TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

938 Aurora Blvd. Cubao, Quezon City

COLLEGE OF COMPUTER STUDIES

Project Compilation

Heart Disease Classification Model using the Random Forest Algorithm

In Partial Fulfillment of the Requirements for

ITE 030 - Data Analytics

by:

Casile, Jasper Riley P.

Ongsiako, Cailo Nehru P.

Paragas, Veronica Maxine D.

Ms. Nila D. Santiago

Instructor

June 2024

I. Introduction

Cardiovascular diseases are prevalent these days, they describe a range of conditions that could affect your heart. The World Health Organization estimates that 17.9 million global deaths are from Cardiovascular diseases (CVDs). Heart diseases are known to be one of the leading causes of death worldwide, day by day, each case of heart disease is increasing at a rapid rate, making accurate prediction of heart disease risk a critical challenge (Jindal H., et al., 2021). However multiple linear regression would be a good statistical method for predicting the likelihood of heart disease based on different factors.

Recent research has increasingly focused on applying machine learning algorithms for heart disease prediction. For instance, Reddy et al. (2021) compared several machine learning classifiers, including logistic regression, decision trees, and random forests, finding that ensemble methods yielded the highest accuracy. Mohan et al. (2022) created a hybrid machine-learning model that combined multiple algorithms to improve prediction accuracy.

Despite advancements in machine learning, multiple linear regression has its benefits. It can manage different predictor variables, is easy to interpret, and provides a clear estimate of the impact of each risk factor on heart disease likelihood. By using multiple linear regression for heart disease prediction, the researchers aim to create a model that is both accurate and easy to understand, helping doctors make better decisions. This method prioritizes improved early detection and guides preventive measures to reduce the impact of heart disease.

II. Business Understanding

Background of the Study

Based on research conducted by WHO in 2021, heart disease is still internationally recognized for taking away almost 18 million lives annually. According to Shu (2017), traditional diagnostic methods are time-consuming and/or invasive making it a major hassle given that it may still be prone to errors. Heart disease prediction is crucial for early diagnosis and prevention, potentially saving millions of lives and reducing healthcare costs. Traditional methods of diagnosing heart disease often rely on manual interpretation of medical data, which can be time-consuming and prone to error (Heart and Stroke Foundation of Canada, 2020). Common diagnostic tests include blood tests, chest X-rays, and electrocardiograms (ECGs), which record the electrical signals in the heart to detect abnormalities (Mayo Clinic, 2022).

The integration of machine learning in healthcare is a growing trend, with algorithms being used to analyze vast amounts of patient data to identify patterns and predict outcomes. Current data analytics solutions for heart disease prediction include logistic regression, decision trees, and neural networks. However, these models often require extensive computational resources and may not be interpretable, which is critical in medical applications (Rajkomar, Dean, & Kohane, 2019).

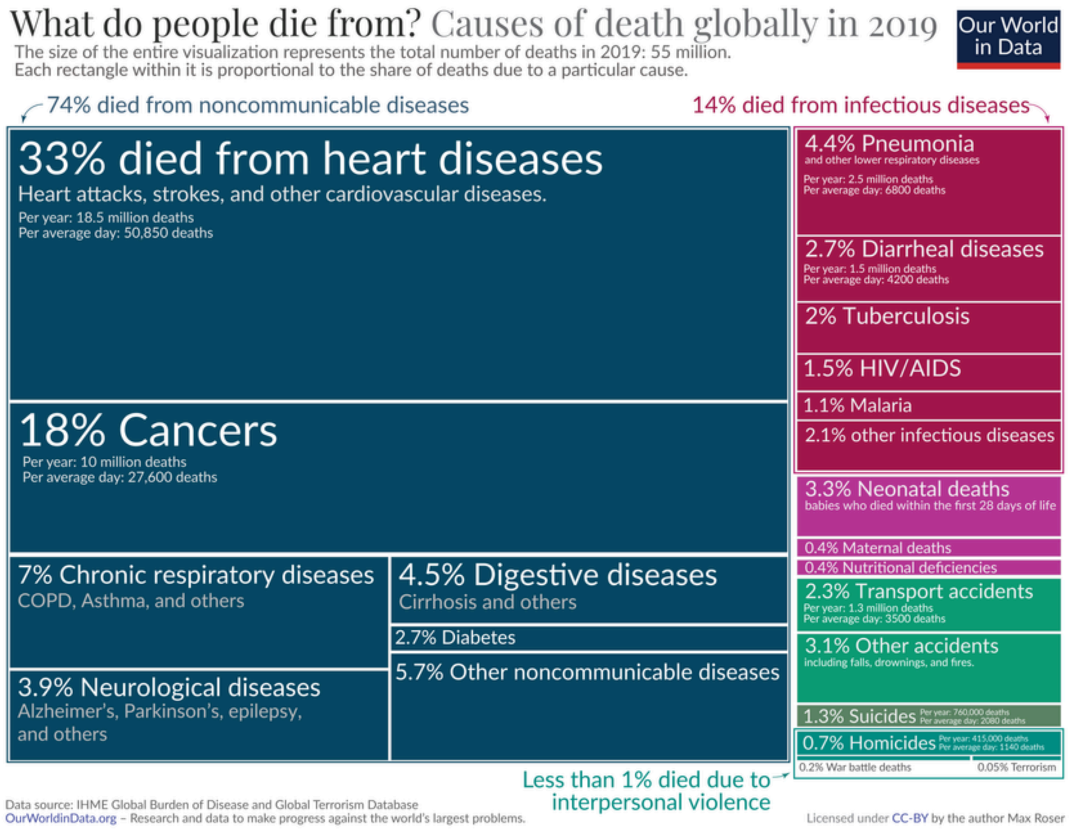
This project intends to fill these gaps by developing a multiple linear regression model that not only predicts heart disease with high accuracy and provides interpretable results but also, provides less expensive diagnostic check-ups. By doing so, it aims to assist healthcare professionals in making informed decisions and improving patient outcomes. The project will focus on optimizing the model for accuracy, handling multicollinearity, and ensuring a reliable prediction. By utilizing data from individuals with heart disease, the model can offer another basis for determining the risk, complementing existing diagnostic methods.

A. Current metrics, trends, or dashboard

As seen in Figure 1, this shows that globally, 33% of deaths are due to heart diseases (e.g.heart attacks, strokes, and other cardiovascular diseases) and it is the most common cause of death, responsible for a third of all deaths globally, a total of around 18 million. This is according to IHME (2019)

Figure 1.

What do people die from? Causes of death globally in 2019



Data source: IHME Global Burden of Disease and Global Terrorism Database
OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the author Max Roser

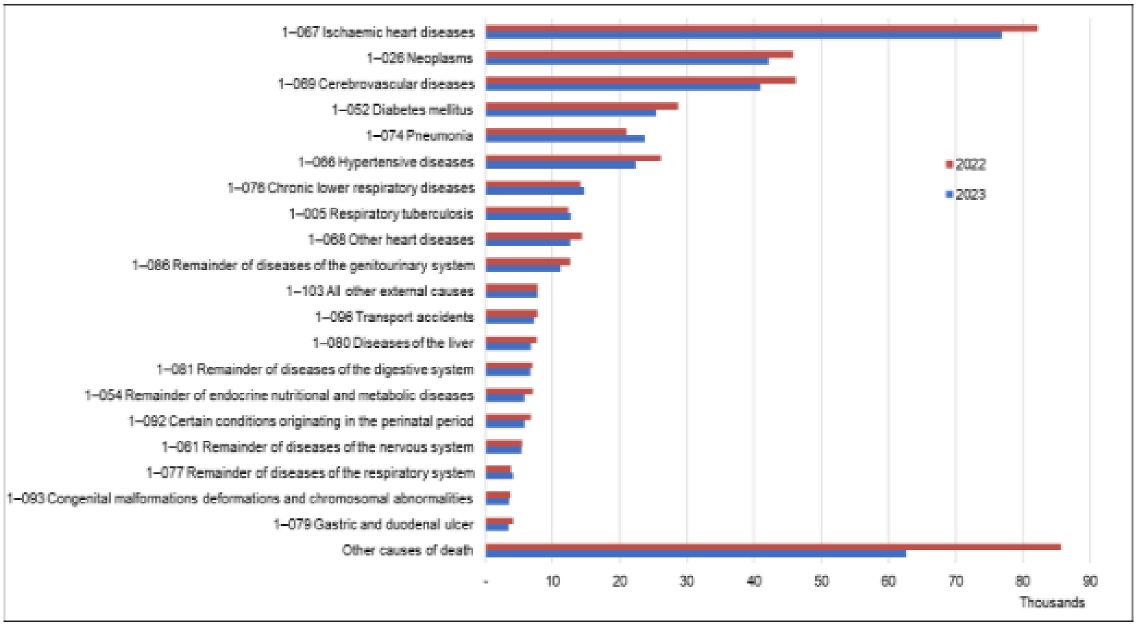
Here in figure 2, consists of the top 20 causes of death in the Philippines starting from January to August in 2022 and 2023. The leading causes were due to ischaemic heart diseases, neoplasms, and cerebrovascular diseases. And according to the Philippine Statistics Authority (2024), Ischemic heart disease is the first cause. From January to

August of 2023, ischaemic heart diseases were the leading cause of death with 76,901 cases or 19.1% were the total deaths in the country.

Figure 2.

All causes of mortality (Top 20), Philippines: January to August, 2022 & 2023.

Figure 1. All Causes of Mortality (Top 20), Philippines: January to August, 2022 and 2023



Source: Philippine Statistics Authority

Note: Symptoms, signs, and abnormal clinical and laboratory findings, not elsewhere classified (R00-R99) are not included in the analysis due to the unspecified nature of these causes.

The high cost of hospitalization and diagnostic testing drastically impacts the economic status and accessibility of healthcare services for many individuals and families. According to Tumanan-Mendoza (2018), a study in the Philippines revealed that hospitalizing patients with congestive heart failure (CHF) is costly. Government hospitals charged between PHP 19,340 and PHP 41,800 per case, while the Philippine Health Insurance Corporation (PhilHealth) covered only PHP 15,700 per case. Non-healthcare expenses, like lost income and transportation, added PHP 10,700 to PHP 14,600 per case. The study estimated the total economic burden due to congestive heart failure hospitalizations, ranging from PHP 851,850,000 to PHP 1,841,563,000. These findings highlight the significant financial strain on patients.

B. Statement of the Problems

Heart disease remains a global concern, contributing to high morbidity and mortality rates. Early detection and accurate detection of possible heart disease are crucial for on-point intervention and prevention strategies. Despite advances in medical diagnostics, there is still a need for effective and efficient classification models that can leverage clinical and demographic data to identify individuals at high risk of developing heart disease.

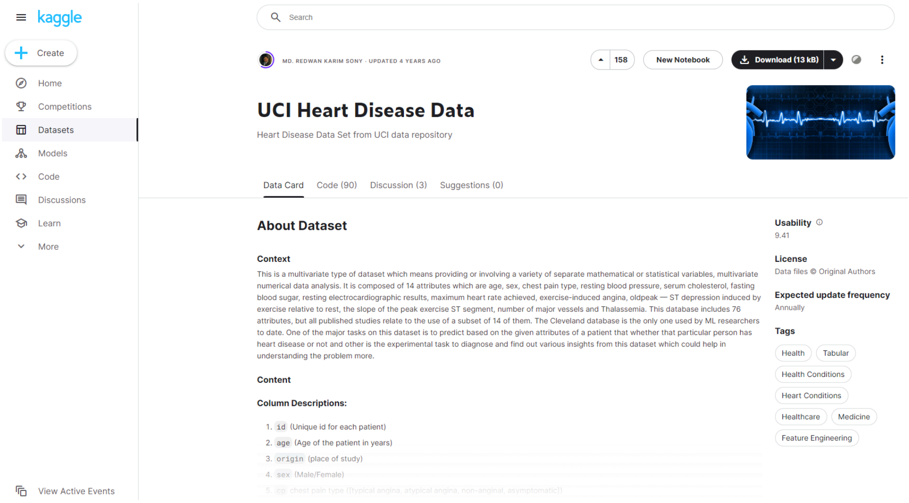
III. Data Sample, Extraction, and Data Mining

The data source comes from the UCI heart disease dataset. The data comes from four different locations, namely: (1) Hungarian Institute of Cardiology in Budapest, Hungary, (2) University Hospital in Zurich, Switzerland, (3) University Hospital in Base, Switzerland, and (3) V.A. Medical Center in Long Beach, California and Cleveland Clinic Foundation. The data from these four locations was combined into a comprehensive dataset which checks for the likelihood that a person has heart disease based on the 14 attributes.

Dataset: *https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data*

Figure 3.

Dataset from Kaggle



MD. REDWAN KARIM SONY · UPDATED 4 YEARS AGO

UCI Heart Disease Data

Heart Disease Data Set from UCI data repository

About Dataset

Context

This is a multivariate type of dataset which means providing or involving a variety of separate mathematical or statistical variables, multivariate numerical data analysis. It is composed of 14 attributes which are age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, oldpeak — ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels and Thalassemia. This database includes 76 attributes, but all published studies relate to the use of a subset of 14 of them. The Cleveland database is the only one used by ML researchers to date. One of the major tasks on this dataset is to predict based on the given attributes of a patient that whether that particular person has heart disease or not and other is the experimental task to diagnose and find out various insights from this dataset which could help in understanding the problem more.

Content

Column Descriptions:

1. `id` (Unique id for each patient)
2. `age` (Age of the patient in years)
3. `origin` (place of study)
4. `sex` (Male/Female)
5. `cp` chest pain type ([typical angina, atypical angina, non-anginal, asymptomatic])

Usability
9.41

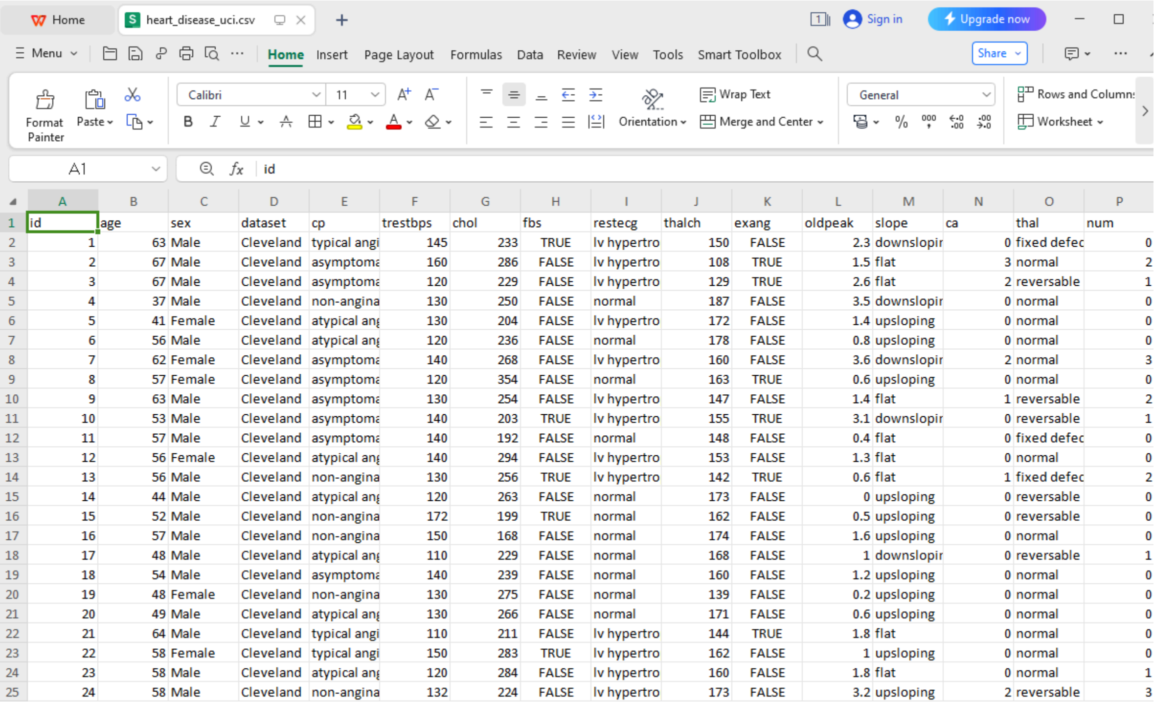
License
Data files © Original Authors

Expected update frequency
Annually

Tags
Health
Tabular
Health Conditions
Heart Conditions
Healthcare
Medicine
Feature Engineering

Figure 4.

Dataset in CSV File



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	num
2	1	63	Male	Cleveland	typical angi	145	233	TRUE	lv hypertro	150	FALSE	2.3	downslopir	0	fixed defec	0
3	2	67	Male	Cleveland	asymptoma	160	286	FALSE	lv hypertro	108	TRUE	1.5	flat	3	normal	2
4	3	67	Male	Cleveland	asymptoma	120	229	FALSE	lv hypertro	129	TRUE	2.6	flat	2	reversable	1
5	4	37	Male	Cleveland	non-angina	130	250	FALSE	normal	187	FALSE	3.5	downslopir	0	normal	0
6	5	41	Female	Cleveland	atypical ang	130	204	FALSE	lv hypertro	172	FALSE	1.4	upsloping	0	normal	0
7	6	56	Male	Cleveland	atypical ang	120	236	FALSE	normal	178	FALSE	0.8	upsloping	0	normal	0
8	7	62	Female	Cleveland	asymptoma	140	268	FALSE	lv hypertro	160	FALSE	3.6	downslopir	2	normal	3
9	8	57	Female	Cleveland	asymptoma	120	354	FALSE	normal	163	TRUE	0.6	upsloping	0	normal	0
10	9	63	Male	Cleveland	asymptoma	130	254	FALSE	lv hypertro	147	FALSE	1.4	flat	1	reversable	2
11	10	53	Male	Cleveland	asymptoma	140	203	TRUE	lv hypertro	155	TRUE	3.1	downslopir	0	reversable	1
12	11	57	Male	Cleveland	asymptoma	140	192	FALSE	normal	148	FALSE	0.4	flat	0	fixed defec	0
13	12	56	Female	Cleveland	atypical ang	140	294	FALSE	lv hypertro	153	FALSE	1.3	flat	0	normal	0
14	13	56	Male	Cleveland	non-angina	130	256	TRUE	lv hypertro	142	TRUE	0.6	flat	1	fixed defec	2
15	14	44	Male	Cleveland	atypical ang	120	263	FALSE	normal	173	FALSE	0	upsloping	0	reversable	0
16	15	52	Male	Cleveland	non-angina	172	199	TRUE	normal	162	FALSE	0.5	upsloping	0	reversable	0
17	16	57	Male	Cleveland	non-angina	150	168	FALSE	normal	174	FALSE	1.6	upsloping	0	normal	0
18	17	48	Male	Cleveland	atypical ang	110	229	FALSE	normal	168	FALSE	1	downslopir	0	reversable	1
19	18	54	Male	Cleveland	asymptoma	140	239	FALSE	normal	160	FALSE	1.2	upsloping	0	normal	0
20	19	48	Female	Cleveland	non-angina	130	275	FALSE	normal	139	FALSE	0.2	upsloping	0	normal	0
21	20	49	Male	Cleveland	atypical ang	130	266	FALSE	normal	171	FALSE	0.6	upsloping	0	normal	0
22	21	64	Male	Cleveland	typical angi	110	211	FALSE	lv hypertro	144	TRUE	1.8	flat	0	normal	0
23	22	58	Female	Cleveland	typical angi	150	283	TRUE	lv hypertro	162	FALSE	1	upsloping	0	normal	0
24	23	58	Male	Cleveland	atypical ang	120	284	FALSE	lv hypertro	160	FALSE	1.8	flat	0	normal	1
25	24	58	Male	Cleveland	non-angina	132	224	FALSE	lv hypertro	173	FALSE	3.2	upsloping	2	reversable	3

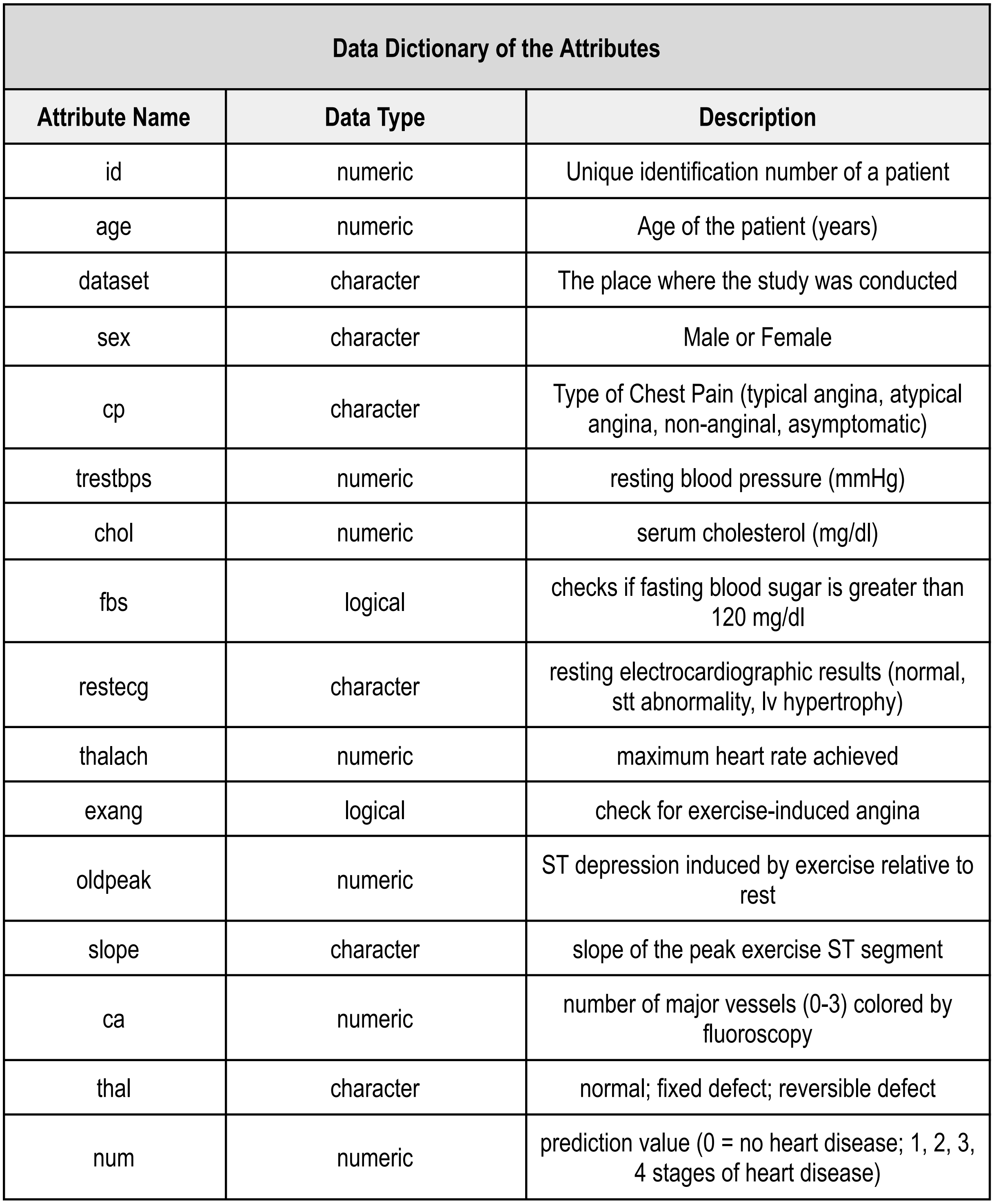
IV. Data Exploration, Preparation, and Transformation

Exploration of Data (Reporting, Summarizing, and Descriptive Analysis)

- **Reporting**
  - The dataset has 16 different attributes with 920 instances. Table 1 tabulates the different attributes on the dataset, their data type, and their description.
  - Figure 7 shows that there are attributes with missing values. These may cause biases or inaccuracies on the prediction if it is not handled properly in the data preparation process.

Table 1

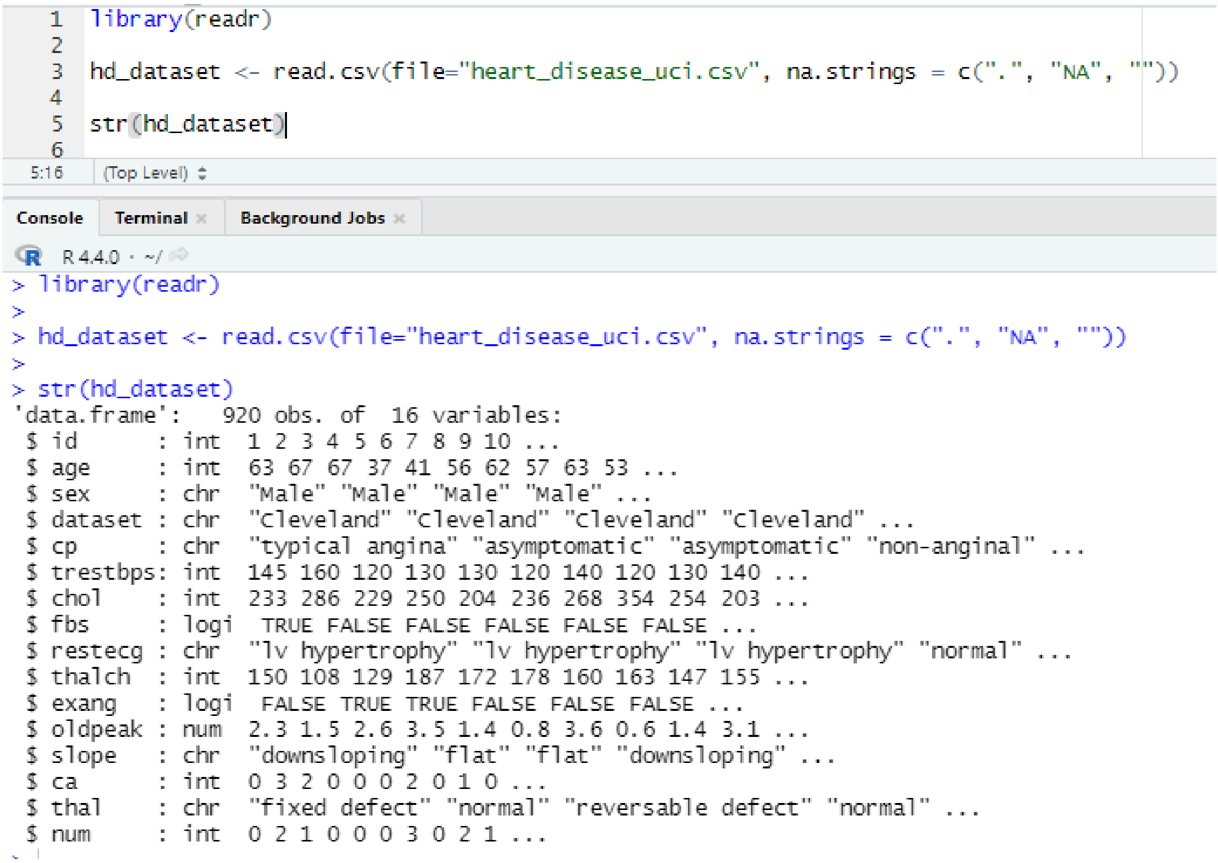
Data Dictionary of the Attributes



Data Dictionary of the Attributes		
Attribute Name	**Data Type**	**Description**
id	numeric	Unique identification number of a patient
age	numeric	Age of the patient (years)
dataset	character	The place where the study was conducted
sex	character	Male or Female
cp	character	Type of Chest Pain (typical angina, atypical angina, non-anginal, asymptomatic)
trestbps	numeric	resting blood pressure (mmHg)
chol	numeric	serum cholesterol (mg/dl)
fbs	logical	checks if fasting blood sugar is greater than 120 mg/dl
restecg	character	resting electrocardiographic results (normal, stt abnormality, lv hypertrophy)
thalach	numeric	maximum heart rate achieved
exang	logical	check for exercise-induced angina
oldpeak	numeric	ST depression induced by exercise relative to rest
slope	character	slope of the peak exercise ST segment
ca	numeric	number of major vessels (0-3) colored by fluoroscopy
thal	character	normal; fixed defect; reversible defect
num	numeric	prediction value (0 = no heart disease; 1, 2, 3, 4 stages of heart disease)

Figure 5

Checking for structure of the dataset with RStudio

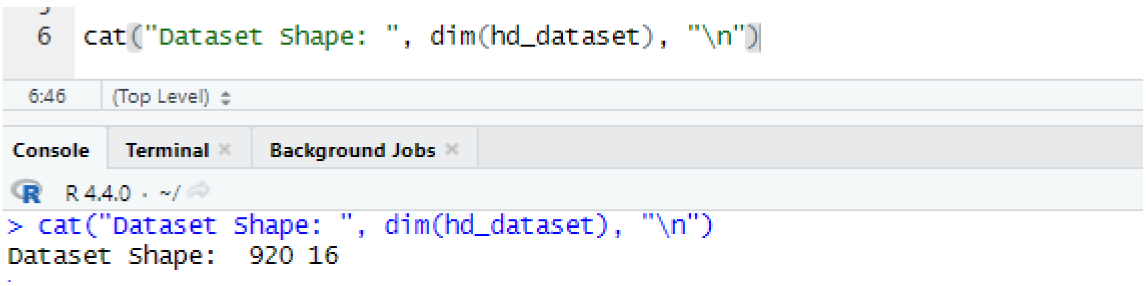


```
library(readr)

hd_dataset <- read.csv(file="heart_disease_uci.csv", na.strings = c(".", "NA", ""))

str(hd_dataset)

```

```
> library(readr)
>
> hd_dataset <- read.csv(file="heart_disease_uci.csv", na.strings = c(".", "NA", ""))
>
> str(hd_dataset)
'data.frame':	920 obs. of  16 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ age     : int  63 67 67 37 41 56 62 57 63 53 ...
 $ sex     : chr  "Male" "Male" "Male" "Male" ...
 $ dataset : chr  "Cleveland" "Cleveland" "Cleveland" "Cleveland" ...
 $ cp      : chr  "typical angina" "asymptomatic" "asymptomatic" "non-anginal" ...
 $ trestbps: int  145 160 120 130 130 120 140 120 130 140 ...
 $ chol    : int  233 286 229 250 204 236 268 354 254 203 ...
 $ fbs     : logi  TRUE FALSE FALSE FALSE FALSE FALSE ...
 $ restecg : chr  "lv hypertrophy" "lv hypertrophy" "lv hypertrophy" "normal" ...
 $ thalch  : int  150 108 129 187 172 178 160 163 147 155 ...
 $ exang   : logi  FALSE TRUE TRUE FALSE FALSE FALSE ...
 $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
 $ slope   : chr  "downsloping" "flat" "flat" "downsloping" ...
 $ ca      : int  0 3 2 0 0 0 2 0 1 0 ...
 $ thal    : chr  "fixed defect" "normal" "reversable defect" "normal" ...
 $ num     : int  0 2 1 0 0 0 3 0 2 1 ...
```

Figure 6

Checking for the shape of the dataset with RStudio

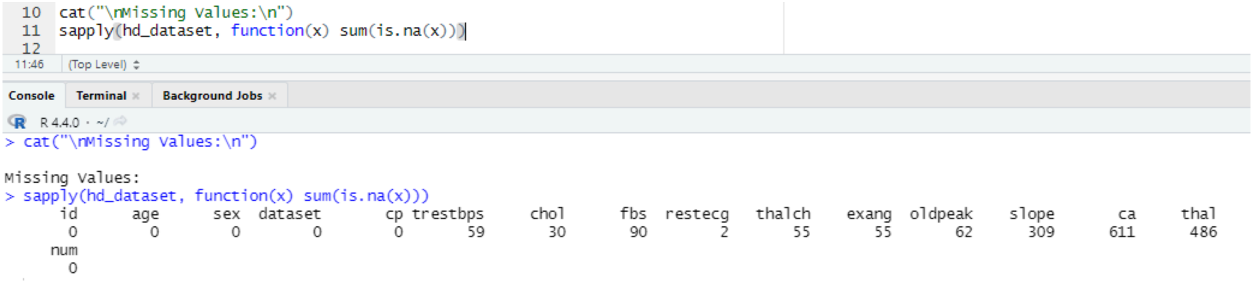


```
cat("Dataset Shape: ", dim(hd_dataset), "\n")
```

```
> cat("Dataset Shape: ", dim(hd_dataset), "\n")
Dataset Shape:  920 16
```

Figure 7

Checking for the number of missing values on each column of the dataset



```
cat("\nMissing Values:\n")
sapply(hd_dataset, function(x) sum(is.na(x)))
```

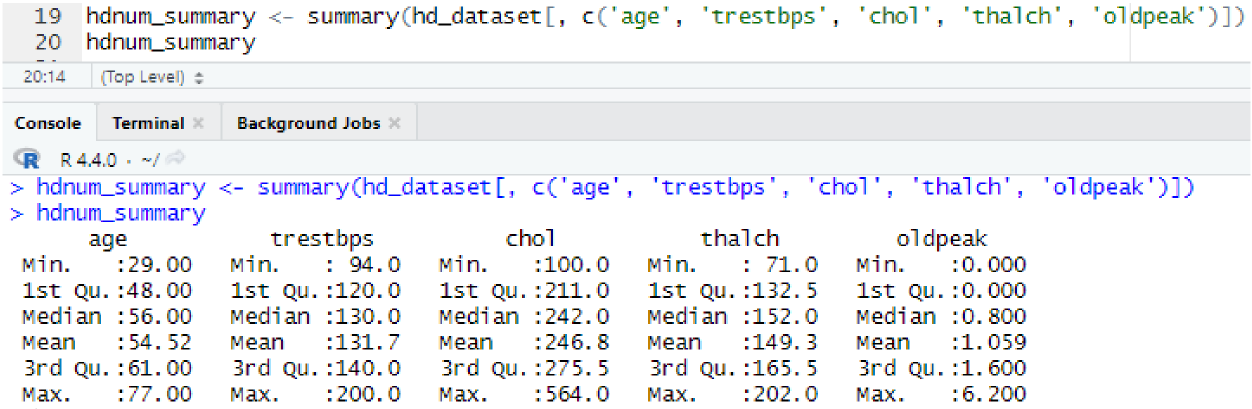
```
> cat("\nMissing Values:\n")

Missing Values:
> sapply(hd_dataset, function(x) sum(is.na(x)))
      id      age      sex  dataset       cp trestbps     chol      fbs  restecg   thalch    exang  oldpeak    slope       ca     thal 
       0        0        0        0        0       59       30       90        2       55       55       62      309      611      486 
     num 
       0 
```

- Summarizing
  - Summarization for Numerical Variables

    Using the summary() function, the summary statistics of the numeric variables on the dataset. Take note that this summarization is before the data cleaning process and may still be changed as the missing values are still not handled.

Figure 8

Statistical summary of the numeric variables on the dataset.



```
hdnum_summary <- summary(hd_dataset[, c('age', 'trestbps', 'chol', 'thalch', 'oldpeak')])
hdnum_summary
```

```
> hdnum_summary <- summary(hd_dataset[, c('age', 'trestbps', 'chol', 'thalch', 'oldpeak')])
> hdnum_summary
      age           trestbps          chol           thalch          oldpeak     
 Min.   :29.00   Min.   : 94.0   Min.   :100.0   Min.   : 71.0   Min.   :0.000  
 1st Qu.:48.00   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:132.5   1st Qu.:0.000  
 Median :56.00   Median :130.0   Median :242.0   Median :152.0   Median :0.800  
 Mean   :54.52   Mean   :131.7   Mean   :246.8   Mean   :149.3   Mean   :1.059  
 3rd Qu.:61.00   3rd Qu.:140.0   3rd Qu.:275.5   3rd Qu.:165.5   3rd Qu.:1.600  
 Max.   :77.00   Max.   :200.0   Max.   :564.0   Max.   :202.0   Max.   :6.200  
```

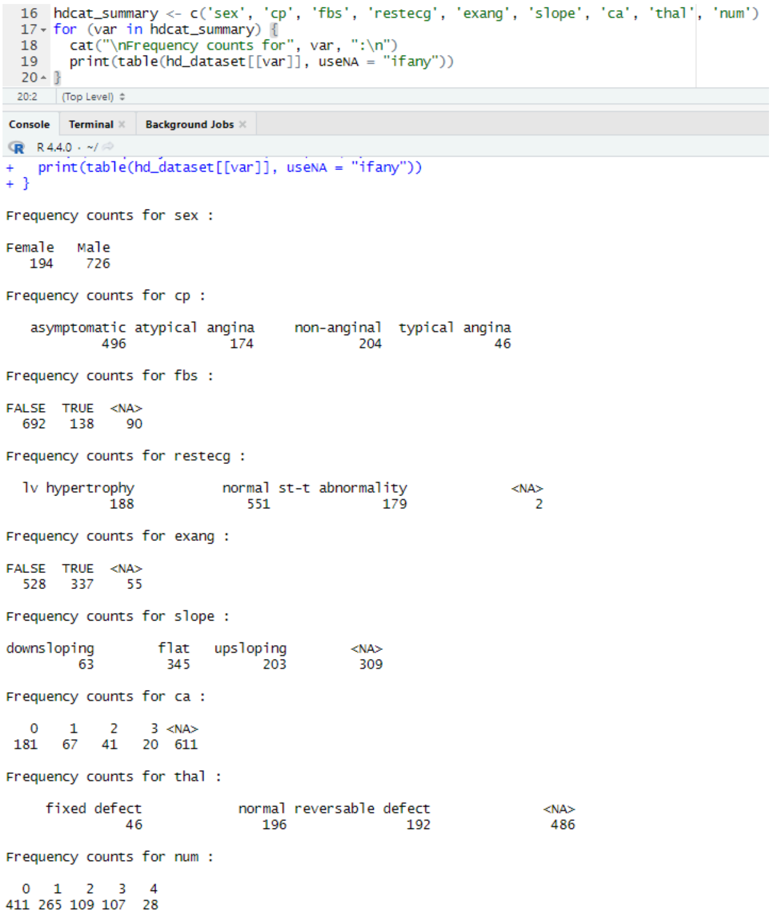
Note: The dataset is yet to be cleaned at this stage, thus, it may yield unreliable results. The researchers followed the order of the document template.

- Summarization for Categorical Variables

  Using the table(), the researchers are able to find the frequency for each possible value in an attribute. By looping this function, thay are able to do the previous function to all attributes defined in the hdcat_summary vector.

Figure 9

Summary of the frequency for the categorical variables on the dataset.



```
hdcat_summary <- c('sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'num')
for (var in hdcat_summary) {
  cat("\nFrequency counts for", var, ":\n")
  print(table(hd_dataset[[var]], useNA = "ifany"))
}
```

```
+   print(table(hd_dataset[[var]], useNA = "ifany"))
+ }

Frequency counts for sex :

Female   Male 
   194    726 

Frequency counts for cp :

   asymptomatic atypical angina     non-anginal  typical angina 
            496             174             204              46 

Frequency counts for fbs :

FALSE  TRUE  <NA> 
  692   138    90 

Frequency counts for restecg :

  lv hypertrophy           normal st-t abnormality             <NA> 
             188              551              179                2 

Frequency counts for exang :

FALSE  TRUE  <NA> 
  528   337    55 

Frequency counts for slope :

downsloping        flat   upsloping        <NA> 
         63         345         203         309 

Frequency counts for ca :

   0    1    2    3 <NA> 
 181   67   41   20  611 

Frequency counts for thal :

     fixed defect           normal reversable defect              <NA> 
               46              196               192               486 

Frequency counts for num :

  0   1   2   3   4 
411 265 109 107  28 
```

Descriptive Analysis

To perform the descriptive analysis, the researchers will perform the data cleaning first to ensure that the insights gained from this section is reliable. For this section, the researchers worked with data that is already cleaned (see the Preparation section for further details).

Figure 10 shows the age distribution for the cleaned dataset. Based on the values given using the summary() function, the average age for this dataset is 54. The youngest patient on this dataset is 29 while the oldest patient is 77. Lastly, 25% of the data from the dataset is equal or below 48 while the 75% of the data is equal or below 61.

Figure 10

Histogram for the age attribute

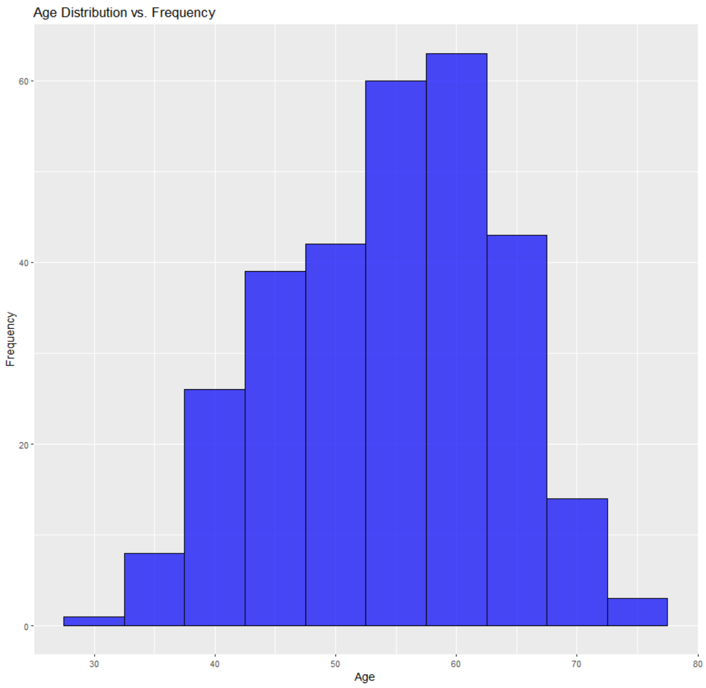


Figure 11 shows the scatter plot for each numeric value, tested against each combination.

Figure 11

Scatter plot for numeric values

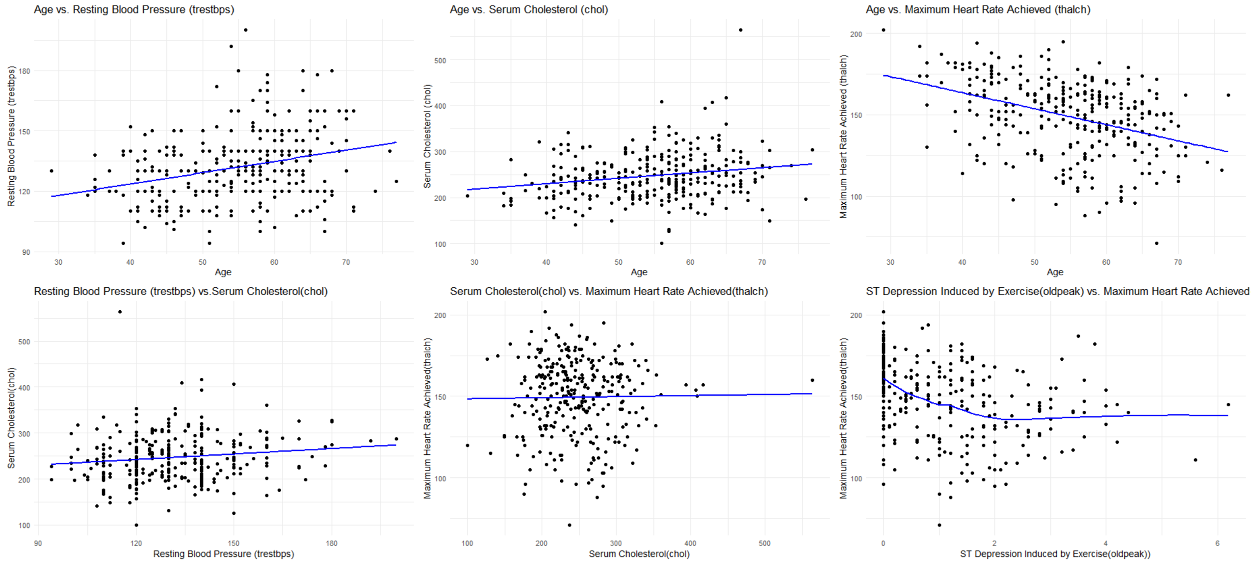


Figure 12 shows the distribution of heart disease based on sex. As stated on Table 1, each number represents the stage of heart disease of that patient. The number 0 represents patients with no heart disease, number 1 for stage 1, and so on. The stacked bar plot is used to represent the population of each stage to easily identify the quantity of each instance.

Figure 12

Distribution of stages of heart disease based on sex

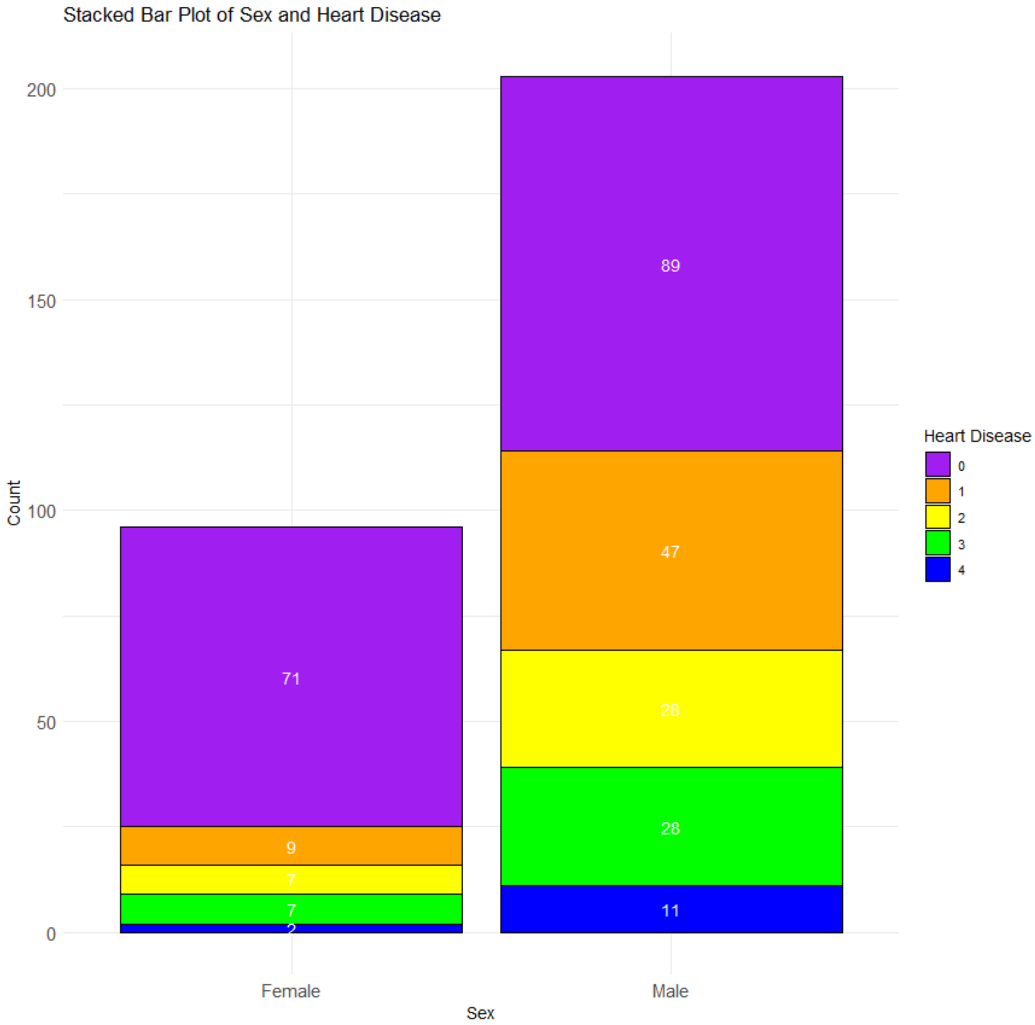
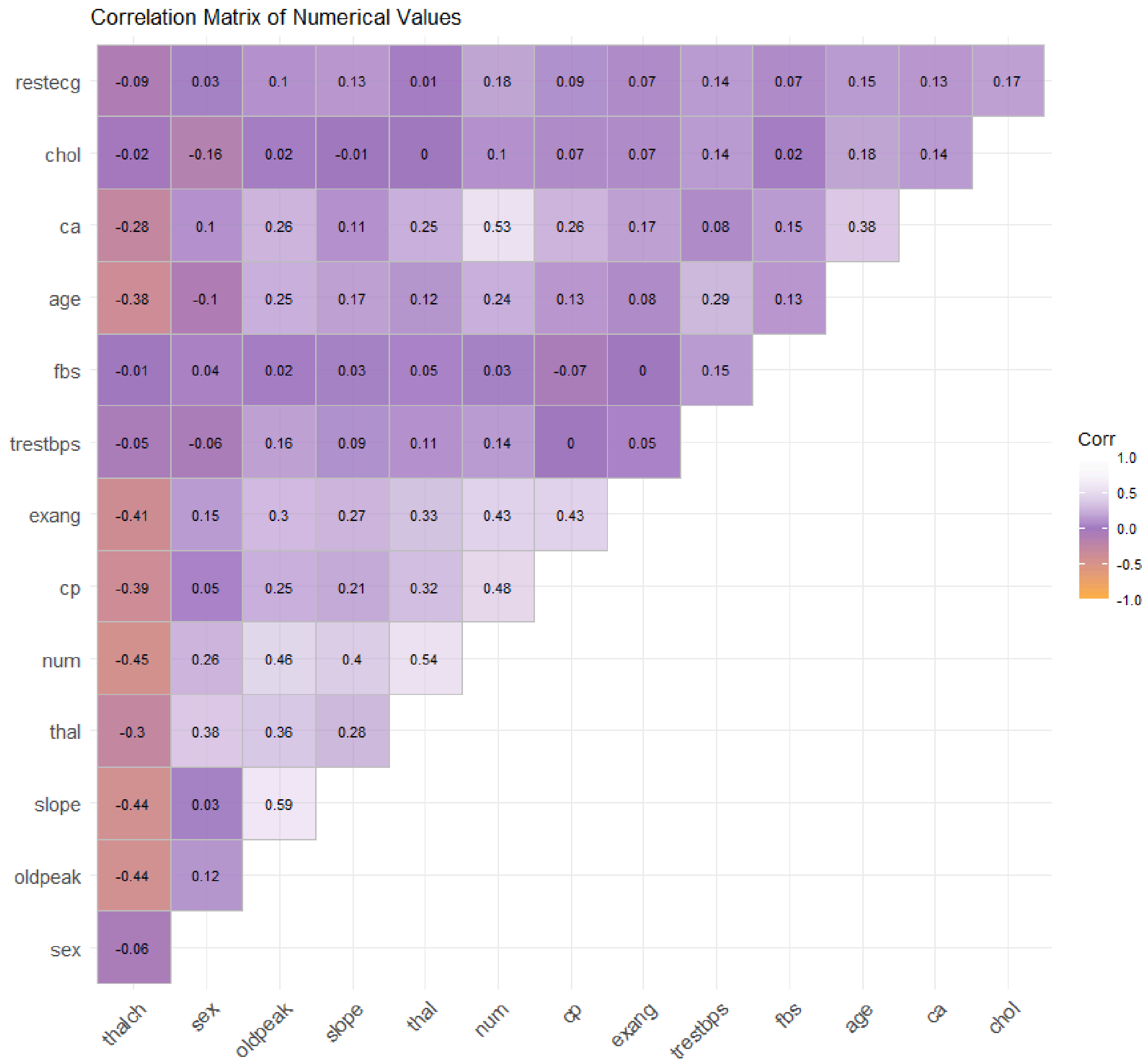


Figure 13 shows the correlation matrix of each of the relevant variables. The values on each square represents the strength of relationship between the variables wherein the value that is closer to 1 represents higher correlation.

Figure 13

Correlation Matrix of Variables



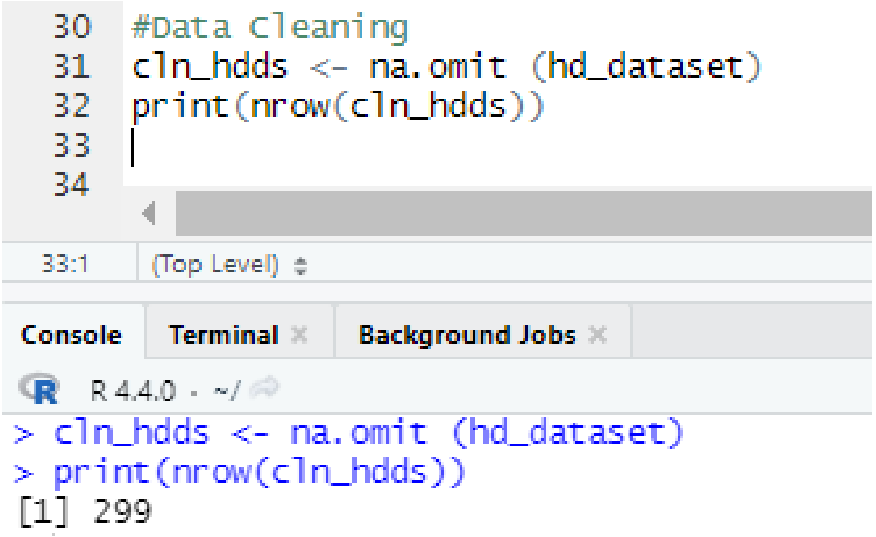
	thalch	sex	oldpeak	slope	thal	num	cp	exang	trestbps	fbs	age	ca	chol
restecg	-0.09	0.03	0.1	0.13	0.01	0.18	0.09	0.07	0.14	0.07	0.15	0.13	0.17
chol	-0.02	-0.16	0.02	-0.01	0	0.1	0.07	0.07	0.14	0.02	0.18	0.14	
ca	-0.28	0.1	0.26	0.11	0.25	0.53	0.26	0.17	0.08	0.15	0.38		
age	-0.38	-0.1	0.25	0.17	0.12	0.24	0.13	0.08	0.29	0.13			
fbs	-0.01	0.04	0.02	0.03	0.05	0.03	-0.07	0	0.15				
trestbps	-0.05	-0.06	0.16	0.09	0.11	0.14	0	0.05					
exang	-0.41	0.15	0.3	0.27	0.33	0.43	0.43						
cp	-0.39	0.05	0.25	0.21	0.32	0.48							
num	-0.45	0.26	0.46	0.4	0.54								
thal	-0.3	0.38	0.36	0.28									
slope	-0.44	0.03	0.59										
oldpeak	-0.44	0.12											
sex	-0.06												

- **Preparation (Data Cleaning)**
    - Handling missing values

      Figure 14 shows that the rows containing null values were removed using the na.omit() function. After printing the number of rows in the dataset, we are left with 299 rows. This is a significant decrease from 920 instances. The researchers realized that the statistical power of the dataset will be lower due to lesser population. However, the researchers are also considering the accuracy and quality of the results on the latter phase of this project, thus, we opted to remove the rows with null values first.

Figure 14

Removing all of the rows with null values using na.omit() function



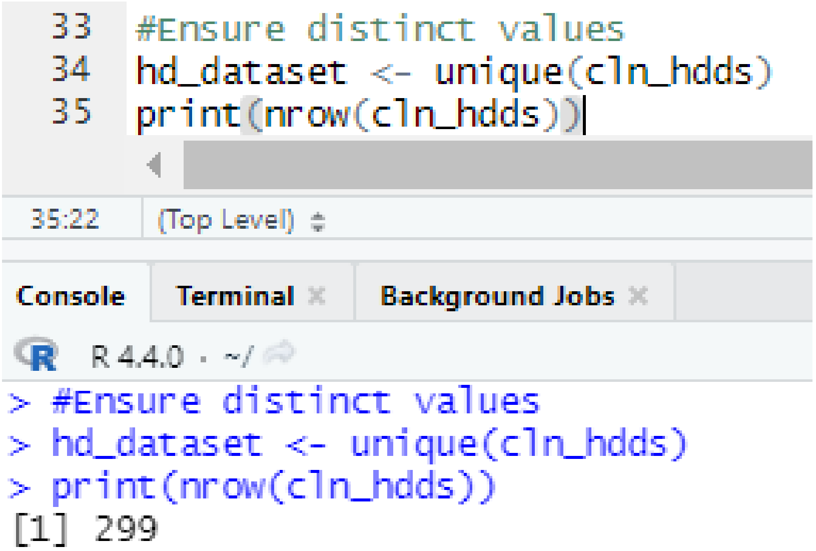
```
#Data Cleaning
cln_hdds <- na.omit (hd_dataset)
print(nrow(cln_hdds))
```

```
> cln_hdds <- na.omit (hd_dataset)
> print(nrow(cln_hdds))
[1] 299
```

- Handling duplicate rows

Figure 15

Using the unique() function to ensure that there are no duplicates



```
#Ensure distinct values
hd_dataset <- unique(cln_hdds)
print(nrow(cln_hdds))
```

```
> #Ensure distinct values
> hd_dataset <- unique(cln_hdds)
> print(nrow(cln_hdds))
[1] 299
```

- Ensuring correct data type

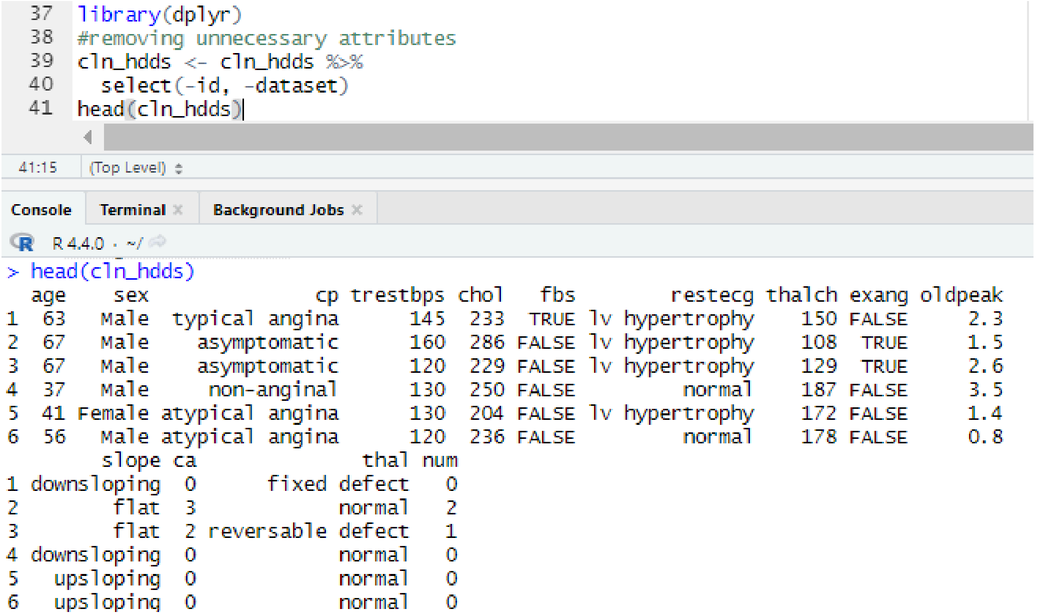
  Figure 5 shows the structure of the entire dataset along with the respective data types of attributes. Through observations, the researchers have identified that the data types for each attribute are correct.

- Identifying and removing unnecessary attributes

  Figure 12 shows how the select() function from the dplyr library was used. The ‘id’ and ‘dataset’ columns were removed as they are unnecessary in identifying the likelihood of having a heart disease.

Figure 16

Removing ‘ id’ and ‘dataset’ columns



```
library(dplyr)
#removing unnecessary attributes
cln_hdds <- cln_hdds %>%
  select(-id, -dataset)
head(cln_hdds)
```

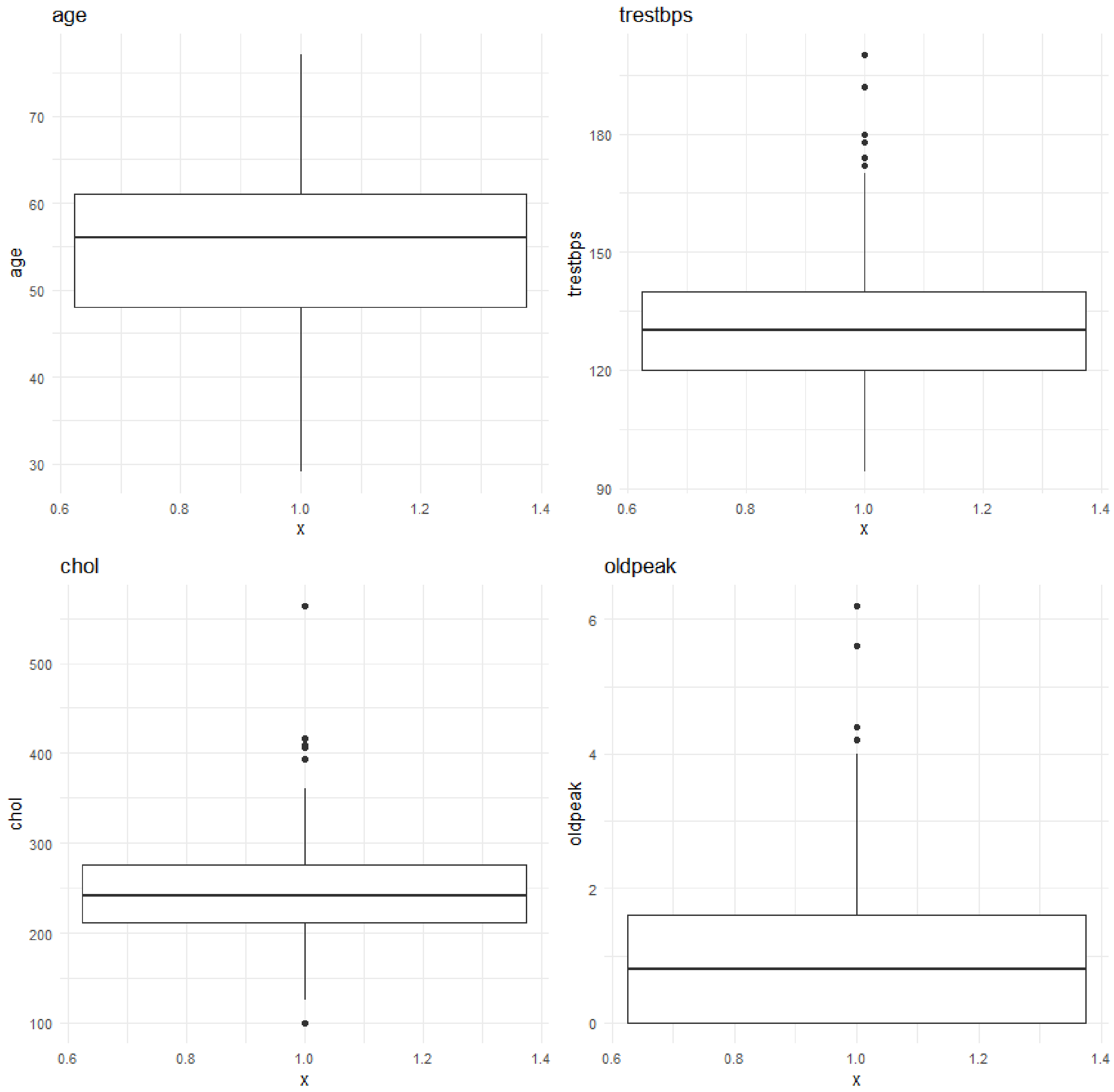
```
> head(cln_hdds)
  age    sex              cp trestbps chol   fbs        restecg thalch exang oldpeak
1  63   Male  typical angina      145  233  TRUE lv hypertrophy    150 FALSE     2.3
2  67   Male    asymptomatic      160  286 FALSE lv hypertrophy    108  TRUE     1.5
3  67   Male    asymptomatic      120  229 FALSE lv hypertrophy    129  TRUE     2.6
4  37   Male     non-anginal      130  250 FALSE         normal    187 FALSE     3.5
5  41 Female atypical angina      130  204 FALSE lv hypertrophy    172 FALSE     1.4
6  56   Male atypical angina      120  236 FALSE         normal    178 FALSE     0.8
        slope ca              thal num
1 downsloping  0      fixed defect   0
2        flat  3            normal   2
3        flat  2 reversable defect   1
4 downsloping  0            normal   0
5   upsloping  0            normal   0
6   upsloping  0            normal   0
```

- Handling outliers

To handle the outliers, the researchers have used box plots for numeric variables. The points outside the ‘whiskers’ represent the outliers. One point to consider, however, is that the origin of these data are medical institutions. It is likely a medical professional is present on the collection of the data. Thus, the outliers will not be removed but will be taken into account when we work further with the project.

Figure 17

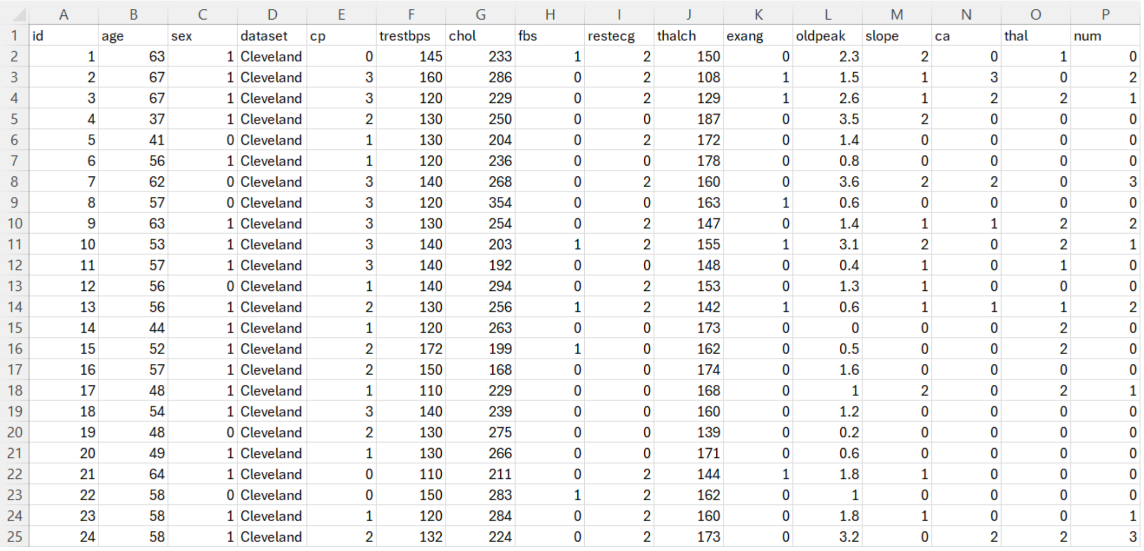
Visual representation of the outliers for numerical values using boxplot



- **Transformation**

Figure 18

Overview of the transformed dataset using Microsoft Excel



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	id	age	sex	dataset	cp	trestbps	chol	fbs	restecg	thalch	exang	oldpeak	slope	ca	thal	num
2	1	63	1	Cleveland	0	145	233	1	2	150	0	2.3	2	0	1	0
3	2	67	1	Cleveland	3	160	286	0	2	108	1	1.5	1	3	0	2
4	3	67	1	Cleveland	3	120	229	0	2	129	1	2.6	1	2	2	1
5	4	37	1	Cleveland	2	130	250	0	0	187	0	3.5	2	0	0	0
6	5	41	0	Cleveland	1	130	204	0	2	172	0	1.4	0	0	0	0
7	6	56	1	Cleveland	1	120	236	0	0	178	0	0.8	0	0	0	0
8	7	62	0	Cleveland	3	140	268	0	2	160	0	3.6	2	2	0	3
9	8	57	0	Cleveland	3	120	354	0	0	163	1	0.6	0	0	0	0
10	9	63	1	Cleveland	3	130	254	0	2	147	0	1.4	1	1	2	2
11	10	53	1	Cleveland	3	140	203	1	2	155	1	3.1	2	0	2	1
12	11	57	1	Cleveland	3	140	192	0	0	148	0	0.4	1	0	1	0
13	12	56	0	Cleveland	1	140	294	0	2	153	0	1.3	1	0	0	0
14	13	56	1	Cleveland	2	130	256	1	2	142	1	0.6	1	1	1	2
15	14	44	1	Cleveland	1	120	263	0	0	173	0	0	0	0	2	0
16	15	52	1	Cleveland	2	172	199	1	0	162	0	0.5	0	0	2	0
17	16	57	1	Cleveland	2	150	168	0	0	174	0	1.6	0	0	0	0
18	17	48	1	Cleveland	1	110	229	0	0	168	0	1	2	0	2	1
19	18	54	1	Cleveland	3	140	239	0	0	160	0	1.2	0	0	0	0
20	19	48	0	Cleveland	2	130	275	0	0	139	0	0.2	0	0	0	0
21	20	49	1	Cleveland	1	130	266	0	0	171	0	0.6	0	0	0	0
22	21	64	1	Cleveland	0	110	211	0	2	144	1	1.8	1	0	0	0
23	22	58	0	Cleveland	0	150	283	1	2	162	0	1	0	0	0	0
24	23	58	1	Cleveland	1	120	284	0	2	160	0	1.8	1	0	0	1
25	24	58	1	Cleveland	2	132	224	0	2	173	0	3.2	0	2	2	3

- **Merging and Clearing (Data Fine Tuning)**

As mentioned in the discussion in section III, the dataset used for this project is already a comprehensive dataset wherein the data from four different locations were aggregated. Thus, the merging step is deemed unnecessary for the project.

V. Modeling, Evaluation, and Validation

A. Modeling

- **Model Selection**

The Random Forest model is a great choice for predicting heart disease because of several important reasons. It is good at dealing with outliers, which are unusual data points that can mess up other models. This is important in medical data where some values might be extremely high or low. Random Forest builds multiple decision trees using different parts of the data, which helps it handle outliers well.

Another reason is that Random Forest can find complex patterns in the data. For example, heart disease risk might depend on a combination of factors like age, cholesterol levels, and lifestyle. Random Forest can understand these complicated relationships better than simpler models.

- **Feature Selection**

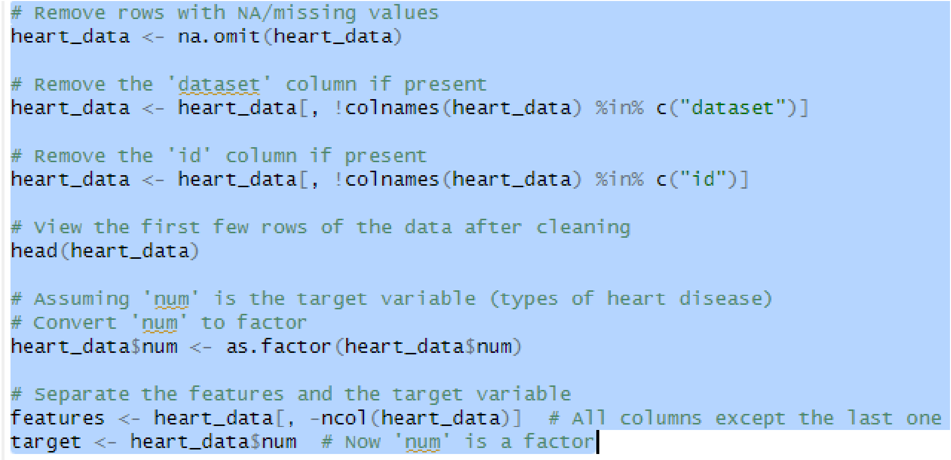
The researchers selected features for the Random Forest model using statistical methods, machine learning techniques, and domain knowledge. They removed irrelevant columns and handled missing values, converting the target variable 'num' to a factor. Statistical methods like correlation analysis identified significant numerical features, while feature importance from Random Forests prioritized impactful ones. Data from healthcare professionals ensured clinically relevant features like age, sex, cholesterol levels, and resting blood pressure were included, making the feature selection process thorough and effective.

- **Data Preprocessing**

To prepare the heart disease dataset for analysis, several steps were taken. First, any incomplete rows with missing values were removed to ensure the dataset was complete. Next, two columns, dataset, and id, were dropped as they didn't contribute to our analysis. The num column, which indicates different types of heart disease, was converted into a categorical variable to classify the types accurately. After this, the dataset was split into two parts: features (all columns except num) and the target variable (num) because this is the variable that represents the types of heart disease. This separation helped in clearly defining what we're predicting (heart disease types) and what we're using to make predictions (the dataset's features). This structured approach ensured our data was ready for building and evaluating predictive models

Figure 19.

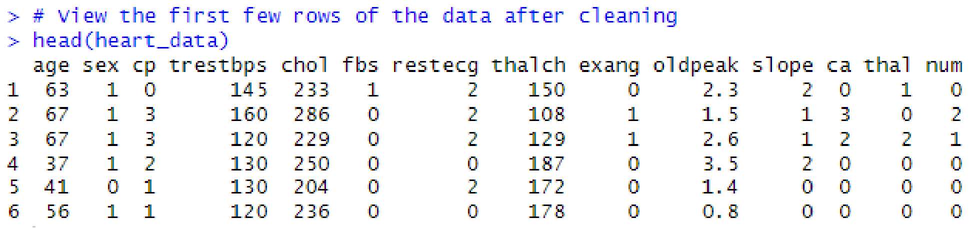
Removing the rows with missing values as well as the ‘dataset’ and ‘id’ columns then assigning feature and target variable/s



```
# Remove rows with NA/missing values
heart_data <- na.omit(heart_data)

# Remove the 'dataset' column if present
heart_data <- heart_data[, !colnames(heart_data) %in% c("dataset")]

# Remove the 'id' column if present
heart_data <- heart_data[, !colnames(heart_data) %in% c("id")]

# View the first few rows of the data after cleaning
head(heart_data)

# Assuming 'num' is the target variable (types of heart disease)
# Convert 'num' to factor
heart_data$num <- as.factor(heart_data$num)

# Separate the features and the target variable
features <- heart_data[, -ncol(heart_data)]  # All columns except the last one
target <- heart_data$num  # Now 'num' is a factor
```

Figure 20.

The first few rows of the data after cleaning



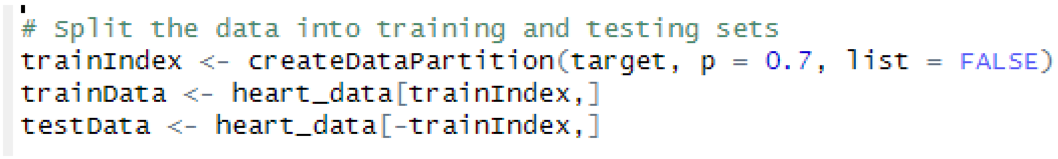
```
> # View the first few rows of the data after cleaning
> head(heart_data)
  age sex cp trestbps chol fbs restecg thalch exang oldpeak slope ca thal num
1  63   1  0      145  233   1       2    150     0     2.3     2  0    1   0
2  67   1  3      160  286   0       2    108     1     1.5     1  3    0   2
3  67   1  3      120  229   0       2    129     1     2.6     1  2    2   1
4  37   1  2      130  250   0       0    187     0     3.5     2  0    0   0
5  41   0  1      130  204   0       2    172     0     1.4     0  0    0   0
6  56   1  1      120  236   0       0    178     0     0.8     0  0    0   0
```

- **Model Training**

The dataset was split into training and testing sets using the c~function, where 70% of the data was allocated to training (trainData) and the remaining 30% to testing (testData).

Figure 21.

Splitting the data

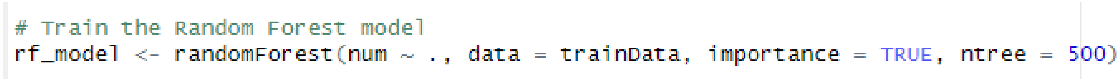


```
# Split the data into training and testing sets
trainIndex <- createDataPartition(target, p = 0.7, list = FALSE)
trainData <- heart_data[trainIndex,]
testData <- heart_data[-trainIndex,]
```

Next, a Random Forest model (rf_model) was trained using the randomForest function. This model predicts the num variable using all other

Figure 22.

Training the random forest model.



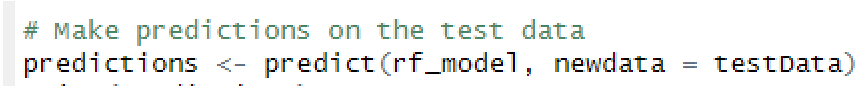
```
# Train the Random Forest model
rf_model <- randomForest(num ~ ., data = trainData, importance = TRUE, ntree = 500)
```

variables (~ .) in the trainData dataset. The model was configured with 500 trees (ntree = 500), and feature importance was computed (importance = TRUE).

After training, predictions were made on the test data (testData) using the predict function based on the rf_model.

Figure 23.

Making predictions on the test data.

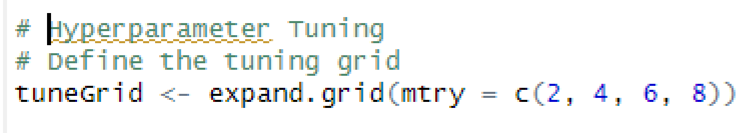


```
# Make predictions on the test data
predictions <- predict(rf_model, newdata = testData)
```

Hyperparameter tuning was performed using cross-validation (trainControl(method = "cv", number = 5)) to optimize the Random Forest model's performance. The tuning grid (tuneGrid) specified different values of mtry (number of variables randomly sampled at each split), allowing the train function to select the best parameters (tuned_rf_model).

Figure 24.

Tuning and defining the tuning grid.

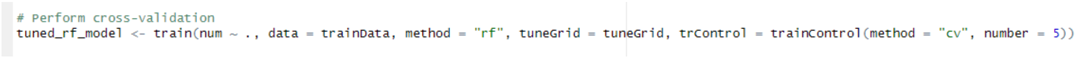


```
# Hyperparameter Tuning
# Define the tuning grid
tuneGrid <- expand.grid(mtry = c(2, 4, 6, 8))
```

Performing cross-validation on the tuned random forest model to make sure it is good at predicting values.

Figure 25.

Performing cross-validation.



```
# Perform cross-validation
tuned_rf_model <- train(num ~ ., data = trainData, method = "rf", tuneGrid = tuneGrid, trControl = trainControl(method = "cv", number = 5))
```

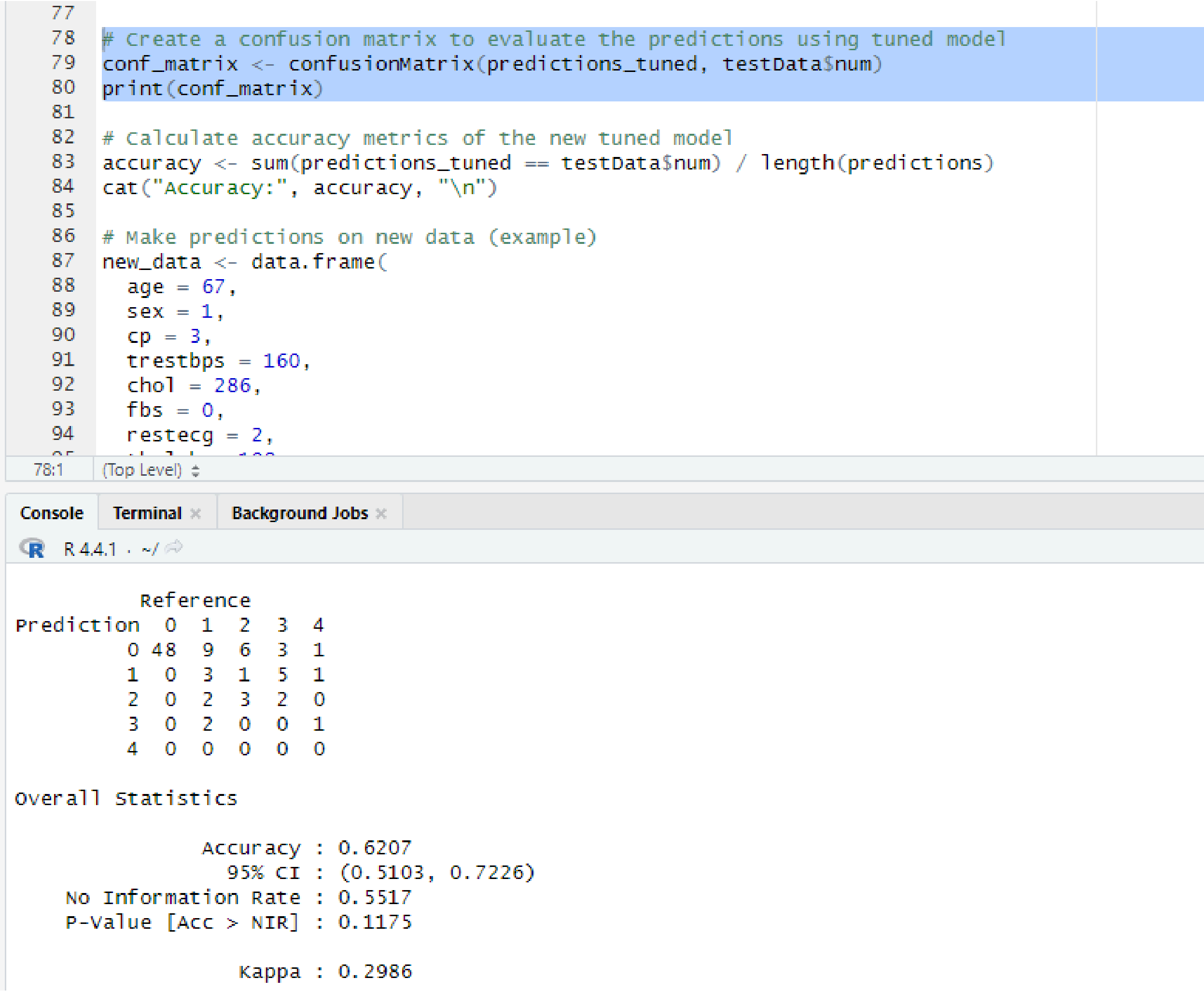
B. Evaluation of the Model Result

- **Evaluation Metrics**

The researchers evaluate model performance using a confusion matrix; the given metrics provide insights into how effectively the model predicts heart disease. The model showed 62% accuracy

Figure 26.

Creating a confusion matrix for the evaluation of predictions



```r

# Create a confusion matrix to evaluate the predictions using tuned model
conf_matrix <- confusionMatrix(predictions_tuned, testData$num)
print(conf_matrix)

# Calculate accuracy metrics of the new tuned model
accuracy <- sum(predictions_tuned == testData$num) / length(predictions)
cat("Accuracy:", accuracy, "\n")

# Make predictions on new data (example)
new_data <- data.frame(
  age = 67,
  sex = 1,
  cp = 3,
  trestbps = 160,
  chol = 286,
  fbs = 0,
  restecg = 2,
```

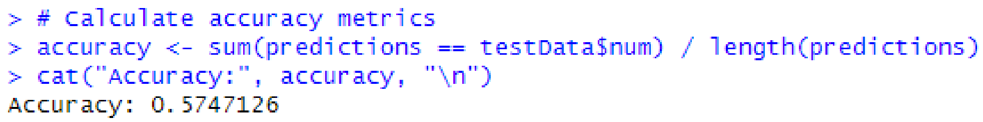
```
          Reference
Prediction  0  1  2  3  4
         0 48  9  6  3  1
         1  0  3  1  5  1
         2  0  2  3  2  0
         3  0  2  0  0  1
         4  0  0  0  0  0

Overall Statistics

               Accuracy : 0.6207
                 95% CI : (0.5103, 0.7226)
    No Information Rate : 0.5517
    P-Value [Acc > NIR] : 0.1175

                  Kappa : 0.2986
```

- **Performance Assessment**

  The researchers will evaluate model performance using metrics such as accuracy, precision, recall, and F1-score for classification tasks. This evaluation will be based on predictions made by the random forest model trained on the heart disease dataset.

Figure 27.

Calculating the accuracy

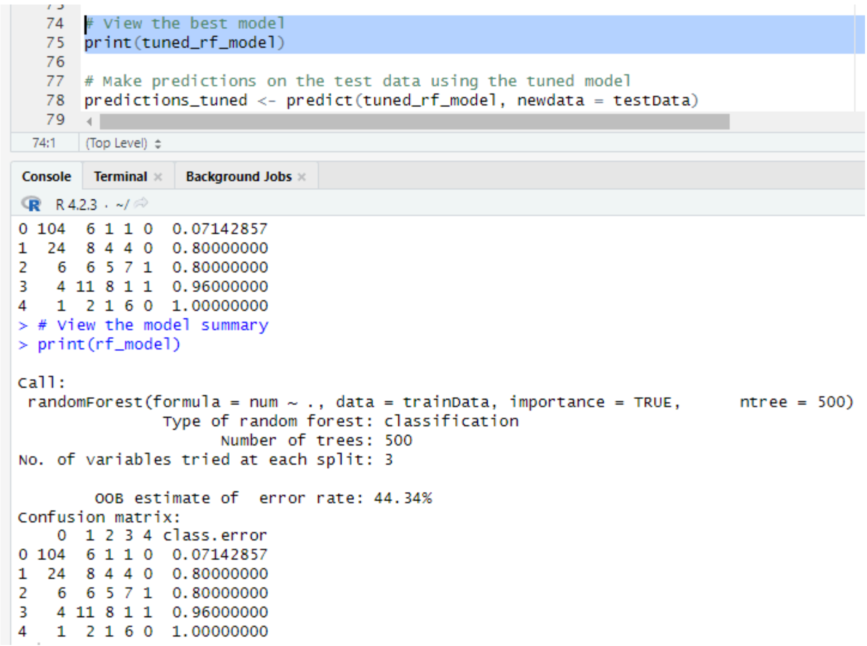


```
> # Calculate accuracy metrics
> accuracy <- sum(predictions == testData$num) / length(predictions)
> cat("Accuracy:", accuracy, "\n")
Accuracy: 0.5747126
```

- **Error Analysis**

The researchers used a confusion matrix for the model’s error analysis.

Figure 28.



```
# View the best model
print(tuned_rf_model)

# Make predictions on the test data using the tuned model
predictions_tuned <- predict(tuned_rf_model, newdata = testData)
```

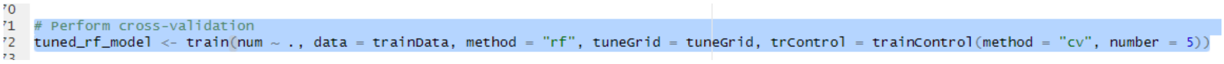
```
0 104   6 1 1 0   0.07142857
1  24   8 4 4 0   0.80000000
2   6   6 5 7 1   0.80000000
3   4 11 8 1 1   0.96000000
4   1   2 1 6 0   1.00000000
> # View the model summary
> print(rf_model)

Call:
 randomForest(formula = num ~ ., data = trainData, importance = TRUE,      ntree = 500)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 44.34%
Confusion matrix:
    0  1 2 3 4 class.error
0 104   6 1 1 0   0.07142857
1  24   8 4 4 0   0.80000000
2   6   6 5 7 1   0.80000000
3   4 11 8 1 1   0.96000000
4   1   2 1 6 0   1.00000000
```

C. Validation

- **Validation Techniques**

  The researchers used a train-test split to validate the model. Moreover, cross-validation was conducted to tune the and improve the model making it robust and accurate.

Figure 29.

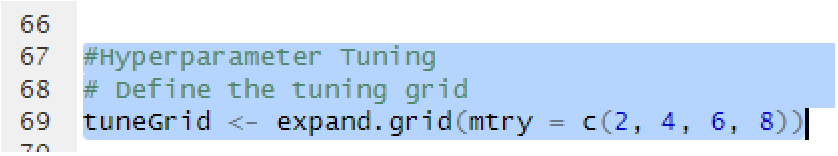


```
# Perform cross-validation
tuned_rf_model <- train(num ~ ., data = trainData, method = "rf", tuneGrid = tuneGrid, trControl = trainControl(method = "cv", number = 5))
```

- **Overfitting and Underfitting**

  The researcher employs the mtry parameter in the random forest model to mitigate overfitting. mtry controls the number of variables randomly sampled at each split of a tree. By limiting the number of features considered for each decision split, the model's complexity is managed.

Figure 30.



```

#Hyperparameter Tuning
# Define the tuning grid
tuneGrid <- expand.grid(mtry = c(2, 4, 6, 8))
```

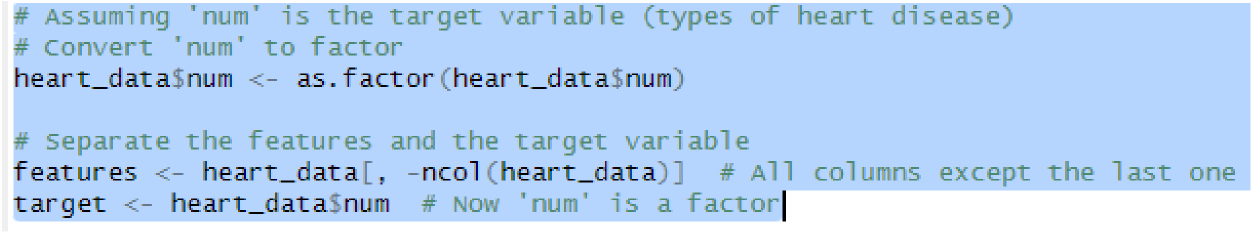
- **External Validation**

  The utilized dataset in the study is already a combined data, hence, the researcher did not have the need to integrate more data resources.

- **Model Interpretability**

Model interpretability was achieved by making the num attribute a factor in which the model treats it as an ordered categorical variable. The features were also separated to specify the variables used to identify if the person has a heart disease or not.

Figure 31.



```
# Assuming 'num' is the target variable (types of heart disease)
# Convert 'num' to factor
heart_data$num <- as.factor(heart_data$num)

# Separate the features and the target variable
features <- heart_data[, -ncol(heart_data)]  # All columns except the last one
target <- heart_data$num  # Now 'num' is a factor
```

VI. Knowledge Presentation and Visualization

Purpose of Knowledge Presentation and Visualization

In a project focused on predicting heart disease, the effective presentation and visualization of data are crucial for making the insights accessible and understandable. Clear and engaging visualizations help healthcare professionals and medical researchers quickly and accurately interpret risk factors, aiding in informed clinical decisions and research. For patients, these visualizations enhance their understanding of their health status and potential risks.

Visualization Tools and Libraries

- ggplot2 - This library was imported to use the ggplot() function which is necessary for performing different types of plots such as histogram, scatter plot, and stacked bar plot.
- ggcorrplot - This library was imported to use the ggcorrplot() function which is necessary to perform the correlation matrix on the part of data exploration.
- gridExtra - This library was imported to neatly display multiple plots at once in a grid-like structure.

Types of Visualizations

- **Charts and Graphs:**
    - Histogram - This graph was used to show the distribution of age within the dataset.

- Scatter Plot - This graph was used to plot the relationship of two numerical attributes in the dataset.
- Stacked Bar Plot - This graph was used to show the distribution of different stages of heart disease based on sex.
- Correlation Matrix - This matrix was used to plot the relationship of all numerical attributes in table form. This table shows the r value between two attributes. If the value is closer to one (1), then there is a stronger likelihood that these attributes affect one another.

A. Summary

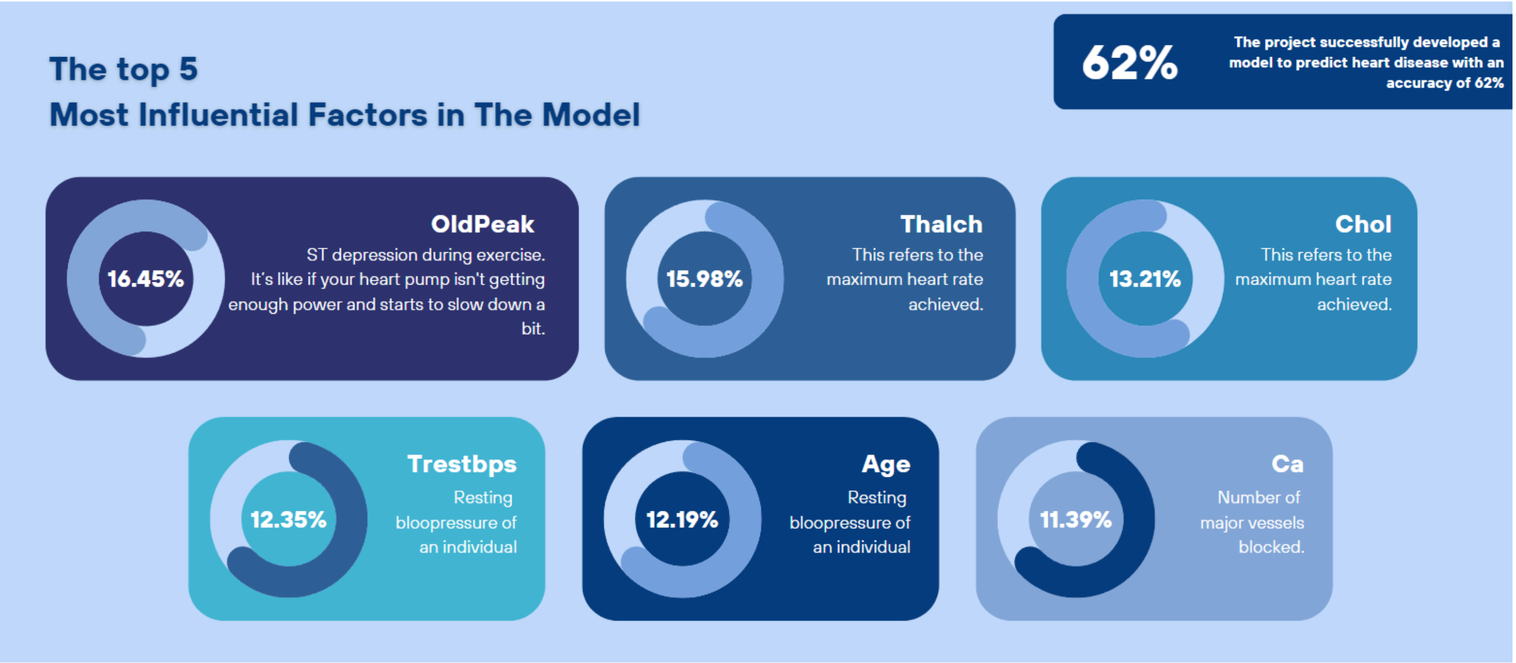
The researchers utilized a dataset containing various attributes such as age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, and num. These features were employed to predict the presence of heart disease. Random Forest was chosen as the modeling technique due to its ability to identify patterns and relationships within noisy datasets. This choice was supported by the researchers' findings that the attributes exhibit weak relationships. The model had a 62% accuracy in predicting heart disease which makes it a reliable model for predicting heart disease.The top factors influencing heart disease prediction are Oldpeak (ST depression during exercise), Trestbps (resting blood pressure), Age, Thalch (maximum heart rate achieved), and Ca (number of major vessels blocked), and Cholesterol These features are critical indicators in assessing heart disease risk in the model.

B. Recommendation

The project successfully developed a model to predict heart disease with an accuracy of 62%. However, future enhancements are advisable and should extend beyond the current features. Researchers recommend collecting more data to enhance accuracy and performance. Future research should consider incorporating additional factors such as family history and lifestyle. Given the challenges in identifying relationships between different attributes, expanding the dataset is crucial to address any imbalances and biases. Collaborating with medical professionals for evaluations and expert insights is also highly recommended.

Figure 32.

Data Story Deck: Top 5 Most Influential Factors in The Model



The top 5
Most Influential Factors in The Model

62% The project successfully developed a model to predict heart disease with an accuracy of 62%

16.45% **OldPeak**
ST depression during exercise. It's like if your heart pump isn't getting enough power and starts to slow down a bit.

15.98% **Thalch**
This refers to the maximum heart rate achieved.

13.21% **Chol**
This refers to the maximum heart rate achieved.

12.35% **Trestbps**
Resting bloopressure of an individual

12.19% **Age**
Resting bloopressure of an individual

11.39% **Ca**
Number of major vessels blocked.

VII. Source Code and Sample Output

```
library(readr)

#Handle missing values

hd_dataset <-
read.csv(file="heart_disease_uci.csv",
na.strings = c(".", "NA", ""))

tail(hd_dataset, 10)

#Ensure distinct values

hd_dataset <- unique(hd_dataset)

head(hd_dataset)

str(hd_dataset)

cat("Dataset Shape: ", dim(hd_dataset), "\n")

cat("\nMissing Values:\n")

sapply(hd_dataset, function(x)
sum(is.na(x)))

hdnum_summary <- summary(hd_dataset[,
c('age', 'trestbps', 'chol', 'thalch', 'oldpeak')])

hdnum_summary

hdcat_summary <- c('sex', 'cp', 'fbs',
'restecg', 'exang', 'slope', 'ca', 'num')

for (var in hdcat_summary) {

  cat("\nFrequency counts for", var, ":\n")

  print(table(hd_dataset[[var]], useNA =
"ifany"))

}
```

```
#Data Cleaning

cln_hdds <- na.omit (hd_dataset)

#Ensure distinct values

hd_dataset <- unique(cln_hdds)

print(nrow(cln_hdds))

#removing unnecessary attributes

library(dplyr)

cln_hdds <- cln_hdds %>%

  select(-id, -dataset)

head(cln_hdds)

#handling outliers for numerical values

library(ggplot2)

library(gridExtra)

numeric_attributes <- c("age", "trestbps",
"chol", "oldpeak")

# Create a list to store plots

plots <- lapply(numeric_attributes,
function(var) {

  ggplot(data = cln_hdds, aes(x = 1, y =
!!sym(var))) +

    geom_boxplot() +
    labs(title = var) +
    theme_minimal()
})
grid.arrange(grobs = plots, ncol = 2)
#Descriptive Analysis
library(ggplot2)
library(gridExtra)
#age histogram
ggplot(cln_hdds, aes(x = age)) +
  geom_histogram(binwidth=5, fill = "blue", color = "black", alpha = 0.7) +
  ggtitle("Age Distribution vs. Frequency") +
  xlab("Age") + ylab("Frequency")
summary(cln_hdds$age)
library(ggplot2)
library(gridExtra)
#age vs trestbps
atr_spear <- cor(cln_hdds$age, cln_hdds$trestbps, method = "spearman")
print(paste("Spearman Correlation:", atr_spear))
atr_cor <- cor(cln_hdds$age, cln_hdds$trestbps, method = "pearson")
print("Pearson Correlation:")
print(atr_cor)
atr_plot <- ggplot(cln_hdds, aes(x = age, y = trestbps)) +
  geom_point() +  # Scatter plot
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add linear trend line
  labs(title = "Age vs. Resting Blood Pressure (trestbps)",
       x = "Age", y = "Resting Blood Pressure (trestbps)") +
  theme_minimal()
#age vs cholesterol
ac_spear <- cor(cln_hdds$age, cln_hdds$chol, method = "spearman")
print(paste("Spearman Correlation:", ac_spear))
ac_cor <- cor(cln_hdds$age, cln_hdds$chol, method = "pearson")
print("Pearson Correlation:")
print(ac_cor)
ac_plot <- ggplot(cln_hdds, aes(x = age, y = chol)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add linear trend line
  labs(title = "Age vs. Serum Cholesterol (chol)",
      x = "Age", y = "Serum Cholesterol (chol)") +

  theme_minimal()

#age vs thalch

atc_spear <- cor(cln_hdds$age, cln_hdds$thalch, method = "spearman")

print(paste("Spearman Correlation:", atc_spear))

atc_cor <- cor(cln_hdds$age, cln_hdds$thalch, method = "pearson")

print("Pearson Correlation:")

print(atc_cor)

atc_plot <- ggplot(cln_hdds, aes(x = age, y = thalch)) +

  geom_point() +

  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add linear trend line

  labs(title = "Age vs. Maximum Heart Rate Achieved (thalch)",

      x = "Age", y = "Maximum Heart Rate Achieved (thalch)") +

  theme_minimal()

#trestbps vs chol

trc_spear <- cor(cln_hdds$trestbps, cln_hdds$chol, method = "spearman")

print(paste("Spearman Correlation:", trc_spear))

trc_cor <- cor(cln_hdds$trestbps, cln_hdds$chol, method = "pearson")

print("Pearson Correlation:")

print(trc_cor)

trc_plot <- ggplot(cln_hdds, aes(x = trestbps, y = chol)) +

  geom_point() +

  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add linear trend line

  labs(title = "Resting Blood Pressure (trestbps) vs.Serum Cholesterol(chol)",

      x = "Resting Blood Pressure (trestbps)", y = "Serum Cholesterol(chol)") +

  theme_minimal()

#chol vs thalach

cht_spear <- cor(cln_hdds$chol, cln_hdds$thalch, method = "spearman")

print(paste("Spearman Correlation:", cht_spear))

cht_cor <- cor(cln_hdds$chol, cln_hdds$thalch, method = "pearson")

print("Pearson Correlation:")

print(cht_cor)

cht_plot <- ggplot(cln_hdds, aes(x = chol, y = thalch)) +

  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add linear trend line
  labs(title = "Serum Cholesterol(chol) vs. Maximum Heart Rate Achieved(thalch)",
       x = "Serum Cholesterol(chol)", y = "Maximum Heart Rate Achieved(thalch)") +
  theme_minimal()
#oldpeak vs thalach
opt_spear <- cor(cln_hdds$oldpeak, cln_hdds$thalch, method = "spearman")
print(paste("Spearman Correlation:", opt_spear))
opt_cor <- cor(cln_hdds$oldpeak, cln_hdds$thalch, method = "pearson")
print("Pearson Correlation:")
print(opt_cor)
opt_plot <- ggplot(cln_hdds, aes(x = oldpeak, y = thalch)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +  # Add linear trend line
  labs(title = "ST Depression Induced by Exercise(oldpeak) vs. Maximum Heart Rate Achieved(thalch)",
       x = "ST Depression Induced by Exercise(oldpeak))", y = "Maximum Heart Rate Achieved(thalch)") +
  theme_minimal()
grid.arrange(atr_plot, ac_plot, atc_plot, trc_plot, cht_plot, opt_plot, ncol = 3)
#correlation matrix
library(ggcorrplot)
numerical_vars <- sapply(cln_hdds, is.numeric)
numerical_data <- cln_hdds[, numerical_vars]
correlation_matrix <- cor(numerical_data, use = "complete.obs", method = "spearman")
cor_vis <- ggcorrplot(correlation_matrix,
                      hc.order = TRUE,
                      type = "upper",
                      lab = TRUE,
                      lab_size = 3,
                      method = "square",
                      colors = c("#ffb346", "#a37ac2"),
                      title = "Correlation Matrix of values",
                      ggtheme = theme_minimal())
cor_vis
library(ggplot2)

cln_hdds$num <- factor(cln_hdds$num)
```

```
ggplot(cln_hdds, aes(x = sex, fill = num)) +

  geom_bar(position = "stack", color =
"black") +

  labs(title = "Stacked Bar Plot of Sex and
Heart Disease",

       x = "Sex", y = "Count") +

  scale_fill_manual(values = c("0" =
"purple", "1" = "orange", "2" = "yellow",
"3" = "green", "4" = "blue"), name = "Heart
Disease") +

  theme_minimal() +

  geom_text(stat = "count", aes(label =
..count..), position = position_stack(vjust =
0.5), color = "white")+

  theme(

    axis.text.x = element_text(size = 12),

    axis.text.y = element_text(size = 12)

  )

head(cln_hdds)

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

#RANDOM FOREST START

if (!requireNamespace("randomForest",
quietly = TRUE)) {

  install.packages("randomForest")

}

if (!requireNamespace("caret", quietly =
TRUE)) {

  install.packages("caret")

}

# Load necessary libraries

library(randomForest)

library(caret)

# Import the CSV file

heart_data <-
read.csv("heart_disease_transformed.csv")

# View the first few rows of the data

head(heart_data)

# Remove rows with NA/missing values

heart_data <- na.omit(heart_data)

# Remove the 'dataset' column if present

heart_data <- heart_data[,
!colnames(heart_data) %in% c("dataset")]

# Remove the 'id' column if present

heart_data <- heart_data[,
!colnames(heart_data) %in% c("id")]

# View the first few rows of the data after
cleaning

head(heart_data)



# Assuming 'num' is the target variable
(types of heart disease)

# Convert 'num' to factor
```

```r
heart_data$num <- as.factor(heart_data$num)

# Separate the features and the target variable

features <- heart_data[, -ncol(heart_data)]  # All columns except the last one

target <- heart_data$num  # Now 'num' is a factor

# Set seed for reproducibility

set.seed(123)

# Split the data into training and testing sets

trainIndex <- createDataPartition(target, p = 0.7, list = FALSE)

trainData <- heart_data[trainIndex,]

testData <- heart_data[-trainIndex,]

# Train the Random Forest model

rf_model <- randomForest(num ~ ., data = trainData, importance = TRUE, ntree = 500)

# View the model summary

print(rf_model)

# Make predictions on the test data

predictions <- predict(rf_model, newdata = testData)



# Create a confusion matrix to evaluate the predictions

conf_matrix <- confusionMatrix(predictions, testData$num)

print(conf_matrix)



# Calculate accuracy metrics

accuracy <- sum(predictions == testData$num) / length(predictions)

cat("Accuracy:", accuracy, "\n")



# (Optional) Hyperparameter Tuning

# Define the tuning grid

tuneGrid <- expand.grid(mtry = c(2, 4, 6, 8))

# Perform cross-validation

tuned_rf_model <- train(num ~ ., data = trainData, method = "rf", tuneGrid = tuneGrid, trControl = trainControl(method = "cv", number = 5))

# View the best model

print(tuned_rf_model)

# Make predictions on the test data using the tuned model

predictions_tuned <- predict(tuned_rf_model, newdata = testData)

# Create a confusion matrix to evaluate the predictions using tuned model
```

```
conf_matrix <-
confusionMatrix(predictions_tuned,
testData$num)

print(conf_matrix)

# Calculate accuracy metrics of the new
tuned model

accuracy <- sum(predictions_tuned ==
testData$num) / length(predictions)

cat("Accuracy:", accuracy, "\n")

# Make predictions on new data (example)

new_data <- data.frame(

  age = 67,

  sex = 1,

  cp = 3,

  trestbps = 160,

  chol = 286,

  fbs = 0,

  restecg = 2,

  thalch = 108,

  exang = 1,

  oldpeak = 1.5,

  slope = 1,

  ca = 3,

  thal = 0

)

# Predict the outcome

new_prediction <- predict(tuned_rf_model,
newdata = new_data)

print(new_prediction)
```

VIII. References

Ahmad, A. A., & Polat, H. (2023). Prediction of heart disease based on machine learning using Jellyfish optimization algorithm. Diagnostics, 13(14), 2392. https://doi.org/10.3390/diagnostics13142392

American Heart Association. (2020). Types of heart disease tests. https://www.heart.org/en/health-topics/heart-attack/diagnosing-a-heart-attack/types-of-heart-disease-tests

Dattani, S., Samborska, V., Ritchie, H., & Roser, M. (2023, December 28). Cardiovascular diseases. Our World in Data. https://ourworldindata.org/cardiovascular-diseases

Heart and Stroke Foundation of Canada. (2020). Heart disease diagnosis and treatment. https://www.heartandstroke.ca/heart/diagnosis

Jindal, H., Agrawal, S., Khera, R., Jain, R., & Nagrath, P. (2021). Heart disease prediction using machine learning algorithms. IOP Conference Series. Materials Science and Engineering, 1022(1), 012072. https://doi.org/10.1088/1757-899x/1022/1/012072

Mayo Clinic. (2022). Heart disease diagnosis. https://www.mayoclinic.org/diseases-conditions/heart-disease/diagnosis-treatment/drc-20353124

Nadakinamani, R. G., Reyana, A., Kautish, S., Vibith, A. S., Gupta, Y., Abdelwahab, S. F., & Mohamed, A. W. (2022). Clinical data analysis for prediction of cardiovascular disease using machine learning techniques. Computational Intelligence and Neuroscience, 2022, 1–13. https://doi.org/10.1155/2022/2973324

Philippine Statistics Authority (2024). Causes of death in the Philippines https://www.psa.gov.ph/system/files/vsd/2-Press%20Release_2023%20Cause%20 of%20Death%20Statistics_as%20of%2031%20October%202023_RMRQ_ONS-signed.pdf

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. *New England Journal of Medicine, 380*(14), 1347-1358. https://doi.org/10.1056 /NEJMra1814259

Roser, M. (2023, December 28). Causes of death globally: what do people die from? Our World in Data. https://ourworldindata.org/causes-of-death-treemap

Shu, T. (Et.al). (2017). Effective heart disease detection based on quantitative computerized traditional Chinese medicine using representation-based classifiers. Evidence-based Complementary and Alternative Medicine, 2017, 1–10. https://doi.org/10.1155/2017/7483639

Tumanan-Mendoza (Et.al). (2018). Economic burden of hospitalization for congestive heart failure among adults in the Philippines. *Heart Asia*, *10*(2), e011039. https://doi.org/10.1136/heartasia-2018-011039

IX. Members' Detailed Contribution

Name	Detailed Tasks
Casile, Jasper Riley P. 	● Documentation ● III. Data Sample, Extraction, and Data Mining ● IV. Data Exploration, Preparation, and Transformation ● VI. Knowledge Presentation and Visualization ○ Visualization Tools and Libraries ○ Types of Visualizations ● VIII. References
Ongsiako, Cailo Nehru P. 	● Documentation ● II. Business Understanding ○ A. Background of the Study ○ B. Current metrics, trends, or dashboard ● V. Modeling, Evaluation, and Validation ● VI. Knowledge Presentation and Visualization ○ Summary ○ Recommendations ● VIII. References
Paragas, Veronica Maxine D. 	● Documentation ● I. Introduction ● II. Business Understanding ○ B. Current metrics, trends, or dashboard ○ C. Statement of the Problems ● PPT Presentation ● V. Modeling, Evaluation, and Validation ● VI. Knowledge Presentation and Visualization ○ Data Story Deck

	● VIII. References