INTRODUCTION

Heart disease, also known as cardiovascular disease, is a major global health concern. It is the leading cause of death worldwide, affecting both men and women across different racial and ethnic groups. In the United States alone, someone dies from cardiovascular disease every 33 seconds, and 695,000 lives were lost to heart disease in 2021. The economic burden is significant, costing the United States billions of dollars each year (Centers for Disease Control and Prevention, n.d.). This issue is not limited to the United States, as cardiovascular disease remains the primary cause of death globally. In Malaysia, there has been a concerning increase in cardiovascular disease-related deaths and illnesses. It is essential to understand both modifiable risk factors like tobacco use, high blood pressure, high cholesterol, obesity, and diabetes, as well as unmodifiable factors such as gender, race, and family history, in order to effectively address this health challenge (Firus Khan et al., 2022). By taking preventive measures and addressing these factors, we can work towards reducing the impact of heart disease on individuals and communities.

This report aims to examine the impact of various factors on the development of heart disease, as provided in the CSV file. The factors of interest include age, sex, chest pain (CP) characterized by discomfort or tightness in the chest area, resting blood pressure (trestbps) which measures the force of blood against the artery walls when the individual is at rest, serum cholesterol (chol) levels, fasting blood sugar (fbs) levels, resting electrocardiographic results (restecg), maximum heart rate achieved (thalach) during exercise, exercise-induced angina (exang), ST depression induced by exercise relative to rest (oldpeak), the slope of the peak exercise ST segment (slope), the number of major blood vessels colored by fluoroscopy (ca), and the presence of thalassemia (thal), a genetic blood disorder.

In this study, we conducted an analysis of heart disease data using the R programming language. The primary objectives were to perform exploratory data analysis, assess the normality of the data, and transform the data into a normal distribution. Through exploratory data analysis, we examined various aspects of the dataset, including data types, missing values, and other relevant characteristics. Additionally, we focused on assessing the normality of one selected variable, following a step-by-step approach. For the remaining variables, a summary table was provided to give an overview of their distribution. Finally, we applied data transformation techniques to normalize the selected variable, demonstrating the successful achievement of normality. By employing these analytical techniques, we aimed to gain insights into the dataset, identify any data anomalies, and ensure the data meets the assumptions required for subsequent statistical analysis.

In conclusion, heart disease is a critical global health issue with far-reaching consequences. It is the leading cause of death worldwide, demanding immediate attention. The economic impact is substantial, highlighting the urgent need for comprehensive measures. Understanding the risk factors and implementing preventive strategies is crucial for reducing the burden of heart disease. This report utilizes the R programming language to analyze the impact of various factors on heart disease development. Through exploratory data analysis, normality assessment, and data transformation, valuable insights are gained to inform further statistical analysis. By deepening our understanding of these factors, we can enhance prevention, diagnosis, and management approaches for this widespread health concern.

EXPLORATORY DATA ANALYSIS (EDA)

To generate a summary of the variables in a data set, the function "summary()" is used. It provides descriptive statistics for each variable, including minimum and maximum values, quartiles, and means, as shown in Figure 1. The summary function is commonly employed to obtain a quick overview of the data and identify any potential issues or patterns. On the other hand, the "str()" function is used to display the structure of the data frame. It provides information about the variables in the data frame, their corresponding data types, and helps in understanding the composition of the dataset.

```
> summary(heartdf)
      age             sex               cp           trestbps         chol            fbs        
 Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000  
 1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000  
 Median :55.00   Median :1.0000   Median :1.000   Median :130.0   Median :240.0   Median :0.0000  
 Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6   Mean   :246.3   Mean   :0.1485  
 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:274.5   3rd Qu.:0.0000  
 Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0   Max.   :564.0   Max.   :1.0000  
    restecg          thalach         exang           oldpeak         slope             ca        
 Min.   :0.0000   Min.   : 71.0   Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:133.5   1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000  
 Median :1.0000   Median :153.0   Median :0.0000   Median :0.80   Median :1.000   Median :0.0000  
 Mean   :0.5281   Mean   :149.6   Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294  
 3rd Qu.:1.0000   3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000  
 Max.   :2.0000   Max.   :202.0   Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000  
      thal           target      
 Min.   :0.000   Min.   :0.0000  
 1st Qu.:2.000   1st Qu.:0.0000  
 Median :2.000   Median :1.0000  
 Mean   :2.314   Mean   :0.5446  
 3rd Qu.:3.000   3rd Qu.:1.0000  
 Max.   :3.000   Max.   :1.0000  
```

```
> str(heartdf)
'data.frame':   303 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
 $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
 $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
 $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

Figure 1 Summary of dataset on the left and structure of dataset on the right

After reviewing the summary and structure of the dataset, it became apparent that certain numerical values lacked clear explanations. To address this issue and enhance the interpretability of the dataset, we referred to the article by Deshmukh (2020), which utilized the same data source. By implementing Deshmukh's approach, we transformed the numerical values into more meaningful data, facilitating easier comprehension of the variables, as shown in Figure 2. The str() function is used to examine the modifications and observe the resulting changes in the dataset's structure.

```
'data.frame':   303 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : chr  "Male" "Male" "Female" "Male" ...
 $ cp      : Factor w/ 4 levels "ASY","ATA","NAP",..: 4 3 2 2 1 1 2 2 3 3 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : chr  "Yes" "No" "No" "No" ...
 $ restecg : Factor w/ 3 levels "LVH","Normal",..: 1 2 1 2 2 2 1 2 2 2 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : chr  "No" "No" "No" "No" ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : Factor w/ 3 levels "Downsloping",..: 1 1 3 3 3 2 2 3 3 3 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thal    : chr  "Fixed Defect" "Normal Blood Flow" "Normal Blood Flow" "Normal Blood Flow" ...
 $ target  : chr  "Yes" "Yes" "Yes" "Yes" ...
```

Figure 2 Structure of dataset after assigns meaningful labels to the variables

As depicted in Figure 3, the purpose of converting the thal variable from a factor to a character data type and replacing the character level "NA" with the NA missing value is to ensure consistency and proper handling of missing data in the dataset. In R, factors are utilized to represent categorical variables with predefined levels. However, if one of the levels is denoted by the character value "NA," it can lead to confusion because NA is the standard convention for representing missing values in R. Thus, this conversion is necessary to transform the factor level 'NA' into an actual missing value character.

```
heartdf$thal <- as.character(heartdf$thal)
is.na(heartdf$thal) <- heartdf$thal == "NA"
```

Figure 3 Converting factor levels character "NA" to real NA values.

To improve readability and comprehension, as illustrated in Figure 4, the variable names were modified to more accurately reflect the nature of the data, thereby reducing the potential for confusion. This practice of utilizing descriptive and meaningful names contributes to the creation of clean, understandable and robust code.

```
new_names <- c("Age","Sex","Chest Pain Type","Resting Blood Pressure","Cholesterol","Fasting Blood Sugar","Resting ECG",
               "Maximum Heart Rate","Exercise Induced Angina","Oldpeak","Slope","Number of Major Vessels",
               "Thalassemia","Heart Disease")

# Assign the new column names to the specified columns
colnames(heartdf) <- new_names
str(heartdf)
```

Figure 4 Changing the variable name

The function `head(f)` is used to display the first few rows of a data frame as shown in Figure 5. Each row represents an observation, and the columns represent variables or attributes of the data. This function provides a concise overview of the dataset, allowing to examine the variable names and the values in the initial rows. It helps to verify if the data is loaded correctly, check for any unexpected values, or get a general sense of the data structure.

```
> head(heartdf)
  Age    Sex Chest Pain Type Resting Blood Pressure Cholesterol Fasting Blood Sugar Resting ECG Maximum Heart Rate
1  63   Male             TA                   145         233                Yes         LVH                150
2  37   Male            NAP                   130         250                 No      Normal                187
3  41 Female            ATA                   130         204                 No         LVH                172
4  56   Male            ATA                   120         236                 No      Normal                178
5  57 Female            ASY                   120         354                 No      Normal                163
6  57   Male            ASY                   140         192                 No      Normal                148
  Exercise Induced Angina Oldpeak       Slope Number of Major Vessels       Thalassemia Heart Disease
1                      No     2.3 Downsloping                      0      Fixed Defect           Yes
2                      No     3.5 Downsloping                      0 Normal Blood Flow           Yes
3                      No     1.4   Upsloping                      0 Normal Blood Flow           Yes
4                      No     0.8   Upsloping                      0 Normal Blood Flow           Yes
5                     Yes     0.6   Upsloping                      0 Normal Blood Flow           Yes
6                      No     0.4        Flat                      0      Fixed Defect           Yes
> |
```

Figure 5 First 6 rows of the dataset

To obtain the results depicted in Figure 6, the sapply() function is employed to determine the data types of variables within the data frame. This aids in data exploration and analysis by returning a vector that contains the class or data type of each column in the data frame. It offers an overview of the data structure and can be used to verify whether the variables have been imported accurately or if any data type conversions are required for subsequent analysis.

```
sapply(heartdf, class)
                Age                 Sex     Chest Pain Type  Resting Blood Pressure         Cholesterol
          "integer"         "character"            "factor"               "integer"           "integer"
Fasting Blood Sugar         Resting ECG Maximum Heart Rate Exercise Induced Angina             Oldpeak
          "integer"           "factor"           "integer"             "character"           "numeric"
            Slope Number of Major Vessels        Thalassemia       Heart Disease
         "factor"             "integer"         "character"         "character"
|
```

Figure 6 Determined the data type and variable information

The lapply() function applies the unique() function to each column of the data frame as shown in Figure 7. The unique() function returns the unique values of a vector or column. Each element of the list will contain a vector of the unique values present in that particular column. This information can be useful for understanding the distinct values in each column, identifying potential data quality issues (such as unexpected or erroneous values), and gaining insights into the distribution or characteristics of the data.

```
unique_vals <- lapply(heartdf, unique)
unique_vals
```

Figure 7 Checking unique data in each variable

The z-score method is utilized to identify outliers within a dataset. It calculates the number of standard deviations a data point deviates from the mean. By setting a threshold, which is 2 in this project, data points with z-scores exceeding the threshold are classified as outliers. These outliers may indicate uncommon or extreme observations that deviate substantially from the majority of the data. The outcomes of the outlier calculation are displayed in Figure 8.

```
outliers
                Age  Resting Blood Pressure             Cholesterol
                 10                     15                      11
Maximum Heart Rate                Oldpeak Number of Major Vessels
                 11                     17                      25
|
```

Figure 8 Determine outliers in dataset

Figure 9 compares the boxplot before and after the removal of outliers to understand the impact of cleaning the dataset. The presence of outliers in the dataset can significantly influence the boxplot by stretching the whiskers and causing elongated or skewed boxplot shapes. These outliers can distort the interpretation of the median, quartiles, and the overall data spread. Following the removal of outliers, a new boxplot is generated using the cleaned dataset. This boxplot represents the distribution of the data with outliers eliminated, providing a more accurate representation of the central tendency and variability. The whiskers of the boxplot align more closely with the majority of the data, resulting in a visual summary that accurately depicts the location of the bulk of the data points and provides a better understanding of the dataset's spread and central tendency.

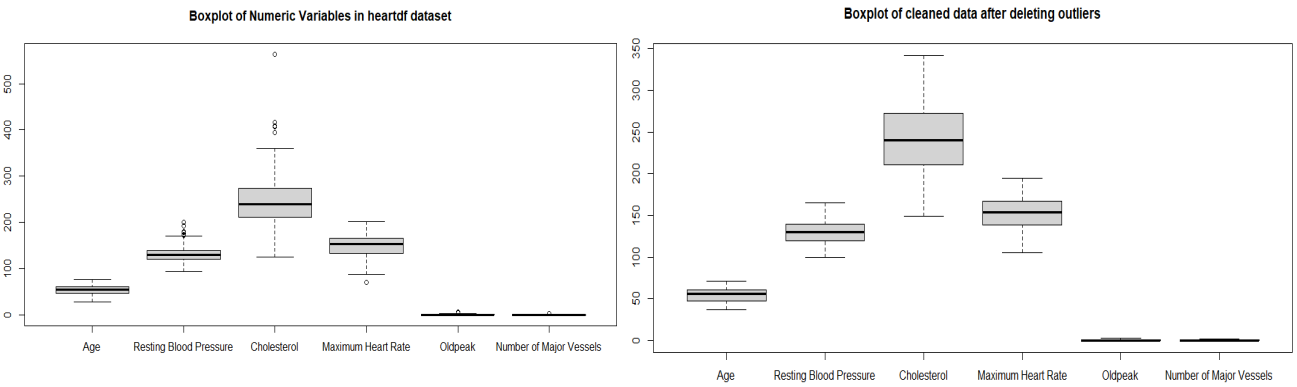

Figure 9 Comparison of boxplot for numeric variables in dataset with outliers on the top and cleaned one at the bottom

After removing the outliers, we proceed to check the missing value by using code shown in Figure 10. The result shows there are 91 missing values in the dataset, hence we need to remove the missing value using the 'complete.cases()' function. Then using the same code in Figure 10, the missing value now is 0.

```
# Check missing value
sum(is.null(heart_cleaned)| is.na(heart_cleaned))
```

```
sum(is.null(heart_cleaned)
] 91
```

Figure 10 Checking missing value

Figure 11 presents the code used to check for duplicate values in the dataset, and the result indicates that there are no duplicates, as the output shows a count of 0. Once the absence of duplicates is confirmed, we can proceed to examine the overall changes in the dataset using the summary() function. The output, depicted in Figure 12, provides a summary of the cleaned dataset. With the dataset now free of duplicates and potentially erroneous values, it is ready for further manipulation, analysis, and exploration.

```
# Check duplicated value
duplicates <- duplicated(heart)
sum(duplicates)
```

```
> sum(duplicates)
[1] 0
```

Figure 11 Checking duplicate value

```
> summary(heart)
      Age            Sex           Chest Pain Type Resting Blood Pressure  Cholesterol
 Min.   :37.00   Length:223         ASY:99          Min.   :100.0          Min.   :149.0
 1st Qu.:47.00   Class :character   ATA:42          1st Qu.:120.0          1st Qu.:212.0
 Median :54.00   Mode  :character   NAP:64          Median :130.0          Median :240.0
 Mean   :53.87                      TA :18          Mean   :128.8          Mean   :243.3
 3rd Qu.:60.00                                      3rd Qu.:140.0          3rd Qu.:269.0
 Max.   :71.00                                      Max.   :160.0          Max.   :342.0
 Fasting Blood Sugar          Resting ECG  Maximum Heart Rate Exercise Induced Angina    Oldpeak
 Length:223          LVH             :108  Min.   :105.0      Length:223              Min.   :0.0000
 Class :character    Normal          :114  1st Qu.:140.0      Class :character        1st Qu.:0.0000
 Mode  :character    ST-T abnormality:  1  Median :156.0      Mode  :character        Median :0.6000
                                           Mean   :152.6                              Mean   :0.8578
                                           3rd Qu.:168.0                              3rd Qu.:1.4500
                                           Max.   :194.0                              Max.   :3.2000
       Slope     Number of Major Vessels Thalassemia        Heart Disease
 Downsloping: 11   Min.   :0.0000         Length:223         Length:223
 Flat       : 97   1st Qu.:0.0000         Class :character   Class :character
 Upsloping  :115   Median :0.0000         Mode  :character   Mode  :character
                   Mean   :0.4843
                   3rd Qu.:1.0000
                   Max.   :2.0000
>
```

Figure 12 Summary of cleaned dataset

After performing outlier removal and handling missing values in the R dataset, several notable changes can be observed in the output of the cleaned dataset, as shown in Figure 12. The summary statistics of the cleaned dataset will provide a more accurate representation of the central tendency and variability of the data. Measures such as mean, median, and quartiles may have changed as a result of the removal of outliers, leading to potentially different values compared to the original dataset. Additionally, the range and spread of the data have also been altered. The cleaned dataset facilitates more robust analyses, reduces the potential for bias or misleading conclusions, and enhances the overall integrity of the results obtained from subsequent data manipulations and analyses.

After the dataset is cleaned, we do some data manipulation by counting the number of variables of categorical data. These code snippets from Figure 13 provide an overview of the counts for different categorical variables in the dataset, allowing for a quick analysis of the distribution and frequency of each category.

```
     Sex   n
1 Female  70
2   Male 153
```

```
  Fasting Blood Sugar   n
1                  No 196
2                 Yes  27
```

```
  Exercise Induced Angina   n
1                      No 159
2                     Yes  64
```

```
  Heart Disease   n
1            No  88
2           Yes 135
```

Figure 13 Data Manipulation: Displaying number of categorical data

NORMALITY

As shown in Figure 14, the Age variable exhibit a relatively symmetric and bell-shaped pattern hence it is safe to assume the data is normally distributed. This histogram depicts a variable that displays a bimodal distribution. A bimodal distribution is characterized by the presence of two distinct peaks or modes, indicating the existence of two different groups or subpopulations within the data. Each peak represents a concentration of values centered around specific ranges or categories. However, it is important to note that identifying departures from normality based solely on visual inspection is a preliminary assessment and should be supported by statistical tests. Additional statistical tests, such as the Shapiro-Wilk test, Jarque-Bera test, and Kolmogorov-Smirnov test, will provide further confirmation of the observed non-normality.

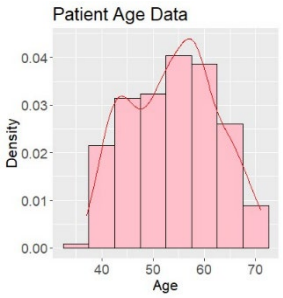

Figure 14 Age-Related Density Distribution of Heart Disease Plot

Skewness and kurtosis are statistical measures that provide information about the shape and distribution of a dataset. Skewness measures the asymmetry of the data, while kurtosis indicates the tailedness or concentration of the distribution. Positive skewness means the data is skewed to the right, negative skewness means it is skewed to the left, and a skewness value of 0 indicates a symmetrical distribution. Positive kurtosis (leptokurtic) indicates heavy tails and more extreme values, while negative kurtosis (platykurtic) indicates lighter tails and a more dispersed distribution. Figure 15 shows the results of skewness and kurtosis for the "age" variable. The skewness value of 0.01885544 indicates a right-skewed distribution. In terms of kurtosis, the value of 2.083659 suggests a moderately peaked distribution compared to a normal distribution. The dataset exhibits slightly heavier tails, a higher concentration of values around the mean, and a more peaked distribution compared to a normal distribution, indicating a potential presence of outliers or extreme values.

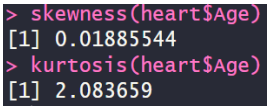

Figure 15 Skewness and Kurtosis Analysis of Heart Disease Data

The Jarque-Bera test is a statistical test used to assess the normality of a dataset. It examines whether a given dataset follows a normal distribution based on skewness and kurtosis. In the Jarque-Bera normality test in R, the null hypothesis states that the data follows a normal distribution, while the alternative hypothesis states that the data does not follow a normal distribution. If the p-value associated with the test is below a specified significance level, α (e.g., α = 0.05), the null hypothesis is rejected, indicating that the data is not normally distributed. In the Jarque-Bera Normality Test result displayed in Figure 16, the test statistic (JB) is calculated as 7.8152, and the associated p-value is 0.02009. Since the p-value (0.02009) is less than the significance level of 0.05, we reject the null hypothesis. This suggests that there is enough evidence to support the claim that the age data does not follow a normal distribution and is greater than a normal distribution. The alternative hypothesis, which is "greater," suggests that the distribution of the age data may have a heavier right tail compared to a normal distribution.

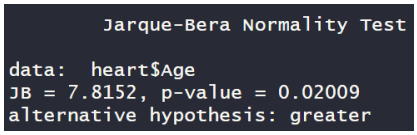

Figure 16 Summary Output of Jarque-Bera Normality Test

The asymptotic one-sample Kolmogorov-Smirnov test compares a sample to a specified distribution to assess if they are significantly different. It examines the maximum vertical difference between the sample's empirical cumulative distribution function (ECDF) and the specified distribution's cumulative distribution function (CDF). The test helps determine the goodness-of-fit between the data and the expected distribution. In Figure 17, "data: HA" represents the analyzed dataset. "D=1" is the test statistic, measuring the largest deviation between the observed data and the expected distribution. "alternative hypothesis: two-sided" indicates a test without a specific direction of difference, allowing for differences in both positive and negative directions. This type of test is useful when there is no prior expectation about the direction of the difference.

Another statistical test used to assess the normality of the dataset is Shapiro-Wilk Normality Test. It examines whether the data follows a normal distribution or significantly deviates from it. It compares the observed data to what is expected under the assumption of normality. The test calculates a statistic and associated p-value, which indicate whether there is evidence to reject the assumption of normality. Similar to the Jarque-Bera Normality Test, if the p-value is below a chosen significance level, α (e.g., α = 0.05), it suggests that the data significantly deviates from a normal distribution. Conversely, a higher p-value suggests that the data is likely to follow a normal distribution. As shown in Figure 17, the Shapiro-Wilk test was conducted on the dataset labelled "HA". The test statistic, denoted as "W," has a value of 0.97305. The associated p-value is calculated to be 0.0002922. Therefore, there is strong evidence to reject the assumption of normality. This indicates that the dataset significantly deviates from a normal distribution and may exhibit non-normal behaviour.

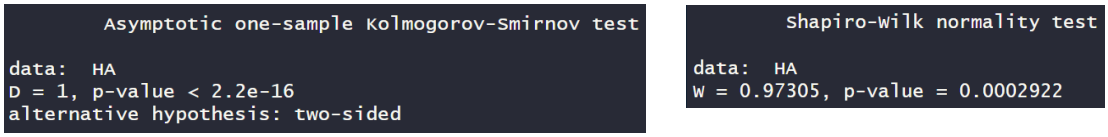

Figure 17 Summary Output of Asymptotic One-Sample Kolmogorov-Smirnov and Shapiro-Wilk Normality Tests

When analysing data in statistics, it is often important to assess whether a dataset follows a normal distribution. The normal distribution, also known as the Gaussian distribution or bell curve, is a common

assumption in many statistical methods. One way to visually examine the normality of a dataset is by using a Q-Q plot. The Q-Q plot, short for quantile-quantile plot, compares the quantiles of the observed data to the quantiles of a theoretical normal distribution. If the data points fall along a straight line in the plot, it suggests that the data is normally distributed. However, deviations from the line indicate departures from normality. By plotting a Q-Q plot for normality, we can assess whether our data meets the assumption of normality, which is crucial for performing accurate statistical analyses and making valid inferences. The Q-Q plot in Figure 18 reveals that the observed data deviates from the expected normal distribution. The points on the plot do not align with a straight line, indicating a departure from normality. This suggests that the dataset does not meet the assumption of normality, which should be taken into consideration when conducting statistical analyses. Adjustments or alternative methods may be necessary to accommodate the departure from normality and ensure the accuracy of subsequent statistical procedures.

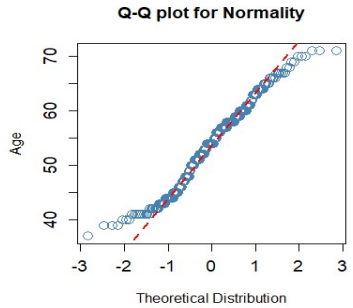

Figure 18 Q-Q Plot For Normality

As shown in Figure 19, the statistical tests conducted on the variables "Age," "Resting Blood Pressure," "Cholesterol," "Maximum Heart Rate," "Oldpeak," and "Number of Major Vessels" indicate significant departures from normality, as evidenced by p-values below the chosen significance level. These variables exhibit non-normal behaviour and caution should be exercised when interpreting results that assume normality. In contrast, the variables "Resting Blood Pressure," "Cholesterol," "Maximum Heart Rate," "Oldpeak," and "Number of Major Vessels" do not show significant departures from normality, allowing for the assumption of normality in further analyses, although the specific characteristics of their distributions should be considered.

	Variable	Test	P_Value	Reject_Null_Hypothesis	Normality
1	Age	Jarque-Bera	2.008821e-02	Yes	Not normal
2	Age	Kolmogorov-Smirnov	0.000000e+00	Yes	Not normal
3	Age	Shapiro-Wilk	2.921920e-04	Yes	Not normal
4	Resting Blood Pressure	Jarque-Bera	1.141446e-01	No	Normal
5	Resting Blood Pressure	Kolmogorov-Smirnov	0.000000e+00	Yes	Not normal
6	Resting Blood Pressure	Shapiro-Wilk	1.617098e-03	Yes	Not normal
7	Cholesterol	Jarque-Bera	9.627973e-02	No	Normal
8	Cholesterol	Kolmogorov-Smirnov	0.000000e+00	Yes	Not normal
9	Cholesterol	Shapiro-Wilk	2.621920e-02	Yes	Not normal
10	Maximum Heart Rate	Jarque-Bera	7.187688e-03	Yes	Not normal
11	Maximum Heart Rate	Kolmogorov-Smirnov	0.000000e+00	Yes	Not normal
12	Maximum Heart Rate	Shapiro-Wilk	1.558850e-04	Yes	Not normal
13	Oldpeak	Jarque-Bera	5.472748e-07	Yes	Not normal
14	Oldpeak	Kolmogorov-Smirnov	0.000000e+00	Yes	Not normal
15	Oldpeak	Shapiro-Wilk	2.653306e-14	Yes	Not normal
16	Number of Major Vessels	Jarque-Bera	7.307932e-11	Yes	Not normal
17	Number of Major Vessels	Kolmogorov-Smirnov	0.000000e+00	Yes	Not normal
18	Number of Major Vessels	Shapiro-Wilk	1.455310e-20	Yes	Not normal

Figure 19 Summary Output of All Numerical Data

Figure 20 combines all q-q plots for the numerical variables in this dataset. By examining the plotted graphs, it becomes apparent that for all the variables involved, it is safe to assume that the data is non-normally distributed. This is evident from the significant deviations of the points from the straight diagonal line. While observing substantial deviations from the straight diagonal line in the q-q plot serves as a strong indication of non-normality, we conduct additional tests to further confirm the non-normal distribution of the data.

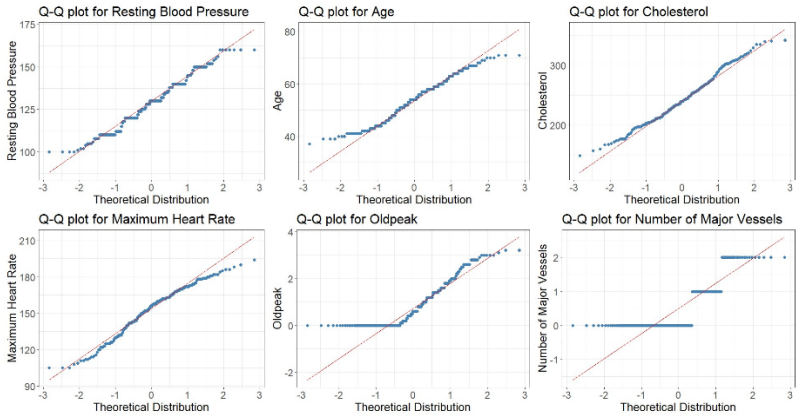

Figure 20 Combination of Q-Q Plot for trestbps, age, chol, thalach, oldpeak and ca

Figure 21 presents a combination of histogram and density graphs for all the numerical data involved in this dataset. Upon analyzing the plotted graph, we observe that Resting blood pressure, Age, Cholesterol, and Maximum Heart Rate display a relatively symmetric and bell-shaped pattern. These variables exhibit characteristics that are indicative of a normal distribution. The histogram and density graphs for these variables align with the familiar bell curve, suggesting that the data points are distributed around a central tendency, with the majority of values concentrated in the middle and tapering off towards the tails. However, for Oldpeak and Number of Major Vessels, the histogram and density graphs do not conform to a normal distribution. The shapes of these graphs indicate that the data points for Oldpeak and Number of Major Vessels are not symmetrically distributed around a central tendency.

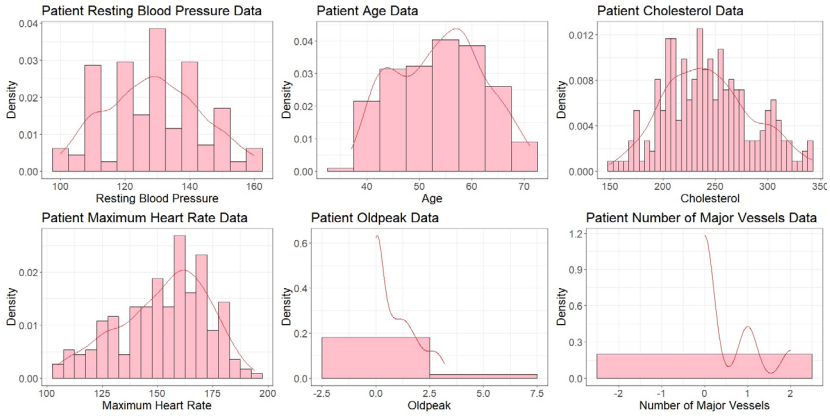

Figure 21 Combination of Density Distribution Plot for trestbps, age, chol, thalach, oldpeak and ca

TRANSFORMATION

We have found that none of the continuous variables in the dataset follow a normal distribution based on the analysis we conducted on those variables. Jarque-Bera, Kolmogorov-Smirnov, and Shapiro-Wilk tests, among others, have all shown that the null hypothesis of normality cannot be accepted.

In this instance, transformation is required to resolve the data's deviation from normality. A more symmetrical and regularly distributed data distribution, which is frequently desired in statistical studies and modelling, can be attained by transforming the variables.The value of transformation rests in its capacity to increase the reliability of normality-assumed statistical tests and models. For accurate results and trustworthy inference, many statistical techniques, including regression analysis, ANOVA, and hypothesis testing, rely on the assumption of normality. The outcomes may be skewed or deceptive when the data contradicts this presumption.

We may be able to attain a closer approximation to normalcy by applying the proper transformations to the variables, which could improve the validity and reliability of later studies. Statistical inferences and

model predictions can be made with more accuracy using data normalisation techniques such as logarithmic, square root, reciprocal, and exp.

Four distinct techniques were used to modify the data's cholesterol variable: reciprocal, exponential, square root, and logarithmic. To ascertain whether the converted variables showed normality or not, three normality tests were then carried out for each transformation technique.

From the summarising output as shown in Figure 22, we can see that, according to the Shapiro-Wilk and Jarque-Bera tests, the variables produced by the logarithmic and square root transformations are both most likely to have a normal distribution at 5% significance level. The Kolmogorov-Smirnov test reveals that they still depart significantly from a normal distribution at 5% significance level. Based on any of the three tests, neither the exponential nor reciprocal transformations produce variables that are likely to have a normal distribution at 5% significance level.

As a summary, the logarithmic and square root transformations seem more appropriate to make the cholesterol variable more normally distributed, according to the tests that were done. A variable need not, however, adhere to all transformation techniques in order to be regarded as normal. The choice is based on the requirements for the data and the analysis, and each transformation addresses particular data properties. The failure of a transformation to reach normalcy does not render an analysis erroneous, but it may point to underlying data features. Statistical tests, visual evaluation, and domain expertise should all be taken into account when choosing which transformation to apply.

Transformation	Shapiro_Wilk_PValue	Shapiro_Wilk_Reject	Shapiro_Wilk_Normal	Jarque_Bera_PValue
Logarithmic	2.018243e-01	No	Yes	0.351104252
Square Root	1.897007e-01	No	Yes	0.293274225
Exponential	1.420182e-31	Yes	No	NaN
Reciprocal	5.913011e-04	Yes	No	0.003227151

Jarque_Bera_Reject	Jarque_Bera_Normal	KS_Test_PValue	KS_Test_Reject	KS_Test_Normal
No	Yes	0	Yes	No
No	Yes	0	Yes	No
<NA>	<NA>	0	Yes	No
Yes	No	0	Yes	No

Figure 22 Summary Output of Normality Test for Transformed Cholesterol Variable

As shown in Figure 23, we used graphical visualisation approaches, such as histograms with density lines, QQ plots, and boxplots for both the log and sqrt transformations, to further confirm the normality of the transformed cholesterol variable. These visualisations show how, after applying these changes, the cholesterol variable moves noticeably closer to a normal distribution.

But it's important to understand that getting close to normality might be difficult for a variety of reasons. It's possible that the original data has inherent traits or patterns that can't be fully changed by transformations alone. Furthermore, normalcy itself is an idealised assumption that might not always hold true in data from the real world. Many real-world phenomena deviate from strict normality as a result of intricate relationships and underpinning mechanisms.

It is vital to recognise that while transformations like log and sqrt might significantly enhance the distributional features of data and approach normalcy, they may not produce a completely normal distribution. As an alternative, these transformations offer a helpful approximation and help to satisfy the underlying assumptions of statistical tests and models. As a result, despite the fact that log and sqrt transformations can improve data normalcy, reaching absolute normality is frequently difficult because of the peculiarities of the data and the restrictions of the transformations.

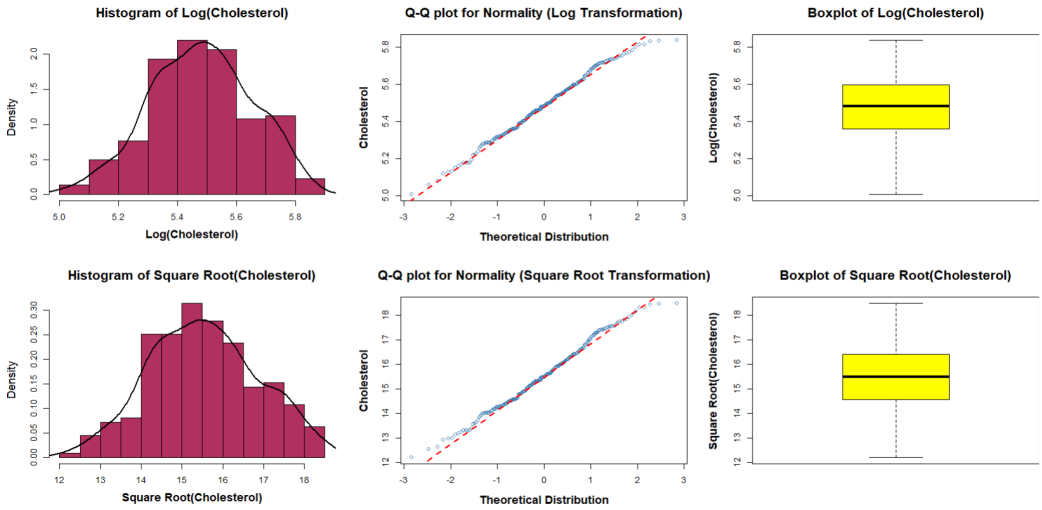

Figure 23 Histogram, Q-Q Plot & Boxplot for Log and Sqrt Cholesterol Variable

CONCLUSION

In conclusion, this report conducted exploratory data analysis, assessed normality, and performed data transformation for the prediction of heart disease development. The assessment of normality revealed that none of the continuous variables followed a normal distribution. Statistical tests such as the Jarque-Bera, Kolmogorov-Smirnov, and Shapiro-Wilk tests confirmed the departure from normality, indicating the need for data transformation. The non-normal behaviour of the variables raised concerns regarding the assumptions underlying subsequent statistical analyses. To address the non-normality issue, data transformation techniques were employed. These transformations aimed to achieve a more symmetric and regularly distributed data distribution, aligning with the assumption of normality. Transformations such as logarithmic, square root, reciprocal, and other appropriate techniques were applied to the variables. These transformations aimed to improve the validity and reliability of subsequent statistical tests and models, ensuring accurate results and reliable inference.

By performing exploratory data analysis, assessing normality, and implementing data transformations, this report enhanced the understanding and interpretation of the dataset for heart disease development. The identification of outliers, non-normal distributions, and subsequent transformation techniques contributed to the creation of a cleaner and more robust dataset for further analysis. However, it is important to note that while data transformations can improve the approximation to normality, they should be applied cautiously and their impact on the research question and subsequent analyses should be carefully considered. Additionally, further validation and evaluation of the transformed variables should be conducted to ensure their suitability for the intended statistical analyses and modelling. Overall, the exploratory data analysis, normality assessment, and data transformation performed in this report have provided valuable insights into the dataset and laid the foundation for subsequent analyses. These steps have contributed to improving the reliability and accuracy of statistical inferences and predictions related to heart disease development.

REFERENCES

Centers for Disease Control and Prevention. (n.d.). Multiple cause of death data on CDC Wonder. Centers for Disease Control and Prevention. https://wonder.cdc.gov/mcd.html

Deshmukh, H. (2020, 6). *Heart Disease UCI-Diagnosis & Prediction | by Hardik Deshmukh*. Towards Data Science. Retrieved June 19, 2023, from https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7

Firus Khan, A. Y., Ramli, A. S., Abdul Razak, S., Mohd Kasim, N. A., Chua, Y. A., Ul-Saufie, A. Z., . . . Nawawi, H. (2022). The Malaysian HEalth and WellBeing AssessmenT (MyHEBAT) Study Protocol: An Initiation of a National Registry for Extended Cardiovascular Risk Evaluation in the Community. *Int J Environ Res Public Health, 19*(18). doi:10.3390/ijerph191811789

APPENDIX A: R-Coding

```
#  Install required library

library(dplyr)     # for data manipulation


#  Read the dataset

heartdf<-read.csv("heart.csv")


#  Explore dataset structure and summary

str(heartdf)

summary(heartdf)


# Changing 0 to "No" and 1 to "Yes" and changing the numbers to meaningful data for specified columns based on the data explaination sources

heartdf$sex<-ifelse(heartdf$sex == 0,"Female","Male")

heartdf$fbs<-ifelse(heartdf$fbs == 0,"No", "Yes")

heartdf$exang<-ifelse(heartdf$exang == 0,"No","Yes")

heartdf$target<-ifelse(heartdf$target == 0,"No","Yes")

heartdf$cp <- factor(heartdf$cp, levels = c(0, 1, 2, 3), labels = c("ASY", "ATA", "NAP", "TA"))

heartdf$restecg<- factor(heartdf$restecg, levels = c(0,1,2), labels = c("LVH", "Normal", "ST-T abnormality"))

heartdf$slope<-factor(heartdf$slope, levels = c(0,1,2), labels = c("Downsloping", "Flat", "Upsloping"))

heartdf$thal <- factor(heartdf$thal,levels = c(0, 1, 2, 3),
                       labels = c(NA,"Fixed Defect","Normal Blood Flow","Reversible Defect"))


# Converting factor levels character "NA" to real NA values.

# Only then NA will be counted as missing value.

heartdf$thal <- as.character(heartdf$thal)

is.na(heartdf$thal) <- heartdf$thal == "NA"



summary(heartdf)

str(heartdf)
```

```
# Change column Names
new_names <- c("Age","Sex","Chest Pain Type","Resting Blood Pressure","Cholesterol","Fasting Blood Sugar","Resting ECG",
               "Maximum Heart Rate","Exercise Induced Angina","Oldpeak","Slope","Number of Major Vessels",
               "Thalassemia","Heart Disease")

# Assign the new column names to the specified columns
colnames(heartdf) <- new_names
str(heartdf)

# Check the first 6 rows of the dataset
head(heartdf)

# Examine data types and variable information:
sapply(heartdf, class)

# Identify unique values for categorical variables:
unique_vals <- lapply(heartdf, unique)
unique_vals

### Checking outliers ###
# Create box plots for numeric variables
numeric_vars <- sapply(heartdf, is.numeric)
boxplot(heartdf[, numeric_vars])

# Identify outliers in numeric variables
threshold <- 2
outliers <- sapply(heartdf[, numeric_vars], function(x) {
  mean_value <- mean(x, na.rm = TRUE)
  sd_value <- sd(x, na.rm = TRUE)
  num_outliers <- sum(x < mean_value - threshold * sd_value | x > mean_value + threshold * sd_value, na.rm = TRUE)
```

```
  num_outliers
})
outliers

# Clean the outliers
heart_cleaned <- heartdf
for (var in names(heartdf)[numeric_vars]) {
  mean_value <- mean(heartdf[[var]], na.rm = TRUE)
  sd_value <- sd(heartdf[[var]], na.rm = TRUE)
  heart_cleaned[[var]][heart_cleaned[[var]] < mean_value - threshold * sd_value | heart_cleaned[[var]]
> mean_value + threshold * sd_value] <- NA
}
# Check boxplot again
# Can Compare the boxplot before and after cleaned. Extend the image panel to display all x-axis
name
boxplot(heart_cleaned[, numeric_vars])

# --------Delete missing & duplicated values -----------

# Check missing value
sum(is.null(heart_cleaned)| is.na(heart_cleaned))  # Check missing value in a data frame or column

# Remove rows with missing values
heart <- heart_cleaned[complete.cases(heart_cleaned), ]

# Check the dimensions of the cleaned dataset
dim(heart)
print(sum(is.na(heart)))

# Check duplicated value
duplicates <- duplicated(heart)
sum(duplicates)
```

```
#  Remove duplicated rows from the dataset in-place
# heart<- heart[!duplicated(heart), ] # No need because there is no duplicate data

# Check duplicated value
print(sum(duplicated(heart)))
dim(heart)

summary(heart)

####  Data manipulation ####
# 1.Display number of Female and Males (categorical data)
sex_counts <- heart %>% count(Sex)
print(sex_counts)

# 2. Display number of Fasting Blood Sugar count (categorical data)
fbs_counts<- heart%>% count(`Fasting Blood Sugar`)
print(fbs_counts)

# 3. Display number of Fasting Blood Sugar count (categorical data)
exang_counts<- heart%>% count(`Exercise Induced Angina`)
print(exang_counts)

# 4. Display number of Fasting Blood Sugar count (categorical data)
target_counts<- heart%>% count(`Heart Disease`)
print(target_counts)
```

```
#Normality data analysis
library(ggplot2)

ggplot(heart, aes(x = Age)) +
  geom_histogram(aes(y = ..density..), binwidth = 5, fill = "pink", color = "black") +
  geom_density(color = "red") +
  labs(title = "Patient Age Data", x = "Age", y = "Density") +
  theme(text = element_text(size = 16),
        axis.text = element_text(size = 14))

summary(heart$Age)
#for one variable, in this example we choose Age
#check skewness & kurtosis
library(moments)
skewness(heart$Age)
kurtosis(heart$Age)

#Null Hypothesis: The dataset has a skewness and kurtosis that matches a normal distribution.
#Alternative Hypothesis: The dataset has a skewness and kurtosis that does not match a normal distribution.
#use Jarque.test in the package moments

jarque.test(heart$Age)

#p-value = 0.06 greater than alpha= 0.05 hence accept H0

#Another approach to check the normality using Q-Q plot
HA<- heart$Age
qqnorm(HA, main = Q -Q plot for Normality, xlab = "Theoretical Distribution", ylab = Age, col = "steelblue", cex = 1.5, cex.axis = 1.2)
qqline(HA, col = red, lwd = 2, lty = 2)
#data falls approximately along a straight line, hence the data can be reasonably approximated by a normal distribution
```

```
#using Kolmogorov-Smirnov (KS)

#Null Hypothesis (H0): The sample dataset follows the specified theoretical distribution.

#Alternative Hypothesis (HA): The sample dataset does not follow the specified theoretical distribution.


ks.test(HA,"pnorm")

#p-value less than 2.2e-16 smaller than alpha= 0.05, reject the null hypothesis


#Shapiro-Wilk test

shapiro.test(HA)

#p-value of 0.006745 (< 0.05), reject null hypothesis

#for all dataset

numeric_vars <- c("Age", "Resting Blood Pressure", "Cholesterol", "Maximum Heart Rate", "Oldpeak", "Number of Major Vessels")


# Calculate skewness

skew_values <- apply(heart[numeric_vars], 2, skewness)


# Calculate kurtosis

kurt_values <- apply(heart[numeric_vars], 2, kurtosis)


# Combine the results into a data frame

skew_kurt_df <- data.frame(Skewness = skew_values, Kurtosis = kurt_values)


# Print the results

print(skew_kurt_df)


#------------------------------------------------------------------------


# Apply Jarque-Bera test to each variable

jb_test_results <- lapply(heart[numeric_vars], function(x) jarque.test(x))
```

```
# Extract test statistics and p-values
test_statistics <- sapply(jb_test_results, function(x) x$statistic)
p_values <- sapply(jb_test_results, function(x) x$p.value)

# Combine the results into a data frame
jb_results_df <- data.frame(Test_Statistic = test_statistics, P_Value = p_values, row.names =
names(heart[numeric_vars]))

# Print the results
print(jb_results_df)
#------------------------------------------------------------------------
library(gridExtra)
library(ggplot2)
library(stringr)

# Create a list to store the Q-Q plots
numeric_plots <- list()

# Loop over each variable in numeric_vars
for (variable in numeric_vars) {
  # Create the Q-Q plot for the current variable
  qqplot <- ggplot(heart, aes(sample = .data[[variable]])) +
    geom_qq(col = "steelblue") +
    geom_qq_line(col = "red") +
    ggtitle(paste0("Q-Q plot for ", variable)) +
    xlab("Theoretical Distribution") +
    ylab(variable) +
    theme_bw() +
    theme(text = element_text(size = 16),
          axis.text = element_text(size = 14),
          plot.title = element_text(hjust = 0.5),  # Center-align the plot title
          axis.title.y = element_text(hjust = 1))  # Right-align the y-axis label
```

```
  # Add the plot to the list
  numeric_plots[[variable]] <- qqplot
}

#combine all to one figure
grid.arrange(
  numeric_plots[["Resting Blood Pressure"]],
  numeric_plots[["Age"]],
  numeric_plots[["Cholesterol"]],
  numeric_plots[["Maximum Heart Rate"]],
  numeric_plots[["Oldpeak"]],
  numeric_plots[["Number of Major Vessels"]],
  nrow = 2, ncol = 3
)

#-------------------------------------------------------------------------------
RP <- heart$`Resting Blood Pressure`
c<- heart$Cholesterol
M<- heart$`Maximum Heart Rate`
o<- heart$Oldpeak
ca<- heart$`Number of Major Vessels`

# Apply Kolmogorov-Smirnov test to each variable
ks.test(RP,"pnorm")
ks.test(c,"pnorm")
ks.test(M,"pnorm")
ks.test(o,"pnorm")
ks.test(ca,"pnorm")

# Apply Shapiro-Wilk test to each variable
shapiro.test(RP)
shapiro.test(c)
```

```
shapiro.test(M)

shapiro.test(o)

shapiro.test(ca)

#summary of all results


library(moments)


numeric_vars <- c("Age", "Resting Blood Pressure", "Cholesterol", "Maximum Heart Rate",
"Oldpeak", "Number of Major Vessels")


results <- data.frame(

  Variable = character(),

  Test = character(),

  P_Value = numeric(),

  Reject_Null_Hypothesis = character(),

  Normality = character(),

  stringsAsFactors = FALSE

)


for (variable in numeric_vars) {

  data <- heart[[variable]]


  # Calculate skewness and kurtosis

  skew <- skewness(data)

  kurt <- kurtosis(data)


  # Perform Jarque-Bera test

  jarque_test <- jarque.test(data)

  jarque_pvalue <- jarque_test$p.value

  jarque_reject <- ifelse(jarque_pvalue < 0.05, "Yes", "No")

  jarque_normal <- ifelse(jarque_reject == "Yes", "Not normal", "Normal")
```

```
  # Perform Kolmogorov-Smirnov test
  ks_test <- ks.test(data, "pnorm")
  ks_pvalue <- ks_test$p.value
  ks_reject <- ifelse(ks_pvalue < 0.05, "Yes", "No")
  ks_normal <- ifelse(ks_reject == "Yes", "Not normal", "Normal")

  # Perform Shapiro-Wilk test
  shapiro_test <- shapiro.test(data)
  shapiro_pvalue <- shapiro_test$p.value
  shapiro_reject <- ifelse(shapiro_pvalue < 0.05, "Yes", "No")
  shapiro_normal <- ifelse(shapiro_reject == "Yes", "Not normal", "Normal")

  # Add the results to the table
  result <- data.frame(
    Variable = variable,
    Test = c("Jarque-Bera", "Kolmogorov-Smirnov", "Shapiro-Wilk"),
    P_Value = c(jarque_pvalue, ks_pvalue, shapiro_pvalue),
    Reject_Null_Hypothesis = c(jarque_reject, ks_reject, shapiro_reject),
    Normality = c(jarque_normal, ks_normal, shapiro_normal),
    stringsAsFactors = FALSE
  )

  results <- rbind(results, result)
}

# Print the table
print(results)
```

```
#hist + density graph for all variables

# Create an empty list to store the plots
numeric_plots <- list()

# Create an empty list to store the plots
numeric_plots <- list()

# Loop through each variable in numeric_vars
for (variable in numeric_vars) {
  # Create the histogram and density plot for the current variable
  hist_density_plot <- ggplot(heart, aes(x = .data[[variable]])) +
    geom_histogram(aes(y = ..density..), binwidth = 5, fill = "pink", color = "black") +
    geom_density(color = "red") +
    labs(title = paste0("Patient ", variable, " Data"), x = variable, y = "Density") +
    theme_bw() +
    theme(axis.text = element_text(size = 14))
  # Add the plot to the list
  numeric_plots[[variable]] <- hist_density_plot
}

#combine all to one figure
grid.arrange(
  numeric_plots[["Resting Blood Pressure"]],
  numeric_plots[["Age"]],
  numeric_plots[["Cholesterol"]],
  numeric_plots[["Maximum Heart Rate"]],
  numeric_plots[["Oldpeak"]],
  numeric_plots[["Number of Major Vessels"]],
  nrow = 2, ncol = 3
)
```

```
#####################################################################

#TRANSFORMATION FOR CHOLESTEROL

library(moments)

# Logarithmic transformation
transformed_log <- log(heart$Cholesterol)
shapiro_test_log <- shapiro.test(transformed_log)
jarque_test_log <- jarque.test(transformed_log)
ks_test_log <- ks.test(transformed_log, "pnorm")

# Square root transformation
transformed_sqrt <- sqrt(heart$Cholesterol)
shapiro_test_sqrt <- shapiro.test(transformed_sqrt)
jarque_test_sqrt <- jarque.test(transformed_sqrt)
ks_test_sqrt <- ks.test(transformed_sqrt, "pnorm")

# Exponential transformation
transformed_exp <- exp(heart$Cholesterol)
shapiro_test_exp <- shapiro.test(transformed_exp)
jarque_test_exp <- jarque.test(transformed_exp)
ks_test_exp <- ks.test(transformed_exp, "pnorm")

# Reciprocal transformation
transformed_reciprocal <- 1/heart$Cholesterol
shapiro_test_reciprocal <- shapiro.test(transformed_reciprocal)
jarque_test_reciprocal <- jarque.test(transformed_reciprocal)
ks_test_reciprocal <- ks.test(transformed_reciprocal, "pnorm")

# Create a table summarizing the results
```

```
results <- data.frame(
  Transformation = c("Logarithmic", "Square Root", "Exponential", "Reciprocal"),
  Shapiro_Wilk_PValue = c(shapiro_test_log$p.value, shapiro_test_sqrt$p.value,
shapiro_test_exp$p.value, shapiro_test_reciprocal$p.value),
  Shapiro_Wilk_Reject = c(ifelse(shapiro_test_log$p.value < 0.05, "Yes", "No"),
                          ifelse(shapiro_test_sqrt$p.value < 0.05, "Yes", "No"),
                          ifelse(shapiro_test_exp$p.value < 0.05, "Yes", "No"),
                          ifelse(shapiro_test_reciprocal$p.value < 0.05, "Yes", "No")),
  Shapiro_Wilk_Normal = c(ifelse(shapiro_test_log$p.value >= 0.05, "Yes", "No"),
                          ifelse(shapiro_test_sqrt$p.value >= 0.05, "Yes", "No"),
                          ifelse(shapiro_test_exp$p.value >= 0.05, "Yes", "No"),
                          ifelse(shapiro_test_reciprocal$p.value >= 0.05, "Yes", "No")),
  Jarque_Bera_PValue = c(jarque_test_log$p.value, jarque_test_sqrt$p.value,
jarque_test_exp$p.value, jarque_test_reciprocal$p.value),
  Jarque_Bera_Reject = c(ifelse(jarque_test_log$p.value < 0.05, "Yes", "No"),
                         ifelse(jarque_test_sqrt$p.value < 0.05, "Yes", "No"),
                         ifelse(jarque_test_exp$p.value < 0.05, "Yes", "No"),
                         ifelse(jarque_test_reciprocal$p.value < 0.05, "Yes", "No")),
  Jarque_Bera_Normal = c(ifelse(jarque_test_log$p.value >= 0.05, "Yes", "No"),
                         ifelse(jarque_test_sqrt$p.value >= 0.05, "Yes", "No"),
                         ifelse(jarque_test_exp$p.value >= 0.05, "Yes", "No"),
                         ifelse(jarque_test_reciprocal$p.value >= 0.05, "Yes", "No")),
  KS_Test_PValue = c(ks_test_log$p.value, ks_test_sqrt$p.value, ks_test_exp$p.value,
ks_test_reciprocal$p.value),
  KS_Test_Reject = c(ifelse(ks_test_log$p.value < 0.05, "Yes", "No"),
                     ifelse(ks_test_sqrt$p.value < 0.05, "Yes", "No"),
                     ifelse(ks_test_exp$p.value < 0.05, "Yes", "No"),
                     ifelse(ks_test_reciprocal$p.value < 0.05, "Yes", "No")),
  KS_Test_Normal = c(ifelse(ks_test_log$p.value >= 0.05, "Yes", "No"),
                     ifelse(ks_test_sqrt$p.value >= 0.05, "Yes", "No"),
                     ifelse(ks_test_exp$p.value >= 0.05, "Yes", "No"),
                     ifelse(ks_test_reciprocal$p.value >= 0.05, "Yes", "No"))
)
```

```
# Print the table

print(results)


# log & sqrt cholesterol


par(mfrow = c(2, 3))


# Logarithmic transformation

hist(transformed_log, freq = FALSE, main = bquote(bold("Histogram of Log(Cholesterol)")), xlab =
bquote(bold("Log(Cholesterol)")), col = "maroon", cex.main = 1.5, cex.lab = 1.3)

lines(density(transformed_log), col = "black", lwd = 1.6)

qqnorm(transformed_log, main = bquote(bold("Q-Q plot for Normality (Log Transformation)")), xlab
= bquote(bold("Theoretical Distribution")), ylab = bquote(bold("Cholesterol")), col = "steelblue",
cex.main = 1.5, cex.lab = 1.3)

qqline(transformed_log, col = red, lwd = 2, lty = 2)

boxplot(transformed_log, main = bquote(bold("Boxplot of Log(Cholesterol)")), ylab =
bquote(bold("Log(Cholesterol)")), col = "yellow", cex.main = 1.5, cex.lab = 1.3)


# Square Root transformation

hist(transformed_sqrt, freq = FALSE, main = bquote(bold("Histogram of Square
Root(Cholesterol)")), xlab = bquote(bold("Square Root(Cholesterol)")), col = "maroon", cex.main =
1.5, cex.lab = 1.3)

lines(density(transformed_sqrt), col = "black", lwd = 1.6)

qqnorm(transformed_sqrt, main = bquote(bold("Q-Q plot for Normality (Square Root
Transformation)")), xlab = bquote(bold("Theoretical Distribution")), ylab =
bquote(bold("Cholesterol")), col = "steelblue", cex.main = 1.5, cex.lab = 1.3)

qqline(transformed_sqrt, col = red, lwd = 2, lty = 2)

boxplot(transformed_sqrt, main = bquote(bold("Boxplot of Square Root(Cholesterol)")), ylab =
bquote(bold("Square Root(Cholesterol)")), col = "yellow", cex.main = 1.5, cex.lab = 1.3)
```