1. Introduction

This report provides a comprehensive analysis on a Student Admission Classification dataset, focusing on data validation, handling missing values, detecting and removing outliers, and performing statistical analysis of numerical and categorical variables. This dataset, Student Admission Classification dataset, is to classify the status of students' college application whether it is to be accepted or rejected. This dataset has six independent variables with four categorical columns and two numeric columns and it has one dependent variable which is status of admission.

2. Objectives

The primary objectives of this report are:

- ✓ To perform data validation and data preprocessing (cleaning) by identifying and removing any duplicate entries, handling missing data appropriately for both numerical and categorical variables and detecting and removing outliers from the dataset.
- ✓ To perform a statistical analysis of numerical variables, calculating measures such as mean, median, mode, variance, and standard deviation.
- ✓ To create visualizations for both categorical and numerical data for better insights.

So the objectives of this report is to carry out data analysis in R, applying knowledge of statistical programming from data importing to reporting key findings.

3. Methodology

1. Data Validation: Checking for inconsistencies such as duplicates and missing values.
2. Data Cleaning: Removing duplicates, handling missing values for both numerical and categorical variables.
3. Outlier Detection: Identifying and removing outliers from the dataset.
4. Statistical Data Analysis: Analyzing the measures of central tendency and dispersion for the numerical variables.
5. Data Visualization: Creating visual representations for both numerical and categorical variables.
6. R Code Integration: All steps will be implemented in R, and relevant outputs will be included.

3.1 Data Import

The dataset is imported using the **read.csv()** function in R, and the structure and summary statistics of the dataset were inspected to understand the data types and identify potential data issues. The dataset is downloaded from: https://www.kaggle.com/datasets/rosiellenpassos/student-admission-data and locally stored as student_data. Then it is loaded into Rstudio as follows.

```r
# Load required libraries
library(ggplot2)

# Load the dataset
student_data <- read.csv("C:/Users/smrc/Desktop/StudentData.csv", na.strings = c("", "NA"))

# View the structure of the dataset
str(student_data)
summary(student_data)

```

Output:

```
> str(student_data)
'data.frame':   403 obs. of  7 variables:
 $ TOEFL_Score             : int  118 107 104 110 103 115 109 101 102 108 ...
 $ University_Rating       : chr  "Very Good" "Very Good" "Good" "Good" ...
 $ Stetement_of_Purpose    : chr  "Excellent" "Very Good" "Good" "Very Good" ...
 $ Letter_of_Recommendation: chr  "Excellent" "Excellent" "Very Good" "Good" ...
 $ CGPA                    : num  3.86 3.55 3.2 3.47 3.28 ...
 $ Research                : chr  "Yes" "Yes" "Yes" "Yes" ...
 $ Status_of_Admission     : chr  "Accepted" "Accepted" "Accepted" "Accepted" ...
>
```

```
> summary(student_data)
  TOEFL_Score    University_Rating  Stetement_of_Purpose Letter_of_Recommendation      CGPA
 Min.   : 92.0   Length:403         Length:403           Length:403                Min.   :2.720
 1st Qu.:103.0   Class :character   Class :character     Class :character          1st Qu.:3.265
 Median :107.0   Mode  :character   Mode  :character     Mode  :character          Median :3.440
 Mean   :107.4                                                                    Mean   :3.437
 3rd Qu.:112.0                                                                    3rd Qu.:3.623
 Max.   :120.0                                                                    Max.   :3.964
 NA's   :3                                                                        NA's   :5
   Research         Status_of_Admission
 Length:403         Length:403
 Class :character   Class :character
 Mode  :character   Mode  :character
```

Originally, the dataset has 403 observations of 7 variables with six independent variables and one dependent variable. There are two numeric columns and four categorical columns contained in the independent variables.

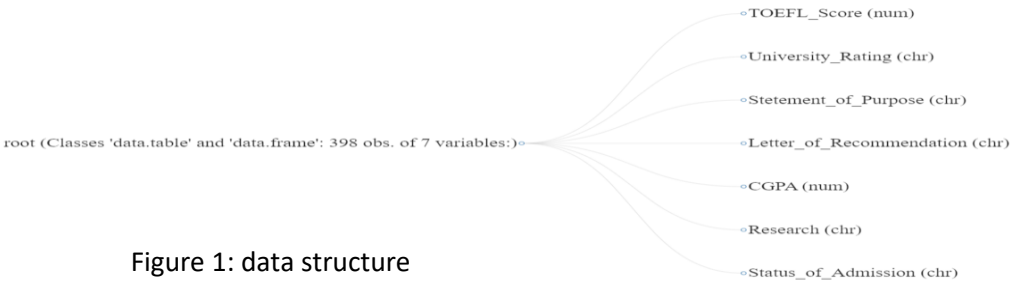

Figure 1: data structure

3.2 **Data Validation**

The dataset was validated for duplicate entries, missing values, and outliers. Each of these issues was addressed as follows:

3.2.1 **Duplicates**

I have checked for any duplicate entries using **duplicated()** function.

```
> sum(duplicated(student_data))
[1] 4
> # Remove duplicate rows
> student_data <- unique(student_data)
```

Four rows of duplicated entries are identified, as we can see in the above output.

Then the duplicated entries are removed using:

```
11  data <- data[!duplicated(data), ]
12
```

And I checked again to make sure its removal.

```
> sum(duplicated(student_data))
[1] 0
```

3.2.2 **Handling Missing Values**

Checking for total missing values in the dataset by using:

```
> colSums(is.na(student_data))
             TOEFL_Score        University_Rating     Stetement_of_Purpose Letter_of_Recommendation
                       3                        1                        3                        0
                    CGPA                 Research      Status_of_Admission
                       5                        3                        0
>
```

15 missing values are identified and missing values in numerical variables were replaced with the column mean, while for categorical variables, missing values were replaced with the mode (most frequent category).

```
# Impute missing numerical values with the mean
student_data$TOEFL_Score[is.na(student_data$TOEFL_Score)] <- mean(student_data$TOEFL_Score, na.rm = TRUE)
student_data$CGPA[is.na(student_data$CGPA)] <- mean(student_data$CGPA, na.rm = TRUE)

# Impute missing categorical values with the most frequent level
student_data$University_Rating[is.na(student_data$University_Rating)] <- names(which.max(table(student_data$University_Rating)))
student_data$Stetement_of_Purpose[is.na(student_data$Stetement_of_Purpose)] <- names(which.max(table(student_data$Stetement_of_Purpose)))
student_data$Letter_of_Recommendation[is.na(student_data$Letter_of_Recommendation)] <- names(which.max(table(student_data$Letter_of_Recommendation)))
student_data$Research[is.na(student_data$Research)] <- names(which.max(table(student_data$Research)))
```

Now, checking again for missing values to make sure that they are replaced.

```
> sum(is.na(student_data))
[1] 0
```

Now, the missing values are replaced.

3.2.3 **Encoding Categorical Variables**

Ordinal categorical variables are converted to factors with ordered levels, ensuring appropriate representation and analysis.

```
# Convert ordinal categorical variables to factors with ordered levels

student_data$University_Rating <- factor(student_data$University_Rating, levels = c("Very Poor", "Poor", "Good", "Very Good", "Excellent"), orderec
student_data$Stetement_of_Purpose <- factor(student_data$Stetement_of_Purpose, levels = c("Poor", "Good", "Very Good", "Excellent"), ordered = TRUE
student_data$Letter_of_Recommendation <- factor(student_data$Letter_of_Recommendation, levels = c("Poor", "Good", "Very Good", "Excellent"), ordere
student_data$Research <- factor(student_data$Research, levels = c("No", "Yes"), ordered = TRUE)
student_data$Status_of_Admission <- factor(student_data$Status_of_Admission, levels = c("Rejected", "Accepted"), ordered = TRUE)
```

The abbove code converts several ordinal categorical variables into factors with explicitly ordered levels. In R, factors are used to handle categorical data, and when the categories have a meaningful order (ordinal data), we can define the levels to represent that order. This process is important for ensuring correct data analysis and modeling since ordinal data needs to reflect the natural ranking of the categories. Here's a detailed explanation of each variable conversion:

University_Rating: The variable University_Rating is transformed into a factor with five ordered levels: "Very Poor", "Poor", "Good", "Very Good", "Excellent". The ordering implies that "Very Poor" has the lowest rating and "Excellent" has the highest, establishing a rank or hierarchy among the categories.

Stetement_of_Purpose: The Stetement_of_Purpose variable is converted into a factor with four ordered levels: "Poor", "Good", "Very Good", "Excellent". Here, the quality of the statement of purpose ranges from "Poor" (lowest) to "Excellent" (highest).

Letter_of_Recommendation: The Letter_of_Recommendation variable is turned into a factor with four ordered levels: "Poor", "Good", "Very Good", "Excellent". This implies a ranking system for recommendation letters, from a poor letter to an excellent one, which can influence a student's admission decision.

Research: The Research variable, which indicates whether a student has research experience, is converted into a factor with two ordered levels: "No" (no research experience), "Yes" (has research experience). This binary variable reflects the absence or presence of research experience, with "Yes" being ranked higher than "No".

Status_of_Admission: The Status_of_Admission variable, which shows whether the student was accepted or rejected, is converted into a factor with two ordered levels: "Rejected", "Accepted". This ordering establishes "Rejected" as the lower level and "Accepted" as the higher level, indicating the admission decision outcome.

3.2.4 Outliers Detection and Removal

Boxplots were used to identify outliers in numerical variables.

```
47  # Generate boxplots for numerical variables
48  boxplot(student_data$TOEFL_Score, main = "Boxplot of TOEFL Score (Before Outlier Removal)")
49  boxplot(student_data$CGPA, main = "Boxplot of CGPA (Before Outlier Removal)")
```

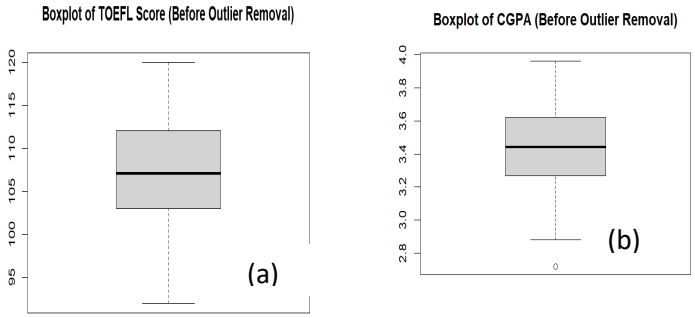

Figure 2: (a) boxplot for TOFL column, (b) boxplot for CGPA column.

The CGPA boxplot shows the presence of outliers as individual points lying outside the whiskers of the boxplots while there is no any outliers for TOEFL Score boxplot.

```
# remove outliers using the IQR method
outlier_remover <- function(x) {
  Q1 <- quantile(x, 0.25)
  Q3 <- quantile(x, 0.75)
  IQR <- IQR(x)
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR
  x[x < lower_bound | x > upper_bound]
}
outliers_CGPA <- outlier_remover(student_data$CGPA)

student_data <- subset(student_data, !(TOEFL_Score %in% outliers_T

# Generate boxplots after outlier removal
boxplot(student_data$CGPA, main = "Boxplot of CGPA (After Outlier
```

The as we can see, the above command is used to remove the outlier from CGPA values.

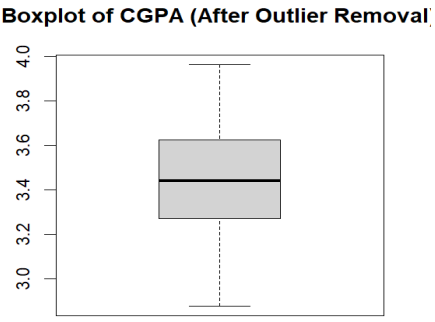

Figure 3: Boxplot of CGPA

outliers_CGPA <- outlier_remover(student_data$CGPA): Applies the outlier_remover function to the "CGPA" column of the student_data dataframe and stores the identified outliers in the outliers_CGPA variable.

boxplot(student_data$CGPA, main = "Boxplot of CGPA (After Outlier Removal)"): Creates a boxplot of the "CGPA" column after outlier removal.

Descriptive Statistics

Descriptive statistics is calculated for numerical variables grouped by admission status.

```
# Installing the modeest package
install.packages("modeest")
# Loading the modeest package
library(modeest)
# Grouping by admission status
admission_groups <- split(student_data, student_data$Status_of_Admission)

# Calculate descriptive statistics for each group
descriptive_stats <- lapply(admission_groups, function(group) {
  summary_data <- summary(group[, c("TOEFL_Score", "CGPA")])
  mode_TOEFL <- mfv(group$TOEFL_Score)
  mode_CGPA <- mfv(group$CGPA)
  sd_data <- apply(group[, c("TOEFL_Score", "CGPA")], 2, sd)
  var_data <- apply(group[, c("TOEFL_Score", "CGPA")], 2, var)
  return(list(Summary = summary_data, Mode_TOEFL = mode_TOEFL, Mode_CGPA = mode_CGPA,
              Standard_Deviation = sd_data, Variance = var_data))
})

# Print the descriptive statistics
print(descriptive_stats)

```

The code installs the **modeest** package and loads it into the R session. This package is required to calculate the mode (most frequent value) of the numerical variables.

The **split()** function groups the dataset **student_data** based on the variable Status_of_Admission (which has values "Accepted" or "Rejected"). This creates two groups: one for students who were accepted and one for those who were rejected.

The lapply() function is used to apply descriptive statistics calculations to each of the two groups (Accepted and Rejected) :

summary(): Provides the minimum, first quartile, median, mean, third quartile, and maximum for two numerical variables: TOEFL_Score and CGPA.

mfv(): Calculates the mode (most frequent value) for the TOEFL_Score and CGPA variables.

apply() with sd: Calculates the standard deviation for TOEFL_Score and CGPA.

apply() with var: Calculates the variance for TOEFL_Score and CGPA.

The **print()** function outputs the descriptive statistics for both the "Accepted" and "Rejected" groups.

Output:

```
> print(descriptive_stats)
$Accepted
$Accepted$Summary
  TOEFL_Score         CGPA
 Min.   : 98.0   Min.   :3.056
 1st Qu.:107.0   1st Qu.:3.426
 Median :110.0   Median :3.560
 Mean   :110.4   Mean   :3.563
 3rd Qu.:114.0   3rd Qu.:3.678
 Max.   :120.0   Max.   :3.964

$Accepted$Mode_TOEFL
[1] 110

$Accepted$Mode_CGPA
[1] 3.504

$Accepted$Standard_Deviation
TOEFL_Score        CGPA
  5.0288264   0.1794997

$Accepted$Variance
TOEFL_Score        CGPA
25.28909516  0.03222014
```

The minimum TOEFL score is 98, and the maximum score is 120.
The median score is 110, and the mean score is 110.4, indicating that most students have a TOEFL score around 110.
The minimum CGPA is 3.056, and the maximum is 3.964.
The median CGPA is 3.560, and the mean CGPA is 3.563, suggesting that students who were accepted have relatively high academic performance.
TOEFL_Score Mode: The most frequent TOEFL score is 110.
CGPA Mode: The most frequent CGPA value is 3.504.
TOEFL_Score: The standard deviation is 5.03, meaning that TOEFL scores among accepted students vary by about 5 points from the mean.
CGPA: The standard deviation is 0.179, indicating that CGPAs are quite consistent across students, with little variation from the mean.
TOEFL_Score: The variance is 25.29, reinforcing the observation that there is moderate variation in TOEFL scores.
CGPA: The variance is 0.032, showing a relatively low spread in CGPAs.

Rejected Group:

```
$Rejected
$Rejected$Summary
  TOEFL_Score         CGPA
 Min.   : 92.0   Min.   :2.880
 1st Qu.: 99.5   1st Qu.:3.144
 Median :102.0   Median :3.228
 Mean   :102.7   Mean   :3.240
 3rd Qu.:106.0   3rd Qu.:3.366
 Max.   :114.0   Max.   :3.688

$Rejected$Mode_TOEFL
[1] 100

$Rejected$Mode_CGPA
[1] 3.2

$Rejected$Standard_Deviation
TOEFL_Score        CGPA
   4.280332    0.165157

$Rejected$Variance
TOEFL_Score        CGPA
18.32124079  0.02727685
```

The minimum TOEFL score is 92, and the maximum is 114.
The median TOEFL score is 102, and the mean is 102.7, indicating that rejected students tend to have lower TOEFL scores compared to accepted students.
The minimum CGPA is 2.880, and the maximum is 3.688.
The median CGPA is 3.228, and the mean is 3.240, suggesting that rejected students generally have lower CGPAs than accepted students.
TOEFL_Score Mode: The most frequent TOEFL score is 100.
CGPA Mode: The most frequent CGPA value is 3.2.
TOEFL_Score: The standard deviation is 4.28, meaning that TOEFL scores among rejected students are less spread out compared to the accepted group.
CGPA: The standard deviation is 0.165, showing slightly less variation in CGPAs compared to the accepted group.
TOEFL_Score: The variance is 18.32, indicating moderate spread in TOEFL scores among rejected students, but lower than the accepted group.
CGPA: The variance is 0.027, reflecting the slightly lower spread in CGPAs among rejected students.

Data Visualization and Analysis

CGPA

```
39  hist(data$CGPA[data$Status_of_Admission == "1"],
40       xlim = c(2.7, 4),
41       breaks = 10,       # Set a specific number of bins
42       main = "Histogram of CGPA for Accepted Students",
43       xlab = "CGPA Score")
44
45  hist(data$CGPA[data$Status_of_Admission == "0"],
46       xlim = c(2.7, 4),
47       breaks = 10,       # Set a specific number of bins
48       main = "Histogram of CGPA for Rejected Students",
49       xlab = "CGPA Score")
```

The above commands are used to visualize the CGPA of both accepted and rejected students.

hist(...): Creates a histogram in R. data$CGPA[data$Status_of_Admission == "1"/"0"]: Subsets the data to select CGPA scores 1 for accepted and 0 for rejected students.

xlim = c(2.7, 4): Sets the x-axis range (CGPA scores) from 2.7 to 4.0.

breaks = 10: Divides the data into 10 bins for visualization.

main =" " Sets the title of the histogram and **xlab = "CGPA Score":** Labels the x-axis.

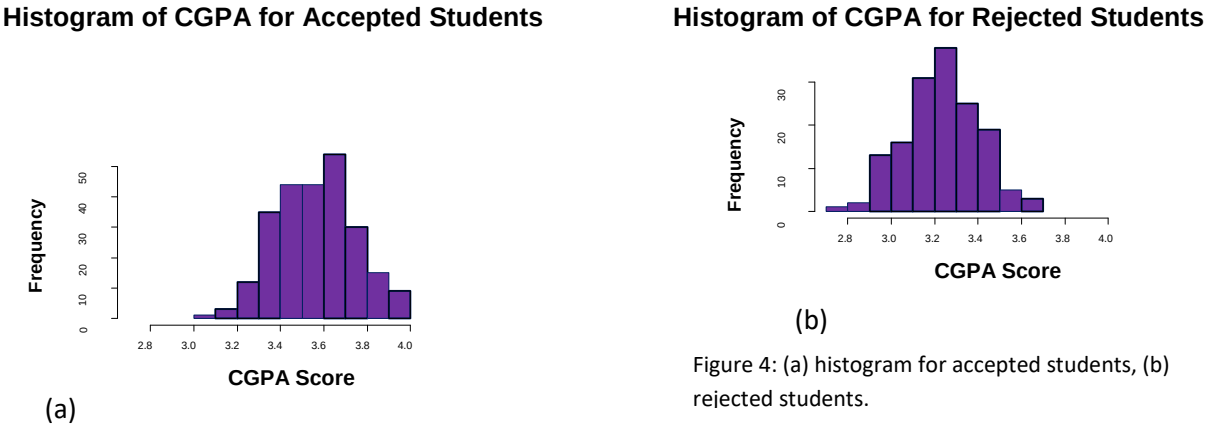

Figure 4: (a) histogram for accepted students, (b) rejected students.

The histogram in (a) shows a right-skewed distribution of CGPA scores for accepted students and histogram in (b) shows left-skewed distribution of CGPA.

1. CGPA and Acceptance:

Clear Relationship: There's a noticeable difference in the CGPA distributions between accepted and rejected students, suggesting CGPA plays a significant role in admission decisions.

Higher CGPA Favored: The histogram for accepted students is shifted to the right (higher CGPA values) compared to the histogram for rejected students. This indicates that applicants with higher CGPAs have a greater chance of acceptance.

2. Distribution Characteristics:

Accepted Students:

The distribution is somewhat right-skewed, meaning there are more students with CGPAs clustered towards the higher end (3.4 to 3.7). A good number of accepted students have CGPAs above 3.5.

Rejected Students:

The distribution appears more clustered around the center, with a peak around 3.2 to 3.4.

Fewer students in this group have CGPAs above 3.6 compared to the accepted group.

3. Overlap and Other Factors:

Overlap: While higher CGPAs are generally favored, there's some overlap in the distributions. This means some students with lower CGPAs were accepted, while others with higher CGPAs were rejected. This highlights that CGPA is likely not the only factor considered for admission. Other factors like standardized test scores, letters of recommendation, essays, and extracurricular activities could also play a role.

Outliers: There don't appear to be any significant outliers (extremely low or high values) in the CGPA scores of accepted students.

Overall, the histogram suggests that a higher CGPA generally increases the likelihood of acceptance, as the majority of accepted students have CGPAs above the midpoint of the displayed range. However, the presence of overlap emphasizes that admission decisions are likely based on a holistic evaluation of applicants

TOEFL Score

```
#Data Visualization
# Boxplot for TOEFL Score by Admission Status
ggplot(student_data, aes(x = Status_of_Admission, y = TOEFL_Score, fill = Status_of_Admission)) +
  geom_boxplot() +
  labs(title = "TOEFL Score Distribution by Admission Status", x = "Admission Status", y = "TOEFL Score") +
  theme_minimal()

```

ggplot(student_data, aes(...)): This initializes the ggplot function, which is used for creating visualizations in R. The student_data dataset is specified as the data source.

aes(x = Status_of_Admission, y = TOEFL_Score, fill = Status_of_Admission) defines the aesthetic mappings for the plot:

x = Status_of_Admission: The admission status (Accepted or Rejected) is mapped to the x-axis.

y = TOEFL_Score: The TOEFL score is mapped to the y-axis.

fill = Status_of_Admission: The boxplot is color-coded based on the admission status.

geom_boxplot(): This function creates a boxplot to display the distribution of TOEFL scores for each admission status. Boxplots are useful for visualizing the summary statistics (median, quartiles, range) and potential outliers.

labs(...): The labs() function is used to add labels and a title to the plot:

title = "TOEFL Score Distribution by Admission Status" sets the plot's title.

x = "Admission Status" labels the x-axis as "Admission Status."

y = "TOEFL Score" labels the y-axis as "TOEFL Score."

theme_minimal(): This applies the minimal theme to the plot, which removes background elements and focuses on a clean, simple design.

Output:

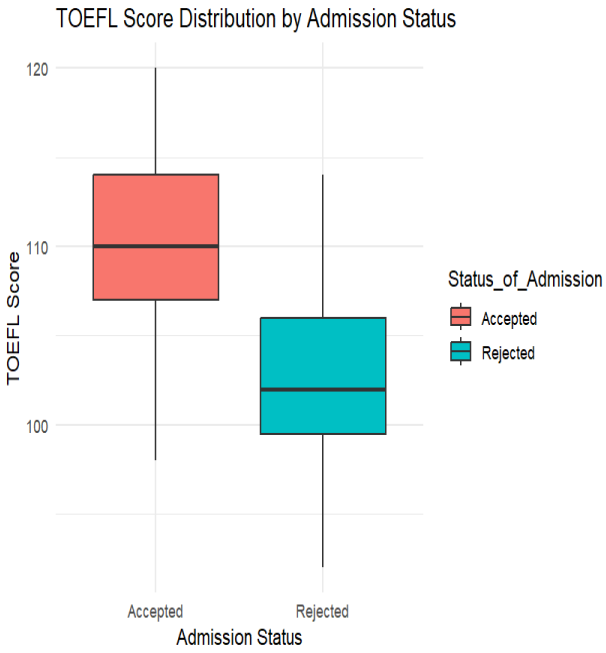

Figure 5 visualizes the distribution of TOEFL scores based on students' Admission Status (Accepted vs. Rejected):

The TOEFL scores of accepted students are generally higher, with the median around 110.

The interquartile range (IQR), represented by the box, shows that the majority of TOEFL scores fall between 107 and 114.

The whiskers indicate that the minimum score for accepted students is approximately 98, and the maximum is 120.

The relatively taller box indicates more spread in the TOEFL scores among accepted students.

Figure 5: Boxplot for TOEFL Score

Rejected students have lower TOEFL scores on average, with a median around 102.

The IQR shows that most rejected students' scores are between 99.5 and 106.

The minimum score for rejected students is about 92, while the maximum is around 114.

The distribution of TOEFL scores among rejected students is more concentrated, as shown by the slightly narrower box.

Insights:

Comparison: Accepted students tend to have higher TOEFL scores than rejected students. The median score for accepted students (110) is notably higher than for rejected students (102).

Variability: While both groups have some overlap in TOEFL scores, accepted students exhibit more variability, especially in the upper range (as shown by the longer whiskers and wider IQR).

Conclusion: TOEFL scores appear to be an important factor in admission decisions, with higher scores being associated with acceptance.

University Rating vs **Admission Status**

```
# Bar plot for University Rating by Admission Status
ggplot(student_data, aes(x = University_Rating, fill = Status_of_Admission)) +
  geom_bar(position = "dodge") +
  labs(title = "University Rating by Admission Status", x = "University Rating", y = "Count") +
  theme_minimal()
```

ggplot(student_data, aes(...)): Initializes a ggplot for data visualization.

aes(x = University_Rating, fill = Status_of_Admission): Maps University_Rating to the x-axis.

fill = Status_of_Admission: The bars are color-coded by Status_of_Admission (Accepted or Rejected).

geom_bar(position = "dodge"): Creates a bar plot where bars for different admission statuses (Accepted or Rejected) are placed side by side (dodged) to compare their counts across university ratings. labs(...): Adds labels and title:

title = "University Rating by Admission Status" sets the plot title.

x = "University Rating" labels the x-axis as "University Rating."

y = "Count" labels the y-axis as "Count."

theme_minimal(): Applies the minimal theme to create a clean and simple design, without distracting background elements.

Output:

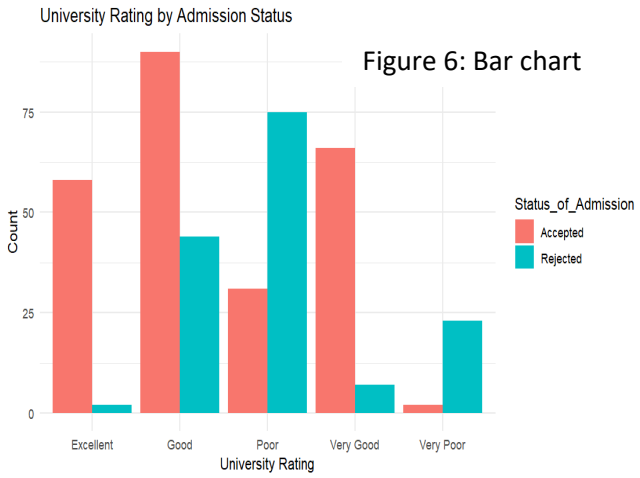

Figure 6: Bar chart

Figure 6 compares the counts of students who were accepted (red) and rejected (blue) across different University Rating categories.

Observations:

Excellent rating: More students were accepted than rejected.

Good rating: A significant number of accepted students compared to rejected, though the gap is not as large.

Poor rating: The count of rejected students is much higher than accepted students.

Very Good rating: Similar pattern to Excellent, with more students accepted than rejected.

Very Poor rating: The majority of students in this category were rejected. This visualization provides insight into how the University Rating affects admission outcomes, showing a clear correlation between higher university ratings and higher acceptance rates. Conversely, lower ratings tend to result in more rejections.

Research Experience vs Admission Status

The following code generates a pie chart to visualize the relationship between research experience and admission status for students in a dataset.

Creating the Pie Chart:

geom_bar(position = "fill"): This creates a stacked bar chart where the height of each segment represents the proportion of students with and without research experience.

coord_polar("y", start = 0): This transforms the bar chart into a pie chart by using polar coordinates, starting from the top (0 degrees).

facet_wrap(~ Status_of_Admission): This creates separate pie charts for each level of the Status_of_Admission variable (e.g., Accepted, Rejected). Each pie chart visualizes the distribution of research experience among students within each admission status category.

```
# Pie chart for Research by Admission Status
ggplot(student_data, aes(x = "", fill = Research)) +
  geom_bar(position = "fill") +
  coord_polar("y", start = 0) +
  facet_wrap(~ Status_of_Admission) +
  labs(title = "Research Experience by Admission Status", x = NULL, y = NULL, fill = "Research Experience") +
  theme_minimal() +
  theme(axis.text.x = element_blank())
```

Output:

Figure 7 provides insights into the relationship between research experience and admission status

- ✓ Each pie chart represents the proportion of students with and without research experience within each admission status category.
- ✓ The segments of the pie chart indicate the percentage of students who have research experience (Yes) versus those who do not (No).

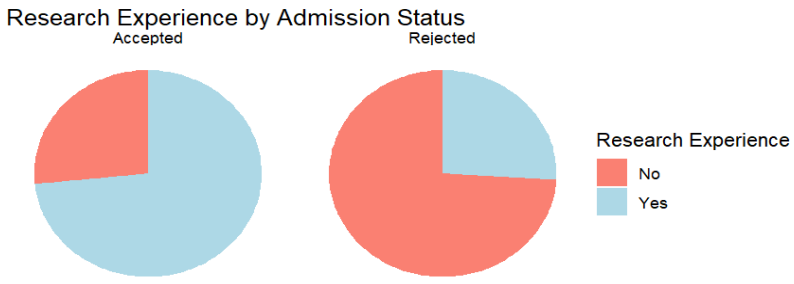

Figure 7: Pie chart of research column

A significantly larger portion of accepted students had research experience. This suggests that having research experience is a strong positive factor in the admission decision.

The majority of rejected students did not have research experience. This reinforces the idea that research experience is a significant factor in the selection process.

Letter of Recommendation vs Admission Status

The following code generates a stacked bar plot using the ggplot2 library in R.

geom_bar(position = "stack"): This adds a bar geometry layer to the plot.

position = "stack": Specifies that the bars for each Letter of Recommendation quality level should be stacked on top of each other, representing the count of both "Accepted" and "Rejected" students within each category.

x = Letter_of_Recommendation: Sets the x-axis to represent the different levels of Letter of Recommendation quality.

fill = Status_of_Admission: Indicates that the bars will be stacked and color-coded based on Admission Status (Accepted or Rejected).

labs(title = "...", x = "...", y = "..."): This adds labels to the plot

```
# Stacked bar plot for Letter of Recommendation by Admission Status
ggplot(student_data, aes(x = Letter_of_Recommendation, fill = Status_of_Admission)) +
  geom_bar(position = "stack") +
  labs(title = "Letter of Recommendation Quality by Admission Status", x = "Letter of Recommendation Quality", y = "Count") +
  theme_minimal()
```

Output:

As illustrated in figure 8, the stacked bar plot visually represents the relationship between the quality of Letters of Recommendation and student admission status

Figure 8: Stack chart

Strong Letters are Crucial: The "Very Good" and “Excellent“ category has the highest bar, with a large proportion of those students being accepted. This suggests that strong letters of recommendation are a very positive factor in the admission process.

Good Letters Still Matter: The "Good" category also shows a higher proportion of accepted students compared to rejected students, indicating that good letters are still important.

Poor Letters are Detrimental: The "Poor" and category has a much higher proportion of rejected students, highlighting that weak letters of recommendation significantly decrease the chances of acceptance. The “very poor” category does not have any accepted students, indicating the impact of recommendation.

Statement of Purpose vs Admission Status

The code generates a stacked bar plot using the ggplot2 library to visualize the relationship between the quality of a student's Statement of Purpose and their admission status.
ggplot(student_data, aes(x = Stetement_of_Purpose, fill = Status_of_Admission)): This initiates the ggplot object. geom_bar(position = "stack"): Adds a bar geometry layer.
position = "stack": Stacks bars for each Statement of Purpose quality level, showing the count of Accepted and Rejected students within each category.
labs(title = "...", x = "...", y = "..."): Adds labels to the plot.

```
# Stacked bar plot for Statement of Purpose by Admission Status
ggplot(student_data, aes(x = Stetement_of_Purpose, fill = Status_of_Admission)) +
  geom_bar(position = "stack") +
  labs(title = "Statement of Purpose Quality by Admission Status", x = "Statement of Purpose Quality", y = "Count") +
  theme_minimal()
```

Output

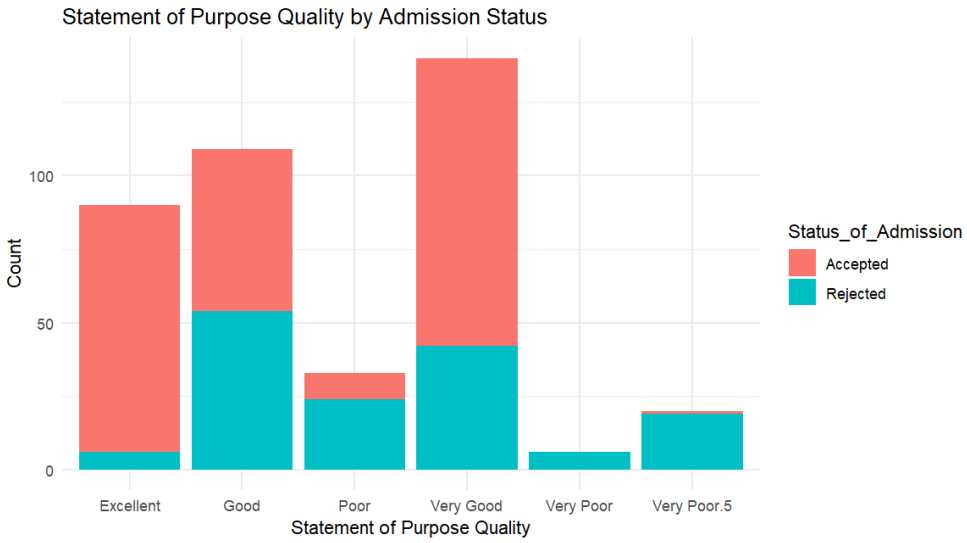

Figure 9, the stacked bar plot clearly shows a relationship between the quality of the Statement of Purpose and the likelihood of being accepted:

Figure 9: stack chart of SOP column

"Very Good" Statements are Key: The "Very Good" category has the tallest bar, with a very high proportion of those students being accepted. This strongly suggests that a strong Statement of Purpose is critical for a successful application.

"**Good" Statements are Still Positive:** The "Good" category also shows a higher proportion of accepted students compared to rejections, indicating that a well-written statement is still advantageous.

Weaker Statements Have Lower Acceptance: The "Excellent", "Poor", "Very Poor", and "Very

Poor" categories all show a greater proportion of rejections. This highlights that a weaker or poorly written Statement of Purpose can significantly hurt an applicant's chances.

The quality of the Statement of Purpose is a very important factor in the admission decision. A strong, well-articulated statement significantly increases the likelihood of acceptance, while a weaker statement can be a major disadvantage.

Conclusion

This analysis performed data validation, cleaning, outlier detection, and exploratory analysis for the given dataset. The following key steps were taken:
Duplicate records were removed.
Missing values were handled appropriately.
Categorical variables were encoded, and numerical variables were analyzed for outliers.
Statistical analysis and visualizations were generated to understand the distribution and spread of the data.
The analysis revealed a strong positive correlation between having research experience and the likelihood of being accepted. This finding highlights the value institutions place on prior research involvement.
Both the quality of Letters of Recommendation and the Statement of Purpose emerged as significant factors in admission decisions. Strong letters and well-articulated statements were consistently associated with higher acceptance rates. Conversely, weaker applications in these areas proved detrimental.