Predicting Heart Disease

Olachi Mbakwe, Justin Constant, Rebande Olusesi and Evan Settipane

2023-12-15

Introduction

Heart disease stands as one of the leading causes of mortality globally, accounting for an estimated 17.9 million deaths each year, which is 31% of all global deaths. This statistic highlights the impact of heart disease not only in terms of prevalence but also in its deep implications for individuals, communities, and healthcare systems worldwide. The high incidence, coupled with the potential to significantly impair quality of life, positions heart disease as a critical area for study, intervention, and innovation in public health. The consequences of heart disease are far-reaching, affecting people of all ages and backgrounds. It predominantly leads to heart attacks and strokes, with a considerable proportion of these deaths occurring prematurely in individuals under 70. The burden it places on healthcare systems, families, and economies is substantial, often leading to prolonged illness, disability, and decreased life expectancy. These ripple effects extend beyond the individual, impacting communities and societies at large.

Understanding and addressing heart disease goes beyond reducing mortality rates; it is about enhancing the quality of life and improving health outcomes for millions worldwide. This becomes increasingly critical given the rising prevalence of risk factors associated with heart disease, such as hypertension, obesity, and diabetes. This project is motivated by a broader concern for public health and the well-being of society. It is driven by the goal of applying machine learning techniques to gain a deeper understanding of heart disease, aiming to identify key indicators and predictors using analytical methods. We intend to employ various machine learning algorithms, each chosen for its ability to reveal different aspects of the data. Through regression-based, tree-based, and unsupervised learning approaches, we aspire to uncover patterns and correlations that might otherwise remain obscured. The insights gained from this study are anticipated to contribute significantly to developing more effective strategies for the prevention, diagnosis, and management of heart disease, ultimately leading to better health outcomes and a decrease in its global impact.

These models have the potential to aid in early detection and informed decision-making in clinical settings, potentially reducing the incidence and severity of heart disease. Our interest in this project stems from this immense potential to make a meaningful difference in the field of public health and the lives of those affected by heart disease.

Engaging with Heart Data set

Population and Data Collection

The data set chosen for this study is the Heart Failure Prediction Data set, an extensive collection of data specifically curated for research on cardiovascular diseases (CVDs). This data set is a culmination of data from five different heart disease studies:

The data set selected for this study is the Heart Failure Prediction Data set, an extensive collection of data curated specifically for research on cardiovascular diseases (CVDs). This data set is culmination of data from five different heart disease studies:

- Cleveland Clinic Foundation (303 observations)
- Hungarian Institute of Cardiology, Budapest (294 observations)
- University Hospital, Zurich, Switzerland (123 observations)
- V.A. Medical Center, Long Beach, California (200 observations)
- Stalog Heart Disease Data set (270 observations)

After the elimination of duplicates, the final data set comprises 918 observations. This diverse collection spans a range of demographics, health backgrounds, and geographical locations, offering a holistic view of the various factors contributing to heart disease.

The data sets were assembled by respected institutions, including the Hungarian Institute of Cardiology in Budapest (Andras Janosi, M.D.), the University Hospital in Zurich and Basel, Switzerland (William Steinbrunn, M.D., and Matthias Pfisterer, M.D.), and the V.A. Medical Center in Long Beach, along with the Cleveland Clinic Foundation (Robert Detrano, M.D., Ph.D.).

The primary objective of compiling this data set was to create a comprehensive resource for research in heart disease prediction. By integrating data sets from varied sources, it provides a more robust and diverse basis for analysis. The data collection involved clinical and diagnostic measurements crucial for understanding cardiovascular health, including demographic information, blood pressure readings, cholesterol levels, and results from electrocardiograms.

Data Selection:

- `Demographic Variables`: The data set includes critical demographic information such as age and gender, providing insights into how heart disease prevalence and risks vary across different populations.
- `Clinical Measurements`: Key health indicators like blood pressure, cholesterol levels, and maximum heart rate are included. These measurements are essential in understanding the physiological factors that may contribute to heart disease.
- `Health History and Lifestyle Factors`: Variables such as fasting blood sugar and exercise-induced angina offer a glimpse into lifestyle and health history aspects that could influence heart disease risks.
- `Electrocardiogram Results`: The inclusion of resting ECG results adds another layer of diagnostic data, aiding in the comprehensive assessment of heart health.
- `Symptom and Pain Types`: Chest pain type, a critical symptom of heart issues, is categorized in the data set, helping to differentiate between various forms of heart disease.

Each variable in the data set has been carefully selected to contribute to a multi-faceted analysis of heart disease. The diversity of data points allows for a robust exploration using machine learning techniques, aiming to identify patterns and correlations critical in predicting heart disease.

Overview of the columns and their meanings:

- `Age`: Age of the patient in years.
- `Sex`: Sex of the patient (M: Male, F: Female).
- `ChestPainType`: Type of chest pain (ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic).
- `RestingBP`: Resting blood pressure in mm Hg.
- `Cholesterol`: Serum cholesterol in mm/dl.
- `FastingBS`: Fasting blood sugar (>120 mg/dl, 1: Yes, 0: No).
- `RestingECG`: Results of resting electrocardiogram (Normal, ST, LVH).
- `MaxHR`: Maximum heart rate achieved.
- `ExerciseAngina`: Exercise-induced angina (Y: Yes, N: No).
- `Oldpeak`: ST depression induced by exercise relative to rest.
- `ST_Slope`: The slope of the peak exercise ST segment (Up: upsloping, Flat: flat, Down: downsloping).
- `HeartDisease`: Presence of heart disease (1: Yes, 0: No).

Exploratory Data Analysis

Exploration 1: Descriptive Statistics

Understanding the distribution of various heart-related measures is crucial for identifying patterns that could be indicative of cardiovascular health risks.

The table below examines the sample minimum (Samp.Min), first quartile (Q1), median (Samp.Med), third quartile (Q3), and sample maximum (Samp.Max).We also consider the median absolute deviation (MAD), sample mean (SAM), standard deviation (SASD), skewness, and sample excess kurtosis. These metrics provide a comprehensive overview of the data distribution.

Table 1: Statistics

Variables	n	Samp.Min	Q1	Samp.Med	Q3	Samp.Max	MAD	SAM	SASD	Skew	Samp.Exc.Kurtosis
Age	918	28.0	47.00	54.0	60.0	77.0	10.38	53.51	9.43	-0.20	-0.40
RestingBP	918	0.0	120.00	130.0	140.0	200.0	14.83	132.40	18.51	0.18	3.23
Cholesterol	918	0.0	173.25	223.0	267.0	603.0	68.20	198.80	109.38	-0.61	0.10
FastingBS	918	0.0	0.00	0.0	0.0	1.0	0.00	0.23	0.42	1.26	-0.41
MaxHR	918	60.0	120.00	138.0	156.0	202.0	26.69	136.81	25.46	-0.14	-0.46
Oldpeak	918	-2.6	0.00	0.6	1.5	6.2	0.89	0.89	1.07	1.02	1.18
HeartDisease	918	0.0	0.00	1.0	1.0	1.0	0.00	0.55	0.50	-0.21	-1.96

Note: Descriptive statistics calculated for numeric variables in the dataset.

Exploration 2: Distributions across data

In this section of our exploratory data analysis, we examine the distributions of the various attributes,

Distribution by Age The histogram below provides a graphic representation of the frequency distribution of age within our dataset. Each bar in the histogram represents an age interval of five years, reflecting the number of individuals within that age range.

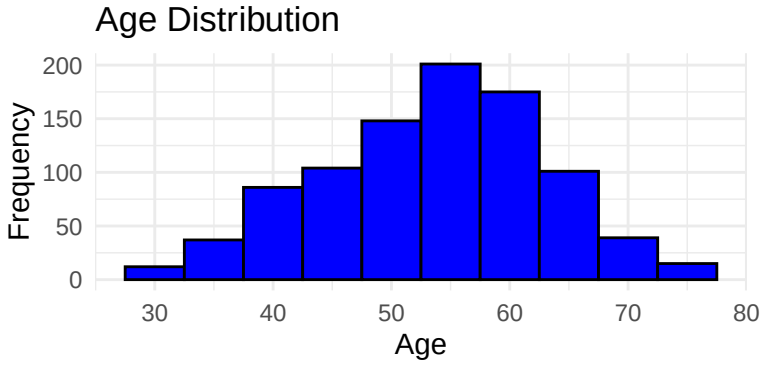


Figure 1: Age Distribution

The distribution reveals a concentration of individuals in the mid-life range (50-60 year age range), with fewer younger and older subjects within the data set.

Distribution by Sex The bar chart below provides a visual between the number of male and female participants. With blue representing males and red representing females.

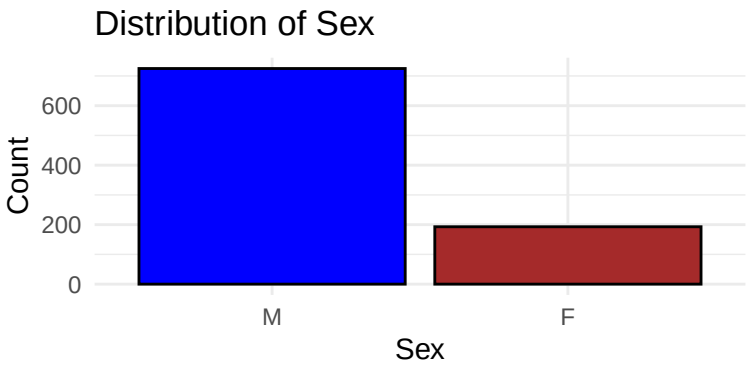


Figure 2: Sex Distribution

Immediately it is evident that there is a disparity in representation within our data set. This could be due to the nature of the data collection in the original studies.

Distribution of Resting ECG The resting electrocardiogram (ECG) is a diagnostic tool that provides insights into the electrical activity of the heart, playing a key role in the detection of various cardiac abnormalities.

The dataset categorizes resting ECG results into three types: Normal, ST (indicating ST-T wave abnormality), and LVH (suggesting left ventricular hypertrophy).

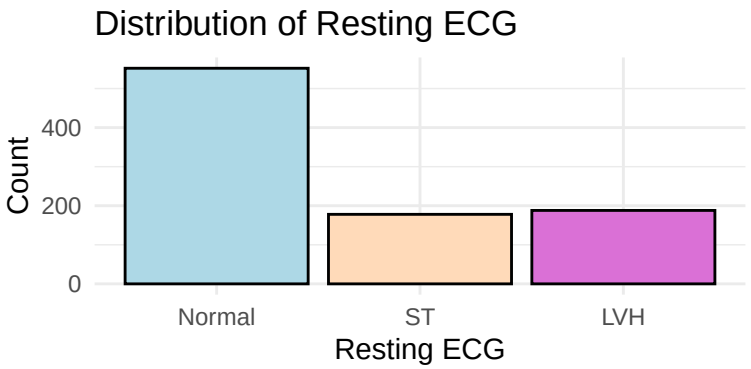


Figure 3: Resting ECG Distribution

The barchart above shows that majority of the subjects, represents those with no immediate ECG abnormalities. A high count in this category can suggest a relatively healthy cohort or, conversely, the potential for asymptomatic conditions that are not detectable through a resting ECG.

Distribution of Chest Pain Type Chest pain, as a symptom, plays a critical role in the diagnosis of cardiovascular conditions. The different categorization of chest pain type is an essential aspect of clinical assessments, as it can be indicative of various underlying cardiac issues.

Chest pain types are classified into four categories: Typical Angina (TA), Atypical Angina (ATA), Non-Anginal Pain (NAP), and Asymptomatic (ASY).

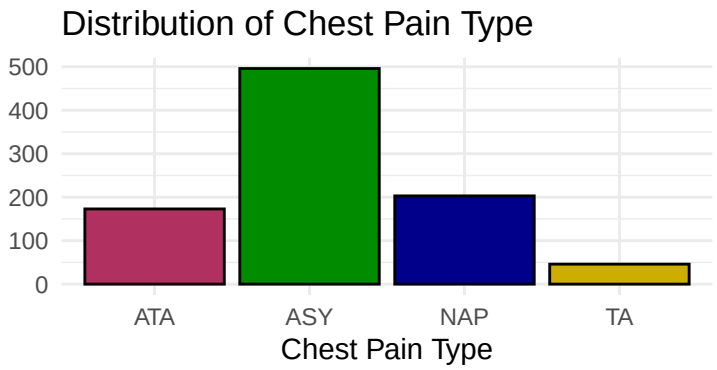


Figure 4: Chest Pain Type Distribution

The bar chart above visualizes the distribution of these categories among the data set. - `Typical Angina (TA)` : represented in yellow, has the least count, suggesting that classical symptoms of heart disease might be less common or underreported in the dataset. - `Atypical Angina (ATA)`: Shown in maroon, this type of chest pain is more prevalent than TA but less than the other two types.

- `Non-Anginal Pain (NAP)`: The blue bar indicates that this type of pain, which is not typically associated with angina, is relatively common among the subjects basically as much as ATA is.
- `Asymptomatic (ASY)` : Colored in green, represents the most common type among the data set subjects, this finding suggests that many individuals with heart disease may not experience the typical symptoms associated with angina.

The high prevalence of asymptomatic individuals (ASY) showcases the importance of screening and preventive measures, as many individuals at risk of Heart Disease may not exhibit clear warning signs.

Distribution of Exercise Angina Exercise-induced angina is a critical clinical indicator often associated with underlying coronary artery disease. Angina during exercise suggests that the heart may not be receiving enough oxygen-rich blood during increased physical activity.

The dataset captures this variable as a binary outcome—Yes (Y) for those who experience exercise-induced angina and No (N) for those who do not.

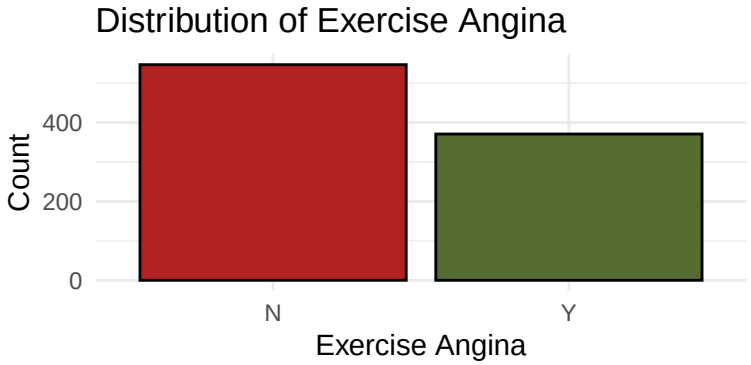


Figure 5: Exercise Angina Distribution

The bar chart visualizes the count of individuals who have experienced exercise-induced angina against those who have not. No Exercise Angina (N) group constitutes the larger portion of the dataset, indicating that a

significant number of subjects did not report angina precipitated by exercise. This could also highlights the potential for asymptomatic heart conditions or exercise patterns that do not provoke angina.

The presence of exercise-induced angina in thE bar chart is clinically significant because a substantial amount of subjects have experienced symptoms that suggest a higher risk for cardiovascular.

Distribution of ST Slope The ST Slope is a metric derived from the electrocardiogram (ECG) during exercise testing.

It is categorized into three distinct types: 'Up' for upsloping, 'Flat', and 'Down' for downsloping.

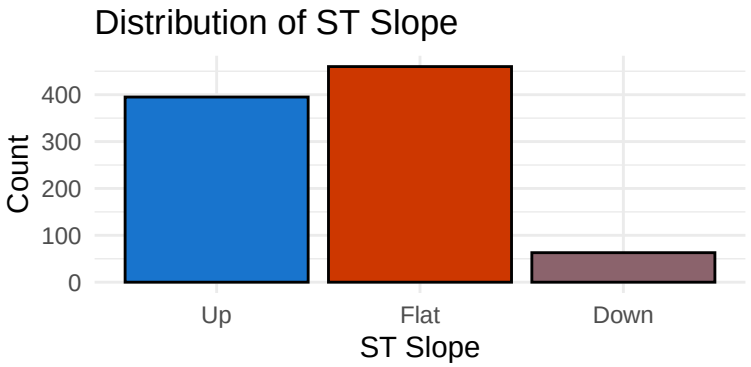


Figure 6: ST Slope Distribution

This bar chart shows the ST Slope categories within our dataset. Upsloping(up) suggests that the individuals' hearts are less likely to have significant coronary artery disease. Flat (Flat) indicates a significant portion of our data sets subjects may have some degree of cardiac concern. Downsloping (Down) ST Slope suggests a sign of higher risk for heart disease. The observed prevalence of upsloping and flat slopes informs us about the general cardiac health of the population in the study.

Distribution of Heart Disease The bar chart categorizes subjects into two groups: those diagnosed with heart disease ('Yes') and those without ('No'). The bars are filled with dark green for 'No' and brown for 'Yes'.

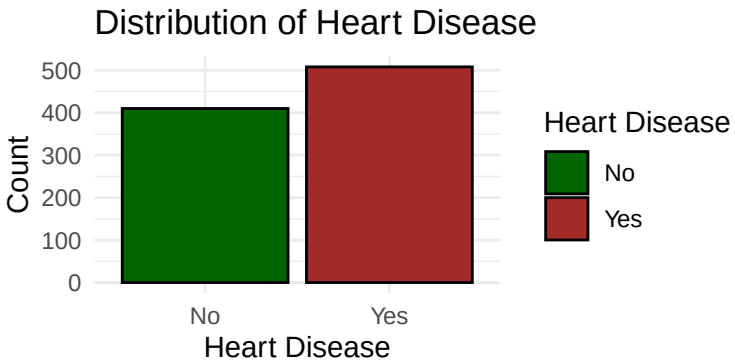


Figure 7: Heart Disease Distribution

The distribution is indicative of the data set's balance concerning the heart disease status of participants, which is helpful in avoiding bias toward one class.

Exploration 3: Heart Disease Distribution against different data

This section is on understanding the relationship between different attributes and heart disease.

Heart Disease Distribution by Age Age is a well-known risk factor for heart disease, with the risk increasing as people get older. To visualize the association between age and heart disease, we want tou use a boxplot

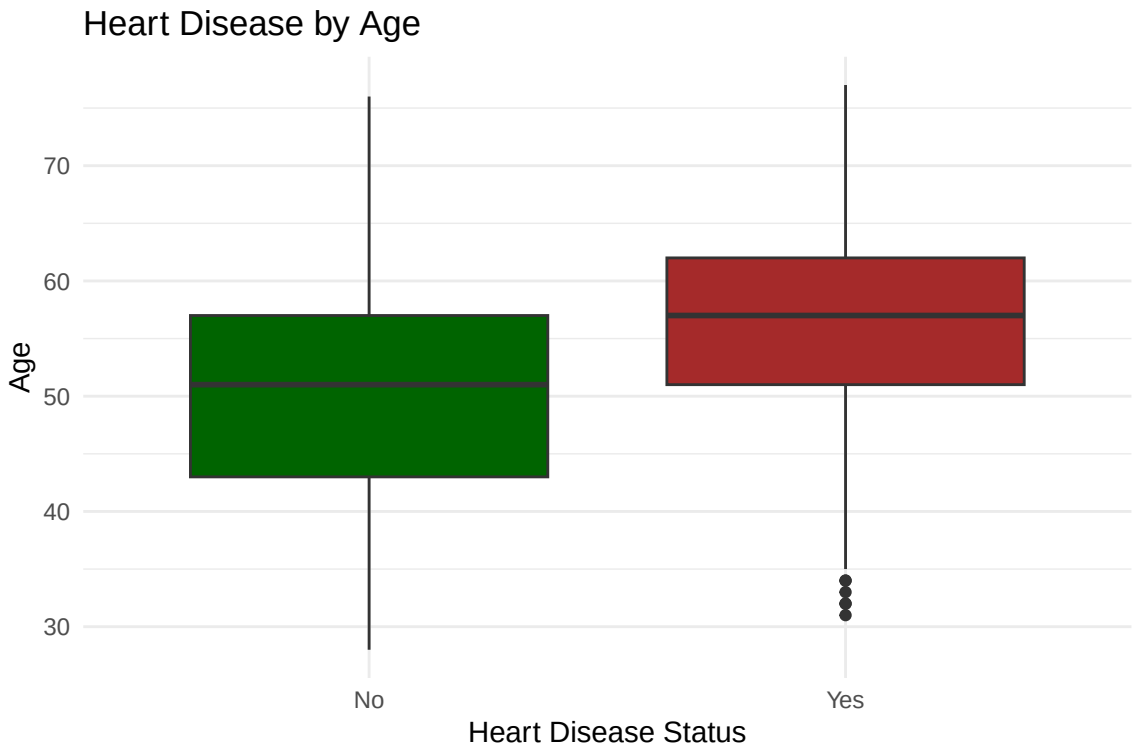


Figure 8: Boxplot of Age by Heart Disease Status

The Boxplot shows for those with heart disease, the median age is higher, and the IQR is broader, indicating a wider spread of ages and a tendency for heart disease to be more prevalent in older individuals.

Heart Disease Distribution by Sex In this analysis, we explore the prevalence of heart disease across different sexes within our dataset.

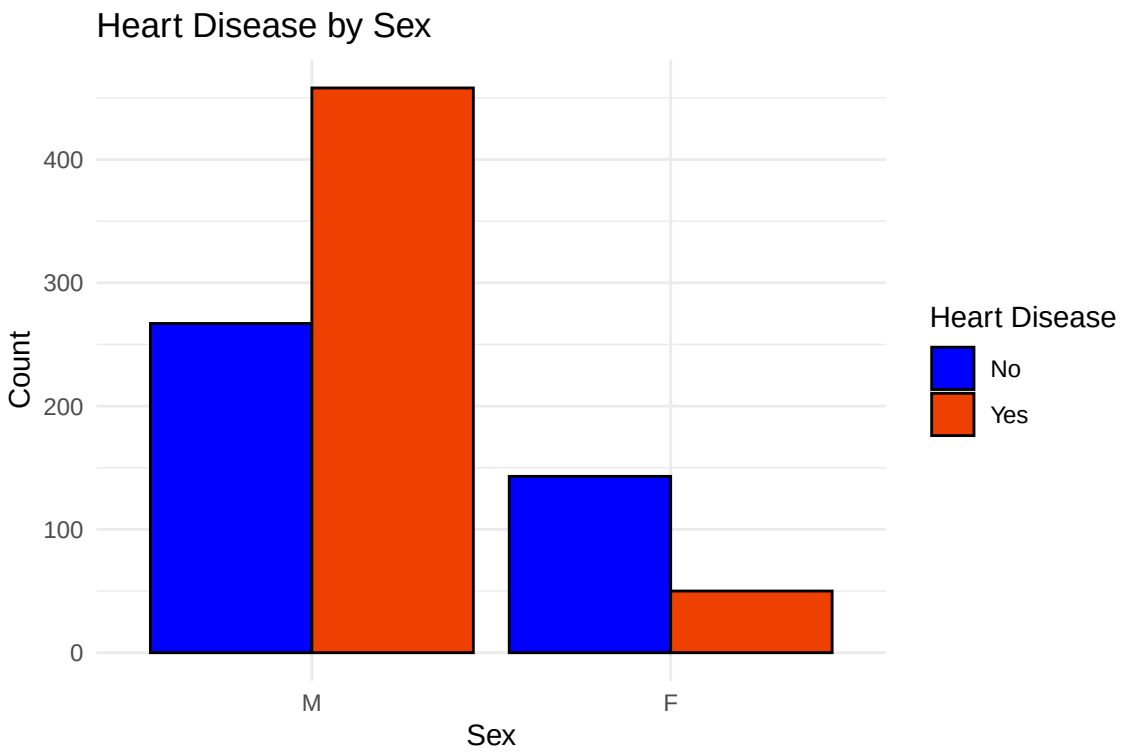


Figure 9: Bar Chart of Heart Disease Distribution by Sex

- `Males (M)`: The blue bar for males without heart disease is significantly shorter than the orange-red bar for males with heart disease. This suggests that within our dataset, a higher proportion of males have been diagnosed with heart disease
- `Females (F)`: For females, the blue bar for those without heart disease is taller than the orange-red bar for those with heart disease, indicating a lower prevalence of diagnosed heart disease in females within this dataset.

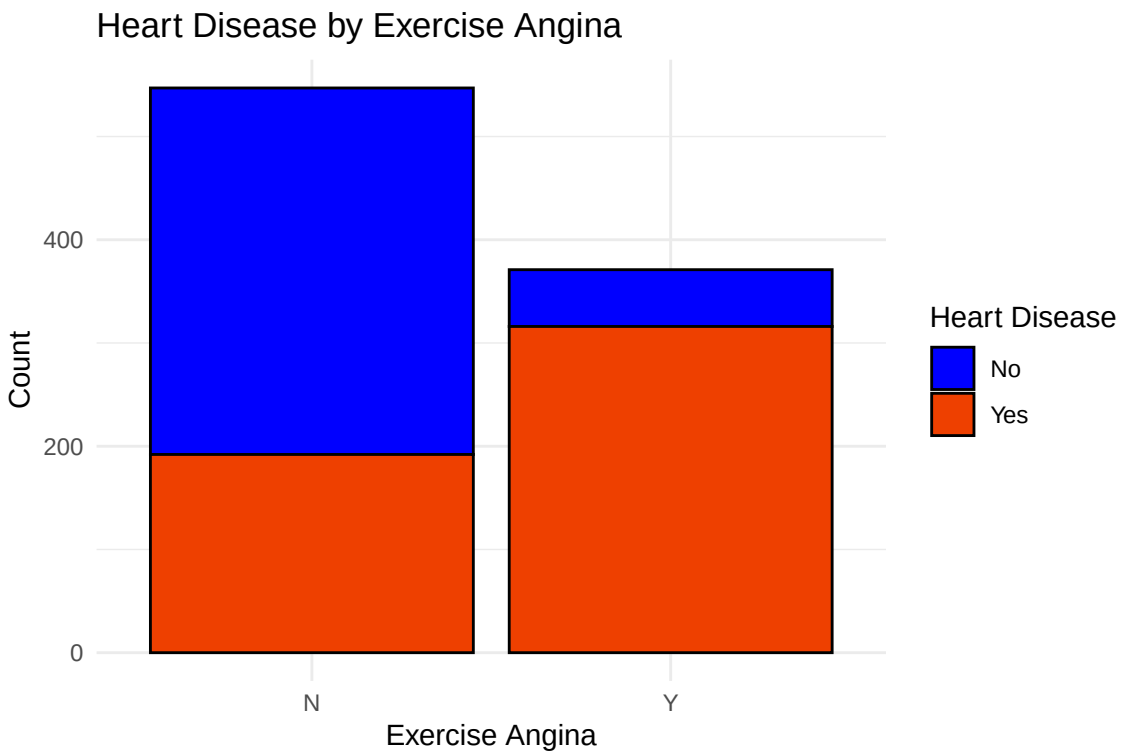


Figure 10: Bar Chart of Exercise Angina Distribution by Heart Disease

Heart Disease Distribution by Exercise Angina The distribution above shows the relationship between exercise-induced angina and heart disease, confirming that angina can be a significant indicator of cardiovascular issues. Specifically if we look at bar associated Exercise Angina (Y) it indicates a strong association between the occurrence of exercise angina and the presence of heart disease whereas the bar associated no Exercise Angina (N) indicates a stronger association between no heart disease occurrence than the presence of heart disease .

Heart Disease Distribution by MaxHR Maximum heart rate (MaxHR) achieved during stress testing is a significant indicator of cardiovascular fitness and heart health.

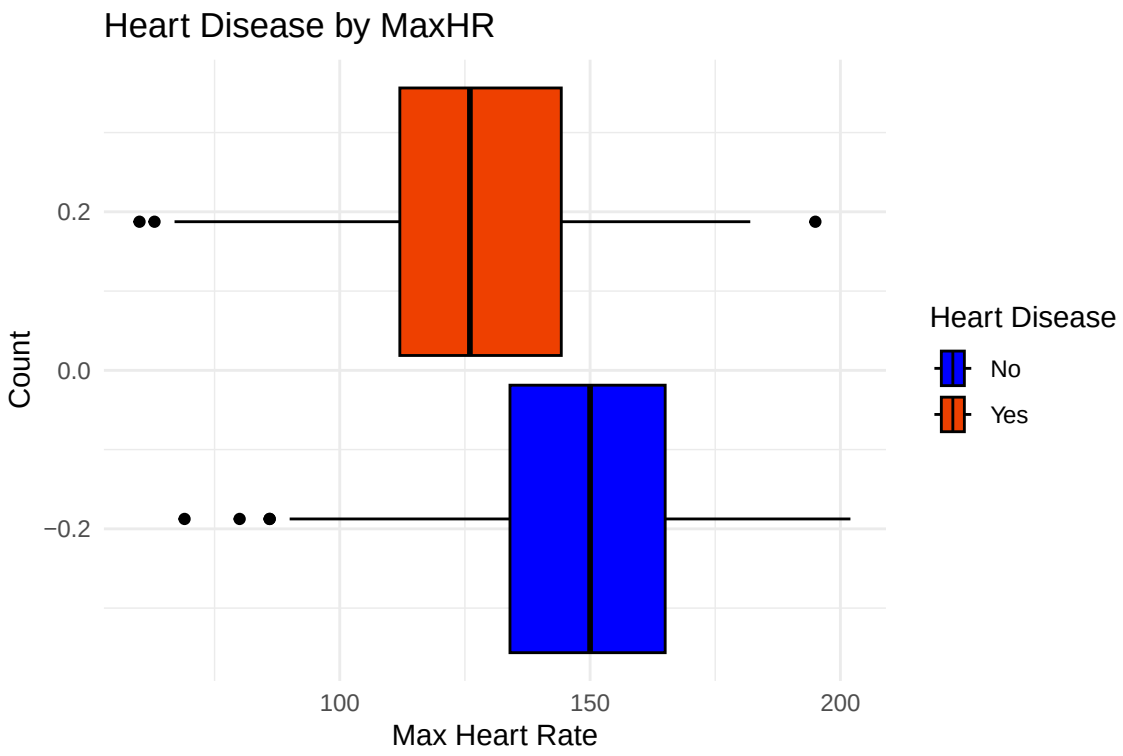


Figure 11: Boxplot of MaxHR by Heart Disease Status

The comparison of MaxHR between the two groups highlights the potential of MaxHR as a predictor of heart disease, suggesting that individuals with lower MaxHRs may be at increased risk.

The segment representing individuals with heart disease shows a generally lower median MaxHR and a wider IQR. This suggests a broader spread of MaxHR values among those with heart disease. As for those without heart disease the median MaxHR is higher in this group, and the interquartile range (IQR) is narrower, which aligns with expectations that a higher MaxHR is associated with better cardiovascular health.

Heart Disease Distribution by Oldpeak Oldpeak is a measurement derived from an exercise stress test and refers to the depression of the ST segment on an ECG.

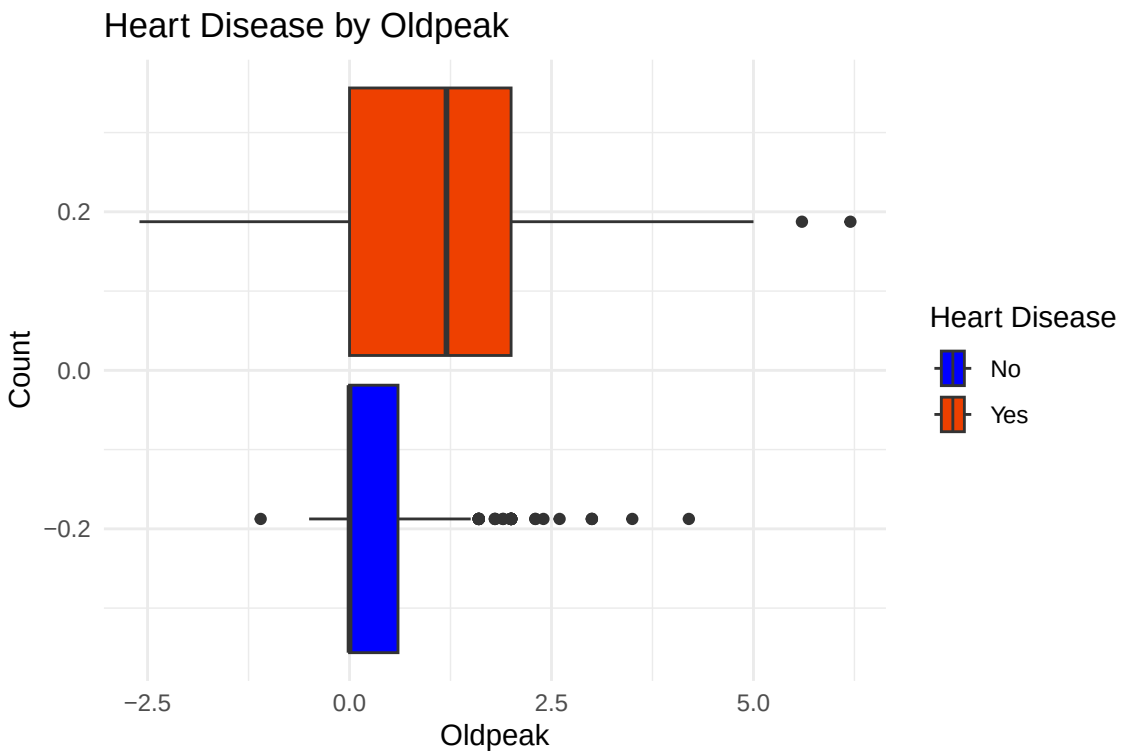


Figure 12: Boxplot of Oldpeak by Heart Disease Status

The comparison of Oldpeak values between individuals with and without heart disease in this dataset demonstrates the potential of Oldpeak as a predictor of heart disease. A higher Oldpeak value correlates with the presence of heart disease, aligning with clinical expectations.

Figure 13: Bar Chart of ST Slope Distribution by Heart Disease

Heart Disease Distribution by ST Slope The ST Slope is an informative clinical measure that, as evidenced by this chart, correlates with heart disease.

Upsloping ST Slope which is typically associated with a lower risk of heart disease, the 'Up' category shows a considerable base in blue, but a significant proportion in orange-red indicates the presence of heart disease. This suggests that while an upsloping ST Slope is generally a good sign, it does not rule out the risk of heart disease completely.The 'Flat' category emphasizies that a flat ST Slope may be a marker of higher risk for heart disease, requiring further clinical investigation. And lastly Downsloping ST Slope although the count for 'Down' is lower, the ratio of heart disease presence is significant and aligns with clinical expectations.

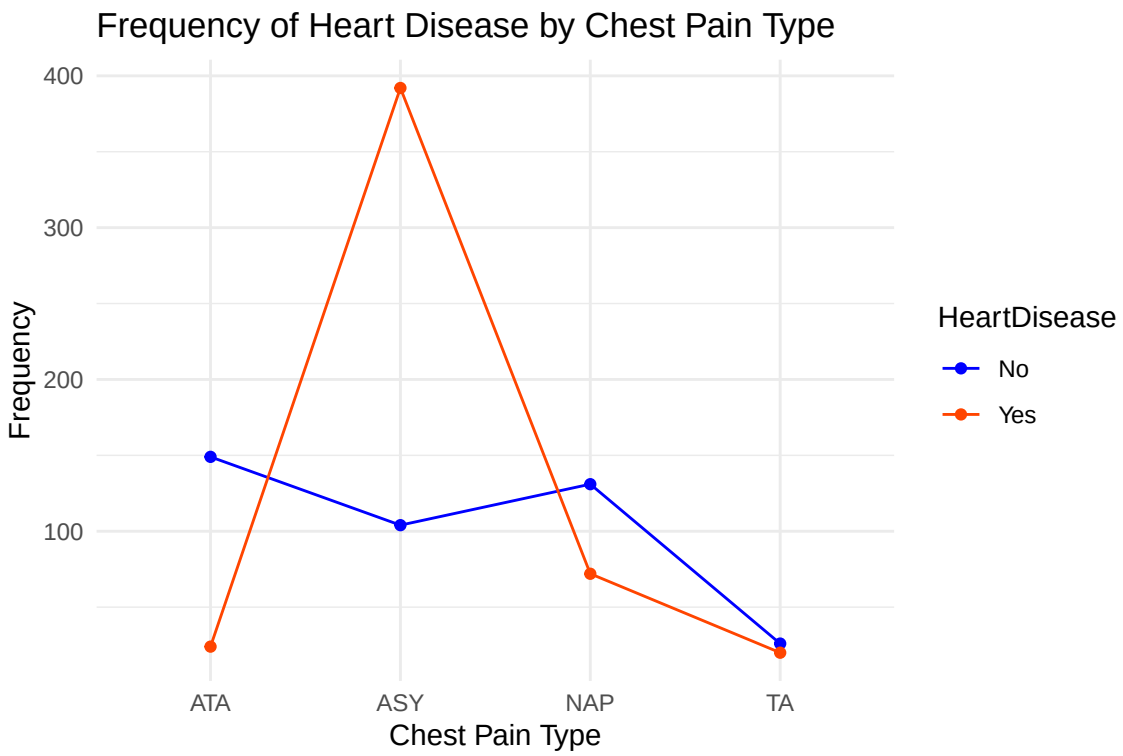


Figure 14: Line Plot of Frequency of Heart Disease by Chest Pain Type

Heart Disease Distribution by Chest Pain Type The frequency of chest pain types relative to heart disease status shows the complexity of cardiovascular diagnosis. While certain types of pain are more indicative of heart disease, the presence of heart disease in asymptomatic individuals cannot be overlooked.

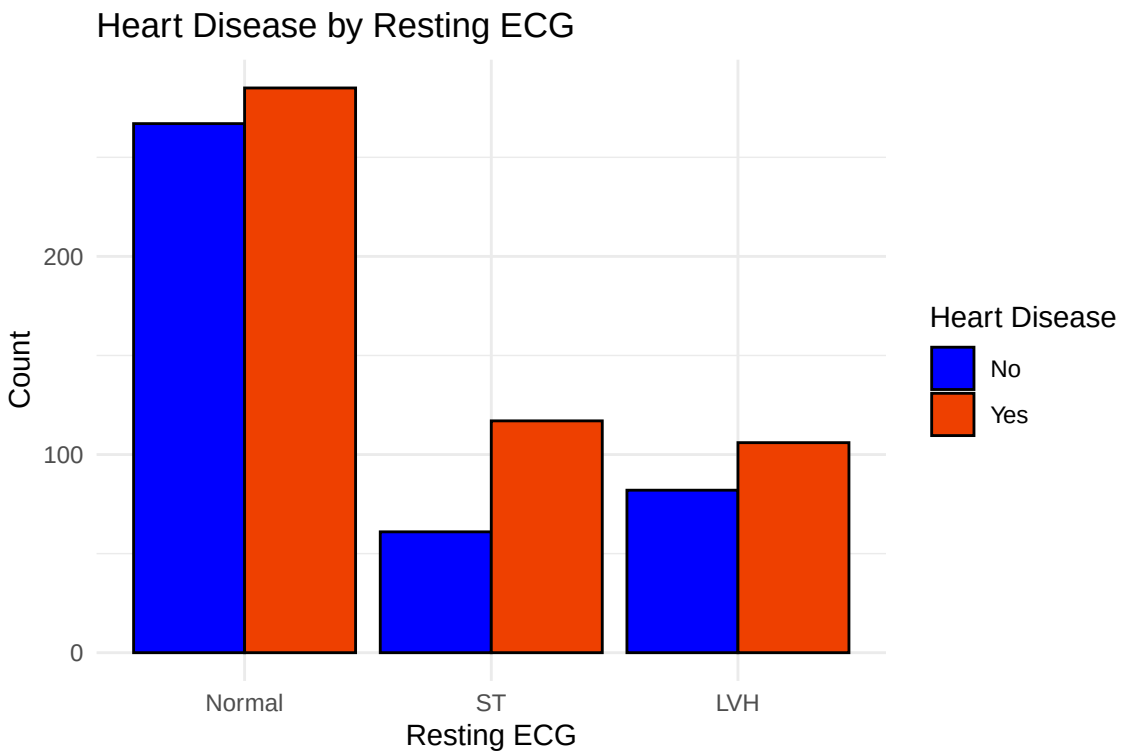


Figure 15: Bar Chart of Resting ECG Distribution by Heart Disease

Heart Disease Distribution by Resting ECG The distribution across these categories emphasizes the importance of resting ECG results as a diagnostic tool in assessing heart disease risk. While a 'Normal' ECG is reassuring, it is not entirely predictive of heart health, as evidenced by the substantial count of heart disease cases within this group.

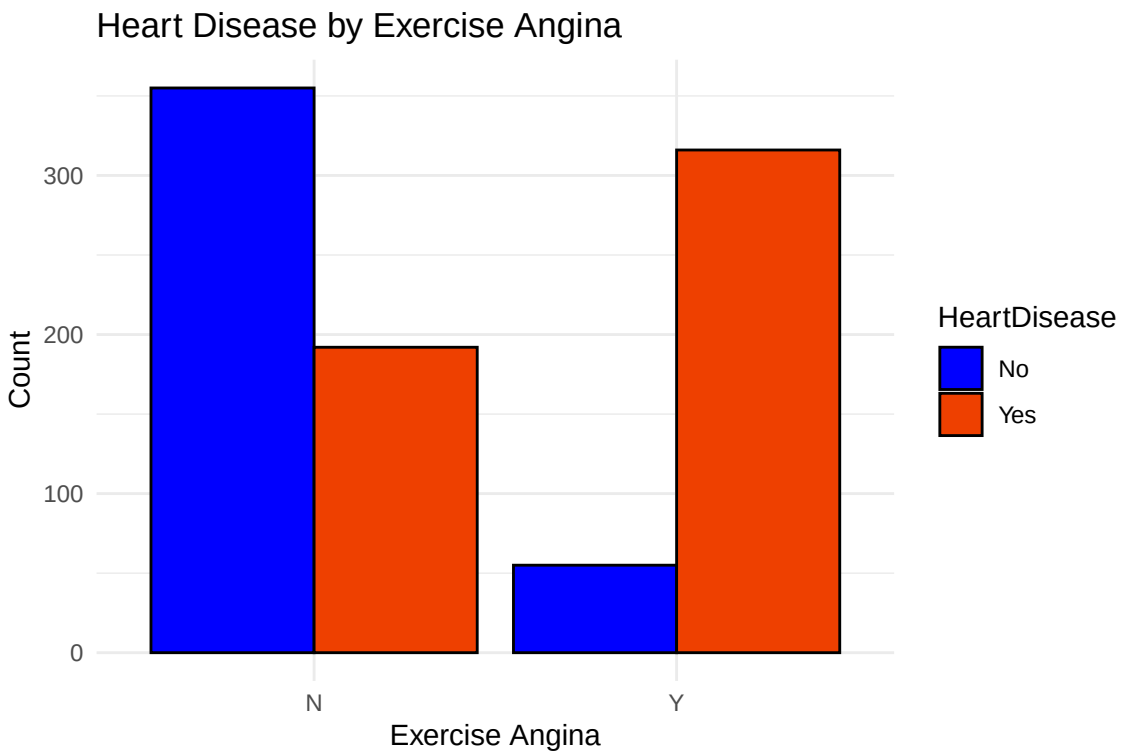


Figure 16: Bar Chart of Exercise Angina Distribution by Heart Disease

Heart Disease Distribution by Exercise Angina The distribution highlights the importance of exercise-induced angina as a symptom in the diagnosis of heart disease. While the presence of angina during exercise is a strong indicator of heart disease, the significant number of heart disease cases without exercise-induced angina calls attention to the need for a cardiac evaluation.

Heart Disease Distribution by Cholesterol Cholesterol levels are a factors in assessing cardiovascular risk; elevated cholesterol is often associated with an increased risk of heart disease due to the potential for arterial plaque buildup, which can lead to coronary artery disease.

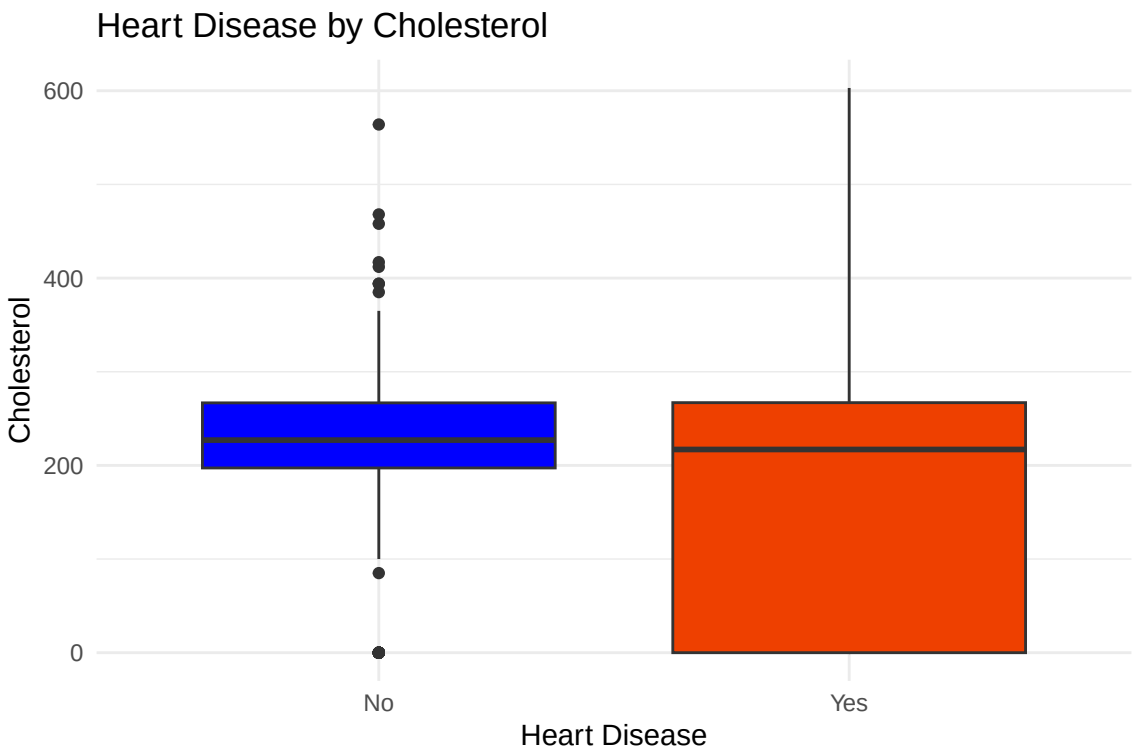


Figure 17: Boxplot of Cholesterol by Heart Disease Status

The median cholesterol level for individuals without heart disease falls within a normal range, and the interquartile range is relatively tight, suggesting a consistent pattern of cholesterol levels among this group. Notably, there are several outliers indicating some individuals with high cholesterol levels but no diagnosed heart disease. For those with heart disease, the median cholesterol level is also within a normal to slightly elevated range, but the interquartile range is broader, indicating more variability in cholesterol levels among affected individuals.

Heart Disease Distribution by Resting Blood Pressure Resting blood pressure is a fundamental measure of cardiovascular health. Elevated resting blood pressure, or hypertension, is a known risk factor for heart disease as it can lead to damage within the arteries and reduced blood flow to the heart.

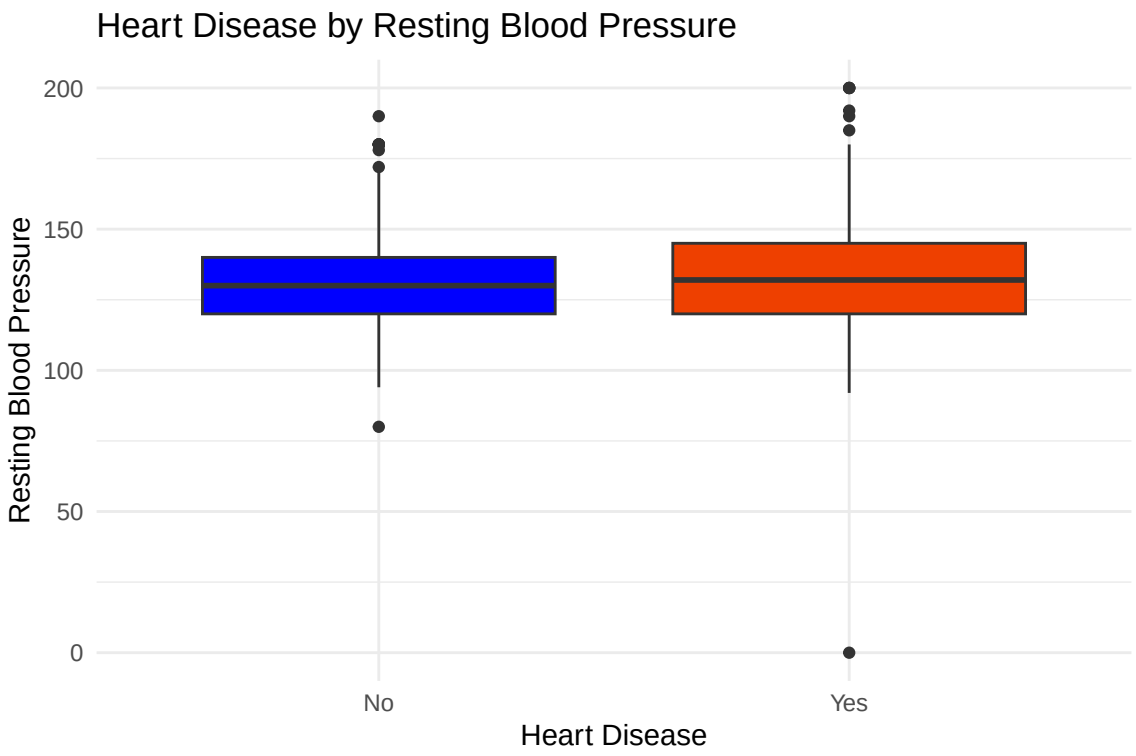


Figure 18: Boxplot of Resting Blood Pressure by Heart Disease Status

This boxplot indicates that while elevated resting blood pressure is associated with heart disease, the relationship is not absolute, as some individuals with high blood pressure do not have heart disease, and vice versa.

Exploration 4: Correlation Plot

The correlation plot provides a visual representation of how closely these health indicators are related to one another.

In the correlation matrix:

Darker blue indicates a stronger positive correlation, where higher values of one variable are associated with higher values of another. Darker red suggests a stronger negative correlation, indicating that higher values of one variable are associated with lower values of another. Lighter shades of blue or red indicate weaker correlations.

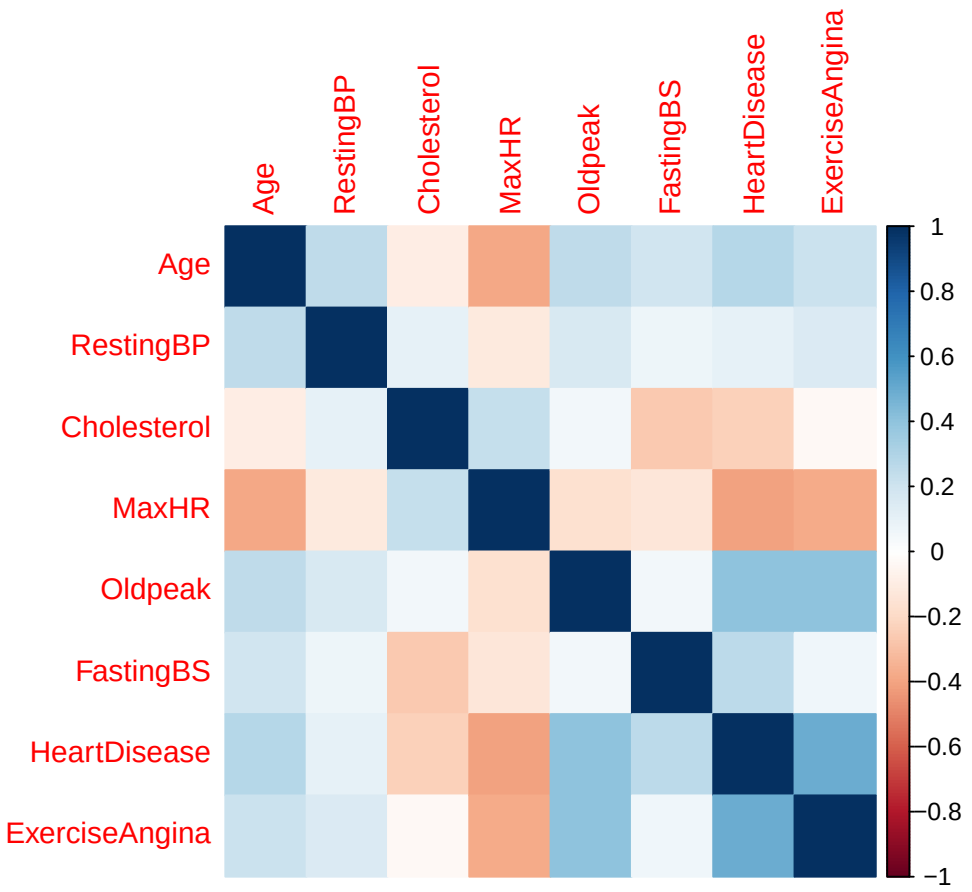


Figure 19: Correlation Matrix of Numerical Variables

There is a notably strong positive correlation between ExerciseAngina (exercise-induced angina) and HeartDisease, suggesting that individuals who experience angina during exercise are more likely to have heart disease. Oldpeak (ST depression induced by exercise relative to rest) also shows a positive correlation with HeartDisease, which aligns with the clinical understanding that ST depression can indicate myocardial ischemia. MaxHR (maximum heart rate achieved) displays a negative correlation with HeartDisease, indicating that lower maximum heart rates during stress tests may be associated with a higher risk of heart disease. Age shows a positive correlation with HeartDisease, reflecting the increased risk of heart conditions with advancing age. Other variables such as RestingBP (resting blood pressure) and Cholesterol show more moderate correlations with HeartDisease

Heatmap of Resting ECG and Chest Pain Type

The heat map provides a visual representation of the frequency of chest pain types across different categories of resting electrocardiogram (ECG) results.

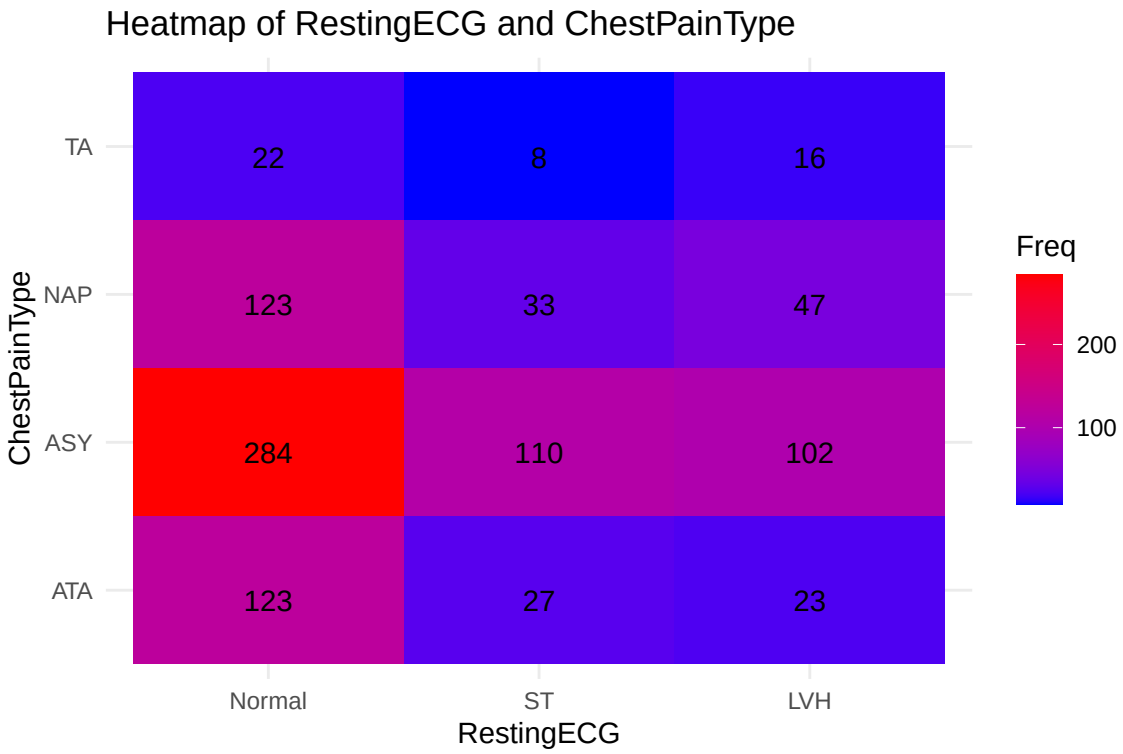


Figure 20: Heatmap of RestingECG and ChestPainType

The heat map illustrates a higher frequency of asymptomatic chest pain (ASY) among individuals with normal resting ECG results, highlighted by the intense color. Atypical angina (ATA) appears to have a uniform distribution across the different ECG results, suggesting a less specific relationship between this type of chest pain and ECG outcomes. Typical angina (TA) is notably less frequent across all ECG categories, indicating its rarity in this data set.

Machine Learning Algorithms

Logistic Regression

Methodology

Given that our goal is to build a classifier that will predict whether a person is going to have heart disease or not (binary), logistic regression provides an excellent algorithm to build our classifier. To provide model validation evidence, we will use an 80% training–20% testing split based upon stratified random sampling. We will do this after removing any cases with missing information.

We will then build two candidate models. The first model will use a single predictor oldpeak based upon prior research. The second candidate model will emerge from a step wise feature selection search pegged to the Akaike Information Criterion (AIC). We will evaluate both of these models and refine them to a final model. We will assess this final model on our testing data.

We will develop two logistic regression models for comparison:

- Model 1: Utilizes ‘Oldpeak’ as a single predictor
- Model 2: Generated from a stepwise feature selection

Results

We will present our results in three parts. First, we'll discuss the two initial models, separately. Then we'll compare the two models and discuss how we refined the models before testing out the refined model with our testing data. For any hypothesis testing and confidence interval construction, we'll control our overall Type I error rate at 7%. For any confusion matrices, we will draw upon a naïve rule where any predicted probability of a person having heart disease greater than 0.5 will classify the person as having heart disease.

Heart Predition: Model 1

Term	Coefficient	Prob./Odds Ratio	Std. Err.	Z	p-value
Intercept	-0.574	0.563	0.105	-5.462	< 0.001
Oldpeak	1.036	2.819	0.100	10.389	< 0.001

Note: Logistic Model 1 Coefficient Table

The analysis involves a logistic regression model for heart disease, specifically focusing on the Oldpeak predictor. The intercept term's coefficient is -0.574, with a corresponding probability of 0.563 and a standard error of 0.105. This implies that when Oldpeak is near 1, the probability of not having heart disease is essentially 0. The coefficient for Oldpeak is 1.036, associated with a probability or odds ratio of 2.819 and a standard error of 0.100. The change in the log-odds for heart disease increases by a factor of 1.036 for each unit increase in Oldpeak. The statistical significance is evident, as the p-value for both the intercept and Oldpeak is less than 0.001. The model also includes a Z-score of -5.462 for the intercept and 10.389 for Oldpeak, further reinforcing their significance in predicting heart disease.

Confusion Matrix Using a confusion matrix we measure the performance of a classification model by showing the actual versus predicted classifications.

The confusion matrix output provides a breakdown of the model's predictions:

- `True Positives (TP)`: The number of patients with heart disease correctly identified by the model.
- `False Negatives (FN)`: The number of patients with heart disease the model incorrectly predicted as no disease
- `False Positives (FP)`: The number of patients without heart disease incorrectly labeled as having it.
- `True Negatives (TN)`: The number of patients without heart disease correctly identified by the model.

Predicted/Actual	Heart Disease	No Heart Disease
Heart Disease	284	85
No Heart Disease	126	239

This model's accuracy is approximately 71% , with a sensitivity of 69% and a specificity of 73% . While Model 1 is better than a fair coin, this model is not far from the fair coin under our naïve rule and could be a lot better.

Heart Predition: Model 2

Term	Coefficient	Prob./Odds Ratio	Std. Err.	Z	p-value
Intercept	-3.368	0.034	0.848	-3.973	< 0.001
ST Slope: Flat	2.904	18.250	0.291	9.985	< 0.001
ST Slope: Down	1.091	2.977	0.515	2.120	0.034
Chest pain type: Asymptomatic	1.801	6.056	0.371	4.859	< 0.001
Chest pain type: Non-Anginal Pain	-0.080	0.923	0.410	-0.195	0.846
Chest pain type: Typical Angina	0.617	1.853	0.589	1.048	0.295
Sex: Female	-1.787	0.167	0.325	-5.499	< 0.001
Fasting Blood Sugar	1.109	3.031	0.321	3.450	< 0.001
Oldpeak	0.454	1.575	0.136	3.328	< 0.001
Exercise-induced angina: Yes	0.866	2.377	0.277	3.124	0.002
Cholesterol	-0.003	0.997	0.001	-2.871	0.004
Age	0.024	1.024	0.014	1.702	0.089

Note: Logistic Model 2 Coefficient Table

From the table a flat ST_Slope has a large positive coefficient, indicating a strong association with the presence of heart disease. Chest pain types Asymptomatic (ASY) is positively associated with heart disease, while Non-anginal pain(NAP) and Typical angina(TA) are not statistically significant.Female gender (SexF) has a negative coefficient, suggesting a lower odds of heart disease compared to males but this could also be because of the lack of alot females in the study.

Attribute such as Fasting blood sugar(FastingBS), Oldpeak (ST depression), and Exercise-induced angina(ExerciseAnginaY) are positively associated with heart disease. Cholesterol shows a very small negative association with heart disease. Age has a small positive coefficient, indicating a slight increase in the odds of heart disease with age.

Tukey-Anscombe Plot

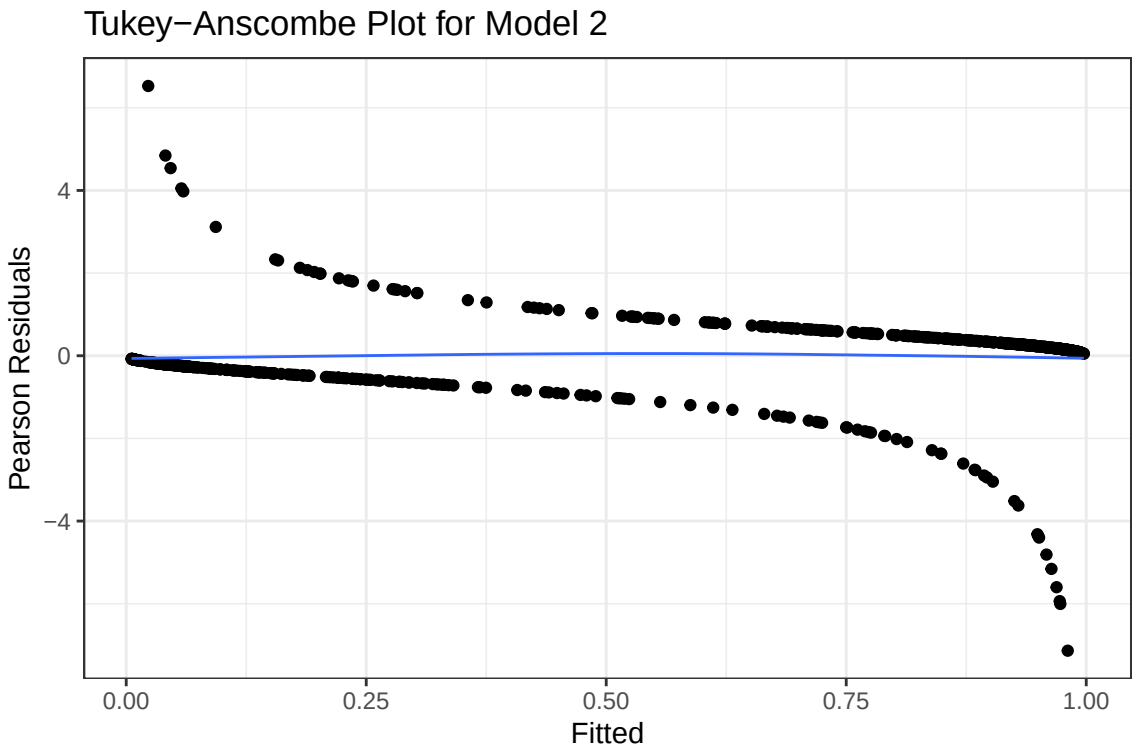


Figure 21: Tukey-Anscombe Plot for Logistic Regression Model 2

Above shows the Tukey-Anscombe plot using Pearson residuals for Model 2. We can see the classical pattern for logistic regression in the plot; the smoothing line does indicate that residuals are centered around zero.

GVIF for Model 2

term	GVIF	Df	GVIF^(1/(2*Df))	squared
ST_Slope	1.576	2	1.121	1.256
ChestPainType	1.216	3	1.033	1.067
Sex	1.179	1	1.086	1.179
FastingBS	1.110	1	1.053	1.110
Oldpeak	1.320	1	1.149	1.320
ExerciseAngina	1.167	1	1.080	1.167
Cholesterol	1.107	1	1.052	1.107
Age	1.082	1	1.040	1.082

The table above shows the generalized variance inflation factors for candidate Model 2. We need to be a bit cautious when examining these values as the current calculations do not adequate account for the higher order interactions. None of the GIVF's are inflated so this points to model 2 being a good model.

Confusion Matrix

Predicted/Actual	Heart Disease	No Heart Disease
Heart Disease	376	47
No Heart Disease	34	277

This model's accuracy is approximately 89%, with a sensitivity of 85% and a specificity of 91% . Model 2 is a lot better than a fair coin.

Model Comparison

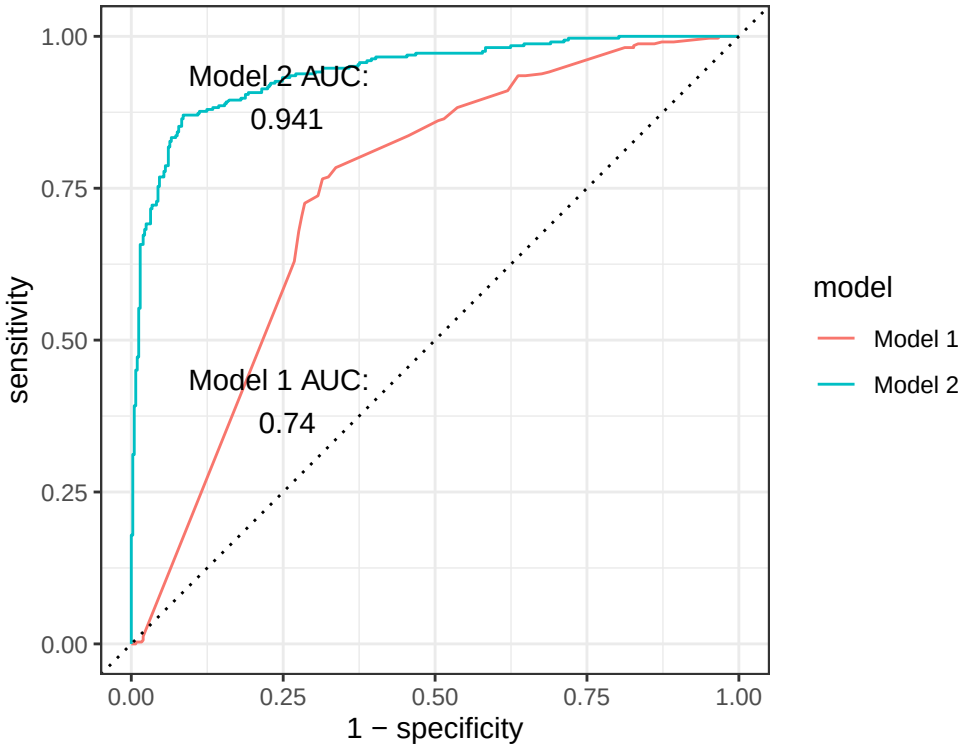


Figure 22: ROC Curves for Models 1 and 2

As the confusion matrices highlight, our two models are quite different from one another, with the multiple logistic regression model (i.e., Model 2) doing far better. The figure above shows the ROC curves and AUC values for our two initial models. As anticipated, the second candidate model (the blue curve) does much better than the first model (the red curve).

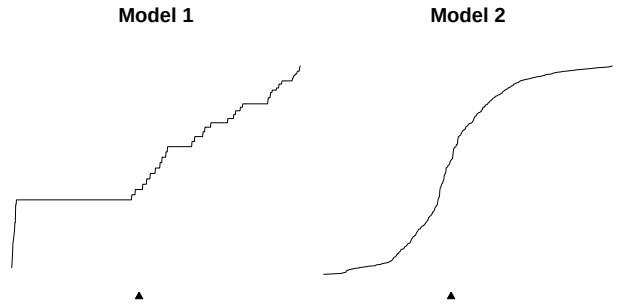


Figure 23: Separation Plots for Models 1 and 2

Separation Plots Based on the previous plots and tables model 2 seems to be a great model to use. Therefore we chose Model 2 as our model to evaluate on the testing data.

Testing

Predicted/Actual	Heart Disease	No Heart Disease
Heart Disease	86	21
No Heart Disease	12	65

Figure 24: Separation Plot for Testing Data on Model 2

This shows the confusion matrix of our testing data and using the naïve decision rule. Our accuracy is approximately 82%, with a sensitivity of 85% and a specificity of 75%. While not as accurate as the training data it isn't that far of. The Separation plot also shows the model does a decent job predicting Heart disease.

Assessing and Evaluating the Candidate Model

Model Fit Metrics For evaluating the models, we employ the following evaluation metrics:

- `Root Mean Squared Error (RMSE)`(Lower is Better): RMSE measures the average prediction error and indicates how well the model fits the testing data.
- `R-squared (R²)`(Higher is Better): R-squared quantifies the proportion of variance in the dependent variable explained by the model. A higher R-squared value suggests a better model fit.
- `Akaike Information Criterion (AIC)`(Lower is Better): AIC is a measure of the model's goodness of fit, considering model complexity. Lower AIC values indicate a better model.

- `Bayesian Information Criterion (BIC)`(Lower is Better): Similar to AIC, BIC is another criterion for model selection, penalizing complex models. Lower BIC values are preferred.

```
## AIC Model 1:  863.885
```

```
## AIC Model 2:  462.5637
```

```
## BIC Model 1:  873.082
```

```
## BIC Model 2:  517.7458
```

```
## Brier Score Model 1:  0.1994835
```

```
## Brier Score Model 2:  0.08878123
```

```
## Pseudo R-squared Model 1:  0.2439421
```

```
## Pseudo R-squared Model 2:  0.7224204
```

This analysis demonstrates that Model 2 significantly outperforms Model 1 across all metrics, indicating it is the more accurate and efficient model for predicting the outcome variable.

In summary, we've built a classifier using multiple logistic regression to predict Heart disease. Our chosen model (Model 2) uses the step wise selection to chose predictors to draw upon from.

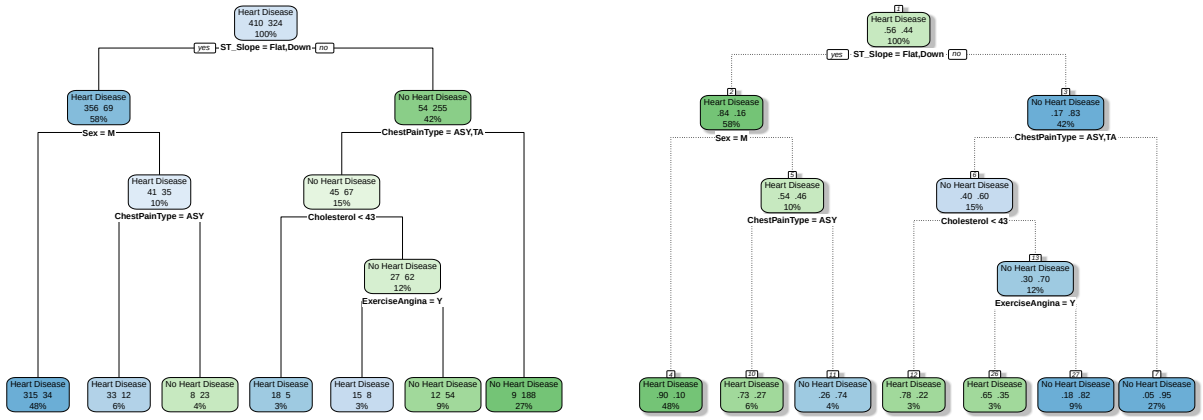
Decision Tree Analysis

Within this area of the project, because of our data set not being the biggest thing ever, we are more prone of over fitting using decision trees.

Part 1: Applying CART

To predict the likelihood of heart disease, we've constructed a decision tree using the rpart package in R exclusively.

Growing Tree The decision tree algorithm constructed with rpart examines a set of variables including Age, Sex, Chest Pain Type, Cholesterol Level, Resting Blood Pressure (RestingBP), Fasting Blood Sugar (FastingBS), Resting Electrocardiogram results (RestingECG), Maximum Heart Rate (MaxHR), Exercise-induced Angina (ExerciseAngina), ST depression induced by exercise relative to rest (Oldpeak), and the slope of the peak exercise ST segment (ST_Slope). By analyzing these predictors, the tree segments the data into subgroups, or nodes, aiming to uncover patterns that assist in predicting the presence or absence of heart disease.



(a) Plot

(b) Rattle plot

Figure 25: Plot of the Decision Tree Model

Visualize our trees Above is a plot for our initial rPart tree. This tree gives us insight in some good statistics about its decision making based on if a person has heart disease or not. Each node within the tree displays two numbers and a corresponding percentage, indicating the distribution of patients with and without heart disease, as well as the proportion of the total dataset represented by that node. For example, a node showing 46% implies that nearly half of the dataset's subjects fall within that particular classification category.

The nodes are also color-coded: blue indicates subjects with heart disease, while green represents those without. The intensity of the color correlates with the distribution within the node - a lighter shade signifies a more balanced distribution between the two classes, whereas a darker shade denotes a more significant skew towards one class.

CP	Num. of splits	Rel. Error	Mean Error	Std. Deviation of Error
0.620	0	1.000	1.000	0.042
0.023	1	0.380	0.380	0.031
0.020	3	0.333	0.392	0.032
0.010	6	0.272	0.327	0.029

Pruning The table above shows as the CP decreases, the decision tree grows more complex with additional splits, and the relative error tends to decrease, indicating a more complex model that may fit the training data better but could be more prone to overfitting.

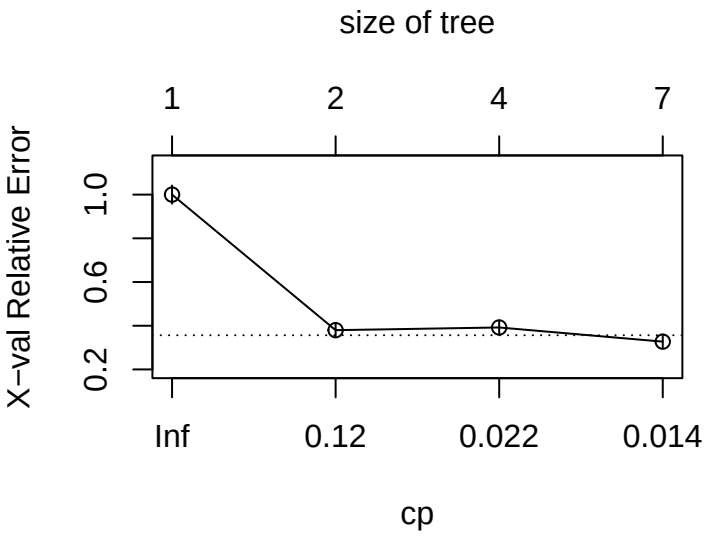


Figure 26: Cost Complexity Results for rpart Tree

Plot View From the plot CP value below the line is 0.014. therefore CP = 0.014

Testing and Predicting

For a more accurate understanding of how well the model fits the context we will test our model on testing data.

	Heart Disease	No Heart Disease
Heart Disease	88	26
No Heart Disease	10	60

Note: Pruned Decision Tree Model confusion Matrix

Building Confusion Matrix The shows the accuracy, sensitivity, and specificity for the final model we created.

model	accuracy	sensitivity	specificity
rpart2_pred	0.804	0.698	0.898

Under the naïve classification rule,The rpart model accuracy is approximately 80.4%, with a sensitivity of approximately 0.698 and a specificity of approximately 89.8%

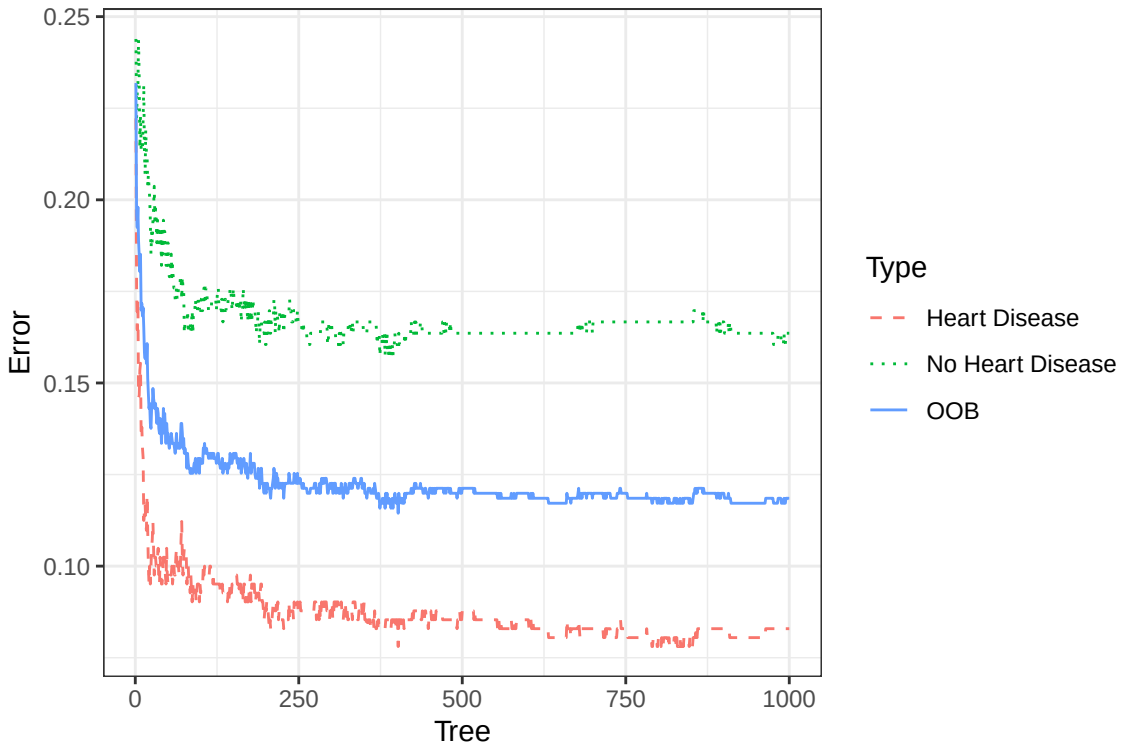


Figure 27: Out-of-bag Error and Misclassification

Random Forest Model The plot above shows the Out-of-Bag Error rates as well as the misclassification error rates for each of the trees in our random forest. As more trees are added to the model, the OOB error rate stabilizes specifically around the 250th tree. This stability suggests that the model has reached a reliable level of performance

Attribute Importance

	Heart Disease	No Heart Disease	MeanDecreaseAccuracy	MeanDecreaseGini
Age	5.618	15.307	14.368	26.055
Sex	22.686	29.782	35.746	14.526
ChestPainType	27.664	36.922	44.123	46.474
RestingBP	5.083	7.596	8.731	24.501
Cholesterol	7.658	20.964	20.149	32.822
FastingBS	13.864	19.435	22.283	7.744
RestingECG	6.054	5.304	8.087	6.919
MaxHR	18.341	5.698	19.096	34.027
ExerciseAngina	15.338	22.803	27.549	23.552
Oldpeak	8.885	41.449	37.622	31.829
ST_Slope	59.221	114.938	118.119	112.728

The table shows us each of the attributes we put into the pool of predictors/factors. The first two columns (labeled “Heart Disease” and “No Heart Disease”) are the prediction errors (via Out-of-Bag cases). The last two columns are the key ones we want to look at. We ultimately want the attributes that lead to the better

model. Thus, creating purer nodes means we want the largest mean decrease in the Gini Index. For the mean decrease in accuracy, we want to imagine what would happen if we deleted that attribute. Thus, larger decreases would result from deleting a more important attribute. The figure below gives a visual display of this information.

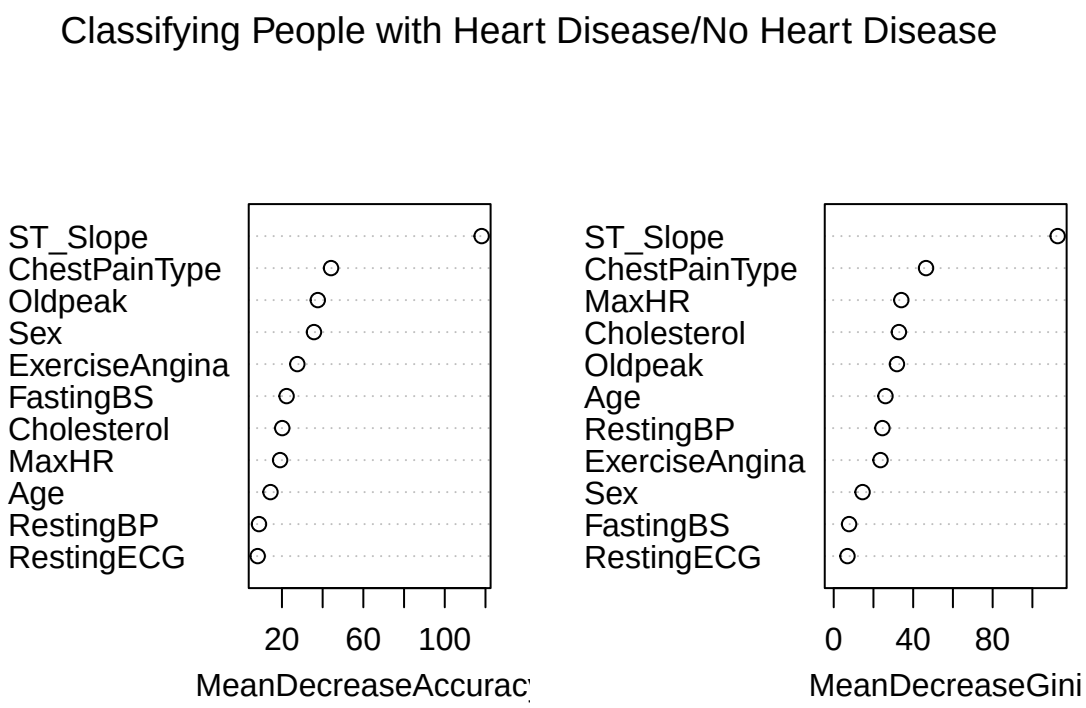


Figure 28: Variable Importance Plot for Model

We can see from the visualization above that ST_Slope appears to be the most significant variable, showing the highest decrease in accuracy and Gini when omitted, indicating a strong relationship with heart disease presence.Sex and Chest Pain Type are also important predictors, for distinguishing between the presence and absence of heart disease, as reflected in their accuracy decrease. MaxHR (Maximum heart rate) and ExerciseAngina (chest pain induced by exercise) also have notable importance in prediction of heart Disease.

Testing/Predicting

Prediction/Supervision	Heart Disease	No Heart Disease
Heart Disease	88	23
No Heart Disease	10	63

model	accuracy	sensitivity	specificity
predicted	0.821	0.733	0.898

The Models accuracy is approximately 82.1%, with a sensitivity of approximately 0.733 and a specificity of approximately 78.9%. which is a 2% increase for accuracy than the prior tree.

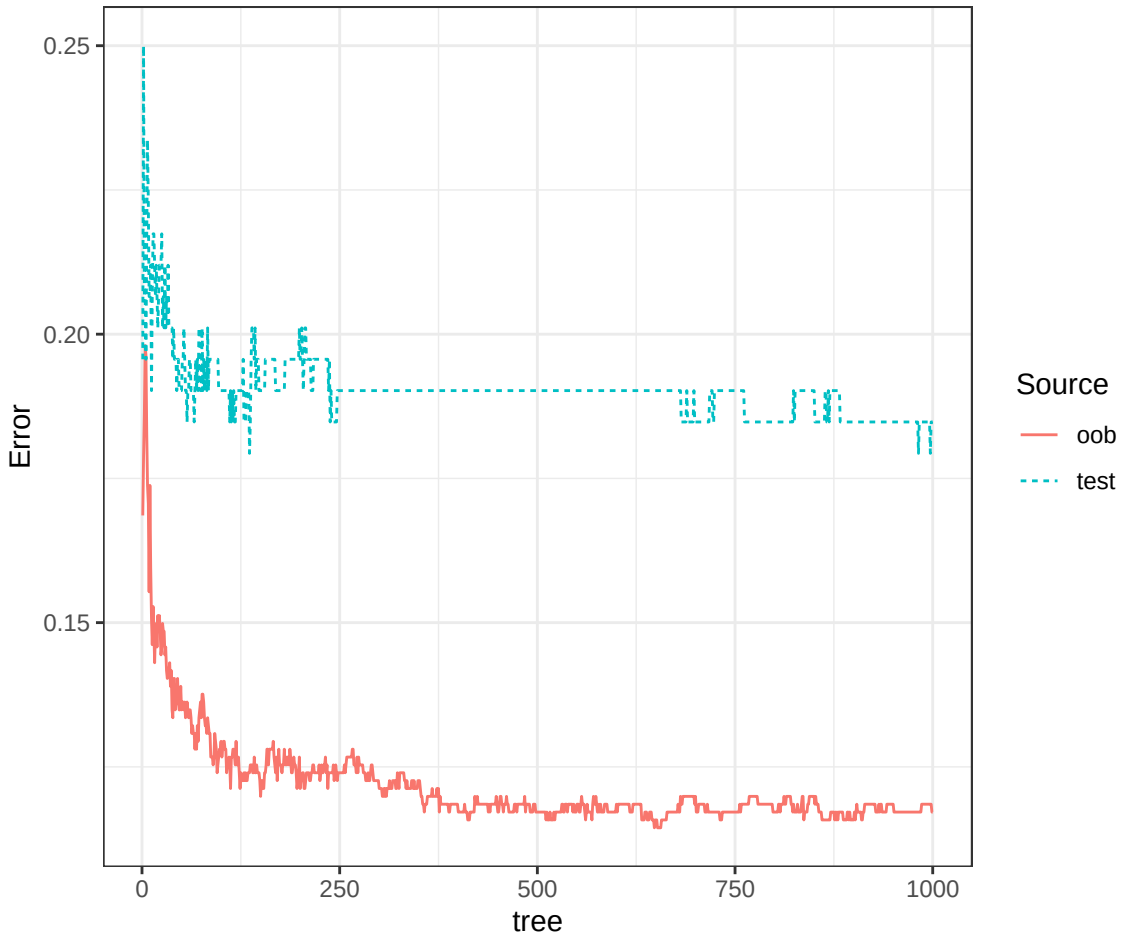


Figure 29: Plot of OOB and Testing Set Errors

Above is a plot for our test set against the OOB again. Within this plot we see a blue line representing the test data, and the red line representing the OOB. We can see that both of their error rates are very low, providing us info that our predictions are almost always correct.

Clustering

In preparation for clustering, we first standardize the quantitative attributes of our heart dataset.This step is crucial for clustering algorithms like K-means, which are sensitive to the scale of the data.

For qualitative attributes, we focus on Sex, ExerciseAngina, FastingBS, and HeartDisease. These variables are binary in nature, making them suitable for conversion into a coded (numeric) format. Specifically, we encode 'F' (female) as 1 and 'M' (male) as 0 for Sex, and similarly, 'Y' (yes) as 1 and 'N' (no) as 0 for ExerciseAngina.

We then select the standardized quantitative variables along with the encoded qualitative variables for clustering.

Distance Matrix For our clustering analysis, we have decided to use the Euclidean distance metric.

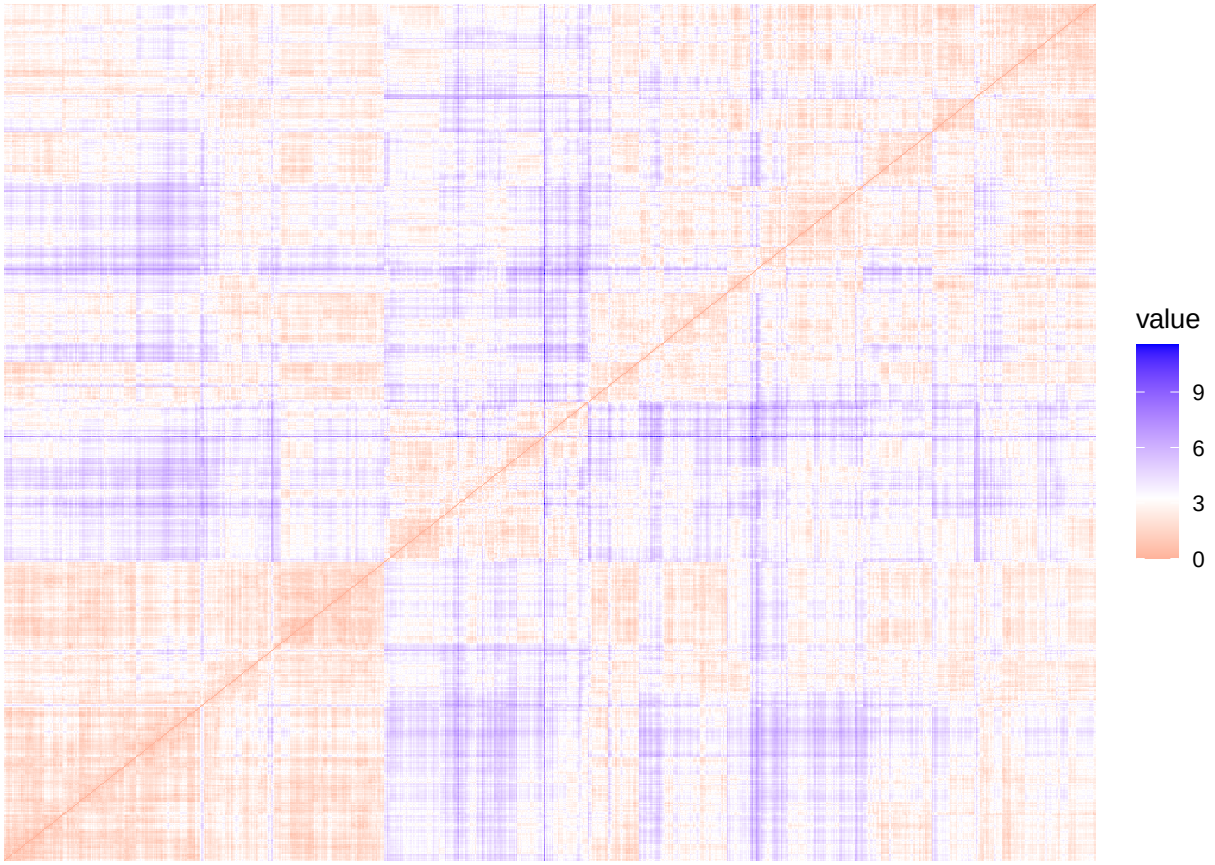


Figure 30: Heat Map of Heart Disease Distance Matrix

The heatmap displays the distance matrix for our heart disease dataset, visualizing the pairwise distances between data points using the Euclidean metric. A quick glance at the heatmap reveals potential clusters as blocks of similar colors. We observe several distinct blocks along the diagonal, where the colors change from light to dark shades. This suggests potential clusters within our dataset. In this cluster there seems to be two prominent clusters.

Hierarchical Clustering

Hierarchical clustering is an algorithm that builds a hierarchy of clusters by progressively merging or splitting existing groups.

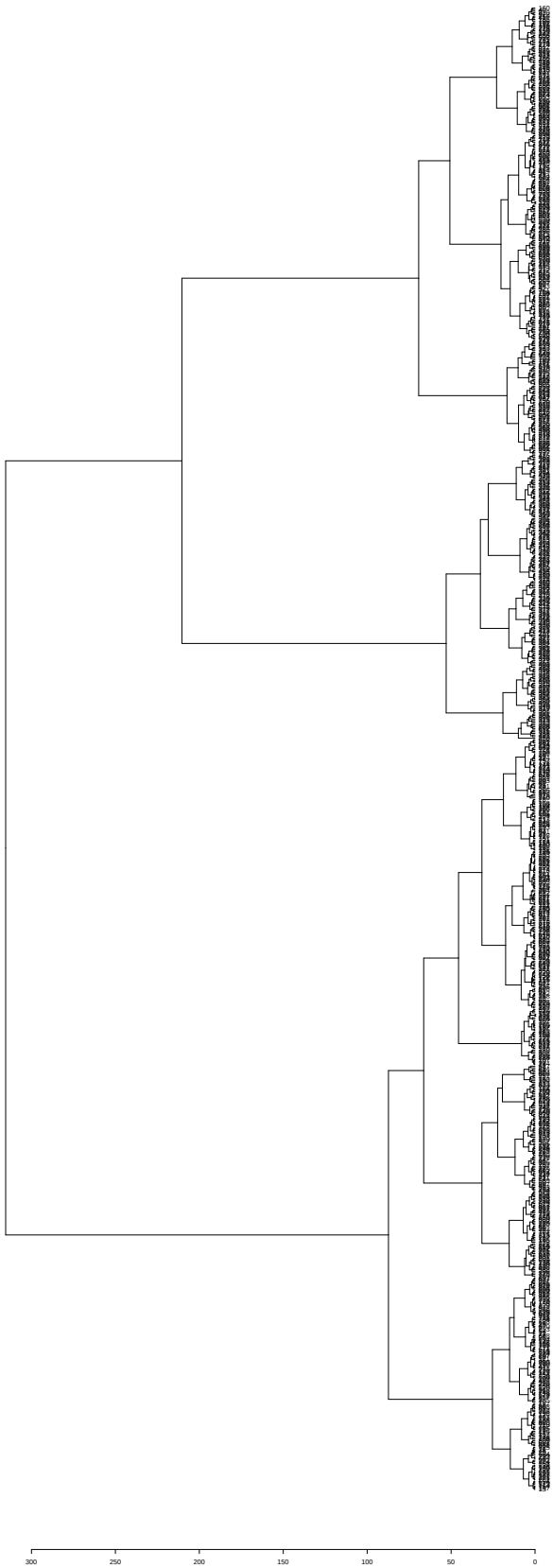


Figure 31: Dendrogram Visualization of Hierarchical Clustering

In this dendrogram, we see a complex structure with many small branches, indicating a dataset with intricate relationships and potential subgroups. The horizontal axis represents the distance or dissimilarity between clusters, with larger values indicating less similarity.

Non-hierarchical Clustering

In non-hierarchical clustering, we aim to partition the dataset into a set number of clusters.

Figure 32: k-Means Cluster Plot

From the plot, it's apparent that there are three distinct groups within the dataset, each represented by different symbols and colors. These clusters could be indicative of underlying patterns within the data. The plot shows the spread of the data points across the two principal dimensions, marked as Dim1 and Dim2. Each cluster is also enveloped in an ellipsis that represents the boundary of the cluster. Also very noticeably we can see that the third group is takes over the plot indicating a very large spreadout cluster.

Tuning Parameters

Scree Plot Using a scree plot to we identify the ideal number of clusters via the k-means algorithm applied to the coreHeart dataset. A scree plot serves as a visual aid in clustering analysis, illustrating how data variation is accounted for across various numbers of clusters (K values). Usually, the x-axis depicts the number of clusters (K), while the y-axis presents a metric assessing the quality of clustering.

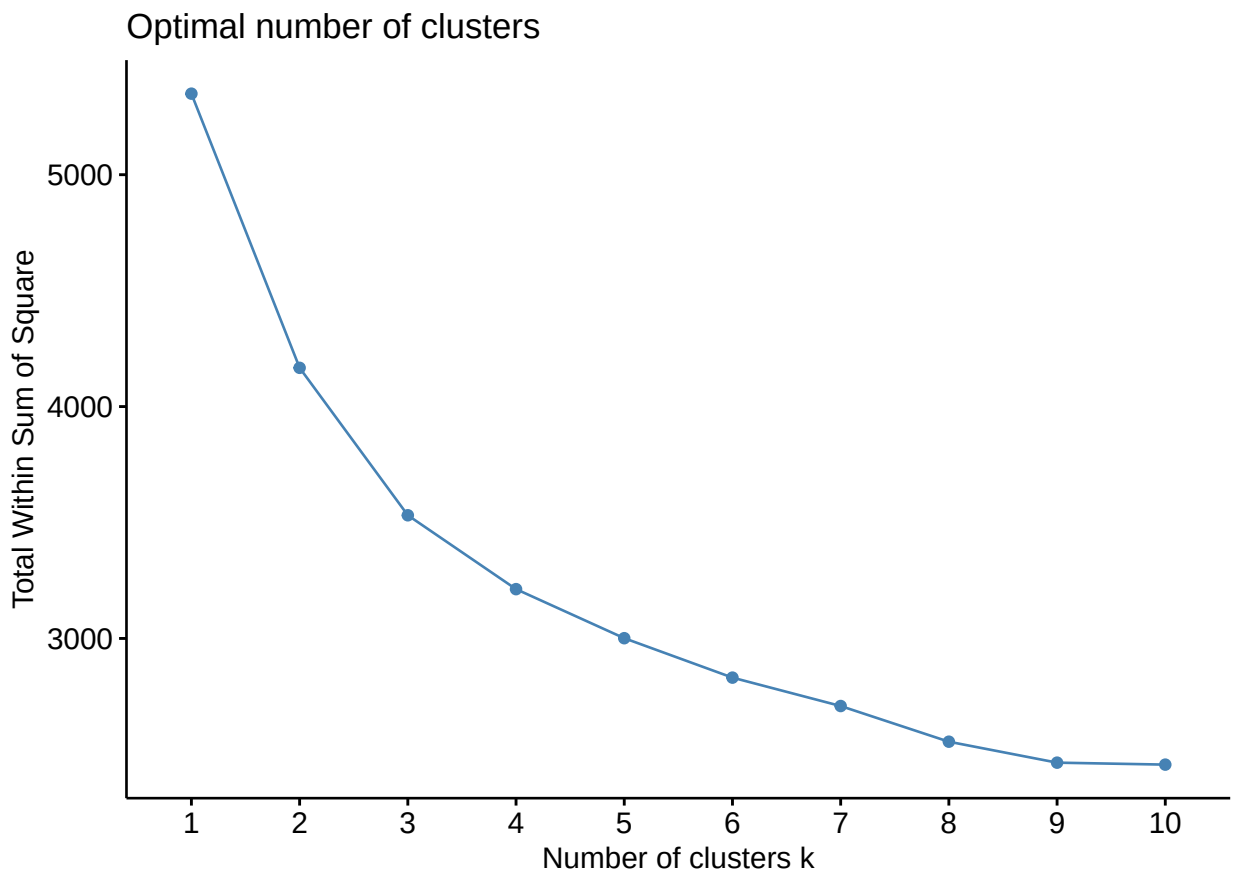


Figure 33: Scree Plot for Choosing Optimal Number of Clusters

The scree plot (Figure X) showcases a steep decline from one to two clusters, using the 'elbow method', a way for determining the number of clusters in a data set.In our analysis, the elbow appears to be at k = 3, after which the decline in WSS becomes less pronounced.

Silhouette Method Average Silhouette method is a means to validate the consistency within clusters of data. It displays a measure of how close each point in one cluster is to points in the neighboring clusters

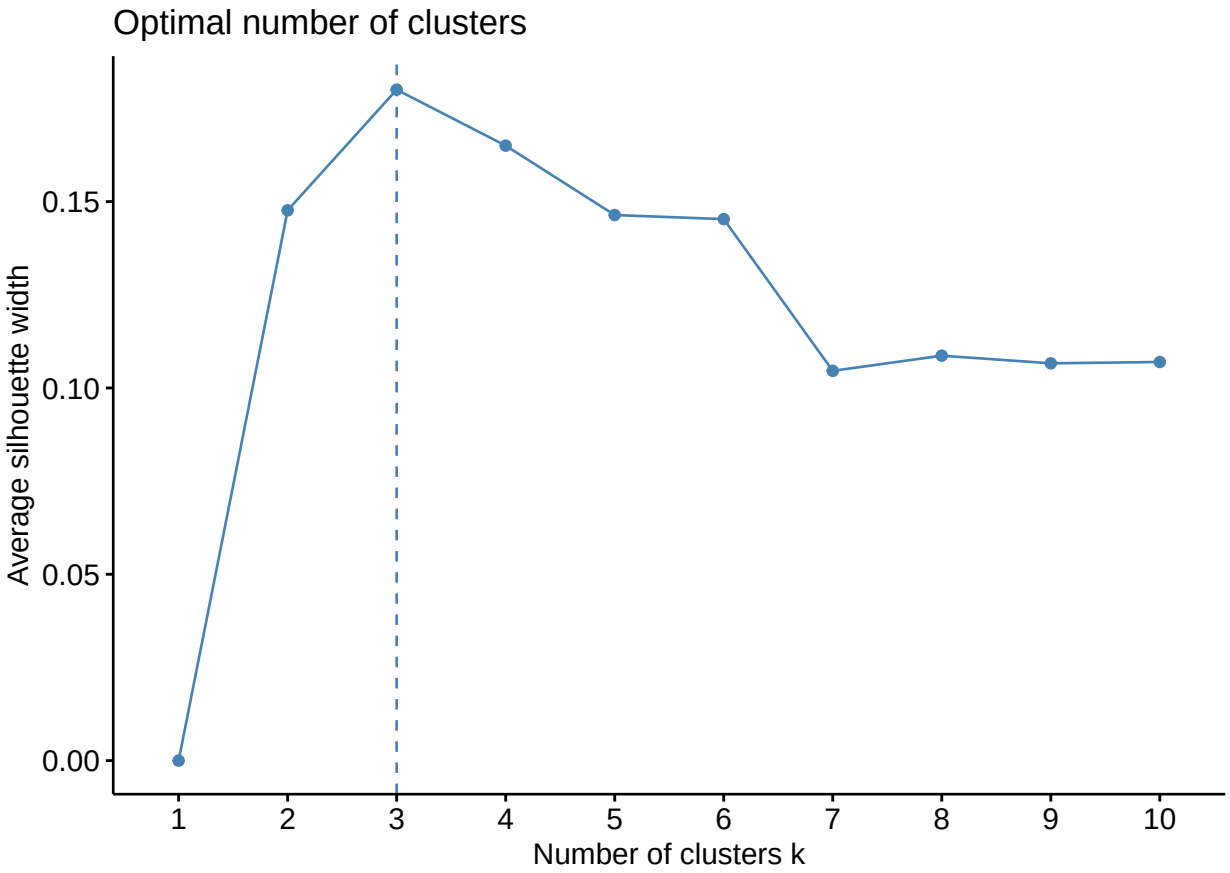


Figure 34: Silhouette Method for Clustering

The silhouette plot displays blue vertical dashed line indicating the average silhouette width for each number of clusters. For the plot above the average silhouette width is 3

Gap Statistics The gap statistic is a method for estimating the number of clusters in a set of data. The gap statistic plot below shows the x-axis which represents the number of clusters (k), and the y-axis which represents the gap statistic values. Each point on the plot is the gap statistic value for a particular number of clusters

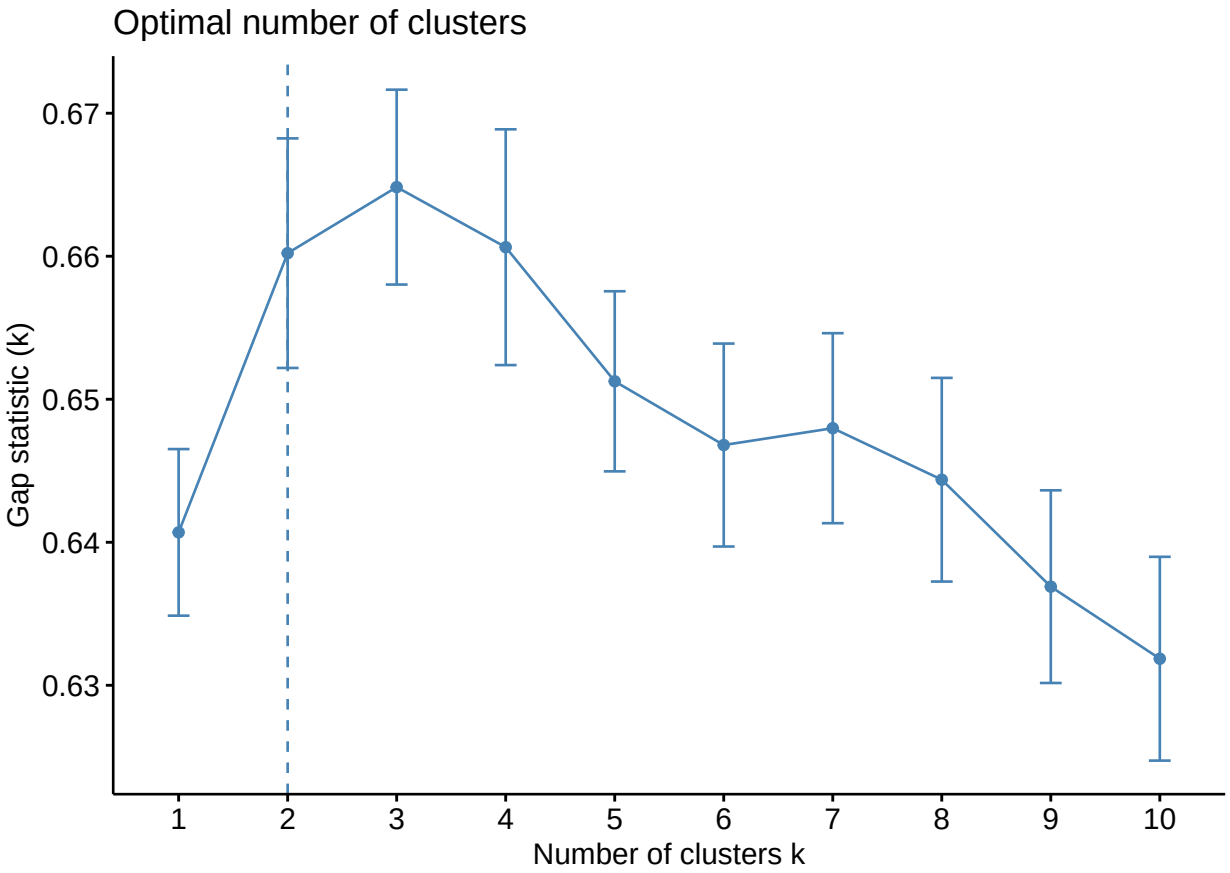


Figure 35: Gap Statistic Plot 1 for Determining Number of Clusters

The optimal number of clusters is typically determined by the first significant local maximum of the gap statistic or the smallest number of clusters such that the gap statistic falls within one standard error of its maximum.The visulisation shows that k is 2

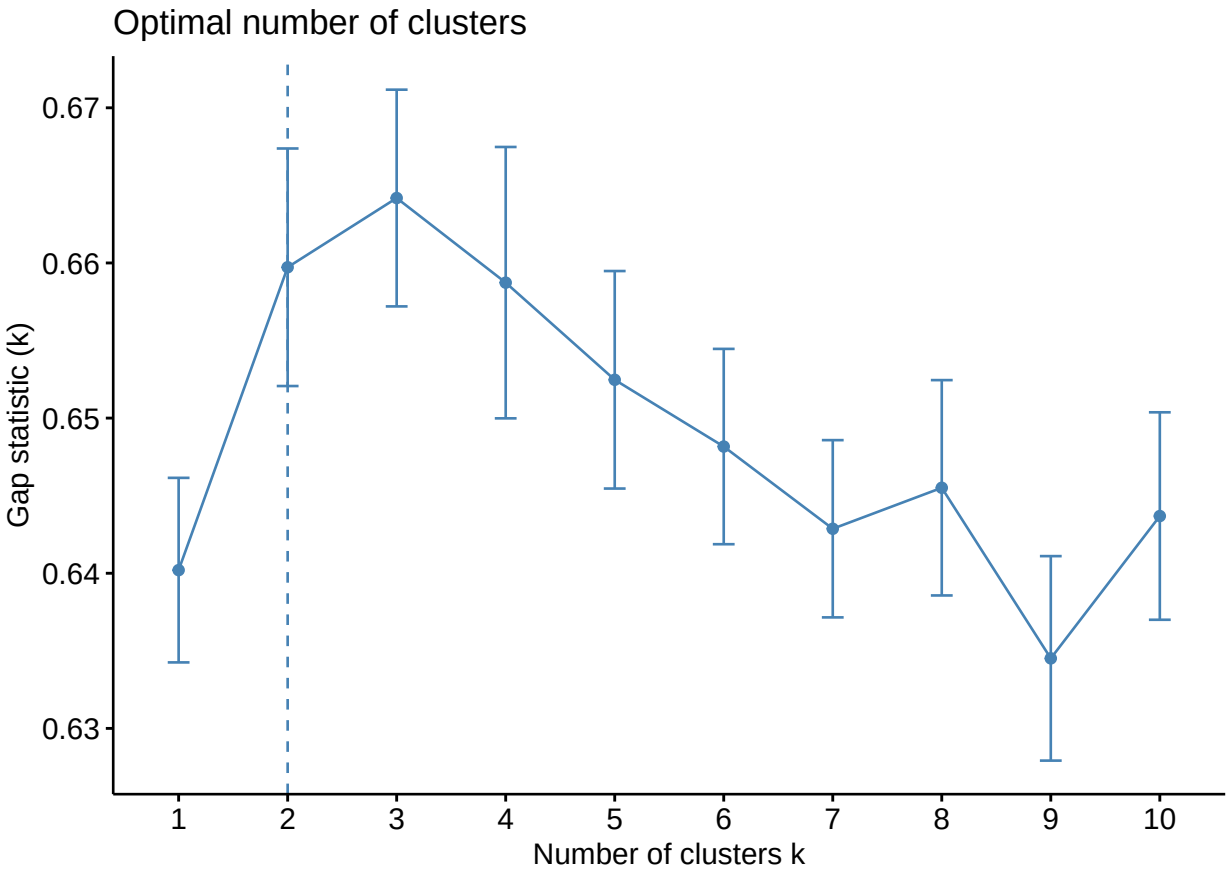


Figure 36: Gap Statistic Plot 2 for Determining Number of Clusters

Like the other Gap statistiv The plot suggests that after k=2, the increase in the gap statistic begins to diminish, indicating that additional clusters may not be significant.

Looking at the four visualization they either point to 3 clusters or 2 clusters. So we are going to investigate both on them for the remainder of analysis.

Hierarchical: Cut the Dendrogram using K value

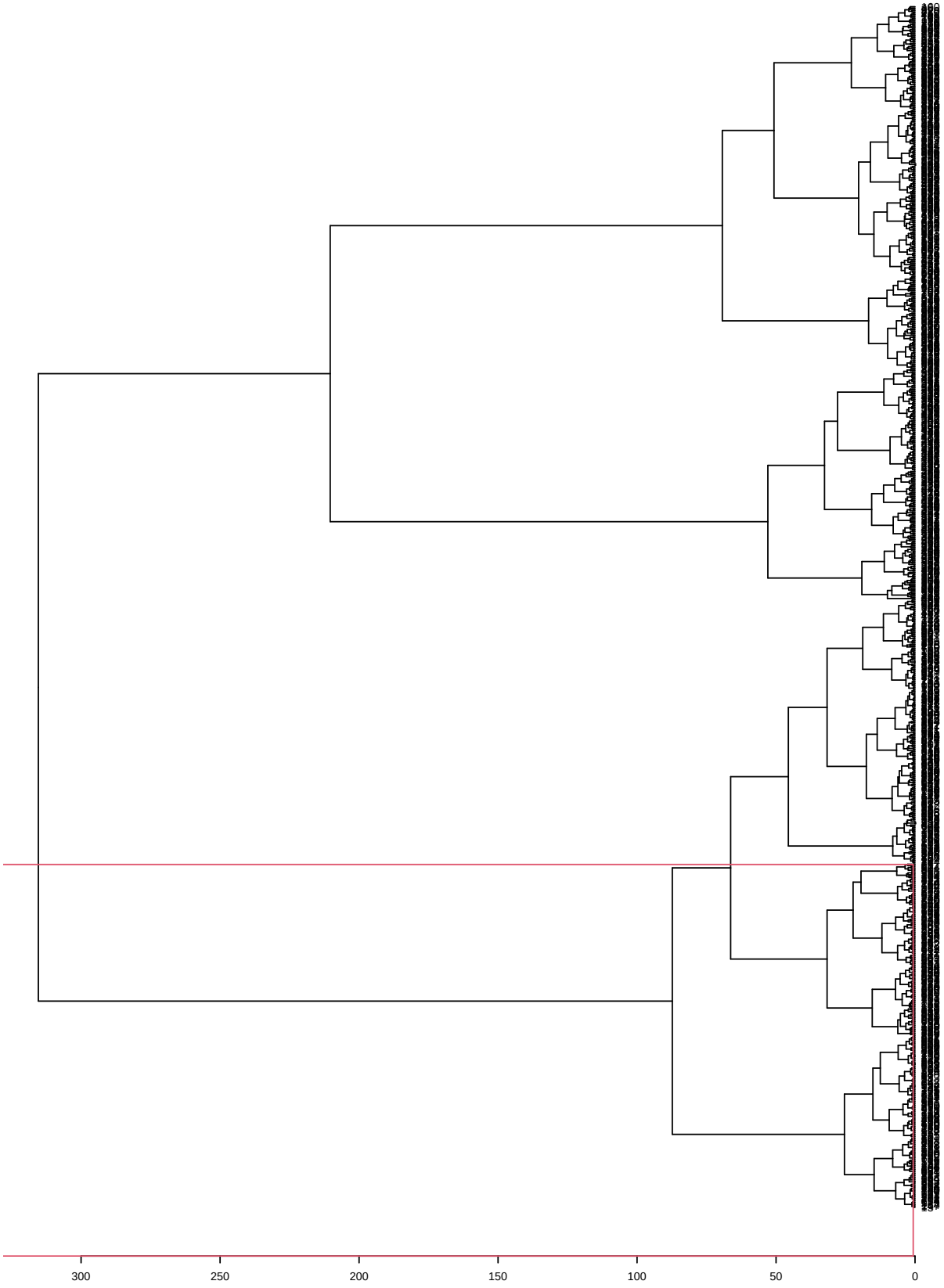


Figure 37: Dendrogram Visualization K = 2

In the dendrogram above, we illustrate the hierarchical clustering of our heart disease dataset. The tree-like structure allows us to observe the natural grouping of data points based on their similarity. Observations are represented as leaves, and the branches denote the points at which clusters are combined. For this we are cutting the tree at k = 2, which would create two clusters, indicated by the horizontal red line.

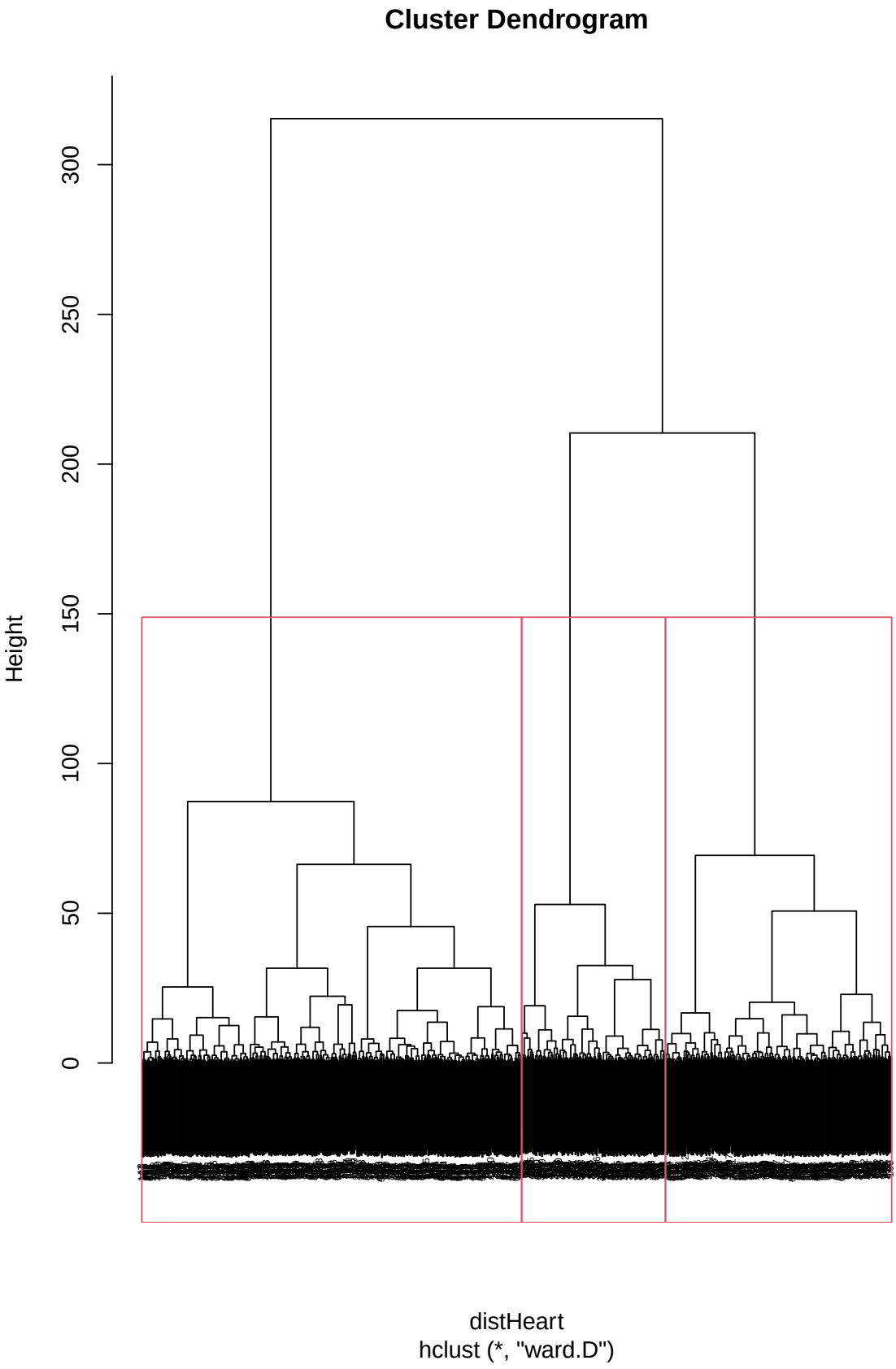
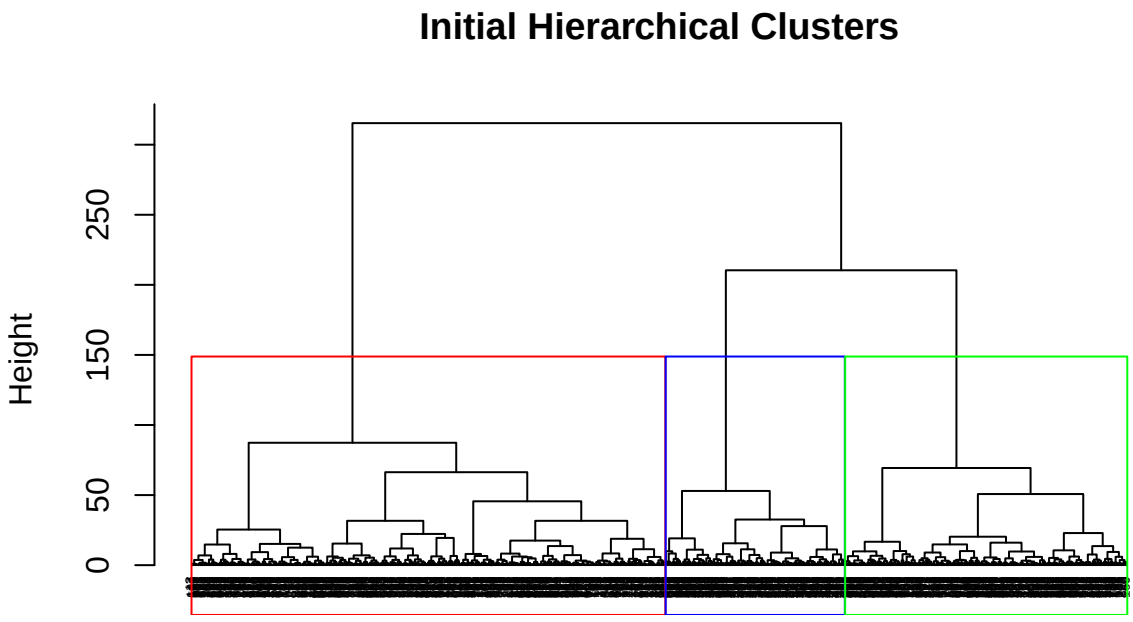


Figure 38: Dendrogram Visualization K = 3

For this dendrogram we have chosen to cut the tree at k = 3, as indicated by the horizontal red lines, which will give us three distinct groups. The height of the merges suggests the distance at which these groups come together, with larger heights representing greater dissimilarity

k-Means: Re-Run the Algorithm

Interweaving Hierarchical and Nonhierarchical Approaches In our clustering analysis, we decided to use a hybrid approach that intertwines hierarchical and nonhierarchical (k-means) clustering techniques. This strategy leverages the strengths of both methods



The dendrogram shows the initial hierarchical clustering with colored rectangles highlighting the clusters identified by the hybrid approach. Rectangles drawn at different heights indicate potential cluster separations, with red, blue, and green indicating different clusters.

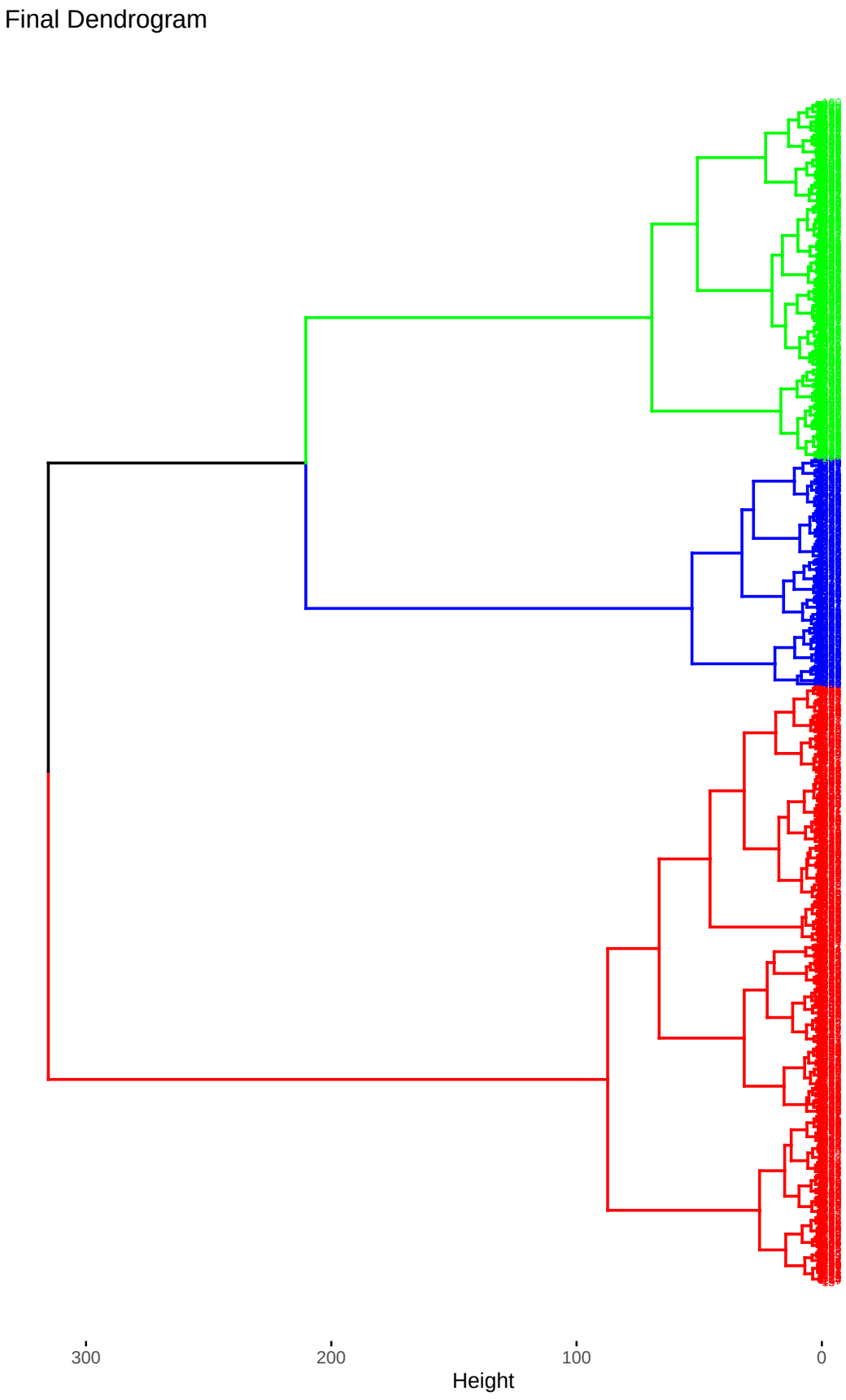


Figure 39: Final Dendrogram Visualization

For our final dendrogram the different colors signify the distinction between clusters, making it easier to visually differentiate between them. Red clusters are likely those that are the most different from others, as indicated by the greater height at which they merge with other clusters. The green and blue clusters represent groups of that are more similar to each other, as they merge at lower heights.

Comparing Modeling Approaches (Logistic Regression, Decision Tree and Clustering)

- `Logistic Regression`: The Logistic regression has stood out the most for its interpretability. We constructed two logistic regression models, each with distinct sets of predictors. Model 1 was a simple logistic regression focusing on 'Oldpeak' as a sole predictor, while Model 2 was a multiple logistic regression model derived from a stepwise selection process.

Model 2's performance was superior, as evidenced by higher accuracy, sensitivity (true positive rate), and specificity (true negative rate) scores.The Area Under the Curve (AUC) from ROC curve analysis was also notably better for Model 2, indicating a greater ability to distinguish between the presence and absence of heart disease.

The variables retained in Model 2, such as ST Slope and chest pain type, are clinically relevant indicators of heart disease, validating the model's practical utility. In summary, logistic regression has proven to be a valuable tool in the analysis of heart disease data.

- Logistic regression Overall Accuracy 89% with a sensitivity of 85% and a specificity of 91% *
- `Decision Tree`: We employed the CART (Classification and Regression Trees) algorithm to develop our decision tree model. The algorithm recursively splits the data into subsets that are as distinct as possible concerning the target variable, which, in our case, was the presence or absence of heart disease. The initial tree, without any constraints, was complex and prone to overfitting. To address this, we pruned the tree using cost complexity pruning, which simplified the model by penalizing overly complex trees.

One of the key strengths of decision trees is their visual interpretability. The tree can be plotted, and each node's decisions can be examined to understand the model's classification rules.

Compared to logistic regression, decision trees handle non-linear relationships between predictors and the outcome variable more naturally.Uunlike logistic regression which outputs probabilities, decision trees classify patients by splitting the dataset into homogenous subsets based on a series of decisions. However, they can be less stable, as small changes in the data can lead to different splits, and hence, a different tree structure.In summary, decision trees serve as an excellent tool for initial data exploration and hypothesis generation.

- Decision Tree overall Accuracy was 82.1% with a sensitivity of approximately 73.3% and a specificity of approximately 78.9%. *
- Clustering Analysis : Clustering techniques, such as K-means and hierarchical clustering, provide an unsupervised learning approach that can uncover structures within the heart disease dataset. Unlike logistic regression and decision trees, which are supervised and predict an outcome based on input variables, clustering groups data points based on feature similarity without predefined labels.

Hierarchical vs. Non-Hierarchical Clustering Our hierarchical clustering analysis revealed detailed data structures, suggesting diverse subgroups within the heart disease. Non-hierarchical clustering, particularly k-means, gave us clear groupings, allowing us to discern distinct patterns in heart disease.

Hybrid Approach This approach provided a comprehensive picture, for balancing both hierarchical clustering with the precision of k-means. This technique helps solidify our understanding of the natural groupings in the data.

Clustering can also inform feature engineering for supervised learning and provide exploratory insights that could lead to new hypotheses about heart disease.In summary, clustering provides a valuable unsupervised approach to uncover hidden structures within heart disease data.

Discussion and Limitations

Logistic Regression excelled in identifying significant predictors and their respective odds ratios, offering a clear interpretation of risk factors. While the decision tree approach provided an intuitive visualization of the decision-making process, highlighting the hierarchical importance of variables. As for clustering it revealed the underlying structure in the data without the guidance of a dependent variable. It helped to identify natural groupings within the patient data that could correspond to different subtypes of heart disease or risk profiles

Limitations for this study firstly because of the data set itself because it wasnt a very big dataset so it didnt provide enough of information that could be needed to model a better analysis for heart disease. Also when doing the clutering it was hard to visualisation the the Dendrogram despite increasing size and changing it view.

Overall though Logistic regression was able to predict more than the decision tree. And comparatively, logistic regression and decision trees offer more direct applicability to prediction tasks in heart disease. Chosen modeling approach should align with the specific objectives of the analysis, whether it be prediction, interpretation, or exploratory analysis. Each method has its strengths and is most powerful when used in comparison with others.

References and Materials Consulted

Fedesoriano. (2021a, September 10). Heart failure prediction dataset. Kaggle. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

U.S. National Library of Medicine. (n.d.). National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/

Author Contributions

The authors of this report would like to acknowledge their individual contributions to the report. Both authors contributed to ongoing discussions about study design and analysis.

- Olachi Mbakwe contributed to the Introduction and Background, Exploration of the Data,Comparing the results, Discussion and Limitations, coding, and writing of the report.
- Rebande Olusesi contributed to the Logistic Regression Section, Coding, and Writing of the report. Discussions and Limitations, coding, and writing of the report.
- Justin Constant contributed to the Clustering Section, Coding, and Writing of the report.
- Evan Settipane to the CART Section(growing tree), Coding, and Writing of the report.

Code Appendix

```
knitr::opts_chunk$set(
  echo = FALSE,
  warning = FALSE,
  message = FALSE,
  fig.align = "center",
```

```
    cache = TRUE
)


# Packages to install ----
## tree, rpart, rpart.plot, partykit, rattle,factoextra and randomForest
#install.packages(
 #  pkgs = c("tree", "rpart", "rpart.plot", "partykit", "rattle", "randomForest","factoextra","cluster"


# Load packages used in this guide ----
packages <- c("readr","dplyr","tidyverse", "knitr", "kableExtra","psych",
              "janitor","tree", "rpart", "rpart.plot", "partykit",
              "rattle", "randomForest", "yardstick","leaps",
              "ggplot2","pROC","separationplot","factoextra","corrplot","vcd","cluster")


invisible(
  lapply(
    X = packages,
    FUN = library,
    character.only = TRUE,
    quietly = TRUE
  )
)

# Set the CRAN mirror
options(repos = "https://cran.rstudio.com/")

# Set Table Option ----
options(knitr.kable.NA = "")


heart <- read_csv("heart.csv")

heart$ChestPainType <- factor(
x = heart$ChestPainType,
levels = c("ATA", "ASY", "NAP","TA")
)

heart$RestingECG <- factor(
x = heart$RestingECG,
levels = c("Normal", "ST","LVH")
)

heart$ST_Slope <- factor(
x = heart$ST_Slope,
levels = c("Up", "Flat", "Down")
)

heart$Sex <- factor(
x = heart$Sex,
levels = c("M", "F")
```

```
)
heart_stats <- psych::describe(heart[, sapply(heart, is.numeric)],
                               na.rm = TRUE,
                               quant = c(0.25, 0.75))


# Convert to data frame
heart_stats <- as.data.frame(heart_stats)

heart_stats$Attribute <- rownames(heart_stats)

# Gets rid of first row
heart_stats <- heart_stats[,-1]

# Makes Table
as.data.frame(heart_stats) %>%
  dplyr::select(
    Attribute, n, min, Q0.25, median, Q0.75, max, mad, mean, sd, skew, kurtosis
  ) %>%
  kable(
    digits = 2,
    booktabs = TRUE,
    row.name = FALSE,
    col.names = c("Variables","n","Samp.Min", "Q1", "Samp.Med","Q3", "Samp.Max","MAD","SAM","SASD","Skew
    align ="lccccccccccc",
    format.args = list(big.mark = ","),
    caption = "Statistics"
  ) %>%
  kable_classic(
    latex_options = c("HOLD_position", "scale_down"),
    "striped",
    full_width = FALSE,
    position = "center"
  ) %>%
  add_header_above() %>%
  footnote(
    general = "Descriptive statistics calculated for numeric variables in the dataset.",
    footnote_as_chunk = TRUE
  )


heart_data <- heart

heart_data$HeartDisease <- ifelse(heart_data$HeartDisease == 1, "Yes", "No")

ggplot(heart_data, aes(x = Age)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  labs(title = "Age Distribution", x = "Age", y = "Frequency") +
  theme_minimal()

ggplot(heart_data, aes(x = Sex)) +
  geom_bar(fill = c("blue", "brown"), color = "black") +
  labs(title = "Distribution of Sex", x = "Sex", y = "Count") +
```

```
  theme_minimal()

ggplot(heart_data, aes(x = RestingECG)) +
  geom_bar(fill = c("lightblue", "peachpuff", "orchid"), color = "black") +
  labs(title = "Distribution of Resting ECG", x = "Resting ECG", y = "Count") +
  theme_minimal()

ggplot(heart_data, aes(x = ChestPainType)) +
  geom_bar(fill = c("maroon", "green4", "dark blue", "gold3"), color = "black") +
  labs(title = "Distribution of Chest Pain Type", x = "Chest Pain Type", y = "") +
  theme_minimal()


ggplot(heart_data, aes(x = ExerciseAngina)) +
  geom_bar(fill = c("firebrick", "darkolivegreen"), color = "black") +
  labs(title = "Distribution of Exercise Angina", x = "Exercise Angina", y = "Count") +
  theme_minimal()

ggplot(heart_data, aes(x = ST_Slope)) +
  geom_bar(fill = c("dodgerblue3", "orangered3", "pink4"), color = "black") +
  labs(title = "Distribution of ST Slope", x = "ST Slope", y = "Count") +
  theme_minimal()
ggplot(heart_data, aes(x = HeartDisease, fill = HeartDisease)) +
  geom_bar(color = "black") +
  scale_fill_manual(values = c("dark green", "brown")) +
  labs(title = "Distribution of Heart Disease", x = "Heart Disease", y = "Count", fill = "Heart Disease
  theme_minimal()
ggplot(heart_data) +
  aes(x = HeartDisease, y = Age) +
  geom_boxplot(fill = c("dark green", "brown")) +
  labs(title = "Heart Disease by Age", x = "Heart Disease Status", y = "Age") +
  theme_minimal()
ggplot(heart_data, aes(x = Sex, fill = HeartDisease)) +
  geom_bar(position = "dodge", color = "black") +
  labs(title = "Heart Disease by Sex", x = "Sex", y = "Count", fill = "Heart Disease") +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()
ggplot(heart_data) +
  aes(x = ExerciseAngina, fill = HeartDisease) +
  geom_bar(color = "black") +
  labs(title = "Heart Disease by Exercise Angina", x = "Exercise Angina", y = "Count", fill = "Heart Di
  scale_fill_hue(direction = 1) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()


ggplot(heart_data) +
  aes(x = MaxHR, fill = HeartDisease) +
  geom_boxplot(color = "black") +
  labs(title = "Heart Disease by MaxHR", x = "Max Heart Rate", y = "Count", fill = "Heart Disease") +
  scale_fill_hue(direction = 1) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()
```

```r
ggplot(heart_data) +
  aes(x = Oldpeak, fill = HeartDisease) +
  geom_boxplot() +
  labs(title = "Heart Disease by Oldpeak", x = "Oldpeak", y = "Count", fill = "Heart Disease") +
  scale_fill_hue(direction = 1) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()


ggplot(heart_data) +
  aes(x = ST_Slope, fill = HeartDisease) +
  geom_bar(color = "black") +
  labs(title = "Heart Disease by ST Slope", x = "ST Slope", y = "Count", fill = "Heart Disease") +
  scale_fill_hue(direction = 1) +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()

# Assuming 'heart' data frame is already loaded and HeartDisease is converted to a factor
chest_pain_counts <- heart_data %>%
  group_by(ChestPainType, HeartDisease) %>%
  summarise(Count = n()) %>%
  ungroup()

ggplot(chest_pain_counts, aes(x = ChestPainType, y = Count, group = HeartDisease, color = HeartDisease))
  geom_line() +
  geom_point() +
  labs(title = "Frequency of Heart Disease by Chest Pain Type", x = "Chest Pain Type", y = "Frequency")
  scale_color_manual(values = c("No" = "blue", "Yes" = "orangered")) +
  theme_minimal()

ggplot(heart_data, aes(x = RestingECG, fill = HeartDisease)) +
  geom_bar(position = "dodge", color = "black") +
  labs(title = "Heart Disease by Resting ECG ", x = "Resting ECG", y = "Count", fill = "Heart Disease")
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()

ggplot(heart_data, aes(x = ExerciseAngina, fill = HeartDisease)) +
  geom_bar(position = "dodge", color = "black") +
  labs(title = "Heart Disease by Exercise Angina", x = "Exercise Angina", y = "Count") +
  scale_fill_manual(values = c("No" = "blue", "Yes" = "orangered2"))+
  theme_minimal()

ggplot(heart_data) +
  aes(x = HeartDisease, y = Cholesterol) +
  geom_boxplot(fill = c("No" = "blue", "Yes" = "orangered2")) +
  labs(title = "Heart Disease by Cholesterol", x = "Heart Disease", y = "Cholesterol") +
  theme_minimal()

ggplot(heart_data, aes(x = HeartDisease, y = RestingBP)) +
  geom_boxplot(fill = c("No" = "blue", "Yes" = "orangered2")) +
  labs(title = "Heart Disease by Resting Blood Pressure", x = "Heart Disease", y = "Resting Blood Pressu
  theme_minimal()
```

```r
heart_data_num <- heart
heart_data_num$ExerciseAngina <- ifelse(heart_data_num$ExerciseAngina == "Y", 1, 0)
heart_data_num$HeartDisease <- as.numeric(heart_data_num$HeartDisease)

# Compute correlation matrix
cor_matrix <- cor(heart_data_num[, c("Age", "RestingBP", "Cholesterol", "MaxHR", "Oldpeak", "FastingBS"

# Visualize the correlation matrix
corrplot(cor_matrix, method = "color")

# Matrix of counts
counts <- table(heart_data$RestingECG, heart_data$ChestPainType)

# Table to a data frame
counts_df <- as.data.frame(counts)
names(counts_df) <- c("RestingECG", "ChestPainType", "Freq")

ggplot(counts_df, aes(x = RestingECG, y = ChestPainType, fill = Freq)) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red") +
  geom_text(aes(label = Freq), vjust = 1) +
  labs(title = "Heatmap of RestingECG and ChestPainType") +
  theme_minimal()



HeartData <- heart %>%
  drop_na() %>%
  mutate(
    tempID = row_number(),
    .before = Age
  )

set.seed(380)
trainingData <- HeartData %>%
  slice_sample(prop = 0.8)

trainingResults <- trainingData

testingData <- HeartData %>%
  filter(!(tempID %in% trainingData$tempID))
testingResults <- testingData

testingResults$HeartDisease <- factor(testingResults$HeartDisease, levels = c(1, 0), labels = c("Heart

trainingResults$HeartDisease <- factor(trainingResults$HeartDisease, levels = c(1, 0), labels = c("Hear

# Form Candidate Model 1
HeartlogModel1 <- glm(
  formula = HeartDisease ~ Oldpeak,
  data = trainingData,
  family = binomial
)
```

```
# Form Candidate Model 2 ----
## Lower bound
### Intercept only
lower <- glm(
  formula = HeartDisease ~ 1,
  data = trainingData,
  family = binomial
)
## Upper bound

upper <- glm(
  formula = HeartDisease ~  Age + Sex + ChestPainType + RestingBP + Cholesterol + FastingBS + RestingEC
  data = trainingData,
  family = binomial
)

## Stepwise search
Heartlogmodel2 <- step(
  object = lower,
  scope = list(
    lower = lower,
    upper = upper
  ),
  data = trainingData,
  direction = "both",
  k = 2,
  trace = 0
)
# Model 1 Coefficient Table ----
as.data.frame(summary(HeartlogModel1)$coefficients) %>%
  rownames_to_column(var = "term") %>%
  rename(coefficient = Estimate) %>%
  mutate(
    prob_odds = case_when(
      coefficient == "(Intercept)" ~ exp(coefficient)/(1 + exp(coefficient)),
      .default = exp(coefficient)
    ),
    .after = coefficient
  ) %>%
  mutate(
    `Pr(>|z|)` = ifelse(
      test = `Pr(>|z|)` < 0.001,
      yes = paste("< 0.001"),
      no = `Pr(>|z|)`
    ),
    term = case_when(
      term == "(Intercept)" ~ "Intercept",
      grepl(x = term, pattern = "Oldpeak") ~ "Oldpeak",
      .default = term
    )
  ) %>%
  kable(
    digits = 3,
```

```
    booktabs = TRUE,
    align = c("l", rep("c", 5)),
    col.names = c("Term", "Coefficient", "Prob./Odds Ratio",
                  "Std. Err.", "Z", "p-value"),
    table.attr = 'data-quarto-disable-processing="true"'
  ) %>%
  kable_classic(
    position = "center",
    latex_options = c("HOLD_position"),
    full_width = FALSE
  )%>%
  add_header_above() %>%
  footnote(
    general = "Logistic Model 1 Coefficient Table",
    footnote_as_chunk = TRUE
  )

# Building confidence intervals for Model 1 coefficients ----
Heartlogmodel1CI <- confint(
  object = HeartlogModel1,
  parm = "Oldpeak",
  level = 0.9
)
# Stored fitted values for Model 1 ----
trainingResults$Heartlogmodel1Pred <- predict(
  object = HeartlogModel1,
  newdata = trainingData,
  type = "response"
)

# Apply naïve rule ----
trainingResults <- trainingResults %>%
  mutate(
    Heartlogmodel1Class = case_when(
      Heartlogmodel1Pred > 0.5 ~ "Heart Disease",
      TRUE ~ "No Heart Disease"
    )
  )

# Build Confusion Matrix for Model 1 ----
trainingResults %>%
  tabyl(var1 = Heartlogmodel1Class, var2 = HeartDisease) %>%
  adorn_title(
    placement = "combined",
    row_name = "Predicted",
    col_name = "Actual"
  ) %>%
  kable(
    booktabs = TRUE,
    align = "c",
    table.attr = 'data-quarto-disable-processing="true"'
  ) %>%
  kable_classic(
```

```
    position = "center",
    latex_options = c("HOLD_position"),
    full_width = FALSE
  )

# Model 2 Coefficient Table ----
as.data.frame(summary(Heartlogmodel2)$coefficients) %>%
  rownames_to_column(var = "term") %>%
  rename(coefficient = Estimate) %>%
  mutate(
    prob_odds = case_when(
      coefficient == "(Intercept)" ~ exp(coefficient)/(1 + exp(coefficient)),
      .default = exp(coefficient)
    ),
    .after = coefficient
  ) %>%
  mutate(
    `Pr(>|z|)` = ifelse(
      test = `Pr(>|z|)` < 0.001,
      yes = paste("< 0.001"),
      no = round(`Pr(>|z|)`, 3)
    ),
    term = case_when(
      term == "(Intercept)" ~ "Intercept",
      grepl(x = term, pattern = "ST_SlopeFlat") ~ "ST Slope: Flat",
      grepl(x = term, pattern = "ST_SlopeDown") ~ "ST Slope: Down",
      grepl(x = term, pattern = "ChestPainTypeASY") ~ "Chest pain type: Asymptomatic",
      grepl(x = term, pattern = "ChestPainTypeNAP" ) ~ "Chest pain type: Non-Anginal Pain",
      grepl(x = term, pattern = "ChestPainTypeTA" ) ~ "Chest pain type: Typical Angina",
      grepl(x = term, pattern = "SexF" ) ~ "Sex: Female",
      grepl(x = term, pattern = "FastingBS") ~ "Fasting Blood Sugar",
      grepl(x = term, pattern = "Oldpeak") ~ "Oldpeak",
      grepl(x = term, pattern = "ExerciseAnginaY") ~ "Exercise-induced angina: Yes",
      grepl(x = term, pattern = "Cholesterol") ~ "Cholesterol",
      grepl(x = term, pattern = "Age") ~ "Age",
      .default = term
    )
  ) %>%
  kable(
    digits = 3,
    booktabs = TRUE,
    align = "lccccc",
    col.names = c("Term", "Coefficient", "Prob./Odds Ratio",
                  "Std. Err.", "Z", "p-value")
  ) %>%
  kableExtra::kable_classic(
    position = "center",
    font_size = 10,
    latex_options = "HOLD_position",
    full_width = FALSE
  )%>%
  add_header_above() %>%
  footnote(
```

```
    general = "Logistic Model 2 Coefficient Table",
    footnote_as_chunk = TRUE
  )

# Build Tukey-Anscombe plot for Model 2 ----
ggplot(
  data = data.frame(
    residuals = residuals(Heartlogmodel2, type = "pearson"),
    fitted = fitted(Heartlogmodel2)
  ),
  mapping = aes(x = fitted, y = residuals)
) +
  geom_point() +
  geom_smooth(
    formula = y ~ x,
    method = stats::loess,
    method.args = list(degree = 1),
    se = FALSE,
    linewidth = 0.5
  ) +
  theme_bw() +
  labs(
    x = "Fitted",
    y = "Pearson Residuals",
    title = "Tukey-Anscombe Plot for Model 2"
  )
# Find GVIF for Model 2 and build a table ----
as.data.frame(car::vif(Heartlogmodel2)) %>%
  rownames_to_column(var = "term") %>%
  mutate( squared = `GVIF^(1/(2*Df))`^2) %>%
  kable(
    digits = 3,
    align = "lcccc",
    booktab = TRUE,
    format.args = list(big.mark = ",")
  ) %>%
  kable_classic(
    position = "center",
    latex_options = c("HOLD_position"),
    full_width = FALSE
  )
# Stored fitted values for Model 2 ----
trainingResults$Heartlogmodel2Pred <- predict(
  object = Heartlogmodel2,
  newdata = trainingData,
  type = "response"
)

# Apply naïve rule ----
trainingResults <- trainingResults %>%
  mutate(
    Heartlogmodel2Class = case_when(
      Heartlogmodel2Pred > 0.5 ~ "Heart Disease",
```

```
      TRUE ~ "No Heart Disease"
    )
  )

# Build Confusion Matrix for Model 2 ----
trainingResults %>%
  tabyl(var1 = Heartlogmodel2Class, var2 = HeartDisease) %>%
  adorn_title(
    placement = "combined",
    row_name = "Predicted",
    col_name = "Actual"
  ) %>%
  kable(
    booktabs = TRUE,
    align = "c"
  ) %>%
  kable_classic(
    position = "center",
    latex_options = c("HOLD_position"),
    full_width = FALSE
  )
# Fit ROC Curves ----
## Model 1
Heartlogmodel1ROC <- roc(
  formula = HeartDisease ~ Heartlogmodel1Pred,
  data = trainingResults
)
Heartlogmodel1ROC_df <- data.frame(
  threshold = Heartlogmodel1ROC$thresholds,
  sensitivity = Heartlogmodel1ROC$sensitivities,
  specificity = Heartlogmodel1ROC$specificities,
  model = "Model 1"
)

## Model 2
Heartlogmodel2ROC <- roc(
  formula = HeartDisease ~ Heartlogmodel2Pred,
  data = trainingResults
)
Heartlogmodel2ROC_df <- data.frame(
  threshold = Heartlogmodel2ROC$thresholds,
  sensitivity = Heartlogmodel2ROC$sensitivities,
  specificity = Heartlogmodel2ROC$specificities,
  model = "Model 2"
)

## Merge into one data frame
rocData <- rbind(Heartlogmodel1ROC_df, Heartlogmodel2ROC_df)

## AUC Data
aucData <- data.frame(
  model = c("Model 1", "Model 2"),
  auc = c(Heartlogmodel1ROC$auc, Heartlogmodel2ROC$auc)
```

```
)

# Make ROC Plot ----
ggplot(
  data = rocData,
  mapping = aes(x = 1 - specificity, y = sensitivity, color = model)
) +
  geom_path() +
  geom_abline(
    slope = 1,
    intercept = 0,
    linetype = "dotted"
  ) +
  geom_text(
    inherit.aes = FALSE,
    data = aucData,
    mapping = aes(label = paste(model, "AUC: \n", round(auc, 3))),
    x = c(0.25, 0.25),
    y = c(0.4, 0.9)
  ) +
  theme_bw() +
  coord_fixed()

par(mfrow = c(1, 2), mar = c(4,0,4,0))
trainingResults$HeartDisease <- as.vector(trainingResults$HeartDisease)

## Model 1
separationplot(
  pred = trainingResults$Heartlogmodel1Pred,
  actual = trainingResults$HeartDisease,
  type = "line",
  line = TRUE,
  col0 = "#eabcc9",
  col1 = "#a5b2ca",
  show.expected = TRUE,
  heading = "Model 1",
  newplot = FALSE
)

## Model 2
separationplot(
  pred = trainingResults$Heartlogmodel2Pred,
  actual = trainingResults$HeartDisease,
  type = "line",
  line = TRUE,
  col0 = "#eabcc9",
  col1 = "#a5b2ca",
  show.expected = TRUE,
  heading = "Model 2",
  newplot = FALSE
)
testingResults$predict <- predict(
  object = Heartlogmodel2,
```

```
  newdata = testingData,
  type = "response"
)
testingResults <- testingResults %>%
  mutate(
    Heartlogmodel2Class = case_when(
      predict > 0.5 ~ "Heart Disease",
      TRUE ~ "No Heart Disease"
    )
  )

# Build Confusion Matrix for Testing Data ----
testingResults %>%
  tabyl(var1 = Heartlogmodel2Class, var2 = HeartDisease) %>%
  adorn_title(
    placement = "combined",
    row_name = "Predicted",
    col_name = "Actual"
  ) %>%
  kable(
    booktabs = TRUE,
    align = "c"
  ) %>%
  kable_classic(
    position = "center",
    latex_options = c("HOLD_position"),
    full_width = FALSE
  )
testingResults$HeartDisease <- as.vector(testingResults$HeartDisease)
par(mar = c(4,0,0,0))
separationplot(
  pred = testingResults$predict,
  actual = testingResults$HeartDisease,
  type = "rect",
  rectborder = "black",
  col0 = "#eabcc9",
  col1 = "#a5b2ca",
  line = TRUE,
  lwd2 = 2,
  show.expected = TRUE,
  newplot = FALSE
)

# Accessing Residuals
residuals_logModel1 <- residuals(HeartlogModel1, type = "response")
residuals_logModel2 <- residuals(Heartlogmodel2, type = "response")


# Model Fit Metrics
# AIC and BIC are already appropriate for logistic regression
aic_1 <- AIC(HeartlogModel1)
aic_2 <- AIC(Heartlogmodel2)
```

```r
bic_1 <- BIC(HeartlogModel1)
bic_2 <- BIC(Heartlogmodel2)


# install.packages("DescTools") # Uncomment if not already installed
library(DescTools)

brier_1 <- BrierScore(HeartlogModel1)
brier_2 <- BrierScore(Heartlogmodel2)

pseudo_r2_1 <- PseudoR2(which = "Nagelkerke", HeartlogModel1)
pseudo_r2_2 <- PseudoR2(which = "Nagelkerke", Heartlogmodel2)

# Print the metrics
cat("AIC Model 1: ", aic_1, "\n")
cat("AIC Model 2: ", aic_2, "\n")

cat("BIC Model 1: ", bic_1, "\n")
cat("BIC Model 2: ", bic_2, "\n")

cat("Brier Score Model 1: ", brier_1, "\n")
cat("Brier Score Model 2: ", brier_2, "\n")

cat("Pseudo R-squared Model 1: ", pseudo_r2_1, "\n")
cat("Pseudo R-squared Model 2: ", pseudo_r2_2, "\n")

trainingData$HeartDisease <- ifelse(trainingData$HeartDisease == 0, "No Heart Disease", "Heart Disease")
testingData$HeartDisease <- ifelse(testingData$HeartDisease == 0, "No Heart Disease", "Heart Disease")

rpartHeart1 <- rpart(
  formula = HeartDisease ~ Age + Sex + ChestPainType + RestingBP + Cholesterol + FastingBS + RestingECG
  data = trainingData,
  method = "class",
  parms = list(split = "information"),
  control = rpart.control()
)


rpart.plot(
  x = rpartHeart1,
  type = 2,
  extra = 101
)

# Using rattle package
fancyRpartPlot(
  model = rpartHeart1,
  main = NULL,
  sub = NULL
)

# Get table elements
invisible(capture.output({cpTable <- printcp(rpartHeart1)}))
```

```r
# Create nice looking table of CP results
kable(
  x = cpTable,
  col.names = c("CP", "Num. of splits", "Rel. Error",
                "Mean Error", "Std. Deviation of Error"),
  digits = 3,
  booktabs = TRUE,
  align = "c"
) %>%
  kable_classic(
    full_width = FALSE
  ) %>%
  kable_styling(latex_options = c("HOLD_position"))
# Plot the CP results from rpart
plotcp(
  x = rpartHeart1,
  minline = TRUE,
  upper = "size"
)
# Pruning the rpart Tree
rpartHeart2 <- prune(
  tree = rpartHeart1,
  cp = 0.014
)
## The rpart package
pred_heartRpart2 <- predict(
  object = rpartHeart2,
  newdata = testingData,
  type = "prob"
)

# Data Wrangling the predictions
heartPredictions <- data.frame(
  rpart2_heart = pred_heartRpart2[, 1],
  rpart2_nonHeart = pred_heartRpart2[, 2]
) %>%
  mutate(
    rpart2_pred = ifelse(
      test = rpart2_heart > rpart2_nonHeart,
      yes = "Heart Disease",
      no = "No Heart Disease"
    )
  )

heartPredictions$rpart2_pred <- as.factor(heartPredictions$rpart2_pred)

# Merge supervision column into predictions data frame
heartPredictions <- cbind(
  tempID = testingData$tempID,
  HeartDisease = testingData$HeartDisease,
  heartPredictions
)
heartPredictions$HeartDisease <- as.factor(heartPredictions$HeartDisease)
```

```r
# Build confusion matrix for second rpart model
conf_mat(
  data = heartPredictions,
  truth = HeartDisease,
  estimate = rpart2_pred
)$table %>%
  kable(
    col.names = c("Heart Disease", "No Heart Disease"),
  digits = 3,
  booktabs = TRUE,
  align = "c"
  ) %>%
  kable_classic_2("striped",
    full_width = FALSE
  )%>%
  kable_styling(latex_options = c("HOLD_position")) %>%
  add_header_above() %>%
  footnote(
    general = "Pruned Decision Tree Model confusion Matrix",
    footnote_as_chunk = TRUE
  )
# Build a data frame with model metrics
heartPreds <- heartPredictions %>%
  dplyr::select(HeartDisease, contains("_pred")) %>%
  pivot_longer(
    cols = !c(HeartDisease),
    names_to = "model",
    values_to = "prediction"
  )

accuracy <- heartPreds %>%
  group_by(model) %>%
  accuracy(
    truth = HeartDisease,
    estimate = prediction
  )

sensitivity <- heartPreds %>%
  group_by(model) %>%
  sensitivity(
    truth = HeartDisease,
    estimate = prediction,
    event_level = "second"
  )

specificity <- heartPreds %>%
  group_by(model) %>%
  specificity(
    truth = HeartDisease,
    estimate = prediction,
    event_level = "second"
  )
```

```
modelMetrics <- bind_rows(
  accuracy,
  sensitivity,
  specificity
)
# Make a nice looking table of model metrics
modelMetrics %>%
  dplyr::select(model, .metric, .estimate) %>%
  pivot_wider(
    id_cols = model,
    names_from = .metric,
    values_from = .estimate
  ) %>%
  kable(
    digits = 3,
    booktabs = TRUE,
    align = "c",
    table.attr = 'data-quarto-disable-processing="true"'
  ) %>%
  kable_classic(
    full_width = FALSE
  )%>%
  kable_styling(latex_options = c("HOLD_position"))
# Using randomForest to use an ensemble method
trainingData$HeartDisease <- as.factor(trainingData$HeartDisease)
heartForest1 <- randomForest(
  formula = HeartDisease ~ Age + Sex + ChestPainType + RestingBP + Cholesterol + FastingBS + RestingECG
  data = trainingData,
  ntree = 1000,
  mtry = 5,
  importance = TRUE,
  do.trace = FALSE,
  keep.forest = TRUE
)
# Create a line plot of OOB Error and Misclassification Rates
as.data.frame(heartForest1$err.rate) %>%
  mutate(
    Tree = row_number(),
    .before = OOB
  ) %>%
  pivot_longer(
    cols = !Tree,
    names_to = "Type",
    values_to = "Error"
  ) %>%
  ggplot(
    mapping = aes(
      x = Tree,
      y = Error,
      color = Type,
      linetype = Type
    )
) +
```

```
  geom_path() +
  theme_bw() +
  scale_linetype_manual(values = c("dashed", "dotted", "solid"))
# Display attribute importance in a table
importance(heartForest1) %>%
  kable(
    digits = 3,
    booktabs = TRUE,
    align = "c",
    table.attr = 'data-quarto-disable-processing="true"'
  ) %>%
  kable_classic(
    full_width = FALSE
  )%>%
  kable_styling(latex_options = c("HOLD_position"))
varImpPlot(
  x = heartForest1,
  main = "Classifying People with Heart Disease/No Heart Disease"
)
# Predict new observations
testingResults <- testingData
testingResults$predicted <- predict(
  object = heartForest1,
  newdata = testingData,
  type = "response"
)
testingResults$HeartDisease <- factor(testingResults$HeartDisease)
# Build Confusion Matrix
conf_mat(
  data = testingResults,
  truth = HeartDisease,
  estimate = predicted
)$table %>%
  kable(
    col.names = c("Prediction/Supervision", "Heart Disease", "No Heart Disease"),
    digits = 3,
    booktabs = TRUE,
    align = "c",
    table.attr = 'data-quarto-disable-processing="true"'
  ) %>%
  kable_classic_2("striped",
    full_width = FALSE
  )%>%
  kable_styling(latex_options = c("HOLD_position"))
# Build a data frame with model metrics
heartTest <- testingResults%>%
  dplyr::select(HeartDisease, contains("predicted")) %>%
  pivot_longer(
    cols = !c(HeartDisease),
    names_to = "model",
    values_to = "prediction"
  )
```

```r
accuracy <- heartTest %>%
  group_by(model) %>%
  accuracy(
    truth = HeartDisease,
    estimate = prediction
  )

sensitivity <- heartTest %>%
  group_by(model) %>%
  sensitivity(
    truth = HeartDisease,
    estimate = prediction,
    event_level = "second"
  )

specificity <- heartTest %>%
  group_by(model) %>%
  specificity(
    truth = HeartDisease,
    estimate = prediction,
    event_level = "second"
  )

modelMetrics <- bind_rows(
  accuracy,
  sensitivity,
  specificity
)
# Make a nice looking table of model metrics
modelMetrics %>%
  dplyr::select(model, .metric, .estimate) %>%
  pivot_wider(
    id_cols = model,
    names_from = .metric,
    values_from = .estimate
  ) %>%
  kable(
    digits = 3,
    booktabs = TRUE,
    align = "c",
    table.attr = 'data-quarto-disable-processing="true"'
  ) %>%
  kable_classic(
    full_width = FALSE
  )%>%
  kable_styling(latex_options = c("HOLD_position"))
testingData <- as.data.frame(testingData)

library(dplyr)
testingData$HeartDisease <- factor(testingData$HeartDisease)
# Using randomForest to do training and testing
heartForest3 <- randomForest(
  x = trainingData[, which(!(names(trainingData) %in% c("tempID", "HeartDisease")))],
```

```
  y = trainingData$HeartDisease,
  xtest <- testingData[, which(!(names(testingData) %in% c("tempID", "HeartDisease")))],
  ytest = testingData$HeartDisease,
  ntree = 1000,
  mtry = 5,
  importance = TRUE,
  do.trace = FALSE,
  keep.forest = TRUE
)

# Plot of OOB and Testing Set Errors
bind_cols(
  oob = as.data.frame(heartForest3$err.rate)$OOB,
  test = as.data.frame(heartForest3$test$err.rate)$Test
) %>%
  mutate(
    tree = row_number(),
    .before = oob
  ) %>%
  pivot_longer(
    cols = !tree,
    names_to = "Source",
    values_to = "Error"
  ) %>%
  ggplot(
    mapping = aes(
      x = tree,
      y = Error,
      color = Source,
      linetype = Source
    )
  ) +
  geom_path() +
  theme_bw()
cleanHeart <- heart %>%
  mutate(
    across(
      .cols = where(is.numeric),
      .fns = ~scale(.x)[,1],
      .names = "{.col}_std"
    )
  )


# Deal with qualities ----
cleanHeart <- cleanHeart %>%
  mutate(
    sex_coded = case_match(
      Sex,
      "F" ~ 1,
      .default = 0
    )
  )
```

```r
cleanHeart <- cleanHeart %>%
  mutate(
    ExerciseAngina_coded = case_match(
      ExerciseAngina,
      "Y" ~ 1,
      .default = 0
    )
  )

coreHeart <- cleanHeart %>%
  dplyr::select(contains("_std"), sex_coded, FastingBS, HeartDisease,ExerciseAngina_coded)
coreHeart <- coreHeart[, -which(names(coreHeart) == "HeartDisease_std")]
coreHeart <- coreHeart[, -which(names(coreHeart) == "FastingBS_std")]
distHeart <- dist(
  x = coreHeart,
  method = "euclidean"
)

fviz_dist(
  dist.obj = distHeart,
  order = TRUE,
  show_labels = FALSE
)

# Hierarchical clustering
hcHeart <- hclust(
  d = distHeart,
  method = "ward.D"
)
# Add customization to base R via dendrogram object ----
par(mar = c(4,0,0,0), cex = 0.5)
plot(
  as.dendrogram(
    object = hcHeart,
    hang = -1
  ),
  type = "rectangle",
  horiz = TRUE
)
set.seed(380)
kmHeart <- kmeans(
  x = distHeart,
  centers = 3,
  iter.max = 10,
  nstart = 25
)

# View k-means cluster results ----
# library(factoextra)
fviz_cluster(
  object = kmHeart,
  data = coreHeart,
  stand = FALSE,
```

```
  geom = "point",
  main = "3-Means Cluster Plot"
) +
  theme_bw()
coreHeart <- as.data.frame(coreHeart)
set.seed(380)
fviz_nbclust(
  x = coreHeart,
  diss = NULL,
  FUNcluster = kmeans,
  method = "wss",
  k.max = 10
)

# Create average silhouette plot for choosing k ----
set.seed(380)
fviz_nbclust(
  x = coreHeart,
  diss = NULL,
  FUNcluster = hcut,
  method = "silhouette",
  k.max = 10
)
# Create Gap Statistic plot for choosing k ----
set.seed(380)
fviz_nbclust(
  x = coreHeart,
  diss = NULL,
  FUNcluster = kmeans,
  method = "gap_stat",
  k.max = 10,
  nboot = 100
)
# Create Gap Statistic plot for choosing k ----
set.seed(380)
HeartGap <- clusGap(
  x = coreHeart,
  FUNcluster = kmeans,
  K.max = 10,
  B = 100
)

# Visualize the gap statistics to help determine the optimal number of clusters
fviz_gap_stat(
  gap_stat = HeartGap,
  maxSE = list(method = "Tibs2001SEmax")
)

# Cut hierarchical clustering at 2 and 3 clusters ----
library(dplyr)
cleanHeart <- cleanHeart %>%
  mutate(
    hc2 = as.factor(cutree(tree = hcHeart, k = 2)),
```

```
    hc4 = as.factor(cutree(tree = hcHeart, k = 3))
  )
par(mar = c(4,0,0,0), cex = 0.5)
plot(
  as.dendrogram(
    object = hcHeart,
    hang = -1
  ),
  type = "rectangle",
  horiz = TRUE
)

rect.hclust(
  tree = hcHeart,
  k = 2,
)

# Base R----
plot(hcHeart, cex = 0.4)
rect.hclust(
  tree = hcHeart,
  k = 3
)
# A second k-means clustering of  ----
k2Heart <- kmeans(
  x = distHeart,
  centers = 2,
  iter.max = 10,
  nstart = 25
)
# Extract the 2- and 4-cluster kmeans solutions ----
## Add categorization to my cleaned data
cleanHeart <- cleanHeart %>%
  mutate(
    k2 = as.factor(k2Heart$cluster),
    k3 = as.factor(kmHeart$cluster)
  )
# Hybrid approach to clustering ----
# library(factoextra)
hybridHeart <- hkmeans(
  x = coreHeart,
  k = 3,
  hc.metric = "euclidean",
  hc.method = "ward.D",
  iter.max = 10
)
# Plot the initial dendrogram for hybrid approach ----
# library(factoextra)
set.seed(380)
hkmeans_tree(
  hkmeans = hybridHeart,
  rect.col = c("red", "blue", "green"),
  cex = 0.4,
```

```
  main = "Initial Hierarchical Clusters"
)
# Plot the final dendrogram for hybrid approach ----
# library(factoextra)
fviz_dend(
  x = hybridHeart,
  cex = 0.4,
  palette = c("red", "blue", "green"),
  rect = FALSE,
  horiz = TRUE,
  repel = TRUE,
  main = "Final Dendrogram"
)
```