General Sir John Kotelawala Defense University

Faculty of Management, Social Sciences and Humanities Department of Languages

BSc in Applied Data Science Communication

Intake 40

Year 2: Semester 1

Fundamentals of Data Mining: LB 1214 (Assignment 1)

Association Rule Mining &

Logistics Regression

Group Members

- D/ADC/23/0010 - C.S.Rathnasekara
- D/ADC/23/0011 -A.M.K.D.Bandara
- D/ADC/23/0019 - B.K.M.M.Hansanee
- D/ADC/23/0027 -P.G.D.Rodrigo

Table of Contents

Association rule mining

Discover the product relationships in the groceries using MBA association Rule Mining.

1.1. INTRODUCTION

A data mining technique called association rule mining helps for intriguing connections or interactions between objects in large datasets. Its focus is to find patterns in transactional data, like online click streams, client purchases, and medical records.

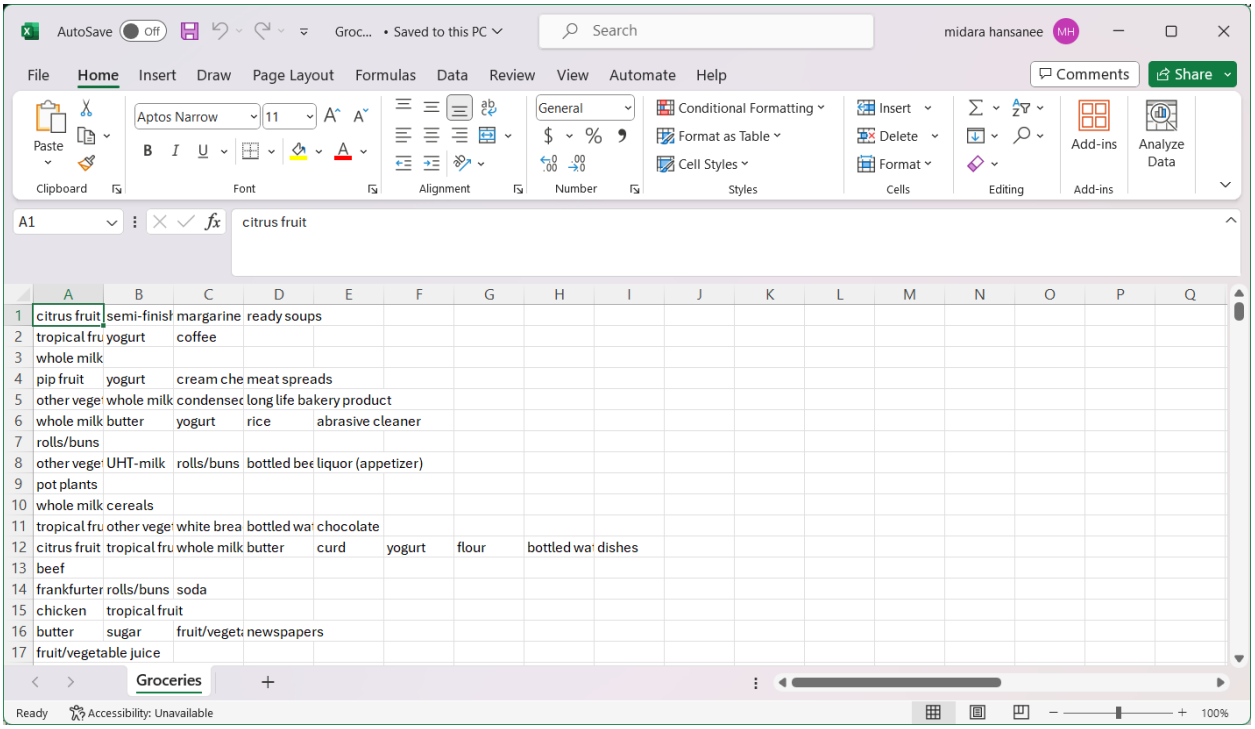
Association rule mining exposes hidden relationships between objects, assisting organizations in making data-driven decisions.
Market basket Analysis in association rule mining used to utilize consumer buying patterns. It looks via transaction history, such as checkout receipts, to determine which items that customers usually purchase together. This indicates trends in what people typically include in their "basket" at the same time.
This method mainly employed for retails to make the most of cross-selling, product placement, and promotions, In Healthcare to determine the correlations between diseases and treatment practices, fraud detection to look for odd activity when transacting with money.

1.2. DATASET

People purchase various types of goods in their day to day lives from grocery stores according to their needs and wants. There are some of the item's customers buy frequently and there are items bought infrequently. These purchases don't give any information regarding what goods customers always buy. Studying on the data set provides knowledge about consumer income, expenditure, and lifestyle etc.

The Groceries dataset contains information on 9835 receipts that represent the transaction of items that customers purchase from a supermarket by one receipt for each customers including the purchased items separately.

Each row defines a transaction while column in row represents an item.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	citrus fruit	semi-finish	margarine	ready soups													
2	tropical fru	yogurt	coffee														
3	whole milk																
4	pip fruit	yogurt	cream che	meat spreads													
5	other vege	whole milk	condensec	long life bakery product													
6	whole milk	butter	yogurt	rice	abrasive cleaner												
7	rolls/buns																
8	other vege	UHT-milk	rolls/buns	bottled bee	liquor (appetizer)												
9	pot plants																
10	whole milk	cereals															
11	tropical fru	other vege	white brea	bottled wa	chocolate												
12	citrus fruit	tropical fru	whole milk	butter	curd	yogurt	flour	bottled wa	dishes								
13	beef																
14	frankfurter	rolls/buns	soda														
15	chicken	tropical fruit															
16	butter	sugar	fruit/veget	newspapers													
17	fruit/vegetable juice																

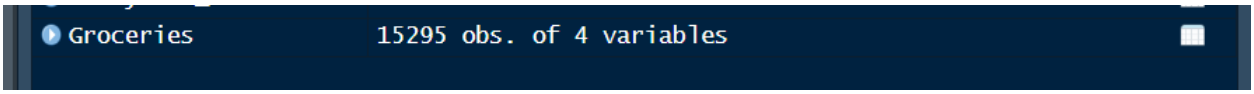
1.3. Explanation and preparation of Datasets

Data pre-proccessing

Data processing is known as the method of converting unprocessed data into processed data.
One of the main components of data mining is data processing. It is prominent to check for null values, inconsistent data, noisy data. The values in the groceries data set are easy to analyze since association rule mining technique usually deal with categorical data,
The values in the groceries data already consists with clean, Not -null values.

Explanation

This data contains 15295 items in 4 variables.



1.4. Data mining

The Association Rule of Data Mining was developed by Agrawal in 1993.
Data mining is a technique that is used in computer and statistical sciences to extract patterns and insights from massive databases. Insightful data analytics from data mining have enhanced business decision-making.
The interesting associations and relations between sizable data sets can be find using Association rule mining.
Market basket analysis is one of the key techniques used by large databases to determine customer purchase behaviors and relationships among items people purchases frequently.

1.5. Implementation of R

To apply association rule of data mining R plat form is used in this assignment. For statistical analysis, data manipulation, and visualization, R is a powerful tool.

R packages

```
4  #load and load thee relavent packages
5
6  install.packages("arules")
7  install.packages("arulesViz")
8
9  library(arules)
10 library(arulesViz)
11 library(datasets)
12 library(ggplot2)
```

“arules” - arules package offers a powerful framework for representing, modifying, and analyze transaction data and patterns.it focuses on frequent items and association rules.most commonly used package in rule mining.

“aruleViz”- The arulesViz package offers a variety of visualization methods for association rules and frequent item sets. It maximize package’s capabilities by providing visualizations that help users to better understand of the outcomes.

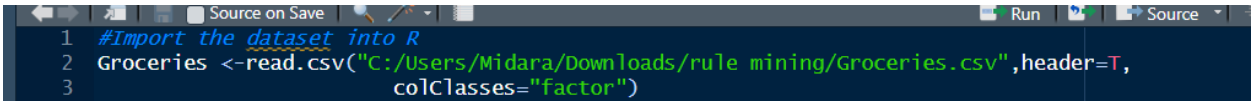
“data” -provides functions to work with the data set.

“ggplot2” – A famous package built in R for the visualization the grammar of graphics that help to understand systems.

1.5.1.set the working directory

session→ set working directory→ choose Working Directory

1.5.2. Import the dataset into R



1.5.3. Inspect the data set.

The parameters of the data set are displayed below as R codes after the file data set as been imported into R

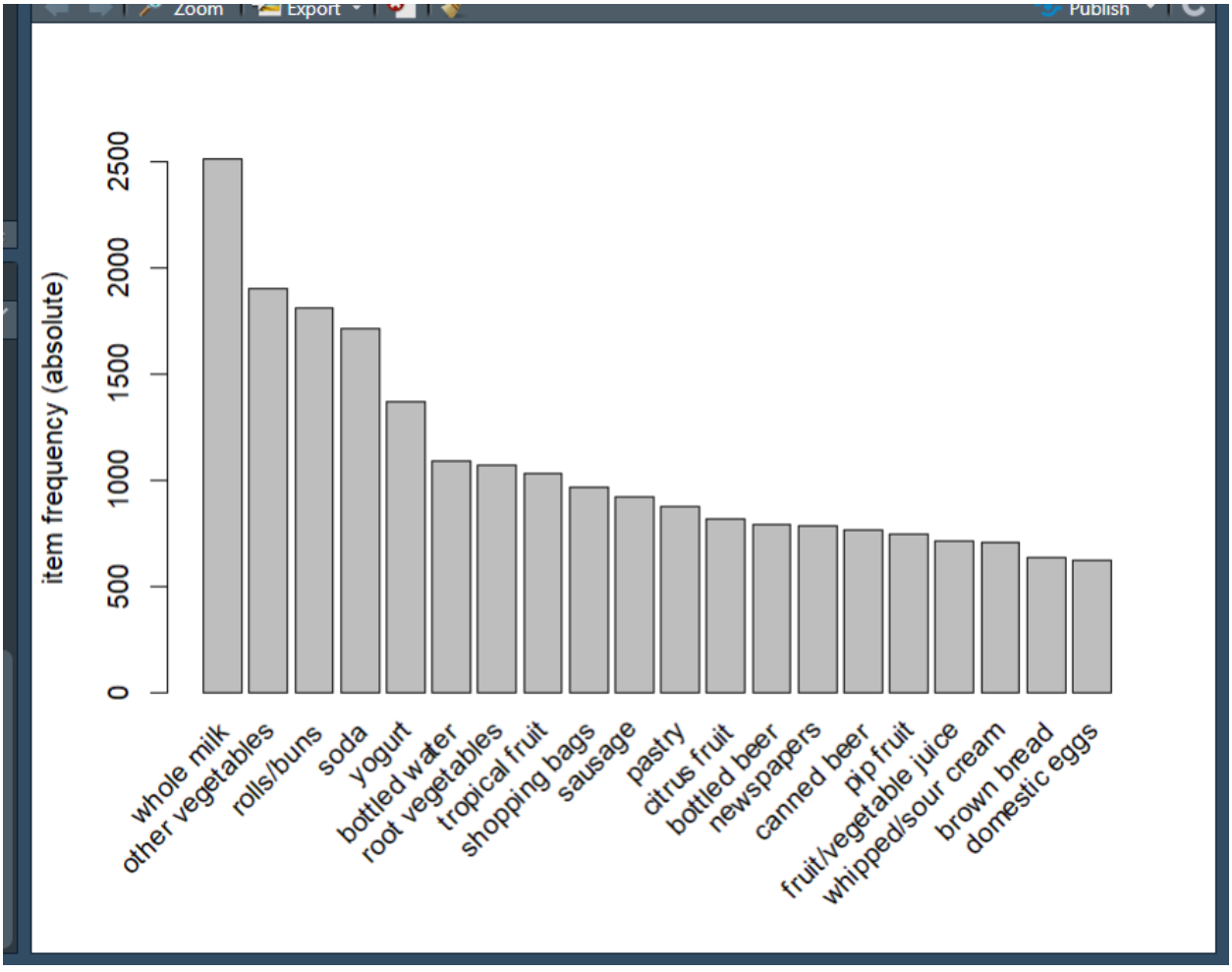
names (Groceries)
head(Groceries)
tail(Groceries)
summary(Groceries)
str(Groceries)
dim(Groceries)

1.5.4. Load the dataset.

```
13
14  #load the Data set
15  data(Groceries)
16
```

1.5.5. Item frequency plot

The item frequency plot defines the frequencies of the items within the dataset. The plot shows the topmost N items in the dataset.




```
17  #Plot and explore the  distribution of the "Purchases " data set
18  itemFrequencyPlot(Groceries,topN=20,type="absolute")
19
```

1.5.6. Apply Apriori Function

A vital data mining method called association rule learning to generate rules is based on the Apriori algorithm. It's based on the idea that every subset of a frequent itemset must likewise be frequent.

Apriori () function executes all the iterations at once.

```
20  #Apply Apriori function- rules
21  rules<-apriori(Groceries)
```

```
> rules<-apriori(Groceries)
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 983

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [8 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [0 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>
```

Returning all rules with a minimum of 0.1% support and 80% confidence requires Supp=0.001 and Conf=0.8 to be passed. We set the rules in decreasing order of confidence and validate the rule summary.

```
23  rules <- apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8))
24  rules<-sort(rules, by="confidence", decreasing=TRUE)
25  summary(rules)
26
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.8    0.1    1 none FALSE            TRUE       5   0.001      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.02s].
writing ... [410 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> 
```

```
> summary(rules)
set of 410 rules

rule length distribution (lhs + rhs):sizes
  3   4   5   6
 29 229 140  12

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   4.000   4.000   4.329   5.000   6.000

summary of quality measures:
    support           confidence        coverage             lift           count
 Min.   :0.001017   Min.   :0.8000   Min.   :0.001017   Min.   : 3.131   Min.   :10.00
 1st Qu.:0.001017   1st Qu.:0.8333   1st Qu.:0.001220   1st Qu.: 3.312   1st Qu.:10.00
 Median :0.001220   Median :0.8462   Median :0.001322   Median : 3.588   Median :12.00
 Mean   :0.001247   Mean   :0.8663   Mean   :0.001449   Mean   : 3.951   Mean   :12.27
 3rd Qu.:0.001322   3rd Qu.:0.9091   3rd Qu.:0.001627   3rd Qu.: 4.341   3rd Qu.:13.00
 Max.   :0.003152   Max.   :1.0000   Max.   :0.003559   Max.   :11.235   Max.   :31.00

mining info:
      data ntransactions support confidence
 Groceries          9835   0.001        0.8
                                                                                call
 apriori(data = Groceries, parameter = list(supp = 0.001, conf = 0.8))
>
```

Summary of rules:

410 rules in total

A summary of quality metrics is given by the ranges of lift, confidence, and assistance.

Data mining details, such as the total amount of data mined and the minimal parameters that were previously established.

We will only list the top 5.

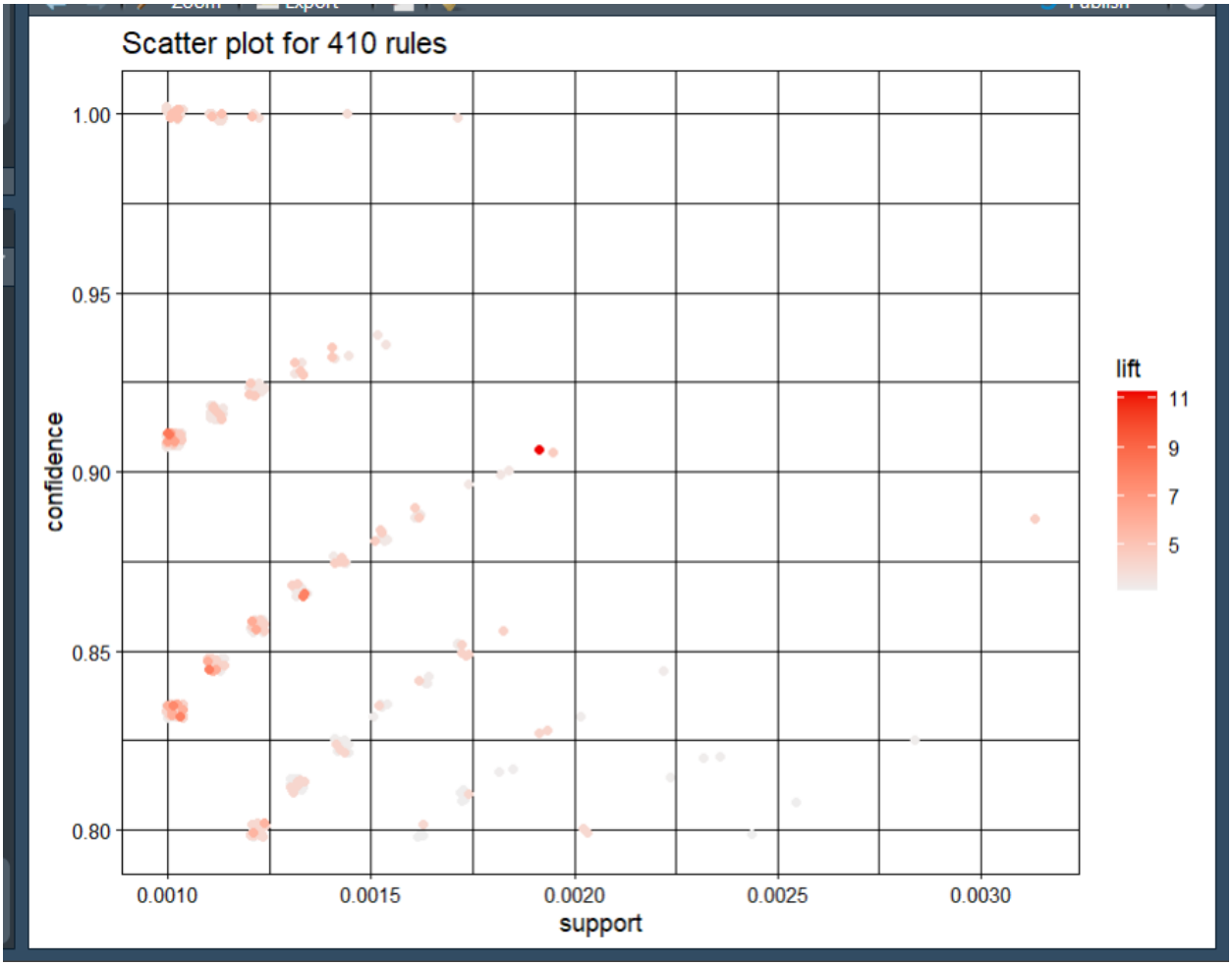
```
26
27  #show the top 5 rules
28  inspect(rules[1:5])
29
```

```
> inspect(rules[1:5])
    lhs                            rhs             support     confidence coverage    lift
[1] {liquor, red/blush wine} => {bottled beer} 0.001931876 0.9047619  0.002135231 11.235269
[2] {curd, cereals}          => {whole milk}   0.001016777 0.9090909  0.001118454  3.557863
[3] {yogurt, cereals}        => {whole milk}   0.001728521 0.8095238  0.002135231  3.168192
[4] {butter, jam}            => {whole milk}   0.001016777 0.8333333  0.001220132  3.261374
[5] {soups, bottled beer}    => {whole milk}   0.001118454 0.9166667  0.001220132  3.587512
```

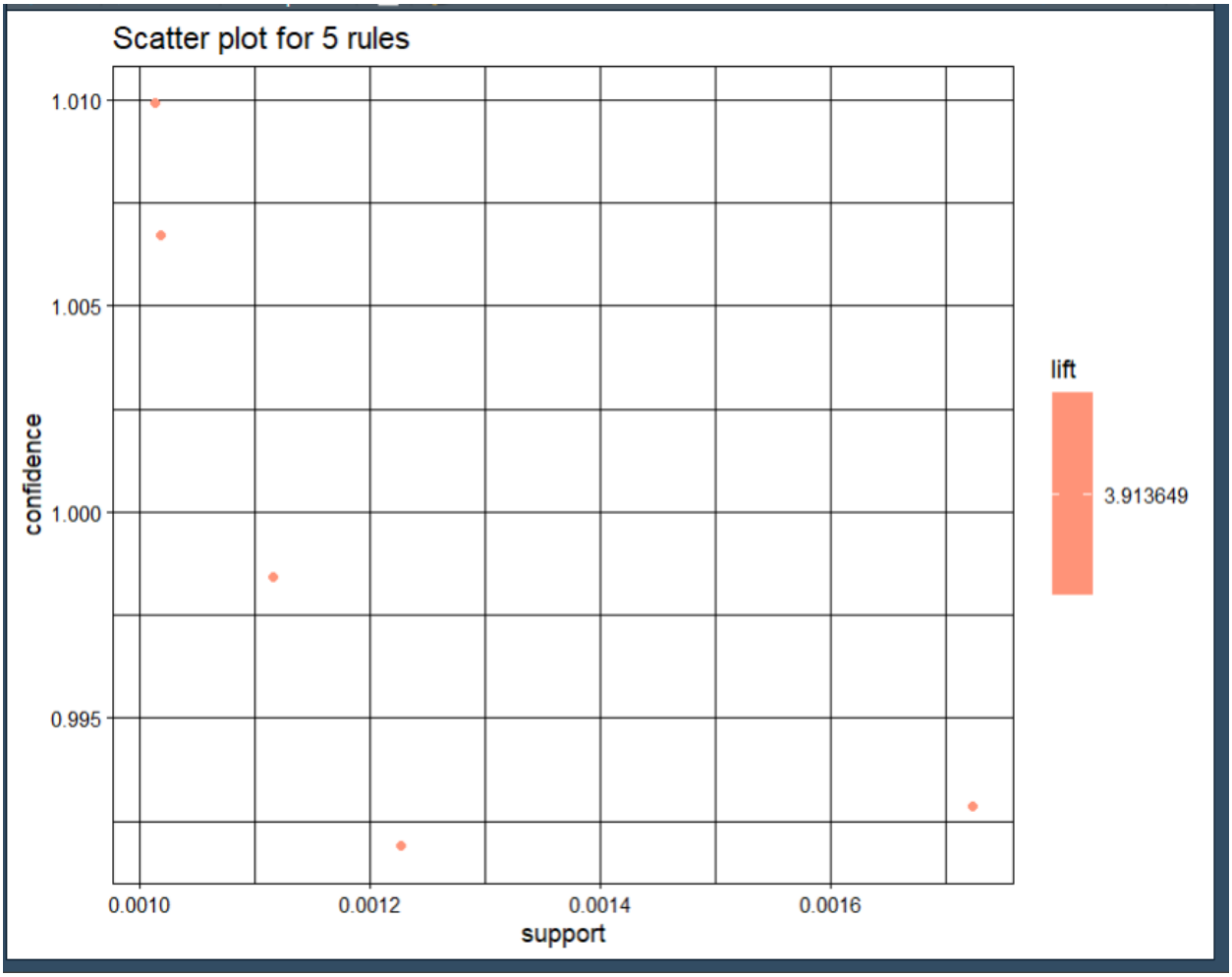
1.7. Result analysis and discussion

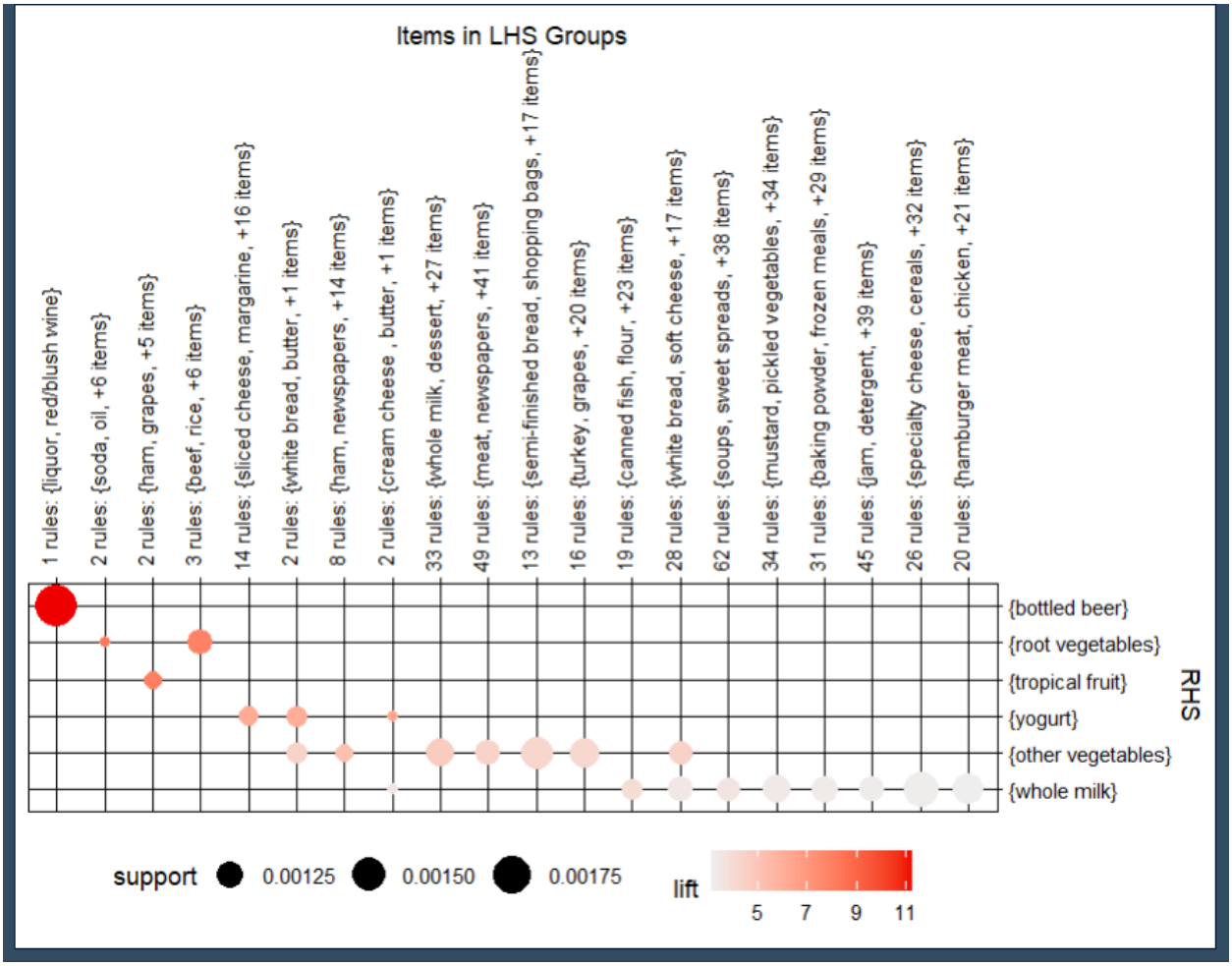
Thousands of rules were created utilizing the dataset; thus, we need to present our findings in a few different ways. We make use of Item Frequency Plot to visualize association rules.

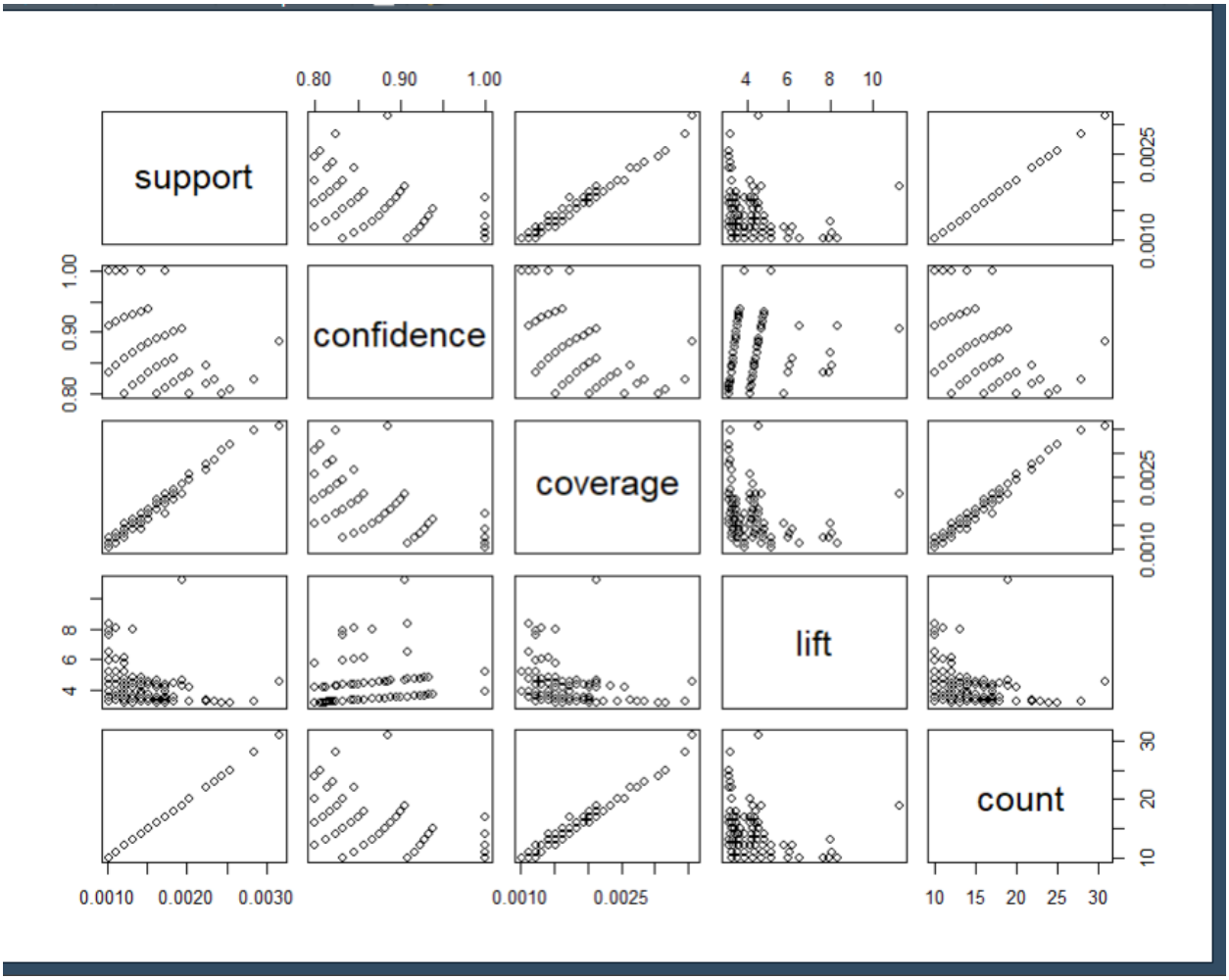
```
#plots of rules
rules <- apriori(Groceries, parameter = list(supp = 0.001, conf = 0.8 ))
rules<-sort(rules, by="confidence", decreasing=TRUE)
#scatterplot of rules
plot(rules)
#scatterplot of top 5 rules
top5<-head(rules,n=5,by="confidence")
plot(top5)
#plot of rules in groups
plot(rules, method="grouped")
#scatter plot matrix to compare the support, confidence,coverage,lift and groups
plot(rules@quality)
```



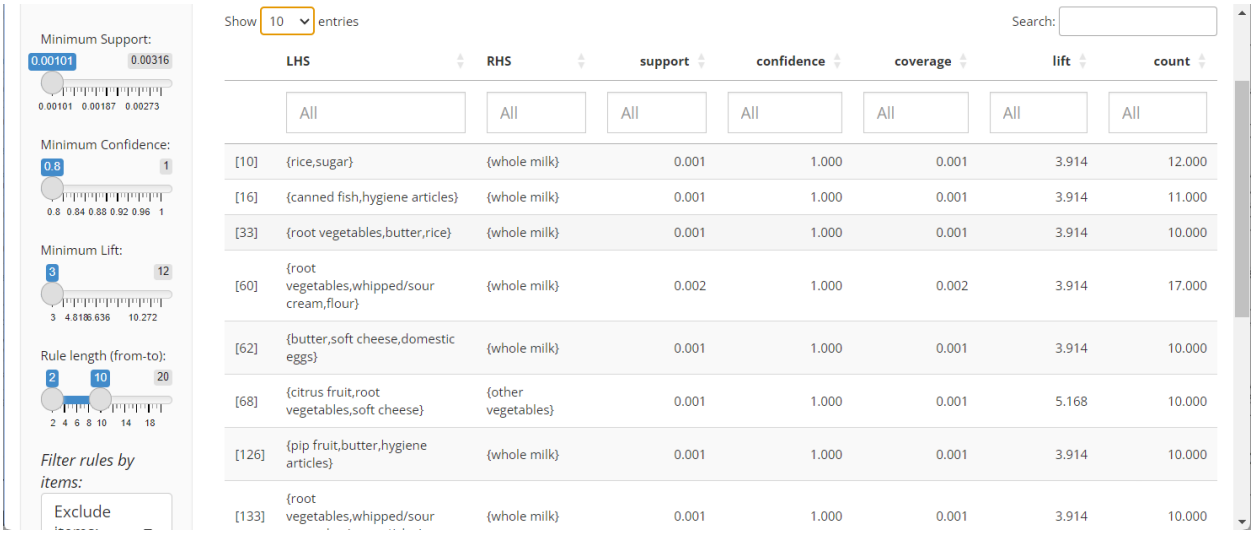
To achieve the intended outcomes thus far, we have combined various programs with the data mining association rule in R software. The objectives of this job are to analyze the results of data mining using the R program's dataset and association rule. At that point, one should predict the outcomes that were attained by use of this dataset.


Scatter plot for 5 rules
1.010
1.005
1.000
0.995
confidence
0.0010
0.0012
0.0014
0.0016
support
lift
3.913649

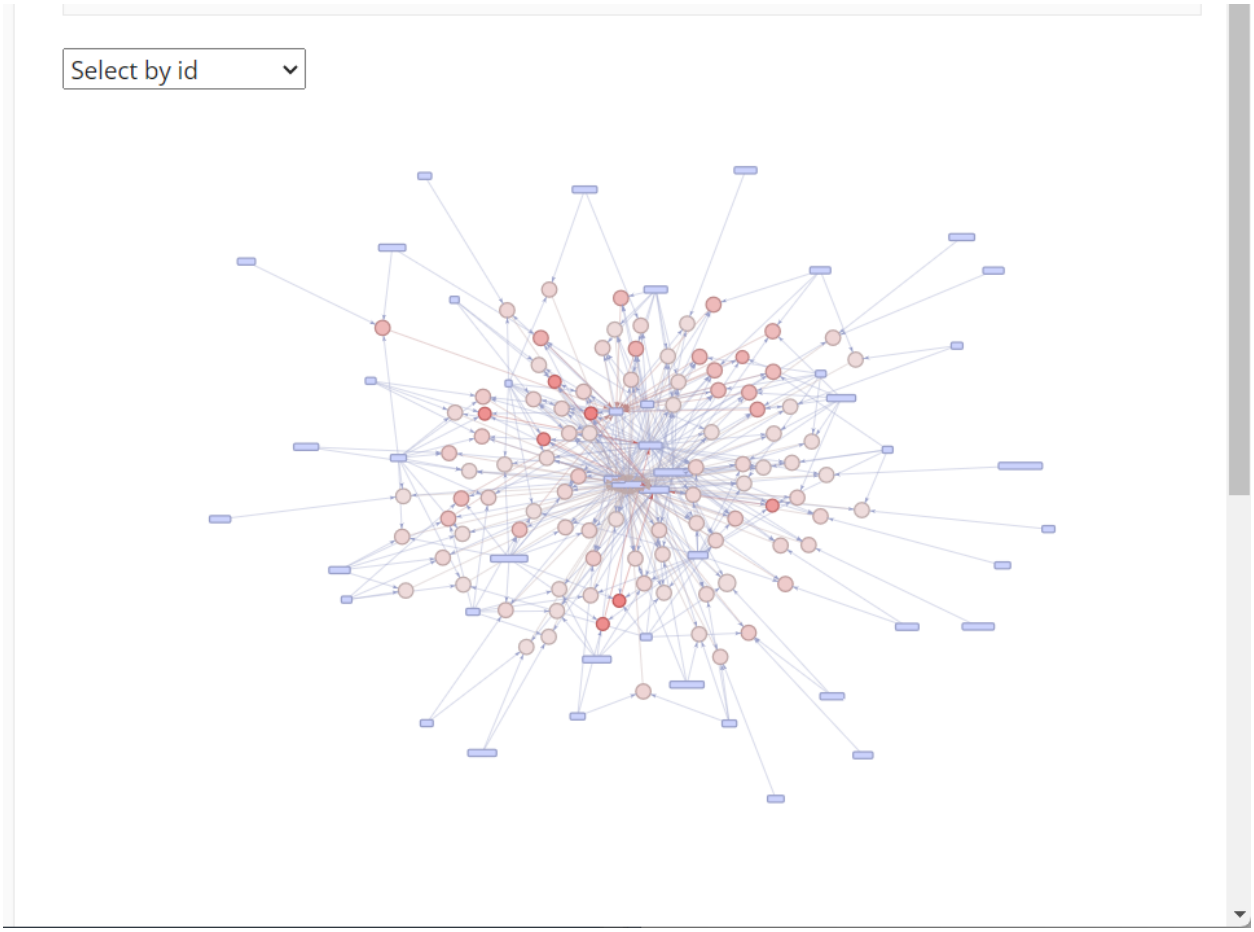

Items in LHS Groups
1 rules: {liquor, red/blush wine}
2 rules: {soda, oil, +6 items}
2 rules: {ham, grapes, +5 items}
3 rules: {beef, rice, +6 items}
14 rules: {sliced cheese, margarine, +16 items}
2 rules: {white bread, butter, +1 items}
8 rules: {ham, newspapers, +14 items}
2 rules: {cream cheese , butter, +1 items}
33 rules: {whole milk, dessert, +27 items}
49 rules: {meat, newspapers, +41 items}
13 rules: {semi-finished bread, shopping bags, +17 items}
16 rules: {turkey, grapes, +20 items}
19 rules: {canned fish, flour, +23 items}
28 rules: {white bread, soft cheese, +17 items}
62 rules: {soups, sweet spreads, +38 items}
34 rules: {mustard, pickled vegetables, +34 items}
31 rules: {baking powder, frozen meals, +29 items}
45 rules: {jam, detergent, +39 items}
26 rules: {specialty cheese, cereals, +32 items}
20 rules: {hamburger meat, chicken, +21 items}
{bottled beer}
{root vegetables}
{tropical fruit}
{yogurt}
{other vegetables}
{whole milk}
RHS
support
0.00125
0.00150
0.00175
lift
5
7
9
11


support
confidence
coverage
lift
count
0.80
0.90
1.00
4
6
8
10
0.0010
0.0025
0.0010
0.0020
0.0030
0.0010
0.0025
10
15
20
25
30

#Datatable


Minimum Support:
0.00101
0.00316
Minimum Confidence:
0.8
1
Minimum Lift:
3
12
Rule length (from-to):
2
10
20
Filter rules by items:
Exclude
Show 10 entries
Search:
LHS
RHS
support
confidence
coverage
lift
count
All
[10] {rice,sugar} {whole milk} 0.001 1.000 0.001 3.914 12.000
[16] {canned fish,hygiene articles} {whole milk} 0.001 1.000 0.001 3.914 11.000
[33] {root vegetables,butter,rice} {whole milk} 0.001 1.000 0.001 3.914 10.000
[60] {root vegetables,whipped/sour cream,flour} {whole milk} 0.002 1.000 0.002 3.914 17.000
[62] {butter,soft cheese,domestic eggs} {whole milk} 0.001 1.000 0.001 3.914 10.000
[68] {citrus fruit,root vegetables,soft cheese} {other vegetables} 0.001 1.000 0.001 5.168 10.000
[126] {pip fruit,butter,hygiene articles} {whole milk} 0.001 1.000 0.001 3.914 10.000
[133] {root vegetables,whipped/sour {whole milk} 0.001 1.000 0.001 3.914 10.000

#Graph



1.8. Conclusion

The groceries dataset has provided important insights into the buying habits of customers. We've discovered frequent item sets and strong association rules that illuminate the connections between various grocery items using methods like Apriori. According to the rules generated, Customers are frequently purchasing whole milk with other groceries.

Businesses can better satisfy customer needs and preferences and improve sales by customizing their products based on the items that customers frequently purchased together. This takes advantage of consumer behavioral patterns.

Association rule mining is a potent technique for deriving meaningful information from massive transactional datasets as empower industry changes, giving companies the ability to make better decisions and stay competitive.

2.Logistic Regression

Predict the occurrence of the Heart Disease, Through Logistic Regression.

2.1 Introduction

An important share of the morbidity and death rates worldwide are caused by heart disease, which is a common and possibly deadly illness. For effective prevention and management techniques, heart disease must be detected and predicted in its early stages. A useful technique for predicting the development of heart disease is logistic regression, a statistical technique that is frequently utilized in predictive modeling to evaluate the likelihood of a discrete ending.

In this summary, we examine the use of logistic regression in predicting the occurrence of heart disease. We'll go over the fundamentals of logistic regression, how it's used to create heart disease prediction models, the factors which impact heart disease, and the significance of predictive modeling in healthcare.

2.2 Dataset

Originally sourced from the UCL Machine Learning Repository, the dataset used to predict heart disease is currently hosted on the "data.world" platform. Using a collection of 14 attributes—13 of which are independent, measurable medical features—its main goal is to determine whether or not a patient has heart disease. Heart disease presence or absence is indicated by the target variable, which is the remaining property. This 270-row dataset is an asset for developing prediction models that aim to increase heart disease early detection and diagnosis. Each row contains data for a distinct patient.

2.3. Explanation and Preparation of Datasets

1. Age – Age in years
2. Sex – Gender (1 = male; 0 = female)
3. Cp – Chest pain type
   - typical angina
   - atypical angina
   - non-anginal pain
   - asymptomatic
4. Trestbps - resting blood pressure (in mm Hg on admission to the hospital)
5. Chol - serum cholesterol in mg/dl
6. Fbs - fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. Restecg - resting electrocardiographic results

   -- normal

   -- having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05
   mV)

   -- Showing probable or definite left ventricular hypertrophy by Estes' criteria

8. Thalach - maximum heart rate achieved

9. Exang – exercise-induced angina (1 = yes; 0 = no)

10. Oldpeak - ST depression induced by exercise relative to rest

11. Slope - the slope of the peak exercise ST segment

    0: upsloping

    1: flat

    2: downsloping

12. Ca - number of major vessels (0-3) colored by fluoroscopy

13. Thal - thal: 1= normal; 2 = fixed defect; 3= reversible defect

14. Heart. Disease - diagnosis of heart disease (angiographic disease status) - The predicted attribute

The "Heart disease" column in the dataset indicates whether a patient has been diagnosed with heart disease or not. "Presence" indicates that a patient has the illness, while "Absence" indicates that a patient does not. Our goal in converting this categorical column to a numeric format is to improve the dataset's readability and analytical abilities. By simplifying computing and modeling methods, this transformation will help us make better use of the data for statistical analysis and predictive analytics.

Presence = 1, Absence = 0

To predict the heart disease of patients who are in this dataset, we use the Logistic Regression method which is

very popular in data mining to categorize the distinct classes of the respective dataset.

2.3 Implementation in R

*Install and activate the packages that are required.

```r
install.packages("party")
install.packages("epitools")
install.packages("ggplot2")
install.packages("GGally")
install.packages("tidyverse")
install.packages("corrplot")
install.packages("RColorBrewer")




library(party)
library(epitools)
library(ggplot2)
library(GGally)
library(tidyverse)
library(corrplot)
library(RColorBrewer)
```

*Load the data set into R

```r

Heart<-read.csv("C:\\Users\\Midara\\Downloads\\regression\\heart.csv")

data(Heart) #loadng  diabetes dataset

```

*Discover the parameters of the data set.

Summary, Dimensions, Structure, first five rows

```
> sum(diag(trn_tab))/sum(trn_tab)
[1] 0.8829268
> summary(Heart)
      age             sex               cp            trestbps          chol            fbs        
 Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000  
 1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000  
 Median :55.00   Median :1.0000   Median :1.000   Median :130.0   Median :240.0   Median :0.0000  
 Mean   :54.37   Mean   :0.6832   Mean   :0.967   Mean   :131.6   Mean   :246.3   Mean   :0.1485  
 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:140.0   3rd Qu.:274.5   3rd Qu.:0.0000  
 Max.   :77.00   Max.   :1.0000   Max.   :3.000   Max.   :200.0   Max.   :564.0   Max.   :1.0000  
    restecg          thalach          exang           oldpeak         slope             ca        
 Min.   :0.0000   Min.   : 71.0   Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000  
 1st Qu.:0.0000   1st Qu.:133.5   1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000  
 Median :1.0000   Median :153.0   Median :0.0000   Median :0.80   Median :1.000   Median :0.0000  
 Mean   :0.5281   Mean   :149.6   Mean   :0.3267   Mean   :1.04   Mean   :1.399   Mean   :0.7294  
 3rd Qu.:1.0000   3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000  
 Max.   :2.0000   Max.   :202.0   Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000  
      thal       target    
 Min.   :0.000   0:138  
 1st Qu.:2.000   1:165  
 Median :2.000          
 Mean   :2.314          
 3rd Qu.:3.000          
 Max.   :3.000          
```

```
24
25  # Discover the parameteres of the data frame
26  summary(Heart)
27  dim(Heart)
28  str(Heart)
29  head(Heart)
30
```

```
> dim(Heart)
[1] 303  14
> str(Heart)
'data.frame':   303 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
 $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
 $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
 $ target  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
> head(Heart)
  age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
1  63   1  3      145  233   1       0     150     0     2.3     0  0    1      1
2  37   1  2      130  250   0       1     187     0     3.5     0  0    2      1
3  41   0  1      130  204   0       0     172     0     1.4     2  0    2      1
4  56   1  1      120  236   0       1     178     0     0.8     2  0    2      1
5  57   0  0      120  354   0       1     163     1     0.6     2  0    2      1
6  57   1  0      140  192   0       1     148     0     0.4     1  0    1      1
>
```

*Convert dependent variable (chd69) to factor and build new dataset with selected variables

.

```r
#convert dependent variable (Class.Variable) to factor
Heart$target<- as.factor(Heart$target)
#build new dataset with selected variables
Heart <- Heart [, c(1,2,3,4,5,6,7,8,9,10,11,12,13,14)]
head(Heart )
View(Heart )
```

*Explore all the variables in the dataset.

Correlogram is a correlation matrix graph useful for highlighting the highly linked variables in a data set. In this figure, the correlation coefficients are shaded according to their respective values. A matrix of correlations can be used to be rearranged in accordance with the degree of correlation that exists between the variables.

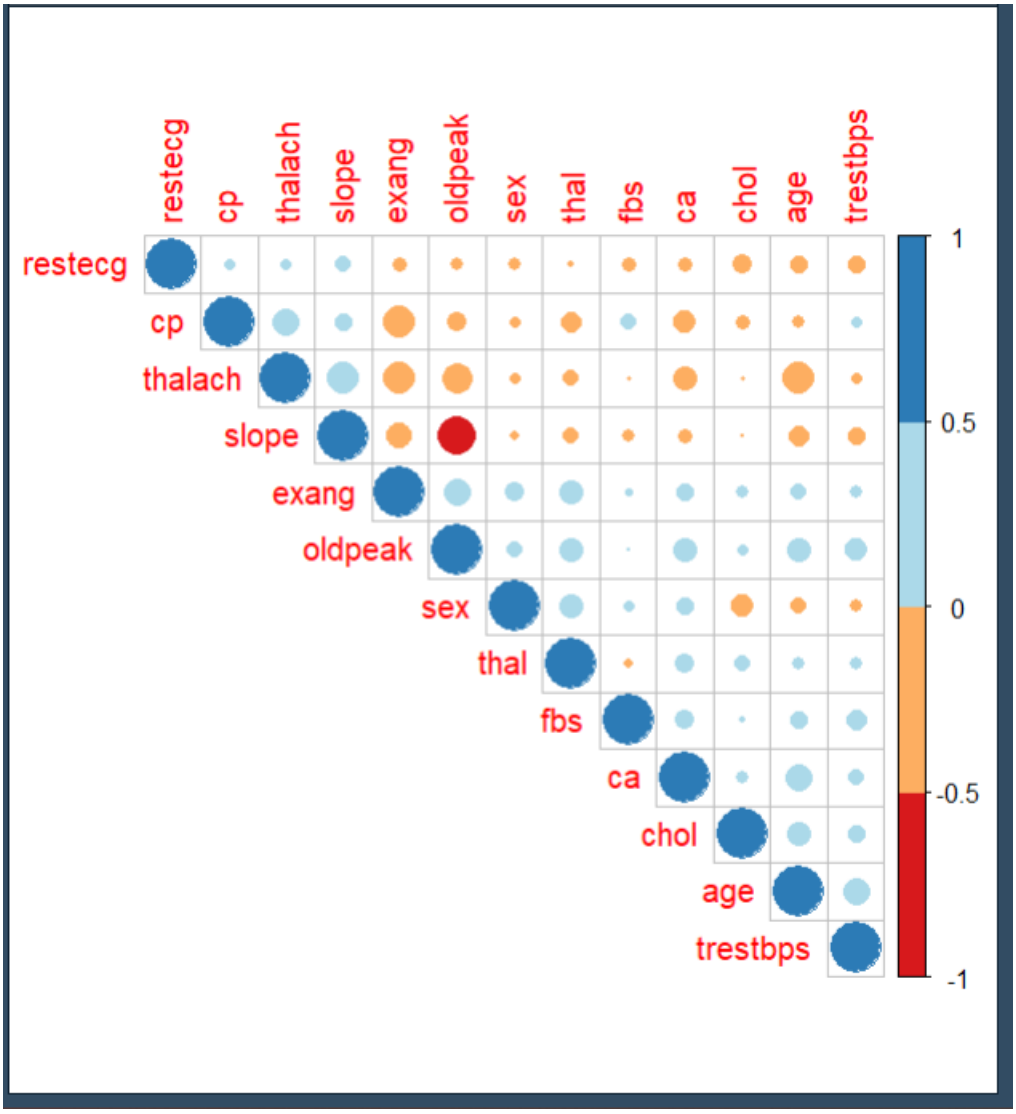
```r
Heart_cor<-cor(Heart [,-14])
View(Heart_cor)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
age	1.00000000	-0.09844660	-0.06865302	0.27935091	0.213677957	0.121307648	-0.11621090	-0.398521938	0.09680083	0.210012567	-0.16881424	0.27632624	0.06800138
sex	-0.09844660	1.00000000	-0.04935288	-0.05676882	-0.197912174	0.045031789	-0.05819627	-0.044019908	0.14166381	0.096092877	-0.03071057	0.11826141	0.21004110
cp	-0.06865302	-0.04935288	1.00000000	0.04760776	-0.076904391	0.094444035	0.04442059	0.295762125	-0.39428027	-0.149230158	0.11971659	-0.18105303	-0.16173557
trestbps	0.27935091	-0.05676882	0.04760776	1.00000000	0.123174207	0.177530542	-0.11410279	-0.046697728	0.06761612	0.193216472	-0.12147458	0.10138899	0.06220989
chol	0.21367796	-0.19791217	-0.07690439	0.12317421	1.000000000	0.013293602	-0.15104008	-0.009939839	0.06702278	0.053951920	-0.00403777	0.07051093	0.09880299
fbs	0.12130765	0.04503179	0.09444403	0.17753054	0.013293602	1.000000000	-0.08418905	-0.008567107	0.02566515	0.005747223	-0.05989418	0.13797933	-0.03201934
restecg	-0.11621090	-0.05819627	0.04442059	-0.11410279	-0.151040078	-0.084189054	1.00000000	0.044123444	-0.07073286	-0.058770226	0.09304482	-0.07204243	-0.01198140
thalach	-0.39852194	-0.04401991	0.29576212	-0.04669773	-0.009939839	-0.008567107	0.04412344	1.000000000	-0.37881209	-0.344186948	0.38678441	-0.21317693	-0.09643913
exang	0.09680083	0.14166381	-0.39428027	0.06761612	0.067022783	0.025665147	-0.07073286	-0.378812094	1.00000000	0.288222808	-0.25774837	0.11573938	0.20675379
oldpeak	0.21001257	0.09609288	-0.14923016	0.19321647	0.053951920	0.005747223	-0.05877023	-0.344186948	0.28822281	1.000000000	-0.57753682	0.22268232	0.21024413

Showing 1 to 11 of 13 entries, 13 total columns

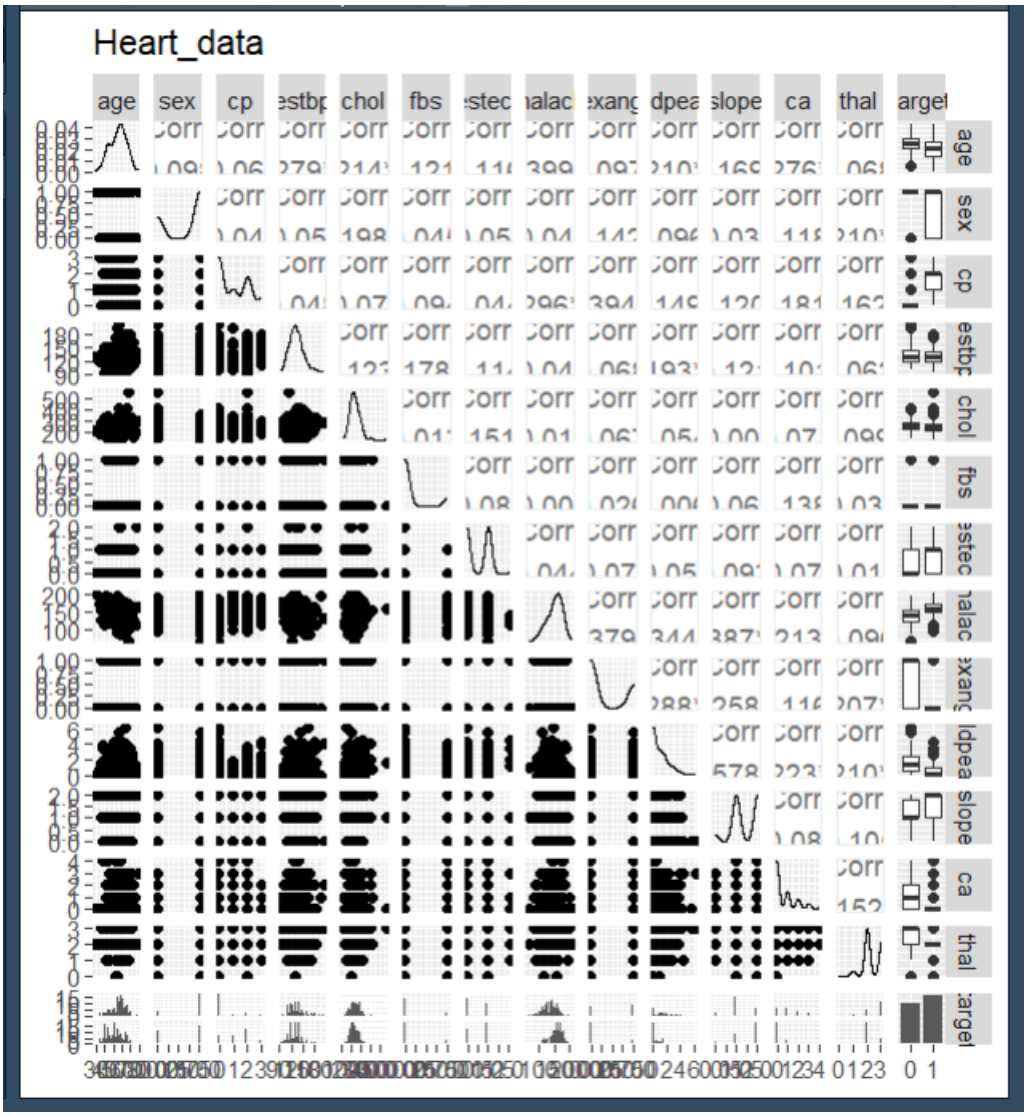
*Visualize Correlation Matrix using Correlogram

```r
corrplot(Heart_cor, type="upper", order="hclust", col=brewer.pal(n=4,
                                                    name="RdYlBu"))
```



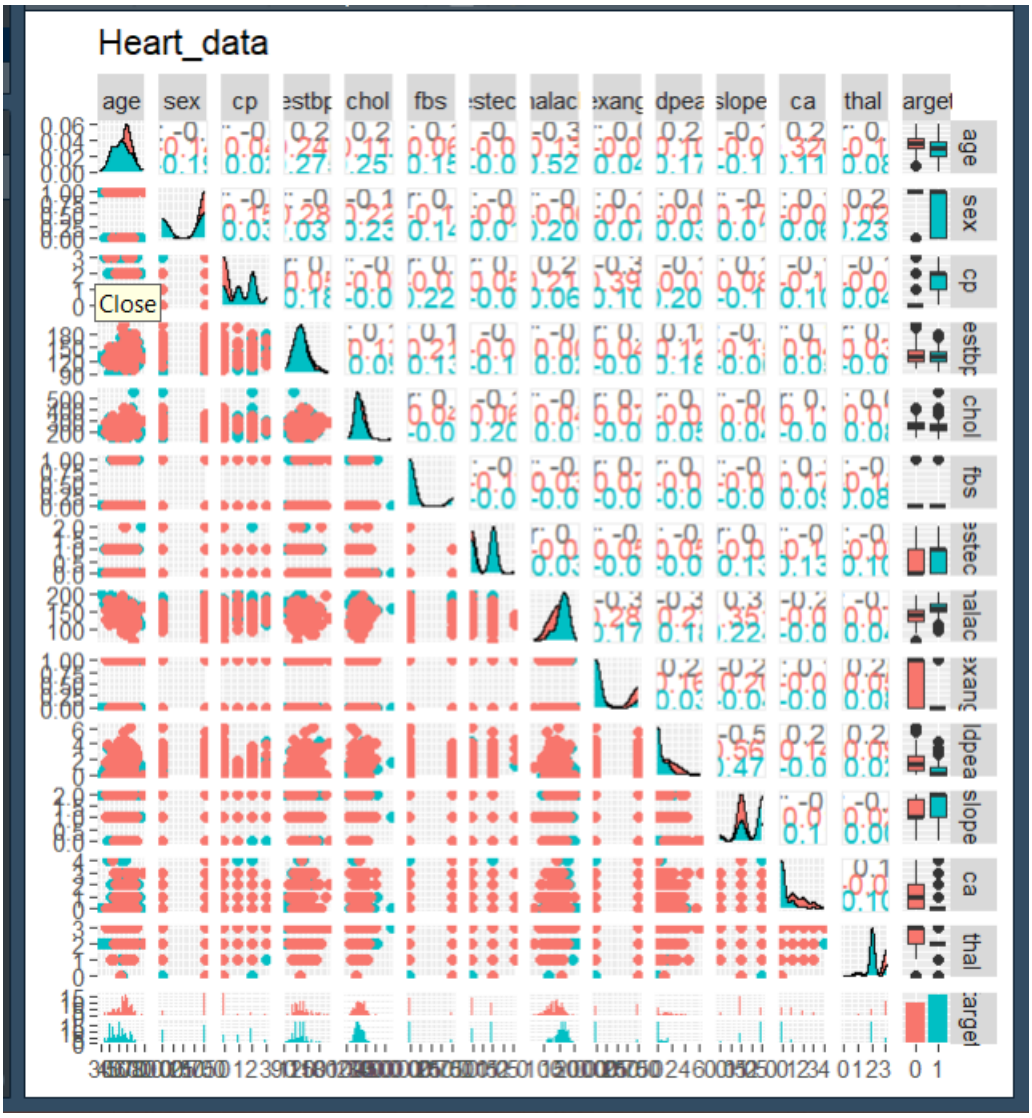
ggpairs() function is a special form of a function that produces a pairwise comparison of multivariate data.

```
62
63  ggpairs(data=Heart , title="Heart_data")
64
```



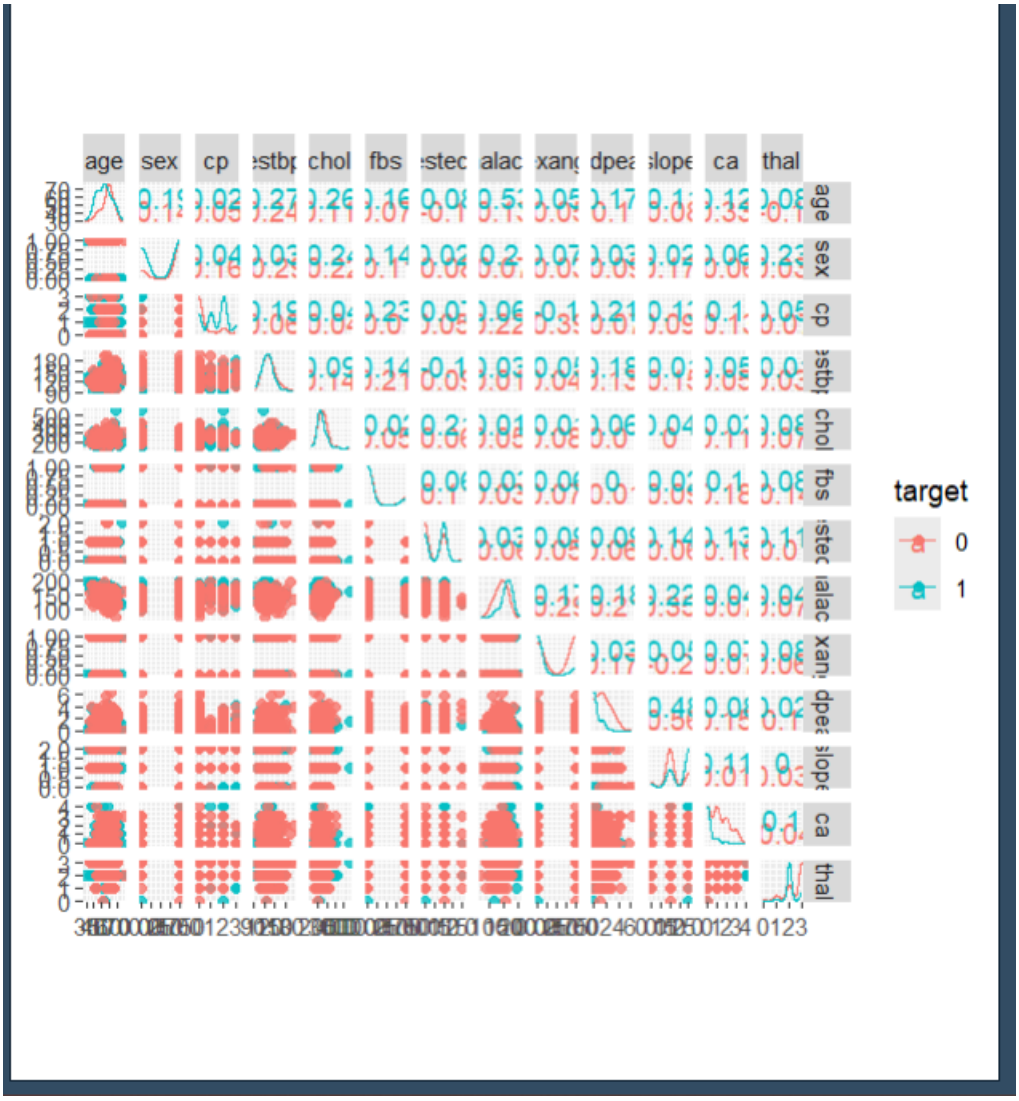
*Compare the variable with target

```r
#Plot

ggpairs(data=Heart,mapping=aes(color=target),title="Heart_data")
```



ggscatmatis like ggpairs but only works for purely numeric multivariate data. creates a matrix with scatterplots in the lower diagonal, densities on the diagonal and correlations written in the upper diagonal.

```
70
71  ggscatmat(data=Heart,color="target",alpha=0.8)
72
```



*Model 1-Build a linear regression model with Age and trestbps variables

These model evaluate the effect between the age and resting blood pressure to occurrence of the heart disease

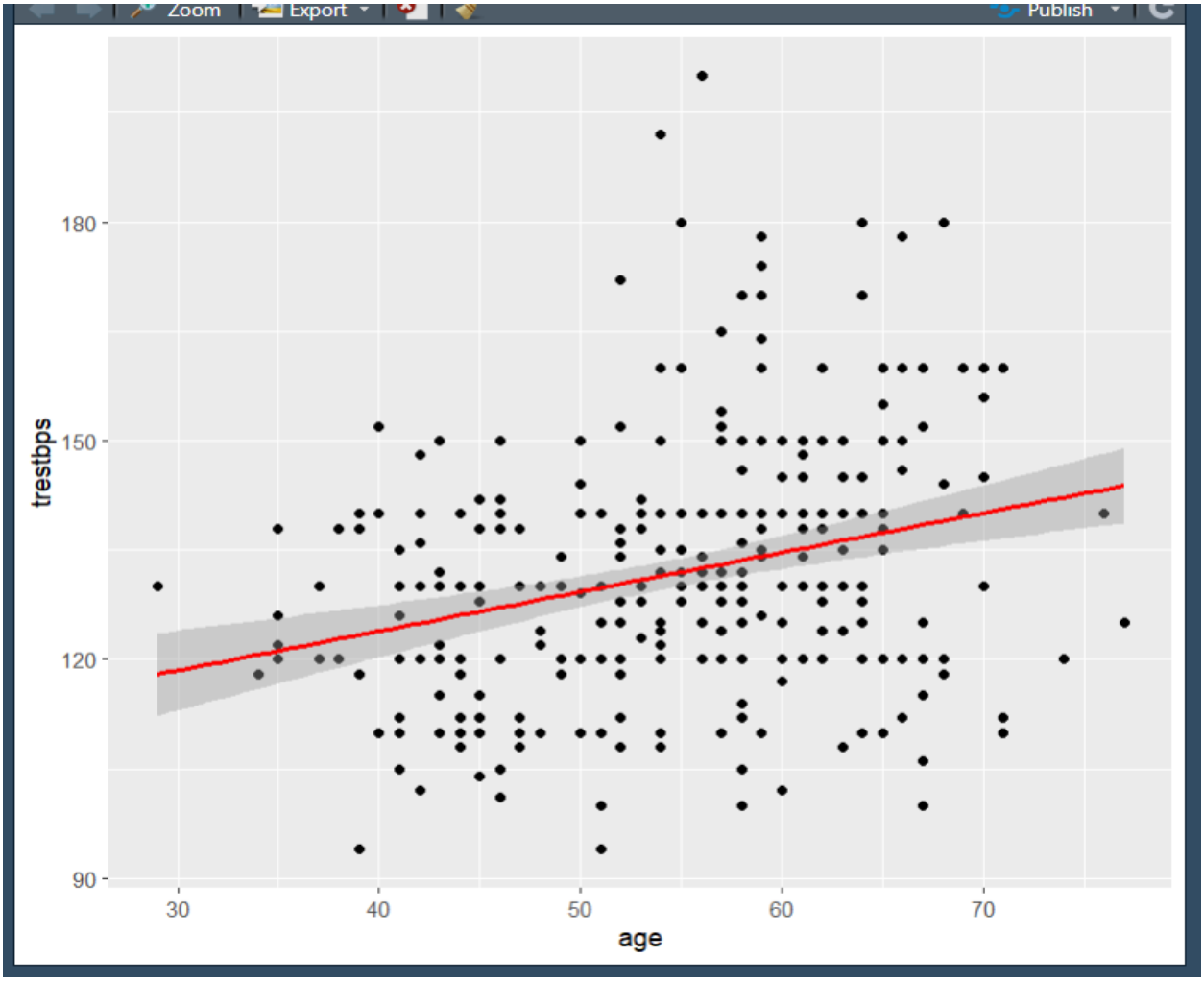
```
69
70  #model 1 - Building a linear regression model with age and trestbps variables
71  lm1 <- lm( age~ trestbps, data = Heart)
72  summary(lm1)
73
74  ggplot(Heart, aes(x =age, y =trestbps)) +
75    geom_point() +
76    geom_smooth(method = "lm", color = "red")
77
```

*Summary of the model

This model predict that investigate the link between age and resting blood pressure (trestbps) through linear regression analysis. Higher levels of resting blood pressure are linked to slightly older ages, and they have a significant effect on age ($p < 0.001$). Though it only accounts for 7.80% of the variability in age, the model has little explanatory power.

```
R 4.3.2 · ~/
> summary(lm1)

Call:
lm(formula = age ~ trestbps, data = Heart)

Residuals:
     Min       1Q   Median       3Q      Max
-25.1314  -6.1441   0.5792   6.3559  23.5919

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.32545    3.80557   9.283  < 2e-16 ***
trestbps     0.14466    0.02866   5.048 7.76e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.735 on 301 degrees of freedom
Multiple R-squared:  0.07804,   Adjusted R-squared:  0.07497
F-statistic: 25.48 on 1 and 301 DF,  p-value: 7.762e-07
```

*Visualization of the model.



*Model 2 -Building a linear regression model with Age vs cholesterol level

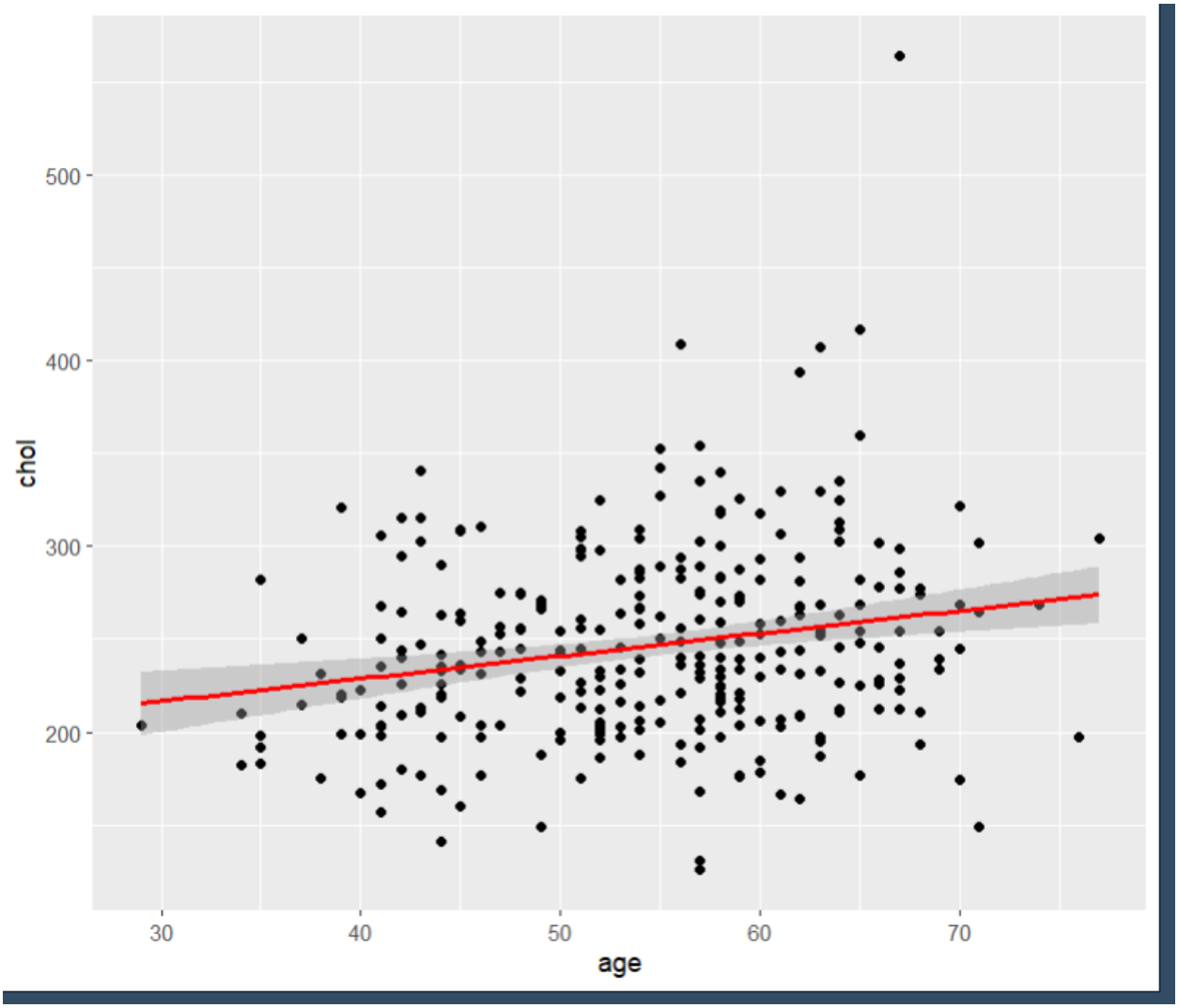
To see the relationship between the age and cholesterol level of people

```
#model 2 -Building a linear regression model with age vs chol
lm1 <- lm( age~ chol ,data = Heart)
summary(lm1)


#Plot linear regression model


ggplot(Heart, aes(x =age, y = chol)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red")
```

*Summary of the model.

By utilizing the Heart dataset, the algorithm develops a linear regression model that predicts age based on cholesterol levels. Age and cholesterol are significantly correlated ($p < 0.001$), according to the results, with somewhat older ages being associated with higher cholesterol levels. The model can only account for about 4.57% of the variability in age, indicating a poor overall explanatory power.

```
R 4.3.2 · ~/

Call:
lm(formula = age ~ chol, data = Heart)

Residuals:
     Min       1Q   Median       3Q      Max
-23.7839  -6.4734   0.4782   6.3221  23.4782

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.145729   2.482848  18.183  < 2e-16 ***
chol         0.037442   0.009867   3.795 0.000179 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.887 on 301 degrees of freedom
Multiple R-squared:  0.04566,   Adjusted R-squared:  0.04249
F-statistic:  14.4 on 1 and 301 DF,  p-value: 0.0001786

>
```



*Establishing a Logistic Regression model

Binary outcome variables can be modeled using the logistic regression technique. To find out if we can predict a person has heart disease for a given set of factors, we will now construct a logistic regression model for heart disease data.
specify data for both the train and validate (test) runs.
split our sample into 30% for validation and 70% for training.

```r
#Now we will divide our sample into 70% Training and 30% Validation parts.
# Load the Heart Disease dataset
# Set the seed for reproducibility
set.seed(42)

# Generate a vector to assign each row to either training or validation set
pd <- sample(2, nrow(Heart), replace = TRUE, prob = c(0.75, 0.25))

# Create the training set
train <- Heart[pd == 2, ]
head(train)

# Create the validation set
validate <- Heart[pd == 2, ]
head(validate)
```

*Inspect the train and validate data sets that was created

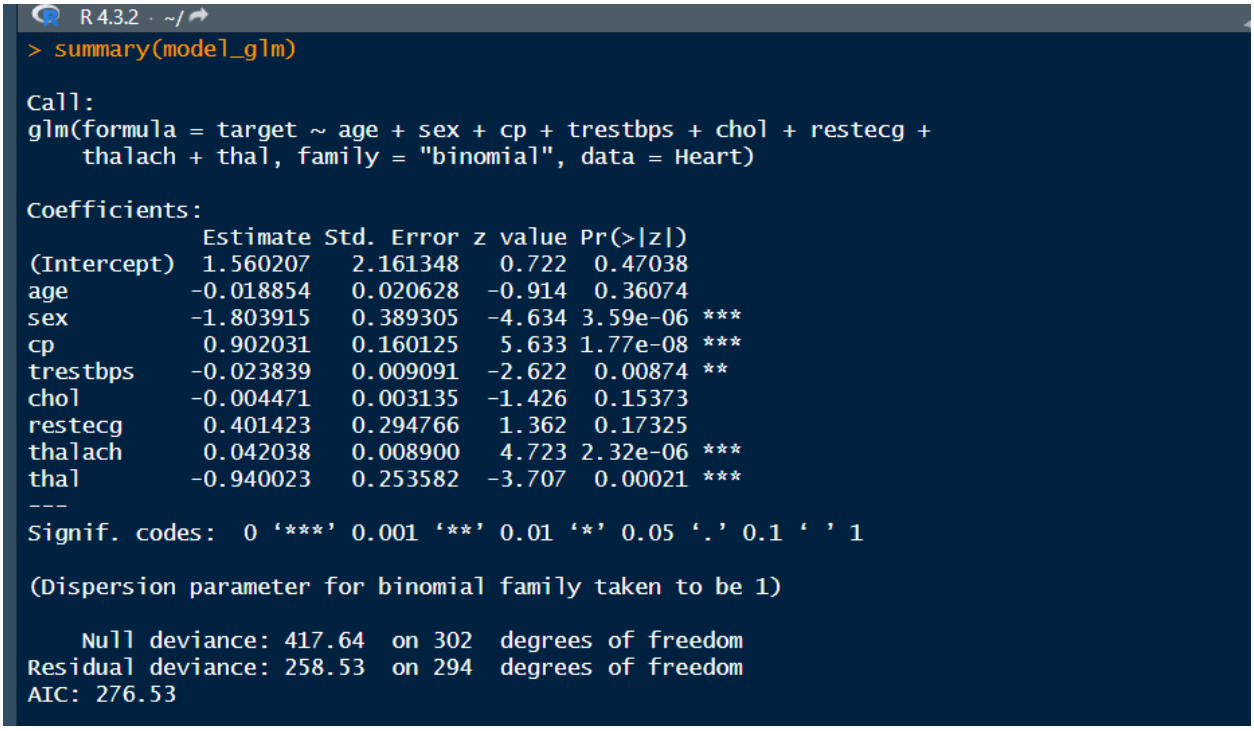
```
> head(train)
   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
1   63   1  3      145  233   1       0     150     0     2.3     0  0    1      1
2   37   1  2      130  250   0       1     187     0     3.5     0  0    2      1
4   56   1  1      120  236   0       1     178     0     0.8     2  0    2      1
13  49   1  1      130  266   0       1     171     0     0.6     2  0    2      1
16  50   0  2      120  219   0       1     158     0     1.6     1  0    2      1
17  58   0  2      120  340   0       1     172     0     0.0     2  0    2      1
>
> # Create the validation set
> validate <- Heart[pd == 2, ]
> head(validate)
   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
1   63   1  3      145  233   1       0     150     0     2.3     0  0    1      1
2   37   1  2      130  250   0       1     187     0     3.5     0  0    2      1
4   56   1  1      120  236   0       1     178     0     0.8     2  0    2      1
13  49   1  1      130  266   0       1     171     0     0.6     2  0    2      1
16  50   0  2      120  219   0       1     158     0     1.6     1  0    2      1
17  58   0  2      120  340   0       1     172     0     0.0     2  0    2      1
```

To determine whether we can infer that a person has heart disease based on certain characteristics construct a logistic regression model.
Logistic regression model creation ought to resemble linear regression model creation in many ways. But glm() is used in place of lm(). Also take note that in a situation including binary classification, we must define family = "binomial".

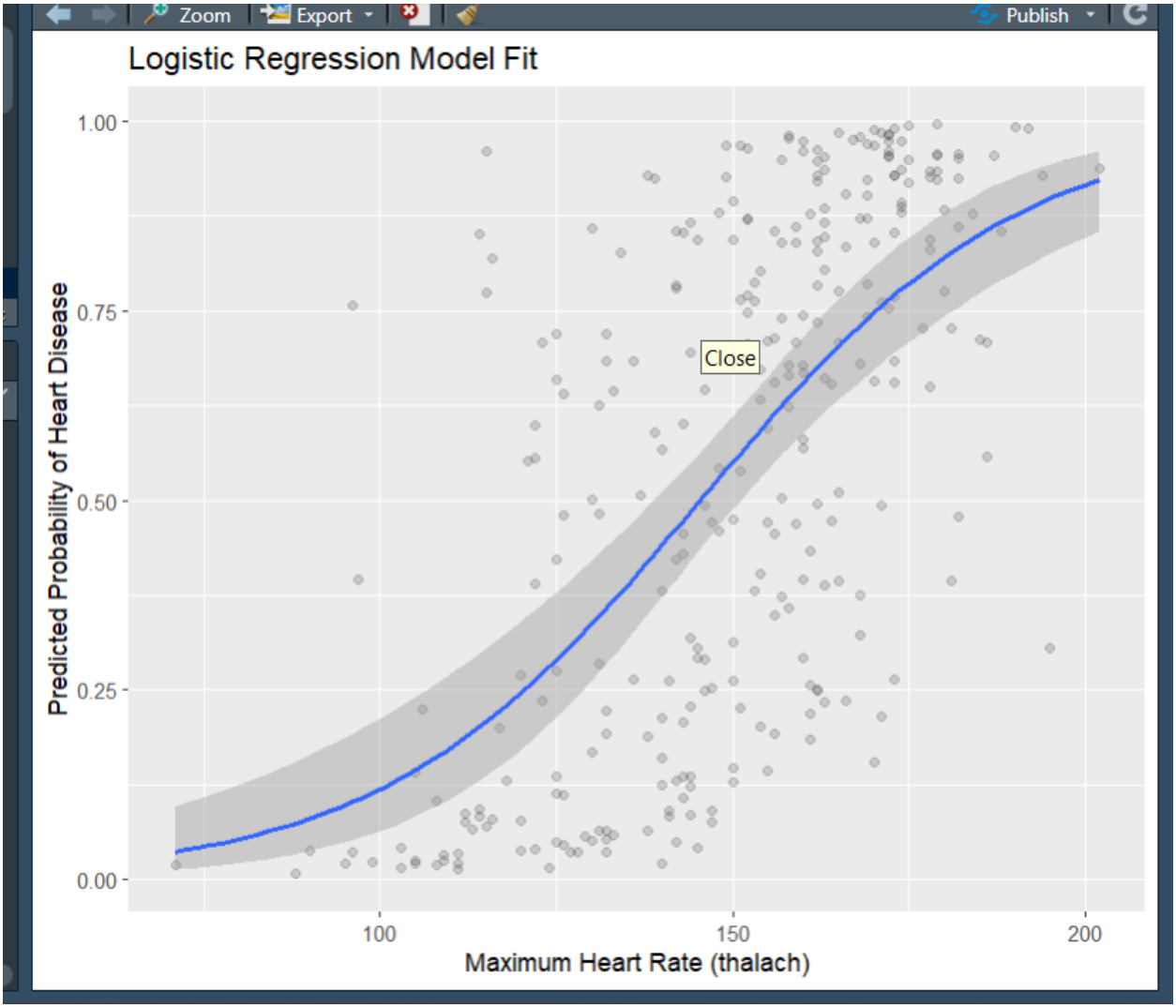
```
#Model 3 - targe

model_glm <- glm( target~age+sex+cp+trestbps+chol+restecg+thalach+thal,
                  data = Heart, family = "binomial")
summary(model_glm)

# Predict probabilities using the model
Heart$predicted_prob <- predict(model_glm, type = "response")

# Visualize the predictions
Heart %>%
  ggplot(aes(x = thalach, y =predicted_prob )) +
  geom_point(alpha = 0.15) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  ggtitle("Logistic Regression Model Fit") +
  xlab("Maximum Heart Rate (thalach)") +
  ylab("Predicted Probability of Heart Disease")
```

*Summary
Using the dataset, this logistic regression model investigates the predictors of a binary target variable. Sex ($p < 0.001$), the type of chest pain (cp) ($p < 0.001$),

resting blood pressure (trestbps) (p = 0.00874), maximum heart rate attained (thalach) (p < 0.001), and thalassemia type (thal) (p = 0.00021) are significant predictors. There are no significant correlations found between other variables including age, cholesterol, or resting electrocardiogram (restecg) values. Based on the residual deviance and AIC values, the model fits to the prediction well.



*Visualization of the regression model



Turn_pred- convert the probabilities to classifications. Traditionally, a midpoint value such as 0.8 is used to “categorize” the probabilities.

Turn_tab-Making predictions on the train dataset.

Sum-Evaluation of the model

The model is trained ,the algorithm creates a confusion matrix ({trn_tab}) to assess its predictions. There are four types of forecasts displayed: true negative (TN), false positive (FP), false negative (FN), and true positive (TP). By adding together, the diagonal members of the confusion matrix and dividing that amount by the number

of predictions made, the accuracy of the model on the training data is computed, yielding an approximate 68.49% accuracy.

```
130
131  #Logistic Regression Model Prediction
132  #First of all we should understand how to use the predict() function with glm().
133  #In order to return probabilities, we must specify type = "response".
134  #As mentioned before, these predicted values are probablilites, not classifications.
135  #We must "manually" convert the probabilities to classifications. Traditionally, a midpoint value s
136
137  trn_pred <- ifelse(predict(model_glm, type = "response") >0.8, "1", "0")
138  trn_pred
139  #Logistic Regression Model Evaluation
140  #One of the best way evaluate a classification models is to compare the actual response values with
141  #This matrix can be generated with the base table() function.
142  #Making predictions on the train dataset.
143  trn_tab <- table(predicted = trn_pred, actual = train$target)
144  trn_tab
145
146  #Model Evaluation
147  sum(diag(trn_tab))/sum(trn_tab)
148  #Making predictions on the validate dataset
```

```
> trn_pred <- ifelse(predict(model_glm, type = "response") >0.8, "1", "0")
> trn_pred
  1   2   4  13  16  17  21  23  24  28  30  32  36  39  44  46  47  49  54  62  63  65  68  85  94
"1" "1" "0" "0" "1" "1" "0" "0" "1" "1" "1" "0" "1" "1" "0" "0" "1" "1" "1" "0" "1" "1" "0" "0" "0"
 95 105 106 112 113 120 127 128 131 134 135 137 139 144 146 148 153 156 161 177 179 180 181 182 187
"0" "1" "1" "1" "1" "0" "0" "1" "1" "0" "0" "1" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0"
188 192 201 203 207 216 220 224 226 235 236 238 250 256 264 267 269 270 272 275 284 291 297
"0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0"
> #Logistic Regression Model Evaluation
> #One of the best way evaluate a classification models is to compare the actual response values with the
predicted ones using a cross-table, which is often called a confusion matrix.
> #This matrix can be generated with the base table() function.
> #Making predictions on the train dataset.
> trn_tab <- table(predicted = trn_pred, actual = train$target)
> trn_tab
         actual
predicted  0  1
        0 27 21
        1  2 23
>
> #Model Evaluation
> sum(diag(trn_tab))/sum(trn_tab)
[1] 0.6849315
> #Making predictions on the validate dataset
```

*Making predictions on the validate dataset.

Sum()-depicts the model evaluation

The validation dataset is used by the algorithm to assess the logistic regression model by model_glm(). It compares the actual and expected target values by generating a confusion matrix (tst_tab). Based on the validation data, the model performs well in terms of prediction, with an accuracy of roughly 84.93%.

```
#Making predictions on the validate dataset.

tst_pred <- ifelse(predict(model_glm, newdata = validate, type =
                           "response") > 0.5, "1", "0")
tst_tab <- table(predicted = tst_pred, actual = validate$target)
tst_tab
#Model Evaluation
sum(diag(tst_tab))/sum(tst_tab)
```

```
> tst_pred <- ifelse(predict(model_glm, newdata = validate, type =
+                                "response") > 0.5, "1", "0")
> tst_tab <- table(predicted = tst_pred, actual = validate$target)
> tst_tab
         actual
predicted  0  1
        0 25  7
        1  4 37
> #Model Evaluation
> sum(diag(tst_tab))/sum(tst_tab)
[1] 0.8493151
> model_glm <- glm(target~ cp, data = train, family = "binomial")
```

*Build a logistic regression model to check whether we can predict a person has heart disease for given all the independent variables.

```
156  # Model 4 - Lets build a logistic regression model to check whether we can predict a person has CHD
157  model_glm <- glm(target~ cp, data = train, family = "binomial")
158  summary(model_glm)
```

With regard to the training dataset's target variable, the logistic regression model investigates the effect of chest pain type (cp). An estimated increase of 1.8273 in the log-odds of the target variable corresponds to every unit increase in chest pain type, indicating a substantial positive relationship ($p < 0.001$) between the two variables. Chest pain type may be a useful predictor of the desired outcome, since the model shows a good fit and a decrease in residual deviation when compared to the null model.showing 71% model accuracy saying that this model can predict occurrence of heart disease at good accuracy.

```
> model_glm <- glm(target~ cp, data = train, family = "binomial")
> summary(model_glm)

Call:
glm(formula = target ~ cp, family = "binomial", data = train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.7495     0.3441  -2.178   0.0294 *
cp            1.8273     0.4543   4.022 5.76e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 98.095  on 72  degrees of freedom
Residual deviance: 67.569  on 71  degrees of freedom
AIC: 71.569

Number of Fisher Scoring iterations: 5

>
```

*Manually convert the probabilities to classifications.

```
summary(model_glm)
#We must "manually" convert the probabilities to classifications.
trn_pred <- ifelse(predict(model_glm, type = "response") >0.5, "1", "0")
#Making predictions on the train set.
trn_tab <- table(predicted = trn_pred, actual = train$target)
trn_tab
#Model Evaluation
sum(diag(trn_tab))/sum(trn_tab)

```

*Summary

the trained model shows 83.56% fit in the model.Depicting that by the development of this model can highly affect to mitigating the risk of heart disease.

```
> #Making predictions on the train set.
> trn_tab <- table(predicted = trn_pred, actual = train$target)
> trn_tab
         actual
predicted  0  1
        0 27 10
        1  2 34
> #Model Evaluation
> sum(diag(trn_tab))/sum(trn_tab)
[1] 0.8356164
```

2.6 Result analysis

Based on the UCI Heart Disease dataset, a logistic regression model's effectiveness and consistency in classifying cases are thoroughly evaluated while analyzing the model's output. To assess the overall efficacy of the model, computations are first made using model assessment measures, including accuracy, precision, recall, F1-score, and AUC-ROC. Following this, a confusion matrix is produced in order to help comprehend the model's classification behavior by offering insights into the distribution of true positive, true negative, false positive, and false negative assumptions. Through a study of the logistic regression model's coefficients, significant features that significantly influence the likelihood of heart disease are identified through feature importance analysis. Discovering the direction and degree of correlation between predictor variables and the desired result is made easier by interpreting these coefficients in light of the domain.

To make sure the model is robust across various data subsets, cross-validation is used to evaluate the model's generalization performance and identify any possible overfitting problems. Additional information on the relative effectiveness of logistic regression for this dataset can be gained through comparisons with various classification algorithms.

When it comes to providing a thorough knowledge of the predictive power of the logistic regression model in the context of heart disease prediction, visualization approaches including ROC curves, precision-recall curves, and feature significance plots are helpful in communicating the findings.

2.7. Conclusion

The present study employed the heart disease dataset from UCI in R to create logistic regression models, which are helpful tools for forecasting the probability of heart disease based on a range of medical indicators. By analyzing the model coefficients, important features that are accurate were found, providing information about the main factors linked to heart disease.

A clear understanding of the connections between input characteristic and disease likelihood was made possible by the interpretability of logistic regression software. In order to guarantee optimal performance and application in medical environments, more testing and comparisons to alternative algorithms are necessary, even though the models show promise in their clinical relevance and predictive powers. A basic structure for risk assessment and decision-making in the management and prevention of heart disease is offered by these logistic regression models.

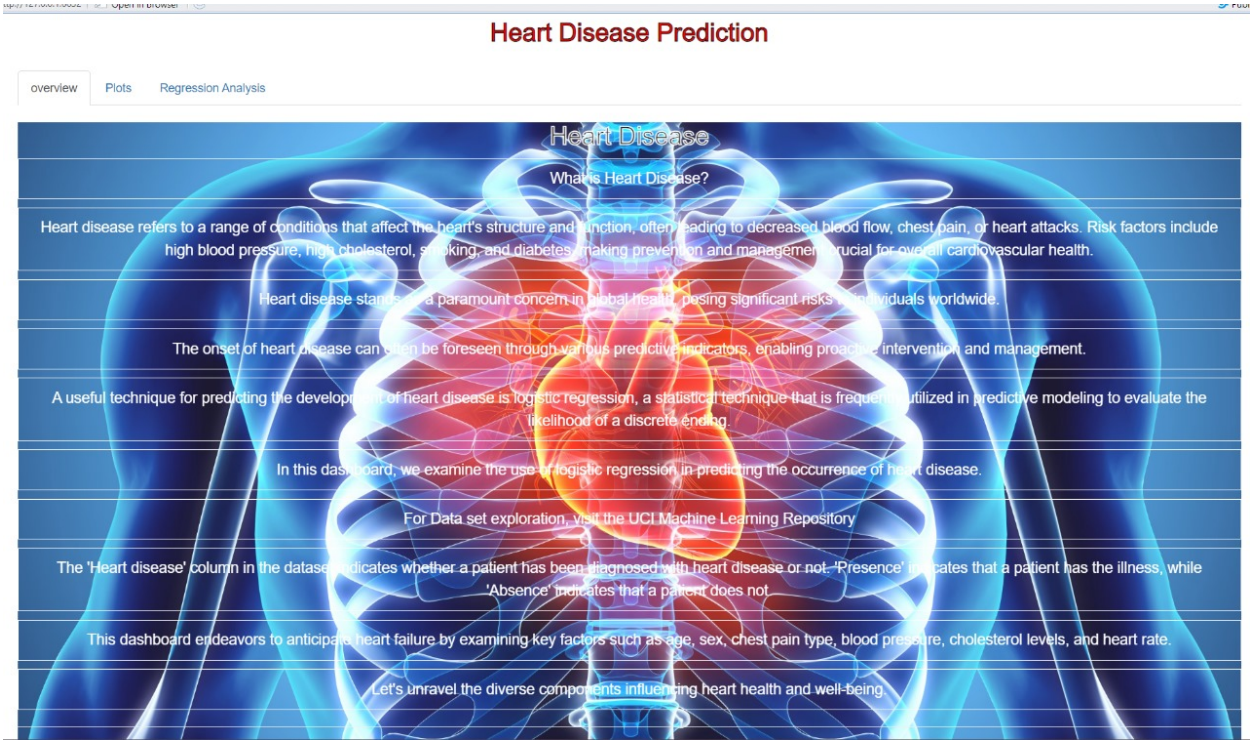
3.Plotly dashboard

3.1 Introduction

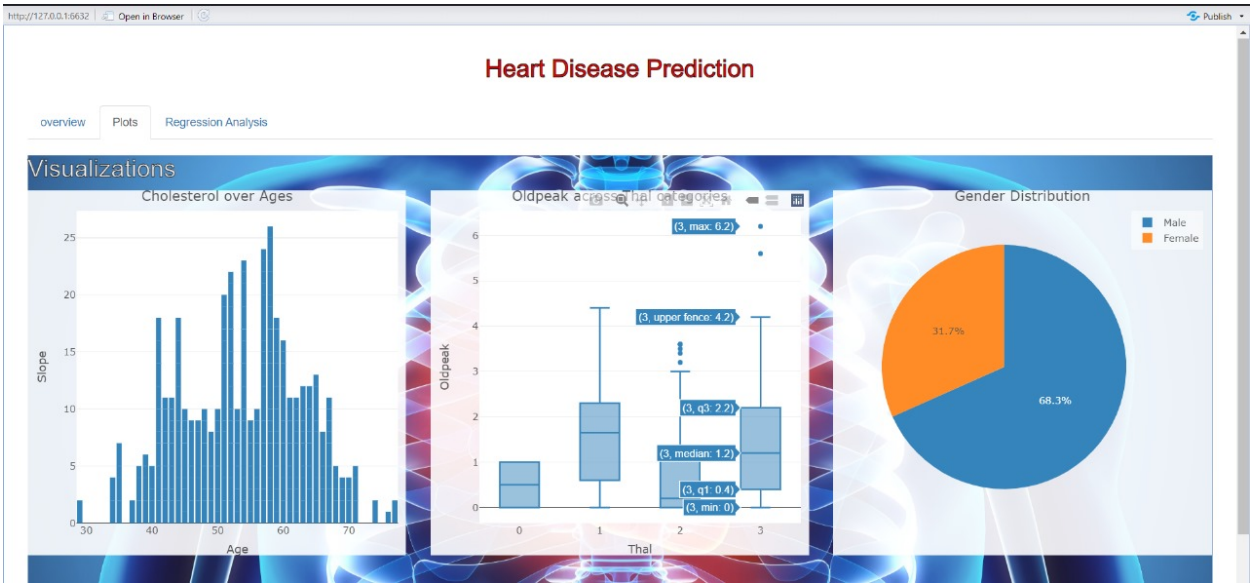
A dashboard is a data visualization tool that makes metrics and key performance indicators (KPIs) visually appealing and easy to understand. It is a crucial tool for performance management and decision-making as it allows users to track progress toward predetermined goals and targets and analyze trends quickly.

We use the heart disease data set in logistic regression to analyze this dashboard.

In this first page, it gives brief introduction about the dashboard and the plots we create.



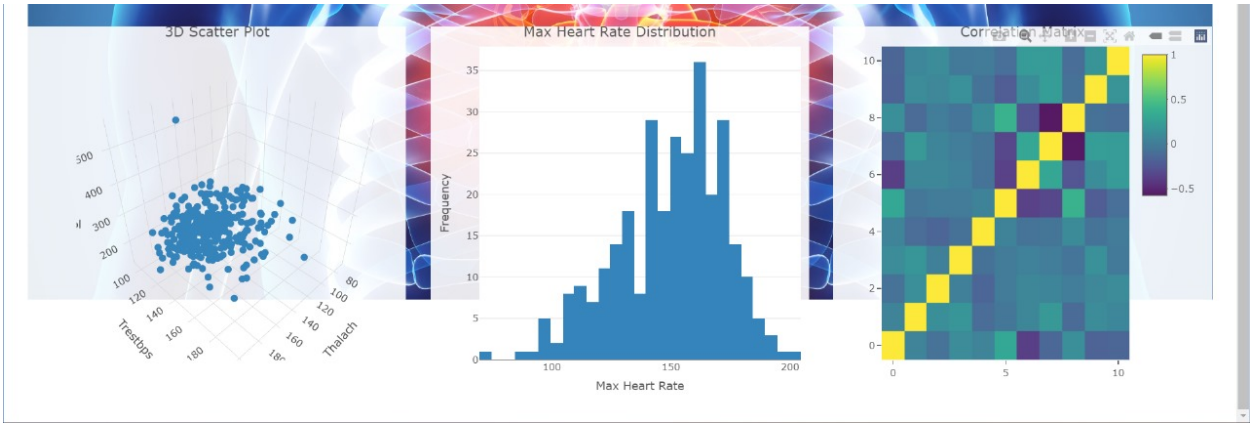
The plots we create to visualize the attributes in the dataset.



The first plot is bar chart. It predicts the type of slope by the age.

The next one is box plot. It predicts the type of thal and the st depression.

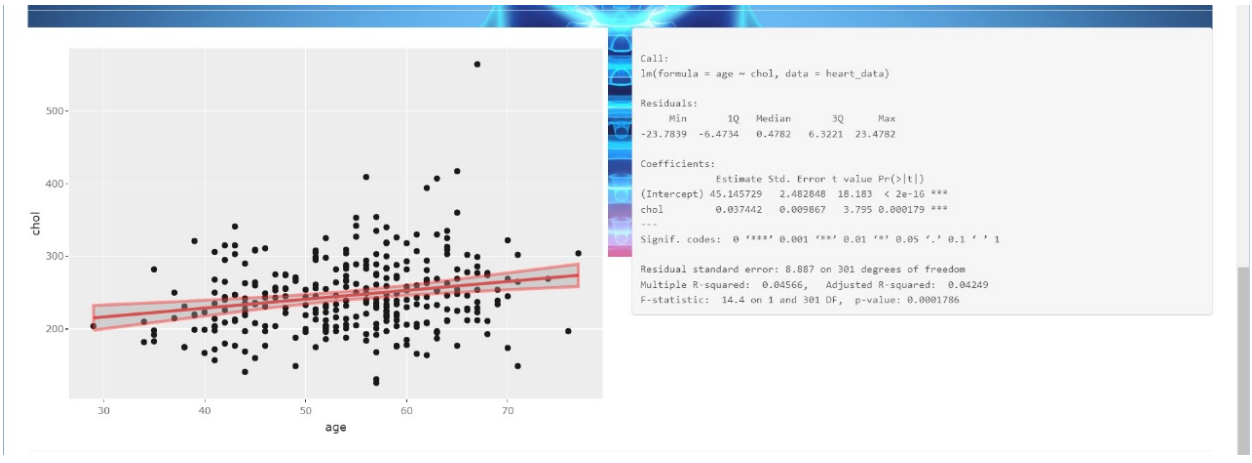
The last plot is pie chart. It defines the gender distribution in the dataset.



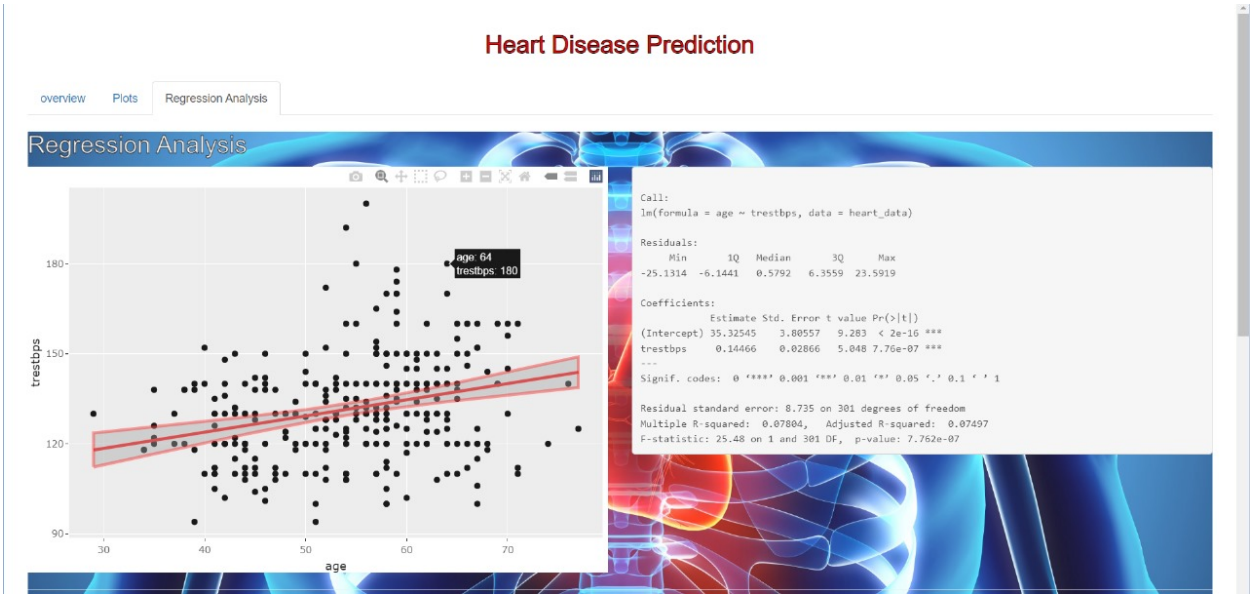
The 3D scatter plot defines the maximum heart rate achieved and the rest in blood pressure of the patient.

The histogram predicts the frequency of the maximum heart rate distribution of the patients.

The correlation matrix depicts the relationship of the attributes in this dataset.

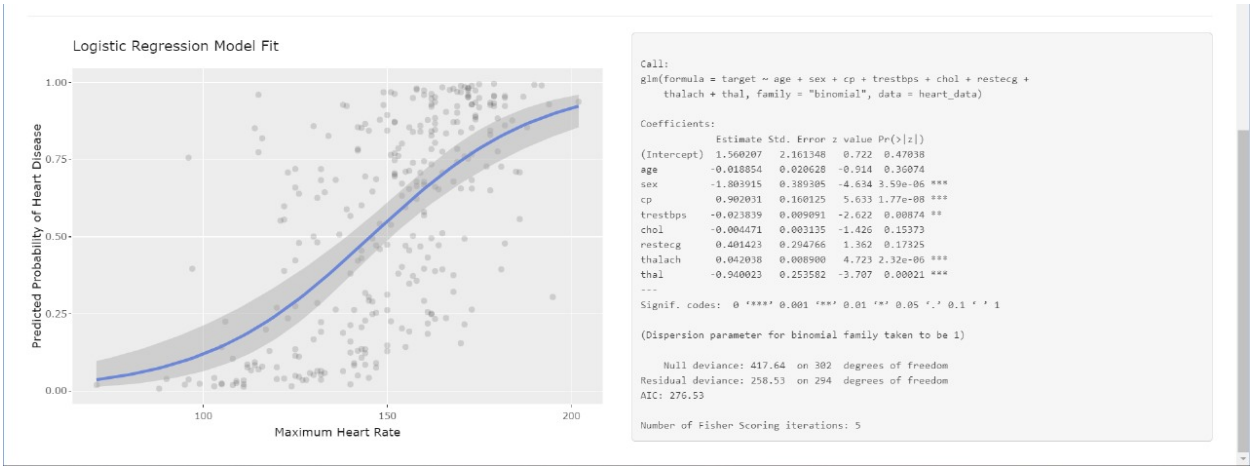


This scatter plot displayed the cholesterol level of the patients. The summary of the linear regression model also displayed.



This scatter plot describes the linear model regression that depicts the relationship rest in blood pressure of the patients according to their age.

The summary of the linear regression model is also displayed.



The last scatter plot depicts the probability of heart disease according to the maximum heart rate. This logistic regression model defines how well our model can be fit to the prediction.

From the following codes we created the dashboard.

```r
# Load required libraries
library(shiny)
library(plotly)
library(ggplot2)
library(readr)
library(dplyr)

# Load the heart disease dataset
heart_data <- read_csv("C:\\Users\\Midara\\Downloads\\regression\\heart.csv")
heart_data$target <- as.factor(heart_data$target)

# Define UI
```

```r
ui <- fluidPage(
  tags$head(
    tags$style(HTML("
      .title-wrapper {
        text-align: center;
        font-size: 3em;
        color:#FF0000; /* Red color */
        -webkit-text-stroke: 1px black;
        width: 100%;
        height: 100px;
        display: flex;
        justify-content: center;
        align-items: center;
      }
      .background {
        background-image: url('https://d.newsweek.com/en/full/1032264/heart-stock.jpg');
        background-size: cover;
        background-position: center;
        background-repeat: no-repeat;
        background-attachment: fixed;
        height: 100vh; /* Set the background height to cover the entire page */
        width: 100%; /* Set the background width to cover the entire page */
        opacity: 0.9;
      }
      .about-content {
        height: 100%;
      }
      .plot-row {
        height: 100%;
      }
```

```
      .plot-row {
        height: 100%;
      }
      .plot-tile {
        padding-bottom: 1cm;
      }
      .plotly .plot-container {
        background-color: rgba(255, 255, 255, 0.8) !important;
      }
    "))
  ),
  div(class = "title-wrapper",
      titlePanel("Heart Disease Prediction")),
  mainPanel(
    style = "width: 100%;",
    style = "height=20px;",
    tags$style(".plot-row { margin-bottom: 1cm; }"),
    tags$style(".plot-tile { padding-bottom: 1cm; }"),
    tabsetPanel(
      tabPanel("overview",
               class = "background",
               align = "center",
               style = "height: 100vh;",
               h2(style = "color: black;-webkit-text-stroke: 1px white;"," Heart Disease "),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"What is Heart Disease?"),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"Heart disease refers to a range of conditions that affect the heart's structure and function, often leading to decreased blood flow,
chest pain, or heart attacks. Risk factors include high blood pressure, high cholesterol, smoking, and diabetes, making prevention and
management crucial for overall cardiovascular health."),
```

```
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white ; border: 1px solid white; padding: 10px;",
"Heart disease stands as a paramount concern in global health, posing significant risks to individuals worldwide."),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"The onset of heart disease can often be foreseen through various predictive indicators, enabling proactive intervention and
management."),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"A useful technique for predicting the development of heart disease is logistic regression, a statistical technique that is frequently
utilized in predictive modeling to evaluate the likelihood of a discrete ending."),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"In this dashboard, we examine the use of logistic regression in predicting the occurrence of heart disease."),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"For Data set exploration, visit the UCI Machine Learning Repository"),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"The 'Heart disease' column in the dataset indicates whether a patient has been diagnosed with heart disease or not. 'Presence' indicates
that a patient has the illness, while 'Absence' indicates that a patient does not."),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"This dashboard endeavors to anticipate heart failure by examining key factors such as age, sex, chest pain type, blood pressure,
cholesterol levels, and heart rate."),
               p(style = "font-size: 18px; line-height: 1.5; margin-bottom: 10px; color: white; border: 1px solid white; padding: 10px;",
"Let's unravel the diverse components influencing heart health and well-being."),

               tags$hr()

      ),
      tabPanel("Plots",
               class = "background",
               h2(style = "color: white;-webkit-text-stroke: 1px black;","Visualizations"),
               fluidRow(
                 style = "height: 40vh;",
                 class = "plot-row",
                 column(width = 4,
```

```
                div(class = "plot-tile",
                    plotlyOutput("plot5", height = "100%", width = "100%"))
            ),
            column(width = 4,
                div(class = "plot-tile",
                    plotlyOutput("plot6", height = "100%", width = "100%"))
            )
          )
  ),
  tabPanel("Regression Analysis",
           class = "background",
           h2(style = "color: white;-webkit-text-stroke: 1px black;","Regression Analysis"),
           fluidRow(
             column(width = 6,
                    plotlyOutput("plot_lm1", height = "500px")
             ),
             column(width = 6,
                    verbatimTextOutput("regression_summary1")
             )
           ),
           tags$hr(),
           fluidRow(
             column(width = 6,
                    plotlyOutput("regression_plot2", height = "500px")
             ),
             column(width = 6,
                    verbatimTextOutput("regression_summary2")
             )
           ),
           tags$hr(),
           fluidRow(
```

```
             fluidRow(
               column(width = 6,
                      plotlyOutput("regression_plot3", height = "500px")
               ),
               column(width = 6,
                      verbatimTextOutput("regression_summary3")
               )
             ),
             tags$hr()
    )
  )
 )
)
```

```
# Define server logic
server <- function(input, output, session) {
  # Scatter plot for Age vs. sex with color by target variable
  output$plot1 <- renderPlotly({
    plot_ly(data = heart_data, x = ~age, y = ~slope, type = "bar", mode = "markers",
            marker = list(size = 10, opacity = 1)) %>%
      layout(title = "Cholesterol over Ages",
             xaxis = list(title = "Age"),
             yaxis = list(title = "Slope"),
             legend = list(title = "Heart Disease"),
             paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
             plot_bgcolor = "rgba(255, 255, 255, 1.0)") # Adjust plot area background color with opacity
  })

  # Box plot for Oldpeak across Thal categories
  output$plot2 <- renderPlotly({
    plot_ly(data = heart_data, x = ~factor(thal), y = ~oldpeak, type = "box") %>%
      layout(title = "type of thal according to ST depression",
             xaxis = list(title = "Thal"),
             yaxis = list(title = "Oldpeak"),
             paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
             plot_bgcolor = "rgba(255, 255, 255, 1.0)") # Adjust plot area background color with opacity
  })

  # Pie chart for Gender Distribution
  output$plot3 <- renderPlotly({
    counts <- table(heart_data$sex)
    labels <- c("Female", "Male")
    plot_ly(labels = labels, values = counts, type = "pie") %>%
      layout(title = "Gender Distribution",
             paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
```

```
             paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
             plot_bgcolor = "rgba(255, 255, 255, 1.0)") # Adjust plot area background color with opacity
  })

  # 3D scatter plot for Thalach, Trestbps, and Chol
  output$plot4 <- renderPlotly({
    plot_ly(data = heart_data, x = ~thalach, y = ~trestbps, z = ~chol,
            type = "scatter3d", mode = "markers", marker = list(size = 5, opacity = 1)) %>%
      layout(title = "3D Scatter Plot",
             scene = list(xaxis = list(title = "Thalach"),
                          yaxis = list(title = "Trestbps"),
                          zaxis = list(title = "Chol")),
             paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
             plot_bgcolor = "rgba(255, 255, 255, 1.0)") # Adjust plot area background color with opacity
  })

  # Histogram for Max Heart Rate Distribution
  output$plot5 <- renderPlotly({
    plot_ly(data = heart_data, x = ~thalach, type = "histogram") %>%
      layout(title = "Max Heart Rate Distribution",
             xaxis = list(title = "Max Heart Rate"),
             yaxis = list(title = "Frequency"),
             paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
             plot_bgcolor = "rgba(255, 255, 255, 1.0)") # Adjust plot area background color with opacity
  })
```

```
# Heatmap for Correlation Matrix
output$plot6 <- renderPlotly({
  corr_matrix <- cor(select(heart_data, -c(age, sex,target)))
  plot_ly(z = corr_matrix, colorscale = "Viridis", type = "heatmap") %>%
    layout(title = "Correlation Matrix",
           paper_bgcolor = "rgba(255, 255, 255, 0.5)", # Adjust background color with opacity
           plot_bgcolor = "rgba(255, 255, 255, 1.0)") # Adjust plot area background color with opacity
})

# Add server logic for regression analysis plots and summaries
# Model 1:
output$plot_lm1 <- renderPlotly({
  ggplot(heart_data, aes(x = age, y = trestbps)) +
    geom_point() +
    geom_smooth(method = "lm", color = "red")
})

output$regression_summary1  <- renderPrint({
  lm1 <- lm(age ~trestbps, data = heart_data)
  summary(lm1)
})

# Model 2:
output$regression_plot2 <- renderPlotly({
  ggplot(heart_data, aes(x = age, y = chol)) +
    geom_point() +
    geom_smooth(method = "lm", color = "red")
})
```

```
output$regression_summary2<- renderPrint({
  lm2 <- lm(age ~ chol, data = heart_data)
  summary(lm2)
})



output$regression_summary3 <- renderPrint({
  model_glm <- glm(target~ age+sex+cp+trestbps+chol+restecg+thalach+thal,
                   data = heart_data, family = "binomial")
  heart_data$predicted_prob <- predict(model_glm, type = "response")
  summary(model_glm)
})
# Model 3:
output$regression_plot3<- renderPlotly({
  # Fit logistic regression model
  model_glm <- glm(target ~ age + sex + cp + trestbps + chol + restecg + thalach + thal,
                   data = heart_data, family = "binomial")

  # Predict probabilities using the model
  heart_data$predicted_prob <- predict(model_glm, type = "response")
  heart_data %>%
    ggplot(aes(x = thalach, y =predicted_prob,)) +
    geom_point(alpha = 0.15) +
    geom_smooth(method = "glm", method.args = list(family = "binomial")) +
    ggtitle("Logistic Regression Model Fit") +
    xlab("Maximum Heart Rate") +
    ylab("Predicted Probability of Heart Disease")
})
```

```
  output$regression_summary3 <- renderPrint({
    model_glm <- glm(target ~ age+sex+cp+trestbps+chol+restecg+thalach+thal,
                     data = heart_data, family = "binomial")
    summary(model_glm)
  })
}

# Run the application
shinyApp(ui = ui, server = server)
```

3.2 Conclusion

It is possible to clearly grasp how input features relate to the likelihood of sickness thanks to the interpretability provided by logistic regression software. Still, more testing and comparisons with other algorithms are required to make sure these models work as well as possible in medical contexts. The models' predictive power and therapeutic relevance are encouraging, even with this need for more assessment. The core framework for risk assessment and decision-making in the management and prevention of heart disease is offered by these logistic regression models.

4. References

1. Association rule mining

Imielinski, T., Agrawal, R., and Swami, A. (1993). mining association rules in big datasets between groups of objects. ACM SIGMOD International Conference on Management of Data, Proceedings, 1993, pp. 207–216. ACM.

Pei, J., Yin, Y., and Han, J. (2000). mining common patterns without producing any candidates. ACM SIGMOD International Conference on Data Management, 2000: Proceedings (pp. 1–12). ACM.

Mannila, H., Agrawal, R., and others (1996). Quick identification of association rules. Progress in Information Exploration and Data Analysis (pages 307–328). Published by AAAI/MIT Press.

2. Logistic regression

David W. Hosmer Jr., Stanley Lemeshow, and Rodney X. Sturdivant, "Applied Logistic Regression," (2013)

J. Scott Long and Jeremy Freese, Regression Models for Categorical Dependent Variables Using Stata (2005)