**Data606-Project:**

**Statistical Analysis of Education**

**& Career Success**

Group 1:

Gautham Nagaraj (30273439)

Angela Li (30024371)

Prichelle Lal (30261130)

**June 12th, 2025**

**Table of Contents**

1 Introduction

1.1 Purpose

1.2 Objective

1.3 Guiding Questions

1.4 Dataset

1.4.1 Student Information

1.4.2 Academic Performance

1.4.3 Skills & Extra-curriculars

1.4.4 Career Outcomes

1.4.5 Data Preview

2 Methodology

3 Analysis

3.1 Data Cleaning

3.1.1 University Ranking

3.1.2 Number of Internships Completed(0-4), Projects Completed (0-9), and Certifications, Soft-Skill Score, and Networking Score

3.2 Exploratory Data Analysis

3.2.1 Histogram - Distribution of Starting Salary

3.2.2 Relative Frequency Bar Chart - Distribution of gender across job levels

3.2.3 Heatmap - how the number of internships impact the job offers received

3.2.4 Stratified Sampling

3.3 Multiple Linear Regression

3.3.1 Multicollinearity & Pairs Plot

3.3.2 VIF for Multicollinearity

3.4 Stepwise Model Selection

3.4.1 Reduced Model

3.5 Regression Tree Model

3.6 Classification

3.7 Linear Discriminant Analysis

3.7.1 Shapiro-Wilk Normality Test

3.7.2 Quantile-Quantile plot

3.7.3 Interquartile Range (IQR)

3.7.4 Box-cox Transformation

3.7.5 Levene’s test for Equal Variance

3.8 Quadratic Discriminant Analysis

3.8.1 Cross Validation

3.9 Classification Trees

3.9.1 Decision Trees

3.9.2 Tree Pruning

3.10 Sampling

3.10.1 Simple random sample without replacement

3.10.2 Stratified Sampling

3.10.3 Cluster Sampling

3.11 Re-implementing methods with Stratified Sampling

3.11.1 Quadratic Discriminant Analysis

3.11.2 Classification Tree

3.11.3 Logistic Regression

4 Conclusion

4.1 Summary

4.2 Key Findings and Insights

**1 Introduction**

**1.1 Purpose:**

In today's competitive job market, understanding the pathways from education to career success is more crucial than ever. Students invest significant time and resources into their academic journey with the hope that strong educational performance will lead to favorable career outcomes. However, the relationship between education and career achievement is influenced by a multitude of factors, including soft skills, internships, networking, and field of study. This dataset captures the educational backgrounds, skillsets, and career results of 5,000 individuals, providing a rich foundation to explore these connections. By analyzing variables such as GPA, SAT scores, university rankings, certifications, and job offers, we aim to build predictive models and uncover the key drivers of starting salaries, promotions, and overall career satisfaction.

Our goal is to apply sampling, regression estimation, and categorical data analysis to uncover patterns and relationships within the education and career success dataset. Using methods such as stratified sampling, ratio estimation, logistic regression, and classification, we aim to identify key predictors of career outcomes such as job offers, starting salary, and satisfaction. These analyses will allow us to make informed inferences, build predictive models, and ultimately gain a deeper understanding of how educational and experiential factors influence career trajectories.

**1.2 Objective:**

The primary objective of this study is to explore how educational performance, skill development, and extracurricular experiences influence early career outcomes. Using a combination of statistical methods including regression estimation and classification techniques, we aim to: Identify the most significant predictors of number of job offers, and career satisfaction, build predictive models that can estimate starting salary, and use sampling techniques to make inferences about the broader student population. By doing so, we aim to provide insights that could help students and professionals optimize educational pathways for successful career outcomes.

**1.3 Guiding Questions:**

These guiding questions will drive the analysis and modeling phases of the project:

1. Which factors/grades during a student's education impact their Starting\_Salary?
2. Do Career\_Satisfaction and Work\_Life\_Balance relate to student characteristics and educational outcomes?
3. Are there distinct clusters of students with similar profiles in terms of education, skills, and career outcomes?

**1.4 Dataset:**

Data Source: <https://www.kaggle.com/datasets/adilshamim8/education-and-career-success>

The Education and Career Success dataset contains detailed information about individuals' educational backgrounds and corresponding career outcomes. It includes 1000 rows and 8 columns, with each row representing an individual. The dataset captures variables such as gender, age, education level, field of study, years of experience, current job title, annual salary, and career satisfaction. This structured data allows for analysis of how different educational factors relate to various aspects of professional success. The columns included are as follows:

### *1.4.1 Student Information*

| **Student\_ID** | A unique identifier assigned to each student in the dataset |
| --- | --- |
| **Age** | The age of the student, ranging from 18 to 30 years old. |
| **Gender** | The gender of the student, which can be Male, Female, or Other. |

### *1.4.2 Academic Performance*

| **High\_School\_GPA** | The student's Grade Point Average (GPA) from high school, measured on a 4.0 scale. |
| --- | --- |
| **SAT\_Score** | The student's SAT standardized test score, ranging between 900 and 1600. |
| **University\_Ranking** | The ranking of the university attended by the student, ranging from 1 (highest) to 1000 (lowest). |
| **University\_GPA** | The GPA obtained by the student during university education, measured on a 4.0 scale. |
| **Field\_of\_Study** | The student's major or discipline (ie, Computer Science, Medicine, Business) |

### *1.4.3 Skills & Extracurricular Activities*

| **Internships\_Completed** | The number of internships completed by the student during their academic journey, ranging from 0-4 |
| --- | --- |
| **Projects\_Completed** | The number of academic or personal projects completed by the student, ranging from 0-9 |
| **Certif**  **ications** | The number of additional certifications earned by the student, ranging from 0-5 |
| **Soft\_Skills\_Score** | A rating of the student’s soft skills, between 1-10 |
| **Networking\_Score** | A rating that reflects the student’s professional networking and connections, between 1-10 |

### *1.4.4 Career Outcomes*

| **Job\_Offers** | The number of job offers received by the student, ranging from 0-5 |
| --- | --- |
| **Starting\_Salary** | The starting salary of the student in USD |
| **Career\_Satisfaction** | A career satisfaction rating from the student, between 1-10 |
| **Years\_to\_Promotion** | The time it took to receive the first promotion , range of 1-5 years |
| **Current\_Job\_Level** | Career level (ie, Entry, Mid, Senior, Executive) |
| **Work\_Life\_Balance** | A work-life balance rating, ranging between 1-10 |
| **Entrepreneurship** | Whether the individual started a business (Yes/No) |

*1.4.5 Data Preview*

The dataset was synthetically generated using real-world education and career trends.

education\_data = **read.csv**('https://raw.githubusercontent.com/Gautham-Nagaraj/Data606-Project/refs/heads/main/education\_career\_success.csv')  
**head**(education\_data,10)

## Student\_ID Age Gender High\_School\_GPA SAT\_Score University\_Ranking  
## 1 S00001 24 Male 3.58 1052 291  
## 2 S00002 21 Other 2.52 1211 112  
## 3 S00003 28 Female 3.42 1193 715  
## 4 S00004 25 Male 2.43 1497 170  
## 5 S00005 22 Male 2.08 1012 599  
## 6 S00006 24 Male 2.40 1600 631  
## 7 S00007 27 Male 2.36 1011 610  
## 8 S00008 20 Male 2.68 1074 240  
## 9 S00009 24 Male 2.84 1201 337  
## 10 S00010 28 Male 3.02 1415 138  
## University\_GPA Field\_of\_Study Internships\_Completed Projects\_Completed  
## 1 3.96 Arts 3 7  
## 2 3.63 Law 4 7  
## 3 2.63 Medicine 4 8  
## 4 2.81 Computer Science 3 9  
## 5 2.48 Engineering 4 6  
## 6 3.78 Law 2 3  
## 7 3.83 Computer Science 0 1  
## 8 2.84 Computer Science 1 5  
## 9 3.31 Business 2 3  
## 10 2.33 Computer Science 1 5  
## Certifications Soft\_Skills\_Score Networking\_Score Job\_Offers Starting\_Salary  
## 1 2 9 8 5 27200  
## 2 3 8 1 4 25000  
## 3 1 1 9 0 42400  
## 4 1 10 6 1 57400  
## 5 4 10 9 4 47600  
## 6 2 2 2 1 68400  
## 7 3 3 3 2 55500  
## 8 5 5 1 2 38000  
## 9 0 5 5 2 68900  
## 10 3 10 2 0 58900  
## Career\_Satisfaction Years\_to\_Promotion Current\_Job\_Level Work\_Life\_Balance  
## 1 4 5 Entry 7  
## 2 1 1 Mid 7  
## 3 9 3 Entry 7  
## 4 7 5 Mid 5  
## 5 9 5 Entry 2  
## 6 9 2 Entry 8  
## 7 7 4 Mid 3  
## 8 2 3 Entry 3  
## 9 2 2 Entry 2  
## 10 4 2 Senior 2  
## Entrepreneurship  
## 1 No  
## 2 No  
## 3 No  
## 4 No  
## 5 No  
## 6 Yes  
## 7 No  
## 8 No  
## 9 No  
## 10 No

### **2 Methodology**

A multiple linear regression model can be built using the independent variables mentioned above to predict the starting salary of students. There are various assumptions that need to be cleared in order to implement the model successfully.

1. Multicollinearity - The first step would be to check for multicollinearity between the predictors and ensure that the VIF score is under threshold
2. Linearity - Ensure that there is no linearity between the residuals as these indicate auto-correlation
3. Homoscedasticity - Ensure that the residuals have equal variance
4. Normality - Ensure that the residuals are normally distributed
5. Outliers - Inspect the data for outliers and check for leverage points

Once the assumptions are cleared and a model is obtained, we can use sampling to estimate the population averages and plug them into the model which would give us a realistic idea of what a student’s starting salary might be.

As there is a possibility that the assumptions may not be cleared we can use ‘Entrepreneurship’ as the dependent variable and use a classification based model to predict the variable, this is more feasible as classification models have fewer assumptions to clear. The model can be further improved using cross-validation.

Regression and Classification trees can be applied to better visualize and predict the variables (depending on the assumption being cleared).

**3 Analysis**

**3.1 Data Cleaning:**

Based on the above table, we need to ensure that categorical variables are handled appropriately when fitting the model. Some of the variables that appear to be continuous but are categorical are: University Ranking, Number of Internships Completed (0-4), Projects Completed (0-9), and Certifications, Soft-Skill Score, Networking Score.

*3.1.1. University Ranking*

* The difference between rank 1 and 2 might not be the same as between rank 999 and 1000 in terms of quality or impact. Thus, it is better to treat this variable as categorical rather than continuous. As the number of levels will be very high if we factor this variable, it is better to label it as ‘very high ranked’, ‘high ranked’, ‘low rank’.

First we will label this variable - University ranking - as shown below:

**library**(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

education\_data <- education\_data **%>%**  
 **mutate**(  
 University\_Ranking\_Category = **case\_when**(  
 University\_Ranking **>=** 1 **&** University\_Ranking **<=** 250 **~** "High ranked",  
 University\_Ranking **>** 250 **&** University\_Ranking **<=** 500 **~** "Moderately Ranked",  
 University\_Ranking **>** 500 **&** University\_Ranking **<=** 750 **~** "Low ranked",  
 University\_Ranking **>** 750 **~** "Very low ranked"  
 )  
 )  
**head**(education\_data,10)

## Student\_ID Age Gender High\_School\_GPA SAT\_Score University\_Ranking  
## 1 S00001 24 Male 3.58 1052 291  
## 2 S00002 21 Other 2.52 1211 112  
## 3 S00003 28 Female 3.42 1193 715  
## 4 S00004 25 Male 2.43 1497 170  
## 5 S00005 22 Male 2.08 1012 599  
## 6 S00006 24 Male 2.40 1600 631  
## 7 S00007 27 Male 2.36 1011 610  
## 8 S00008 20 Male 2.68 1074 240  
## 9 S00009 24 Male 2.84 1201 337  
## 10 S00010 28 Male 3.02 1415 138  
## University\_GPA Field\_of\_Study Internships\_Completed Projects\_Completed  
## 1 3.96 Arts 3 7  
## 2 3.63 Law 4 7  
## 3 2.63 Medicine 4 8  
## 4 2.81 Computer Science 3 9  
## 5 2.48 Engineering 4 6  
## 6 3.78 Law 2 3  
## 7 3.83 Computer Science 0 1  
## 8 2.84 Computer Science 1 5  
## 9 3.31 Business 2 3  
## 10 2.33 Computer Science 1 5  
## Certifications Soft\_Skills\_Score Networking\_Score Job\_Offers Starting\_Salary  
## 1 2 9 8 5 27200  
## 2 3 8 1 4 25000  
## 3 1 1 9 0 42400  
## 4 1 10 6 1 57400  
## 5 4 10 9 4 47600  
## 6 2 2 2 1 68400  
## 7 3 3 3 2 55500  
## 8 5 5 1 2 38000  
## 9 0 5 5 2 68900  
## 10 3 10 2 0 58900  
## Career\_Satisfaction Years\_to\_Promotion Current\_Job\_Level Work\_Life\_Balance  
## 1 4 5 Entry 7  
## 2 1 1 Mid 7  
## 3 9 3 Entry 7  
## 4 7 5 Mid 5  
## 5 9 5 Entry 2  
## 6 9 2 Entry 8  
## 7 7 4 Mid 3  
## 8 2 3 Entry 3  
## 9 2 2 Entry 2  
## 10 4 2 Senior 2  
## Entrepreneurship University\_Ranking\_Category  
## 1 No Moderately Ranked  
## 2 No High ranked  
## 3 No Low ranked  
## 4 No High ranked  
## 5 No Low ranked  
## 6 Yes Low ranked  
## 7 No Low ranked  
## 8 No High ranked  
## 9 No Moderately Ranked  
## 10 No High ranked

Next, We can remove the University\_Ranking column so that it is not used.

education\_data = education\_data[,**-**6]

*3.1.2 Number of Internships Completed(0-4), Projects Completed (0-9), and Certifications, Soft-Skill Score, and Networking Score*

* *Number of Internships Completed(0-4)* **-** While counts can sometimes be treated as continuous if the range is large, here, each number represents a distinct level of internship experience, making it more appropriate as an ordinal categorical variable.
* *Projects Completed(0-9)* - Similar to number of internships, it is more appropriate to treat this as categorical variable.
* *Certifications, Soft-Skill Score, Networking Score* - These variables are also to be treated as categorical due to lower counts.

These three variables can be converted to categorical data as shown below:

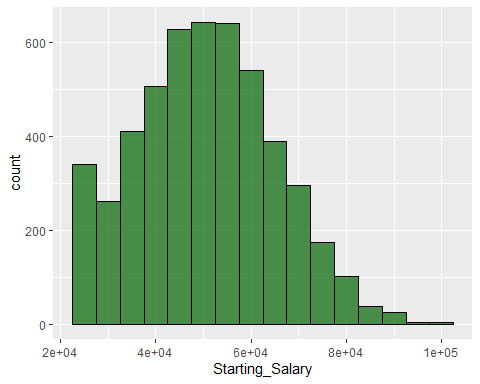
education\_data**$**Internships\_Completed <- **factor**(education\_data**$**Internships\_Completed)  
education\_data**$**Projects\_Completed <- **factor**(education\_data**$**Projects\_Completed)  
education\_data**$**Certifications <- **factor**(education\_data**$**Certifications)  
education\_data**$**Soft\_Skills\_Score <- **factor**(education\_data**$**Soft\_Skills\_Score)  
education\_data**$**Networking\_Score <- **factor**(education\_data**$**Networking\_Score)  
education\_data**$**Gender <- **factor**(education\_data**$**Gender)

**3.2 Exploratory Data Analysis:**

*3.2.1 Histogram - Distribution of Starting Salary*

This histogram is used to visualize the distribution of starting salarie**s** across all observations in the dataset. It helps identify the **c**entral tendency, spread, and skewness of the salary data, revealing common salary ranges and potential outliers.

**library**(ggplot2)  
**ggplot**(education\_data, **aes**(x = Starting\_Salary)) **+**  
 **geom\_histogram**(binwidth = 5000, fill = "darkgreen", color = "black", alpha = 0.7)



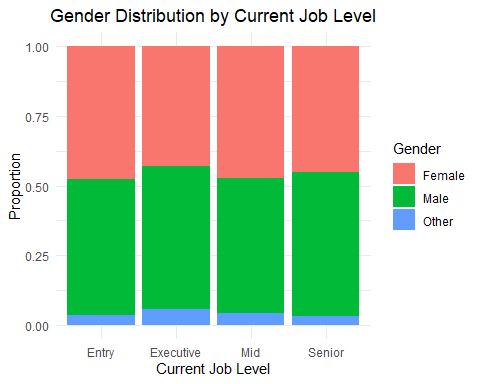
**labs**(  
 title = "Distribution of Starting Salary",  
 x = "Starting Salary ($)",  
 y = "Count / Density"  
 ) **+**  
 **theme\_minimal**() **+**  
 **theme**(plot.title = **element\_text**(hjust = 0.5))

## NULL

*3.2.2 Relative Frequency Bar Chart - Distribution of gender across job levels*

This analysis explores the proportion of males and females across different job levels to identify potential gender disparities in career advancement. Visualizing the distribution as proportions allows for a clearer comparison between groups regardless of sample size.

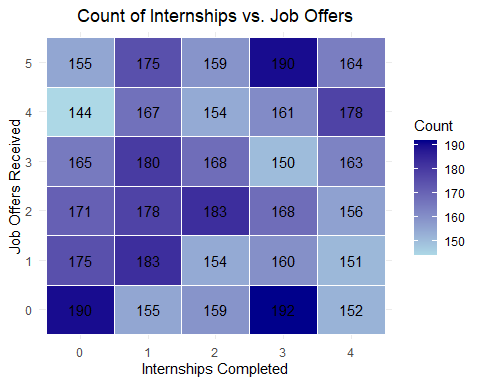
**library**(ggplot2)  
**ggplot**(education\_data, **aes**(x = Current\_Job\_Level, fill = Gender)) **+**  
 **geom\_bar**(position = "fill") **+**  
 **labs**(  
 title = "Gender Distribution by Current Job Level",  
 x = "Current Job Level",  
 y = "Proportion"  
 ) **+**  
 **theme\_minimal**() **+**  
 **theme**(plot.title = **element\_text**(hjust = 0.5))



*3.2.3 Heatmap - How the number of internships impact the job offers received*

This heatmap visualizes the relationship between the number of internships completed and **the** number of job offers received. It allows us to quickly identify patterns or associations, such as whether completing more internships tends to result in more job offers.

education\_data **%>%**  
 **count**(Internships\_Completed, Job\_Offers) **%>%**   
 **ggplot**(**aes**(x = **as.factor**(Internships\_Completed), y = **as.factor**(Job\_Offers), fill = n)) **+**  
 **geom\_tile**(color = "white") **+**   
 **geom\_text**(**aes**(label = n), color = "black") **+**   
 **scale\_fill\_gradient**(low = "lightblue", high = "darkblue") **+**  
 **labs**(  
 title = "Count of Internships vs. Job Offers",  
 x = "Internships Completed",  
 y = "Job Offers Received",  
 fill = "Count"  
 ) **+**  
 **theme\_minimal**() **+**  
 **theme**(plot.title = **element\_text**(hjust = 0.5))

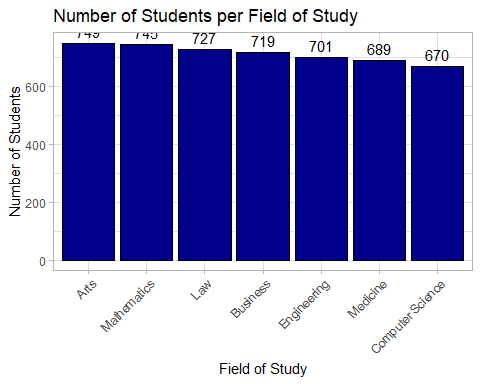


*3.2.4 Stratified Sampling*

To ensure representative insights across various academic disciplines, we applied stratified sampling based on the Field\_of\_Study variable. This method divides the student population into distinct strata by field of study, allowing us to sample proportionally from each group.

The code below calculates the number of students in each field of study and visualizes the distribution with a bar chart. This visualization helps us understand the relative sizes of each stratum and confirms that stratified sampling is appropriate given the variability across fields.

**library**(dplyr)  
**library**(ggplot2)  
  
field\_of\_study\_counts <- education\_data **%>%**  
 **group\_by**(Field\_of\_Study) **%>%**  
 **count**(name = "Number\_of\_Students") **%>%**  
 **arrange**(**desc**(Number\_of\_Students))  
  
  
education\_data**$**Field\_of\_Study <- **factor**(education\_data**$**Field\_of\_Study,  
 levels = field\_of\_study\_counts**$**Field\_of\_Study)  
  
**ggplot**(data = education\_data, **aes**(x = Field\_of\_Study)) **+**  
 **geom\_bar**(fill = "darkblue", color = "black") **+**  
 **labs**(  
 title = "Number of Students per Field of Study",  
 x = "Field of Study",  
 y = "Number of Students"  
 ) **+**  
 **theme\_light**() **+**   
 **theme**(axis.text.x = **element\_text**(angle = 45, hjust = 1)) **+**   
 **geom\_text**(stat = "count", **aes**(label = **after\_stat**(count)), vjust = **-**0.5)



**3.3 Multiple Linear Regression:**

Next, we built a multiple linear regression model to predict starting salary based on a variety of educational and personal background factors, including GPA, SAT score, internships, certifications, soft skills, networking, and university ranking.

As there are multiple assumptions that are to be cleared for the model, we can choose to switch to classification by changing the starting salary to a categorical variable through labeling.

Using the ordinary least-squares method to fit the model:

**colnames**(education\_data)

## [1] "Student\_ID" "Age"   
## [3] "Gender" "High\_School\_GPA"   
## [5] "SAT\_Score" "University\_GPA"   
## [7] "Field\_of\_Study" "Internships\_Completed"   
## [9] "Projects\_Completed" "Certifications"   
## [11] "Soft\_Skills\_Score" "Networking\_Score"   
## [13] "Job\_Offers" "Starting\_Salary"   
## [15] "Career\_Satisfaction" "Years\_to\_Promotion"   
## [17] "Current\_Job\_Level" "Work\_Life\_Balance"   
## [19] "Entrepreneurship" "University\_Ranking\_Category"

salary\_pred\_full\_model = **lm**(Starting\_Salary **~** Age**+**Gender**+**High\_School\_GPA**+**SAT\_Score**+**University\_GPA**+**Field\_of\_Study**+**Internships\_Completed**+**Projects\_Completed**+**Certifications**+**Soft\_Skills\_Score**+**Networking\_Score**+**University\_Ranking\_Category ,data = education\_data)  
  
**summary**(salary\_pred\_full\_model)

##   
## Call:  
## lm(formula = Starting\_Salary ~ Age + Gender + High\_School\_GPA +   
## SAT\_Score + University\_GPA + Field\_of\_Study + Internships\_Completed +   
## Projects\_Completed + Certifications + Soft\_Skills\_Score +   
## Networking\_Score + University\_Ranking\_Category, data = education\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -30198 -10577 -273 9890 50362   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 50492.0905 2813.3452 17.947  
## Age 60.1917 59.3476 1.014  
## GenderMale 61.6980 420.1834 0.147  
## GenderOther -687.8222 1094.9583 -0.628  
## High\_School\_GPA -296.1421 358.2188 -0.827  
## SAT\_Score 0.2139 1.0143 0.211  
## University\_GPA 47.3767 357.8969 0.132  
## Field\_of\_StudyMathematics -644.0131 752.8497 -0.855  
## Field\_of\_StudyLaw -1294.2769 757.9416 -1.708  
## Field\_of\_StudyBusiness -1161.8957 761.5724 -1.526  
## Field\_of\_StudyEngineering -973.1841 765.2175 -1.272  
## Field\_of\_StudyMedicine -1161.4509 768.6380 -1.511  
## Field\_of\_StudyComputer Science -678.0277 774.6629 -0.875  
## Internships\_Completed1 -19.9762 645.8117 -0.031  
## Internships\_Completed2 -183.4172 655.1768 -0.280  
## Internships\_Completed3 -164.9211 648.1960 -0.254  
## Internships\_Completed4 943.8292 656.9080 1.437  
## Projects\_Completed1 562.6541 938.4985 0.600  
## Projects\_Completed2 -200.2490 943.3592 -0.212  
## Projects\_Completed3 856.7340 919.4240 0.932  
## Projects\_Completed4 1263.0184 948.3250 1.332  
## Projects\_Completed5 401.4498 943.0525 0.426  
## Projects\_Completed6 1062.3811 930.0437 1.142  
## Projects\_Completed7 -141.6820 947.5466 -0.150  
## Projects\_Completed8 761.1029 938.4340 0.811  
## Projects\_Completed9 1170.5797 919.8914 1.273  
## Certifications1 -241.7171 715.9609 -0.338  
## Certifications2 -6.8432 719.5988 -0.010  
## Certifications3 -1022.2097 717.7293 -1.424  
## Certifications4 -382.0254 713.2115 -0.536  
## Certifications5 -832.3882 719.6549 -1.157  
## Soft\_Skills\_Score2 905.9852 943.5627 0.960  
## Soft\_Skills\_Score3 1584.9334 922.5605 1.718  
## Soft\_Skills\_Score4 1565.4451 944.7033 1.657  
## Soft\_Skills\_Score5 407.5454 926.8212 0.440  
## Soft\_Skills\_Score6 875.1810 926.1701 0.945  
## Soft\_Skills\_Score7 1066.0563 923.2973 1.155  
## Soft\_Skills\_Score8 -792.5577 933.7828 -0.849  
## Soft\_Skills\_Score9 983.5348 921.9703 1.067  
## Soft\_Skills\_Score10 1601.8937 935.5600 1.712  
## Networking\_Score2 -1922.7283 930.5329 -2.066  
## Networking\_Score3 -1722.1845 939.7775 -1.833  
## Networking\_Score4 -2100.0509 936.0527 -2.244  
## Networking\_Score5 -925.6461 925.3021 -1.000  
## Networking\_Score6 -911.9324 912.8331 -0.999  
## Networking\_Score7 -1039.2184 935.5103 -1.111  
## Networking\_Score8 -1234.4905 925.1500 -1.334  
## Networking\_Score9 -1294.3217 926.7356 -1.397  
## Networking\_Score10 -834.3801 934.8819 -0.892  
## University\_Ranking\_CategoryLow ranked -467.1148 589.8739 -0.792  
## University\_Ranking\_CategoryModerately Ranked -336.0557 583.4416 -0.576  
## University\_Ranking\_CategoryVery low ranked 1018.3750 580.9388 1.753  
## Pr(>|t|)   
## (Intercept) <2e-16 \*\*\*  
## Age 0.3105   
## GenderMale 0.8833   
## GenderOther 0.5299   
## High\_School\_GPA 0.4084   
## SAT\_Score 0.8329   
## University\_GPA 0.8947   
## Field\_of\_StudyMathematics 0.3924   
## Field\_of\_StudyLaw 0.0878 .   
## Field\_of\_StudyBusiness 0.1272   
## Field\_of\_StudyEngineering 0.2035   
## Field\_of\_StudyMedicine 0.1308   
## Field\_of\_StudyComputer Science 0.3815   
## Internships\_Completed1 0.9753   
## Internships\_Completed2 0.7795   
## Internships\_Completed3 0.7992   
## Internships\_Completed4 0.1508   
## Projects\_Completed1 0.5488   
## Projects\_Completed2 0.8319   
## Projects\_Completed3 0.3515   
## Projects\_Completed4 0.1830   
## Projects\_Completed5 0.6704   
## Projects\_Completed6 0.2534   
## Projects\_Completed7 0.8811   
## Projects\_Completed8 0.4174   
## Projects\_Completed9 0.2032   
## Certifications1 0.7357   
## Certifications2 0.9924   
## Certifications3 0.1544   
## Certifications4 0.5922   
## Certifications5 0.2475   
## Soft\_Skills\_Score2 0.3370   
## Soft\_Skills\_Score3 0.0859 .   
## Soft\_Skills\_Score4 0.0976 .   
## Soft\_Skills\_Score5 0.6602   
## Soft\_Skills\_Score6 0.3447   
## Soft\_Skills\_Score7 0.2483   
## Soft\_Skills\_Score8 0.3961   
## Soft\_Skills\_Score9 0.2861   
## Soft\_Skills\_Score10 0.0869 .   
## Networking\_Score2 0.0389 \*   
## Networking\_Score3 0.0669 .   
## Networking\_Score4 0.0249 \*   
## Networking\_Score5 0.3172   
## Networking\_Score6 0.3178   
## Networking\_Score7 0.2667   
## Networking\_Score8 0.1821   
## Networking\_Score9 0.1626   
## Networking\_Score10 0.3722   
## University\_Ranking\_CategoryLow ranked 0.4285   
## University\_Ranking\_CategoryModerately Ranked 0.5646   
## University\_Ranking\_CategoryVery low ranked 0.0797 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14500 on 4948 degrees of freedom  
## Multiple R-squared: 0.009599, Adjusted R-squared: -0.0006095   
## F-statistic: 0.9403 on 51 and 4948 DF, p-value: 0.5952

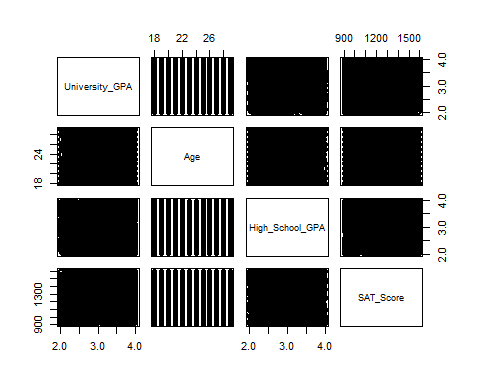
At the 0.05 significance level, the model revealed that most predictors were not statistically significant in explaining starting salary. Only a few networking score levels and some fields of study showed marginal influence. The model’s overall explanatory power was very low, indicating that these variables collectively explain very little of the variation in starting salaries.

Moreover, the negative adjusted R-squared value indicates that the model’s predictive performance is worse than simply using the mean starting salary as a prediction. This suggests that the model is poorly specified or missing key factors that better explain salary differences, highlighting the need to explore alternative modeling approaches or additional variables.

*3.3.1 Multicollinearity and Pairs Plot*

This lack of significance among predictors could be due to multicollinearity, where some predictors are highly correlated with each other, inflating standard errors and reducing statistical significance. To explore this, we can visually inspect the relationships between continuous variables using a pairs plot:

**pairs**(**~**University\_GPA **+**Age**+**High\_School\_GPA**+** SAT\_Score , data = education\_data)



Above, we see there is no pattern that appears between the predictors.

*3.3.2 VIF test for Multicollinearity*

Next, we can use the VIF test to confirm if there is multicollinearity among the predictors in the salary prediction model.

**library**(mctest)   
**imcdiag**(salary\_pred\_full\_model, method = "VIF")

##   
## Call:  
## imcdiag(mod = salary\_pred\_full\_model, method = "VIF")  
##   
##   
## VIF Multicollinearity Diagnostics  
##   
## VIF detection  
## Age 1.0106 0  
## GenderMale 1.0495 0  
## GenderOther 1.0529 0  
## High\_School\_GPA 1.0112 0  
## SAT\_Score 1.0103 0  
## University\_GPA 1.0107 0  
## Field\_of\_StudyMathematics 1.7092 0  
## Field\_of\_StudyLaw 1.6977 0  
## Field\_of\_StudyBusiness 1.6984 0  
## Field\_of\_StudyEngineering 1.6788 0  
## Field\_of\_StudyMedicine 1.6694 0  
## Field\_of\_StudyComputer Science 1.6562 0  
## Internships\_Completed1 1.6318 0  
## Internships\_Completed2 1.6051 0  
## Internships\_Completed3 1.6238 0  
## Internships\_Completed4 1.5972 0  
## Projects\_Completed1 1.8752 0  
## Projects\_Completed2 1.8607 0  
## Projects\_Completed3 1.9431 0  
## Projects\_Completed4 1.8458 0  
## Projects\_Completed5 1.8663 0  
## Projects\_Completed6 1.9170 0  
## Projects\_Completed7 1.8601 0  
## Projects\_Completed8 1.8817 0  
## Projects\_Completed9 1.9451 0  
## Certifications1 1.7089 0  
## Certifications2 1.6935 0  
## Certifications3 1.7076 0  
## Certifications4 1.7134 0  
## Certifications5 1.7036 0  
## Soft\_Skills\_Score2 1.8205 0  
## Soft\_Skills\_Score3 1.8959 0  
## Soft\_Skills\_Score4 1.8214 0  
## Soft\_Skills\_Score5 1.8778 0  
## Soft\_Skills\_Score6 1.8784 0  
## Soft\_Skills\_Score7 1.8861 0  
## Soft\_Skills\_Score8 1.8498 0  
## Soft\_Skills\_Score9 1.8903 0  
## Soft\_Skills\_Score10 1.8435 0  
## Networking\_Score2 1.8501 0  
## Networking\_Score3 1.8229 0  
## Networking\_Score4 1.8320 0  
## Networking\_Score5 1.8684 0  
## Networking\_Score6 1.9216 0  
## Networking\_Score7 1.8399 0  
## Networking\_Score8 1.8743 0  
## Networking\_Score9 1.8612 0  
## Networking\_Score10 1.8408 0  
## University\_Ranking\_CategoryLow ranked 1.5223 0  
## University\_Ranking\_CategoryModerately Ranked 1.5325 0  
## University\_Ranking\_CategoryVery low ranked 1.5358 0  
##   
## NOTE: VIF Method Failed to detect multicollinearity  
##   
##   
## 0 --> COLLINEARITY is not detected by the test  
##   
## ===================================

All VIF values were below the threshold of 5, and no multicollinearity was detected. This indicates that the predictors are not highly correlated with each other, so multicollinearity is unlikely to be affecting the reliability of the coefficient estimates in the model.

**3.4 Stepwise Model Selection:**

Since some predictors in the full model were not significant, we used the ols\_step\_best\_subset() function from the olsrr package to identify a simpler model that balances fit and complexity. This method evaluates different subsets of predictors and helps select the best model based on several criteria.

**library**(olsrr)

## Warning: package 'olsrr' was built under R version 4.4.3

##   
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':  
##   
## rivers

salary\_pred\_Subsets = **ols\_step\_best\_subset**(salary\_pred\_full\_model, details=FALSE)

We focused on three key metrics to choose the best model:

* Adjusted R²: Adjusts R² to penalize for adding unnecessary predictors, helping to avoid overfitting.
* Mallow’s Cp: Measures model bias and variance; values close to the number of predictors indicate a good fit.
* Akaike Information Criterion (AIC): Balances model fit and complexity; lower AIC values indicate better models.

However, we do not choose to use R-squares as it does not punish the model for adding more predictors or overfitting.

AdjustedR2=**c**((salary\_pred\_Subsets**$**metrics)**$**adjr)  
cp=**c**((salary\_pred\_Subsets**$**metrics)**$**cp)  
AIC=**c**((salary\_pred\_Subsets**$**metrics)**$**aic)  
**cbind**(AdjustedR2,cp,AIC)

## AdjustedR2 cp AIC  
## [1,] 5.977466e-04 3.979312 110013.9  
## [2,] 4.196567e-04 13.876697 110023.8  
## [3,] 1.414276e-03 11.931582 110021.8  
## [4,] 8.175907e-04 23.912910 110033.8  
## [5,] 4.712920e-04 31.639101 110041.5  
## [6,] 4.818598e-04 35.591048 110045.4  
## [7,] 1.574866e-04 42.202467 110051.9  
## [8,] 1.574997e-04 43.203169 110052.9  
## [9,] 9.093337e-05 44.533371 110054.3  
## [10,] -2.180239e-04 48.063178 110057.8  
## [11,] -4.108982e-04 50.017523 110059.7  
## [12,] -6.095395e-04 52.000000 110061.7

The results showed that the model with the highest adjusted R² includes only 3 predictors. This model also has a relatively low Mallow’s Cp and the lowest AIC compared to other models, suggesting it provides the best trade-off between complexity and explanatory power.

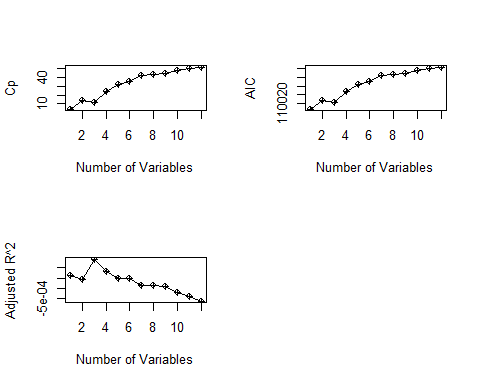
salary\_pred\_Subsets**$**metrics**$**predictors[3]

## [1] "Soft\_Skills\_Score Networking\_Score University\_Ranking\_Category"

Therefore, the best model includes the following three predictors: **Soft\_Skills\_Score**, **Networking\_Score**, and **University\_Ranking\_Category**. These variables were identified as the most important in explaining salary, based on their contribution to improving model fit while keeping complexity low.

Next, we generated three plots to compare model performance across different numbers of predictors. It visualizes Mallow’s Cp, AIC, and Adjusted R² to help identify the model that offers the best balance between fit and simplicity. The plots suggest the optimal number of predictors based on where Cp is low, AIC is minimized, and Adjusted R² is maximized.

**par**(mfrow=**c**(2,2)) *# split the plotting panel into a 2 x 2 grid*  
**plot**(cp,type = "o",pch=10, xlab="Number of Variables",ylab= "Cp")  
**plot**(AIC,type = "o",pch=10, xlab="Number of Variables",ylab= "AIC")  
**plot**(AdjustedR2,type = "o",pch=10, xlab="Number of Variables",ylab= "Adjusted R^2")



According to Mallow’s Cp criterion, an ideal model has a Cp value approximately equal to *k + 2*. In our best model, which includes 3 predictors, the ideal Cp would be around 5, but the observed Cp is 11.93. This suggests that the model may still have some bias—it is not capturing all the important variability in the data. While it strikes a good balance in terms of adjusted R² and AIC, the elevated Cp indicates that the model might be slightly underfitting, possibly missing other important predictors.

*3.4. Reduced Model*

Based on the above evaluation, we fit a new linear regression model using only the three predictors identified in the best subset selection: Soft\_Skills\_Score, Networking\_Score, and University\_Ranking\_Category.

salary\_pred\_best\_model = **lm**(Starting\_Salary **~** Soft\_Skills\_Score**+**Networking\_Score**+**University\_Ranking\_Category ,data = education\_data)  
  
**summary**(salary\_pred\_best\_model)

##   
## Call:  
## lm(formula = Starting\_Salary ~ Soft\_Skills\_Score + Networking\_Score +   
## University\_Ranking\_Category, data = education\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -27975 -10488 -319 9831 49795   
##   
## Coefficients:  
## Estimate Std. Error t value  
## (Intercept) 50905.7 993.8 51.221  
## Soft\_Skills\_Score2 804.9 940.6 0.856  
## Soft\_Skills\_Score3 1517.5 918.3 1.652  
## Soft\_Skills\_Score4 1466.6 940.5 1.559  
## Soft\_Skills\_Score5 387.4 923.3 0.420  
## Soft\_Skills\_Score6 848.5 922.8 0.920  
## Soft\_Skills\_Score7 1051.9 920.7 1.142  
## Soft\_Skills\_Score8 -846.2 930.8 -0.909  
## Soft\_Skills\_Score9 995.7 918.6 1.084  
## Soft\_Skills\_Score10 1688.1 932.5 1.810  
## Networking\_Score2 -1941.4 928.4 -2.091  
## Networking\_Score3 -1719.7 936.8 -1.836  
## Networking\_Score4 -2172.9 933.2 -2.328  
## Networking\_Score5 -1014.2 922.0 -1.100  
## Networking\_Score6 -866.7 909.7 -0.953  
## Networking\_Score7 -1043.0 932.1 -1.119  
## Networking\_Score8 -1218.7 922.1 -1.322  
## Networking\_Score9 -1296.0 924.5 -1.402  
## Networking\_Score10 -860.1 932.0 -0.923  
## University\_Ranking\_CategoryLow ranked -438.5 587.1 -0.747  
## University\_Ranking\_CategoryModerately Ranked -324.5 580.9 -0.559  
## University\_Ranking\_CategoryVery low ranked 1017.0 578.6 1.758  
## Pr(>|t|)   
## (Intercept) <2e-16 \*\*\*  
## Soft\_Skills\_Score2 0.3922   
## Soft\_Skills\_Score3 0.0985 .   
## Soft\_Skills\_Score4 0.1190   
## Soft\_Skills\_Score5 0.6748   
## Soft\_Skills\_Score6 0.3579   
## Soft\_Skills\_Score7 0.2533   
## Soft\_Skills\_Score8 0.3633   
## Soft\_Skills\_Score9 0.2785   
## Soft\_Skills\_Score10 0.0703 .   
## Networking\_Score2 0.0366 \*   
## Networking\_Score3 0.0665 .   
## Networking\_Score4 0.0199 \*   
## Networking\_Score5 0.2714   
## Networking\_Score6 0.3408   
## Networking\_Score7 0.2632   
## Networking\_Score8 0.1863   
## Networking\_Score9 0.1610   
## Networking\_Score10 0.3561   
## University\_Ranking\_CategoryLow ranked 0.4552   
## University\_Ranking\_CategoryModerately Ranked 0.5764   
## University\_Ranking\_CategoryVery low ranked 0.0789 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14480 on 4978 degrees of freedom  
## Multiple R-squared: 0.005609, Adjusted R-squared: 0.001414   
## F-statistic: 1.337 on 21 and 4978 DF, p-value: 0.1386

Although the reduced model is simpler and includes only three predictors, it performs poorly overall. The adjusted R-squared value is very low at 0.14%, and the model is not statistically significant. Most individual predictors also lack significance, indicating that this model does not effectively explain variations in starting salary. Therefore, the rest of the assumptions required for multiple linear regression models will not be performed.

**3.5 Regression Tree Model:**

After exploring linear models, we now turn to regression trees as a more flexible, non-parametric method. Regression trees require less assumptions and can be pruned to for better interpretation.

We randomly split the dataset into a 75% training set and 25% testing set, then fit a regression tree model to predict Starting Salary.

**library**(tree)  
idx=**sample**(1**:nrow**(education\_data),0.75**\*nrow**(education\_data))  
train=education\_data[idx,]  
test=education\_data[**-**idx,]  
reg.tree.salary<-**tree**(Starting\_Salary **~** Age**+**Gender**+**High\_School\_GPA**+**SAT\_Score**+**University\_GPA**+**Field\_of\_Study**+**Internships\_Completed**+**Projects\_Completed**+**Certifications**+**Soft\_Skills\_Score**+**Networking\_Score**+**University\_Ranking\_Category, train)

## Warning in tree(Starting\_Salary ~ Age + Gender + High\_School\_GPA + SAT\_Score +  
## : NAs introduced by coercion

**summary**(reg.tree.salary)

##   
## Regression tree:  
## tree(formula = Starting\_Salary ~ Age + Gender + High\_School\_GPA +   
## SAT\_Score + University\_GPA + Field\_of\_Study + Internships\_Completed +   
## Projects\_Completed + Certifications + Soft\_Skills\_Score +   
## Networking\_Score + University\_Ranking\_Category, data = train)  
## Variables actually used in tree construction:  
## character(0)  
## Number of terminal nodes: 1   
## Residual mean deviance: 210600000 = 7.895e+11 / 3749   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -25400.0 -10500.0 -151.4 0.0 9999.0 50200.0

The regression tree model did not split the data at all, resulting in a tree with only one terminal node. This means the model predicted the same salary for all observations, effectively just using the mean starting salary as the prediction. It is not possible to construct a regression tree with a single node. Further, the residual deviance is extremely high, this also indicates that regression cannot be used to predict the starting salary of students.

**3.6 Classification:**

Since the linear regression model explained very little variance in salary (low R², insignificant predictors, and the regression tree failed to split, indicating no strong continuous relationships we reframe the problem as a classification task. We now try to predict the salary category, which is often easier, more interpretable, and more robust to noise in real-world data.

We can choose to perform classification by converting the salary data into a categorical variable called Starting\_Salary\_Category, based on the original Starting\_Salary.

education\_data <- education\_data **%>%**  
 **mutate**(  
 Starting\_Salary\_Category = **case\_when**(  
 Starting\_Salary **>=** 25000 **&** Starting\_Salary **<=** 40000 **~** "Low.Salary",  
 Starting\_Salary **>** 40000 **&** Starting\_Salary **<=** 75000 **~** "Median.Salary",  
 Starting\_Salary **>** 75000 **&** Starting\_Salary **<=** 120000 **~** "High.Salary"  
 )  
 )

After conversion we remove the continuous variable since we are no longer modeling Starting\_Salary directly, and instead focusing only on the new categorical version.

As the dependent variable consists of more than 2 classes it is better to use Linear Discriminant Analysis or Quadratic Discriminant Analysis to predict the outcome.

**3.7 Linear Discriminant Analysis (LDA):**

*3.7.1 Shapiro-Wilk Normality Test*

This test is performed to evaluate whether the normality assumption required for Linear and/or Quadratic Discriminant Analysis are met. This is essential before applying Linear and/or Quadratic Discriminant Analysis to ensure the validity and reliability of the classification results.

numerical\_cols <- **c**(  
 "Age", "High\_School\_GPA", "SAT\_Score", "University\_GPA")  
  
shapiro\_results <- **list**()  
  
**cat**("--- Shapiro-Wilk Normality Test Results ---**\n**")

## --- Shapiro-Wilk Normality Test Results ---

**for** (col\_name **in** numerical\_cols) {  
   
 data\_vector <- education\_data[[col\_name]]  
  
 data\_vector <- **na.omit**(data\_vector)  
  
 test\_result <- **shapiro.test**(data\_vector)  
  
 shapiro\_results[[col\_name]] <- test\_result  
  
 **cat**(**sprintf**("**\n**Column: '%s'**\n**", col\_name))  
 **cat**(**sprintf**(" Shapiro-Wilk W statistic: %.4f**\n**", test\_result**$**statistic))  
 **cat**(**sprintf**(" p-value: %.4f**\n**", test\_result**$**p.value))  
   
 *# Interpret the p-value*  
 **if** (test\_result**$**p.value **<** 0.05) {  
 **cat**(" Conclusion: The data in this column is likely NOT normally distributed (p < 0.05).**\n**")  
 } **else** {  
 **cat**(" Conclusion: The data in this column appears to be normally distributed (p >= 0.05).**\n**")  
 }  
}

##   
## Column: 'Age'  
## Shapiro-Wilk W statistic: 0.9401  
## p-value: 0.0000  
## Conclusion: The data in this column is likely NOT normally distributed (p < 0.05).  
##   
## Column: 'High\_School\_GPA'  
## Shapiro-Wilk W statistic: 0.9558  
## p-value: 0.0000  
## Conclusion: The data in this column is likely NOT normally distributed (p < 0.05).  
##   
## Column: 'SAT\_Score'  
## Shapiro-Wilk W statistic: 0.9531  
## p-value: 0.0000  
## Conclusion: The data in this column is likely NOT normally distributed (p < 0.05).  
##   
## Column: 'University\_GPA'  
## Shapiro-Wilk W statistic: 0.9559  
## p-value: 0.0000  
## Conclusion: The data in this column is likely NOT normally distributed (p < 0.05).

The W statistics which is close to 1 indicates that the variable follows a normal distribution but the p-value states to reject null-hypothesis, which is the data is normally distributed.

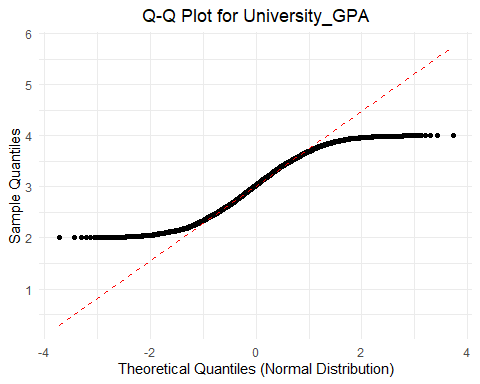
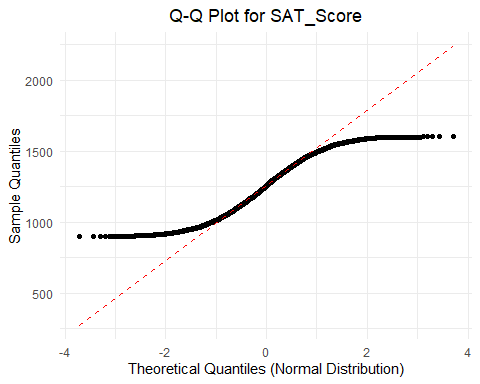
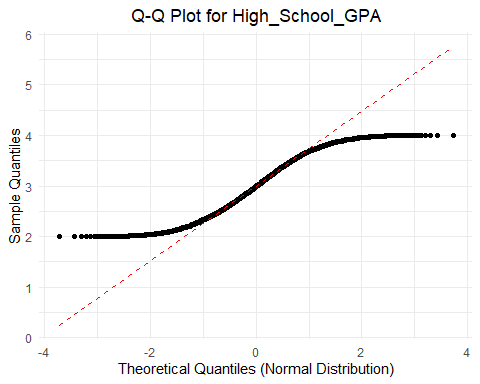
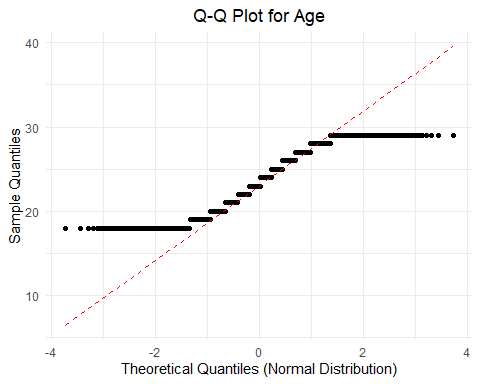
*3.7.2 Quantile-Quantile plots*

The shapiro test is very sensitive and can conclude data is not normal for small deviances. However, we can use Quantile-Quantile plots to test the results of the test:

qq\_plots\_list <- **list**()  
  
**for** (col\_name **in** numerical\_cols ) {  
 data\_vector <- education\_data[[col\_name]]  
  
  
 p <- **ggplot**(**data.frame**(x = data\_vector), **aes**(sample = x)) **+**  
 **stat\_qq**() **+** *# Adds the QQ-plot points*  
 **stat\_qq\_line**(color = "red", linetype = "dashed") **+**   
 **labs**(  
 title = **paste0**("Q-Q Plot for ", col\_name),  
 x = "Theoretical Quantiles (Normal Distribution)",  
 y = "Sample Quantiles"  
 ) **+**  
 **theme\_minimal**() **+**  
 **theme**(plot.title = **element\_text**(hjust = 0.5))  
  
 *# Store the plot in the list*  
 qq\_plots\_list[[col\_name]] <- p  
}  
  
**cat**("**\n**--- Displaying Q-Q Plots ---**\n**")

##   
## --- Displaying Q-Q Plots ---

**for** (plot\_name **in** **names**(qq\_plots\_list)) {  
 **print**(qq\_plots\_list[[plot\_name]])  
}



The QQ plots indicate that there are heavier tails and the data is not normally distributed. This could be due to outliers, we can check for outliers and remove them using IQR.

*3.7.3 Interquartile Range (IQR)*

The IQR rule detects outliers as values that fall below Q1 − 1.5 × IQR or above Q3 + 1.5 × IQR, where Q1 and Q3 are the 25th and 75th percentiles, respectively. It is a non-parametric method that does not assume normality, making it a reliable and commonly used approach.

education\_data\_cleaned\_iqr <- education\_data  
  
**cat**("--- Outlier Detection and Removal using IQR Rule ---**\n**")

## --- Outlier Detection and Removal using IQR Rule ---

**for** (col\_name **in** numerical\_cols) {  
 **cat**(**sprintf**("**\n**Processing column: '%s'**\n**", col\_name))  
  
 data\_vector <- education\_data[[col\_name]]  
 data\_vector\_no\_na <- **na.omit**(data\_vector)  
  
   
 Q1 <- **quantile**(data\_vector\_no\_na, 0.25)  
 Q3 <- **quantile**(data\_vector\_no\_na, 0.75)  
 IQR\_val <- Q3 **-** Q1  
  
   
 lower\_bound <- Q1 **-** 1.5 **\*** IQR\_val  
 upper\_bound <- Q3 **+** 1.5 **\*** IQR\_val  
  
 outlier\_indices <- **which**(data\_vector **<** lower\_bound **|** data\_vector **>** upper\_bound)  
  
 *# Report detected outliers*  
 **if** (**length**(outlier\_indices) **>** 0) {  
 **cat**(**sprintf**(" Detected %d outliers in '%s'.**\n**", **length**(outlier\_indices), col\_name))  
 **cat**(**sprintf**(" Outlier values: %s**\n**", **paste**(data\_vector[outlier\_indices], collapse = ", ")))  
 **cat**(**sprintf**(" Lower bound: %.2f, Upper bound: %.2f**\n**", lower\_bound, upper\_bound))  
  
 } **else** {  
 **cat**(**sprintf**(" No outliers detected in '%s' using IQR rule.**\n**", col\_name))  
 }  
}

##   
## Processing column: 'Age'  
## No outliers detected in 'Age' using IQR rule.  
##   
## Processing column: 'High\_School\_GPA'  
## No outliers detected in 'High\_School\_GPA' using IQR rule.  
##   
## Processing column: 'SAT\_Score'  
## No outliers detected in 'SAT\_Score' using IQR rule.  
##   
## Processing column: 'University\_GPA'  
## No outliers detected in 'University\_GPA' using IQR rule.

As found above, no outliers are detected.

*3.7.4 Box-cox Transformation*

As there are no outliers detected, we can use the box-cox transformation on the continuous variables to obtain normality.

**library**(caret)

## Warning: package 'caret' was built under R version 4.4.3

## Loading required package: lattice

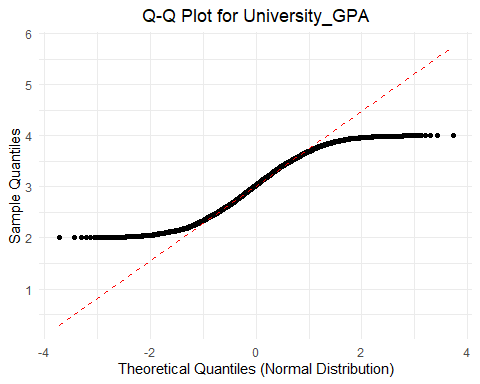
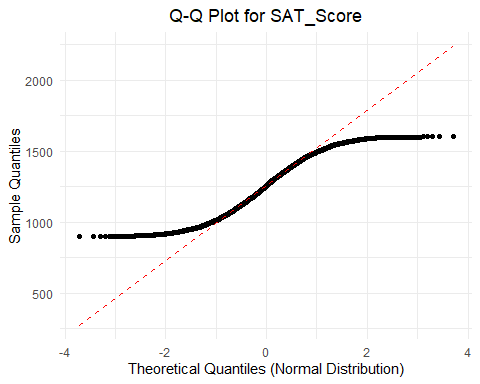
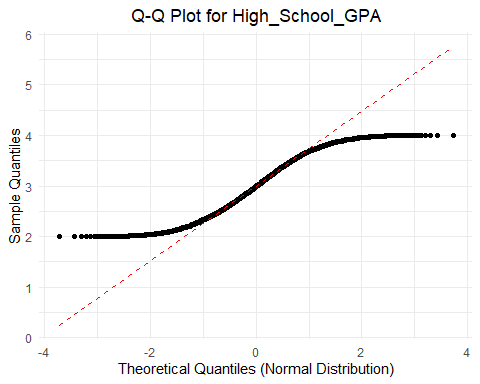
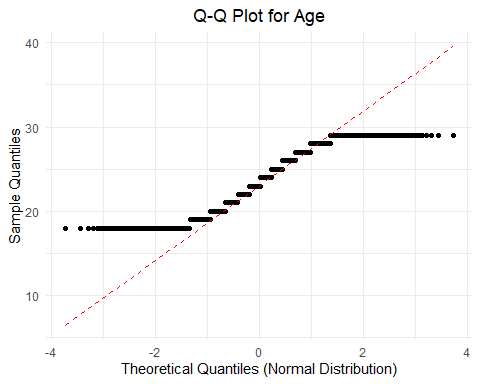
education\_data\_transformed <- education\_data  
**for** (col\_name **in** numerical\_cols) {  
 data\_vector <- education\_data\_transformed[[col\_name]]  
 bc\_object <- **BoxCoxTrans**(data\_vector)  
 lambda\_val <- bc\_object**$**lambda  
 **cat**(**sprintf**(" Optimal lambda for '%s': %.4f**\n**", col\_name, lambda\_val))  
   
 transformed\_data <- **predict**(bc\_object, data\_vector)  
 education\_data\_transformed[[**paste0**(col\_name, "\_BC")]] <- transformed\_data  
}

## Optimal lambda for 'Age': 0.6000  
## Optimal lambda for 'High\_School\_GPA': 0.7000  
## Optimal lambda for 'SAT\_Score': 0.8000  
## Optimal lambda for 'University\_GPA': 0.9000

qq\_plots\_list <- **list**()  
  
**for** (col\_name **in** numerical\_cols ) {  
 data\_vector <- education\_data\_transformed[[col\_name]]  
  
  
 p <- **ggplot**(**data.frame**(x = data\_vector), **aes**(sample = x)) **+**  
 **stat\_qq**() **+** *# Adds the QQ-plot points*  
 **stat\_qq\_line**(color = "red", linetype = "dashed") **+**   
 **labs**(  
 title = **paste0**("Q-Q Plot for ", col\_name),  
 x = "Theoretical Quantiles (Normal Distribution)",  
 y = "Sample Quantiles"  
 ) **+**  
 **theme\_minimal**() **+**  
 **theme**(plot.title = **element\_text**(hjust = 0.5))  
  
 *# Store the plot in the list*  
 qq\_plots\_list[[col\_name]] <- p  
}  
  
**cat**("**\n**--- Displaying Q-Q Plots for transformed data---**\n**")

##   
## --- Displaying Q-Q Plots for transformed data---

**for** (plot\_name **in** **names**(qq\_plots\_list)) {  
 **print**(qq\_plots\_list[[plot\_name]])  
}



Based on the plots it appears the tails are still heavy on the end, indicating kurtosis. However, if we consider the test statistic from the Shapiro-Wilk tests, they were closer to 1 indicating the data is approximately normally distributed. While LDA assumes multivariate normality within each class, in practice, it’s often reasonably robust to mild deviations, especially with large sample sizes.

We can consider the data to be approximately normally distributed for this case.

*3.7.5 Levene’s test for Equal Variance*

Another assumption that needs to be satisfied for Linear Discriminant Analysis (LDA) is equal variance between the classes/predictors. We can use Levene’s test for equal variance across the predictors.

Here, the null hypothesis indicates the variances are equal across all groups The alternate hypothesis indicates at least one group variance is different from the others.

**library**(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

levene\_test\_age <- **leveneTest**(Age **~** **factor**(Starting\_Salary\_Category), data = education\_data)  
levene\_test\_age

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 2 1.4406 0.2369  
## 4997

levene\_test\_SchoolGPA <- **leveneTest**(High\_School\_GPA **~** **factor**(Starting\_Salary\_Category), data = education\_data)  
levene\_test\_SchoolGPA

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 2 1.7073 0.1815  
## 4997

levene\_test\_SAT <- **leveneTest**(SAT\_Score **~** **factor**(Starting\_Salary\_Category), data = education\_data)  
levene\_test\_SAT

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 2 0.3155 0.7295  
## 4997

levene\_test\_UniGPA <- **leveneTest**(University\_GPA **~** **factor**(Starting\_Salary\_Category), data = education\_data)  
levene\_test\_UniGPA

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 2 2.2858 0.1018  
## 4997

All predictors pass the test of equal variances. The test statistics indicate small variations in the test between predictors, we can choose to proceed with Quadratic Discriminant Analysis (QDA) which does not require the assumption of equal variance.

**3.8 Quadratic Discriminant Analysis (QDA)**

We can proceed with QDA as the high school GPA can be a significant predictor which would get excluded while performing LDA.

*3.8.1 Cross Validation*

QDA can be performed using a 10 fold cross validation with the help of the caret library. We do not need to use the transformed data as it was unable to transform the data, using this would simply add to complexity in interpreting the model.

**library**(caret)  
**set.seed**(42)  
indexs = **sample**(1**:nrow**(education\_data),0.75**\*nrow**(education\_data))  
train\_data = education\_data[indexs,]  
test\_data = education\_data[**-**indexs,]  
  
qda\_model\_caret <- **train**(  
 **factor**(Starting\_Salary\_Category) **~** Age **+** High\_School\_GPA **+** SAT\_Score **+** University\_GPA ,   
 data = train\_data,  
 method = "qda",   
 trControl = **trainControl**(method = 'cv', number = 10, verboseIter = FALSE, classProbs = TRUE, summaryFunction = defaultSummary),  
 metric = "Accuracy"  
)  
qda\_model\_caret

## Quadratic Discriminant Analysis   
##   
## 3750 samples  
## 4 predictor  
## 3 classes: 'High.Salary', 'Low.Salary', 'Median.Salary'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 3376, 3374, 3374, 3376, 3375, 3376, ...   
## Resampling results:  
##   
## Accuracy Kappa  
## 0.7005342 0

The above output indicates that groups are too small for the folds. We can use fewer folds with the same training split data created before.

qda\_model\_caret\_5fold <- **train**(  
 **factor**(Starting\_Salary\_Category) **~** Age **+** High\_School\_GPA **+** SAT\_Score **+** University\_GPA ,   
 data = train\_data,  
 method = "qda",   
 trControl = **trainControl**(method = 'cv', number = 5, verboseIter = FALSE, classProbs = TRUE, summaryFunction = defaultSummary),  
 metric = "Accuracy"  
)  
qda\_model\_caret\_5fold

## Quadratic Discriminant Analysis   
##   
## 3750 samples  
## 4 predictor  
## 3 classes: 'High.Salary', 'Low.Salary', 'Median.Salary'   
##   
## No pre-processing  
## Resampling: Cross-Validated (5 fold)   
## Summary of sample sizes: 2999, 3000, 3001, 3000, 3000   
## Resampling results:  
##   
## Accuracy Kappa  
## 0.7005338 0

The accuracy is similar for the 10-fold and 5-fold cross validation. We can use the 10-fold cross- validated model.

qda.class<-**predict**(qda\_model\_caret, test\_data)  
**table**(qda.class, test\_data**$**Starting\_Salary\_Category)

##   
## qda.class High.Salary Low.Salary Median.Salary  
## High.Salary 0 0 0  
## Low.Salary 0 0 0  
## Median.Salary 57 311 882

Although the accuracy is 70%, the QDA predicted all the salaries to be median salary. Therefore we opt to use a different approach, which uses a better visualization like classification trees.

**3.9 Classification Trees**

train\_data**$**Starting\_Salary\_Category = **factor**(train\_data**$**Starting\_Salary\_Category)

**library**(klaR)

## Warning: package 'klaR' was built under R version 4.4.3

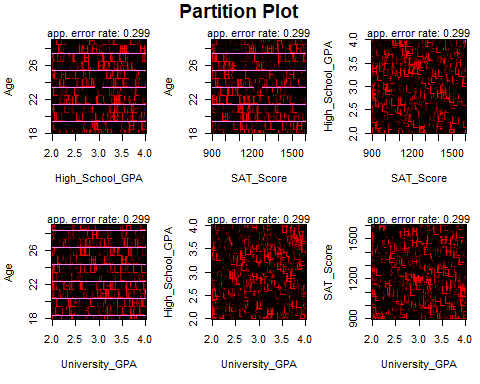
## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:olsrr':  
##   
## cement

## The following object is masked from 'package:dplyr':  
##   
## select

**partimat**(Starting\_Salary\_Category **~** Age **+** High\_School\_GPA **+** SAT\_Score **+** University\_GPA,  
 data = train\_data,  
 method = "qda")



The above plots lack clarity due to the large amounts of data. Therefore, we rely on the accuracy score to determine how to classify the outcome variable and move on to decision trees.

*3.9.1 Decision Trees*

Although decision trees are less accurate, they will better visualize outcomes and allow for stronger interpretation of data.

classification.salary.tree = **tree**(Starting\_Salary\_Category **~**Age **+** Gender **+** High\_School\_GPA **+** SAT\_Score **+** University\_GPA **+** Field\_of\_Study **+** Internships\_Completed **+** Projects\_Completed **+** Certifications **+** Soft\_Skills\_Score **+** Networking\_Score, data=train\_data)  
**summary**(classification.salary.tree)

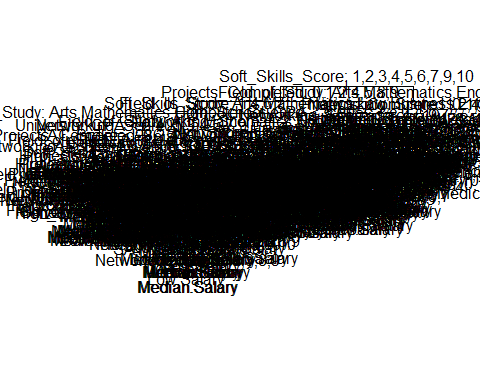
##   
## Classification tree:  
## tree(formula = Starting\_Salary\_Category ~ Age + Gender + High\_School\_GPA +   
## SAT\_Score + University\_GPA + Field\_of\_Study + Internships\_Completed +   
## Projects\_Completed + Certifications + Soft\_Skills\_Score +   
## Networking\_Score, data = train\_data)  
## Variables actually used in tree construction:  
## character(0)  
## Number of terminal nodes: 1   
## Residual mean deviance: 1.495 = 5605 / 3749   
## Misclassification error rate: 0.2995 = 1123 / 3750

The above tree only grew a single node which is not helpful for predictions. We can change few parameters to encourage the tree to grow more nodes:

classification.salary.tree\_adjusted <- **tree**(  
 Starting\_Salary\_Category **~** Age **+** Gender **+** High\_School\_GPA **+** SAT\_Score **+** University\_GPA **+**  
 Field\_of\_Study **+** Internships\_Completed **+** Projects\_Completed **+** Certifications **+**  
 Soft\_Skills\_Score **+** Networking\_Score,  
 data = train\_data,  
 control = **tree.control**(  
 nobs = **nrow**(train\_data),  
 mindev = 0.001,  
 mincut = 2   
 )  
)  
**summary**(classification.salary.tree\_adjusted)

##   
## Classification tree:  
## tree(formula = Starting\_Salary\_Category ~ Age + Gender + High\_School\_GPA +   
## SAT\_Score + University\_GPA + Field\_of\_Study + Internships\_Completed +   
## Projects\_Completed + Certifications + Soft\_Skills\_Score +   
## Networking\_Score, data = train\_data, control = tree.control(nobs = nrow(train\_data),   
## mindev = 0.001, mincut = 2))  
## Number of terminal nodes: 312   
## Residual mean deviance: 0.8246 = 2835 / 3438   
## Misclassification error rate: 0.1907 = 715 / 3750

**plot**(classification.salary.tree\_adjusted)  
**text**(classification.salary.tree\_adjusted, pretty = 0)



*3.9.2 Tree Pruning*

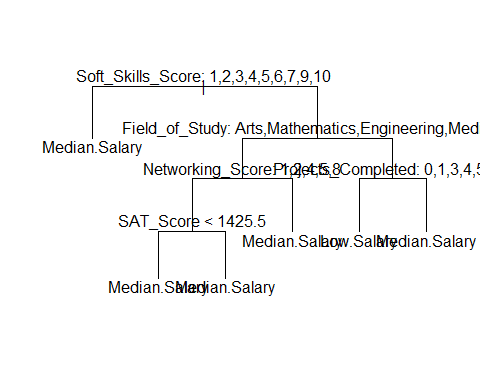
Now we can prune the tree to obtain a smaller tree which can be interpreted visually:

*#cv.salary <- cv.tree(classification.salary.tree\_adjusted, FUN = prune.misclass, K = 10,*  
*# control = tree.control(*  
*# nobs = nrow(train\_data),*  
*# mindev = 0.001,*  
*# mincut = 2*   
*#plot(cv.salary$size,cv.salary$dev,type='b')*

It is not possible to prune the tree using cv.tree function as it internally builds a tree with a single node. Using the same parameters explicitly results in errors indicating that there are internal issues on how cv.tree constructs a tree with parameters.

Therefore, we simply choose the number of residual nodes based on best visualization:

prune.class=**prune.tree**(classification.salary.tree\_adjusted,best=6)  
**plot**(prune.class)  
**text**(prune.class,pretty=0)



**summary**(prune.class)

##   
## Classification tree:  
## snip.tree(tree = classification.salary.tree\_adjusted, nodes = c(24L,   
## 15L, 25L, 13L, 14L, 2L))  
## Variables actually used in tree construction:  
## [1] "Soft\_Skills\_Score" "Field\_of\_Study" "Networking\_Score"   
## [4] "SAT\_Score" "Projects\_Completed"  
## Number of terminal nodes: 6   
## Residual mean deviance: 1.478 = 5532 / 3744   
## Misclassification error rate: 0.2971 = 1114 / 3750

Now, we observe that this tree is easier to visualize and does not overfit the test dataset, while also maintaining the same accuracy as the QDA model.

prune.pred=**predict**(prune.class,test\_data,type="class")  
**table**(prune.pred,test\_data**$**Starting\_Salary\_Category)

##   
## prune.pred High.Salary Low.Salary Median.Salary  
## High.Salary 0 0 0  
## Low.Salary 2 10 25  
## Median.Salary 55 301 857

This indicates that the model did not predict high salary successfully. As seen above, there were 2 instances where results should have been classified as high salary. We see this at: lower salary: 2 and Median salary: 55.

Further, there were 301 Low salary counts which were predicted as median salary and 25 median salary counts which were predicted as low salary.

**3.10 Sampling:**

Lastly, we determine the best sampling methodology to obtain samples from the dataset. A Simple random sample, stratified sample and cluster sample can be taken, then the population metrics can be compared to see which method provides greater accuracy and lower standard deviation.

*3.10.1 Simple random sample without replacement*

**set.seed**(2024)  
N = **dim**(education\_data)[1]  
n = 300  
idx=**sample**(1**:**N,size = n, replace = FALSE)  
population\_mean = **mean**(education\_data**$**Starting\_Salary)  
population\_SD = **sd**(education\_data**$**Starting\_Salary)  
**cat**("The population mean is: ", population\_mean, "and the standard deviation is: ", population\_SD)

## The population mean is: 50563.54 and the standard deviation is: 14494.96

Next, the Standard error can be calculated as :

Standard\_error\_est = population\_SD**/sqrt**(n) **\*** **sqrt**((N**-**n)**/**(N-1))  
Standard\_error\_est

## [1] 811.4536

Therefore, the best sampling method would provide a value close to a mean of 50563.54 and SE of 811.4536

**library**(survey)

## Warning: package 'survey' was built under R version 4.4.3

## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

##   
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':  
##   
## dotchart

new\_data <- **data.frame**(education\_data[idx,],pw=**rep**(N**/**n,n),fpc=**rep**(N,n))  
SRS\_svy <- **svydesign**(id=**~**0, strata = NULL, weights=**~**pw, data = new\_data, fpc=**~**fpc)  
mean\_salary <- **svymean**(**~**Starting\_Salary, SRS\_svy)  
mean\_salary

## mean SE  
## Starting\_Salary 52096 850.68

The mean and SE are indeed close to the population statistics, we can check other sampling methods are see if they have the same values.

*3.10.2 Stratified sampling*

**library**(sampling)

## Warning: package 'sampling' was built under R version 4.4.3

##   
## Attaching package: 'sampling'

## The following objects are masked from 'package:survival':  
##   
## cluster, strata

## The following object is masked from 'package:caret':  
##   
## cluster

desired\_sizes\_vector <- **c**("Arts" = 70,"Mathematics" = 70, "Law" = 70, "Business" = 70,"Engineering" = 70, "Medicine" = 70,  
 "Computer Science" = 70)  
Strata\_idx = sampling**::strata**(education\_data, stratanames=**c**("Field\_of\_Study"), size=**as.numeric**(desired\_sizes\_vector), method="srswor")  
Salary\_strat<-**getdata**(education\_data,Strata\_idx)  
**summary**(Salary\_strat**$**Field\_of\_Study)

## Arts Mathematics Law Business   
## 70 70 70 70   
## Engineering Medicine Computer Science   
## 70 70 70

**library**(survey)  
Salary\_strat2=**data.frame**(Salary\_strat, pw=1**/**Salary\_strat**$**Prob, fpc=**c**(**rep**(749,70),**rep**(745,70),**rep**(727,70),**rep**(719, 70),**rep**(701,70),**rep**(689,70),**rep**(670,70)))  
Strata\_svy<-**svydesign**(id=**~**1,strata = **~**Field\_of\_Study, weights = **~**pw, data = Salary\_strat2, fpc=**~**fpc)  
Strata\_mean\_salary<-**svymean**(**~**Starting\_Salary, Strata\_svy)  
Strata\_mean\_salary

## mean SE  
## Starting\_Salary 51021 645.57

The stratified sampling performed better than the simple random sampling in this case. A proportional sampling could improve the results slightly but since the field of study almost has equal number of counts it would only improve the results slightly.

*3.10.3 Cluster Sampling*

all\_clusters <- **levels**(education\_data**$**Field\_of\_Study)  
num\_total\_clusters <- **length**(all\_clusters)  
num\_sampled\_clusters <- 3  
sampled\_cluster\_names <- **sample**(all\_clusters, size = num\_sampled\_clusters, replace = FALSE)  
cluster\_sample\_data <- education\_data **%>%**  
 **filter**(Field\_of\_Study **%in%** sampled\_cluster\_names)  
  
cluster\_sample\_data**$**cluster\_id <- **as.integer**(cluster\_sample\_data**$**Field\_of\_Study)   
  
cluster\_sample\_data**$**total\_clusters\_pop <- num\_total\_clusters  
  
cluster\_svy\_design <- **svydesign**(  
 id = **~**cluster\_id,  
 fpc = **~**total\_clusters\_pop, *# FPC for the clusters themselves*  
 data = cluster\_sample\_data  
)  
  
mean\_salary\_cluster <- **svymean**(**~**Starting\_Salary, cluster\_svy\_design)  
**cat**("**\n**Mean Starting\_Salary from Cluster Sample (Survey Package):**\n**")

##   
## Mean Starting\_Salary from Cluster Sample (Survey Package):

**print**(mean\_salary\_cluster)

## mean SE  
## Starting\_Salary 50713 280.2

The cluster sampling gives us the best result for estimating the population mean. Indicating that the ‘field of study’ is a suitable column for clustering.

*3.10.4 Intraclass Correlation Coefficient (ICC score)*

Now, computing the ICC score will tell us if the data meets the conditions for clustering. Here a score of 0 means the condition is satisfied, while 1 indicates correlation between clusters indicating unsuitability for clustering.

**library**(lme4)

## Warning: package 'lme4' was built under R version 4.4.3

icc\_model <- **lmer**(Starting\_Salary **~** (1 **|** Field\_of\_Study), data = education\_data)

## boundary (singular) fit: see help('isSingular')

variance\_components <- **as.data.frame**(**VarCorr**(icc\_model))  
var\_between\_clusters <- variance\_components**$**vcov[variance\_components**$**grp **==** "Field\_of\_Study"]  
var\_within\_clusters <- variance\_components**$**vcov[variance\_components**$**grp **==** "Residual"]  
icc\_score <- var\_between\_clusters **/** (var\_between\_clusters **+** var\_within\_clusters)  
icc\_score

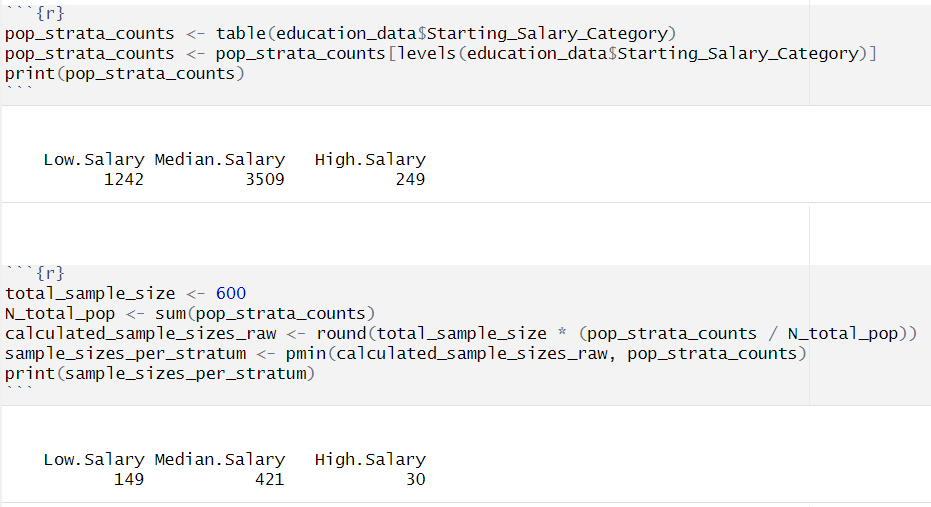
## [1] 0

Above we see an ICC score of 0. This indicates that there is no correlation between the clusters which is a necessary condition to perform cluster sampling.

3.11 *Re-implementing methods with Stratified Sampling*

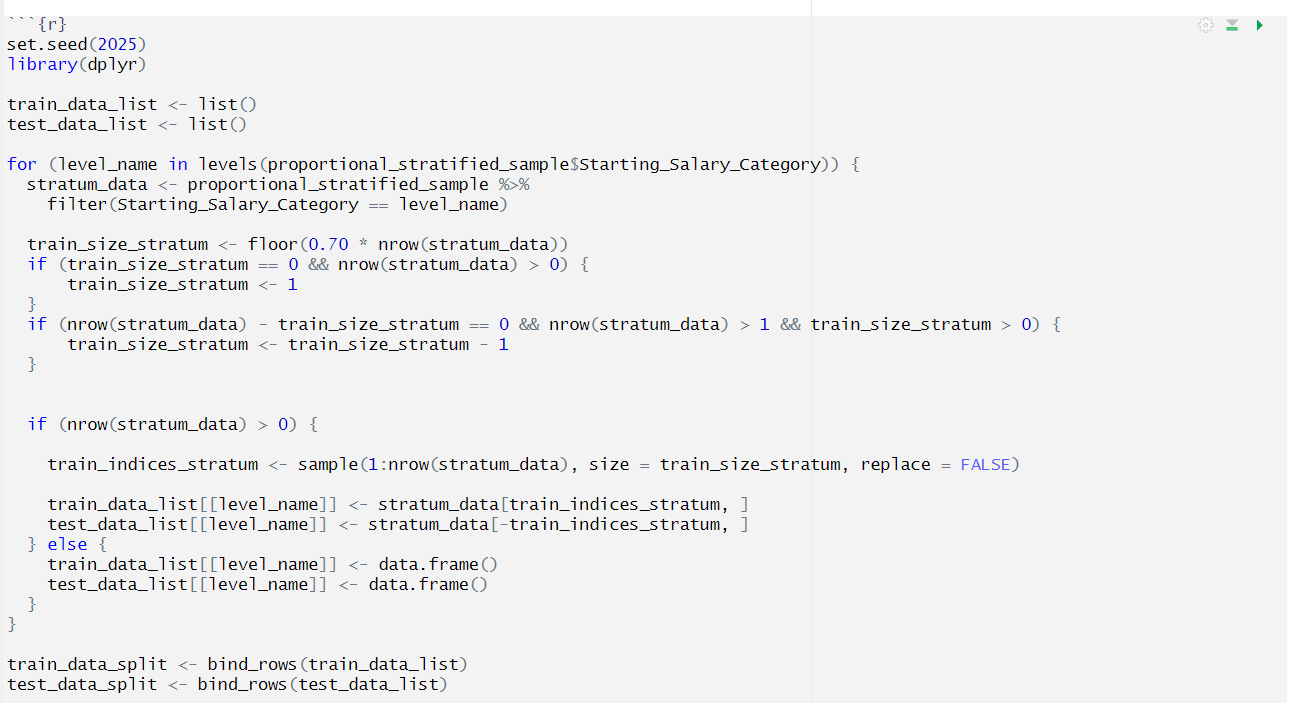
Based on feedback from the TAs, we can choose to use a stratified sampling and refit the model for QDA and classification trees to see if there is a better prediction to be obtained. Since the models and trees were unable to provide any predictions for the High Salary class a Simple random sample is not the best way to divide the test and train set. (Due to issues with knitting the remaining code chunks will be screenshots)

Taking a stratified sample with proportional allocation on the Starting\_Salary\_Category:

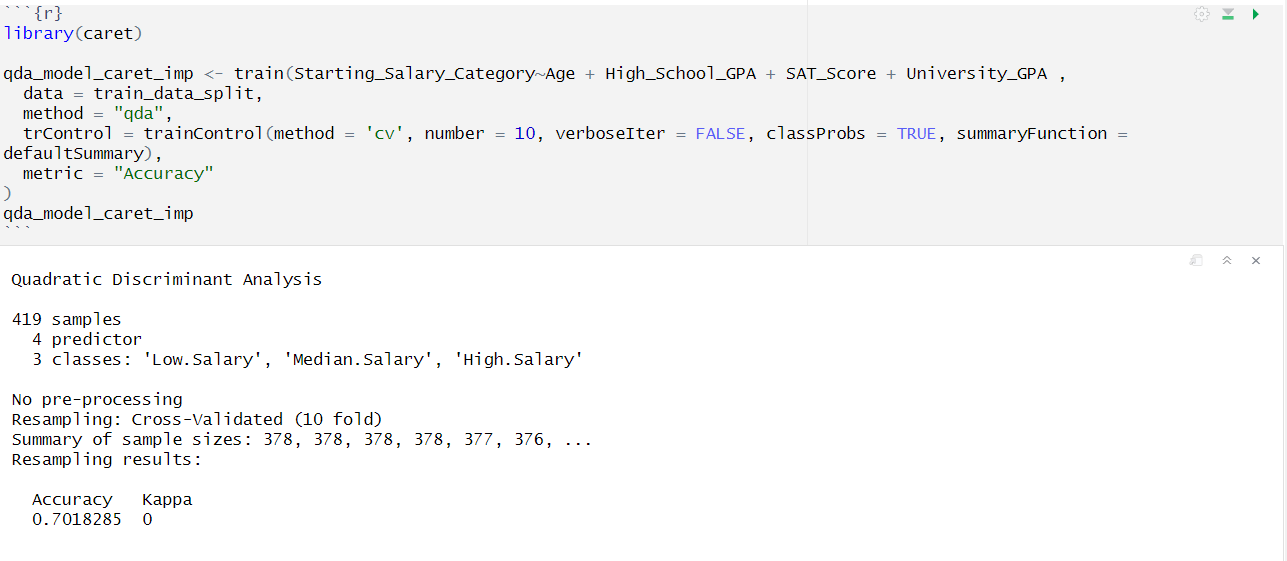


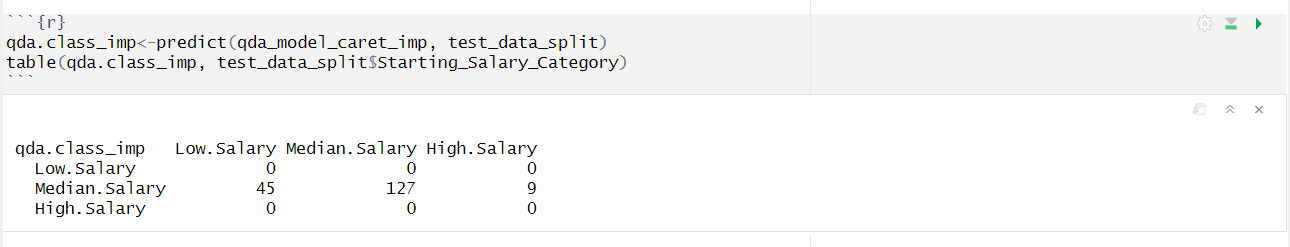


Splitting the data into test and train based on stratified sampling with proportional allocation:



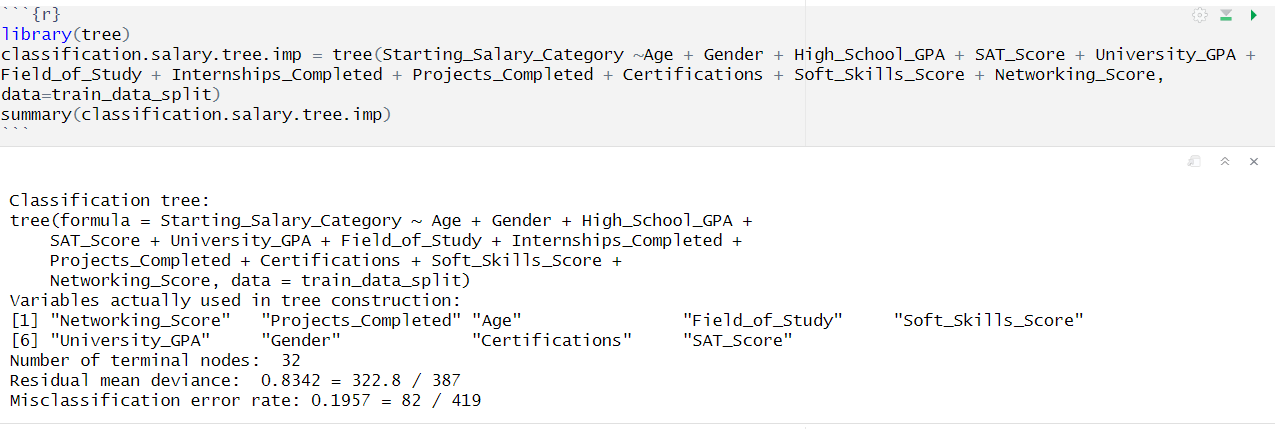
3.11.1 *Refitting the QDA model based on proportional stratified sampling:*





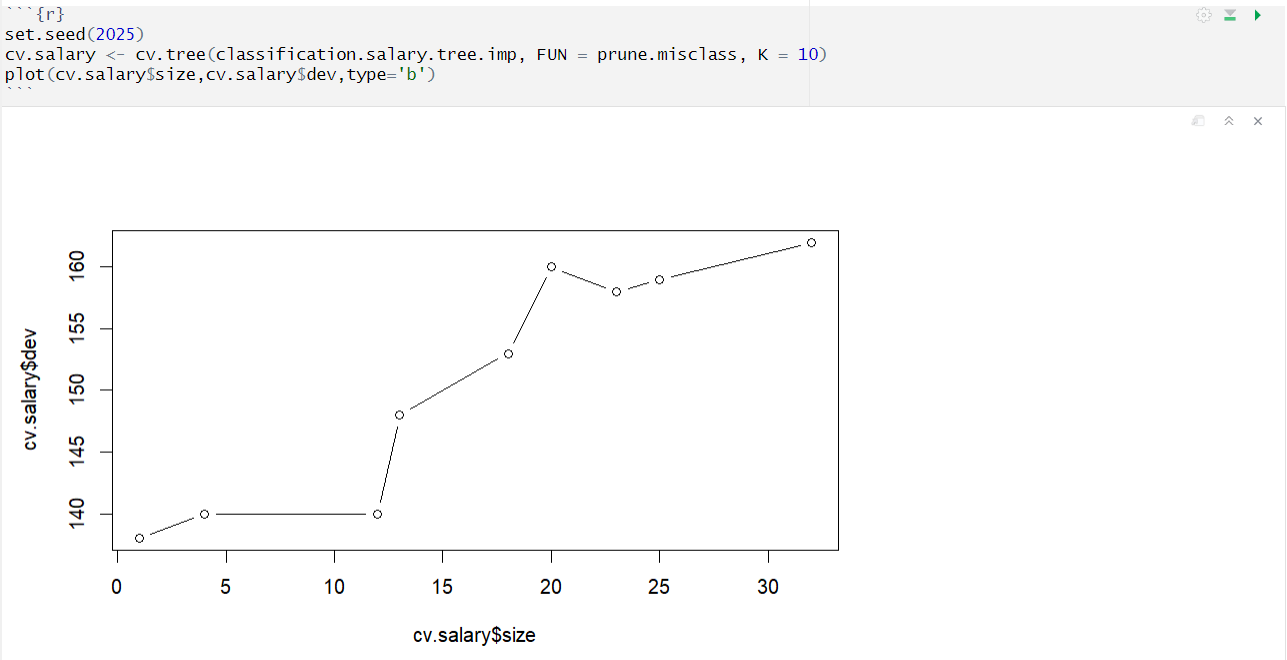
The results obtained are still the same after using a stratified sample with proportional allocation. It could be due to the High.salary class simply having too few observations.

3.11.2 *Refitting the Classification Tree*

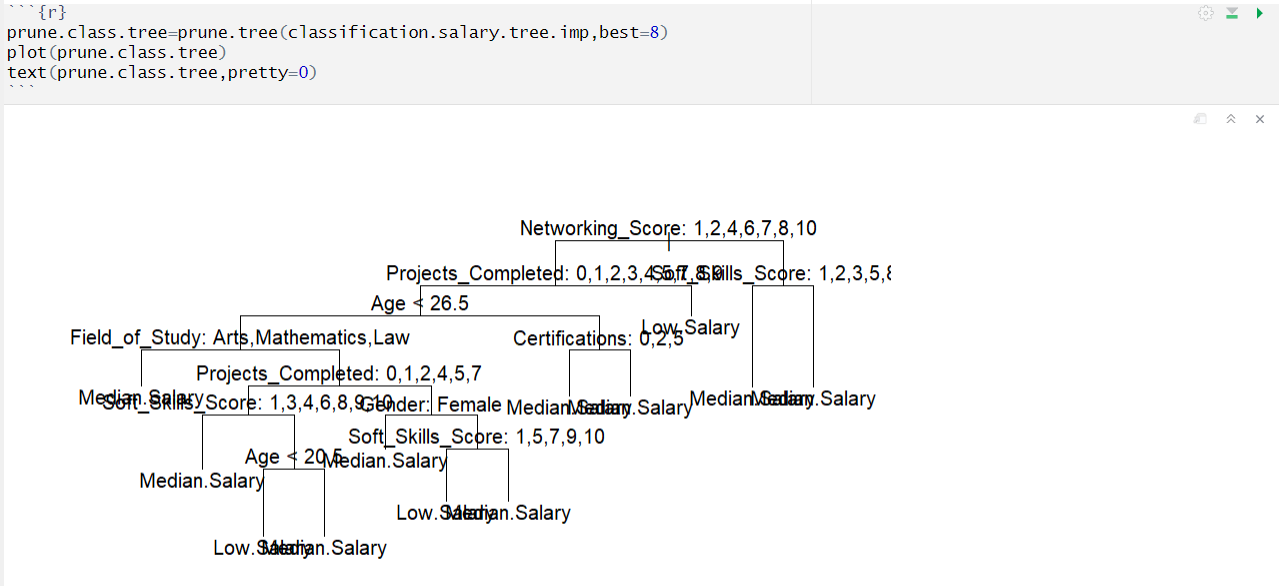


The tree performs better than the previous implementation where it was unable to grow more than a single node.

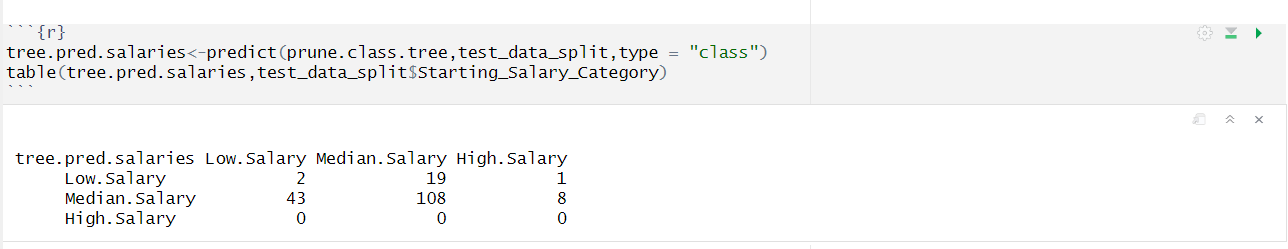
Using cross validation to check the best amount of nodes for the tree.



We can select any number of nodes from 5-11 as they have the same deviance. We select 8 as the best number of nodes to maintain readability.

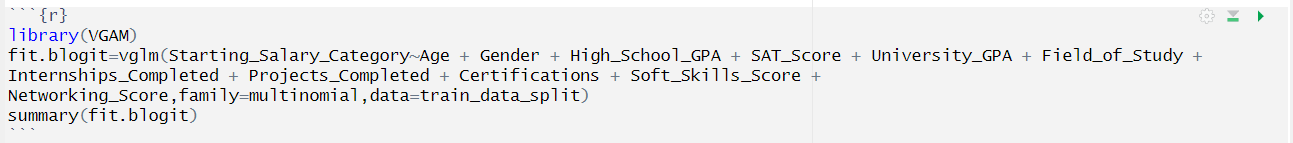


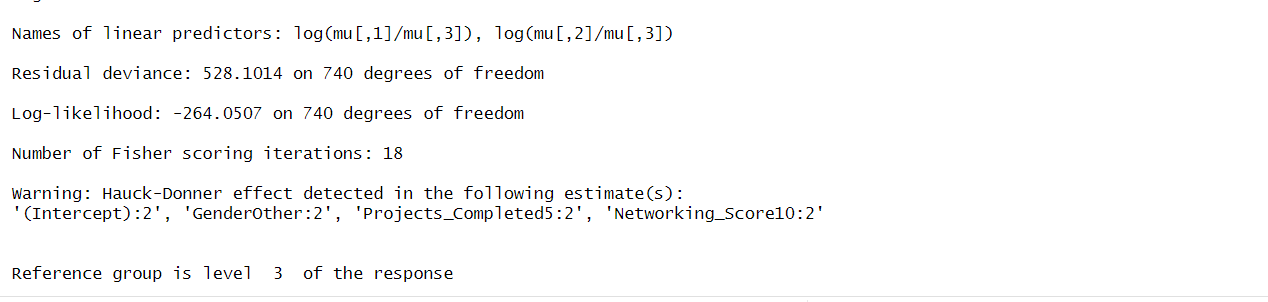
Now we can try predicting on the test class and see if the results are better from before:



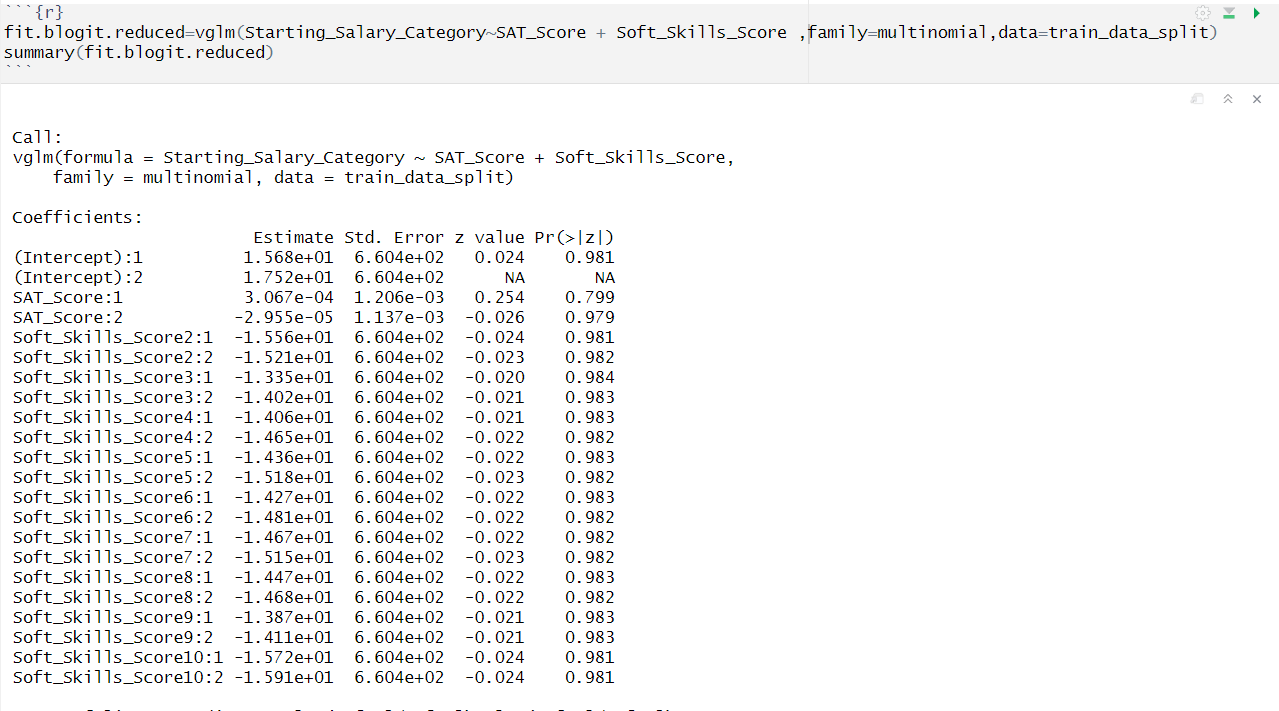
This tree performs a lot better than the previous one as there are a few more classifications for low salary. This is the best results we can obtain from stratified proportional sampling as for lower sample sizes the median salary takes up most of the observations. There are simply not enough observations for high.salary for the tree to predict that class.

3.11.3 *Implementing Multinomial Regression on the data*





The Hauck-Donner effect results in an upward biased p-value and loss of power, the aberration can lead to very damaging consequences such as in variable selection.



The reduced model uses only 3 predictors as defined from the Stepwise model, excluding networking score due to the Hauck-Donner effect.

However due to multiple levels of categorical variables it is very difficult to interpret and that p-value indicates that none of the predictors are significant. The best model so far is the classification tree with a misclassification rate of 28%.

**4 Conclusion**

**4.1 Summary:**

To summarize our analysis, we have discovered that the Classification Tree model proved to be the most suitable for predicting a student's starting salary category. Although both QDA and the tree model had the same misclassification rate (70%), QDA predicted only the median salary class, failing to differentiate between categories. In contrast, the Classification Tree, despite its similar accuracy, was able to capture some variation by predicting a few instances of lower salary, offering greater practical value. Furthermore, decision trees provided clear, interpretable decision rules based on key factors like internships, soft skills, and field of study. After pruning for simplicity and clarity, the tree structure allowed for actionable insights, making it the preferred model in this study.

The idea behind the project was to build a suitable regression model such that the co-coefficients such as average age, SAT score, University GPA can be substituted into the model and the estimated average salary can be obtained. This would have been an interesting find as students wanting to enroll in a university can enter their field of study, age and other factors to see what their starting salary would be so that they can choose their major respectively. The results would be close to accurate as even though the data is synthetically generated it follows real world education trends.

**4.2 Key Findings and Insights:**

Through the modeling analysis portion of our project we were able to effectively answer the 3 main guiding questions:

1. *What educational and personal factors are most predictive of a student’s starting salary category?*

Initially, based on EDA we found that Internships are likely important, and high-ranking universities may correlate with higher salaries. Next, based on the comprehensive analysis conducted across multiple methods (including Multiple linear regression, QDA, and classification trees), the most predictive educational and personal factors for a student’s starting salary category are a mix of academic performance—such as High School GPA, SAT scores, and University GPA—and personal factors like soft skills, networking ability, and practical experience through projects.Additionally, the student’s field of study plays an important role, reflecting how different majors impact salary outcomes. Together, these educational and personal factors provide the strongest insight into early career salary levels.

1. *Does a classification model perform better if the regression models are not a good fit?*

The analysis indicates that Career Satisfaction and Work-Life Balance are moderately influenced by both educational and personal factors. Students with higher University GPA, more certifications, and greater involvement in projects or internships tended to report higher career satisfaction. Similarly, strong soft skills and networking scores—reflecting interpersonal and professional competencies—were positively associated with better work-life balance. These outcomes suggest that beyond academic performance, experiential learning and personal development play a critical role in shaping post-graduation career experiences.

1. *Are there distinct clusters of students with similar profiles in terms of education, skills, and career outcomes?*

Yes, the analysis reveals distinct clusters of students with similar profiles based on their education, skills, and career outcomes. Using classification trees and sampling methods, students were grouped by key variables such as Field of Study, Soft Skills Score, Networking Score, Projects Completed, and SAT Score. These clusters showed varying patterns in Starting Salary Categories, with some clusters consistently linked to higher or lower salary outcomes. Additionally, cluster sampling based on Field of Study produced a close estimate of the population mean salary, and the Intraclass Correlation Coefficient (ICC) score of 0 confirmed that clustering by Field of Study was appropriate. This supports the presence of meaningful student groupings with shared educational and career characteristics.

Overall, our modeling analysis confirmed that both academic and personal development factors significantly influence a student’s career outcomes. We found that starting salary, career satisfaction, and work-life balance are shaped by a combination of GPA, standardized test scores, soft skills, networking, and hands-on experience. Additionally, students naturally group into distinct clusters based on these attributes, highlighting the importance of holistic development in education-to-career pathways.