CAB220 se2 2020: Portfolio 2

n10212361 - Jamie Martin

Due 11:59pm Sunday 27 September 2020

About this Assessment item

This individual Assessment item accounts for 20% of your overall grade of CAB220 and aims to assess your knowledge and skills in

- Following clear instructions
- Summarizing data
- Statistical hypothesis testing
- Exploratory data analysis
- Linear regression
- Logistic regression
- Using `R` for data science, as described in
    - R for Data Science (https://r4ds.had.co.nz/) by Garrett Grolemund and Hadley Wickham
    - ggplot2: Elegant Graphics for Data Analysis (https://ggplot2-book.org/) by Hadley Wickham

While every attempt has been made to present this Assessment clearly and correctly we may update this Assessment item with additional points of clarification or correction.

- Any such changes will be announced on BlackBoard.

How to submit this Assessment item

In the following instructions, replace the string `10212361` with your 7 or 8 digit student number so the Teaching Team can identify your submission.

1. Complete the assessment by writing `R` and `Rmarkdown` code below the "Assessment Tasks" heading to solve these tasks
2. Replace the text on line 3 `put your student number and name here` with your student number and name
3. Rename the file containing your completed submission `CAB220_20se2_Portfolio2_n10212361.Rmd`
4. Knit your submission into `.html` output,
   open that output in your web browser and
   save it in PDF format to the file `CAB220_20se2_Portfolio2_n10212361.pdf`
    - If you prefer, you can knit your submission directly to PDF but you will need $LaTeX$ installed to do that
5. Zip both the `.Rmd` and `.pdf` you have created into a file called `CAB220_20se2_Portfolio2_n10212361.zip`
6. Submit your `.zip` file using the submission link in **Blackboard > Assessment > Portfolio 2**

Make sure that you understand and can execute these steps well before the due date.

About the data

The fictitious data set for this Assessment consists of observations of first-year University students with variables *case ID*, *Attrition*, *Degree Type*, *Achieved Credit Points*, *Attendance Type*, *Age*, *Failed Credit Points*, *International student*, *First in family in university*, *Gender*, *GPA*, *OP Score*, *Socio Economic Status*, *Teaching Period Admitted*, and *Faculty*.

- This dataset is provided in `./data/students.csv`

About the tasks and their marks

There are 7 Tasks worth marks as follows:

Task	**Marks**
Task 1: Following clear instructions	1
Task 2: Summarizing data	5
Task 3: Statistical hypothesis testing	2
Task 4: Exploratory data analysis	2
Task 5: Linear regression	5
Task 6: Logistic regression	4
Task 7: Using `R` for data science	1

The Assessment tasks

Task 1: Follow the instructions

1. Submit your work according to the instructions in the “How to submit this Assessment item” section

Task 2: Summarizing data

2. Appropriately summarise each variable (except *case ID*) using tables and/or graphs

```
read.csv(file.choose(),header = TRUE) -> data

# remove student id
data[,-1] -> data

# assign numerical ids
data %>% as.tbl() %>% rownames_to_column() -> data.id
```

```
## Warning: `as.tbl()` is deprecated as of dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

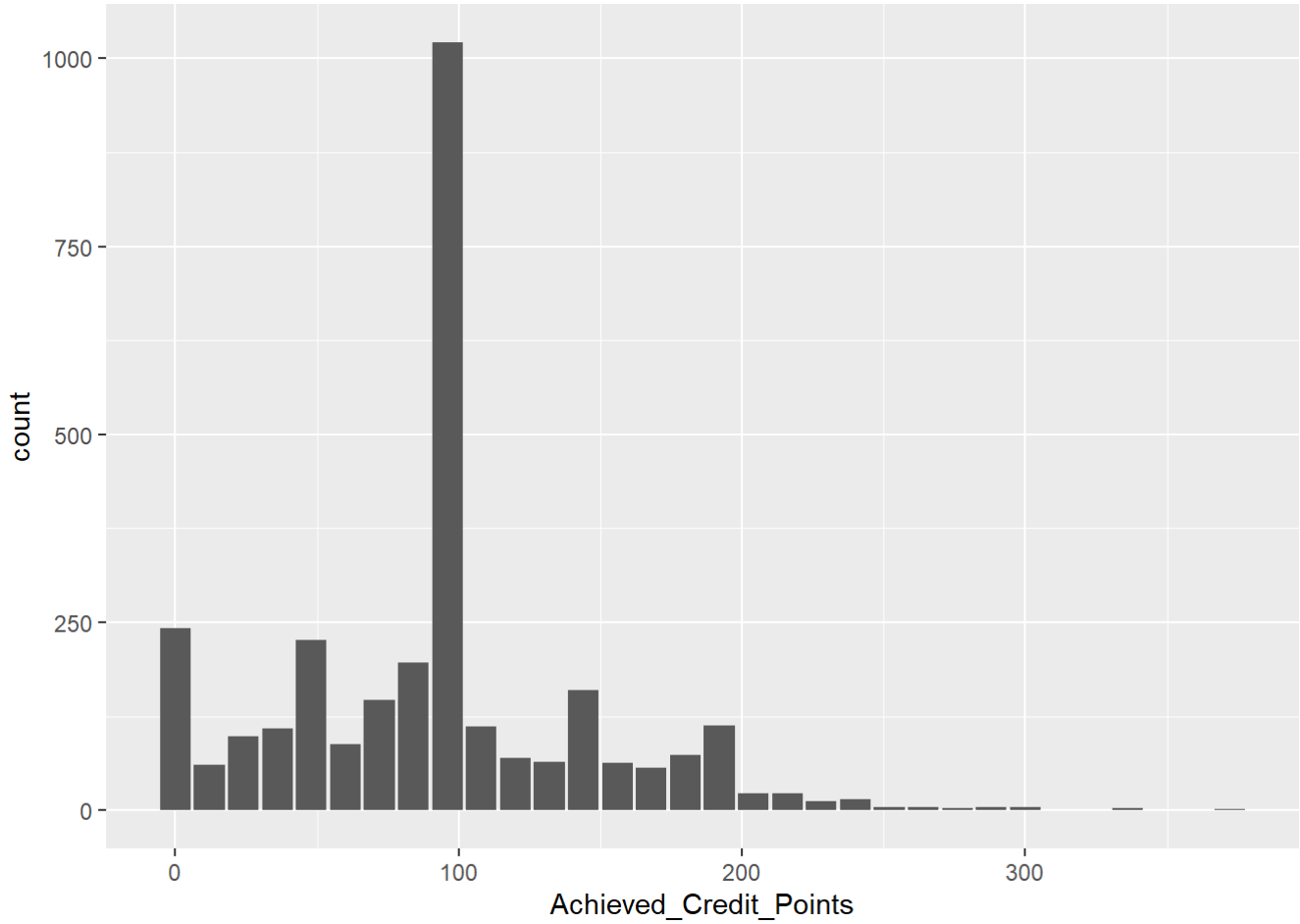
```
# mutate char variables into factors [1,2,3...]
data %>%
  mutate(
    Attrition               = factor(Attrition),
    Degree_Type             = factor(Degree_Type),
    Attendance_Type         = factor(Attendance_Type),
    International_student   = factor(International_student),
    First_in_family         = factor(First_in_family),
    Gender                  = factor(Gender),
    Socio_Economic_Status   = factor(Socio_Economic_Status),
    Teaching_Period_Admitted = factor(Teaching_Period_Admitted),
    Faculty                 = factor(Faculty)
  ) -> data
```

```
# summarise the continuous data points
summary(data[c(3,5,6,10,11)])
```

```
##  Achieved_Credit_Points      Age        Failed_Credit_Points      GPA
##  Min.   :  0.00         Min.   :18.00   Min.   :  0.00       Min.   :0.000
##  1st Qu.: 60.00         1st Qu.:19.00   1st Qu.:  0.00       1st Qu.:4.120
##  Median : 96.00         Median :20.00   Median :  0.00       Median :4.880
##  Mean   : 92.88         Mean   :22.72   Mean   :  7.98       Mean   :4.546
##  3rd Qu.:108.00         3rd Qu.:23.00   3rd Qu.: 12.00       3rd Qu.:5.620
##  Max.   :372.00         Max.   :86.00   Max.   :108.00       Max.   :7.000
##     OP_Score
##  Min.   : 1.00
##  1st Qu.: 6.00
##  Median : 9.50
##  Mean   :10.76
##  3rd Qu.:15.00
##  Max.   :25.00
```

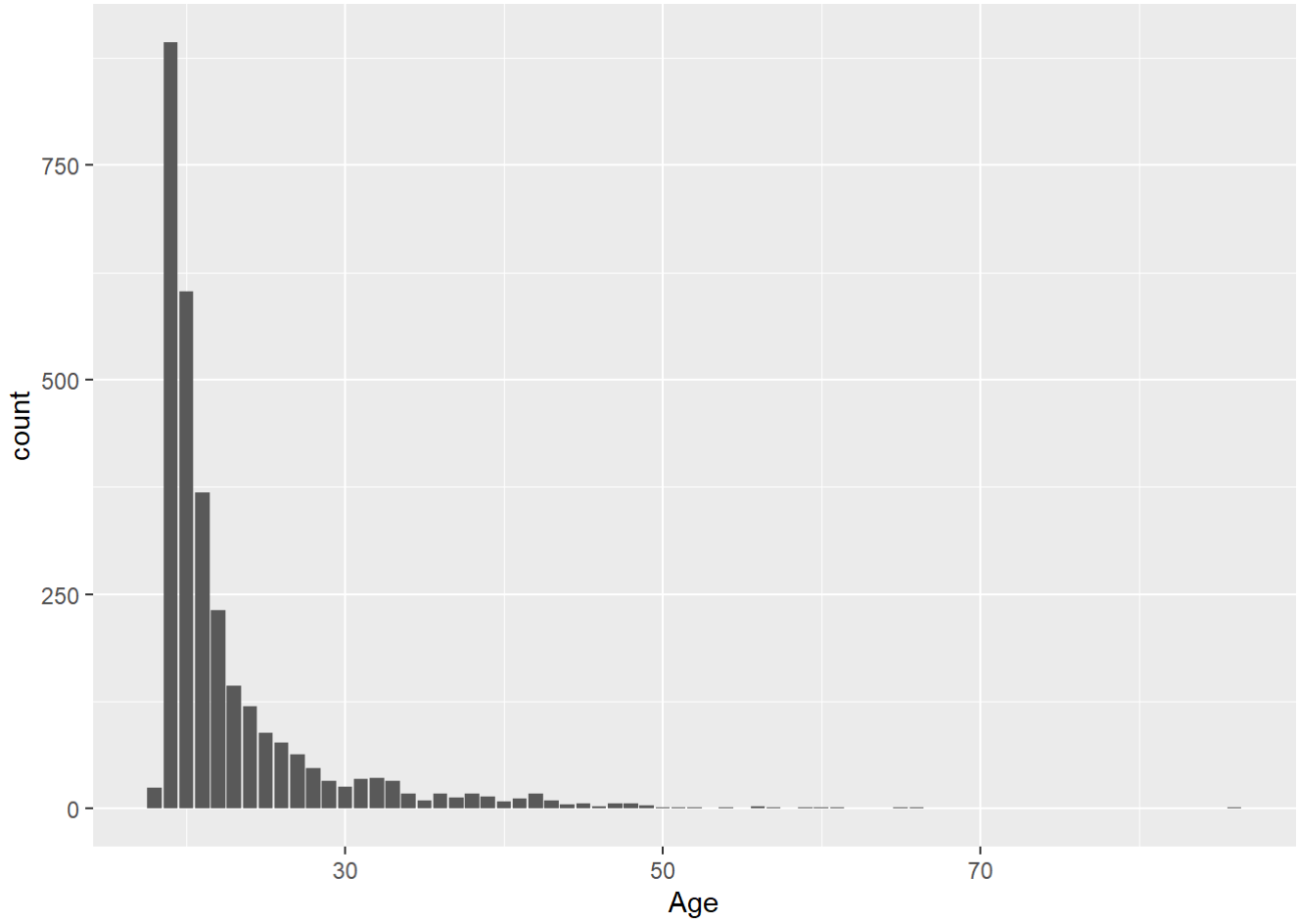
```
# plot Achieved_Credit_Points in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Achieved_Credit_Points), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



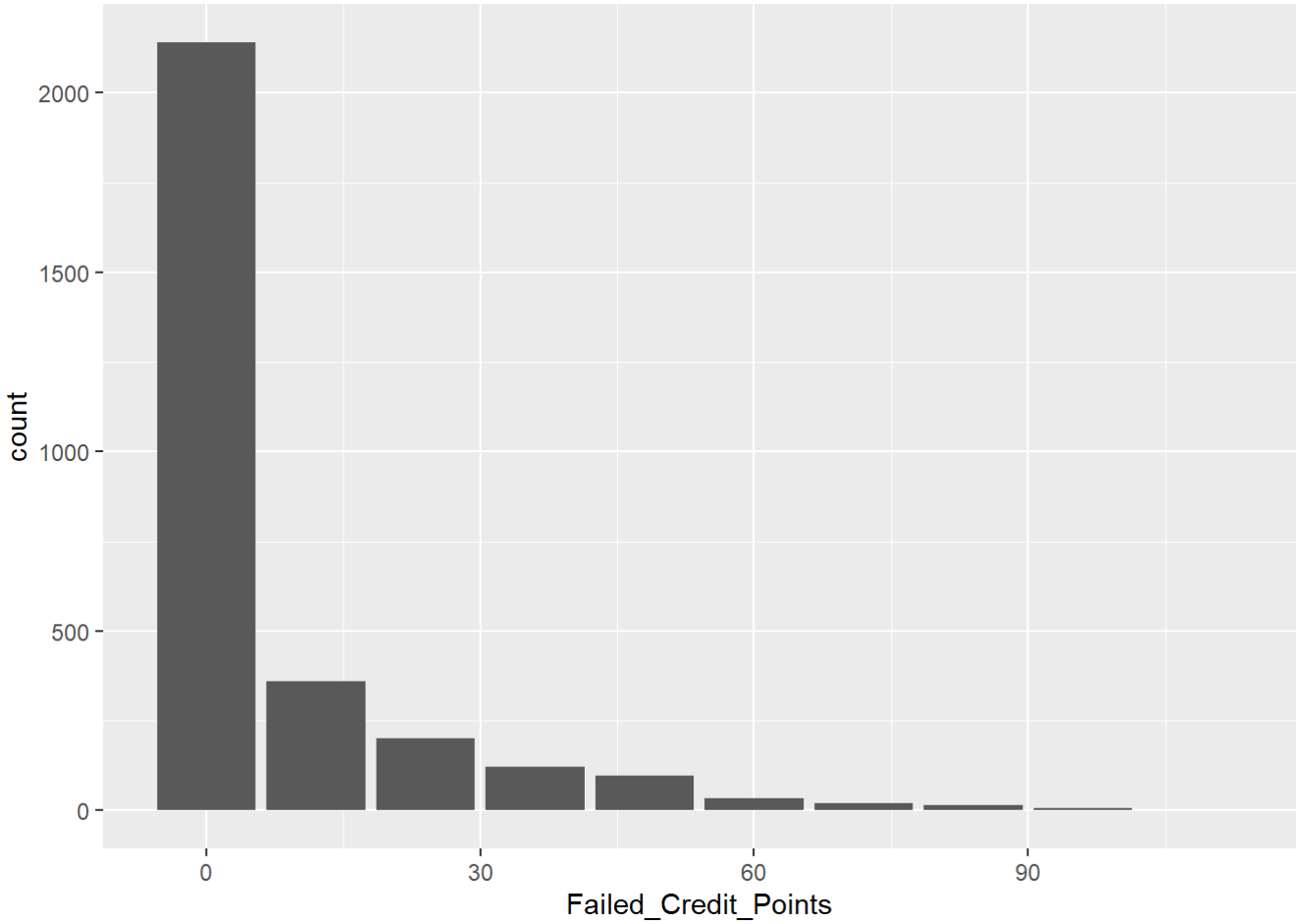

```
# plot Age in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Age), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



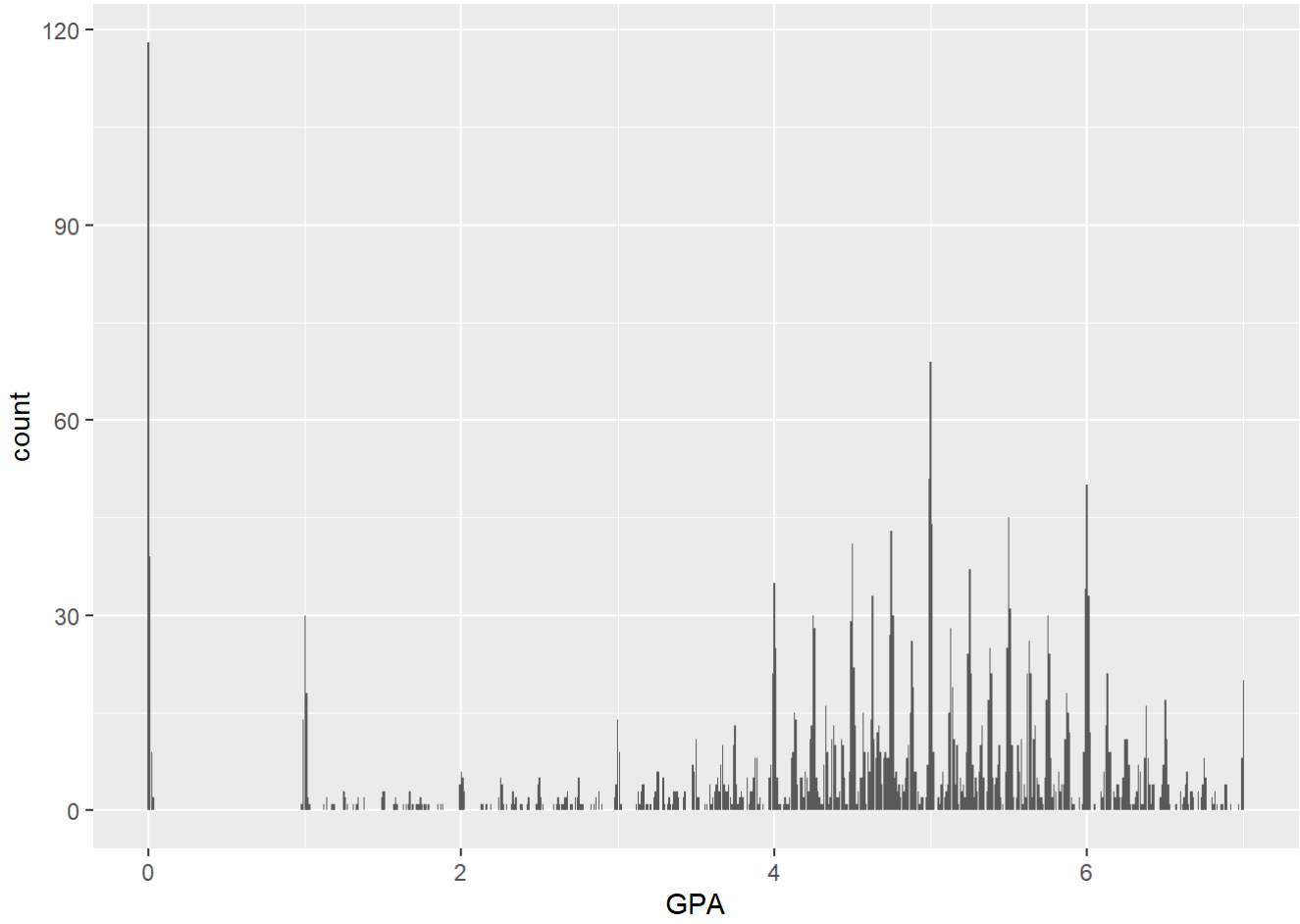

```
# plot Failed_Credit_Points in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Failed_Credit_Points), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



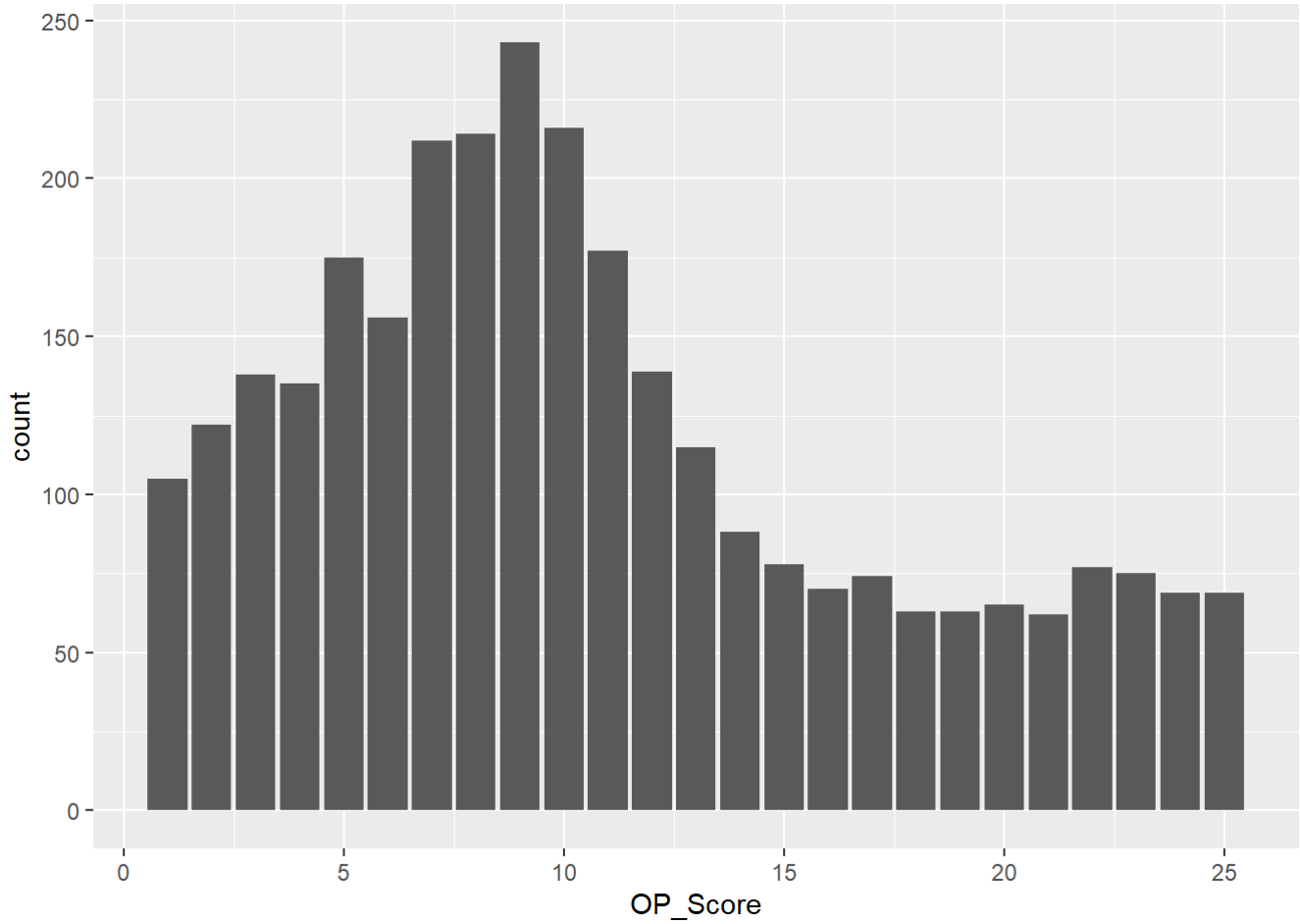

```
# plot GPA in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=GPA), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
# plot OP_Score in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=OP_Score), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



From these 5 tables we can infer:

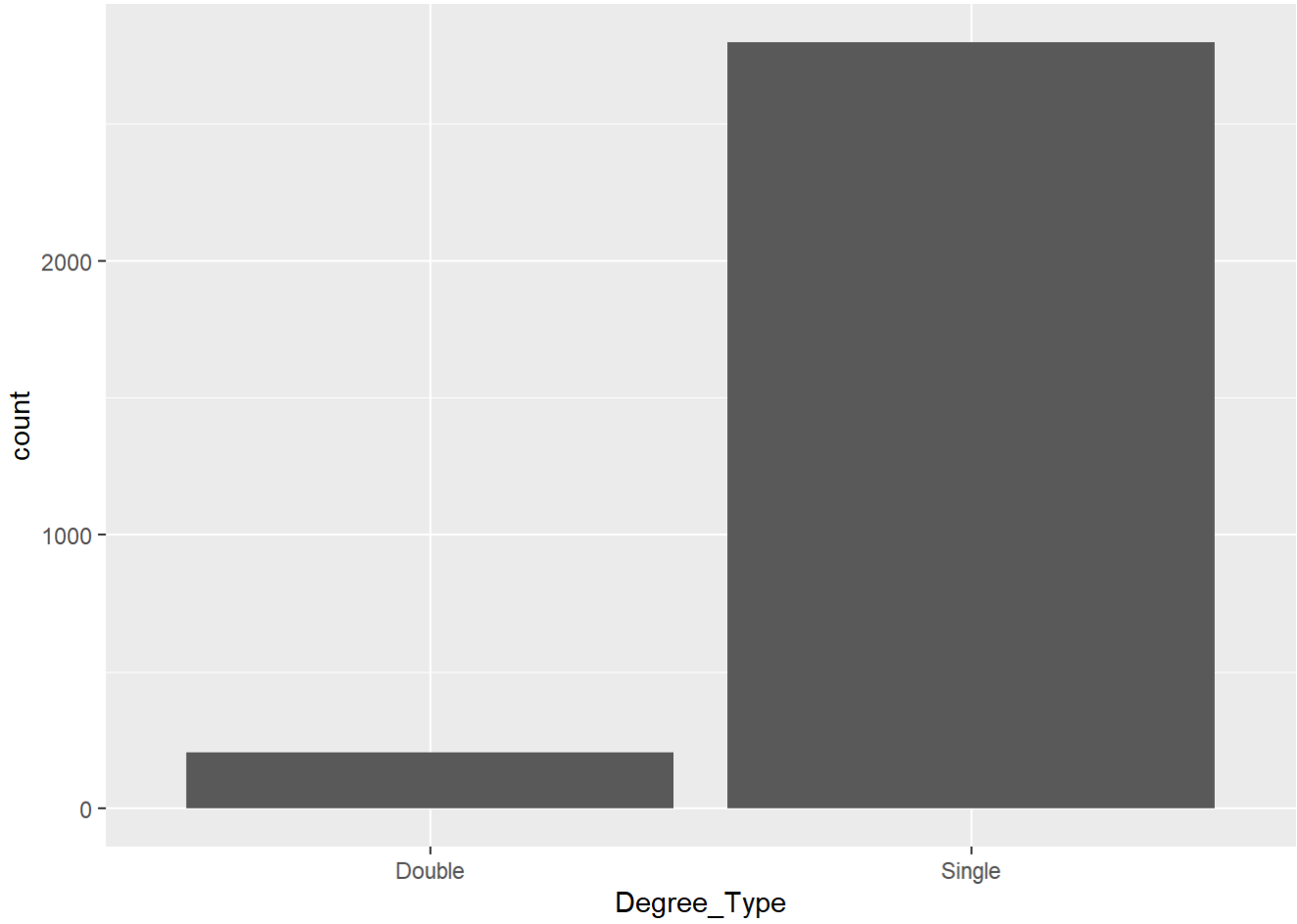
- The majority of students study a 96 credit point unit. The general length of a single degree
- The majority of students are young adults between the age of 19 - 23
- The majority of students are able to study without failure
- The majority of students score a 4 or a 5
- In comparison to the GPA system, the OP system is skewed, with a greater number of students achieving higher (lower numerical) score.

```
# create table of Degree_Type
table(data$Degree_Type)
```

```
##
## Double Single
##    204   2796
```

```
# plot Degree_Type in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Degree_Type), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



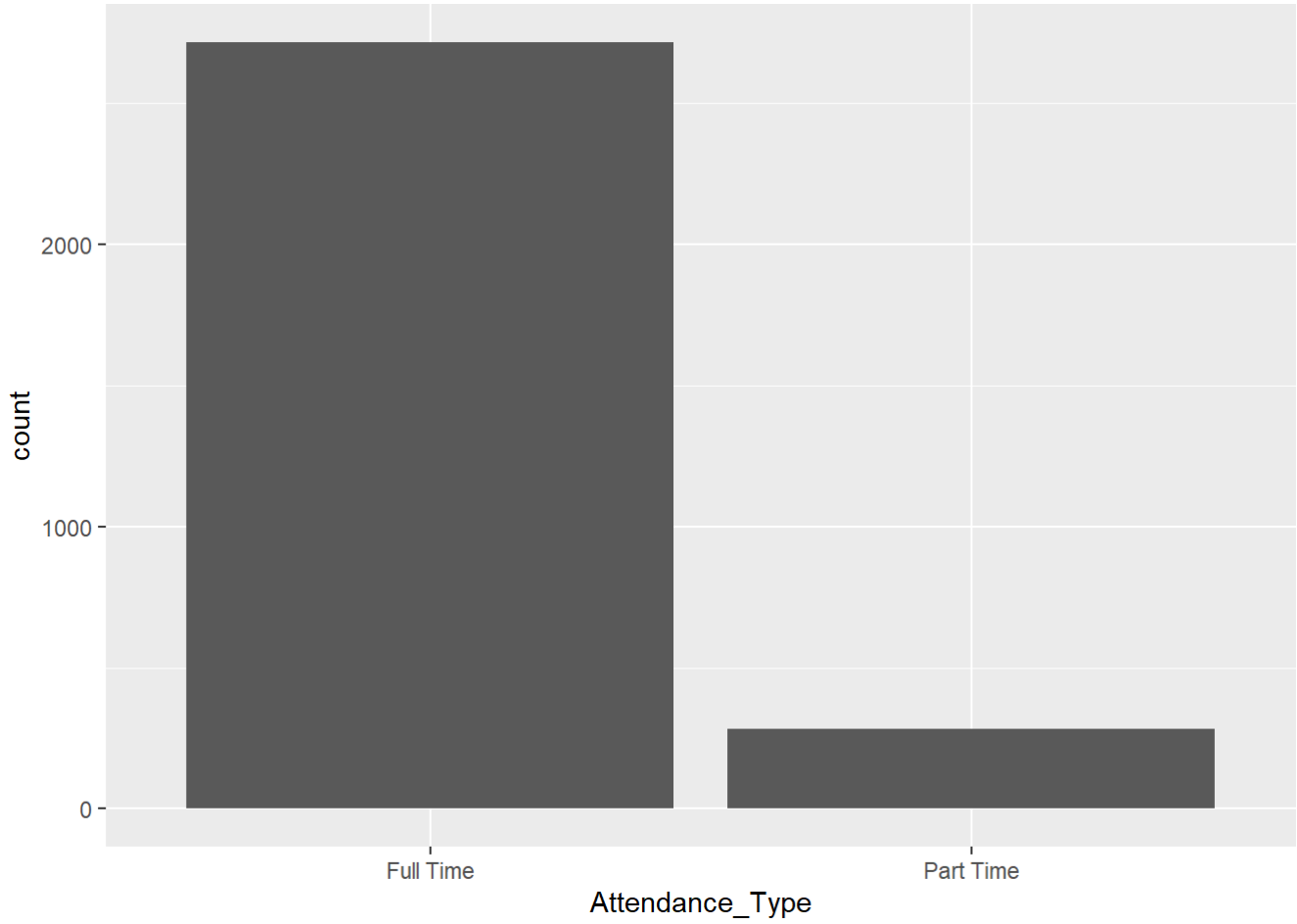
This shows roughly 1 in 15 people take a double degree

```
# create table of Attendance_Type data
table(data$Attendance_Type)
```

```
##
## Full Time Part Time
##      2716       284
```

```
# plot Attendance_Type in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Attendance_Type), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



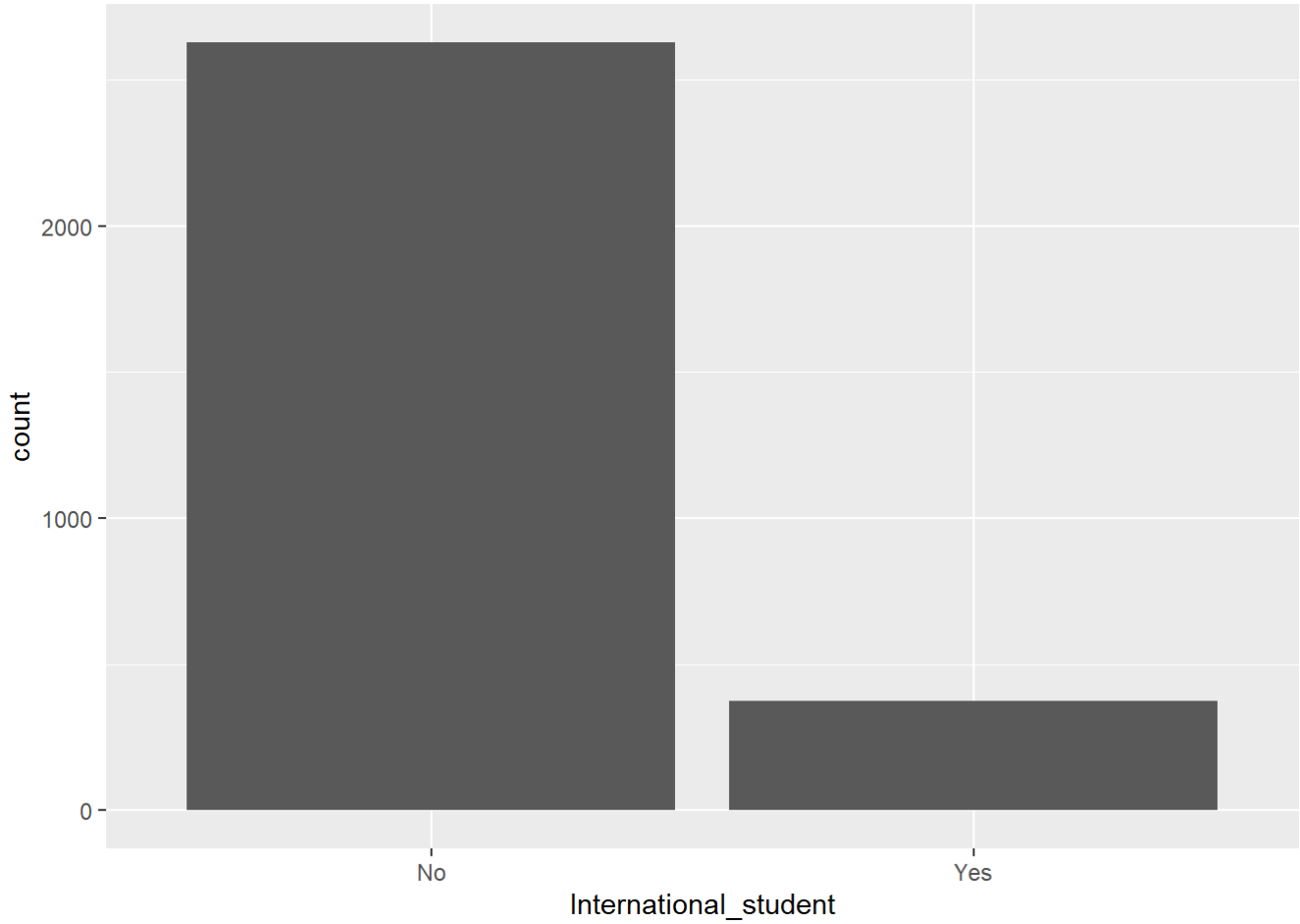
This shows roughly 1 in 11 students study part time.

```
# create table of International_student data
table(data$International_student)
```

```
##
##   No  Yes
## 2627  373
```

```
# plot International_student in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=International_student), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



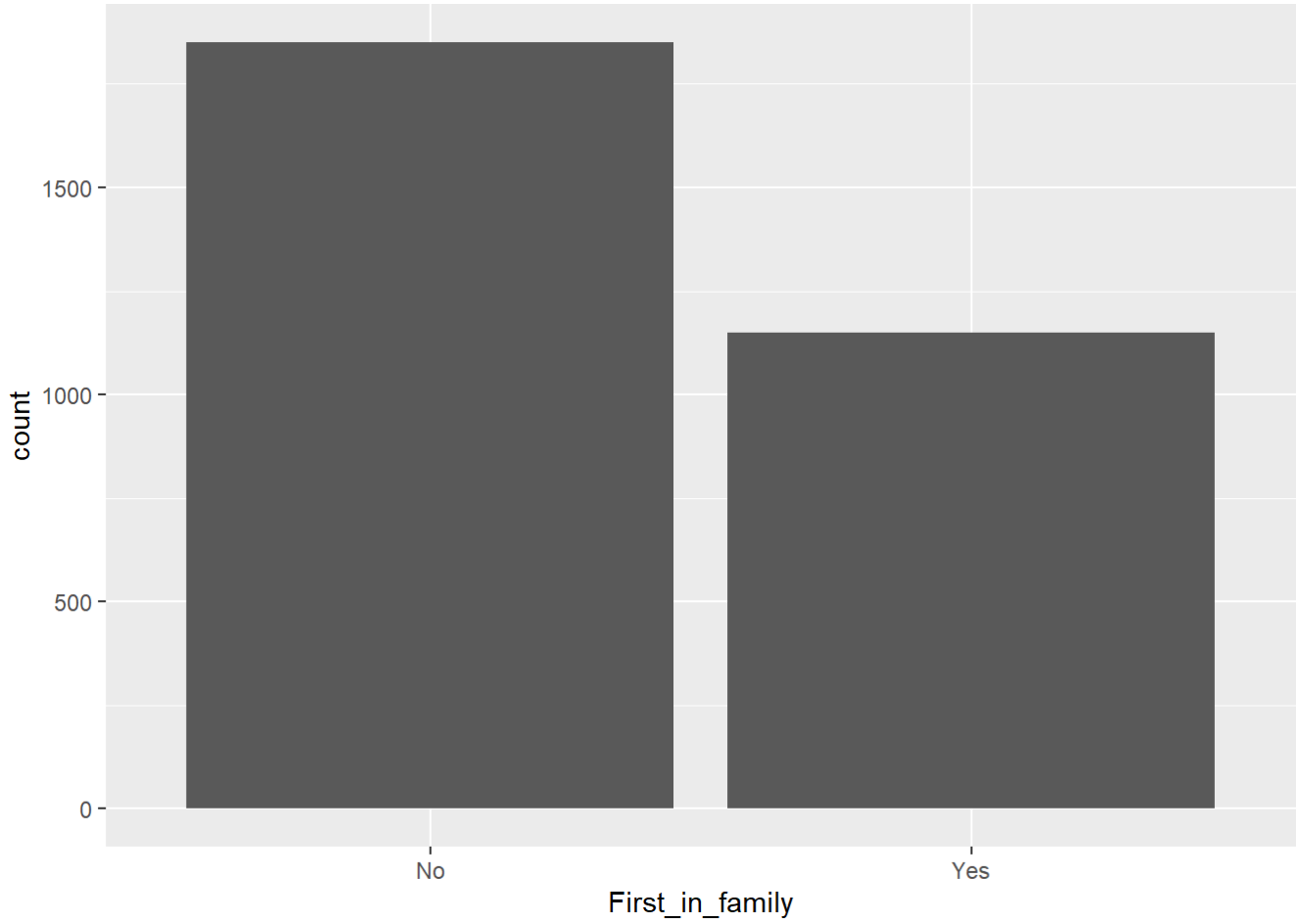
This shows students are predominantly national. Roughly 1 in 8 are international.

```
# create table of First_in_family data
table(data$First_in_family)
```

```
##
##   No  Yes
## 1850 1150
```

```
# plot First_in_family in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=First_in_family), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



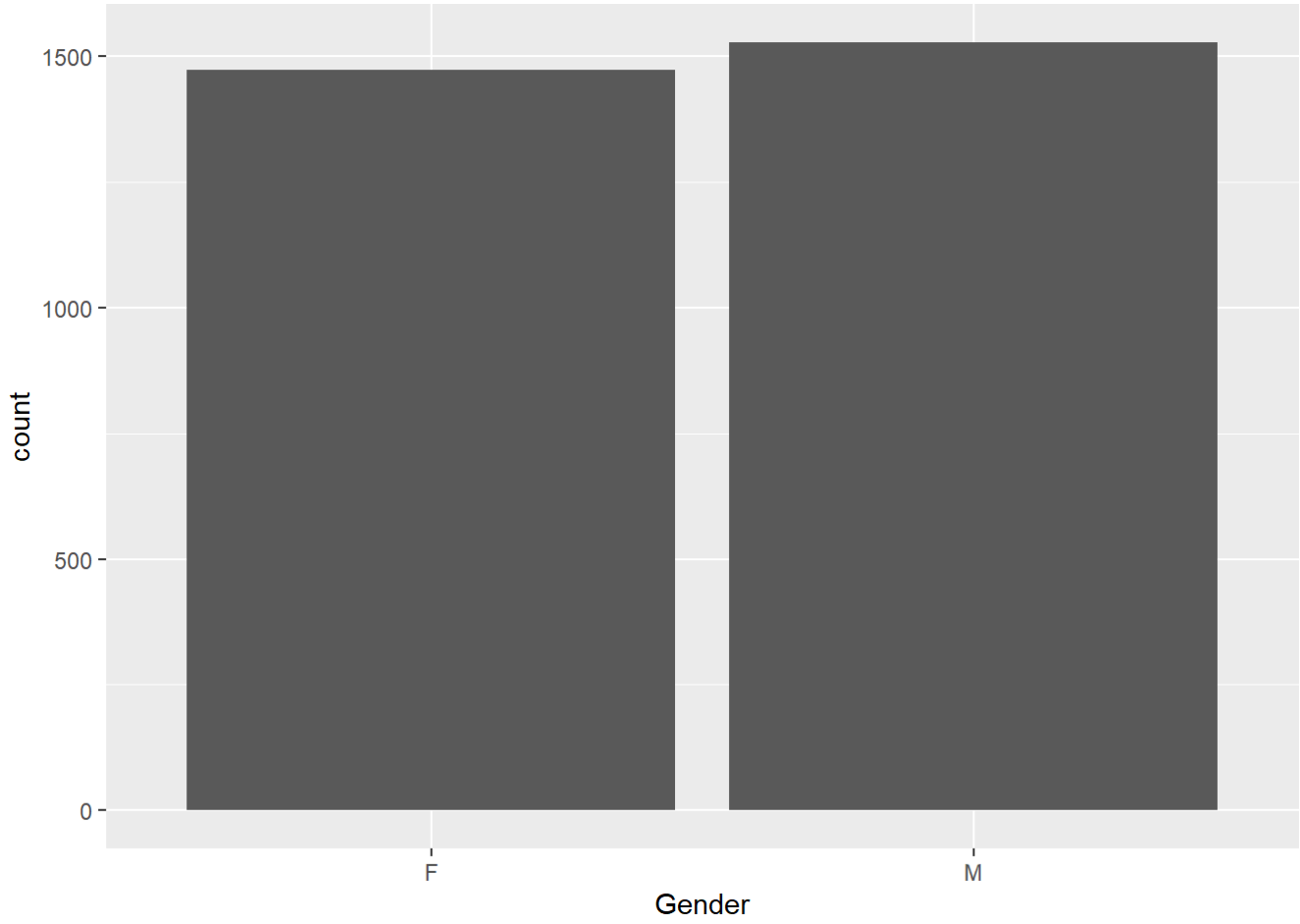
Students are more likely to not be the first in their family, but there are still a lot of students who are the first to study in their family

```
# create table of Gender data
table(data$Gender)
```

```
##
##    F    M
## 1473 1527
```

```
# plot Gender in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Gender), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



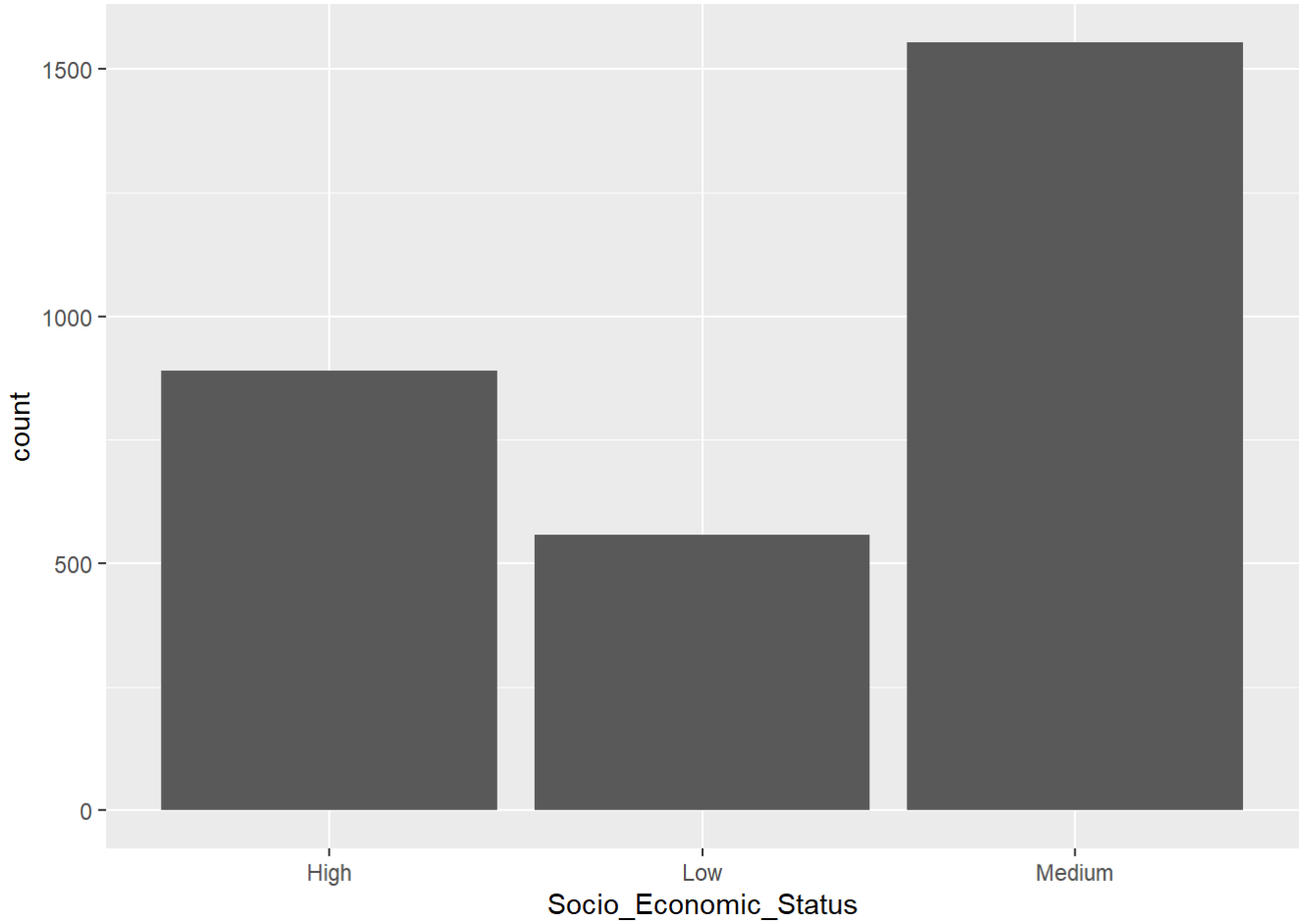
The split between gender is fairly even either side.

```
# create table of Socio_Economic_Status data
table(data$Socio_Economic_Status)
```

```
##
##   High    Low Medium
##    889    557   1554
```

```
# plot Socio_Economic_Status in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Socio_Economic_Status), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



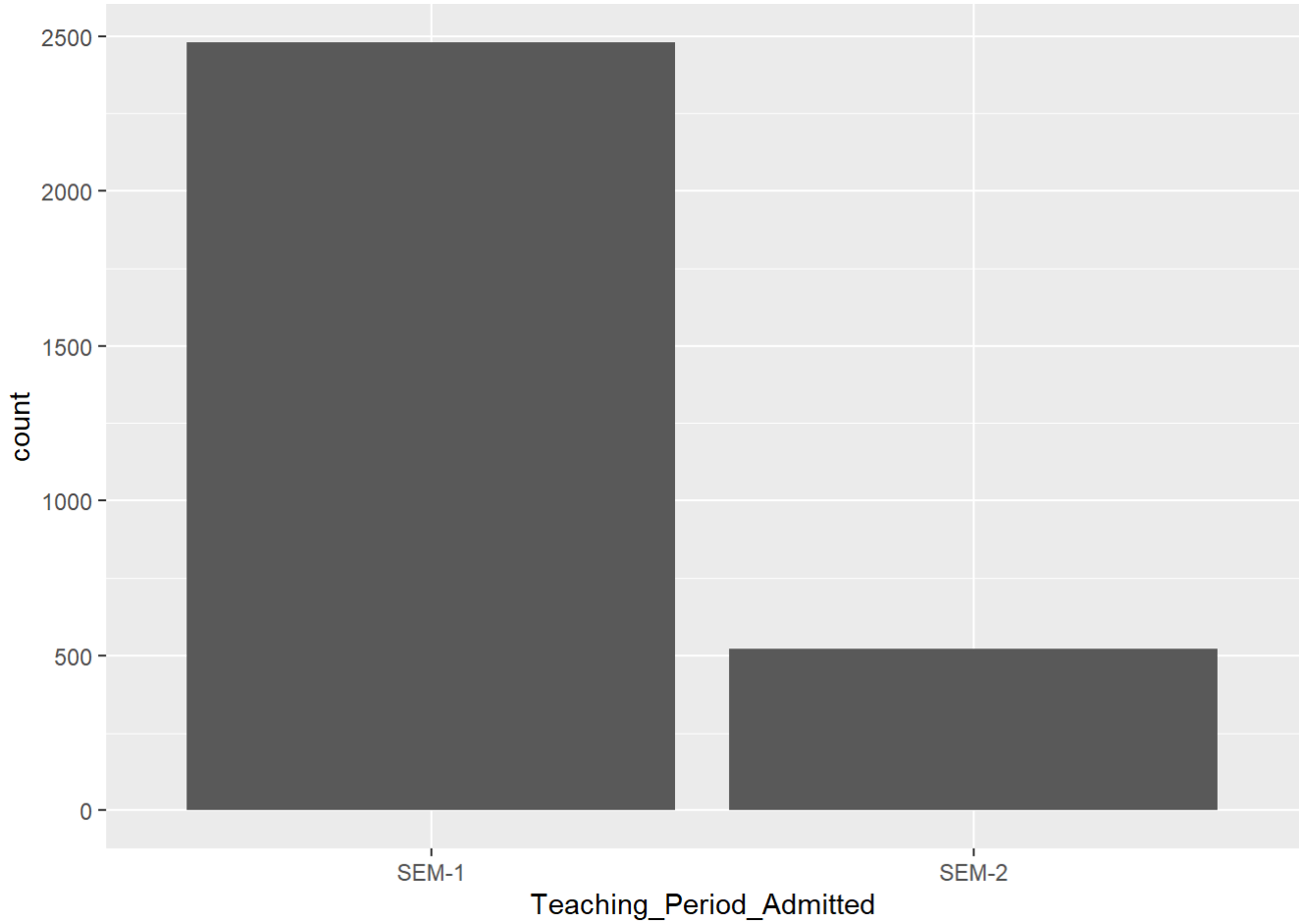
Roughly 50% of students are from a Medium Socio-Economic backgrounds.

```
# create table of Teaching_Period_Admitted data
table(data$Teaching_Period_Admitted)
```

```
##
## SEM-1 SEM-2
##  2479   521
```

```
# plot Teaching_Period_Admitted in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Teaching_Period_Admitted), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



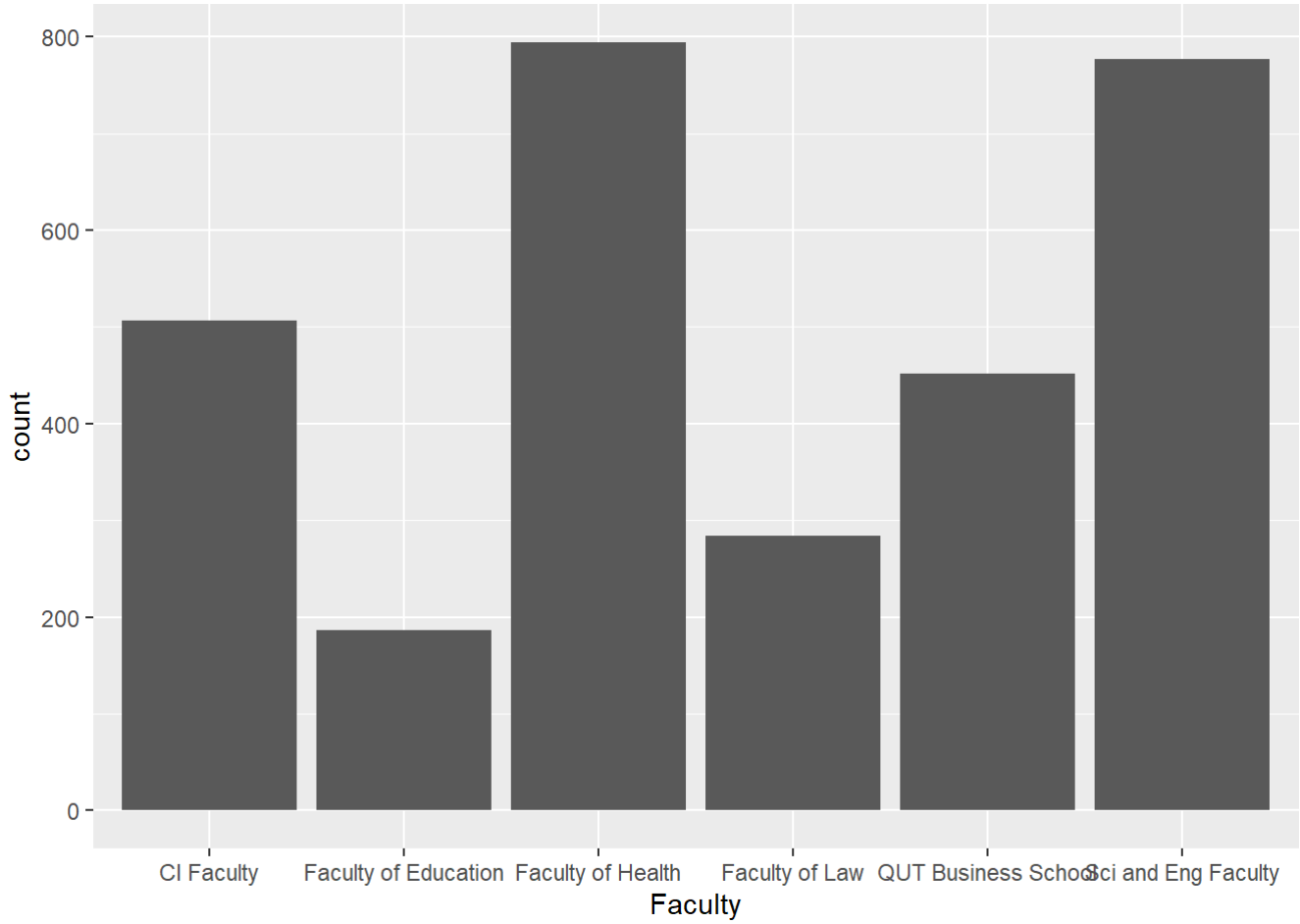
Most students start in Semester 1. Roughly 1 in 6 start in SEM-2

```
# create table of faculty data
table(data$Faculty)
```

```
##
##           CI Faculty Faculty of Education    Faculty of Health
##                  507                  186                  794
##       Faculty of Law  QUT Business School  Sci and Eng Faculty
##                  284                  452                  777
```

```
# plot faculties in count histogram
ggplot(data=data) +
  geom_histogram(aes(x=Faculty), stat="count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

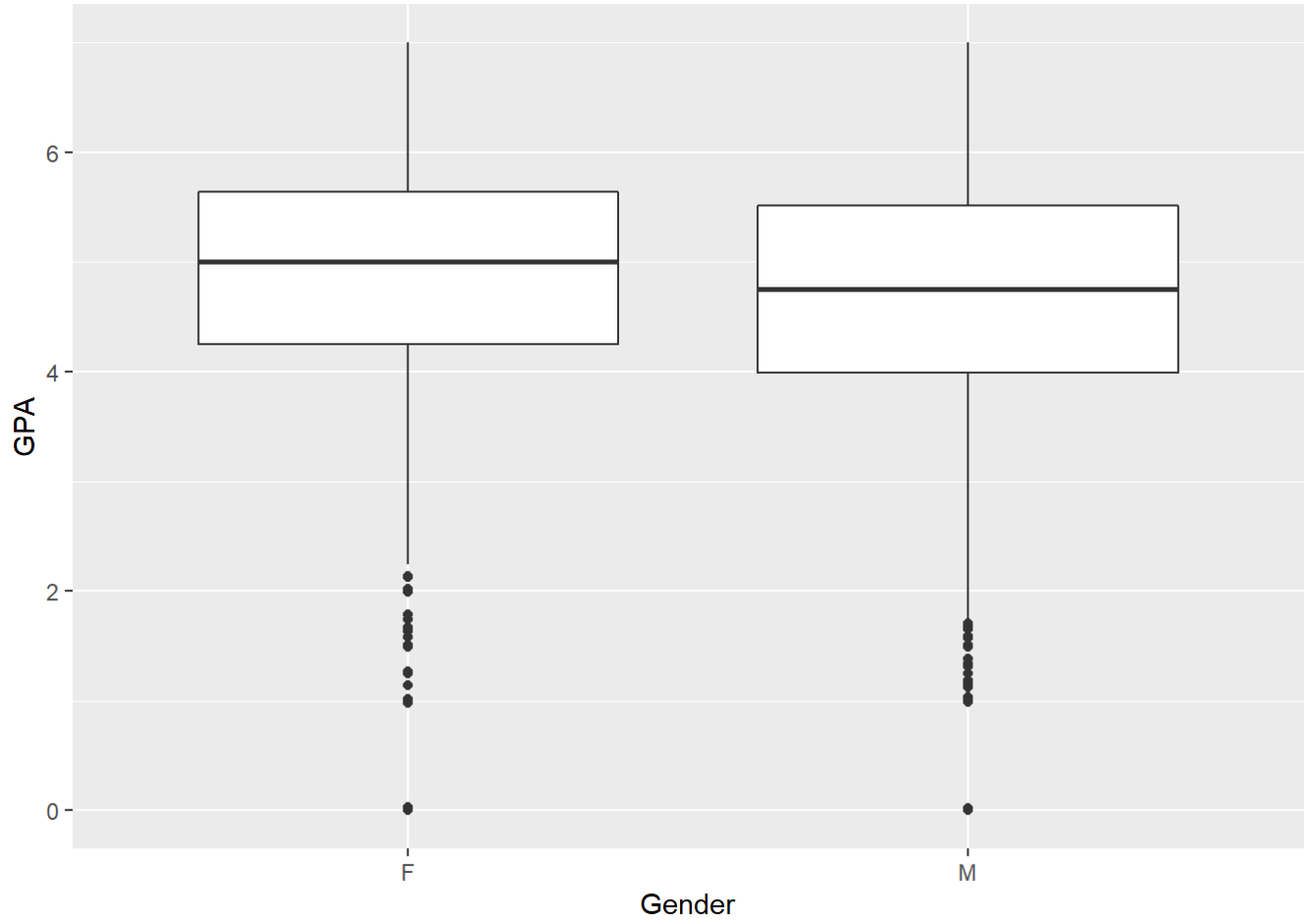


The roughly 50% of students are studying degrees from the Faculty of Health or Sci and Eng

Task 3: Statistical hypothesis testing

3. Compare average GPA between male and female students using an informative graph and an appropriate statistical test.
   Interpret your findings.

```
# create boxplot of Gender vs GPA
ggplot(data=data, aes(x=Gender, y=GPA)) +
  geom_boxplot()
```



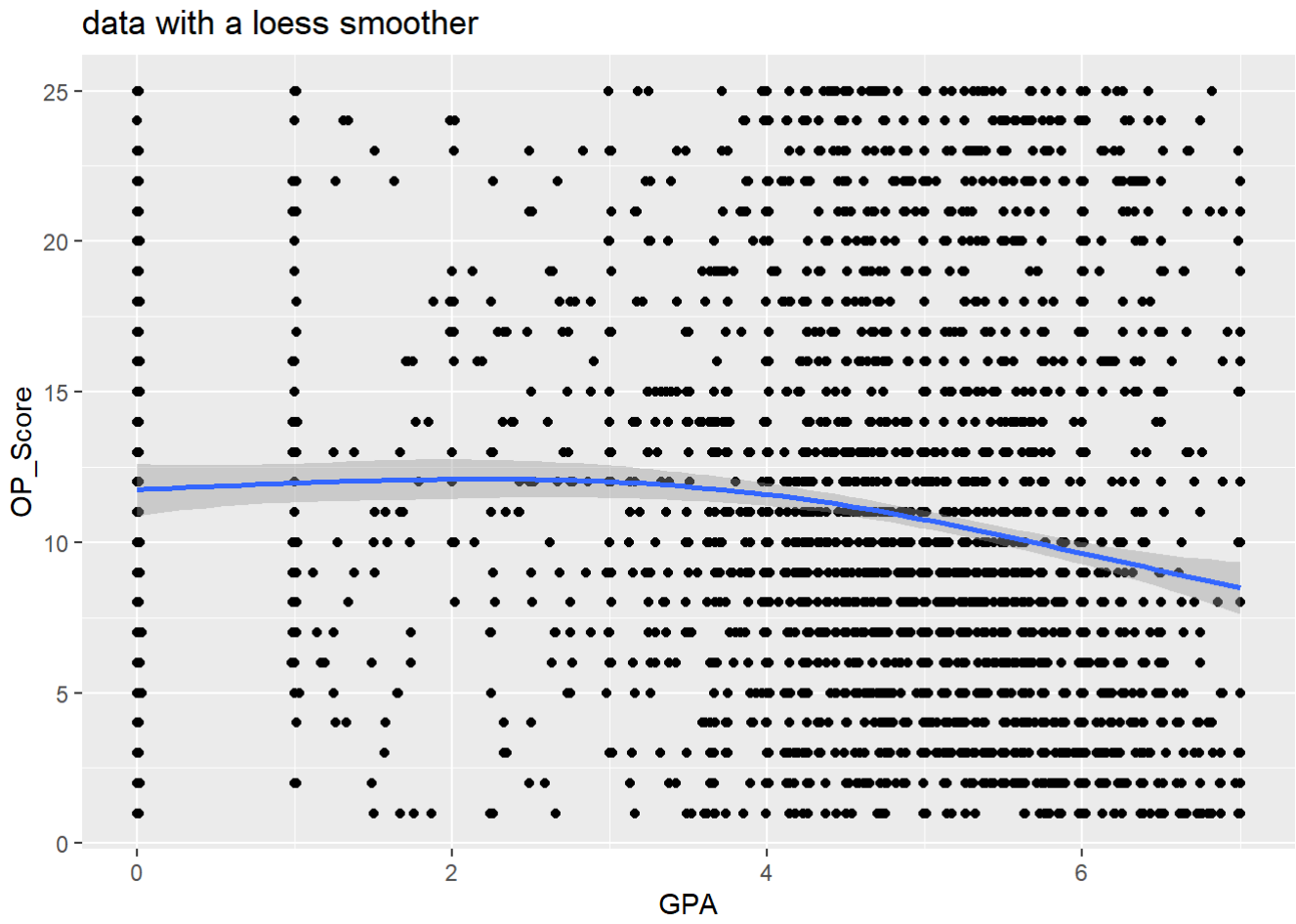
In the box-plot above one can see that females average a slightly higher GPA than males. The largest GPA spread of females pushes closer towards 7, with a more concentrated areas of GPA score. On the other hand, males have a wider lower average spread and a larger range.

Task 4: Exploratory data analysis

4. Explore the relationship between OP Score and GPA using a graph.
   Describe the relationship.

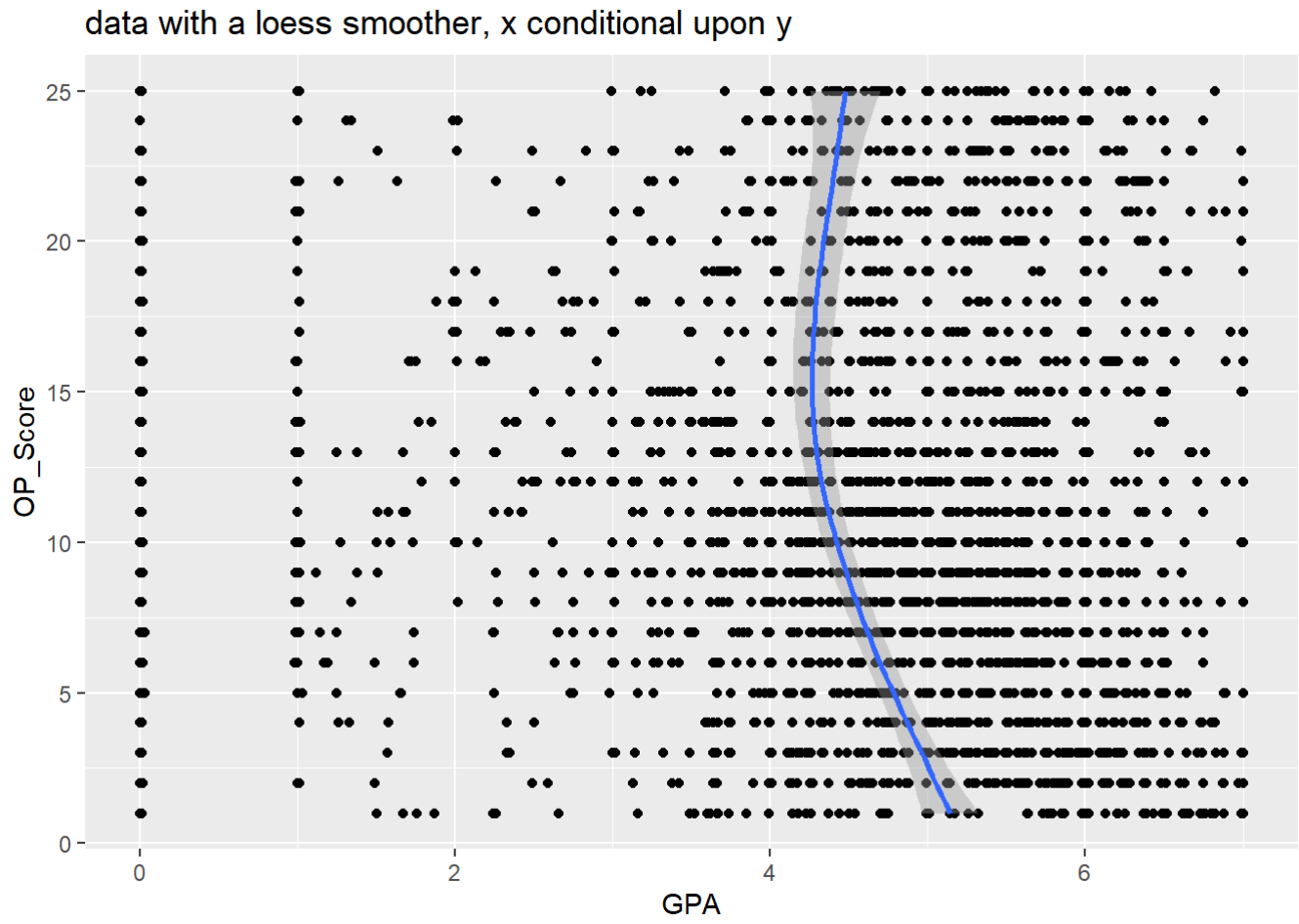
```
# create scatterplot of GPA vs OP_Score
p <- ggplot(data=data, aes(x=GPA, y=OP_Score)) + geom_point()

# create loess smoother to scatterplot
p + geom_smooth()                          + ggtitle("data with a loess smoother")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
# create loess smoother, x conditional upon y to scatterplot
p + geom_smooth(orientation = "y")         + ggtitle("data with a loess smoother, x condition
al upon y")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



From these graphs, a lower (numerically speaking) OP_Score can somewhat correlate to a higher GPA. Do note that the weighting on this trend isn't strong. It can also be seen that those with a lower (numerically speaking) OP_Score, tend to score a higher GPA than those with a median OP_Score, as seen in `data with a loess smoother, x condition upon y` graph. This may be due to how International students OP_Score is taken - as a default 25 - or indicate that students with a higher (numerically) OP_Score perform better than those with a median OP_Score.

It can also be seen that on the `data with a loess smoother` , where students with a GPA towards 0, can be seen to have achieved a higher OP. Although it could be that this occurrence is due to students who have yet to score a GPA, due to being in the middle of their first-year semester.

Task 5: Linear regression

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

```
# extract GPA list
GPA = data$GPA
# extract Achieved_Credit_Points list
Achieved_Credit_Points = data$Achieved_Credit_Points

# correlate
cor(Achieved_Credit_Points, GPA)
```

```
## [1] 0.4897386
```

```
# extract Failed_Credit_Points
Failed_Credit_Points = data$Failed_Credit_Points

# correlate
cor(Failed_Credit_Points, GPA)
```

```
## [1] -0.4737265
```

```
# extract OP_Score
OP_Score = data$OP_Score

# correlate
cor(OP_Score, GPA)
```

```
## [1] -0.1172122
```

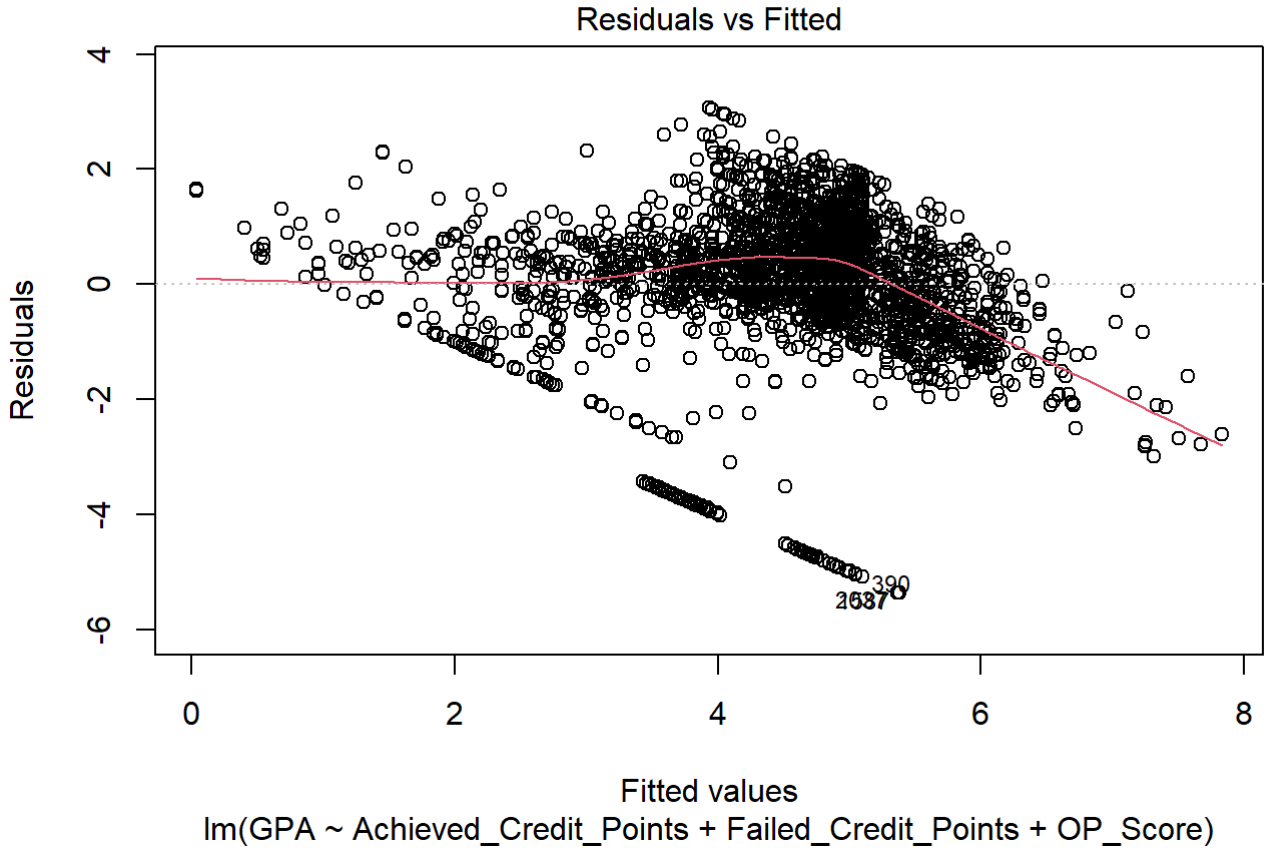
The predictor chosen would need to be a strong data value that could have significant impact on the analysis. The strongest that come to mind are the `OP_Score`, `Achieved_Credit_Points`, and `Failed_Credit_Points`. We can see using `cor()` above, that the `Achieved_Credit_Points` and `Failed_Credit_Points` have a high correlation with the `GPA`. There are four assumptions associated with a linear regression model. These are:

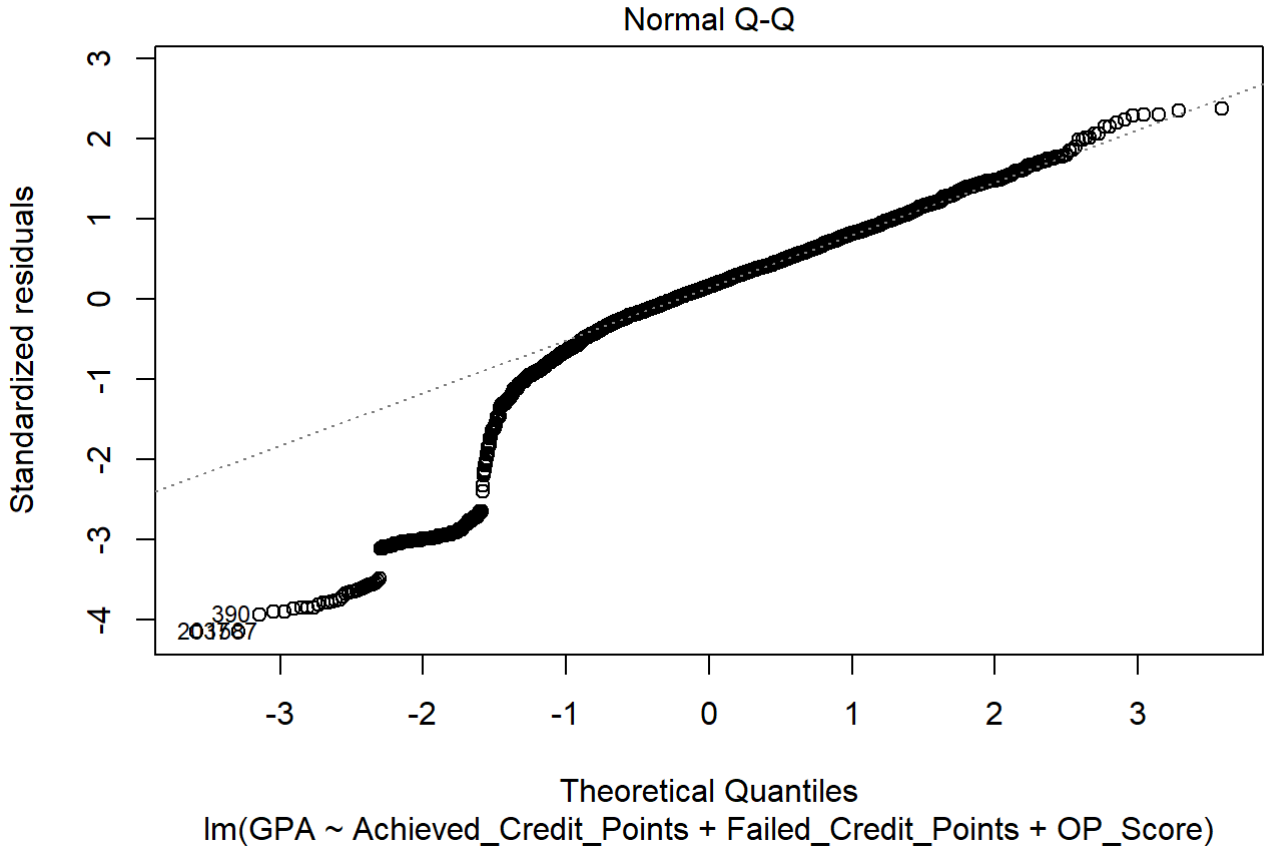
- Linearity: we assume the relationship between `x` and the mean of `y` is linear
- Homoscedasticity: we assume the variance of residual is the same of any value of `x`
- Independence: we assume observations are independent of each other
- Normality: we assume for any fixed value of `x`, `y` is normally distributed

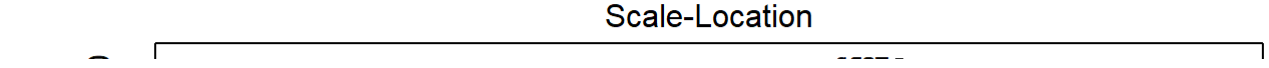
```
# create the linear model of GPA by all
gpa.lm <- lm(GPA ~ Achieved_Credit_Points + Failed_Credit_Points + OP_Score, data=data)

# print summary of model
summary(gpa.lm)
```

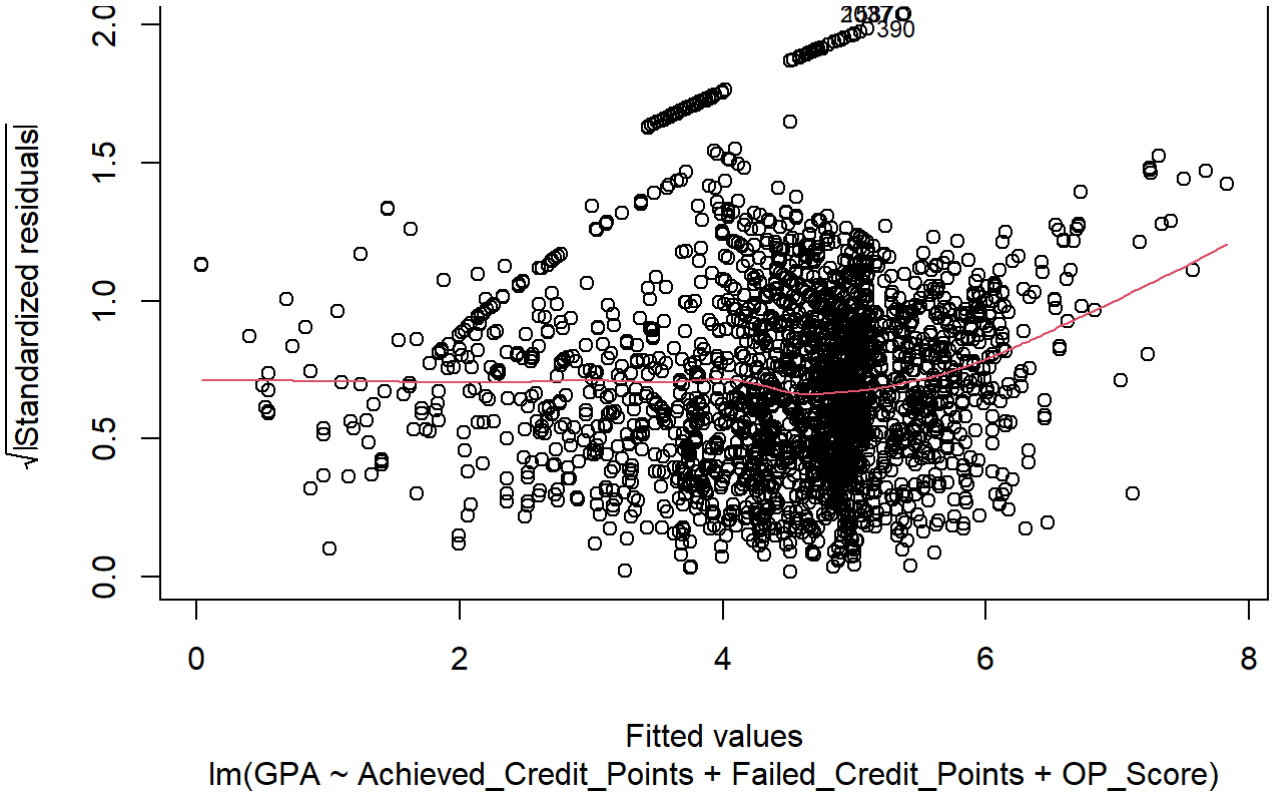
```
##
## Call:
## lm(formula = GPA ~ Achieved_Credit_Points + Failed_Credit_Points +
##     OP_Score, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3690 -0.3933  0.1977  0.7528  3.0673
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             4.0405029  0.0644439  62.698  < 2e-16 ***
## Achieved_Credit_Points  0.0112393  0.0004521  24.861  < 2e-16 ***
## Failed_Credit_Points   -0.0345811  0.0015426 -22.418  < 2e-16 ***
## OP_Score               -0.0244005  0.0036964  -6.601 4.81e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.293 on 2996 degrees of freedom
## Multiple R-squared:  0.3625, Adjusted R-squared:  0.3618
## F-statistic: 567.8 on 3 and 2996 DF,  p-value: < 2.2e-16
```

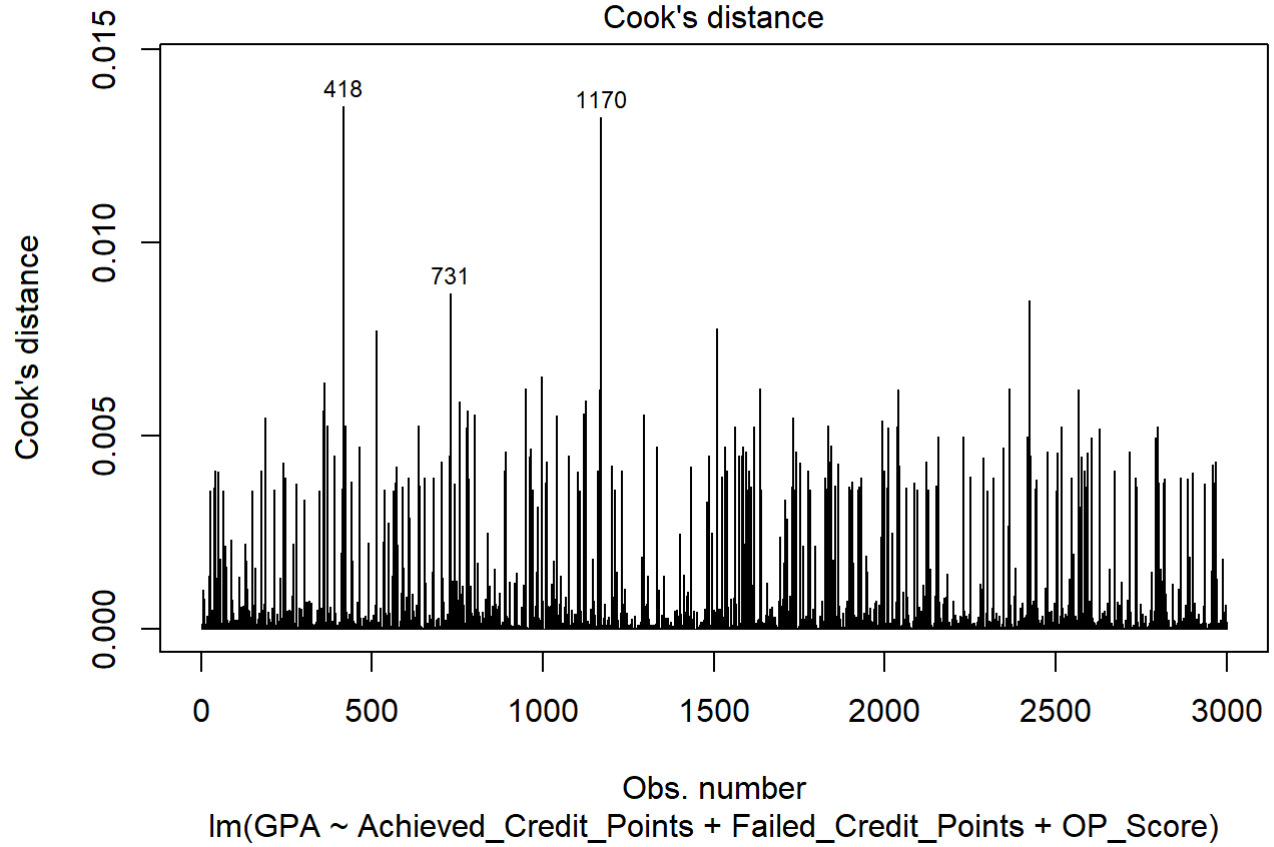
```
# plot model graphs
plot(gpa.lm, which=1:4)
```


Residuals vs Fitted
Residuals
Fitted values
lm(GPA ~ Achieved_Credit_Points + Failed_Credit_Points + OP_Score)


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(GPA ~ Achieved_Credit_Points + Failed_Credit_Points + OP_Score)


Scale-Location





The summary output overviews a brief summary of the residuals and estimated regression results. The residuals do not appear to be strongly symmetrical with the minimum residuals being nearly twice as much as the maximum.

The coefficients show the resultants for each property by `GPA`.

The `Intercept Estimate` is the expected value of `GPA`, taking into consideration all three tested properties. The `Estimate`'s of the three properties show the effect each property has on the `GPA` of our model. Negative values show that property negatively affects the `GPA`, whilst positive values show it will positively affect the

GPA. Naturally, failing units (increasing `Failed_Credit_Points`) would decrease a students `GPA`. The value tells us by how much a property change will affect the `GPA`.

The `Std. Error` measures the average amount that the estimates vary from the actual mean value of the resultant variable. In other words, the expected difference that each property may vary. From this we can see a relatively low `Std. Error` for each property. From this we can see that the `Std. Error` of `Achieved_Credit_Points` varies very little.

From the `t value` we can see a measure of how many standard deviations the coefficient is away from 0. Ideally we want each property to be as far away from 0 - as this indicates we can reject the null hypothesis - meaning we can declare a relationship between `GPA` and a particular property. We can see in the coefficient output that `Achieved_Credit_Points` and `Failed_Credit_Points` are the most far away from 0, which could indicate a relationship may exist.

The `Pr(>|t|)` coefficient describes the probability of observing any value equal or larger that `t`. A smaller `p-value` indicates that observing a relationship between the predictor and resultant due to chance is unlikely. In the model above, the `Achieved_Credit_Points` and `Failed_Credit_Points` are the closest to 0. A small p-value for the intercept and slop indicates that we can reject the null hypothesis allowing us to conclude that there may be a relationship between these properties.

The `Residuals Vs Fitted` graph shows that the residual spread widens as we move towards the center. The data spread also grows smaller as we move left to right. There are a few potential outliers that are clumped together at the bottom of the residuals. Our model is shaped in the middle of the fitted values, which may cause potential issues. However the majority of the shaped data is around the 0 line without any particularly large residuals, so trimming the data of the various outliers may be a potential possibility for refinement of this model.

The `Normal Q-Q` plot shows our model is predominantly normally distributed, however the data spikes toward the lower left of the graph, which may affect results from the linear model. This may be skewed because our data is predominantly students with a GPA of 4 or greater, the outliers all sit below a GPA of 4.

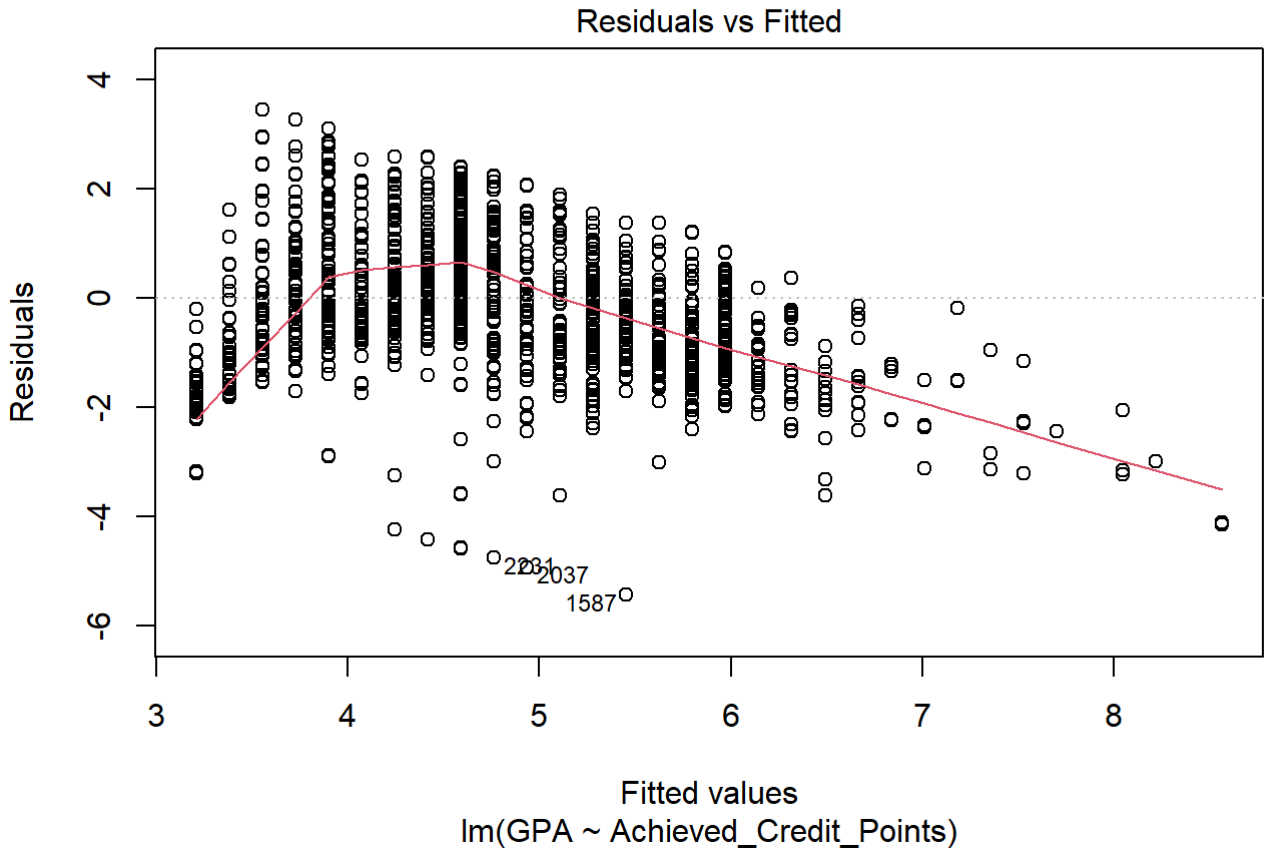
The `Scale-Location` plot shows our residuals aren't spread equally along the predictor range. The first three quadrants show homoscedasticity, having a general uniform variance. In the last quadrant the data is heteroscedastic (non-uniform variance), skewing as the residuals narrow.

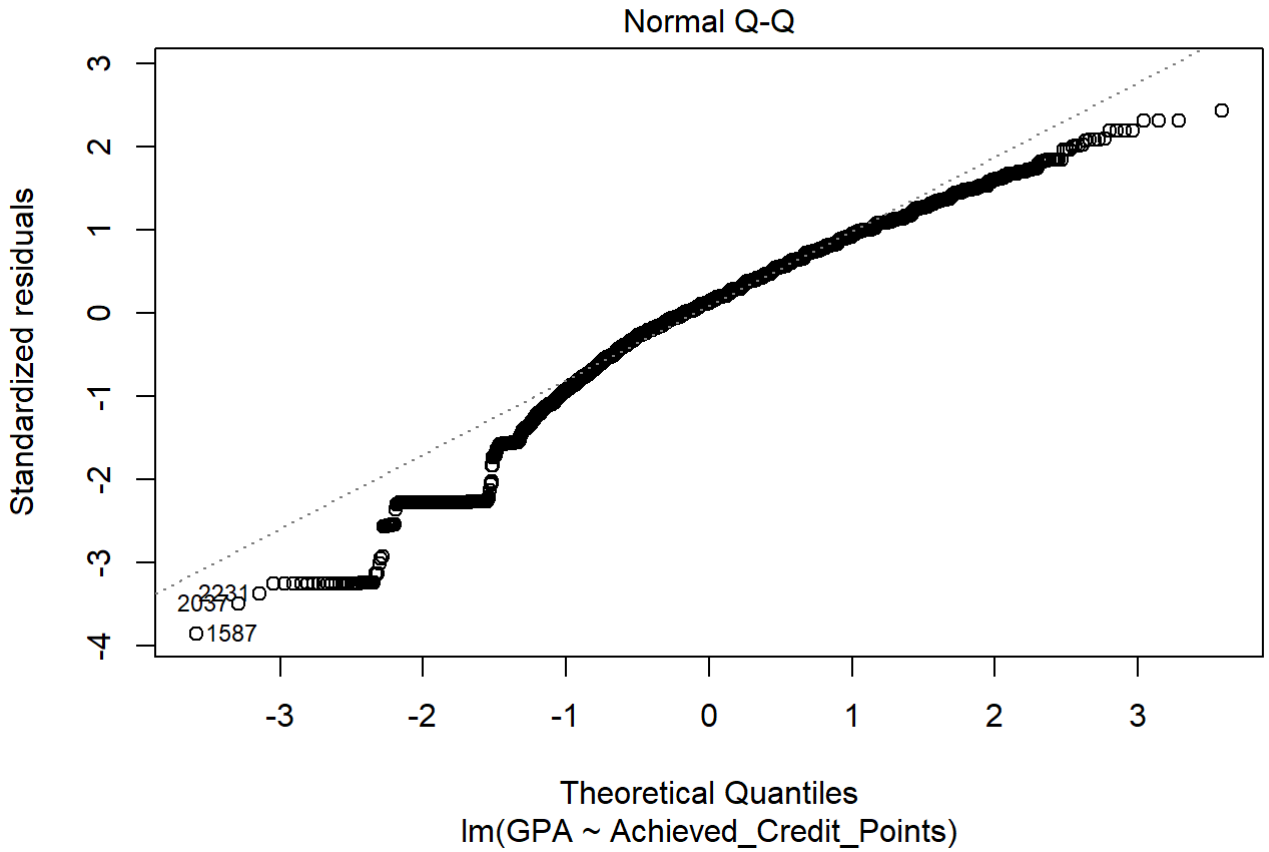
The `Cook's distance` graph shows outliers 418, 731, and 1170, which could be influencing the models regression. Removing them would likely noticeably alter the regression results.

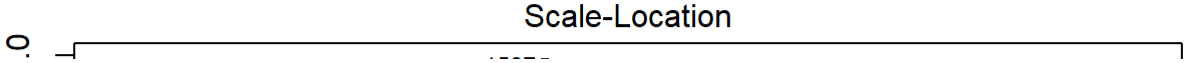
```
# create the linear model of GPA by Achieved_Credit_Points
gpa_achieved.lm <- lm(GPA ~ Achieved_Credit_Points, data=data)

# print summary of model
summary(gpa_achieved.lm)
```

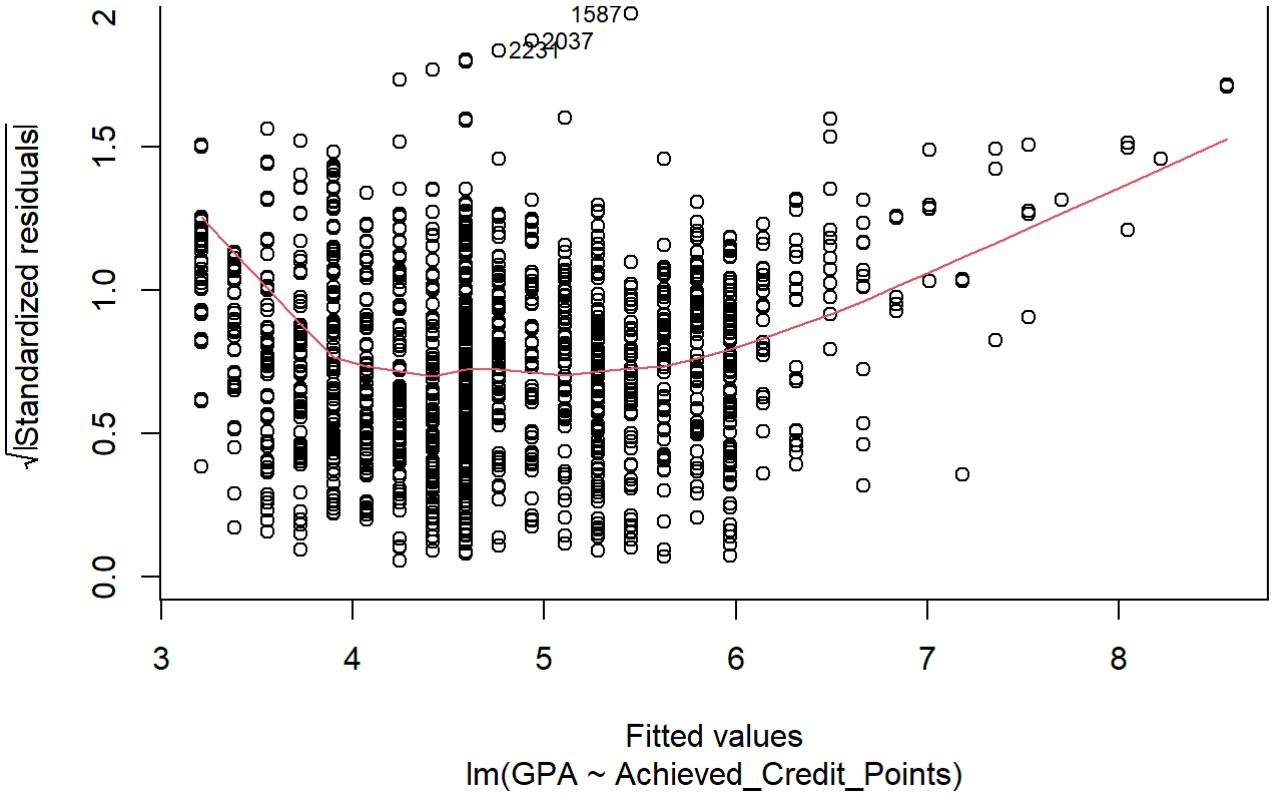
```
##
## Call:
## lm(formula = GPA ~ Achieved_Credit_Points, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4444 -0.7138  0.1792  0.9892  3.4455
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3.209102   0.050529   63.51   <2e-16 ***
## Achieved_Credit_Points 0.014393   0.000468   30.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.411 on 2998 degrees of freedom
## Multiple R-squared:  0.2398, Adjusted R-squared:  0.2396
## F-statistic: 945.9 on 1 and 2998 DF,  p-value: < 2.2e-16
```

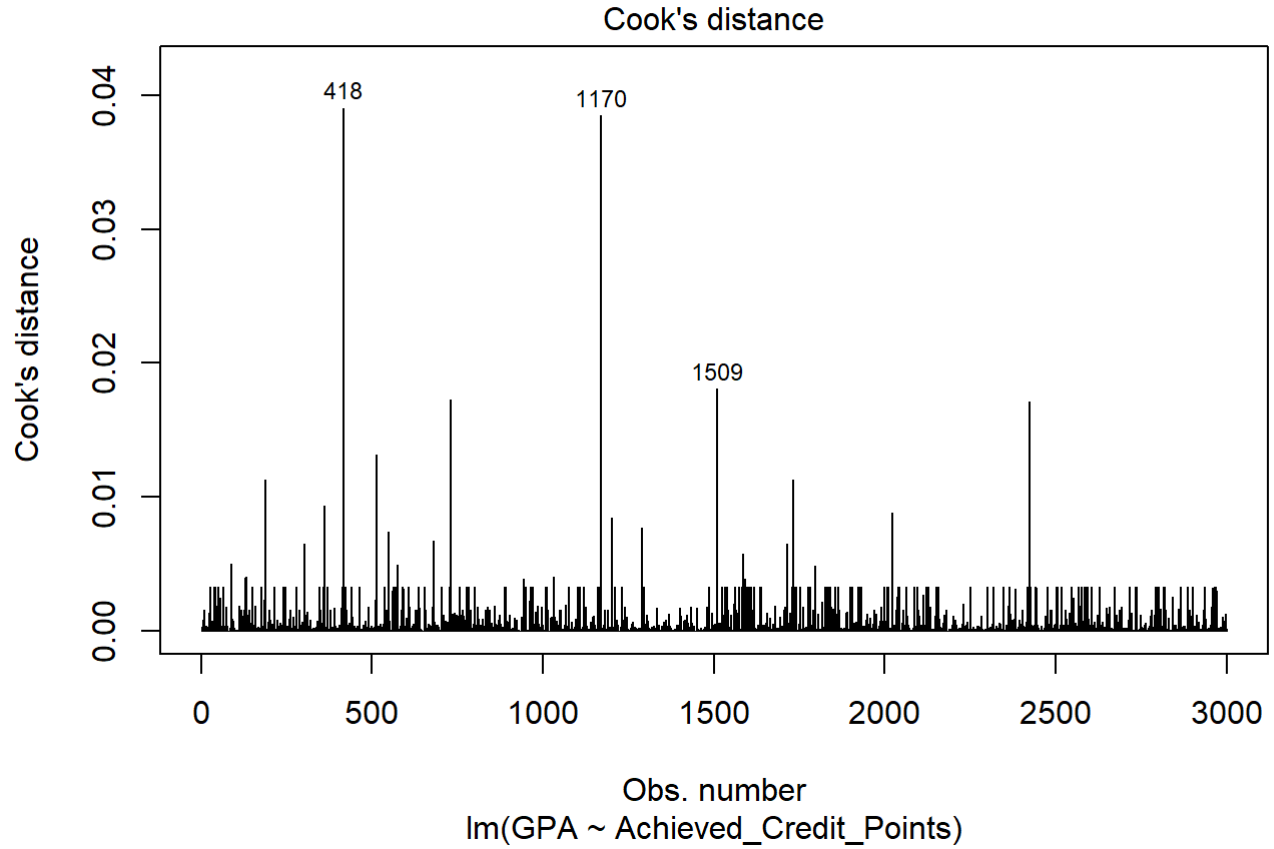
```
# plot model graphs
plot(gpa_achieved.lm, which=1:4)
```


Residuals vs Fitted
Residuals
Fitted values
lm(GPA ~ Achieved_Credit_Points)
2231
2037
1587


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(GPA ~ Achieved_Credit_Points)
2231
2037
1587


Scale-Location





We can see in this linear model of `GPA` by `Achieved_Credit_Points`, the `t-value` is quite high. From this we can infer that a relationship may exist between the two values. The `Std. Error` tells us that the models results won't vary to much. A small p-value for the intercept and slop indicates that we can reject the null hypothesis allowing us to conclude that there may be a relationship between these two properties.

The `Residual standard error` tells us we are not capable of perfectly predicting the resultant `GPA` given the `Achievied_Credit_Points`. The value given here is the average amount of deviation from the true regression line. Thus the `GPA` may differ by `1.411` from the predicted output.

The `Multiple R-squared` and `Adjusted R-squared` provide a measure of how well the model is fitting the data input. `0.2396` indicates that roughly `23%` of the variance found in the resultant `GPA` can be explained by the `Achieved_Credit_Points`. A number near zero repesents a regression that does not explain the variance in the result well.

The `F-statistic` indicates whether there is a strong relationship between the predictor and the resultant `GPA`. A large `F-statistic` such as `945.9` allows us to ascertain that there may be a relationship between the predictor and the resultant `GPA`. Due to a large dataset of 3000 students, a `F-statistic` of such difference from 1 is sufficient enough to indicate a relationship between the two properties.

The `Residuals Vs Fitted` graph shows that the residuals get smaller as we move left to right. There are a few potential outliers that are clumped together at the bottom of the residuals. Our model is shaped towards the left of the fitted values and the higher end of the residuals, which may cause potential issues. However the majority of the shaped data is around the 0 line without any particularly large residuals, so trimming the data of the various outliers may be a potential possibility for refinement of this model.

As shown above we can see the `Normal Q-Q` graph. Given more data for students of a GPA of less than 4, or by trimming the outliers we may result in a more normally distributed plot. However, using the `Achieved_Credit_Points` we get a *predominantly* normalised graph.

The `Scale-Location` plot shows our residuals aren't spread equally along the predictor range. Apart from the section section between 4 and 6 of the fitted values, the data is heteroscedastic (non-uniform variance), skewing as the residuals narrow.

The `Cook's distance` graph shows outliers 418, 1170, and 1509, which could be influencing the models regression. Removing them would likely noticeably alter the regression results.

Obviously the model given here is not optimised. We can improve the models output, by trimming outliers that skew the result. Investigating new variables to include, transforming variables and comparing various models could lead us to a stronger relationship. With this, stronger predictive analytics could be run to determine a students `GPA`.

Task 6: Logistic regression

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

```
# create the logistic model of Attrition by all possibilities
attr.glm <- glm(Attrition ~ ., data=data, family="binomial")

# summarise the Attrition linear model
summary(attr.glm)
```

```
##
## Call:
## glm(formula = Attrition ~ ., family = "binomial", data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5059   0.2048   0.4427   0.5838   1.8453
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    0.952491   0.411158   2.317  0.02053 *
## Degree_TypeSingle             -0.781539   0.279440  -2.797  0.00516 **
## Achieved_Credit_Points         0.017028   0.001686  10.101  < 2e-16 ***
## Attendance_TypePart Time       1.456609   0.244047   5.969 2.39e-09 ***
## Age                           -0.025167   0.009140  -2.753  0.00590 **
## Failed_Credit_Points          -0.016833   0.003151  -5.343 9.16e-08 ***
## International_studentYes       0.640910   0.222607   2.879  0.00399 **
## First_in_familyYes             0.027904   0.110754   0.252  0.80108
## GenderM                        0.158249   0.113899   1.389  0.16472
## GPA                            0.061937   0.039610   1.564  0.11789
## OP_Score                      -0.005148   0.008663  -0.594  0.55233
## Socio_Economic_StatusLow      -0.099750   0.155055  -0.643  0.52002
## Socio_Economic_StatusMedium    0.050424   0.124444   0.405  0.68534
## Teaching_Period_AdmittedSEM-2  0.453131   0.157703   2.873  0.00406 **
## FacultyFaculty of Education    0.614264   0.258418   2.377  0.01745 *
## FacultyFaculty of Health       0.177929   0.160017   1.112  0.26616
## FacultyFaculty of Law         -0.374510   0.209556  -1.787  0.07391 .
## FacultyQUT Business School     0.509131   0.196877   2.586  0.00971 **
## FacultySci and Eng Faculty     0.377837   0.167826   2.251  0.02436 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2825.3  on 2999  degrees of freedom
## Residual deviance: 2297.5  on 2981  degrees of freedom
## AIC: 2335.5
##
## Number of Fisher Scoring iterations: 5
```

From the `Deviance Residuals` we can infer if the proposed model has a good fit, from the deviance being small. Since the median deviance residual is close to zero, this means that our model is not biased. However, note the `Min` and `Max` values, these could skew results for predictions that fall outside of the middle 2 quadrants because of the variance on both ends.

From the coefficients given, we can use the `Estimate` to indicate the expected `Attrition` value given various properties of the data set. From this, negative and positive values show how that property affects the `Attrition`. Naturally, those `GPA` increasing would also increase the probability of a student retaining their studies. The value tells us by how much a property change will affect the `Attrition`. We can see that the `Part Time Attendance_Type` has a greater change than most other `Estimate`s.

From the `Std. Error` we can determine the average amount of variance each property has. We can see `Achieved_Credit_Points` has very small variance and won't differ too much. This just ensures that each time the model is run, the `Estimate` won't differ much from what is given now.

From the `z value`, we want significant properties to be as far away from 0 - as this indicates we can reject the null hypothesis - meaning we can infer a relationship between `Attrition` and that particular property. We can see in the coefficient output that `Achieved_Credit_Points`, `Failed_Credit_Points` and `Attendance_Type(Part Time)` are the furthest away from 0, which could indicate a relationship may exist for each property with the `Attrition`. The negative values indicate it negatively affects a students `Attrition` meaning `Failed_Credit_Points` can indicate that a student is more likely to discontinue their studies, whilst students who study part time and accrue more credit points are potentially more likely to continue their studies.

`Degree_TypeSingle`, `Age`, `International_studentYes`, `Teaching_Period_AdmittedSEM-2`, `FacultyFaculty of Education`, `FacultyQUT Business School`, and `FacultySci and Eng Faculty` may be some other properties to investigate as to why they are so signficant, as well as if we can infer a relationship between them and a student's `Attrition`.

The `Pr(>|t|)` coefficient describes the probability of observing any value equal or larger that `z`. A smaller `p-value` indicates that observing a relationship between the predictor and resultant due to chance is unlikely. In the model above, the `Achieved_Credit_Points`, `Failed_Credit_Points`, and `Attendance_TypePart Time` are the closest to 0. A small p-value for the intercept and slop indicates that we can reject the null hypothesis allowing us to conclude that there may be a relationship between these properties. Further investigation may indicate the validity on these properties or whether we should reject them.

In regards to the `Null deviance` and `Residual deviance`, this can indicate whether the model is well trained. A low `Null deviance` shows how well the result is predicted by the model using only the `Intercept`. For this model our `Null deviance` is quite large. The `Residual deviance` tells us how well we can predict our output using the intercept and various property inputs. The `Residual deviance` is quite large, which may cause issues with our model. The difference between the `Null deviance` and `Residual deviance` is quite small comparatively too, which tells us our input variables aren't that helpful for predicting `Attrition` results.

The `AIC` measure indicates the complexity and fit of a model. A low `AIC` can indicate a model with a low complexity and a good fit. However this is generally used for comparing models trained on the same dataset. The model with the lower `AIC` describes the variance in the data better, and is most likely a better fit.

The `Fisher Scoring iterations` shows that the model converged after 5 iterations, indicating that it didn't run into any significant issues when running.

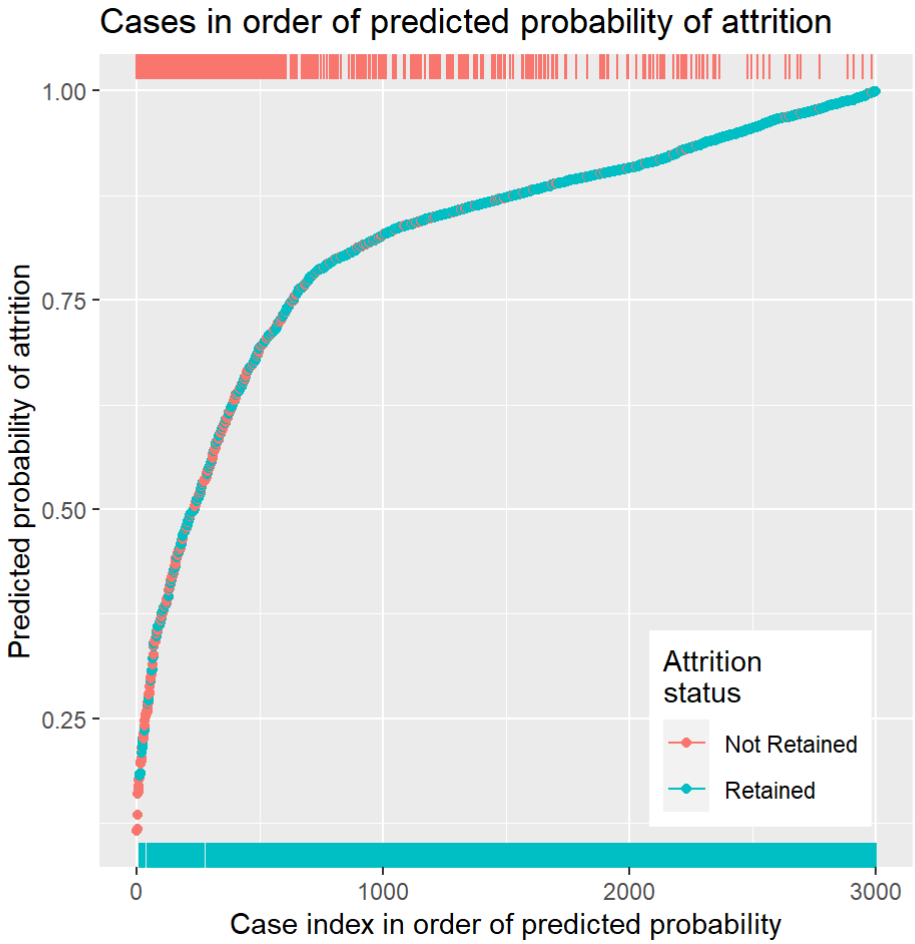
```
confint(attr.glm)
```

```
## Waiting for profiling to be done...
```

```
##                                    2.5 %       97.5 %
## (Intercept)                     0.15639694  1.771897271
## Degree_TypeSingle              -1.36136335 -0.260038776
## Achieved_Credit_Points          0.01378878  0.020400290
## Attendance_TypePart Time        0.99818676  1.958433931
## Age                            -0.04292232 -0.007069094
## Failed_Credit_Points           -0.02305002 -0.010691035
## International_studentYes        0.21907864  1.094305659
## First_in_familyYes             -0.18852732  0.245832441
## GenderM                        -0.06463420  0.382064633
## GPA                            -0.01606352  0.139308053
## OP_Score                       -0.02206968  0.011906371
## Socio_Economic_StatusLow       -0.40266551  0.205601432
## Socio_Economic_StatusMedium    -0.19475779  0.293354898
## Teaching_Period_AdmittedSEM-2   0.14854227  0.767319469
## FacultyFaculty of Education     0.12162588  1.137742025
## FacultyFaculty of Health       -0.13627751  0.491463230
## FacultyFaculty of Law          -0.78394707  0.038343719
## FacultyQUT Business School      0.12649363  0.899136911
## FacultySci and Eng Faculty      0.04876881  0.707119383
```

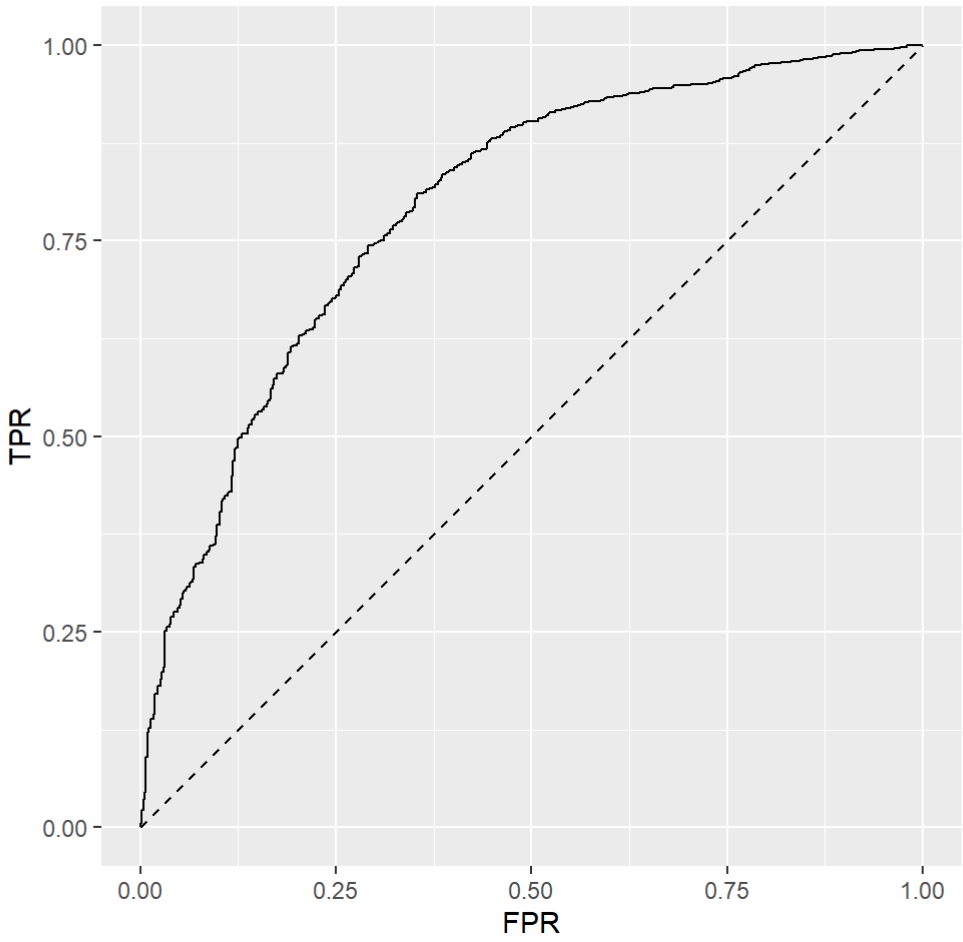
The confidence intervals tell us where the model has the most confidence for predictions based on each property. It tells us how well the model/data represents the population. We can see that the confidence of many properties has a rather slim span. With most properties spanning one or two quadrants. Collecting more data in areas where the dataset doesn't lie would allow the model to generate more confidence for other quadrants.

```
# generate predictions based on dataset
tibble(
    prob.of.attr =attr.glm$fitted.values,
    Attrition    =data$Attrition,
) -> attr.glm.pred

# arrange predictions into ranked predictions
attr.glm.pred %>%
  arrange(prob.of.attr) %>%
  mutate(index=row_number()) -> attr.glm.ranked

# plot ranked predictions
ggplot(data=attr.glm.ranked, aes(x=index, y=prob.of.attr)) +
  geom_point(aes(color=Attrition)) +
  geom_rug(data=filter(attr.glm.ranked, Attrition=="Retained"), sides="b", aes(color=Attritio
n)) +
  geom_rug(data=filter(attr.glm.ranked, Attrition!="Retained"), sides="t", aes(color=Attritio
n)) +
  labs(
    title="Cases in order of predicted probability of attrition",
    x="Case index in order of predicted probability",
    y="Predicted probability of attrition"
    )+
  scale_color_discrete(name="Attrition\nstatus") +
  # http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/#changing-the-position-of-the-legend
  theme(legend.justification=c(1,0), legend.position=c(0.95,0.05), aspect.ratio=1)
```



From this we can see how well our model separates retained and not retained students. We can see that the majority of the `Not Retained` students lie towards the left of the plot. However, it can be noted that the spread of `Retained` students is fairly even across the entirety of the dataset, which may cause issues in predictions as the model may have little confidence to make accurate predictions based on certain input properties. This could result in False Positive results due to a majority of the data spread being Retained.

```
# function to generate ROC
simple.roc <- function(labels, scores){
  ordered.scores <- order(scores, decreasing=TRUE)
  labels <- labels[ordered.scores]
  tibble(
    TPR=c(0,cumsum(labels))/sum(labels),
    FPR=c(0,cumsum(!labels))/sum(!labels),
    labels=c(NA,labels),
    score=c(Inf, scores[ordered.scores])
  )
}

# generate ROC plot using Attrition predictions
attr.glm.ROC <- simple.roc(attr.glm.pred$Attrition=="Retained", attr.glm.pred$prob.of.attr)

# generate diagonal line
diagonal <- data.frame(FPR=c(0,1), TPR=c(0,1))

# generate ROC plot
ggplot(data=attr.glm.ROC, aes(x=FPR, y=TPR)) +
  geom_step() +
  coord_equal(xlim=c(0,1), ylim=c(0,1)) +
  geom_line(data=diagonal, lty=2) +
  labs(xlab="False Positive Rate (FPR)", ylab="True Positive Rate (TPR)")
```



The ROC curve above shows the performance of the model at all thresholds of `True Positive` and `False Positive` rates. A model whose predictions are 100% wrong would have an AUC (Area under curve) of 0.0. On the otherhand, a model whose predictions are 100% correct would have an AUC of 1.0. Models that

give a curve close to the top-left can indicate a better goodness of fit, as well as potentially more accurate prediction performance. Those curves closer to the dashed line can be seen as a worse performance and thus less accurate in predictions.

This can be used to compare different models, and summarise the performance of each using a single measure. In this ROC curve above, we can see that the model curves less that what would sit for a

Task 7: Using `R` for data science

7. In completing this Assessment, use the principles and approaches espoused in R for Data Science (https://r4ds.had.co.nz/) by Garrett Grolemund and Hadley Wickham to ensure that your work can be run, reproduced and understood by the Teaching Team.