CAB-220

Fundementals of Data Science

Portifolio 2

September 30, 2019

Kaspar Kielland [*n10588281*]

CONTENTS

PACKAGES AND LIBRARY'S

```
org_data <- read.csv("Portfolio\ 2\ data.csv", header = TRUE)

install.packages("ggpubr", "tidyverse", "hexbin")
library("ggpubr", "e1071", "corrplot", "DAAG", "ROCR", "gridExtra", "dplyr", "
   tidyverse", "hexbin")

attach(data)
```

1 TASK 1

1.1 Task description

Summarise the information in each variable (except case ID) using a table or an appropriate statistical graph.

1.2 Code

```
#----- Task 1 -----
#### GPA ####
density_GPA <- data %>%
  ggplot( aes(x=GPA))+
  xlab("GPA") +
  ylab("Density") +
  geom_density(fill="#13BFC4",
               color="#e9ecef",
               alpha=0.9) +
  geom_vline(xintercept = mean(GPA),
             color="#13BFC4") +
  geom_text(aes(round(mean(GPA), 2), 0.3,
                label = paste("Mean: ", round(mean(GPA), 2)),
                hjust = 1.5),
            size = 4,
            color = "#13BFC4") +
  theme_minimal()
density_GPA <- density_GPA +
  ggtitle("GPA")

#### OP score ####
density_OP_score <- data %>%
  ggplot( aes(x=OP_Score)) +
  xlab("OP score") +
  ylab("Density") +
  geom_density(fill="#F7766D",
               color="#e9ecef",
               alpha=0.9) +
  geom_vline(xintercept = mean(OP_Score),
             color="#F7766D") +
  geom_text(aes(round(mean(OP_Score), 2), 0.06,
                label = paste("Mean: ", round(mean(OP_Score), 2)),
                hjust = -0.2),
            size = 4, color = "#F7766D") +
  theme_minimal()
density_OP_score <- density_OP_score +
  ggtitle("OP Score")

#### Achived credit points ####
histogram_Achieved_Credit_Points <- data %>%
  ggplot(aes(x=Achieved_Credit_Points)) +
  xlab("Achieved credit points") +
  ylab("Number of occurences") +
  geom_histogram(fill="#02BC38",
                 color="#e9ecef",
                 alpha=0.9) +
  geom_vline(xintercept = mean(Achieved_Credit_Points),
```

```
             color="#02BC38") +
  geom_text(aes(round(mean(Achieved_Credit_Points), 2), 1000,
                label = paste("Mean: ", round(mean(Achieved_Credit_Points), 2)
    ),
                hjust = -0.1),
            size = 4,
            color = "#02BC38") +
  theme_minimal()
histogram_Achieved_Credit_Points <- histogram_Achieved_Credit_Points +
  ggtitle("Achived Credit Points")

#### Failed credit points ####
histogram_Failed_Credit_Points <- data %>%
  ggplot( aes(x=Failed_Credit_Points)) +
  xlab("Failed credit points") +
  ylab("Number of occurences") +
  geom_histogram(fill="#F7766D",
                 color="#e9ecef",
                 alpha=0.9) +
  geom_vline(xintercept = mean(Failed_Credit_Points),
             color="#F7766D") +
  geom_text(aes(round(mean(Failed_Credit_Points), 2), 1750,
                label = paste("Mean: ", round(mean(Failed_Credit_Points), 2)),
                hjust = -0.1),
            size = 4, color = "#F7766D") +
  theme_minimal()
histogram_Failed_Credit_Points <- histogram_Failed_Credit_Points +
  ggtitle("Failed Credit Points")

#### Age ####
density_Age <- data %>%
  ggplot( aes(x=Age)) +
  coord_cartesian(xlim = c(min(Age),
                           max(Age)),
                  expand = FALSE) +
  xlab("Age") +
  ylab("Density") +
  geom_density(fill="#6599FF",
               color="#e9ecef",
               alpha=0.9) +
  geom_vline(xintercept = mean(Age),
             color="#6599FF") +
  geom_text(aes(round(mean(Age), 0), 0.25,
                label = paste("Mean: ", round(mean(Age), 0)),
                hjust = -0.1,
                vjust = 1.5),
            size = 4,
            color = "#6599FF") +
  theme_minimal()
density_Age <- density_Age +
  ggtitle("Age")

#### Attrition Distrubiation ####
df_Attrition <- data %>%
  group_by(Attrition) %>%
```

```
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative
    sum as element
df_Attrition <- df_Attrition %>%
  arrange(desc(Attrition)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for distrubiation
pie_chart_Attrition <- ggplot(df_Attrition,
                              aes(x = "",
                                  y = prop,
                                  fill = Attrition)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_Attrition <- pie_chart_Attrition +
  labs(fill = "Attrition") +
  ggtitle("Attrition")

#### Degree_Type Distrubiation ####
df_Degree_Type <- data %>%
  group_by(Degree_Type) %>%
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative
    sum as element
df_Degree_Type <- df_Degree_Type %>%
  arrange(desc(Degree_Type)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for distrubiation
pie_chart_Degree_Type <- ggplot(df_Degree_Type,
                                aes(x = "",
                                    y = prop,
                                    fill = Degree_Type)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_Degree_Type <- pie_chart_Degree_Type +
  labs(fill = "Degree type") +
  ggtitle("Degree type")

#### Attendance_Type Distrubiation ####
df_Attendance_Type <- data %>%
  group_by(Attendance_Type) %>%
  summarise(counts = n())
```

```r
# Updates df with percantage distrubiation and a vector with the cumulative sum as element
df_Attendance_Type <- df_Attendance_Type %>%
  arrange(desc(Attendance_Type)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)

# Creates pie chart for gender distrubiation
pie_chart_Attendance_Type <- ggplot(df_Attendance_Type,
                                    aes(x = "",
                                        y = prop,
                                        fill = Attendance_Type)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_Attendance_Type <- pie_chart_Attendance_Type +
  labs(fill = "Attendance type") +
  ggtitle("Attendance type")

#### International_student Distrubiation ####
df_International_student <- data %>%
  group_by(International_student) %>%
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative sum as element
df_International_student <- df_International_student %>%
  arrange(desc(International_student)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for gender distrubiation
pie_chart_International_student <- ggplot(df_International_student,
                                          aes(x = "",
                                              y = prop,
                                              fill = International_student)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_International_student <- pie_chart_International_student +
  labs(fill = "International student") +
  ggtitle("International student")

#### Gender Distrubiation ####
df_Gender <- data %>%
  group_by(Gender) %>%
  summarise(counts = n())
```

```r
# Updates df with percantage distrubiation and a vector with the cumulative sum as element
df_Gender <- df_Gender %>%
  arrange(desc(Gender)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for distrubiation
pie_chart_Gender <- ggplot(df_Gender,
                           aes(x = "",
                               y = prop,
                               fill = Gender)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y",
              start = 0) +
  theme_void()
pie_chart_Gender <- pie_chart_Gender +
  scale_fill_discrete(name = "Gender",
                      labels = c("Female", "Male")) +
  ggtitle("Gender")

#### Socio_Economic_Status Distrubiation ####
df_Socio_Economic_Status <- data %>%
  group_by(Socio_Economic_Status) %>%
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative sum as element
df_Socio_Economic_Status <- df_Socio_Economic_Status %>%
  arrange(desc(Socio_Economic_Status)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for gender distrubiation
pie_chart_Socio_Economic_Status <- ggplot(df_Socio_Economic_Status,
                                          aes(x = "",
                                              y = prop,
                                              fill = Socio_Economic_Status)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()

pie_chart_Socio_Economic_Status <- pie_chart_Socio_Economic_Status +
  labs(fill = "Socio economic status") +
  ggtitle("Socio economic status")


#### Teaching._Period_Admitted Distrubiation ####
```

```
df_Teaching._Period_Admitted <- data %>%
  group_by(Teaching._Period_Admitted) %>%
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative
   sum as element
df_Teaching._Period_Admitted <- df_Teaching._Period_Admitted %>%
  arrange(desc(Teaching._Period_Admitted)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for gender distrubiation
pie_chart_Teaching._Period_Admitted <- ggplot(df_Teaching._Period_Admitted,
                                              aes(x = "",
                                                  y = prop,
                                                  fill = Teaching._Period_
   Admitted)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_Teaching._Period_Admitted <- pie_chart_Teaching._Period_Admitted +
  labs(fill = "Teaching period admitted") +
  ggtitle("Teaching period admitted")

#### Faculty Distrubiation ####
df_Faculty <- data %>%
  group_by(Faculty) %>%
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative
   sum as element
df_Faculty <- df_Faculty %>%
  arrange(desc(Faculty)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for distrubiation
pie_chart_Faculty <- ggplot(df_Faculty,
                            aes(x = "",
                                y = prop,
                                fill = Faculty)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_Faculty <- pie_chart_Faculty +
  labs(fill = "Faculty") +
  ggtitle("Faculty")
```

```
#### First_in_family ####
df_First_in_family <- data %>%
  group_by(First_in_family) %>%
  summarise(counts = n())
# Updates df with percantage distrubiation and a vector with the cumulative sum as element
df_First_in_family <- df_First_in_family %>%
  arrange(desc(First_in_family)) %>%
  mutate(prop = round(counts/sum(counts), 3),
         lab.ypos = cumsum(prop) - 0.5*prop)
# Creates pie chart for distrubiation
pie_chart_First_in_family <- ggplot(df_First_in_family,
                                    aes(x = "",
                                        y = prop,
                                        fill = First_in_family)) +
  geom_bar(width = 1,
           stat = "identity",
           color = "transparent") +
  geom_text(aes(y = lab.ypos,
                label = prop*100),
            color = "white") +
  coord_polar("y", start = 0) +
  theme_void()
pie_chart_First_in_family <- pie_chart_First_in_family +
  labs(fill = "First in family") +
  ggtitle("First in family")


#### All charts ####
combined_charts <- grid.arrange(
  arrangeGrob(pie_chart_Gender, pie_chart_Attrition, pie_chart_Degree_Type, ncol = 3),
  arrangeGrob(pie_chart_Teaching._Period_Admitted, pie_chart_Attendance_Type, pie_chart_International_student, ncol = 3),
  arrangeGrob(pie_chart_Socio_Economic_Status, pie_chart_First_in_family, pie_chart_Faculty, ncol = 3),
  arrangeGrob(histogram_Achieved_Credit_Points, histogram_Failed_Credit_Points, ncol = 2),
  arrangeGrob(density_Age, ncol = 1),
  arrangeGrob(density_OP_score, density_GPA, ncol = 2),
  nrow = 6)
```

Listing 1. Code behind **Figure 1**

1.3 Output

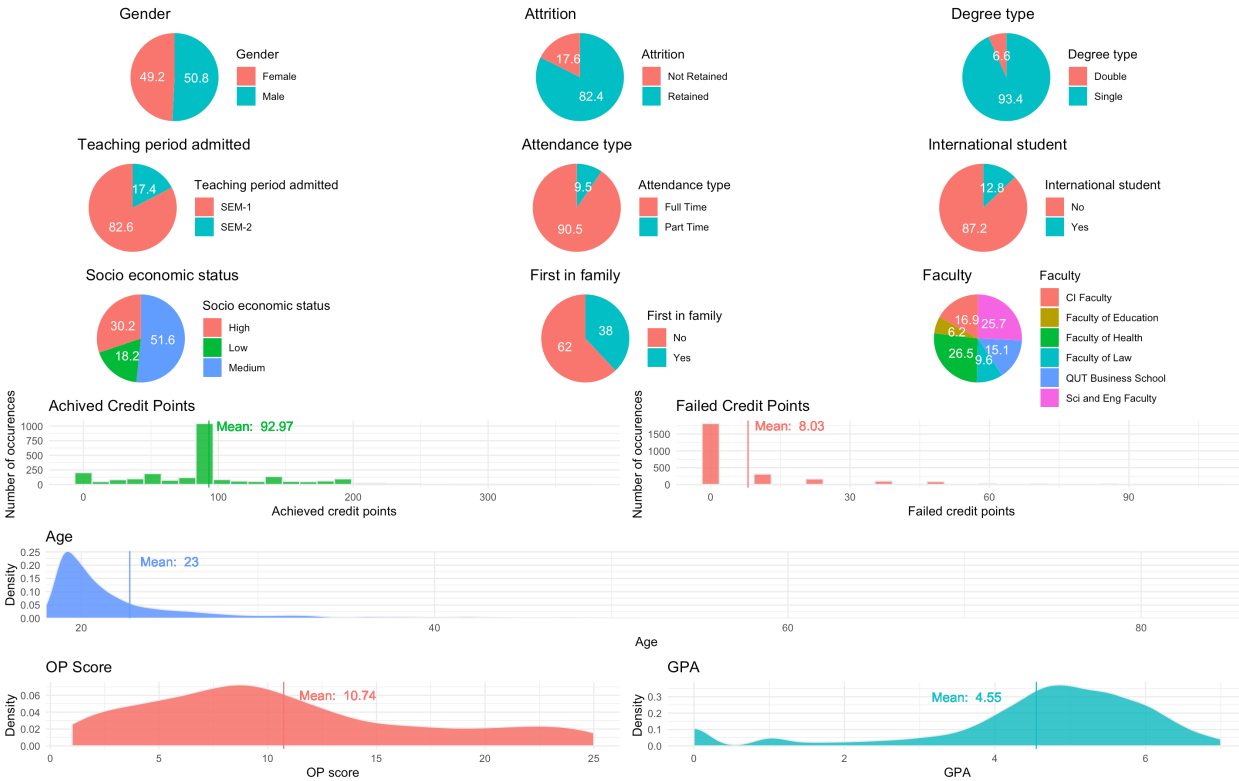

Figure 1. Data summarise

1.4 Summary

By using the following code we can get a summarised overview of our data.

```
summary(data)
```

The output in R gives us information about categories' frequencies of categorical variables and educational statistics, and are listed in Table 1.

Attrition	
Not retained	448
Ratained	2102

Degree type	
Double	169
Single	2381

Attendance type	
Full Time	2308
Part Time	242

International student	
No	2223
Yes	327

First in family	
No	1580
Yes	978

Gender	
Female	1254
Male	1296

Teaching period admitted	
SEM-1	2107
SEM-2	443

Socio economic status	
High	771
Medium	1316
Low	463

Achieved Credit Points	
Min.	0
1st Qu.	60
Median	96
Mean	92.97
3rd Qu.	108
Max.	378

Failed Credit Points	
Min.	0
1st Qu.	0
Median	0
Mean	8.033
3rd Qu.	12
Max.	108

Age	
Min.	18
1st Qu.	19
Median	20
Mean	22.74
3rd Qu.	23
Max.	86

GPA	
Min.	0
1st Qu.	4.13
Median	4.88
Mean	4.55
3rd Qu.	5.63
Max.	7

OP Score	
Min.	1
1st Qu.	6
Median	9
Mean	10.74
3rd Qu.	15
Max.	25

Faculty	
CI Faculty	430
Faculty of Education	158
Faculty of Health	677
Faculty of Law	244
QUT Business School	385
Sci and Eng Faculty	656

Table 1. Summary of data

2 TASK 2

2.1 Task description

Compare average GPA between male and female students using a graph, conduct a statistical test, and interpret its results.

2.2 Code

```r
#----- Task 2 -----
means <- aggregate(GPA ~  Gender, data, mean)
medians <- aggregate(GPA ~  Gender, data, median)

box_plot_GPA_Gender <- ggplot(data,
                              aes(x=Gender,
                                  y=GPA,
                                  fill=Gender)) +
  geom_boxplot(alpha=0.8,
               fatten = 0,
               notch = TRUE,
               varwidth = FALSE) +
  stat_summary(fun.y=mean,
               geom="errorbar",
               aes(ymax = ..y..,
                   ymin = ..y..),
               width = 0.75,
               size = 1,
               linetype = "solid") +
  stat_summary(geom = "text",
               label = paste("Mean: ", round(means$GPA, 2)),
               fun.y = mean,
               vjust=1.2) +
  stat_summary(geom = "text",
               label = paste("Median: ", round(medians$GPA, 2)),
               fun.y = median,
               vjust=-1.2) +
  theme(legend.position="right",
        axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank()) +
  scale_fill_discrete(name = "Gender",
                      labels = c("Female", "Male"))

box_plot_GPA_Gender +
  ggtitle("Comparison of average GPA between\nfemale and male students",
          subtitle = "The respective mean is represented by the black horizontal line\nThe respective median is reprenseted by the notch") +
  scale_y_continuous(breaks=seq(0,7,1))
```

Listing 2. Code behind the box plot chart shown by **Figure 2**

2.3 Output

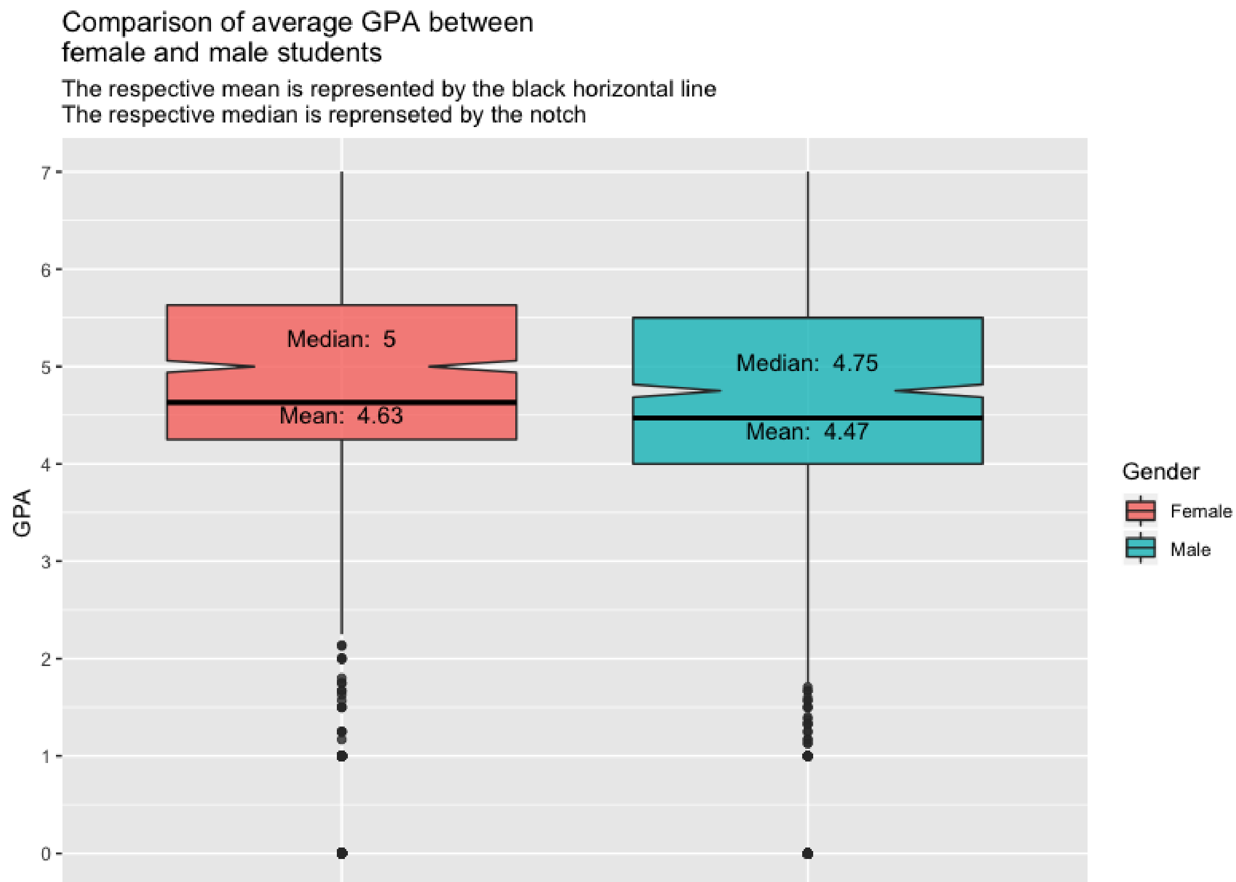

Figure 2. Comparison of GPA between genders

2.4 Statistical test

Before conducting the statistical test we need to define our null and alternative hypotheses. We first define our null hypothesis, H_0, to be "Female students gets on average the same GPA result as male students", and our alternative hypothesis, H_A, to be "Female students gets on average a higher GPA result then male students". Now that we have our null hypothesis, H_0, alternative hypothesis, H_A, we need to define our significance level, α. For this statistical test we set our significance level, α, to be 5%.

Our data to be used in this statistical test are summarized as followed:

H_0 : *Female students gets on average the same GPA result as male students*

H_A : *Female students gets on average a higher GPA result then male students*

$\alpha = 5\%$

```
t.test(GPA ~ Gender)
```

By conducting a Welch Two Sample t-test using the above code we get the following result in the console window:

```
  Welch Two Sample t-test

data:  GPA by Gender
t = 2.4454, df = 2539.7, p-value = 0.01453
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.03111718 0.28297210
sample estimates:
mean in group F mean in group M
       4.629282        4.472238
```

2.5 Interpreting the results

The result from Figure 2 shows us that female students, on average, gets a higher GPA then male students. Both the mean, median and inter-quartile range (IQR) are higher for the female students then for the male students. However, both genders have a lowest low of 0 and a highest high of 7.

The result from the statistical test gives us a p-value of 0.0145 or 1.45% which is lesser then our significance level; $\alpha > p - value \Leftrightarrow 5\% > 1.45\%$. Therefore we can conclude with saying that our alternative hypothesis, H_A, have passed and accept that female students do get, on average, a higher GPA result than male student.

3 TASK 3

3.1 Task description

Explore the relationship between OP Score and GPA using a graph, describe the relationship.

3.2 Code

```
#----- Task 3 -----
library(hexbin)
library(RColorBrewer)

bin<-hexbin(GPA, OP_Score, xbins=20)
my_colors=colorRampPalette(rev(brewer.pal(11,'Spectral')))
plot(bin, style = "colorscale", legend = 1.2, lcex = 1,
     mincnt = 1, maxcnt = 40,
     colorcut = seq(0, 1, length = min(9, max(bin@count))),
     border = 1,
     colramp = my_colors,
     xlab = "GPA", ylab = "OP Score",
     main = "Relationship between\nOP Score and GPA")
```

Listing 3. Code behind the hexbin chart shown by **Figure 3**

3.3 Output

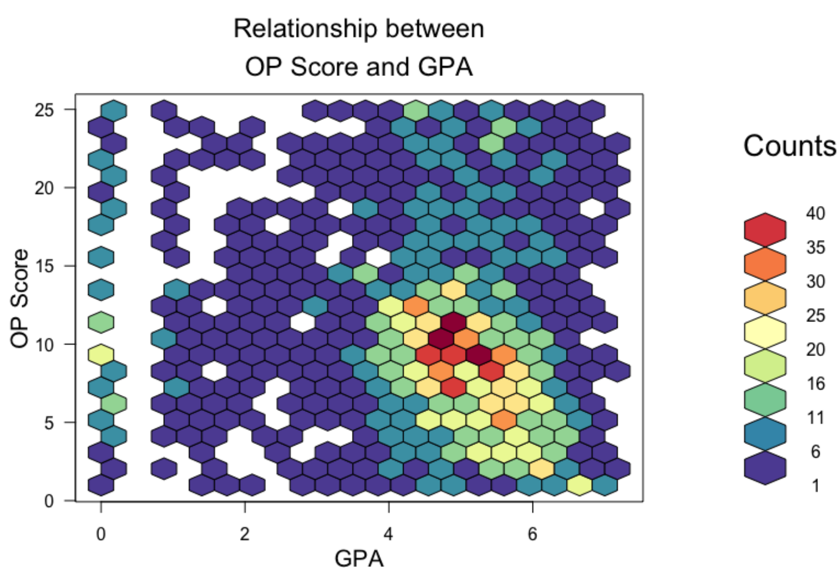

Figure 3. Hexbin chart displaying the relationship between OP Score and GPA

3.4 Interpreting the result

By studying the relationship between OP Score and GPA using the hexbin chart above, we can clearly see that the majority of students have a GPA between 4 and 6. These students have an OP Score lying between 1 and 15.

4 TASK 4

4.1 Task description

Develop a linear regression model of GPA using the given data. You need to describe your choice of predictors, examine your model's assumptions, assess model fit, and interpret the final model's regression coefficients.

4.2 Graphical analysis

This section have been conducted by following Selva Prabhakarans' "Complete Introduction to Linear Regression in R" [1].

4.2.1 Visualising the relationships

Before we can develop a linear regression model we should visualise the relationship between our response variable, GPA, and predictor variables. Since we have multiple potential predictor variables we need to draw a scatter plot along with the line of best fit for each of them.

Code

```r
#Using Scatter Plot to visualise the relationship
par(mfrow = c(2, 2))  # Set up a 2 x 2 plotting space

smoothScatter(Achieved_Credit_Points, y=GPA, xlab = "Achived Credit Points",
    main = "GPA ~ Achived Credit Points")+
  with(data, lines(loess.smooth(Achieved_Credit_Points, GPA), col = "red"))

smoothScatter(Age, y=GPA, main = "GPA ~ Age")+
  with(data, lines(loess.smooth(Age, GPA), col = "red"))

smoothScatter(Failed_Credit_Points, y=GPA, xlab = "Failed Credit Points", main
     = "GPA ~ Failed Credit Points")+
  with(data, lines(loess.smooth(Failed_Credit_Points, GPA), col = "red"))

smoothScatter(OP_Score, y=GPA, xlab = "OP Score", main = "GPA ~ OP Score")+
  with(data, lines(loess.smooth(OP_Score, GPA), col = "red"))
```

Listing 4. Code behind the box plot chart shown by **Figure 4**

Output

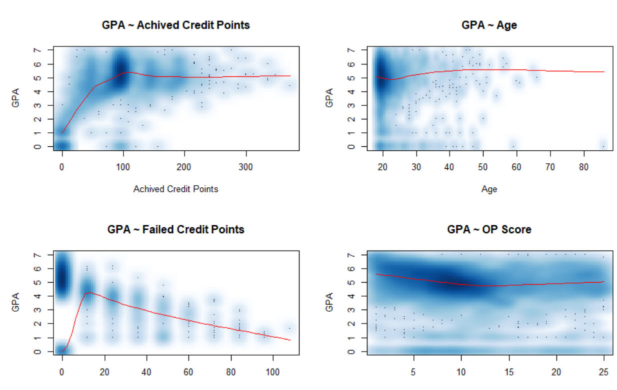

Figure 4. Smooth scatter plots for potentially useful predictors variables along with a line of best fit in red

From figure 4 we can see that our predictor variable to best predict the GPA will be the "Failed Credit Points" variable. These two variables present, overall, a negative relationship.

4.2.2 Checking for outliers

Code

```
# Using Boxplot to check for outliers
par(mfrow = c(1, 2))  # Set up a 1 x 2 plotting space
boxplot(Failed_Credit_Points, main = "Failed Credit Points", sub=paste("
    Outlier rows: ", boxplot.stats(Failed_Credit_Points)$out))
boxplot(GPA, main = "GPA", sub=paste("Outlier rows: ", boxplot.stats(GPA)$out)
    )
```

Listing 5. Code behind the box plot chart shown by **Figure 5**

Output

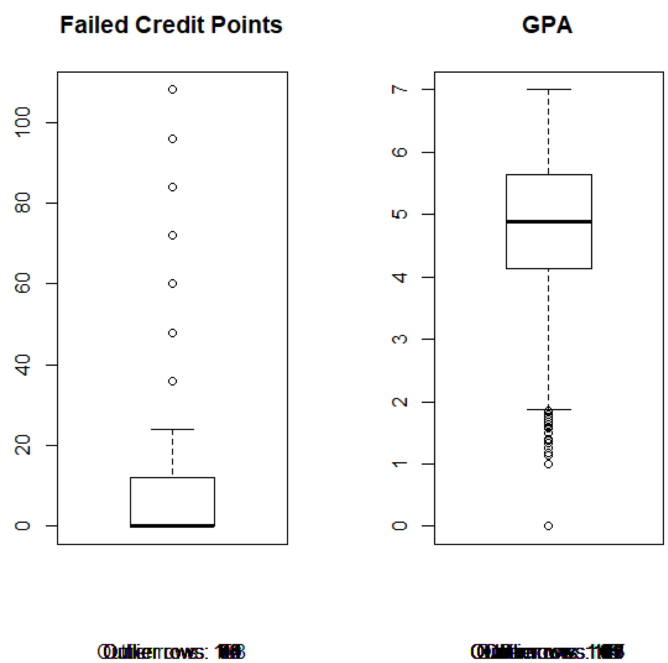

Figure 5. Boxplot of Failed Credit Points and GPA

4.2.3 Check If Response Variable Is Close To Normal

Code

```
# Using Density Plot to check of response variable is close to normal
library(e1071)
par(mfrow=c(1,2))
plot(density(GPA), main="Density Plot: GPA", ylab="Frequency", sub=paste("
    Skewness:", round(e1071::skewness(GPA), 2)))
polygon(density(GPA), col="red")

plot(density(Failed_Credit_Points), main="Density Plot: Failed Credit Points",
     ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(Failed_
    Credit_Points), 2)))
```

```
polygon(density(Failed_Credit_Points), col="red")
```

Listing 6. Code behind the box plot chart shown by **Figure 6**

Output

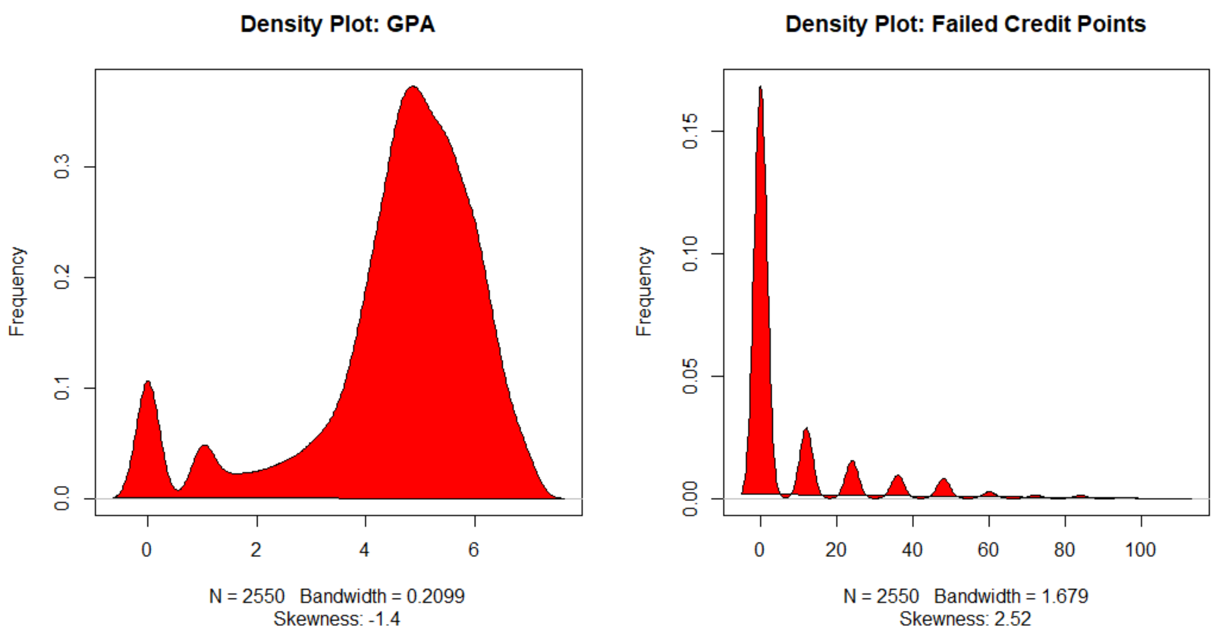

Figure 6. Density plot for GPA and Failed Credit Points

4.3 Correlation

By using the following code we find that Pearson's linear correlation to be -0.473, which is a weak negative correlation.

```
cor(GPA, Failed_Credit_Points)
```

4.4 The linear regression model

```
linearMod <- lm(GPA ~ Failed_Credit_Points)
summary(linearMod)
```

By executing the above code we will get the following output:

```
Call:
lm(formula = GPA ~ Failed_Credit_Points)

Residuals:
    Min      1Q  Median      3Q     Max
-4.9293 -0.2593  0.2081  0.7725  2.4056

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.929342   0.031576  156.11   <2e-16 ***
Failed_Credit_Points -0.047290   0.001743  -27.13   <2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.429 on 2548 degrees of freedom
Multiple R-squared:  0.2241,  Adjusted R-squared:  0.2238
F-statistic:   736 on 1 and 2548 DF,  p-value: < 2.2e-16
```

As we can see from the output above, both our p-values are less then the pre-determined statistical significance level of 0.05. We can therefore say that our model will be statistically significant. However, our R-squared are fairly low with only 0.224 which means that we should not rely solely on this test to estimate possible GPA outcomes.

By using the two components "Intercept" and "Failed_Credit_Points", also known as the *beta coefficients*, we can develop a function to calculate the GPA by using Failed Credit Points as a parameter.

$$GPA = Intercept + (\beta * Failed\_Credit\_Points)$$
$$\Downarrow$$
$$GPA = 4.929 + (-0.047 * Failed\_Credit\_Points)$$

Now that we have our linear regression line we can draw it up with our scatter plot to visualise using the following code.

```
plot(Failed_Credit_Points, GPA)
abline(4.929, -0.047)
```

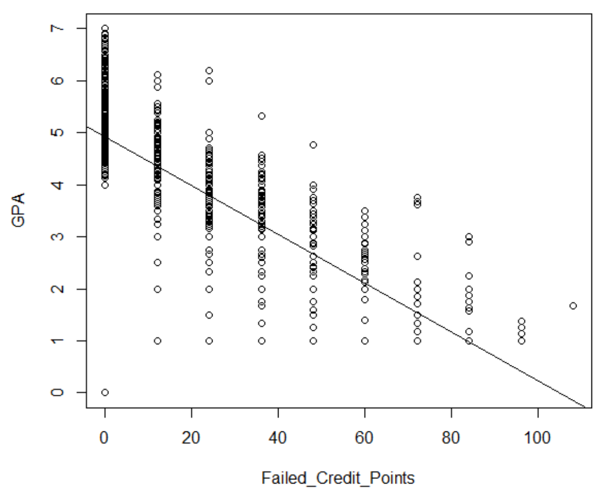

Figure 7. Scatter plot of relation between GPA and Failed Credit Points along with our linear regression line

With this visualisation we can clearly see that the GPA will be lower for students who have many failed credit points, and higher for students with few or none failed credit points.

5 TASK 5

5.1 Task description

Develop a logistic regression model to predict Attrition. You need to describe your choice of predictors, assess model fit, and interpret the final model's regression coefficients.

5.2 Model fitting

We first split our data in to two parts; Training data and Test data. Where Training data contains 80% and the remaining 20% are to be used with our Test data. This is done by executing the below code.

```
# Create Training data and Test data
set.seed(100)  # setting seed to reproduce results of random sampling
trainingRowIndex <- sample(1:nrow(data), 0.8*nrow(data))  # row indices for
    training data
trainingData <- data[trainingRowIndex, ]  # model training data
testData  <- data[-trainingRowIndex, ]   # test data
```

Now we need to fit our model, this can be done by executing the below code.

```
logistic <- glm(Attrition ~ ., data=trainingData, family=binomial(link='logit
    '))
summary(logistic)
```

We have now obtained the results from our fitted model which is shown below.

```
Call:
glm(formula = Attrition ~ ., family = binomial(link = "logit"),
    data = trainingData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4995   0.2041   0.4308   0.5803   1.8828

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     1.030e+00  5.151e-01   1.999  0.04557 *
ID                             -1.502e-04  8.888e-05  -1.690  0.09103 .
Degree_TypeSingle              -8.894e-01  3.469e-01  -2.564  0.01035 *
Achieved_Credit_Points          1.637e-02  2.027e-03   8.079 6.52e-16 ***
Attendance_TypePart Time        1.356e+00  2.881e-01   4.706 2.53e-06 ***
Age                            -2.126e-02  1.101e-02  -1.930  0.05355 .
Failed_Credit_Points           -1.614e-02  3.848e-03  -4.193 2.75e-05 ***
International_studentYes        7.979e-01  2.826e-01   2.824  0.00474 **
First_in_familyYes              7.267e-02  1.350e-01   0.538  0.59031
GenderM                         1.670e-01  1.402e-01   1.191  0.23357
GPA                             9.047e-02  4.752e-02   1.904  0.05692 .
OP_Score                       -4.433e-03  1.058e-02  -0.419  0.67523
Socio_Economic_StatusLow       -5.796e-03  1.921e-01  -0.030  0.97593
Socio_Economic_StatusMedium    -9.001e-03  1.503e-01  -0.060  0.95223
Teaching._Period_AdmittedSEM-2  3.504e-01  1.881e-01   1.862  0.06254 .
FacultyFaculty of Education     7.109e-01  3.221e-01   2.207  0.02730 *
FacultyFaculty of Health        2.103e-01  1.945e-01   1.081  0.27954
FacultyFaculty of Law          -3.151e-01  2.528e-01  -1.246  0.21260
FacultyQUT Business School      5.658e-01  2.365e-01   2.392  0.01675 *
FacultySci and Eng Faculty      4.465e-01  2.046e-01   2.182  0.02910 *
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1913.5  on 2039  degrees of freedom
Residual deviance: 1550.8  on 2020  degrees of freedom
AIC: 1590.8

Number of Fisher Scoring iterations: 5
```

5.3 Interpreting the result

From the results from the model fitting we can see that only the variables *Achieved Credit Points*, *Attendance Type*, *Failed Credit Points* and *International Student* are of statistically significance. Of these statistically significant variables we see that *Attendance Type* has the lowest p-value suggesting a strong association of the students attendance with the probability of being retained.

Now we can analyze the table of deviance to see how our model is doing against the null model.

```
anova(logistic, test="Chisq")
```

The function call above prompts us with the following output.

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: Attrition

Terms added sequentially (first to last)


                           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                        2039     1913.5
ID                          1    2.268      2038     1911.3 0.1320472
Degree_Type                 1   12.632      2037     1898.7 0.0003792 ***
Achieved_Credit_Points      1  262.255      2036     1636.4 < 2.2e-16 ***
Attendance_Type             1   23.335      2035     1613.1 1.361e-06 ***
Age                         1    3.003      2034     1610.0 0.0830865 .
Failed_Credit_Points        1   18.832      2033     1591.2 1.428e-05 ***
International_student       1    8.816      2032     1582.4 0.0029863 **
First_in_family             1    0.292      2031     1582.1 0.5892548
Gender                      1    3.839      2030     1578.3 0.0500845 .
GPA                         1    5.485      2029     1572.8 0.0191823 *
OP_Score                    1    0.138      2028     1572.7 0.7105264
Socio_Economic_Status       2    0.011      2026     1572.6 0.9945336
Teaching._Period_Admitted   1    3.661      2025     1569.0 0.0556895 .
Faculty                     5   18.152      2020     1550.8 0.0027620 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.4 ROC Curve

```
p <- predict(logistic, newdata=subset(testData), type="response")
pr <- prediction(p, testData$Attrition)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```

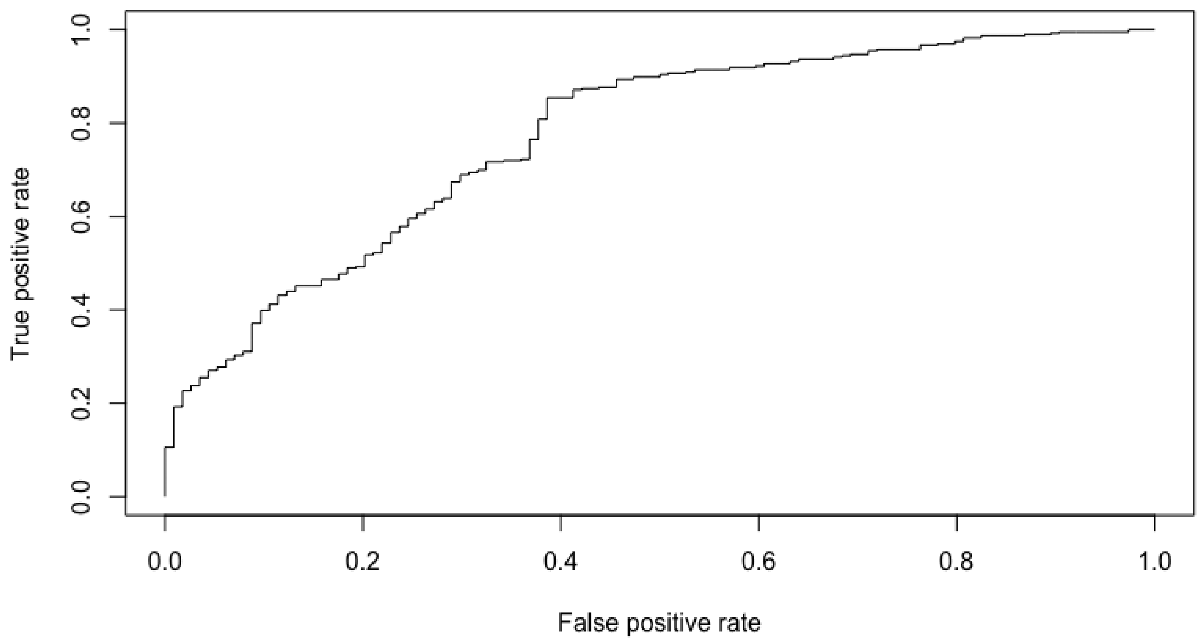

Figure 8. ROC curve for our test data

5.4.1 AUC

```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
[1] 0.7723064
```

Since our AUC are closer to 1 than to 0.5, we can say that our model have an accepteble predictive ability.

REFERENCES

[1] S. Prabhakaran. (2017). Complete introduction to linear regression in r. Accessed: 21.08.19, [Online]. Available: https://www.machinelearningplus.com/machine-learning/complete-introduction-linear-regression-r/.