vector(placementData$Paid_Classes)
)

ggplot(df_address, aes(x = stringr::str_to_title(student_paid), fill = stringr::str_to_title(student_address))) +
  geom_histogram(alpha = 0.5, position = "identity", stat = "count", color = "black") +
  scale_fill_manual(values = c("#970005", "#000052")) +
  labs(
    x = "Having Paid Classes",
    y = "Frequency",
    title = "Overlap Histogram of Student having Paid Classes by Address",
    fill = "Address"
  ) +
  geom_text(
    aes(label = after_stat(count)),
    stat = "count",
    position = position_dodge(width = 0.9),
    vjust = 0.5,
    size = 4
  )
```

The code above demonstrates on how to create an overlapped histogram of **Students' Address to Paid Classes** using GGPlot2 library.

The output of the code is as below:



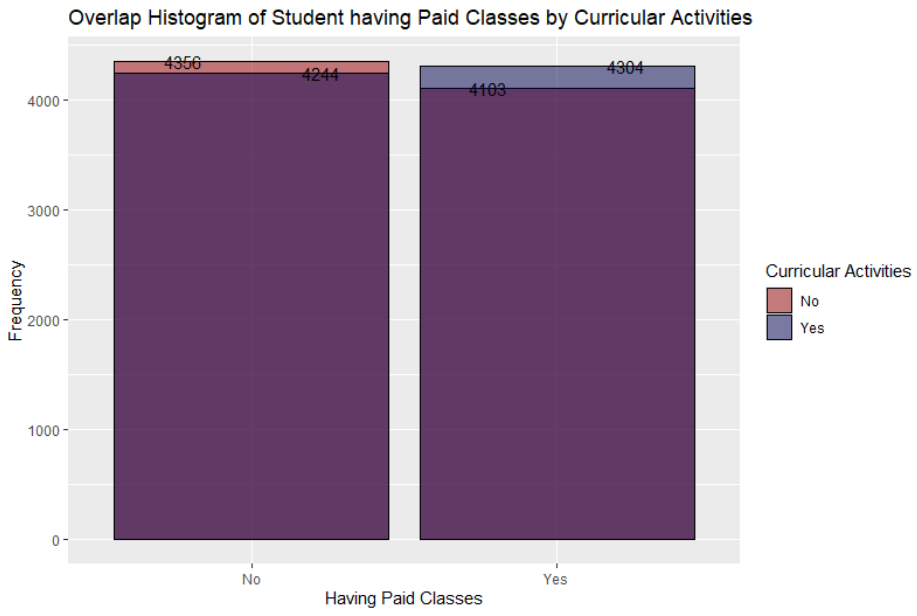

From this overlapped histogram, it is clear that students who lived in rural areas doesn't affect them to have paid classes. On the other hand, the number of students who lived in urban areas and don't have paid classes are greater than students who lived in urban areas and have paid classes. Traffic congestion and time taken to travel from one place to another place in urban areas might be the cause of students who lived in urban areas and not having paid classes.

## 7.8 Curricular Activities

```
# - Curricular activities (Histogram)
df_activities <- data.frame(
  student_activities = as.vector(placementData$Curricular_Activities),
  student_paid = as.vector(placementData$Paid_Classes)
)

ggplot(
  df_activities,
  aes(
    x = stringr::str_to_title(student_paid),
    fill = stringr::str_to_title(student_activities)
  )) +
  geom_histogram(alpha = 0.5, position = "identity", stat = "count", color = "black") +
  scale_fill_manual(values = c("#970005", "#000052")) +
  labs(
    x = "Having Paid Classes",
    y = "Frequency",
    title = "Overlap Histogram of Student having Paid Classes by Curricular Activities",
    fill = "Curricular Activities"
  ) +
  geom_text(
    aes(label = after_stat(count)),
    stat = "count",
    position = position_dodge(width = 0.9),
    vjust = 0.5,
    size = 4
  )
```

The code above demonstrates on how to create an overlapped histogram of **Students' Address to Paid Classes** using GGPlot2 library.

The output of the code is as below:




From this overlapped histogram, it is clear that students who have curricular activities are more likely to go for paid classes whereas students who don't have curricular activities are more likely to not go for paid classes.

## 7.9 Analysis Conclusion

Based on the frequency polygons and histograms, there are multiple conclusions can be deduced.

Firstly, **students' mother who has education at least primary to post graduate education and students' father who has primary and degree education** discourage students to go for paid classes.

Secondly, **students' mother who is currently a housewife or working in services and other industries and student's father who is currently working in services and other industries** also discourage students to go for paid classes.

Thirdly, **students who don't have family support** doesn't go for paid classes as this is logical because it can be financial issues which disallow students to go for paid classes while **students who have internet access** also don't go for paid classes.

Next, **students that students who lived in rural areas** doesn't affect them to have paid classes. On the other hand, the number of **students who lived in urban areas and don't have paid classes** are **greater** than **students who lived in urban areas and have paid classes**.

Finally, **students who have curricular activities are more likely to go for paid classes** whereas **students who don't have curricular activities are more likely to not go for paid classes**.

# 8.0 Question 7 – What will affect students' salary?

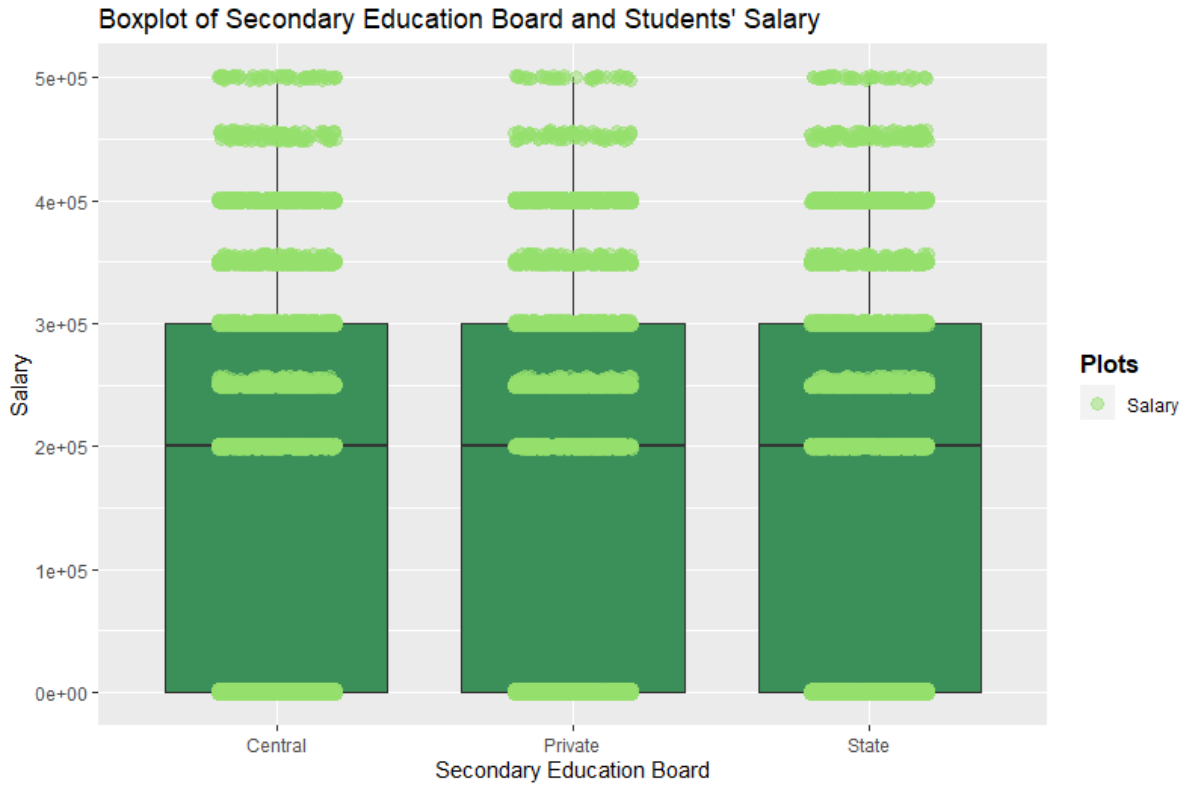
Multiple factors from the imported dataset are used to investigate what will affect students' salary by creating **scatterplot and other previous chart type** as data visualization.

## 8.1 Secondary Education Board

```
# - Secondary eduboard (Boxplot)
secondary_edu_table <- table(placementData$Secondary_Education_Board)
secondary_edu_name <- as.vector(names(secondary_edu_table))

df_secondary <- data.frame(
  secondary_x = secondary_edu_name,
  secondary_y = as.vector(placementData$Salary)
)

ggplot(df_secondary, aes(x = secondary_x, y = secondary_y)) +
  geom_boxplot(fill = "#3B905A") +
  geom_jitter(aes(color = "Salary"), width = 0.2, alpha = 0.5, size = 3) +
  labs(
    x = "Secondary Education Board",
    y = "Salary",
    title = "Boxplot of Secondary Education Board and Students' Salary",
    color = "Plots"
  ) +
  scale_color_manual(values = c("#95E06C")) +
  theme(legend.title = element_text(face = "bold", size = 12))
```

The code above demonstrates on how to create a boxplot of **Secondary Education Board to Students' Salary** using the GGPlot2 library. The **geom_boxplot()** function is to create boxplot with a custom colour for the boxes using **fill()** function. The **geom_jitter()** is to create a scatterplot with random jittering while inside this function, **aes(color = "Salary")** sets the color of the points to be plotted to "Salary". The **width** is to set the width of the jittering to be 0.2 units which means the points will be randomly scattered within a range of 0.1 units +/- from their actual x-axis value. The **alpha** is to control the transparency of the points and **size** is to control how big each point will be.

The output of the code is as below:



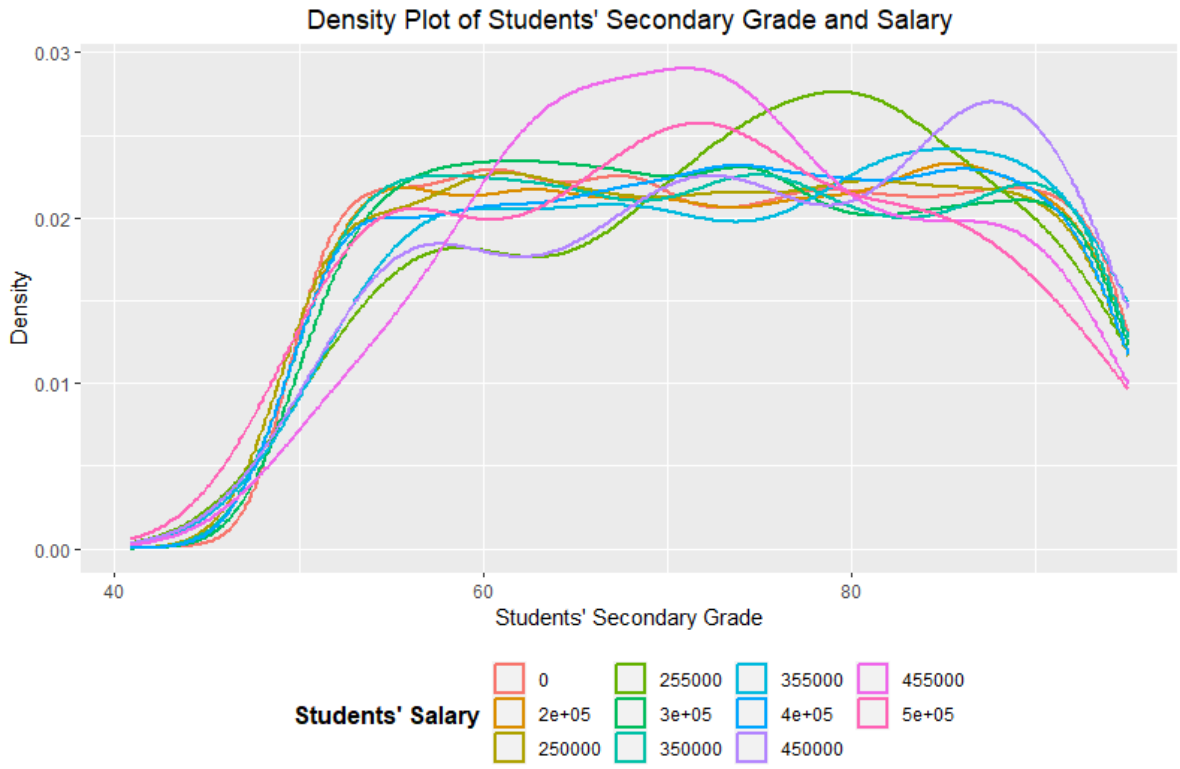

From this boxplot, it is clear that secondary education board doesn't affect students' salary.

## 8.2 Secondary Grade

```
# - Secondary grade (Density Plot)
df_secondary_grade <- data.frame(
  student_secondary_grade = as.vector(placementData$Secondary_Grade_Percentage),
  student_salary = as.vector(placementData$Salary)
)

ggplot(
    df_secondary_grade,
    aes(x = student_secondary_grade, y = after_stat(density), color = factor(student_salary))
  ) +
  geom_density(alpha = 0.6, linewidth = 1) +
  labs(
    x = "Students' Secondary Grade", y = "Density",
    title = "Density Plot of Students' Secondary Grade and Salary"
  ) +
  scale_color_discrete(name = "Students' Salary") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Secondary School Grade to Students' Salary** using the GGPlot2 library.

The output of the code is as below:



From this density plot, it is clear that there are multiple peaks for students who have a salary of 255K, 450K, 455K, and 500K. Firstly, the density line peak for 255K is around students who achieve between 70 to 85 marks in their secondary school. Secondly, the density line peak for 450K is around students who achieve 80 marks or above in their secondary school. Thirdly, the density line peak for 455K is around students who achieve between 60 to 80 marks while for 500K is around students who achieve between 65 to 75 marks.

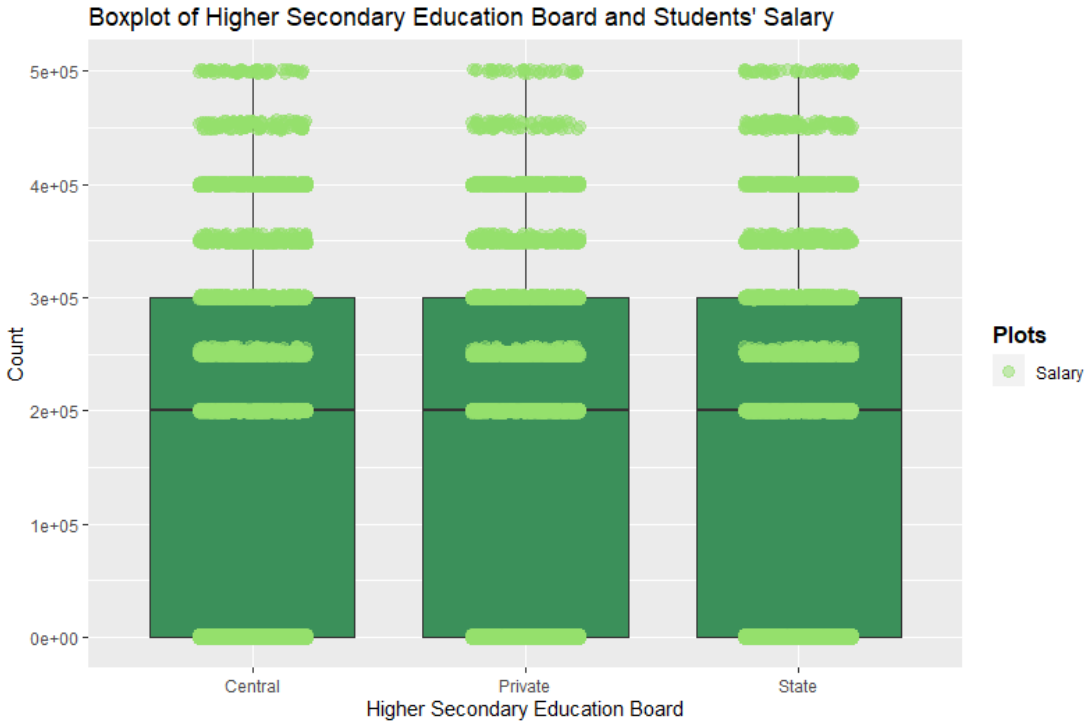
Assuming that there are no other external factor interfering with students' salary, a conclusion can be made. Students' secondary school grade will affect students' salary because the higher the students' secondary school grade, the higher the students' salary is.

## 8.3 Higher Secondary Education Board

```
# - Higher secondary eduboard (Boxplot)
higher_secondary_edu_table <- table(placementData$Higher_Secondary_Education_Board)
higher_secondary_edu_name <- as.vector(names(higher_secondary_edu_table))

df_higher_secondary <- data.frame(
  higher_secondary_x = higher_secondary_edu_name,
  higher_secondary_y = as.vector(placementData$Salary)
)

ggplot(df_higher_secondary, aes(x = higher_secondary_x, y = higher_secondary_y)) +
  geom_boxplot(fill = "#3B905A") +
  geom_jitter(aes(color = "Salary"), width = 0.2, alpha = 0.5, size = 3) +
  labs(
    x = "Higher Secondary Education Board",
    y = "Count",
    title = "Boxplot of Higher Secondary Education Board and Students' Salary",
    color = "Plots"
  ) +
  scale_color_manual(values = c("#95E06C")) +
  theme(legend.title = element_text(face = "bold", size = 12))
```

The code above demonstrates on how to create a boxplot of **Higher Secondary Education Board to Students' Salary** using the GGPlot2 library.

The output of the code is as below:



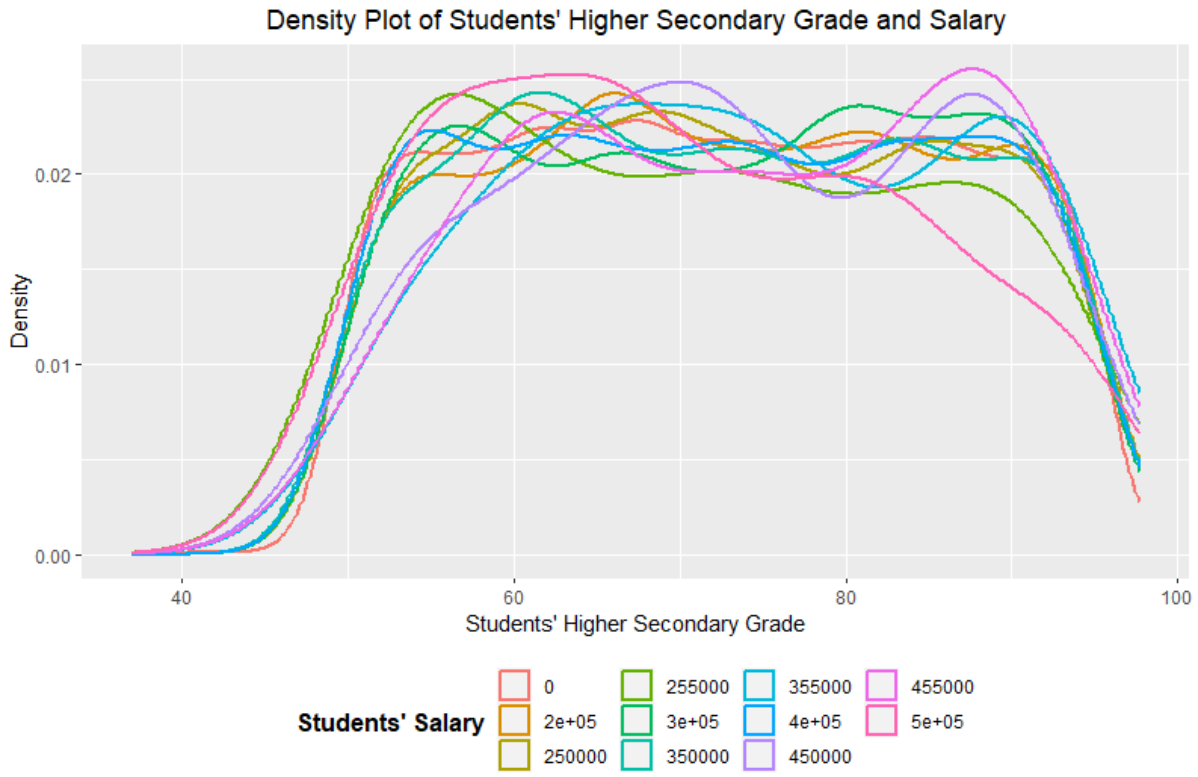

From this boxplot, it is clear that higher secondary education board doesn't affect students' salary.

## 8.4 Higher Secondary Grade

```
# - Higher secondary grade (Density Plot)
df_higher_secondary_grade <- data.frame(
  student_higher_secondary_grade = as.vector(placementData$Higher_Secondary_Grade_Percentage),
  student_salary = as.vector(placementData$Salary)
)

ggplot(
    df_higher_secondary_grade,
    aes(x = student_higher_secondary_grade, y = after_stat(density), color = factor(student_salary))
  ) +
  geom_density(alpha = 0.6, linewidth = 1) +
  labs(
    x = "Students' Higher Secondary Grade", y = "Density",
    title = "Density Plot of Students' Higher Secondary Grade and Salary"
  ) +
  scale_color_discrete(name = "Students' Salary") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Higher Secondary School Grade to Students' Salary** using the GGPlot2 library.

The output of the code is as below:



From this density plot, it is clear that there is a significant peak for students who have a salary of 455K and a drop for students who have a salary of 500K respectively. Both density line peak and drop started from students who achieve 80 marks or above in their higher secondary school.

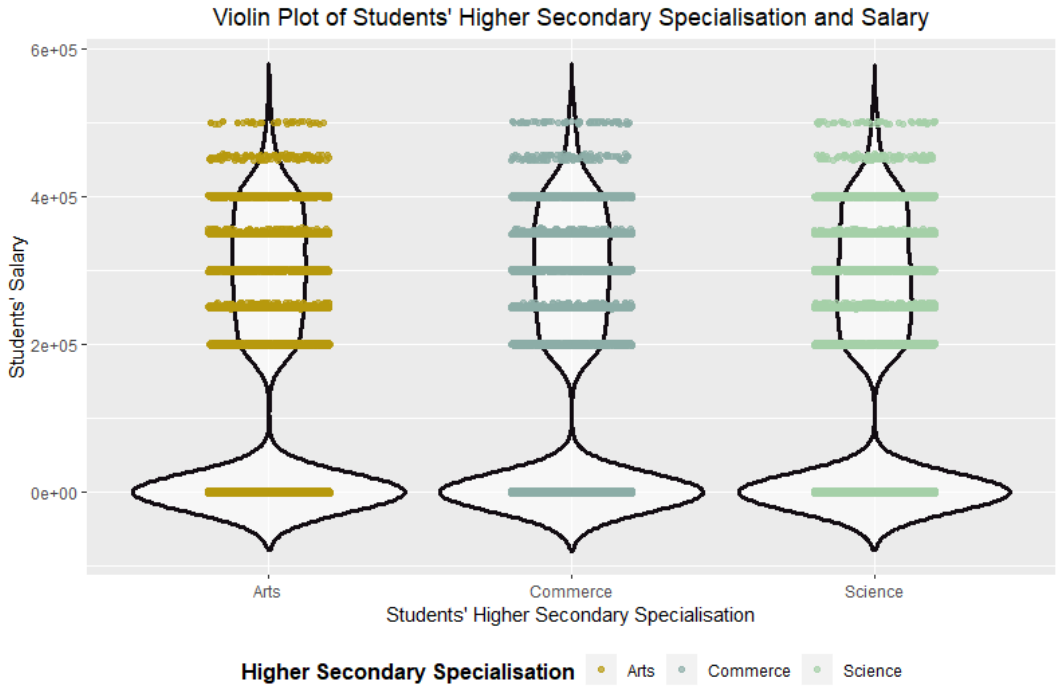
This suggests that while higher grade in higher secondary school is associated with higher salaries up to a certain point, there may be a point of diminishing returns where achieving extremely high grades does not necessarily lead to correspondingly high salaries.

## 8.5 Higher Secondary Specialisation

```
# - Higher secondary specialisation (Violin Plot)
df_higher_secondary_specialisation <- data.frame(
  student_higher_secondary_specialisation = as.vector(placementData$Higher_Secondary_Specialism),
  student_salary = as.vector(placementData$Salary)
)

ggplot(
    df_higher_secondary_specialisation,
    aes(x = factor(student_higher_secondary_specialisation), y = student_salary)
  ) +
  geom_violin(trim = FALSE, alpha = 0.6, color = "#110B11", linewidth = 1.2) +
  geom_jitter(aes(color = student_higher_secondary_specialisation), width = 0.2, alpha = 0.7) +
  labs(
    x = "Students' Higher Secondary Specialisation",
    y = "Students' Salary",
    title = "Violin Plot of Students' Higher Secondary Specialisation and Salary",
    color = "Higher Secondary Specialisation"
  ) +
  scale_color_manual(values = c("#B7990D", "#8CADA7", "#A5D0A8")) +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a violin plot of **Higher Secondary Specialisation to Students' Salary** using the GGPlot2 library.

The output of the code is as below:



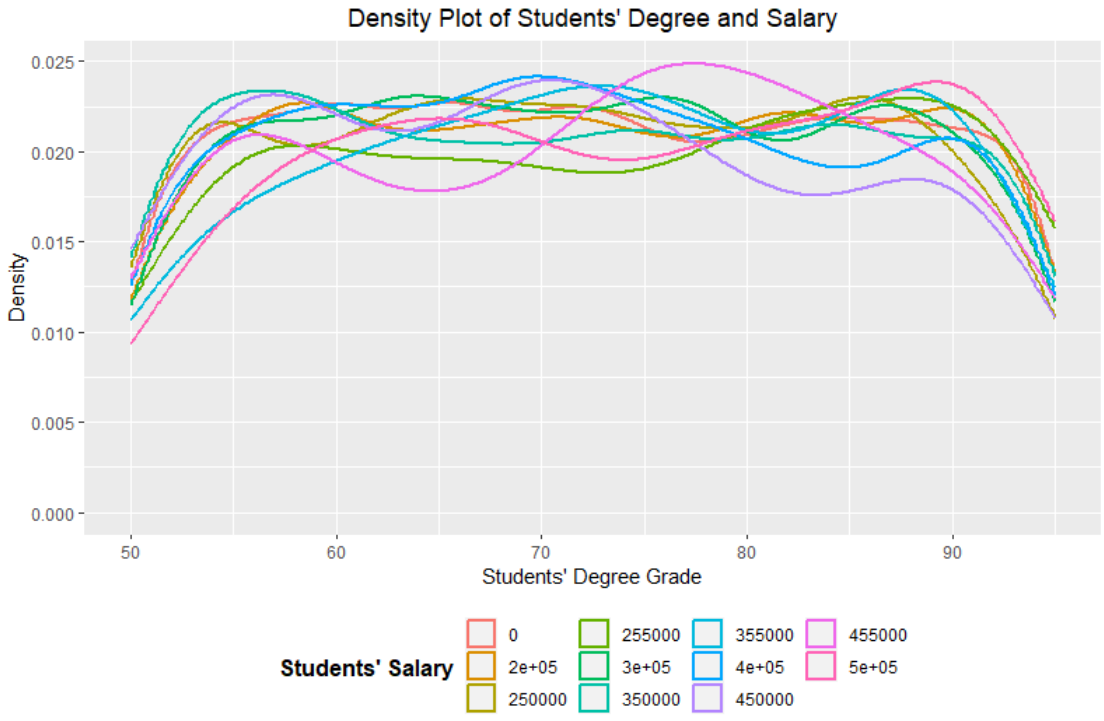

From this violin plot, it is clear that higher secondary specialisation doesn't affect students' salary.

## 8.6 Degree Grade

```
# - Degree grade (Density Plot)
df_degree_grade <- data.frame(
  student_degree_grade = as.vector(placementData$Degree_Grade_Percentage),
  student_salary = as.vector(placementData$Salary)
)

ggplot(df_degree_grade, aes(x = student_degree_grade, y = after_stat(density), color = factor(student_salary))) +
  geom_density(alpha = 0.6, linewidth = 1) +
  labs(x = "Students' Degree Grade", y = "Density", title = "Density Plot of Students' Degree and Salary") +
  scale_color_discrete(name = "Students' Salary") +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a density plot of **Degree Grade to Students' Salary** using the GGPlot2 library.

The output of the code is as below:




From this density plot, it is clear that the density line for students who have a salary of 455K has a drop for students who achieve grade between 60 to 70 marks and has a peak for students who achieve grade between 75 to 85 marks.

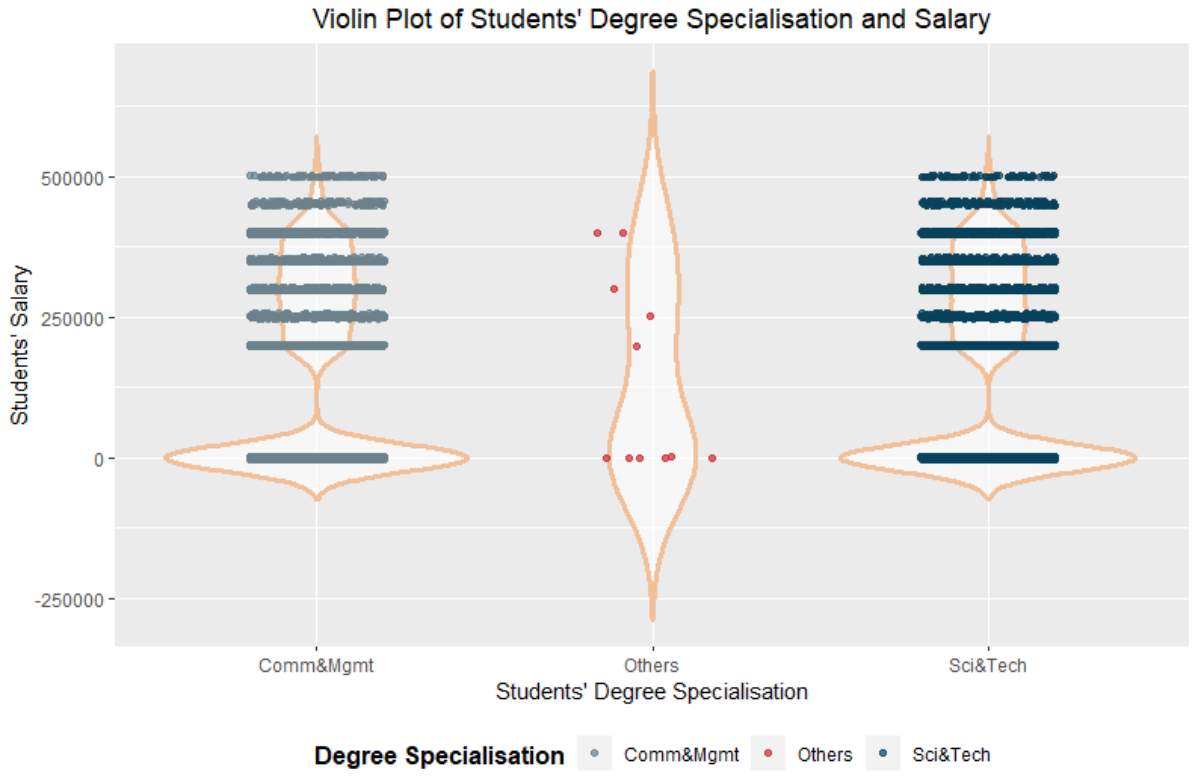
This is obvious and logical that the organization prefers students who achieve high degree marks rather than those who don't.

## 8.7 Degree Specialisation

```r
# - Degree specialization (Violin Plot)
df_degree_specialisation <- data.frame(
  student_degree_specialisation = as.vector(placementData$Degree_Specialism),
  student_salary = as.vector(placementData$Salary)
)

ggplot(df_degree_specialisation, aes(x = factor(student_degree_specialisation), y = student_salary)) +
  geom_violin(trim = FALSE, alpha = 0.6, color = "#F1BF98", linewidth = 1.2) +
  geom_jitter(aes(color = student_degree_specialisation), width = 0.2, alpha = 0.7) +
  labs(
    x = "Students' Degree Specialisation",
    y = "Students' Salary",
    title = "Violin Plot of Students' Degree Specialisation and Salary",
    color = "Degree Specialisation"
  ) +
  scale_color_manual(values = c("#6B818C", "#CC2936", "#08415C")) +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a violin plot of **Degree Specialisation to Students' Salary** using the GGPlot2 library.

The output of the code is as below:




From this violin plot, it is clear that students' degree specialisation doesn't affect the students' salary because students who studied other degree specialisation still get a salary.

## 8.8 Master Grade

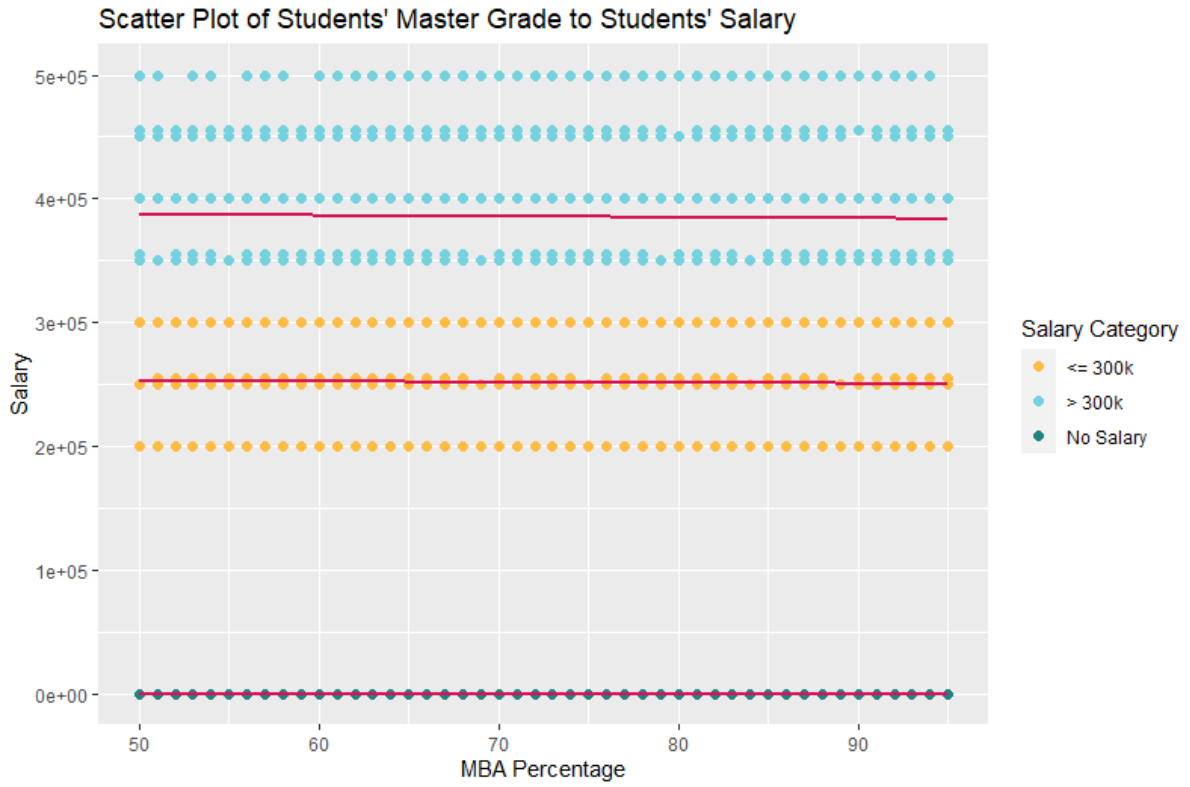
```
# - Master grade (Scatterplot)
df_master <- data.frame(
  student_mba = c(placementData$Master_Grade_Percentage),
  student_salary = c(placementData$Salary),
  # Create a categorical variable based on the salary
  salary_category = cut(placementData$Salary, breaks = c(0, 300000, Inf), labels = c("Category A", "Category B"))
)

# Create the plot
ggplot(df_master, aes(x = student_mba, y = student_salary, color = salary_category)) +
  geom_point(size = 2, shape = 16) +
  geom_smooth(aes(group = salary_category), method = "lm", se = FALSE, color = "#D81159") +
  scale_color_manual(values = c("#FFBC42", "#73D2DE"),
                     name = "Salary Category",
                     labels = c("<= 300k", "> 300k", "No Salary"),
                     drop = FALSE,
                     na.value = "#218380") +
  labs(title = "Scatter Plot of Students' Master Grade to Students' Salary", x = "MBA Percentage", y = "Salary")
```

The code above demonstrates on how to create a scatterplot of **Master Grade to Students' Salary** using the GGPlot2 library. A new variable, **salary_category**, is declared to create a category based on the salary which divides into **"No Salary", "Below 300K", and "300K or above"** using the **cut()** function along with **breaks** and **labels** parameters.

The **geom_smooth()** function is to set **salary_category** in order to group the data points by salary category to create regression lines. The **method = "lm"** is to indicate that a linear model should be used and **se = FALSE** to remove the shaded confidence interval around the regression line.

The **drop = FALSE** is to prevent empty factor levels from being dropped from the legend, and the **na.value** argument is set to a color value to specify the color for missing values.

The output of the code is as below:



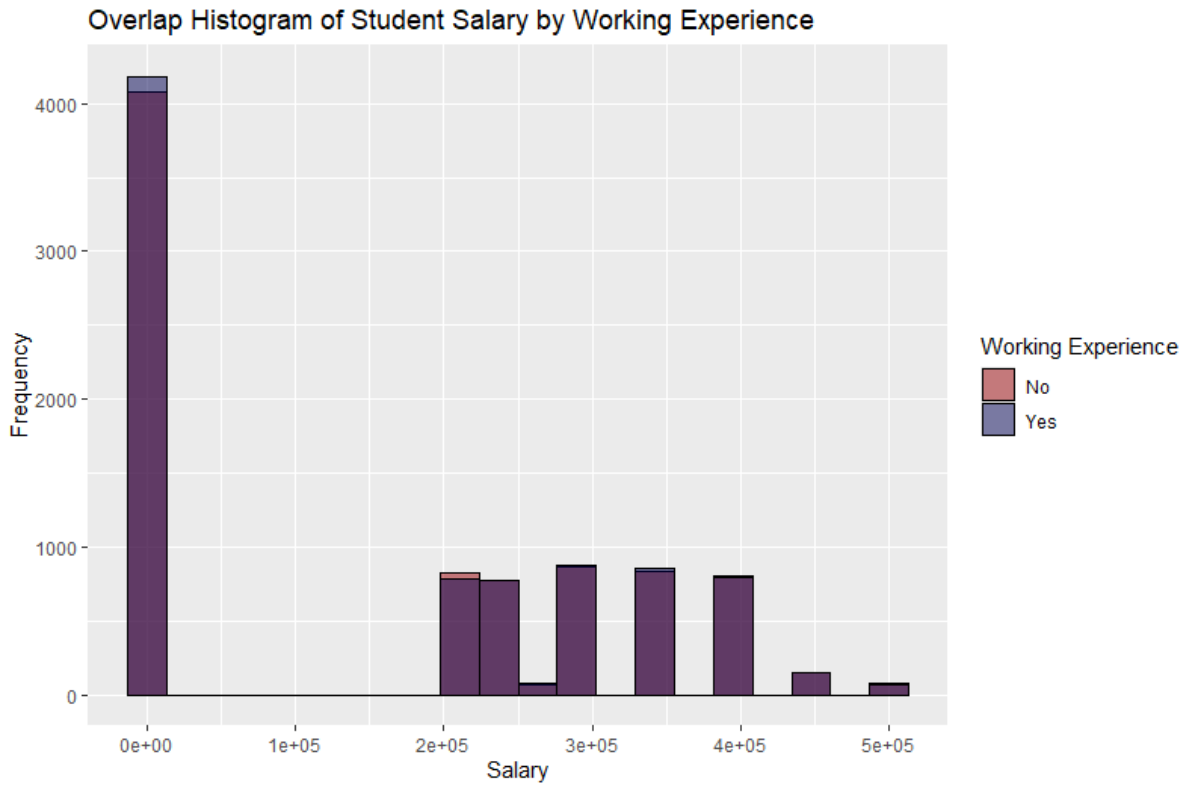

From this scatterplot, it is clear that all three regression lines are horizontal which means students' master grade doesn't affect students' salary as there are no relationship.

## 8.9 Work Experiences

```
# - Work experiences (Histogram)
df_workex <- data.frame(
  student_workex = as.vector(placementData$Working_Experience),
  student_salary = as.vector(placementData$Salary)
)
# Create an overlap histogram
ggplot(df_workex, aes(x = student_salary, fill = student_workex)) +
  geom_histogram(alpha = 0.5, position = "identity", bins = 20, color = "black") +
  scale_fill_manual(values = c("#970005", "#000052")) +
  labs(
    x = "Salary",
    y = "Frequency",
    title = "Overlap Histogram of Student Salary by Working Experience",
    fill = "Working Experience"
  )
```

The code above demonstrates on how to create a histogram of **Students' Working Experience to Students' Salary** using the GGPlot2 library.

The output of the code is as below:



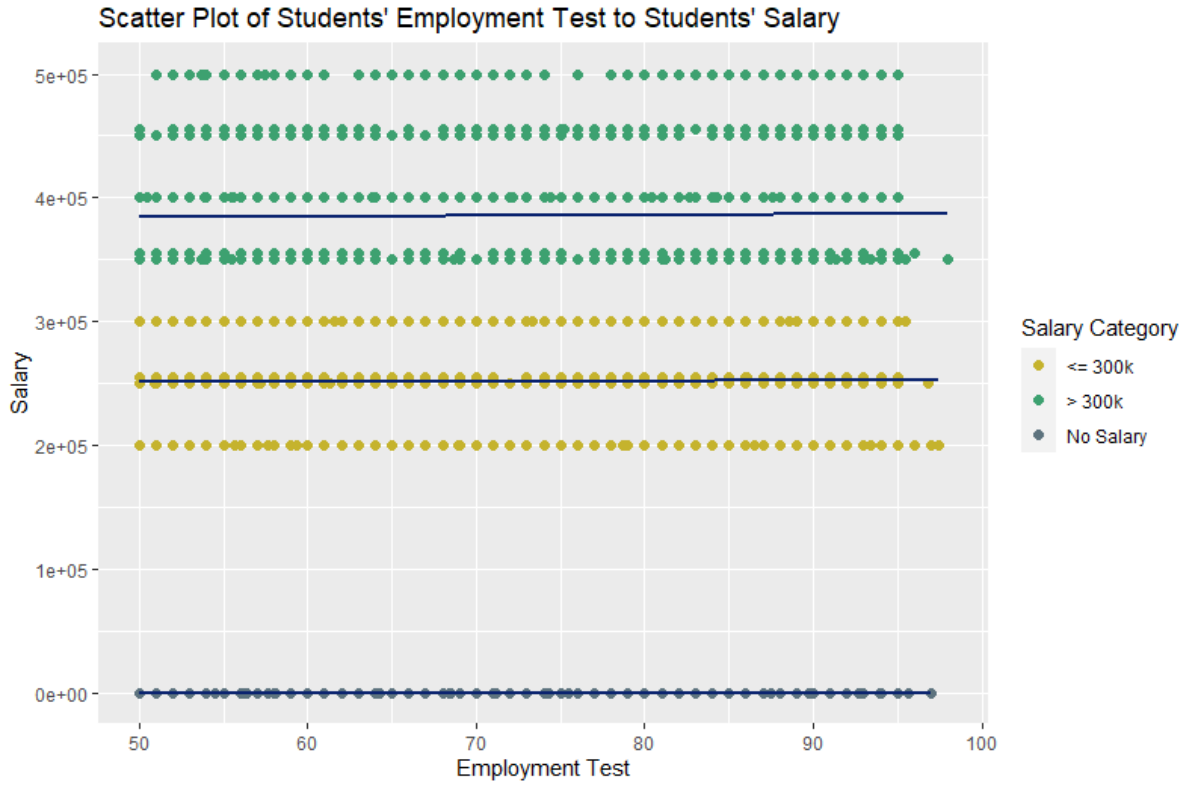

From this histogram, it is clear that students' working experience doesn't affect students' salary.

## 8.10 Employment Test

```
# - Employment test (Scatterplot)
df_employtest <- data.frame(
  student_employtest = c(placementData$Employment_Test),
  student_salary = c(placementData$Salary),
  salary_category = cut(placementData$Salary, breaks = c(0, 300000, Inf), labels = c("Category A", "Category B"))
)

ggplot(df_employtest, aes(x = student_employtest, y = student_salary, color = salary_category)) +
  geom_point(size = 2, shape = 16) +
  geom_smooth(aes(group = salary_category), method = "lm", se = FALSE, color = "#0A236F") +
  scale_color_manual(values = c("#C5B32D", "#3DA170"),
                     name = "Salary Category",
                     labels = c("<= 300k", "> 300k", "No Salary"),
                     drop = FALSE,
                     na.value = "#5D737E") +
  labs(title = "Scatter Plot of Students' Employment Test to Students' Salary", x = "Employment Test", y = "Salary")
```

The code above demonstrates on how to create a scatterplot of **Students' Employment Test to Students' Salary** using the GGPlot2 library.

The output of the code is as below:



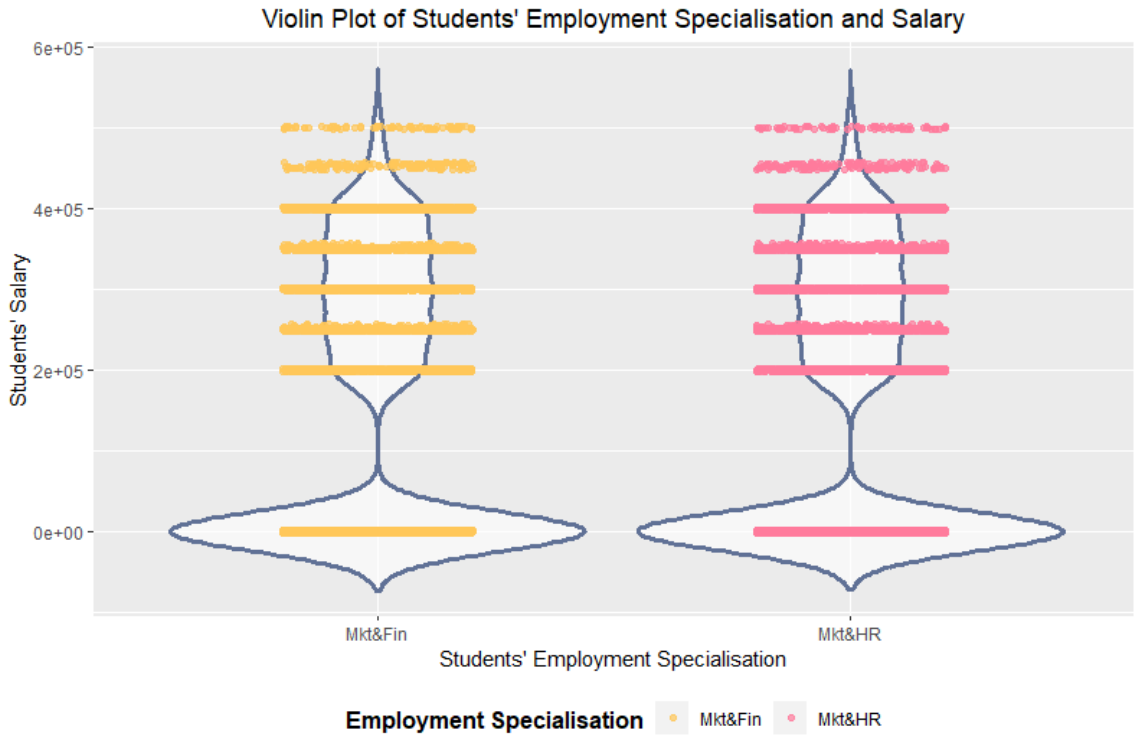

From this scatterplot, it is clear that all three regression lines are horizontal which means students' employment test doesn't affect students' salary as there are no relationship.

## 8.11 Employment Specialisation

```
# - Employment specialisation (Violin Plot)
df_employ_specialisation <- data.frame(
  student_employ_specialisation = as.vector(placementData$Working_Specialism),
  student_salary = as.vector(placementData$Salary)
)

ggplot(df_employ_specialisation, aes(x = factor(student_employ_specialisation), y = student_salary)) +
  geom_violin(trim = FALSE, alpha = 0.6, color = "#607196", linewidth = 1.2) +
  geom_jitter(aes(color = student_employ_specialisation), width = 0.2, alpha = 0.7) +
  labs(
    x = "Students' Employment Specialisation",
    y = "Students' Salary",
    title = "Violin Plot of Students' Employment Specialisation and Salary",
    color = "Employment Specialisation"
  ) +
  scale_color_manual(values = c("#FFC759", "#FF7B9C")) +
  theme(
    plot.title = element_text(hjust = 0.5),
    legend.title = element_text(size = 12, face = "bold"),
    legend.position = "bottom"
  )
```

The code above demonstrates on how to create a violin plot of **Students' Employment Specialisation to Students' Salary** using the GGPlot2 library.

The output of the code is as below:




From this violin plot, it is clear that students' employment specialisation doesn't affect students' salary.

## 8.12 Analysis Conclusion

Based on all graphs, three factors, **secondary school grade, higher secondary school grade and degree grade**, will affect students' salary.

Students who have a salary of 255K, 450K, 455K, and 500K. Firstly, more students who have a salary of 255K achieve between 70 to 85 marks in their secondary school. Secondly, more students who have a salary of 450K achieve 80 marks or above in their secondary school. Thirdly, more students who have a salary of 455K achieve between 60 to 80 marks while for 500K achieve between 65 to 75 marks.

There is a significant peak for students who have a salary of 455K and a drop for students who have a salary of 500K respectively. Both density line peak and drop started from students who achieve 80 marks or above in their higher secondary school.

Students who have a salary of 455K has a drop for students who achieve degree grade between 60 to 70 marks and has a peak for students who achieve degree grade between 75 to 85 marks.

# 9.0 Additional Features

## 9.1 Reshape2 Library

According to Anderson, S. C. (2013), reshape2 is an R library written by Hadley Wickham and it is useful to help data analyst to convert data between wide and long formats. This is because long format data are more suitable for generating graphs and easy to manage. Since **ggplot2** library is used in this documentation, reshape2 is chosen to help to transform data into long format so that it becomes a **"tidy data"** and then can be used to create graph easier.

## 9.2 Hmisc Library

Based on Furman University (n.d.), Hmisc library or Harell Miscellaneous library is useful for data analysis, high-level graphics and utility operations. It also contains functions for computing sample size and power, and many more beneficial functions for data manipulation. Data can be summarized before creating a graph where in this documentation, data is summarized by Hmisc library before generating a frequency polygon.

## 9.3 Stringr Library

Referring to a website created by Tidyverse (n.d.), stringr library allows data analyst to format strings easily and create a clean and neat graph. Stringr is used in this documentation to format strings into title for axes, title, labels etc. to enable viewers to have a better readability.

## 10.0 Placement Data Analysis Verdict

The analysis conducted reveals that several factors, such as academic performance in secondary and higher secondary school, access to the internet, and participation in curricular activities, can influence a student's placement within an organization. On the other hand, factors such as age, working experience, family support, and taking paid classes have little or no significant impact on a student's ability to secure a placement.

Furthermore, the research shows that the educational level of a student's parents has a notable effect on the student's master's grade, with the most significant impact being observed when the mother has no education. Additionally, the father's role in motivating the student to excel in society is crucial.

The impact of paid classes on students is mixed, with students who have family support and engage in curricular activities being more likely to attend paid classes. Conversely, those without family support or internet access tend not to enrol in such classes.

Finally, the analysis indicates that academic performance in secondary and higher secondary school, as well as degree grades, can impact a student's salary. Different levels of academic performance are associated with varying salary ranges.

Overall, the analysis provides valuable insights into the various factors that can influence a student's education and career, highlighting the significance of academic performance, access to resources, and parental education level in shaping a student's future prospects.

# 11.0 References

## 11.1 Data and Algorithms References

STHDA. (n.d.). Wiki, R Software. *Statical Tools for High-throughput Data Analysis*. Retrieved from: http://www.sthda.com/english/wiki/r-software

RDocumentation. (n.d.). RDocumentation. Retrieved from: https://www.rdocumentation.org/

GeeksForGeeks. (n.d.) R programming language – introduction. *GeeksForGeeks*. Retrieved from: https://www.geeksforgeeks.org/r-programming-language-introduction/

## 11.2 Research References

Anderson, S. C. (2013, October 19). An introduction to reshape2. *Sean C. Anderson.* Retrieved from: https://seananderson.ca/2013/10/19/reshape/

Furman University. (n.d.) Overview of Hmisc library. *Furman University*. Retrieved from: http://math.furman.edu/~dcs/courses/math47/R/library/Hmisc/html/Overview.html