7BUIS008C.2 Data Mining and Machine Learning

Coursework Two

Vishvaka Neomal Ranasinghe

2019677 (IIT)

w1790596 (UOW)

1st Task: Data Set Selection and Visualization

Dataset used: UCI Heart disease [link: https://archive.ics.uci.edu/ml/datasets/heart+Disease]

Introduction to the dataset:

This dataset gives 13 variables along with a target condition of having or not having heart disease.

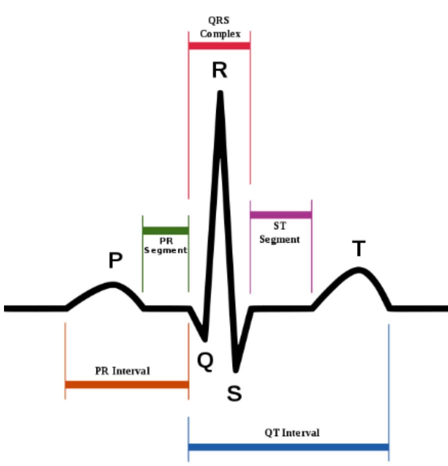
```
> heart<-read.csv("D:/Projects/heart.csv")
> names(heart)
 [1] "age"      "sex"      "cp"       "trestbps" "chol"     "fbs"      "restecg"  "thalach"  "exang"
[10] "oldpeak"  "slope"    "ca"       "thal"     "target"
```

The column definitions are as follows:

- age: Age of the patient in years.
- sex: 1 = male and 0 = female
- cp: Chest pain type
    - 1 → Typical Angina
    - 2 → Atypical Angina
    - 3 → Non-anginal Pain
    - 4 → Asymptomatic

Angina, also known as angina pectoris, is chest pain or pressure, usually due to not enough blood flow to the heart muscle. (Source: Wikipidia)

- trestbps: Resting blood pressure in mm Hg on admission to the hospital
- chol: Serum cholesterol in mg/dl
- fbs: Fasting blood sugar level greater than 120 mg/dl
    - 0 → False
    - 1 → True
- restecg: Resting electrocardiographic results
    - 0 → Normal
    - 1 → Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)



In electrocardiography, the ST segment connects the QRS complex and the T wave and has a duration of 5ms to 150ms.

<u>Interpretation</u>

- The normal ST segment has a slight upward concavity.
- Flat, down sloping, or depressed ST segments may indicate coronary ischemia.

(Source: Wikipidia)

    - 2 → Showing probable or definite left ventricular hypertrophy by [Estes' criteria]().
- thalach: Maximum heart rate achieved in beats per minute
- exang: Exercise induced angina
    - 0 → No
    - 1 → Yes
- oldpeak: ST depression induced by exercise relative to rest
- slope: The slope of the peak exercise ST segment
    - 1 → Upsloping
    - 2 → Flat
    - 3 → Down sloping
- ca: Number of major vessels [0-3] colored by fluoroscopy
- thal: Thalium stress test result
    - 1 → Fixed defect
    - 2 → Normal
    - 3 → Reversable defect
- num: Diagnosis of heart disease (angiographic disease status)
    - 0 → Less than 50% diameter narrowing
    - 1 → Greater than 50% diameter narrowing

str(heart)

```
> 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  : int  1 1 1 1 1 1 1 1 1 1 ...
```

head(heart)

```
> head(heart)
    age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal target
254  67   1  0      100  299   0       0     125     1     0.9     1  2    2      0
54   44   0  2      108  141   0       1     175     0     0.6     1  0    2      1
259  62   0  0      150  244   0       1     154     1     1.4     1  0    2      0
44   53   0  0      130  264   0       0     143     0     0.4     1  0    2      1
291  61   1  0      148  203   0       1     161     0     0.0     2  1    3      0
200  65   1  0      110  248   0       0     158     0     0.6     2  2    1      0
```

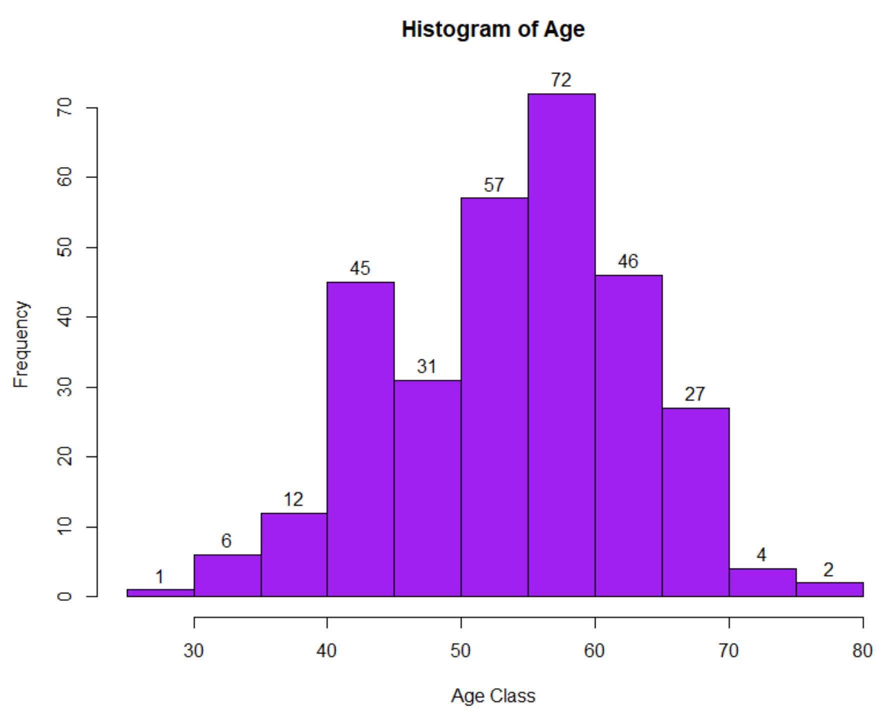
```
summary(heart)
```

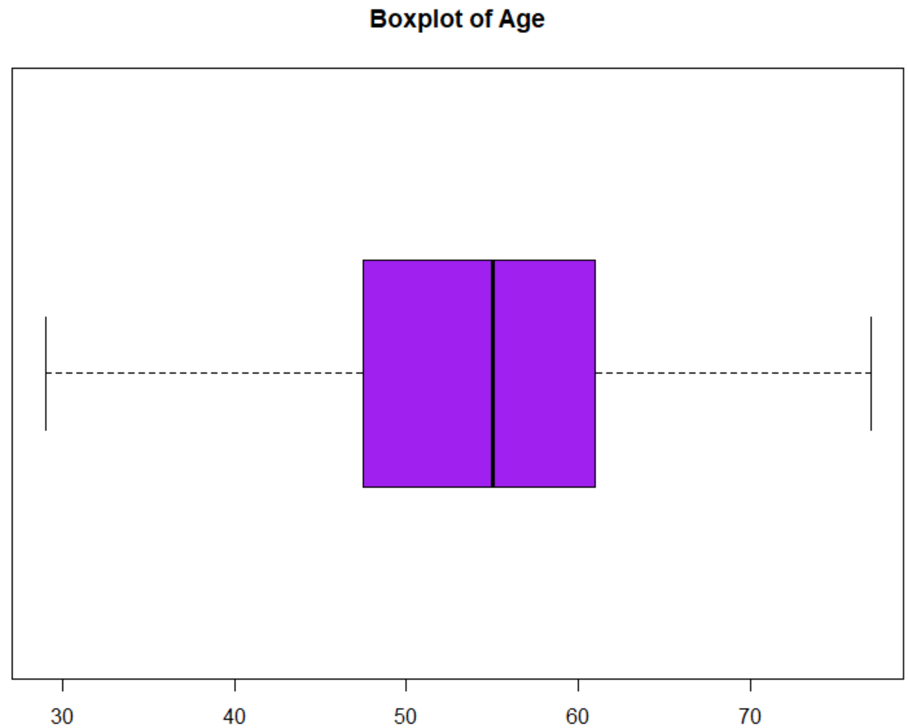
```
> summary(heart)
      age             sex               cp            trestbps          chol            fbs        
 Min.   :29.00   Min.   :0.0000   Min.   :0.000   Min.   : 94.0   Min.   :126.0   Min.   :0.0000  
 1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:120.0   1st Qu.:211.0   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   Min.   :0.0000  
 1st Qu.:2.000   1st Qu.:0.0000  
 Median :2.000   Median :1.0000  
 Mean   :2.314   Mean   :0.5446  
 3rd Qu.:3.000   3rd Qu.:1.0000  
 Max.   :3.000   Max.   :1.0000  
```

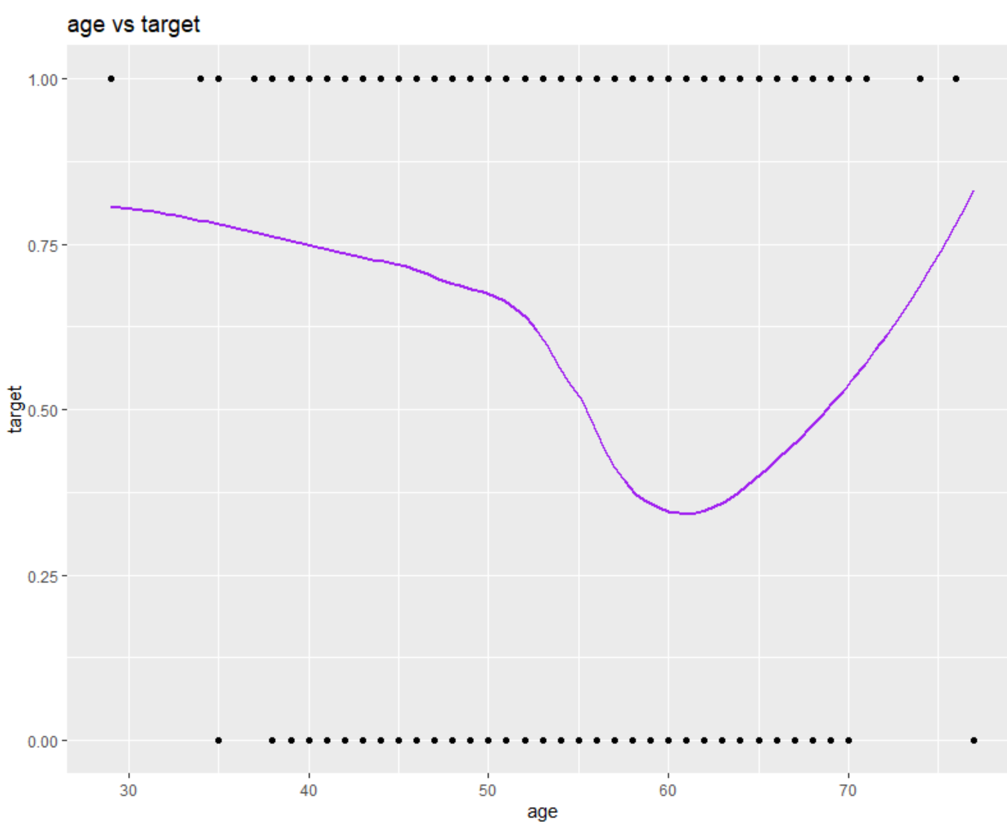
Age Analysis

```
> range(heart$age)
[1] 29 77
> summary(heart$age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  29.00   47.50   55.00   54.37   61.00   77.00 
> sd(heart$age)
[1] 9.082101
> var(heart$age)
[1] 82.48456
> cor(heart$age,heart$target)
[1] -0.2254387
> chisq.test(heart$age,heart$target)

        Pearson's Chi-squared test

data:  heart$age and heart$target
X-squared = 50.129, df = 40, p-value = 0.1309

Warning message:
In chisq.test(heart$age, heart$target) :
  Chi-squared approximation may be incorrect
> hist(heart$age,labels=TRUE,main="Histogram of Age",xlab="Age Class",ylab="Frequency",col="purple")
> boxplot(heart$age,horizontal=TRUE,col="purple",main="Boxplot of Age")
```

- Minimum age is 29 and maximum age is 77, average age is 54.37. Majority of the population is between age group 55 and 60 years.
- There is negative correlation between age and target. This implies that when get older probability of heart attack is decreasing.
- By observing the curve, we can see that from age 30 to 60 probability of heart attack is decreasing and from 60 again probability is increasing. After 70 chance of heart attack is more.
- Using Chi squared test we get a probability value of 0.13. There for we can conclude that target is independent of the age.



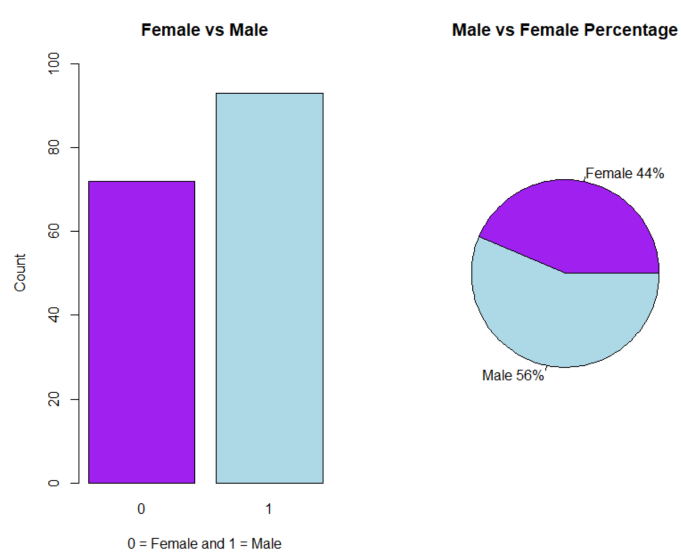

Boxplot of Age
30
40
50
60
70

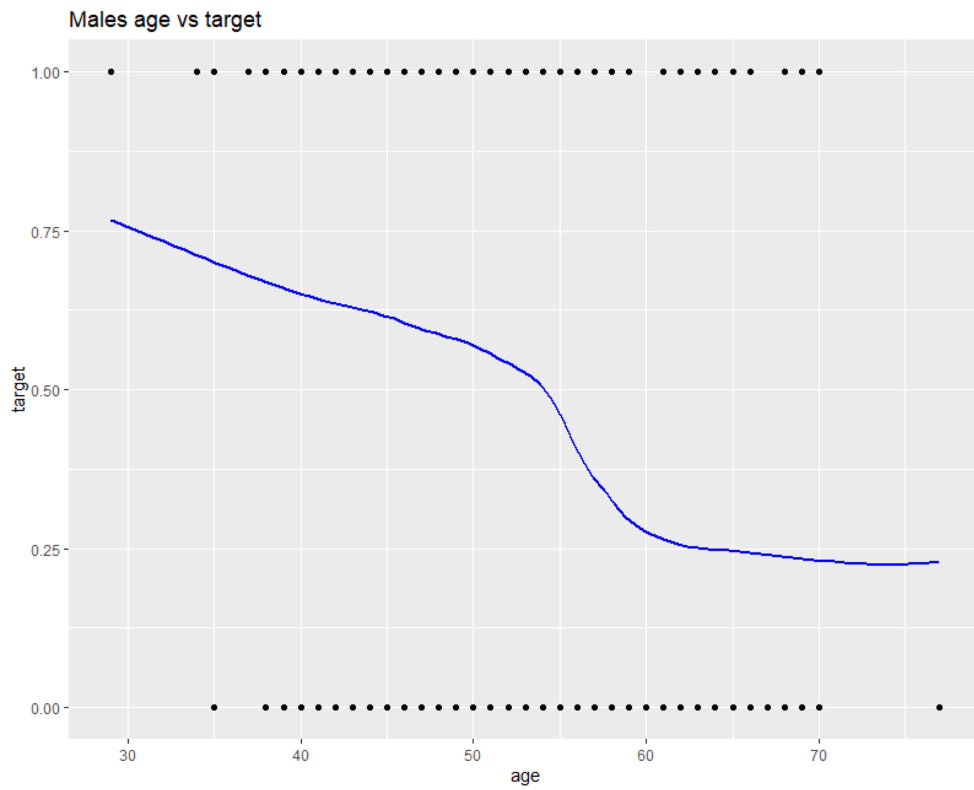

age vs target
1.00
0.75
0.50
0.25
0.00
target
30
40
50
60
70
age

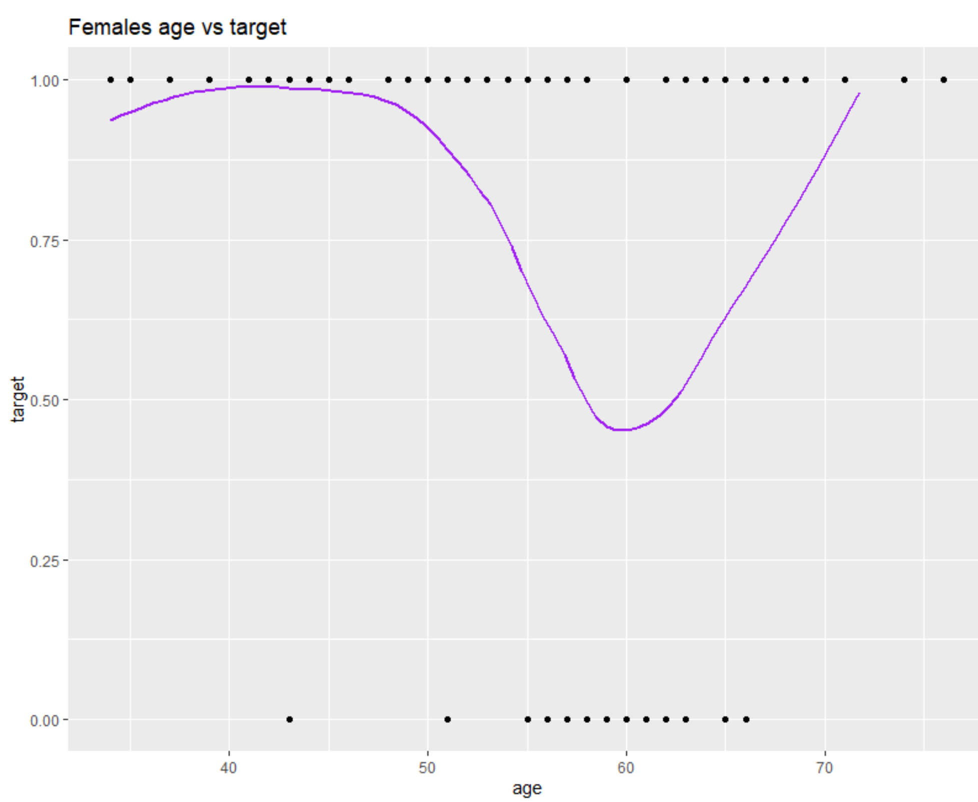
Gender(sex) Analysis

```
> heart_attacks <- heart[heart$target==1,]
> length(heart_attacks$sex)
[1] 165
> length(heart_attacks[heart_attacks$sex==0,]$sex)
[1] 72
> length(heart_attacks[heart_attacks$sex==1,]$sex)
[1] 93
>
> a <- table(heart_attacks$sex)
> par(mfrow=c(1,2))
> barplot(a,
+         col=c("purple","lightblue"),
+         xlab="0 = Female and 1 = Male",
+         ylab="Count",
+         ylim=range(pretty(c(0, a))), #to adjust y-axis scale
+         main="Female vs Male")
>
> percentages <- round(a/sum(a)*100)
> labels <- paste(c("Female","Male")," ",percentages,"%",sep="")
> pie(a,labels=labels,col=c("purple","lightblue"),main="Male vs Female Percentage")
>
> males <- heart[heart$sex==1,]
> a <- ggplot(males,aes(x=age,y=target)) + geom_point() + geom_smooth(color="blue")
> b <- a+scale_x_continuous(name="age") + scale_y_continuous(name="target")
> b + ggtitle("Males age vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
>
> females <- heart[heart$sex==0,]
> a <- ggplot(females,aes(x=age,y=target)) + geom_point() + geom_smooth(color="purple")
> b <- a+scale_x_continuous(name="age") + scale_y_continuous(name="target")
> b + ggtitle("Females age vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- Out of 165 Heart Attack observations 72 are Female and 93 are male.
- Looking at the barplot we can see that males have a higher proportion when compared to female. Also, in Pie chart we can see that proportion of female is 44% while male is 56%.
- Using "*males age vs target*" plot we can see that there is a significant drop in the probability of a heart attack as males grow older.
- As for females there is a drop of the probability of a heart attack from 46 to 60 years and then rapidly increases as they grow older.



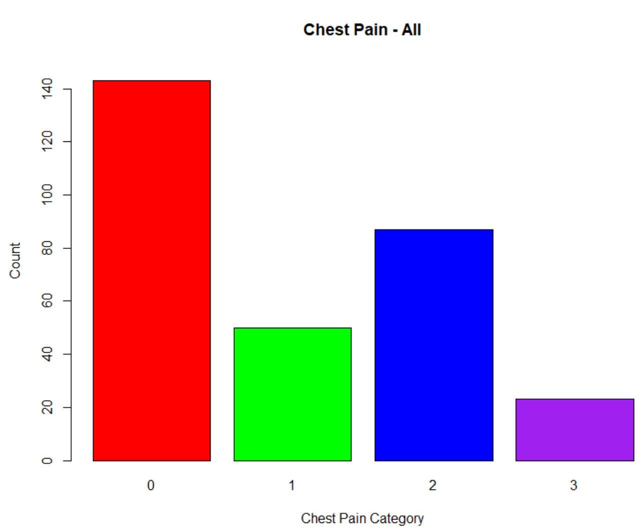

Males age vs target
target
1.00
0.75
0.50
0.25
0.00
30
40
50
60
70
age

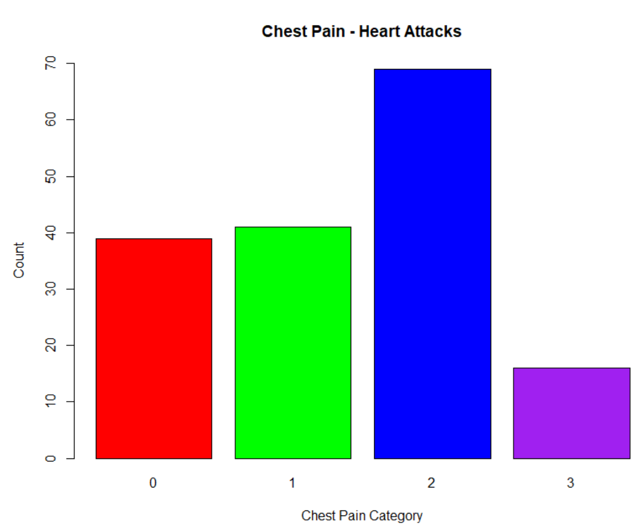

Females age vs target
target
1.00
0.75
0.50
0.25
0.00
40
50
60
70
age

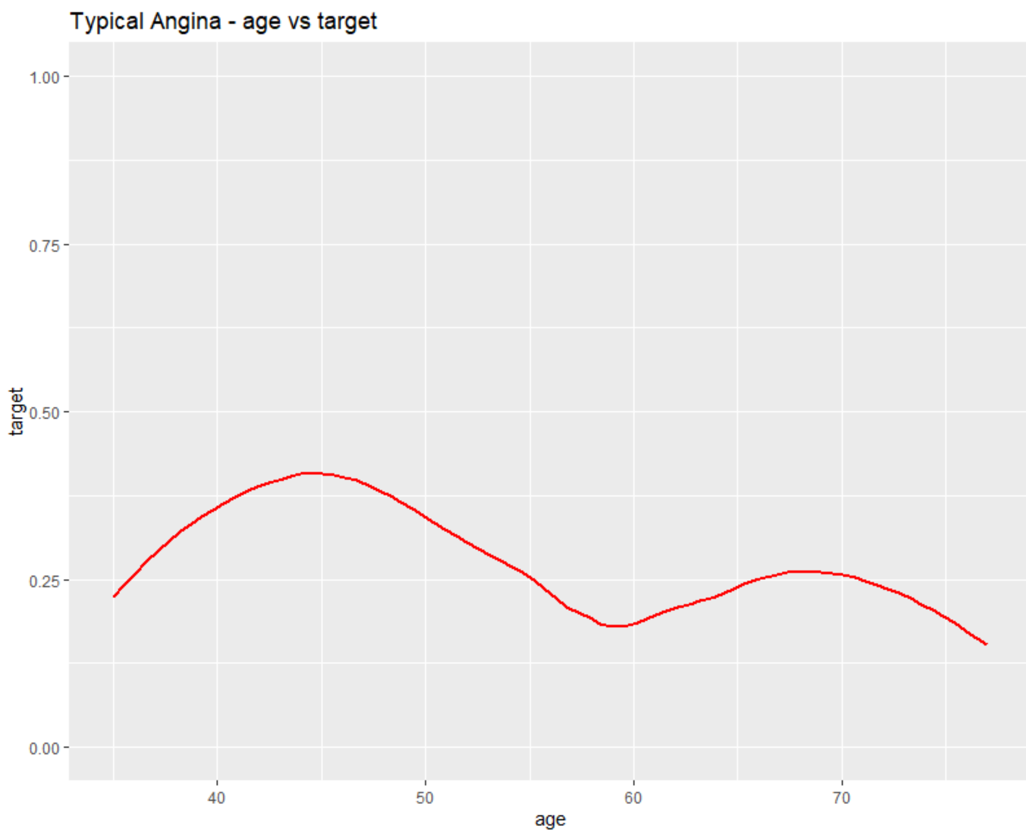
Chest Pain(cp) Analysis

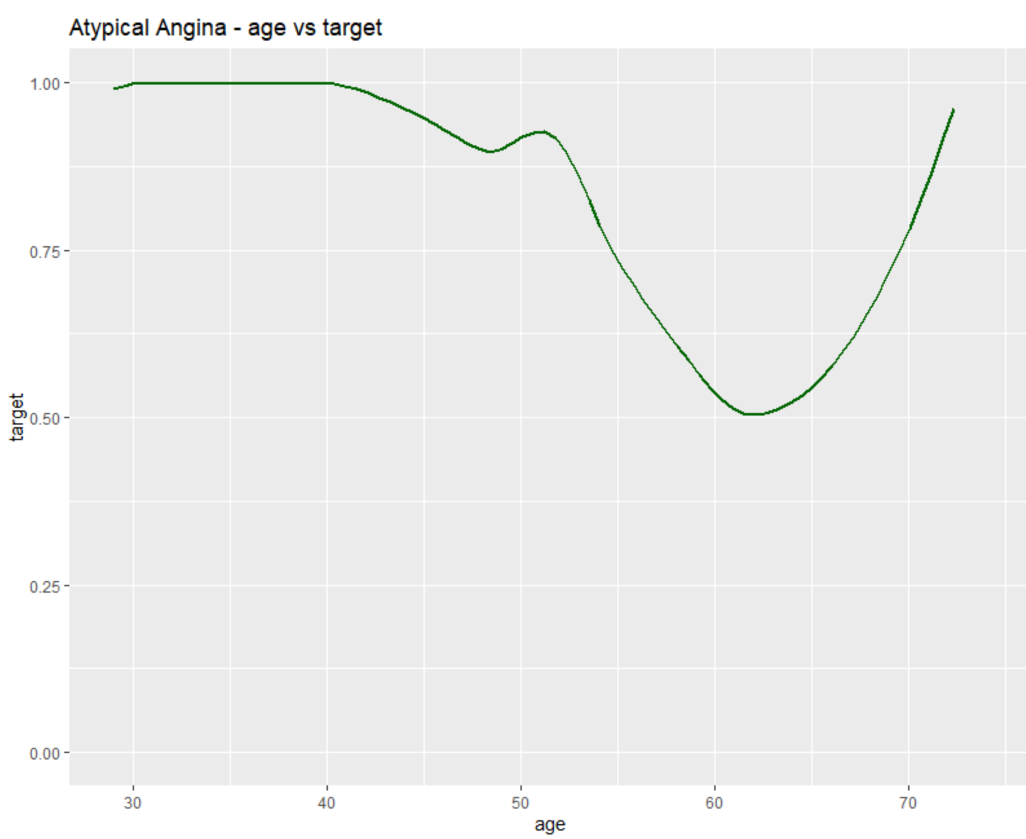
```
> heart_attacks <- heart[heart$target==1,]
> heart$cp <- factor(heart$cp)
> heart_attacks$cp <- factor(heart_attacks$cp)
>
> a <- table(heart$cp)
> barplot(a,
+         col=c("red","green","blue","purple"),
+         main="Chest Pain - All",
+         xlab="Chest Pain Category",
+         ylab="Count",
+         ylim=range(pretty(c(0, a))) #to adjust y-axis scale
+         )
>
> b <- table(heart_attacks$cp)
> barplot(b,
+         col=c("red","green","blue","purple"),
+         main="Chest Pain - Heart Attacks",
+         xlab="Chest Pain Category",
+         ylab="Count",
+         ylim=range(pretty(c(0, b))) #to adjust y-axis scale
+         )
>
> typical_angina <- heart[heart$cp==0,]
> a <- ggplot(typical_angina,aes(x=age,y=target)) + geom_smooth(color="red",se=FALSE)
> b <- a+scale_x_continuous(name="age") + scale_y_continuous(name="target",limit=c(0,1))
> b + ggtitle("Typical Angina - age vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
>
> atypical_angina <- heart[heart$cp==1,]
> a <- ggplot(atypical_angina,aes(x=age,y=target)) + geom_smooth(color="darkgreen",se=FALSE)
> b <- a+scale_x_continuous(name="age") + scale_y_continuous(name="target",limit=c(0,1))
> b + ggtitle("Atypical Angina - age vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
>
> nonanginal_pain <- heart[heart$cp==2,]
> a <- ggplot(nonanginal_pain,aes(x=age,y=target)) + geom_smooth(color="blue",se=FALSE)
> b <- a+scale_x_continuous(name="age") + scale_y_continuous(name="target",limit=c(0,1))
> b + ggtitle("Non-anginal Pain - age vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
>
> asymptomatic <- heart[heart$cp==3,]
> a <- ggplot(asymptomatic,aes(x=age,y=target)) + geom_smooth(color="purple",se=FALSE)
> b <- a+scale_x_continuous(name="age") + scale_y_continuous(name="target",limit=c(0,1))
> b + ggtitle("Asymptomatic - age vs target")
```

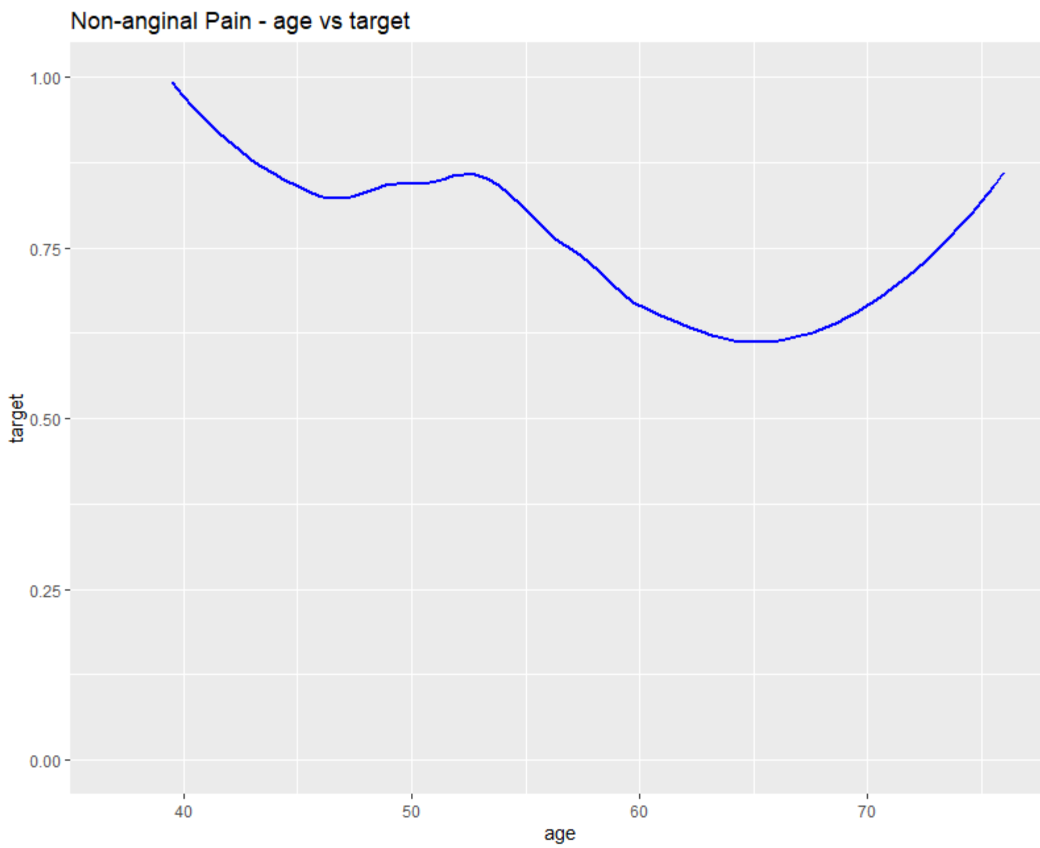
- There is 4 category of Chest pain starting from 0 up to 3.
- From barplot we can observer that, most people have typical angina (Type 0) chest pain, but most of the people who had non-anginal (Type 2) chest pain had a heart attack.

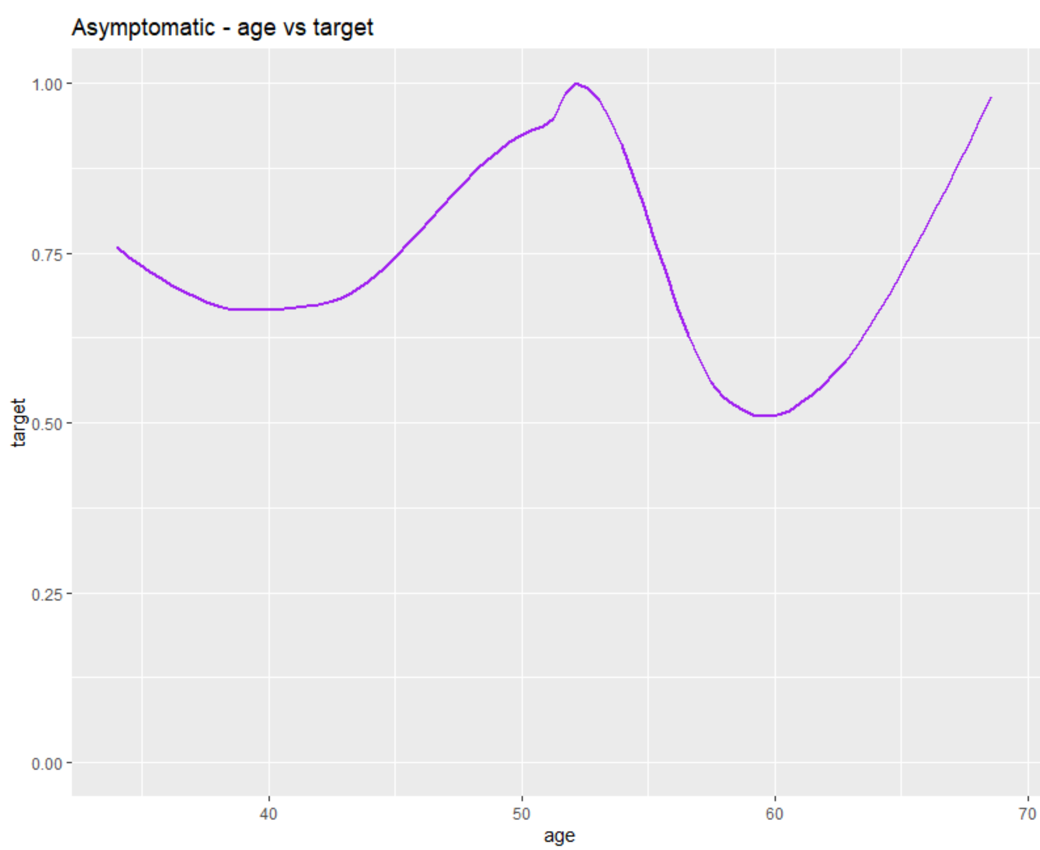





Typical Angina - age vs target
1.00
0.75
0.50
0.25
0.00
target
40
50
60
70
age


Atypical Angina - age vs target
1.00
0.75
0.50
0.25
0.00
target
30
40
50
60
70
age

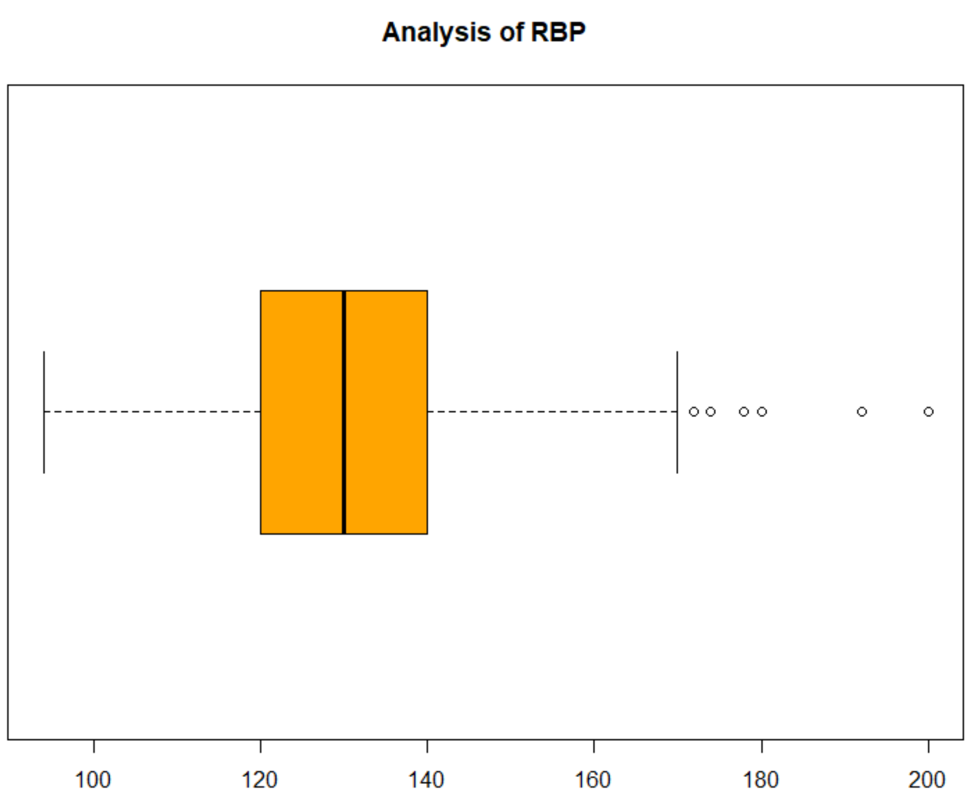

Non-anginal Pain - age vs target
1.00
0.75
0.50
0.25
0.00
target
40
50
60
70
age

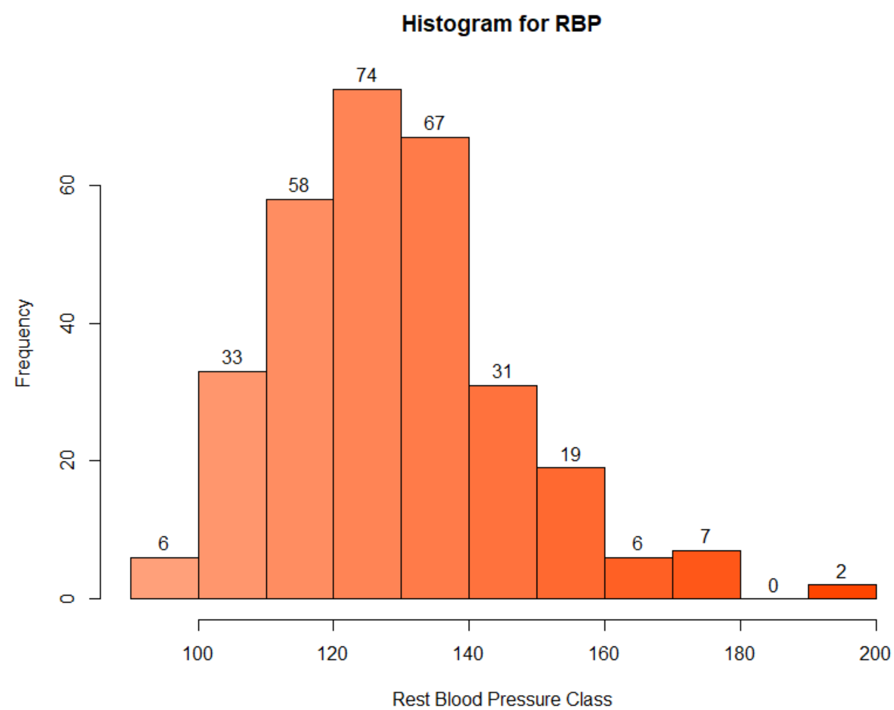

Asymptomatic - age vs target
1.00
0.75
0.50
0.25
0.00
target
40
50
60
70
age

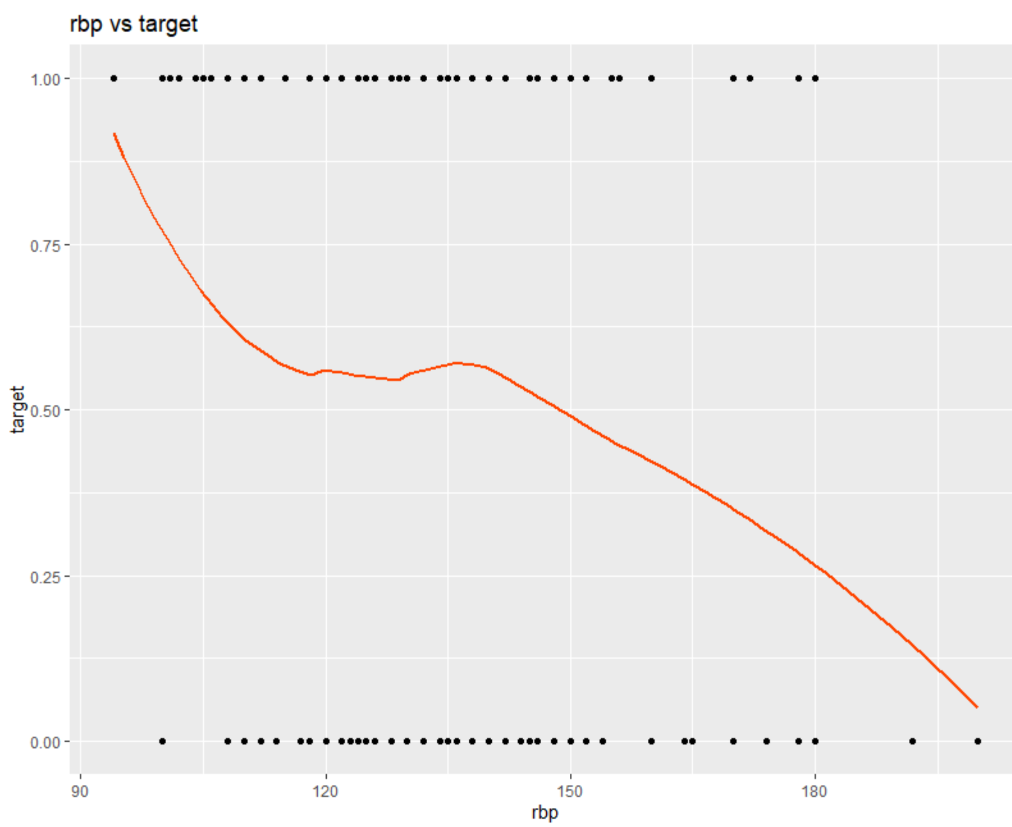
Rest Blood Pressure(trestbps) Analysis

```
> summary(heart$trestbps)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   94.0   120.0   130.0   131.6   140.0   200.0 
> cor(heart$trestbps,heart$age)
[1] 0.2793509
> boxplot(heart$trestbps,
+         col="orange",
+         main="Analysis of RBP",
+         horizontal=TRUE)
> 
> colfunc <- colorRampPalette(c("lightsalmon", "orangered"))
> hist(heart$trestbps,col=colfunc(11),
+      main="Histogram for RBP",
+      xlab="Rest Blood Pressure Class",
+      ylab="Frequency",
+      labels=TRUE)
> 
> a <- ggplot(heart,aes(x=trestbps,y=target))+geom_point()+geom_smooth(color="orangered",se=FALSE)
> b <- a+scale_x_continuous(name="rbp")+scale_y_continuous(name="target",limit=c(0,1))
> b + ggtitle("rbp vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- Minimum resting blood pressure is 94, maximum is 200 and average RBP is 131.6.
- There is low positive correlation between RBP and Target, on increasing resting blood pressure chance of getting heart attack will increase.
- We can clearly see in histogram; Maximum number of Population have Rest Blood Pressure between 120 and 140.
- People having RBP between 95 and 110 are more likely to get Heart Attack.
- By observing the curve of RBP vs Target, probability of a probability is decreasing after RBP 135.



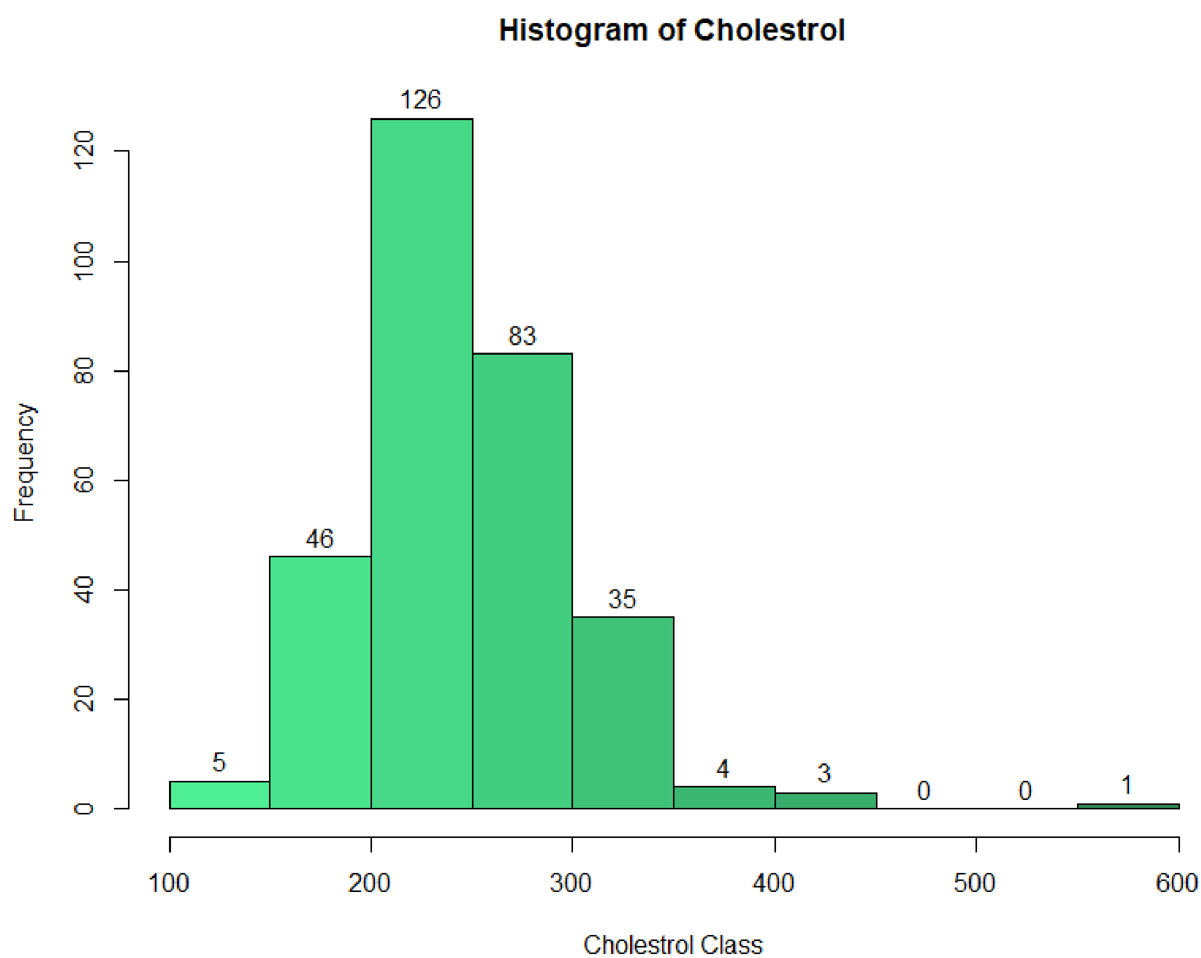

Histogram for RBP
Frequency
Rest Blood Pressure Class
6
33
58
74
67
31
19
6
7
0
2
0
20
40
60
100
120
140
160
180
200

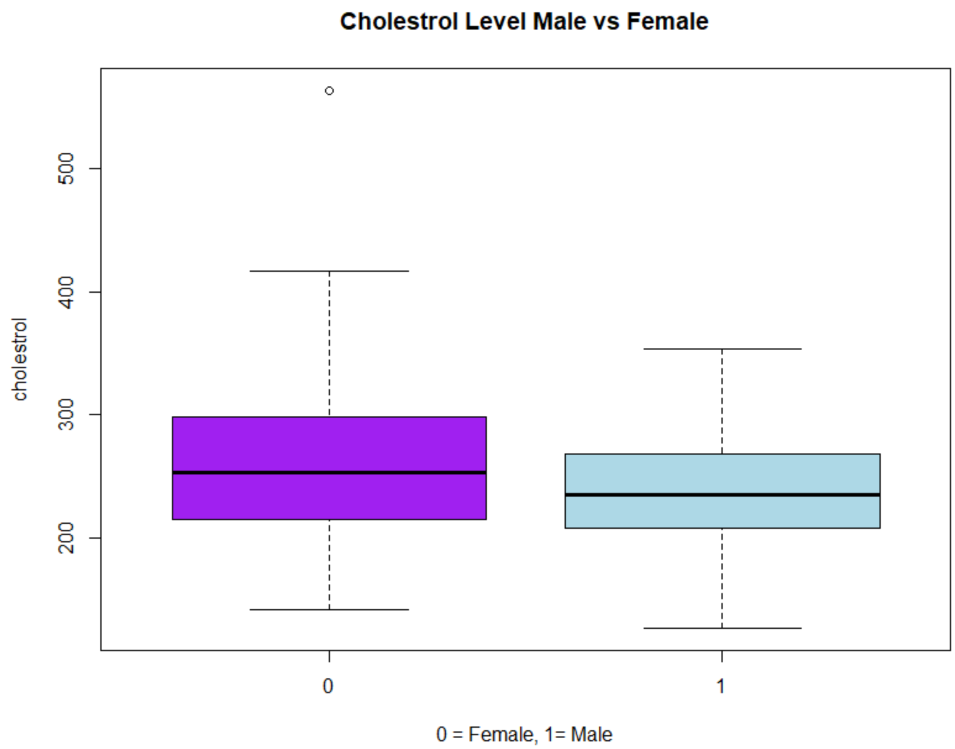

rbp vs target
target
rbp
1.00
0.75
0.50
0.25
0.00
90
120
150
180

Serum Cholesterol(chol) Analysis

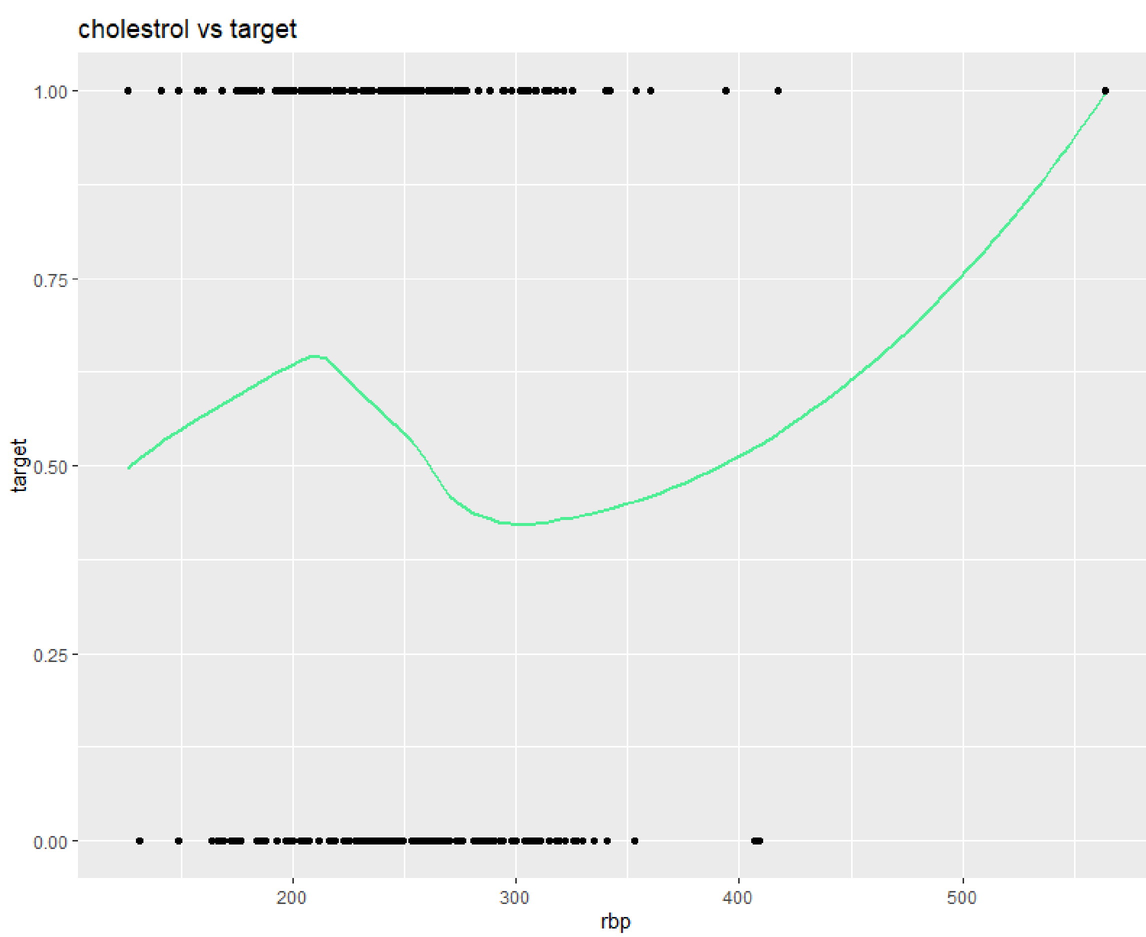
```
> summary(heart$chol)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  126.0   211.0   240.0   246.3   274.5   564.0 
> boxplot(heart$chol~heart$sex,col=c("purple","lightblue"),
+         main="Cholestrol Level Male vs Female",
+         xlab="0 = Female, 1= Male",
+         ylab="cholestrol")
> 
> colfunc <- colorRampPalette(c("seagreen2", "seagreen4"))
> hist(heart$chol,
+      main="Histogram of Cholestrol",
+      xlab="Cholestrol Class",
+      ylab="Frequency",
+      col=colfunc(10),labels=TRUE)
> 
> a <- ggplot(heart,aes(x=chol,y=target))+geom_point()+geom_smooth(color="seagreen2",se=FALSE)
> b <- a+scale_x_continuous(name="rbp")+scale_y_continuous(name="target",limit=c(0,1))
> b + ggtitle("cholestrol vs target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- Minimum cholesterol level is 126, maximum is 564 and average is 246.3.
- We can see Boxplot for analysis separately for males and females, and from it we can observe that males have lower cholesterol than females.
- We can observe in histogram that maximum population have cholesterol between 200 and 250.
- In smooth curve we can clearly see that probability of heart attack is increasing after cholesterol level 300.





Cholestrol Level Male vs Female
cholestrol
200
300
400
500
0
1
0 = Female, 1= Male

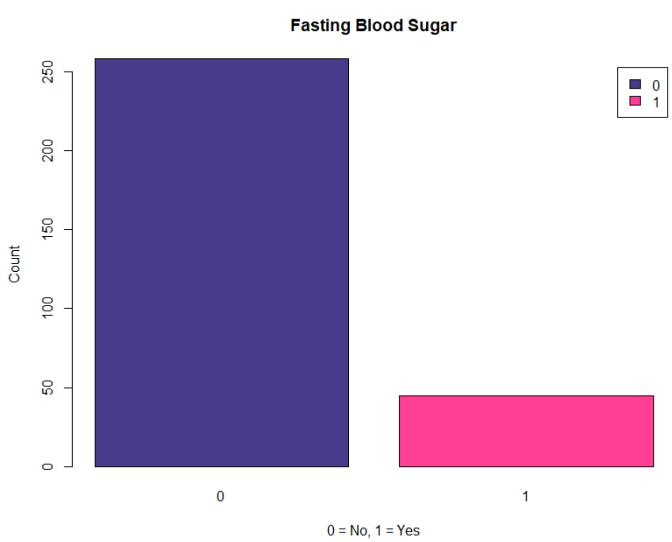


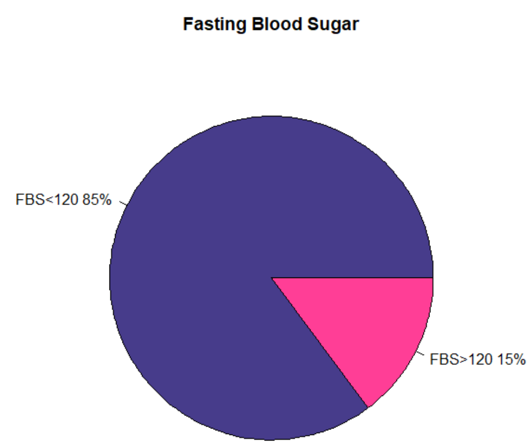
cholestrol vs target
target
0.00
0.25
0.50
0.75
1.00
200
300
400
500
rbp

Fasting Blood Sugar Level(fbs) Analysis

```
> total_non_target <- heart[heart$target==0,]$fbs
> total_non_target_high_fbs <- heart[heart$target==0&heart$fbs==1,]$fbs
> total_non_target_norm_fbs <- heart[heart$target==0&heart$fbs==0,]$fbs
> length(total_non_target_high_fbs) / length(total_non_target) * 100
[1] 15.94203
> length(total_non_target_norm_fbs) / length(total_non_target) * 100
[1] 84.05797
>
> total_target <- heart[heart$target==1,]$fbs
> total_target_high_fbs <- heart[heart$target==1&heart$fbs==1,]$fbs
> total_target_norm_fbs <- heart[heart$target==1&heart$fbs==0,]$fbs
> length(total_target_high_fbs) / length(total_target) * 100
[1] 13.93939
> length(total_target_norm_fbs) / length(total_target) * 100
[1] 86.06061
>
> cor(heart$fbs,heart$target)
[1] -0.02804576
>
> fbs <- table(heart$fbs)
> barplot(fbs,
+         main="Fasting Blood Sugar",
+         xlab="0 = No, 1 = Yes",
+         ylab="Count",
+         col=c("slateblue4","violetred1"),
+         legend=rownames(fbs))
>
> percentages <- round(fbs/sum(fbs)*100)
> labls <- paste(c("FBS<120","FBS>120")," ",percentages,"%",sep="")
> pie(fbs,labls,main="Fasting Blood Sugar",col=c("slateblue4","violetred1"))
>
```

- Fasting blood sugar is a categorical variable in which 0 means level is less than 120mg/dl and 1 means it is greater than 120mg/dl.
- In bar chart and pie chart it is clearly visible that maximum (85%) people have fasting blood sugar less than 120mg/dl.
- From all heart attacks having fasting blood sugar more than 120mg/dl accounts only 13.93%, and there is very low correlation between fasting blood sugar level and target (-0.28)

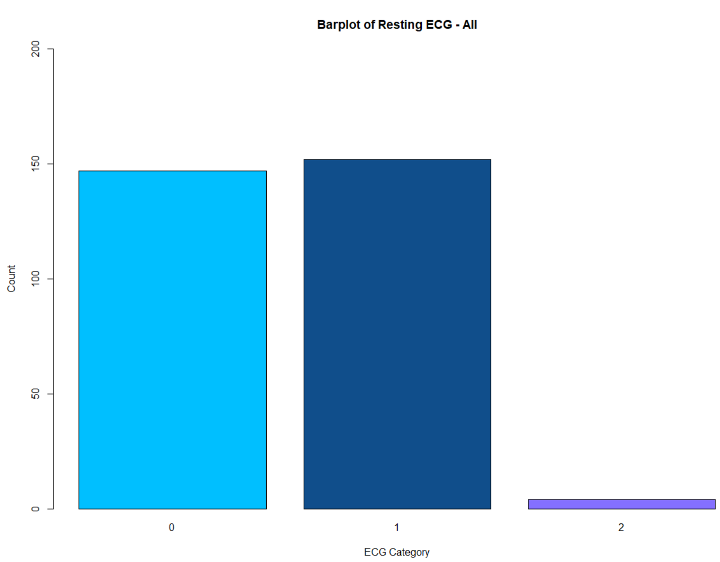


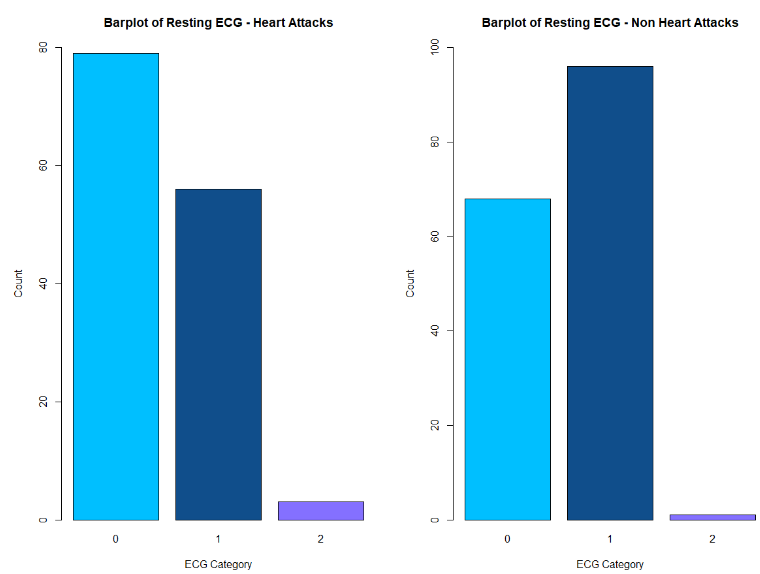


Resting Electrocardiographic Results(restecg) Analysis

```
> a <- heart[heart$target==0,]
> b <- heart[heart$target==1,]
>
> barplot(table(heart$restecg),
+         main="Barplot of Resting ECG - All",
+         xlab="ECG Category",
+         ylab="Count",
+         col=c("deepskyblue","dodgerblue4","lightslateblue"))
>
> par(mfrow=c(1,2))
> barplot(table(a$restecg),
+         main="Barplot of Resting ECG - Heart Attacks",
+         xlab="ECG Category",
+         ylab="Count",
+         col=c("deepskyblue","dodgerblue4","lightslateblue"))
>
> barplot(table(b$restecg),
+         main="Barplot of Resting ECG - Non Heart Attacks",
+         xlab="ECG Category",
+         ylab="Count",
+         col=c("deepskyblue","dodgerblue4","lightslateblue"))
```

- There are 3 categories in this variable 0,1 and 2.
- Category 2 of ECG is very less and category 1 or 2 are nearly same.
- However, when looking at heart attack records and non-heart attack records, having category 2 ECG there is a higher chance of a heart attack.

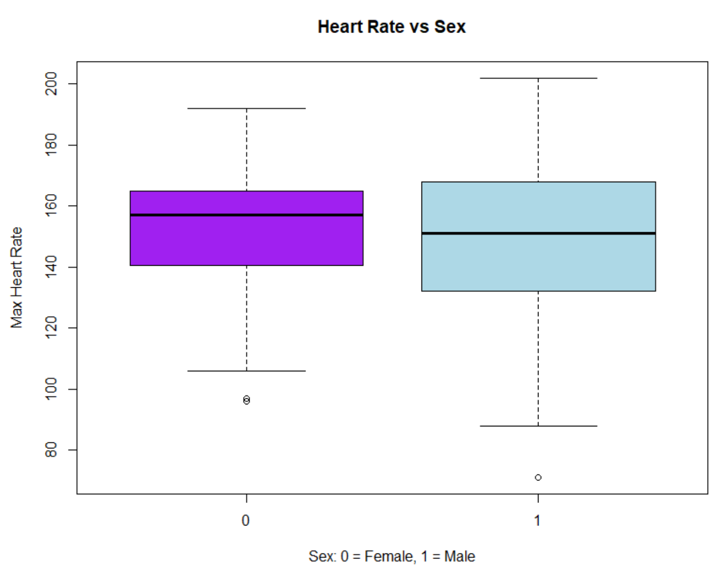


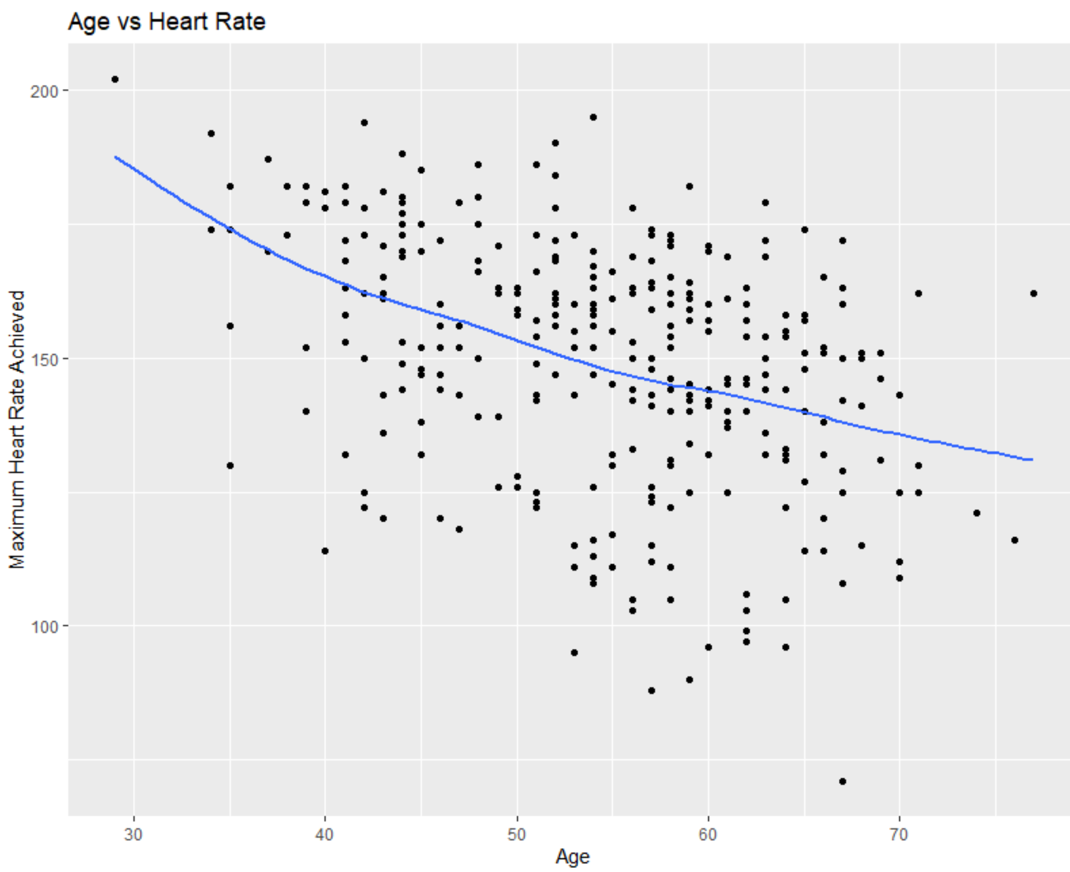


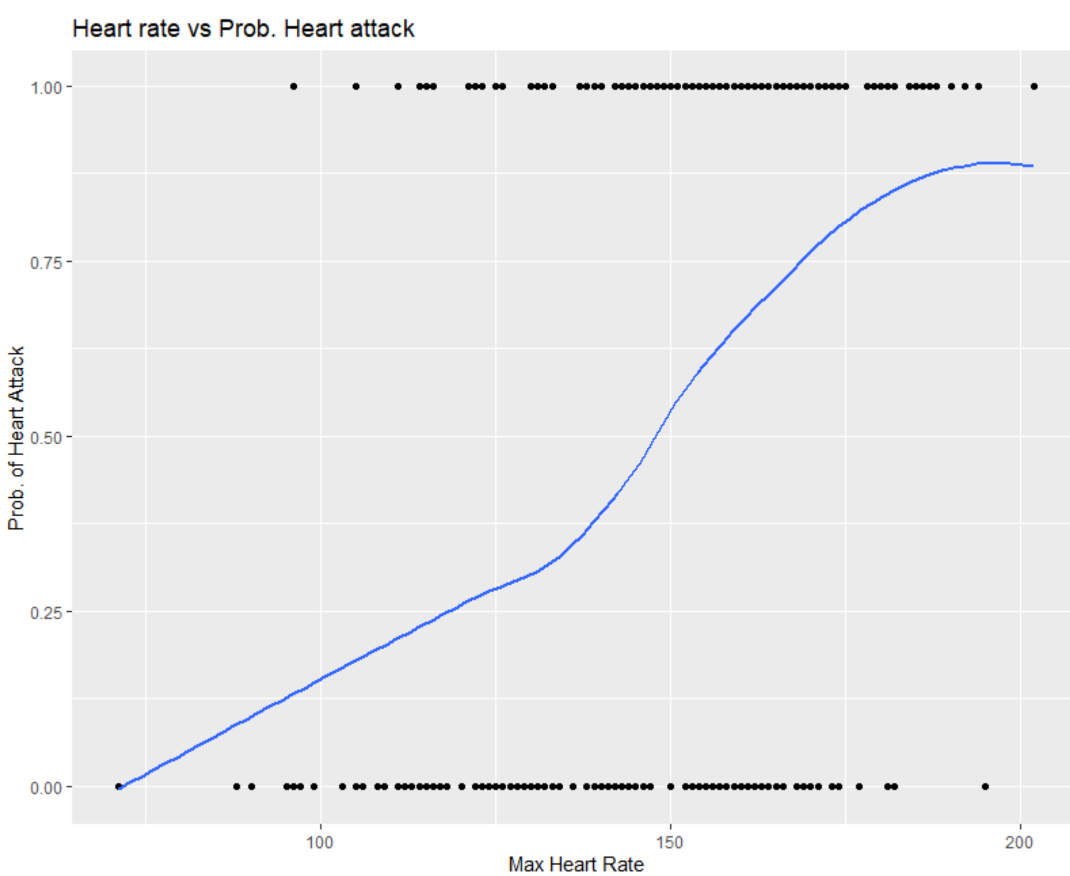
Maximum Heart Rate(thalach) Analysis

```
> class(heart$thalach)
[1] "integer"
> head(heart$thalach)
[1] 145 151 144 147 159 151
> summary(heart$thalach)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   71.0   133.5   153.0   149.6   166.0   202.0 
> cor(heart$age,heart$thalach)
[1] -0.3985219
> chisq.test(heart$age,heart$thalach)

	Pearson's Chi-squared test

data:  heart$age and heart$thalach
X-squared = 3945.4, df = 3600, p-value = 3.829e-05

Warning message:
In chisq.test(heart$age, heart$thalach) :
  Chi-squared approximation may be incorrect
> 
> boxplot(heart$thalach~as.factor(heart$sex),
+         main="Heart Rate vs Sex",
+         xlab="Sex: 0 = Female, 1 = Male",
+         ylab="Max Heart Rate",
+         col=c("purple","lightblue"))
> 
> a ← ggplot(heart,aes(x=age,y=thalach))+geom_point()+geom_smooth(se=FALSE)
> b ← a+scale_x_continuous(name="Age")+scale_y_continuous(name="Maximum Heart Rate Achieved")
> b + ggtitle("Age vs Heart Rate")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
> 
> a ← ggplot(heart,aes(x=thalach,y=target))+geom_point()+geom_smooth(se=FALSE)
> b ← a+scale_x_continuous(name="Max Heart Rate")+scale_y_continuous(name="Prob. of Heart Attack")
> b + ggtitle("Heart rate vs Prob. Heart attack")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- Maximum heart rate achieved continuous data with a minimum of 71, maximum of 202 and average of 149.6.
- Females normally have a higher maximum heart Rate achieved than males, this can be seen clearly from the box plot.
- As a person get older the maximum heart Rate achieved is lower as shown by the "Age vs Heart Rate" curve, this is further backed by the negative correlation (-0.3985219) between age and maximum heart rate achieved.
- We can also observe that as the heart rate increase probability of getting Heart attack is increasing greatly as well.



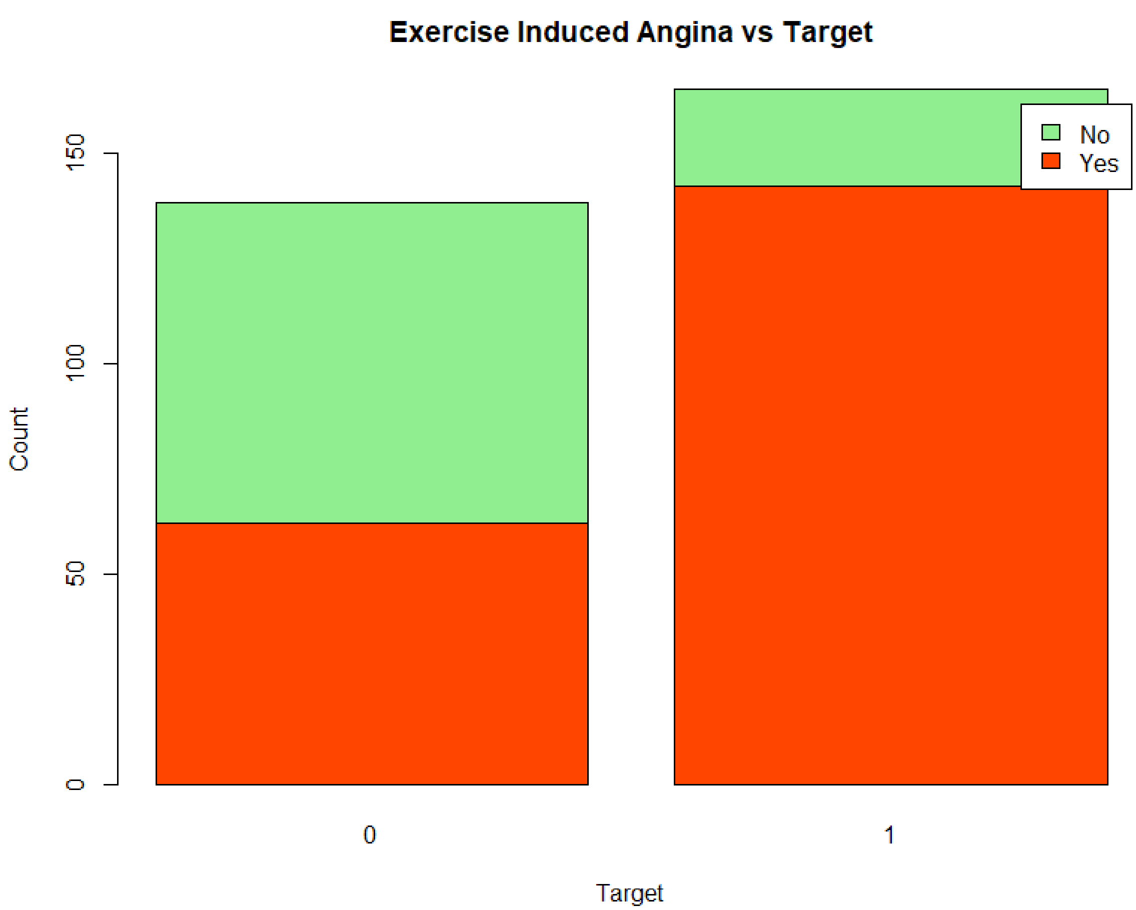

Age vs Heart Rate
200
150
100
Maximum Heart Rate Achieved
30
40
50
60
70
Age

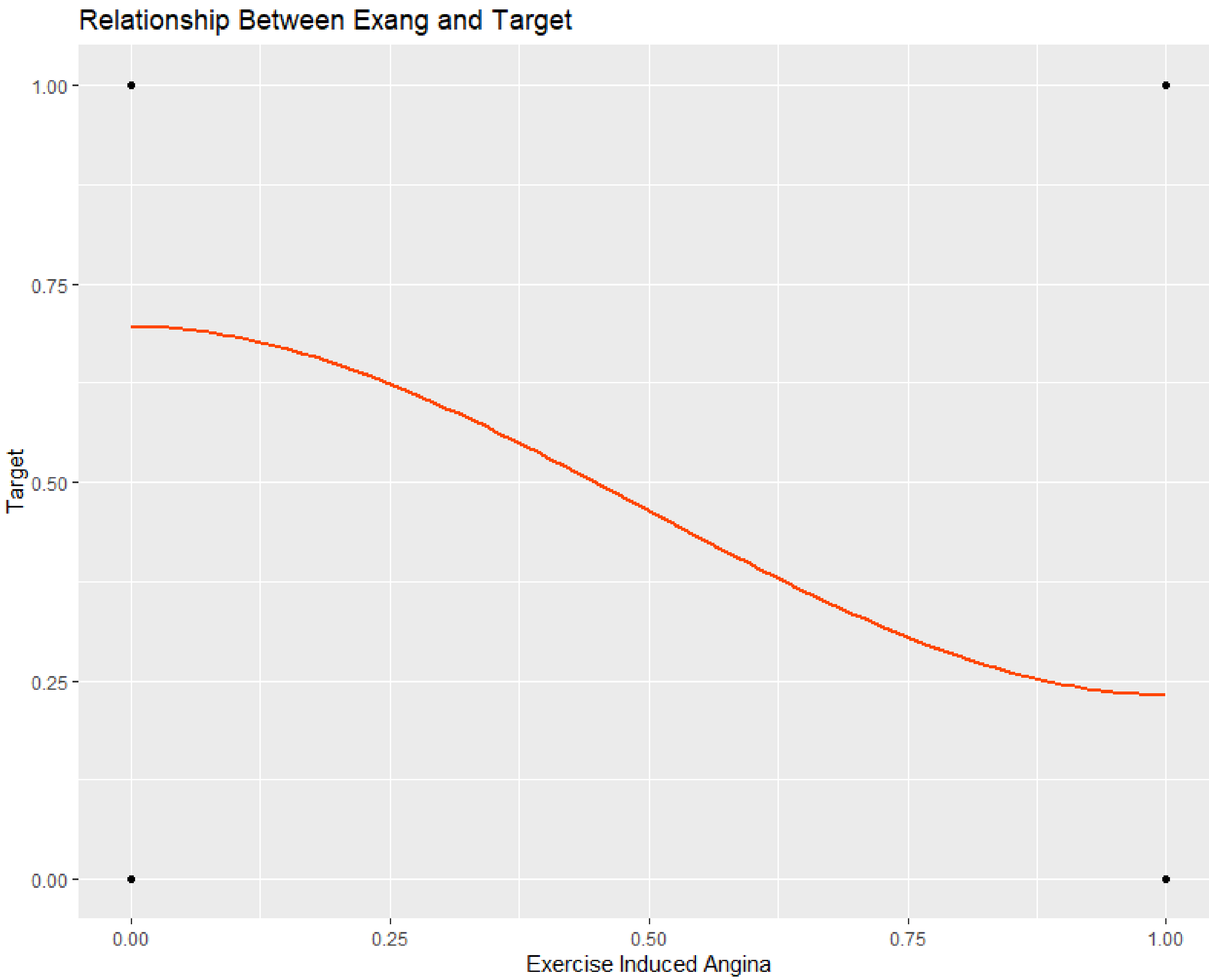

Heart rate vs Prob. Heart attack
1.00
0.75
0.50
0.25
0.00
Prob. of Heart Attack
100
150
200
Max Heart Rate

Exercise Induced Angina(exang) Analysis

```
> class(heart$exang)
[1] "integer"
> head(heart$exang)
[1] 1 0 1 0 0 1
> barplot(table(heart$exang,heart$target),
+         legend=c("Yes", "No"),
+         col=c("orangered1","palegreen2"),
+         main="Exercise Induced Angina vs Target",
+         xlab="Target",
+         ylab="Count")
>
> a <- ggplot(heart,aes(x=exang,y=target))+geom_point()+geom_smooth(color="orangered1",se=FALSE)
> b <- a + scale_x_continuous(name="Exercise Induced Angina")+scale_y_continuous(name="Target")
> b + ggtitle("Relationship Between Exang and Target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- Exercise Induced Angina is categorical variable with 0 = No exercise induced angina and 1 = Exercise induced angina
- We can clearly see in stacked bar plot that people with exercise induced angina is more likely to get heart attack.
- From relationship between exang and target curve with increase in exercise induced angina, chance of Heart attack is increasing.

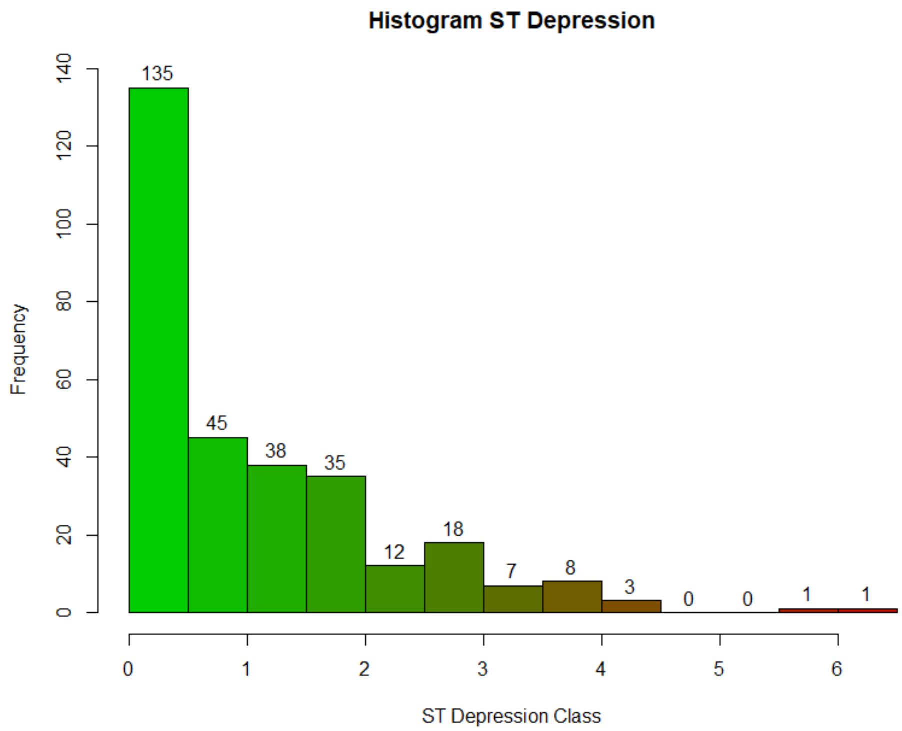


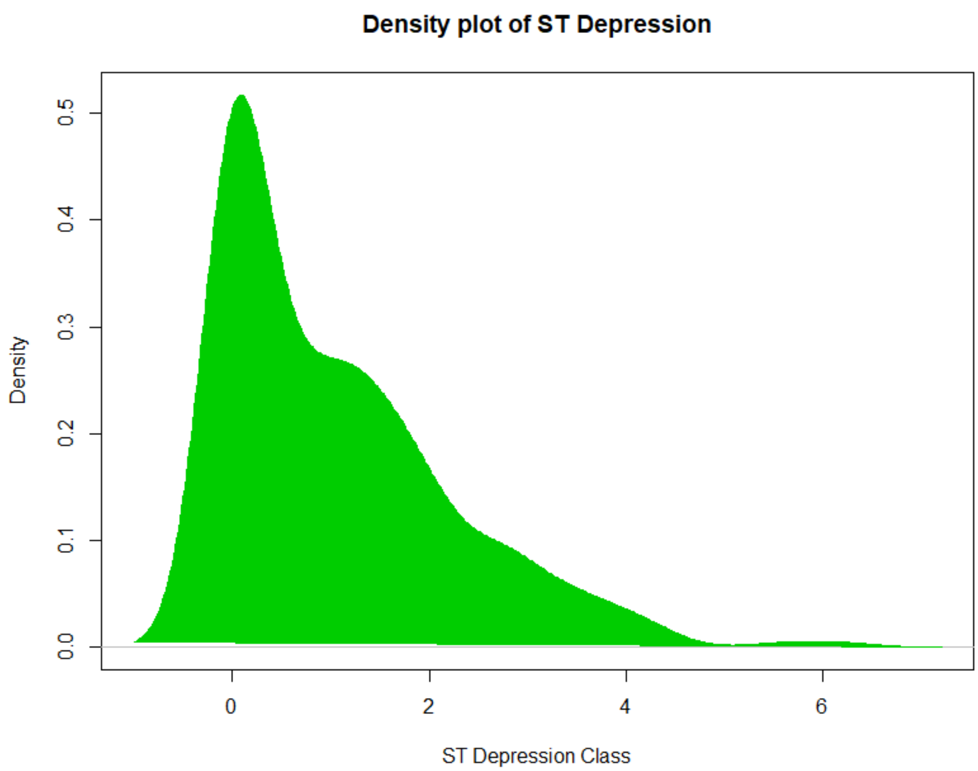

Relationship Between Exang and Target
1.00
0.75
0.50
0.25
0.00
Target
0.00
0.25
0.50
0.75
1.00
Exercise Induced Angina

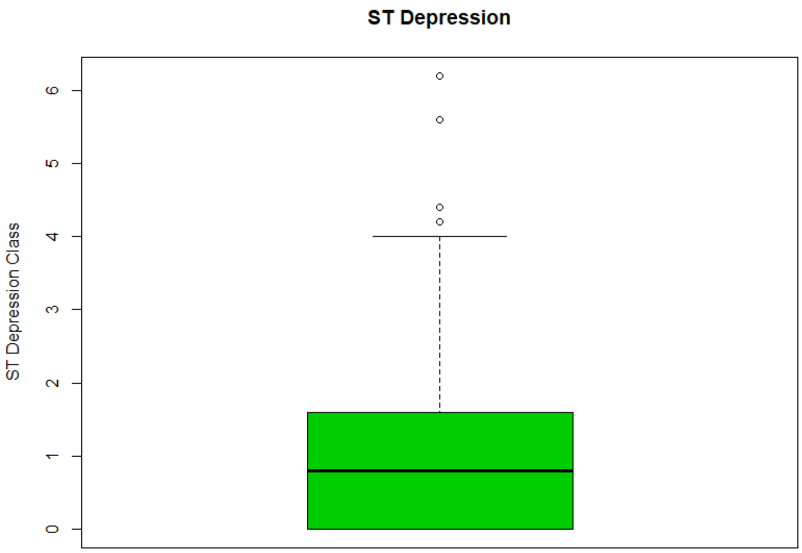
ST Depression Induced by Exercise Relative to Rest(oldpeak) Analysis

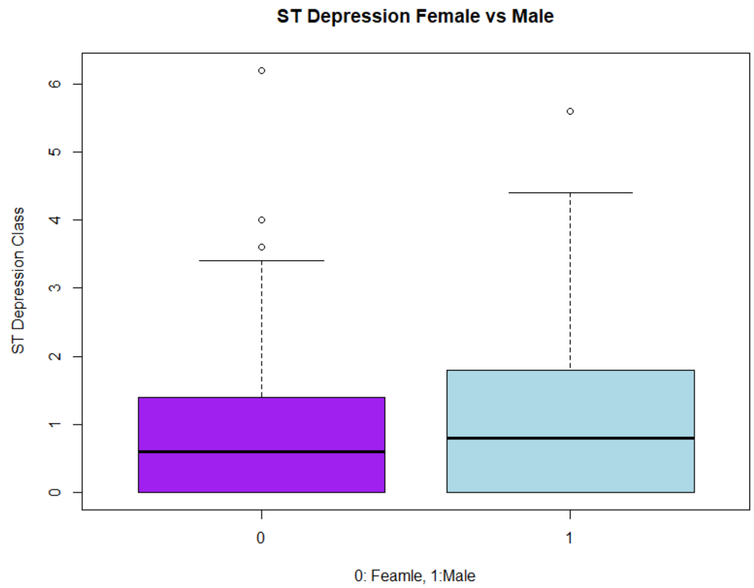
```
> class(heart$oldpeak)
[1] "numeric"
> head(heart$oldpeak,20)
 [1] 2.1 1.4 0.0 0.0 0.8 0.9 0.8 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.5 0.0 0.0 1.0
> summary(heart$oldpeak)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.80    1.04    1.60    6.20
> range(heart$oldpeak)
[1] 0.0 6.2
> sd(heart$oldpeak)
[1] 1.161075
> var(heart$oldpeak)
[1] 1.348095
>
> colfunc <- colorRampPalette(c("green3", "red3"))
> hist(heart$oldpeak,
+      main="Histogram ST Depression",
+      xlab="ST Depression Class",
+      ylab="Frequency",
+      col=colfunc(14),
+      labels=TRUE)
>
> plot(density(heart$oldpeak),
+      main="Density plot of ST Depression",
+      xlab="ST Depression Class",
+      ylab="Density")
> polygon(density(heart$oldpeak),col="green3",border="green3")
>
> boxplot(heart$oldpeak,
+         main="ST Depression",
+         ylab="ST Depression Class",
+         col="green3")
>
> boxplot(heart$oldpeak~heart$sex,
+         main="ST Depression Female vs Male",
+         col=c("purple","lightblue"),
+         xlab="0: Feamle, 1:Male",
+         ylab="ST Depression Class")
>
> a <- ggplot(heart,aes(x=oldpeak,y=target))+geom_point()+geom_smooth(color="green3",se=FALSE)
> b <- a+scale_x_continuous(name="ST Depression Class")+scale_y_continuous(name="Prob. of Heart Attack",limit=c
(0,1))
> b + ggtitle("Relation between oldpeak and heart attack")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

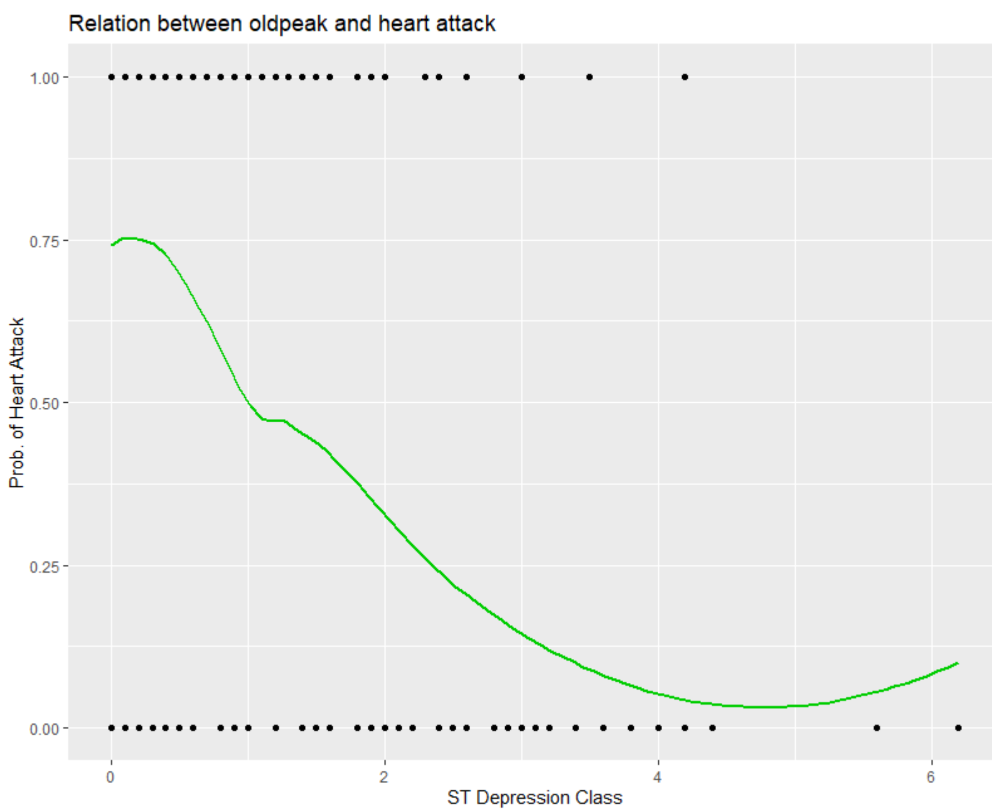
- ST depression induced by exercise relative to rest is numeric variable with a minimum of 0 and a maximum 6.20.
- Majority of the people have a ST depression of 0 to 0.5. This can be observed by the "Histogram ST depression".
- This is not normally distributed; data is skewed to the right as shown by the density plot.
- By "ST depression Female vs Male" we can observe that its higher is males than in females.
- In general, on increasing ST depression induced by exercise relative to rest, the probability of heart attack is decreasing as show by the "Relation between oldpeak and heart attack" curve.


Histogram ST Depression
Frequency
ST Depression Class
0
20
40
60
80
100
120
140
135
45
38
35
12
18
7
8
3
0
0
1
1
0
1
2
3
4
5
6


Density plot of ST Depression
Density
ST Depression Class
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6


ST Depression
ST Depression Class
0
1
2
3
4
5
6

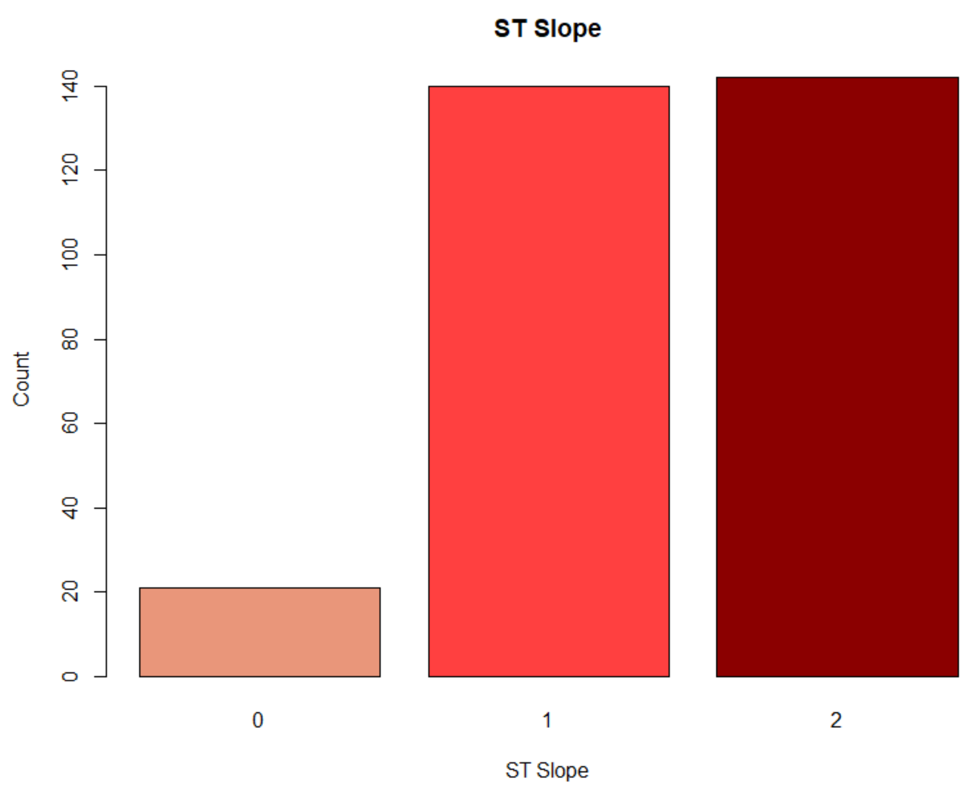

ST Depression Female vs Male
ST Depression Class
0
1
2
3
4
5
6
0
1
0: Feamle, 1:Male

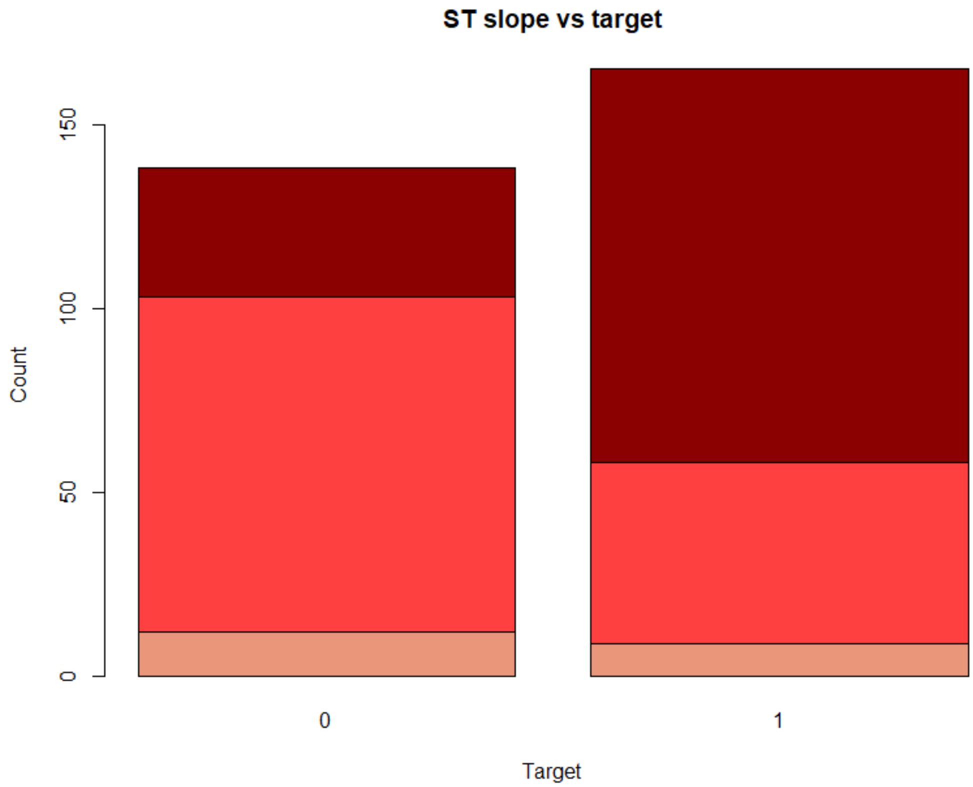

Relation between oldpeak and heart attack
Prob. of Heart Attack
1.00
0.75
0.50
0.25
0.00
0
2
4
6
ST Depression Class

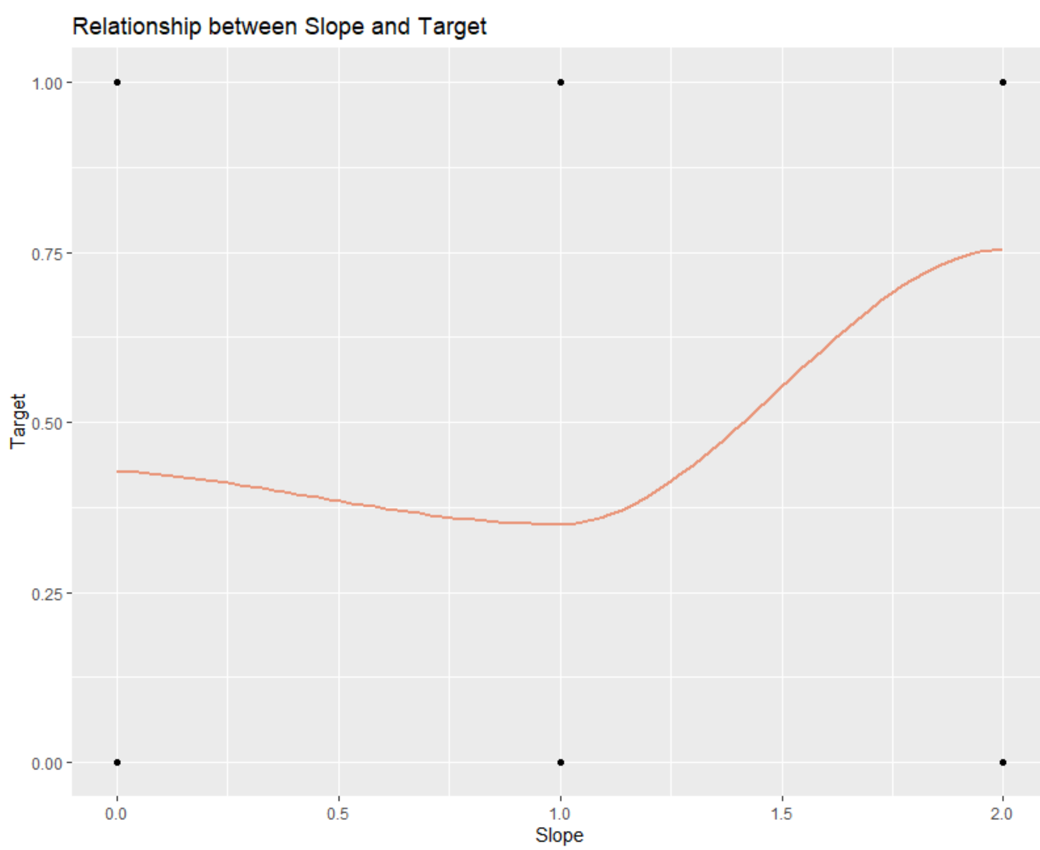
The Slope of the Peak Exercise ST Segment(slope) Analysis

```
> class(heart$slope)
[1] "integer"
> head(heart$slope)
[1] 2 1 0 2 0 1
> unique(heart$slope)
[1] 2 1 0
>
> barplot(table(heart$slope),
+         main="ST Slope",
+         xlab="ST Slope",
+         ylab="Count",
+         col=c("darksalmon","brown1","darkred"))
>
> barplot(table(heart$slope,heart$target),
+         main="ST slope vs target",
+         col=c("darksalmon","brown1","darkred"),
+         xlab="Target",
+         ylab="Count")
>
> a <- ggplot(heart,aes(x=slope,y=target))+geom_point()+geom_smooth(color="darksalmon",se=FALSE)
> b <- a + scale_x_continuous(name="Slope")+scale_y_continuous(name="Target")
> b + ggtitle("Relationship between Slope and Target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- The Slope of the Peak Exercise ST Segment categorical variable with 3 values 0,1 and 2.
- Majority of the population either has a ST slope of type 1 or 2, but type 2 has slightly higher count.
- By observing the ST slope vs target plot we can see that category 2 is more likely to get heart attack and 1 category is less likely to get heart attack.
- In curve "Relation between oldpeak and heart attack" after category 1 probability of heart attack is increasing.



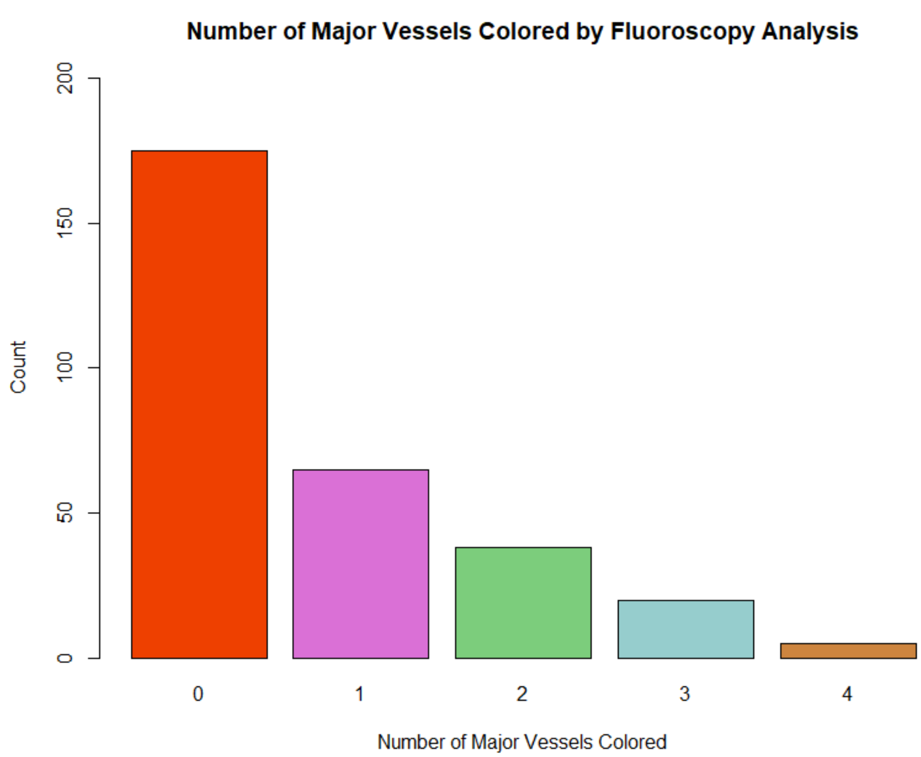

ST slope vs target
Count
0
50
100
150
0
1
Target

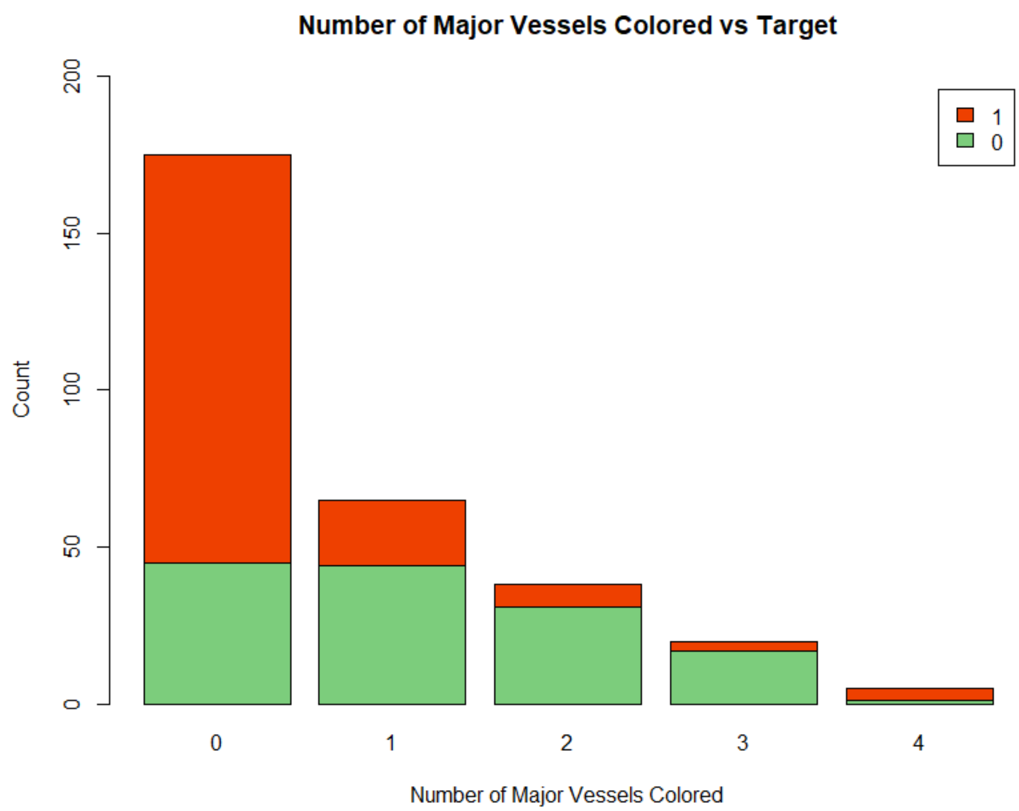

Relationship between Slope and Target
Target
0.00
0.25
0.50
0.75
1.00
0.0
0.5
1.0
1.5
2.0
Slope

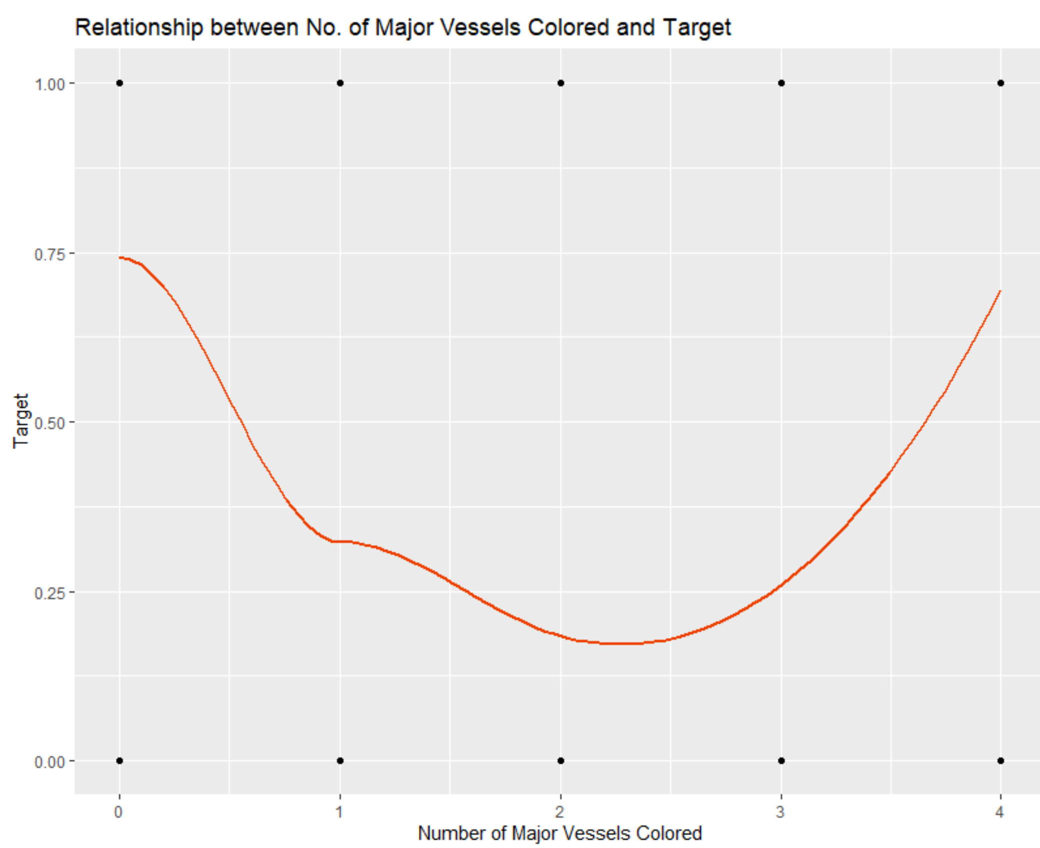
Number of Major Vessels Colored by Fluoroscopy(ca) Analysis

```
> class(heart$ca)
[1] "integer"
> unique(heart$ca)
[1] 0 2 1 3 4
>
> barplot(table(heart$ca),
+         main="Number of Major Vessels Colored by Fluoroscopy Analysis",
+         xlab="Number of Major Vessels Colored",
+         col=c("orangered2","orchid","palegreen3","paleturquoise3","peru"),
+         ylim=range(pretty(c(0, 170))), #to adjust y-axis scale
+         ylab="Count")
>
> barplot(table(heart$target,heart$ca),
+         main="Number of Major Vessels Colored vs Target",
+         xlab="Number of Major Vessels Colored",
+         col=c("palegreen3","orangered2"),
+         legend=c(0,1),
+         ylim=range(pretty(c(0, 170))), #to adjust y-axis scale
+         ylab="Count")
>
> a <- ggplot(heart,aes(x=ca,y=target))+geom_point()+geom_smooth(color="orangered2",se=FALSE)
> b <- a+scale_x_continuous(name="Number of Major Vessels Colored")+scale_y_continuous(name="Target")
> b + ggtitle("Relationship between No. of Major Vessels Colored and Target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- Number of Major Vessels Colored by Fluoroscopy is a categorical variable with 5 values of 0,1,2,3 and 4
- Majority of the population is in the 0th category
- Population in 0th category is most likely to get heart attack.
- The least risk population to get heart attack is in 2nd category.



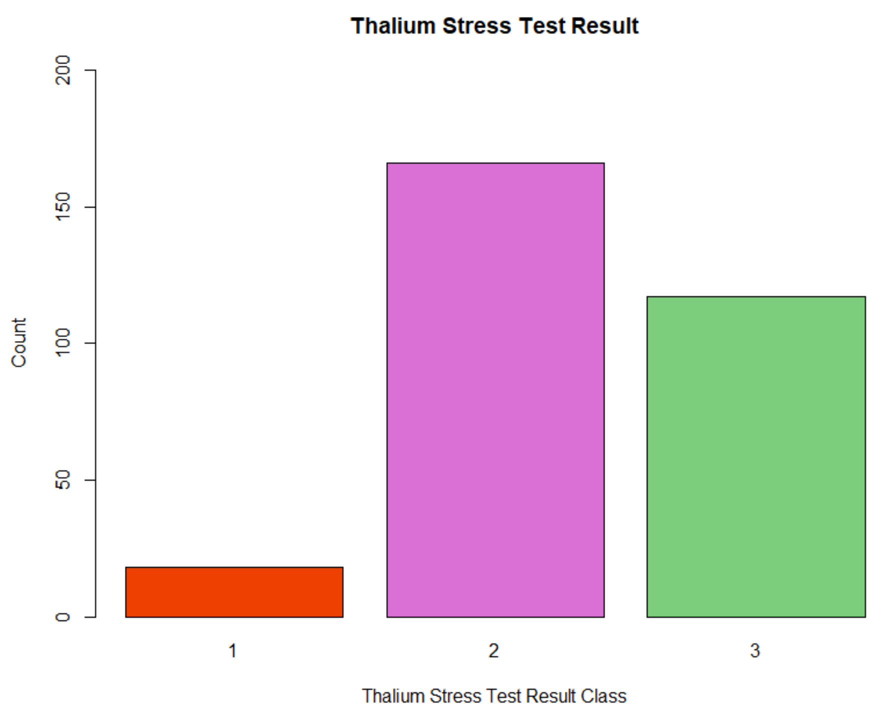

Number of Major Vessels Colored vs Target
Count
0
50
100
150
200
1
0
0
1
2
3
4
Number of Major Vessels Colored

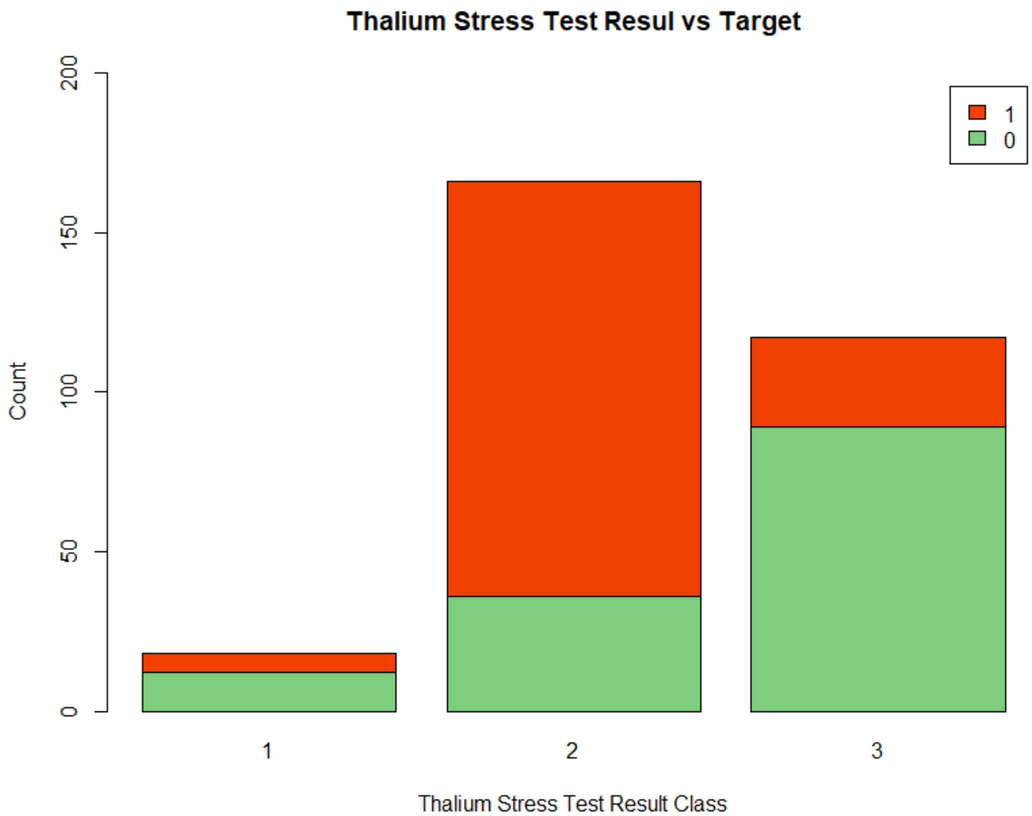

Relationship between No. of Major Vessels Colored and Target
Target
0.00
0.25
0.50
0.75
1.00
0
1
2
3
4
Number of Major Vessels Colored

Thalium Stress Test Result(thal) Analysis

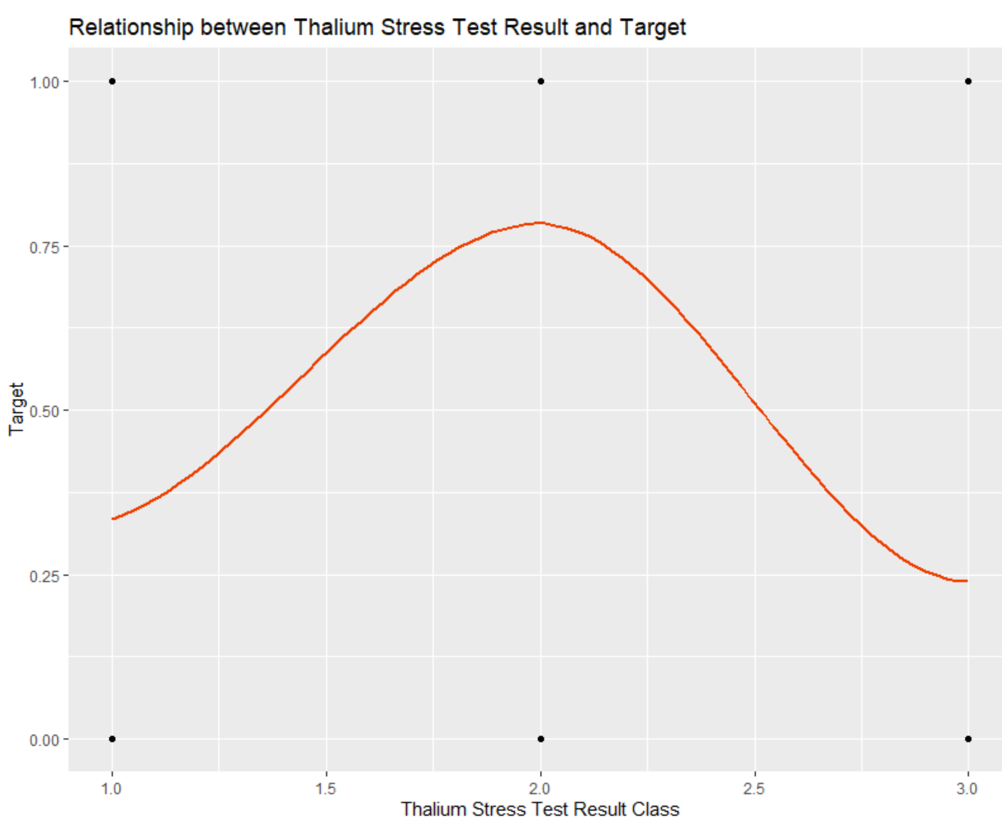
```
> class(heart$thal)
[1] "integer"
> unique(heart$thal)
[1] 2 3 1 0
> length(heart[heart$thal==0,]$thal)
[1] 2
>
> a <- heart[heart$thal!=0,]
> barplot(table(a$thal),
+         main="Thalium Stress Test Result",
+         xlab="Thalium Stress Test Result Class",
+         col=c("orangered2","orchid","palegreen3","paleturquoise3","peru"),
+         ylim=range(pretty(c(0, 170))), #to adjust y-axis scale
+         ylab="Count")
>
> barplot(table(a$target,a$thal),
+         main="Thalium Stress Test Resul vs Target",
+         xlab="Thalium Stress Test Result Class",
+         col=c("palegreen3","orangered2"),
+         legend=c(0,1),
+         ylim=range(pretty(c(0, 170))), #to adjust y-axis scale
+         ylab="Count")
>
> a <- ggplot(a,aes(x=thal,y=target))+geom_point()+geom_smooth(color="orangered2",se=FALSE)
> b <- a+scale_x_continuous(name="Thalium Stress Test Result Class")+scale_y_continuous(name="Target")
> b + ggtitle("Relationship between Thalium Stress Test Result and Target")
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

- This is a categorical variable with 3 values as 1,2 and 3.
- There are 2 data records with missing values for this and for the analysis it was removed.
- Most of the population has a normal (2) result for Thalium stress test, while most of the cases with a positive result (1 and 3) is reversable.
- Majority of the population with a heart attack has a normal result for the Thalium stress test.
- With regards to having a positive result (1 and 3) for the Thalium stress test, the probability of getting a heart is higher for fixed defect (1)





Thalium Stress Test Resul vs Target
Count
0
50
100
150
200
1
0
1
2
3
Thalium Stress Test Result Class

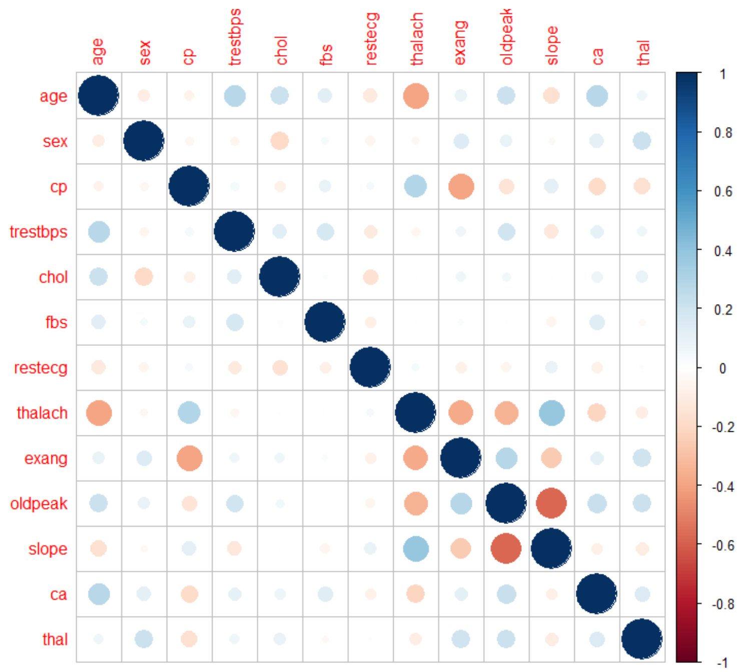


Relationship between Thalium Stress Test Result and Target
Target
0.00
0.25
0.50
0.75
1.00
1.0
1.5
2.0
2.5
3.0
Thalium Stress Test Result Class

Feature Selection

Correlation

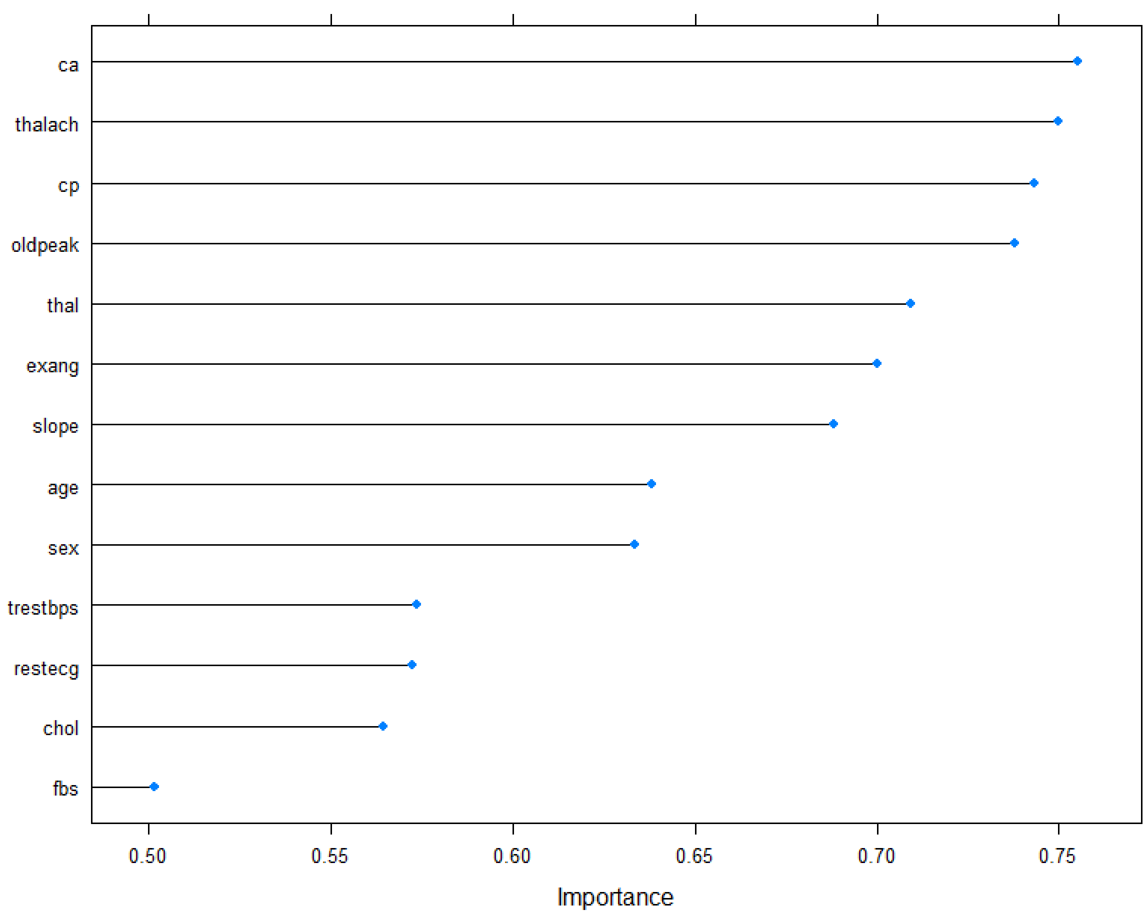
```
> heart<-read.csv("D:/Projects/heart.csv")
>
> # Correlation Plot
> correlation <- cor(heart[,1:13])
> corrplot(correlation)
> findCorrelation(correlation, cutoff=0.75)
integer(0)
>
```



Correlation was calculated for all attributes and highest correlation is between **slope** (*The slope of the peak exercise ST segment*) and **oldpeak** (*ST depression induced by exercise relative to rest*) which is -0.578. Since the highest absolute correlation value is less than ***0.75,*** no attributes are needed to be removed due to high correlation.

Features by importance

```
heart<-read.csv("D:/Projects/heart.csv")

# Data Pre processing: Removing corrupted rows
length(heart$target)
length(heart[heart$ca!=4&heart$thal!=0,]$target)
heart <- heart[heart$ca!=4&heart$thal!=0,]

# Data Pre processing: Convert columns to factors
str(heart)
heart$sex      <- as.factor(heart$sex)
heart$cp       <- as.factor(heart$cp)
heart$fbs      <- as.factor(heart$fbs)
heart$restecg <- as.factor(heart$restecg)
heart$exang    <- as.factor(heart$exang)
heart$slope    <- as.factor(heart$slope)
heart$thal     <- as.factor(heart$thal)
heart$ca       <- as.factor(heart$ca)
heart$target   <- as.factor(heart$target)
str(heart)

# Rank Features By Importance
set.seed(1024)
control <- trainControl(method="repeatedcv",number=10,repeats=3)
model <- train(target~.,data=heart,method="lvq",trControl=control)
importance <- varImp(model,scale=FALSE)
plot(importance)
```

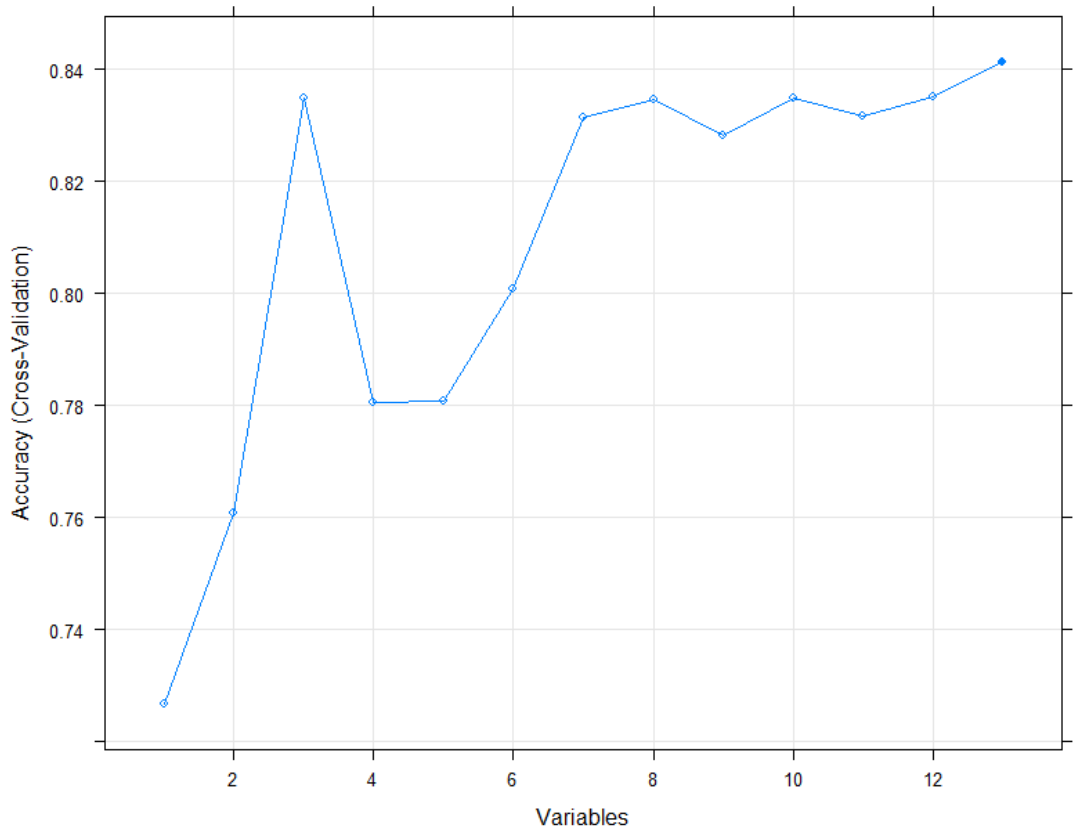


- All features except fasting blood sugar level has an importance greater than 50%.
- The top four features are
    - **ca**: Number of major vessels colored by fluoroscopy
    - **thalach**: Maximum heart rate achieved in beats per minute
    - **cp**: Chest pain type
    - **oldpeak**: ST depression induced by exercise relative to rest

We can observe that **fbs** (*Fasting blood sugar level*) has significantly less has less imports when compared to other attributes.

Automatic feature selection

```
> set.seed(1024)
> control <- trainControl(method="repeatedcv",number=10,repeats=3)
> model <- train(target~.,data=heart,method="lvq",trControl=control)
> importance <- varImp(model,scale=FALSE)
> plot(importance)
> set.seed(2048)
> control <- rfeControl(functions=rfFuncs,method="cv",number=10)
> results <- rfe(x=heart[,1:13],y=heart[,14],sizes=c(1:13),rfeControl=control)
> print(results)

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD KappaSD Selected
         1   0.7267 0.4441    0.04746 0.09847
         2   0.7609 0.5171    0.09940 0.20355
         3   0.8349 0.6654    0.07600 0.15518
         4   0.7805 0.5561    0.06946 0.13837
         5   0.7808 0.5569    0.06822 0.13689
         6   0.8007 0.5956    0.05355 0.10881
         7   0.8314 0.6586    0.04686 0.09468
         8   0.8347 0.6655    0.05051 0.10181
         9   0.8282 0.6526    0.06144 0.12288
        10   0.8348 0.6662    0.06180 0.12496
        11   0.8316 0.6593    0.06236 0.12565
        12   0.8351 0.6674    0.08192 0.16276
        13   0.8414 0.6788    0.06350 0.12838        *

The top 5 variables (out of 13):
   ca, thal, cp, oldpeak, sex

> predictors(results)
 [1] "ca"       "thal"     "cp"       "oldpeak"  "sex"      "thalach"  "exang"    "slope"    "age"
[10] "trestbps" "restecg"  "chol"     "fbs"
> plot(results, type=c("g", "o"))
```



Running RFS on the attributes shows that using all 13 attributes yields the highest accuracy of 84%.

By looking at all 3 methods of analysis I believe taking all 13 attributes into the model training will yield the best results.

2^{nd} **Task**: Formation of Training and Test Sets

```
# Load the Data
heart<-read.csv("D:/Projects/heart.csv")

# Data Pre processing: Removing corrupted rows
length(heart$target)
[1] 303
length(heart[heart$ca!=4&heart$thal!=0,]$target)
[1] 296
heart <- heart[heart$ca!=4&heart$thal!=0,]

# Data Pre processing: Convert columns to factors
str(heart)
'data.frame':	296 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
 $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
 $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
 $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ thal    : int  1 2 2 2 2 1 2 3 3 2 ...
 $ target  : int  1 1 1 1 1 1 1 1 1 1 ...

heart$sex     <- as.factor(heart$sex)
heart$cp      <- as.factor(heart$cp)
heart$fbs     <- as.factor(heart$fbs)
heart$restecg <- as.factor(heart$restecg)
heart$exang   <- as.factor(heart$exang)
heart$slope   <- as.factor(heart$slope)
heart$thal    <- as.factor(heart$thal)
heart$ca      <- as.factor(heart$ca)
heart$target  <- as.factor(heart$target)

levels(heart$sex)     <- c('F','M')
levels(heart$cp)      <- c('TA','ATA','NAP','AS')
levels(heart$fbs)     <- c('NO','YES')
levels(heart$restecg) <- c('NORM','ABNORM','VH')
levels(heart$exang)   <- c('NO','YES')
levels(heart$slope)   <- c('UP','FLT','DOWN')
levels(heart$thal)    <- c('FIX','NORM','REVDEF')
levels(heart$ca)      <- c('NONE','ONE','TWO','THREE')
```

```
levels(heart$target)  <- c('NO', 'YES')

str(heart)
'data.frame':     296 obs. of  14 variables:
 $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
 $ sex     : Factor w/ 2 levels "F","M": 2 2 1 2 1 2 1 2 2 2 ...
 $ cp      : Factor w/ 4 levels "TA","ATA","NAP",..: 4 3 2 2 1 1 2 2 3 3 ...
 $ trestbps: int  145 130 130 120 120 140 140 120 172 150 ...
 $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
 $ fbs     : Factor w/ 2 levels "NO","YES": 2 1 1 1 1 1 1 1 2 1 ...
 $ restecg : Factor w/ 3 levels "NORM","ABNORM",..: 1 2 1 2 2 2 1 2 2 2 ...
 $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
 $ exang   : Factor w/ 2 levels "NO","YES": 1 1 1 1 2 1 1 1 1 1 ...
 $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
 $ slope   : Factor w/ 3 levels "UP","FLT","DOWN": 1 1 3 3 3 2 2 3 3 3 ...
 $ ca      : Factor w/ 4 levels "NONE","ONE","TWO",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ thal    : Factor w/ 3 levels "FIX","NORM","REVDEF": 1 2 2 2 2 1 2 3 3 2 ...
 $ target  : Factor w/ 2 levels "NO","YES": 2 2 2 2 2 2 2 2 2 2 ...

# Seting up training and testing datasets
set.seed(4096)
intrain <- createDataPartition(y=heart$target,p=0.75,list=FALSE)
training <- heart[intrain,]
testing <- heart[-intrain,]
dim(training)
[1] 222  14

dim(testing)
[1] 74 14

# Repeated CV for Bagging type classifier
set.seed(128)
bagging_control <-
trainControl(method="repeatedcv",number=10,repeats=3)

# Repeated CV for Stacking type classifier
set.seed(128)
stacking_control <-
trainControl(method="repeatedcv",number=10,repeats=3,savePredictions='
final',classProbs=TRUE)
```

Notes:

- For data pre-processing, categorical attributes were converted to factors with meaningful values and there were 7 rows with missing values, and they were removed. Total records after cleaning is 296.
- Training data set consists of 75% of total records, which is around 222 records
- Testing data set is the remining 25% of total records, which is around 74 records
- For both bagging and stacking train control was 10 folder cross validation repeated 3 times.

3rd Task: Build Train and Test a Bagging type Classifier

```
# Random Forest
set.seed(256)
rf <- train(target~., data=training, method="rf", metric="Accuracy",
trControl=bagging_control)

# Bagged CART
set.seed(256)
treebag <-
train(target~.,data=training,method="treebag",metric="Accuracy",trCont
rol=bagging_control)

# Summarize Results
bagging_results <- resamples(list(treebag=treebag, rf=rf))
summary(bagging_results)
```

```
Call:
summary.resamples(object = bagging_results)

Models: treebag, rf
Number of resamples: 30

Accuracy
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
treebag 0.6521739 0.7272727 0.7727273 0.7915679 0.8542490 0.9545455    0
rf      0.6818182 0.7840909 0.8636364 0.8364954 0.8636364 1.0000000    0

Kappa
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
treebag 0.2868217 0.4590164 0.5416348 0.5779081 0.7048956 0.9090909    0
rf      0.3304348 0.5658706 0.7226891 0.6685711 0.7272727 1.0000000    0
```

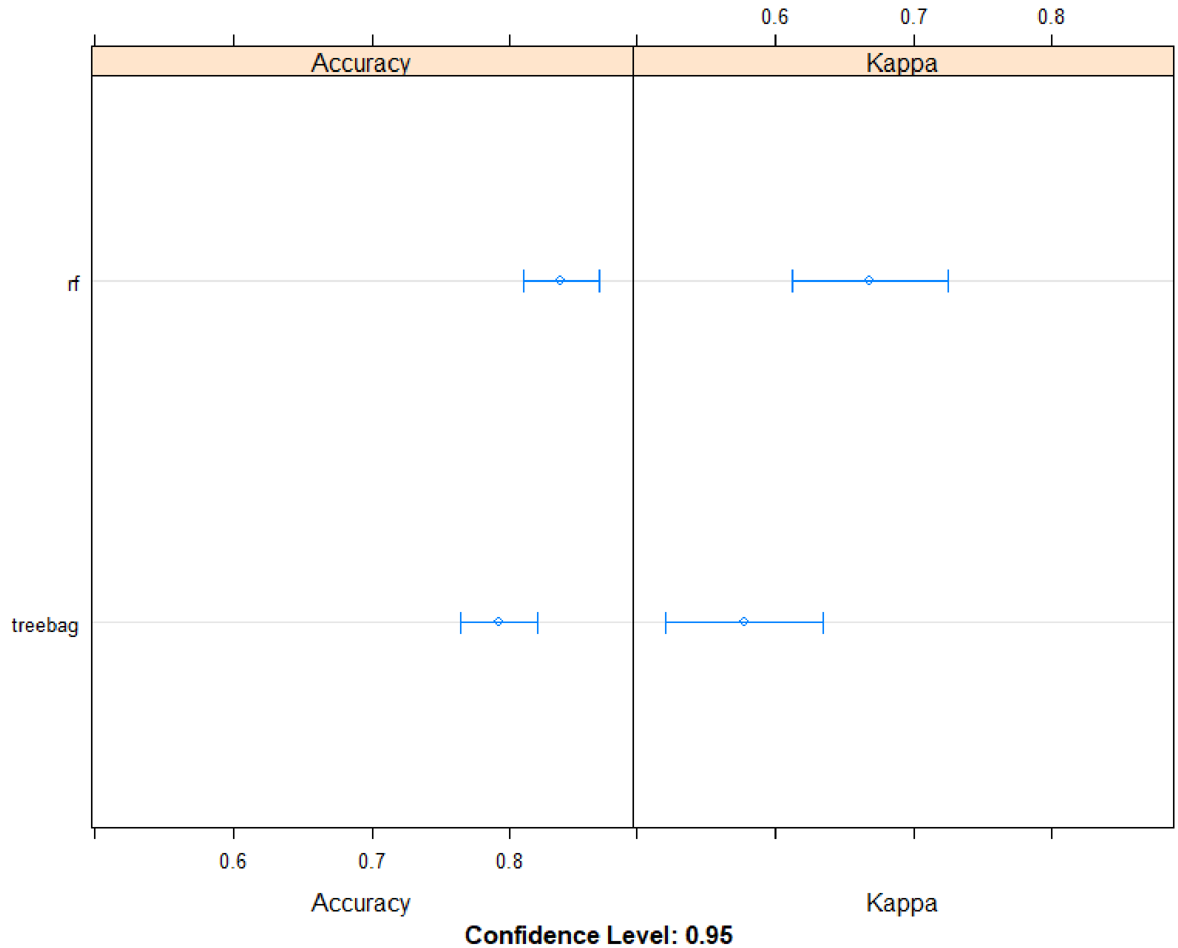
```
dotplot(bagging_results)

# Testing Random Forest
pred<-predict(rf,newdata=testing)
confusionMatrix(data=pred,testing$target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  24   9
       YES 10  31

               Accuracy : 0.7432
                 95% CI : (0.6284, 0.8378)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 0.0002685
```

```
                  Kappa : 0.4819

 Mcnemar's Test P-Value : 1.0000000

            Sensitivity : 0.7059
            Specificity : 0.7750
         Pos Pred Value : 0.7273
         Neg Pred Value : 0.7561
             Prevalence : 0.4595
         Detection Rate : 0.3243
   Detection Prevalence : 0.4459
      Balanced Accuracy : 0.7404

       'Positive' Class : NO
```

```
# Testing Bagged CART
pred<-predict(treebag,newdata=testing)
confusionMatrix(data=pred,testing$target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  21   7
       YES 13  33

               Accuracy : 0.7297
                 95% CI : (0.6139, 0.8265)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 0.0006555

                  Kappa : 0.4486

 Mcnemar's Test P-Value : 0.2635525

            Sensitivity : 0.6176
            Specificity : 0.8250
         Pos Pred Value : 0.7500
         Neg Pred Value : 0.7174
             Prevalence : 0.4595
         Detection Rate : 0.2838
   Detection Prevalence : 0.3784
      Balanced Accuracy : 0.7213

       'Positive' Class : NO
```



Plot Bagging Results

Random Forest Accuracy: 74.32%
Confusion Matrix for Random Forest

	YES	NO
YES	31	10
NO	9	24

Bagged CART Accuracy: 72.97%
Confusion Matrix for Bagged CART

	YES	NO
YES	33	13
NO	7	21

4th Task: Build Train and Test a Stacking type Classifier

```
# Stacking Algorithms
set.seed(512)
stacking_algorithms <- c('rpart', 'knn', 'nb')

# Training 'rpart', 'knn' and 'nb' models in parallel
models <- caretList(target~., data=training,
trControl=stacking_control, methodList=stacking_algorithms)
results <- resamples(models)
summary(results)
```

```
Call:
summary.resamples(object = results)

Models: rpart, knn, nb
Number of resamples: 30

Accuracy
           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
rpart 0.5652174 0.6931818 0.7727273 0.7611989 0.8181818 0.9565217    0
knn   0.3913043 0.6156126 0.6818182 0.6657444 0.7272727 0.8636364    0
nb    0.6818182 0.8003953 0.8636364 0.8528034 0.9090909 1.0000000    3

Kappa
            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
rpart  0.1221374 0.3882464 0.5416348 0.5153318 0.6393443 0.9132075    0
knn   -0.2196970 0.2308176 0.3474108 0.3224263 0.4406780 0.7317073    0
nb     0.3304348 0.5949281 0.7226891 0.7017306 0.8181694 1.0000000    3
```

```
dotplot(results)

# Testing Stacking CART
pred<-predict(models$rpart,newdata=testing)
confusionMatrix(data=pred,testing$target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  26  13
       YES  8  27

               Accuracy : 0.7162
                 95% CI : (0.5995, 0.815)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 0.001502

                  Kappa : 0.4349

 Mcnemar's Test P-Value : 0.382733
```

```
            Sensitivity : 0.7647
            Specificity : 0.6750
         Pos Pred Value : 0.6667
         Neg Pred Value : 0.7714
             Prevalence : 0.4595
         Detection Rate : 0.3514
   Detection Prevalence : 0.5270
      Balanced Accuracy : 0.7199

       'Positive' Class : NO
```

```
# Testing Naive Bayes
pred<-predict(models$nb,newdata=testing)
confusionMatrix(data=pred,testing$target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  28   9
       YES  6  31

               Accuracy : 0.7973
                 95% CI : (0.6878, 0.8819)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 3.778e-06

                  Kappa : 0.5946

 Mcnemar's Test P-Value : 0.6056

            Sensitivity : 0.8235
            Specificity : 0.7750
         Pos Pred Value : 0.7568
         Neg Pred Value : 0.8378
             Prevalence : 0.4595
         Detection Rate : 0.3784
   Detection Prevalence : 0.5000
      Balanced Accuracy : 0.7993

       'Positive' Class : NO
```

```
# Testing K-NN
pred<-predict(models$knn,newdata=testing)
confusionMatrix(data=pred,testing$target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  18  14
       YES 16  26
```

```
               Accuracy : 0.5946
                 95% CI : (0.4741, 0.7073)
    No Information Rate : 0.5405
    P-Value [Acc > NIR] : 0.2075

                  Kappa : 0.1802

 Mcnemar's Test P-Value : 0.8551

            Sensitivity : 0.5294
            Specificity : 0.6500
         Pos Pred Value : 0.5625
         Neg Pred Value : 0.6190
             Prevalence : 0.4595
         Detection Rate : 0.2432
   Detection Prevalence : 0.4324
      Balanced Accuracy : 0.5897

       'Positive' Class : NO
```

```
# Correlation Between Results
modelCor(results)
```

```
            rpart         knn          nb
rpart 1.000000000 0.007780966 0.000523101
knn   0.007780966 1.000000000 0.042365404
nb    0.000523101 0.042365404 1.000000000
```

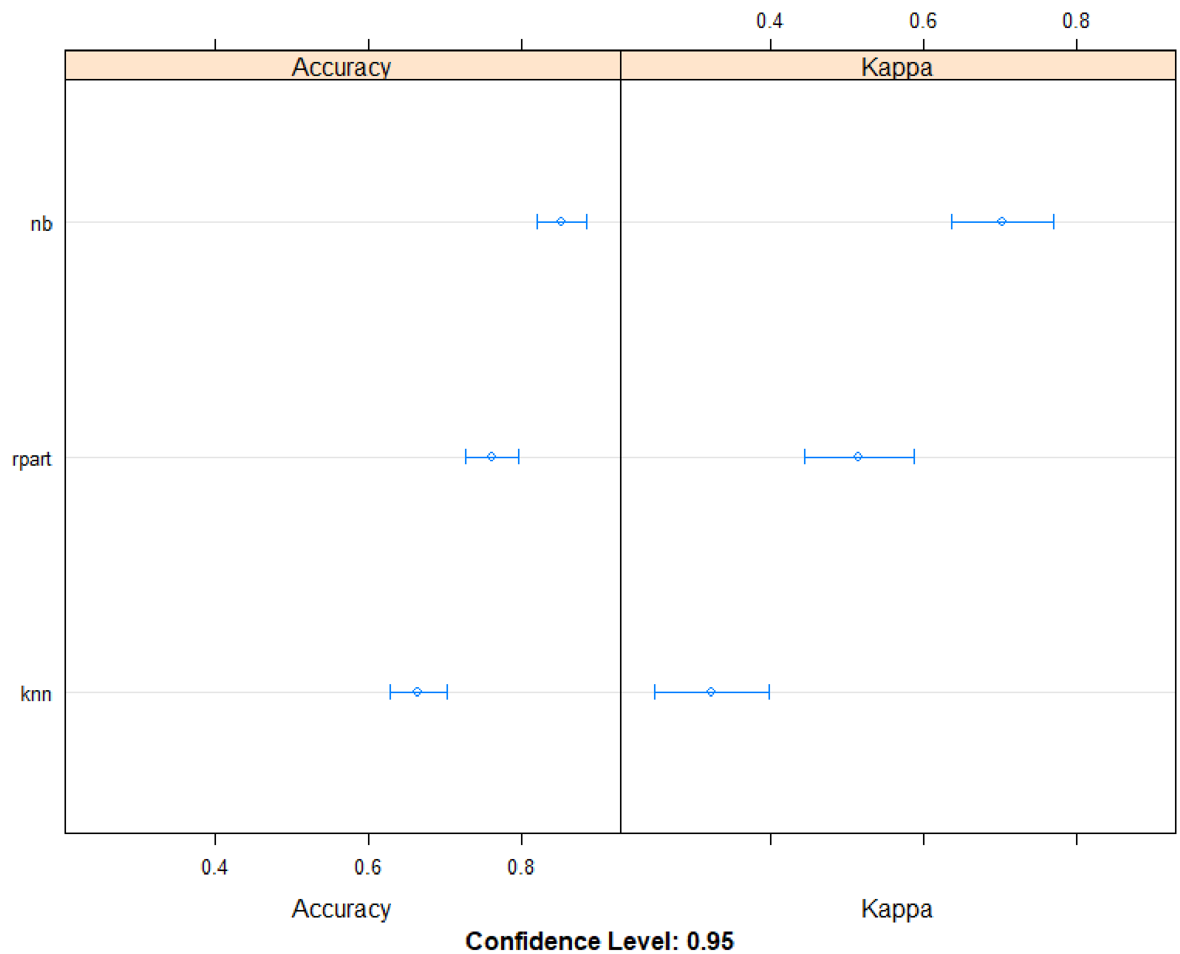
```
splom(results)

# Combining the predictions of the classifiers using a simple linear model
set.seed(512)
stack_glm <- caretStack(models, method="glm", metric="Accuracy", trControl=stacking_control)
print(stack_glm)
```

```
A glm ensemble of 3 base models: rpart, knn, nb

Ensemble results:
Generalized Linear Model

666 samples
  3 predictor
  2 classes: 'NO', 'YES'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 600, 599, 600, 599, 599, 599, ...
Resampling results:
```

```
  Accuracy   Kappa    
  0.8488072  0.6938677
```

```
# Testing linear model combined predictors 'rpart', 'knn' and 'nb
pred<-predict(stack_glm,newdata=testing)
confusionMatrix(data=pred,testing$target)
```

```
Confusion Matrix and Statistics

          Reference
Prediction NO YES
       NO  28   9
       YES  6  31
                                          
               Accuracy : 0.7973          
                 95% CI : (0.6878, 0.8819)
    No Information Rate : 0.5405          
    P-Value [Acc > NIR] : 3.778e-06       
                                          
                  Kappa : 0.5946          
                                          
 Mcnemar's Test P-Value : 0.6056          
                                          
            Sensitivity : 0.8235          
            Specificity : 0.7750          
         Pos Pred Value : 0.7568          
         Neg Pred Value : 0.8378          
             Prevalence : 0.4595          
         Detection Rate : 0.3784          
   Detection Prevalence : 0.5000          
      Balanced Accuracy : 0.7993          
                                          
       'Positive' Class : NO              
```



Plot Stacking Results

Stacking CART Accuracy: 71.62%
Confusion Matrix for Stacking CART

	YES	NO
YES	27	8
NO	13	26

Naive Bayes Accuracy: 79.73%
Confusion Matrix for Naive Bayes

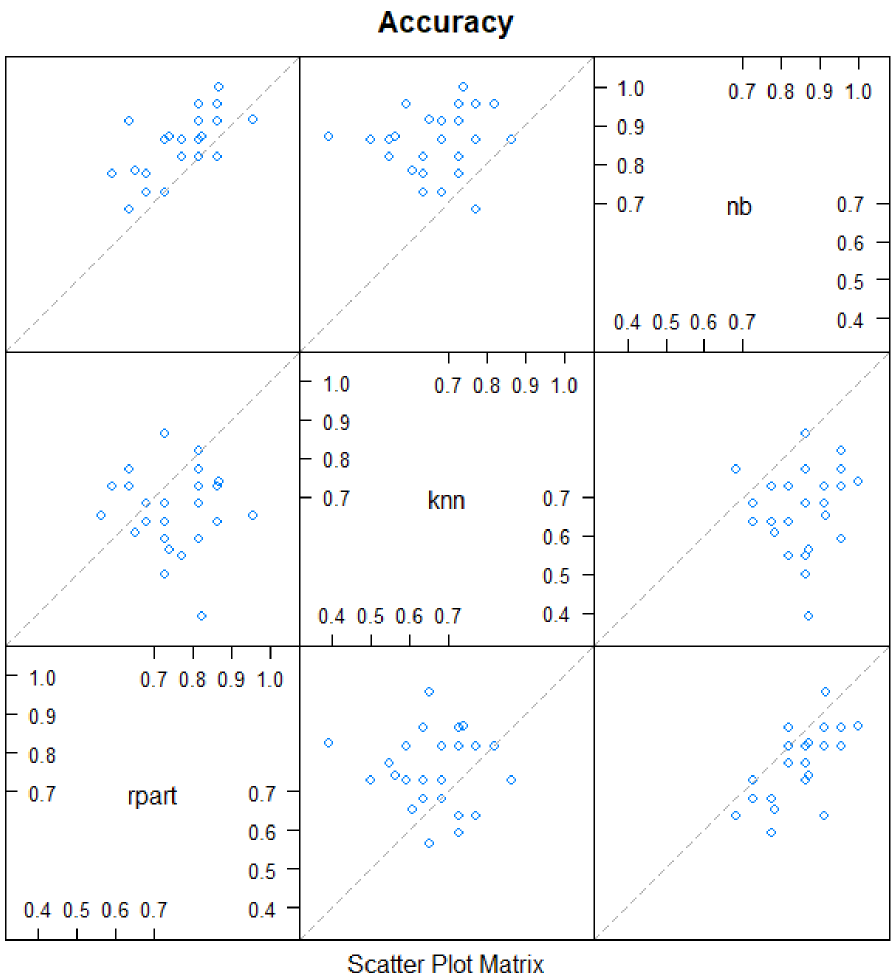
	YES	NO
YES	31	6
NO	9	28

K-NN Accuracy: 59.46%
Confusion Matrix for K-NN

	YES	NO
YES	26	16
NO	14	18

Correlation Between Results

	rpart	knn	nb
rpart	1	0.007780966	0.000523101
knn	0.007780966	1	0.042365404
nb	0.000523101	0.042365404	1



<u>For the 3 algorithms have < 0.7 correlation. Therefore, combining them will improve the prediction accuracy.</u>

Combined predictors model (Stacking CART, Naive Bayes and K-NN) Accuracy: 79.73%
Confusion Matrix combined predictors model:

	YES	NO
YES	31	6
NO	9	28

5th Task: Measure Performance

Confusion matrix estimation

Random Forest

	YES	NO
YES	31	10
NO	9	24

Bagged CART

	YES	NO
YES	33	13
NO	7	21

Stacking CART

	YES	NO
YES	27	8
NO	13	26

Naive Bayes

	YES	NO
YES	31	6
NO	9	28

K-NN

	YES	NO
YES	26	16
NO	14	18

Combined predictors model

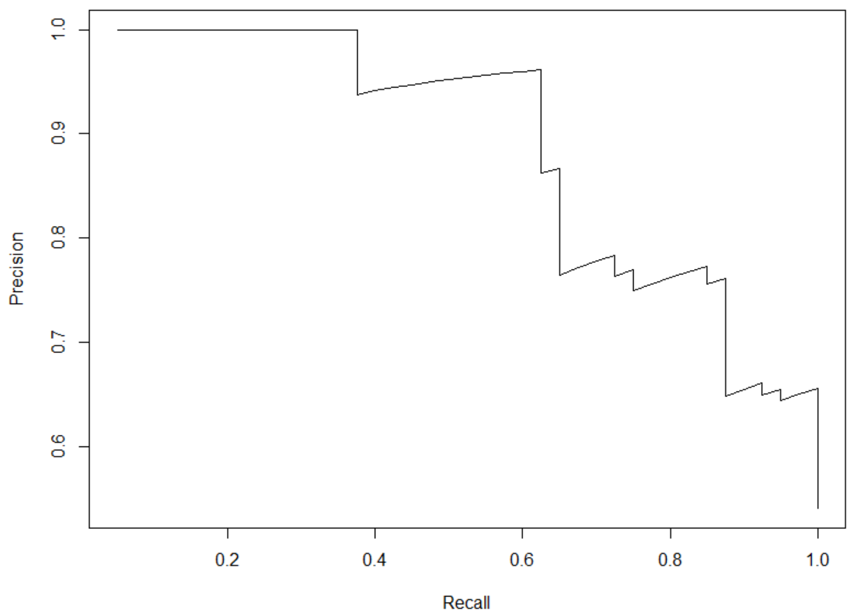
	YES	NO
YES	31	6
NO	9	28

```
### 5th Task: Measure Performance
# Re-run the predictions with type set to probability
rf_pred_prob        <- predict(rf,newdata=testing,type="prob")
treebag_pred_prob   <- predict(treebag,newdata=testing,type="prob")
rpart_pred_prob   <- predict(models$rpart,newdata=testing,type="prob")
nb_pred_prob        <- predict(models$nb,newdata=testing,type="prob")
knn_pred_prob       <- predict(models$knn,newdata=testing,type="prob")
stack_glm_pred_prob <- predict(stack_glm,newdata=testing,type="prob")

measure_performance <- function(pred_prob)
{
  perf <- prediction(pred_prob,testing$target)

  perf.prec_rec <- performance(perf,measure="prec",x.measure='rec')
  plot(perf.prec_rec)

  perf.acc <- performance(perf,measure="acc")
  plot(perf.acc)

  perf.roc = performance(perf,measure="tpr",x.measure="fpr")
  plot(perf.roc)

  perf.auc = performance(perf,measure="auc")
  perf.rauc <- perf.auc@y.values
  perf.rauc
}
```

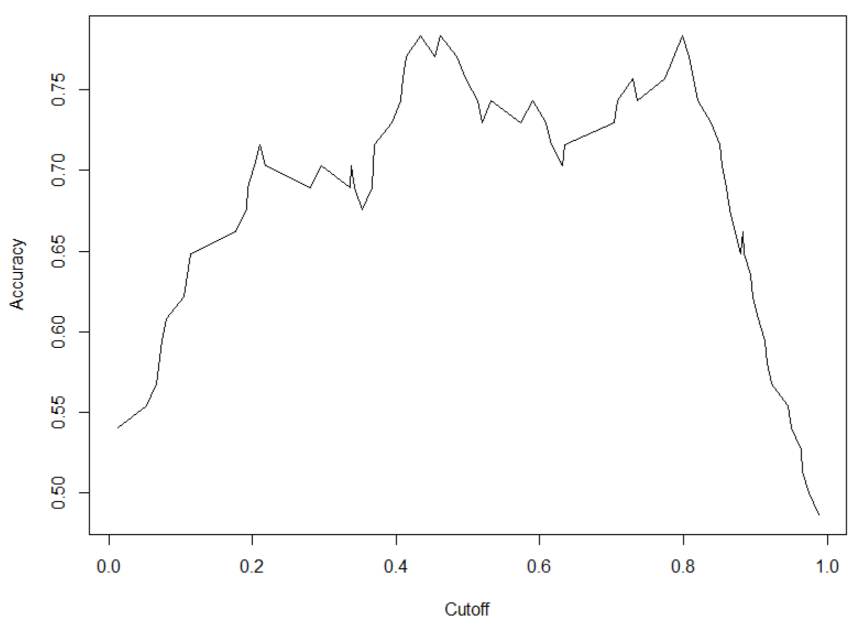
Note: ROCR requires estimated probabilities (or log odds) and the labels are binary values. So, I re-ran the predictions with type set to “prob”.
The `measure_performace` function takes in the p probabilities for the truth label and draws the required plots and prints the RAUC.

Random Forest

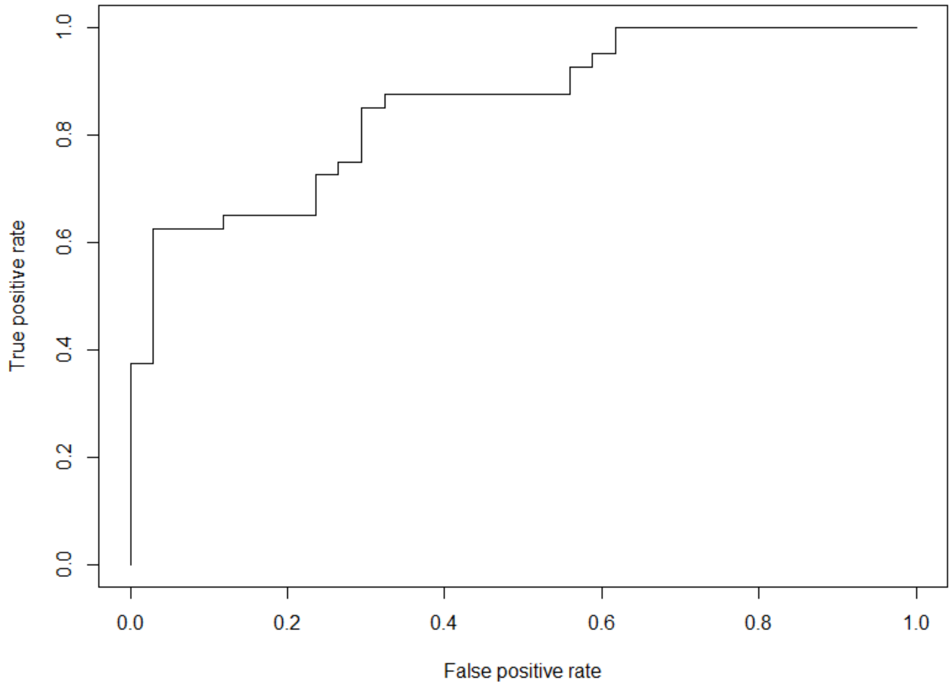
Precision Vs. Recall



Accuracy



Receiver Operating Characteristic Curve



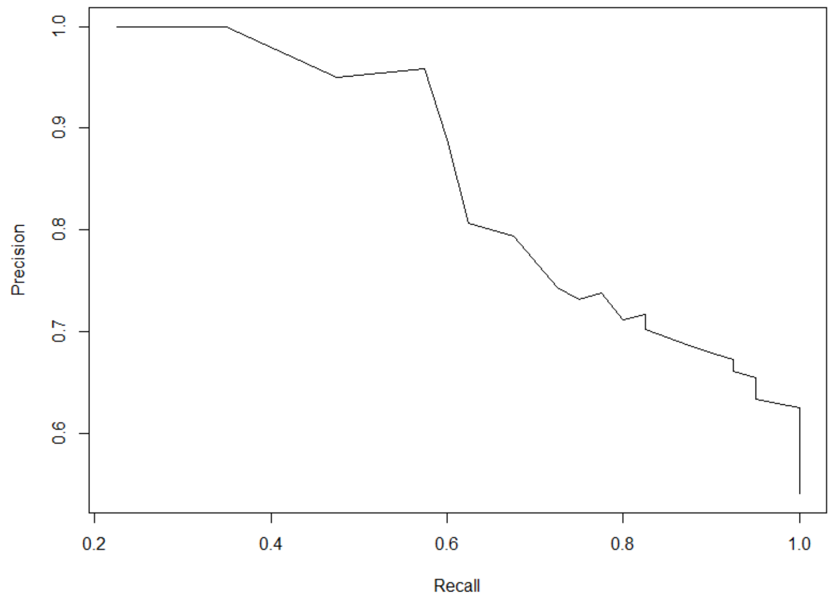
RAUC: 0.8544118

Training time: 9.6 seconds

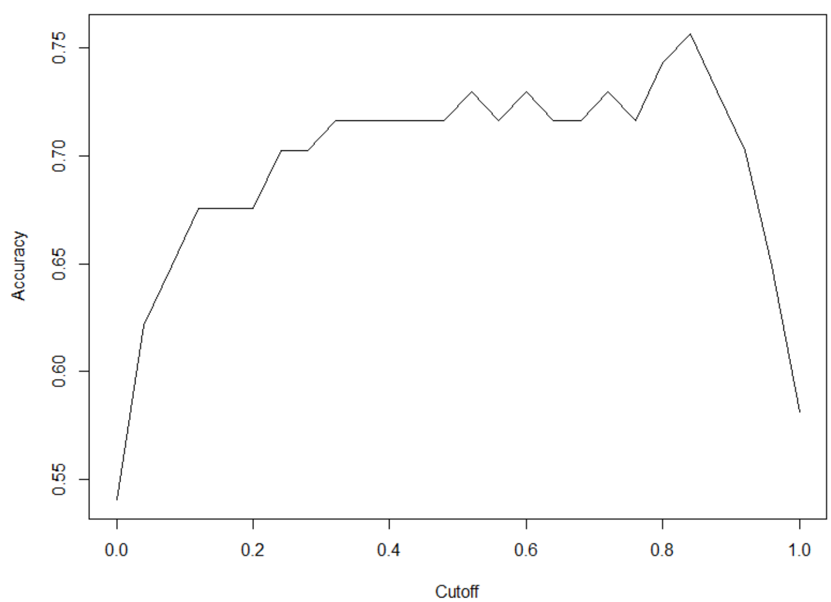
```
set.seed(256)
start_time <- Sys.time()
rf <- train(target~., data=training, method="rf", metric="Accuracy",
trControl=bagging_control)
end_time <- Sys.time()
end_time - start_time
 Time difference of 9.617178 secs
```

Bagged CART

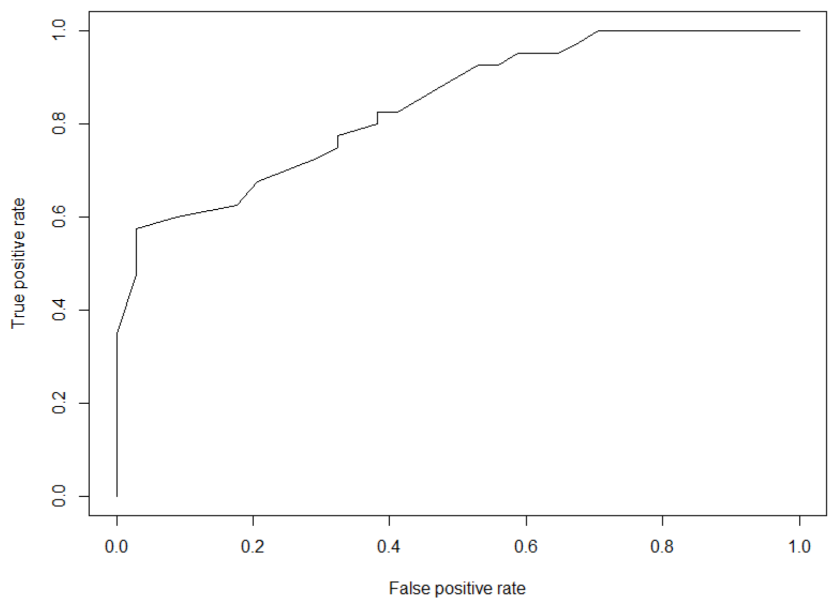
Precision Vs. Recall



Accuracy



Receiver Operating Characteristic Curve



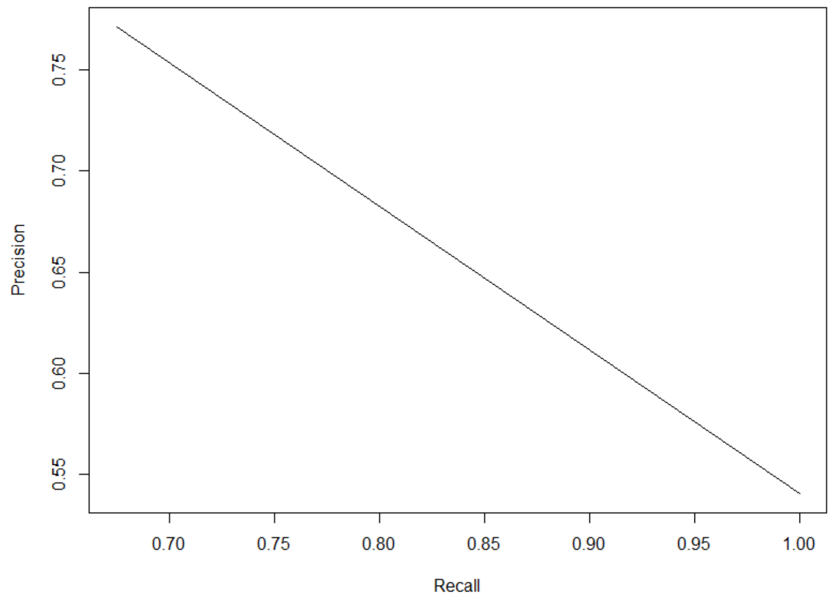
RAUC: 0.8389706

Training time: 3.9 seconds

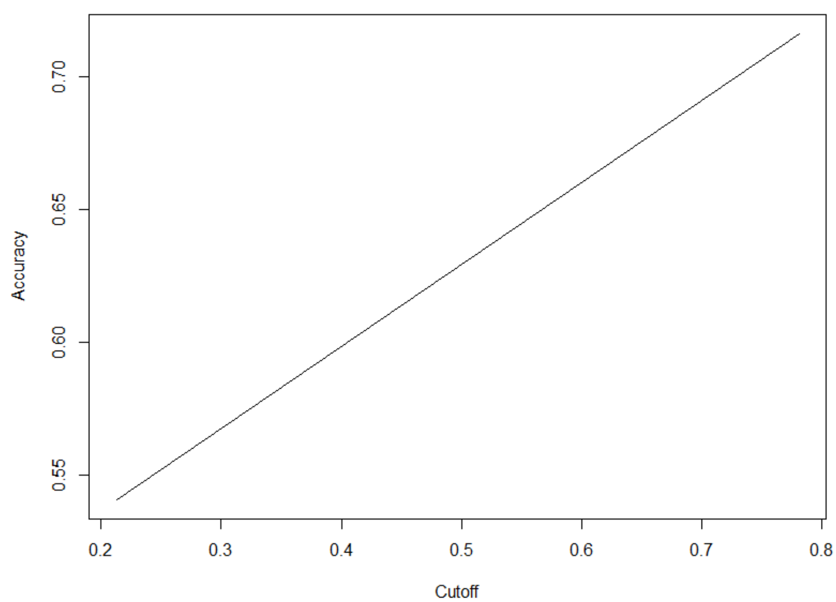
```
set.seed(256)
start_time <- Sys.time()
treebag <-
train(target~.,data=training,method="treebag",metric="Accuracy",trCont
rol=bagging_control)
end_time <- Sys.time()
end_time - start_time
Time difference of 3.869591 secs
```

<u>Stacking CART</u>

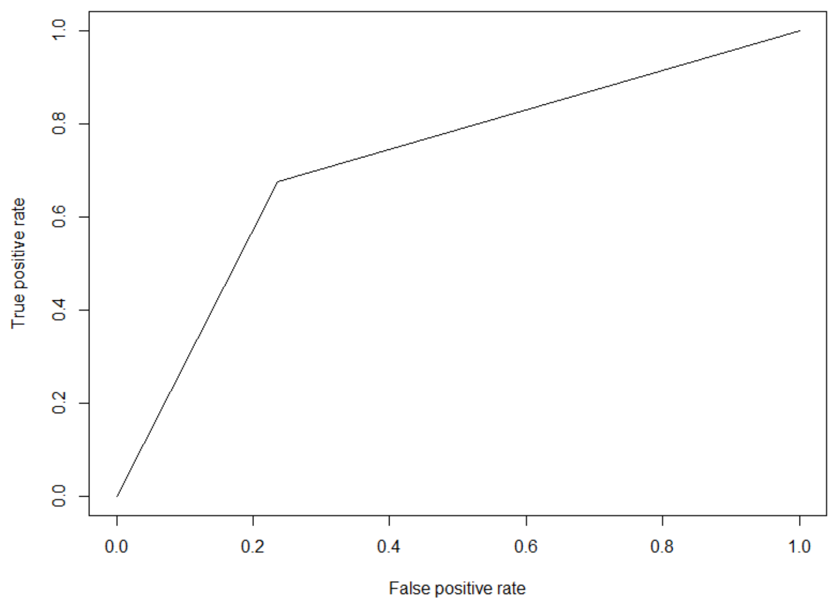
Precision Vs. Recall



Accuracy



Receiver Operating Characteristic Curve



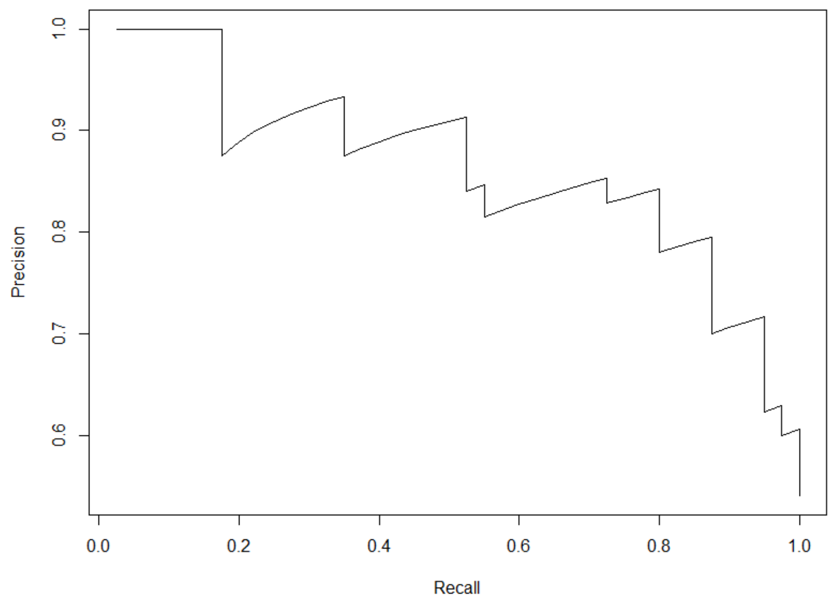
RAUC: 0.7198529

Training time: 5.3seconds (For all 3 algorithms - 'rpart', 'knn', 'nb')

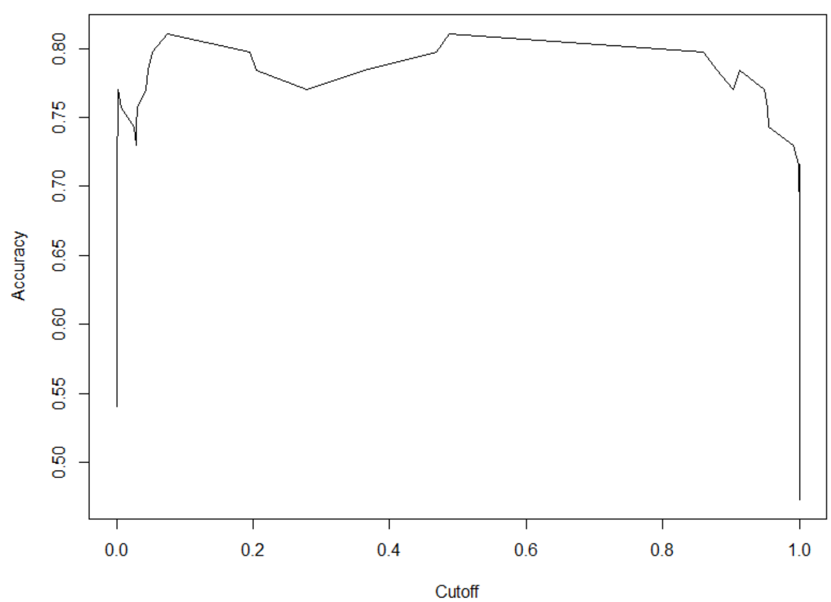
```
start_time <- Sys.time()
models <- caretList(target~., data=training,
trControl=stacking_control, methodList=stacking_algorithms)
end_time <- Sys.time()
end_time - start_time
Time difference of 5.342518 secs
```

<u>Naive Bayes</u>

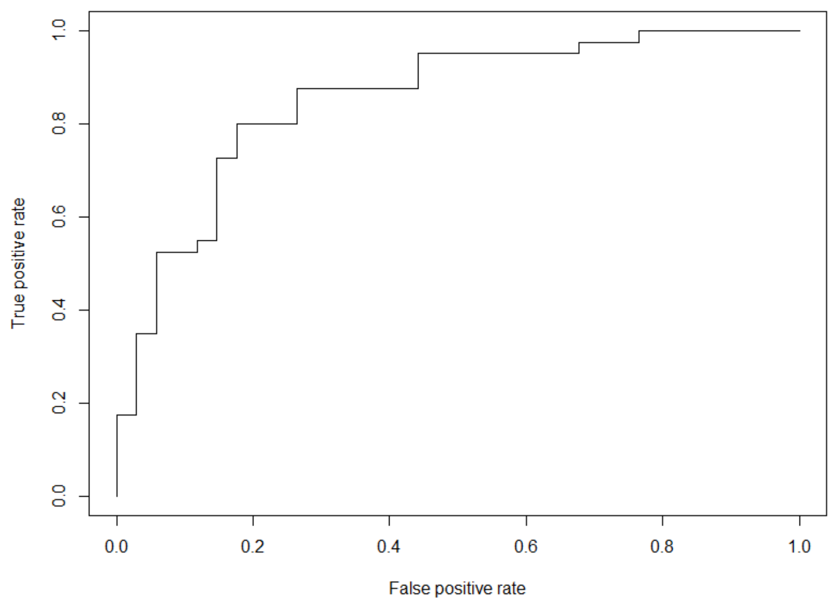
Precision Vs. Recall



Accuracy



Receiver Operating Characteristic Curve



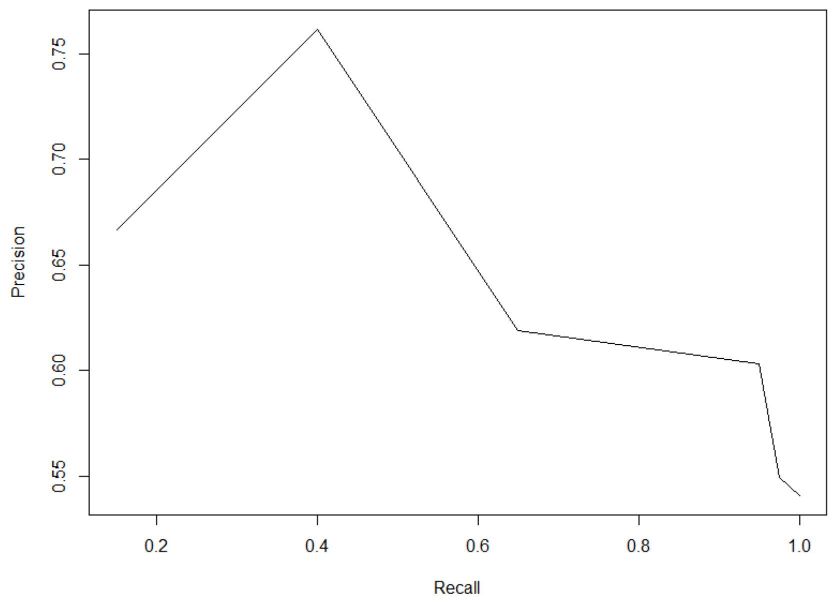
RAUC: 0.8536765

Training time: 5.3seconds (For all 3 algorithms - 'rpart', 'knn', 'nb')

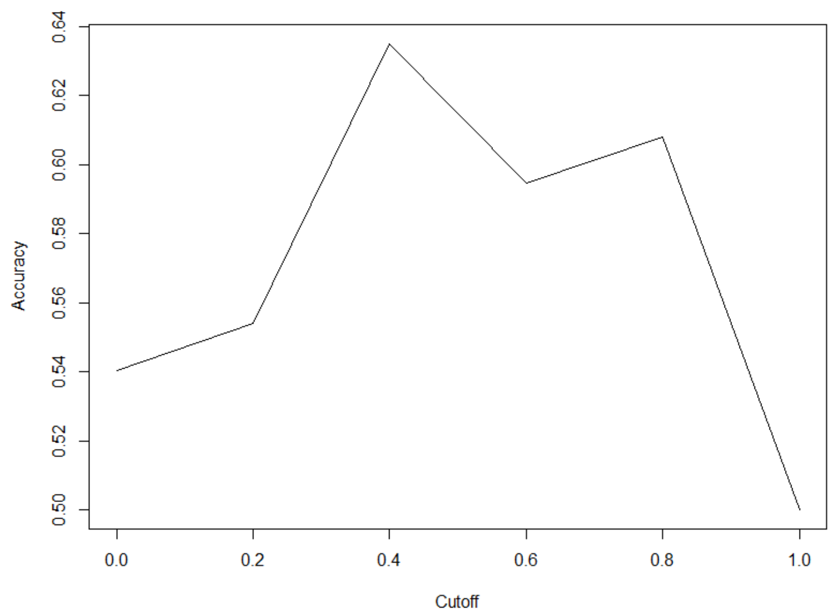
```
start_time <- Sys.time()
models <- caretList(target~., data=training,
trControl=stacking_control, methodList=stacking_algorithms)
end_time <- Sys.time()
end_time - start_time
Time difference of 5.342518 secs
```

<u>K-NN</u>

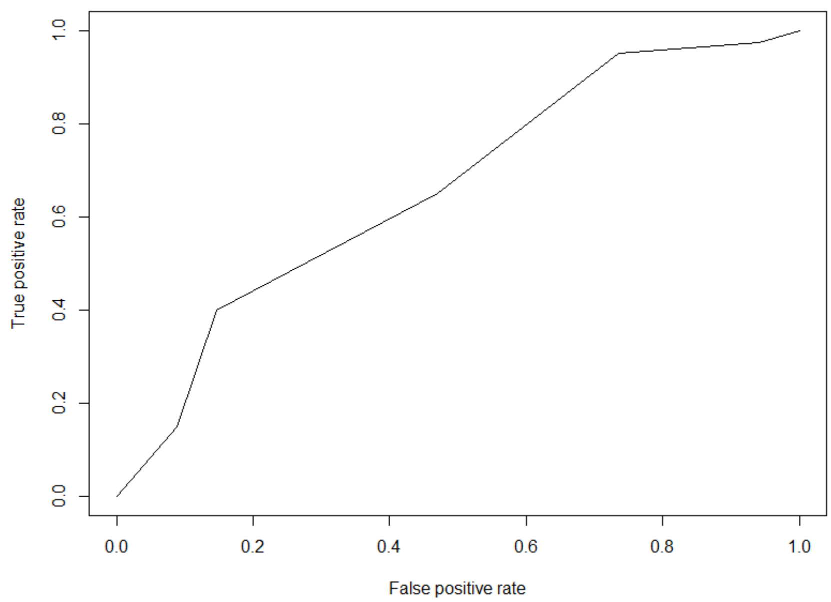
Precision Vs. Recall



Accuracy



Receiver Operating Characteristic Curve



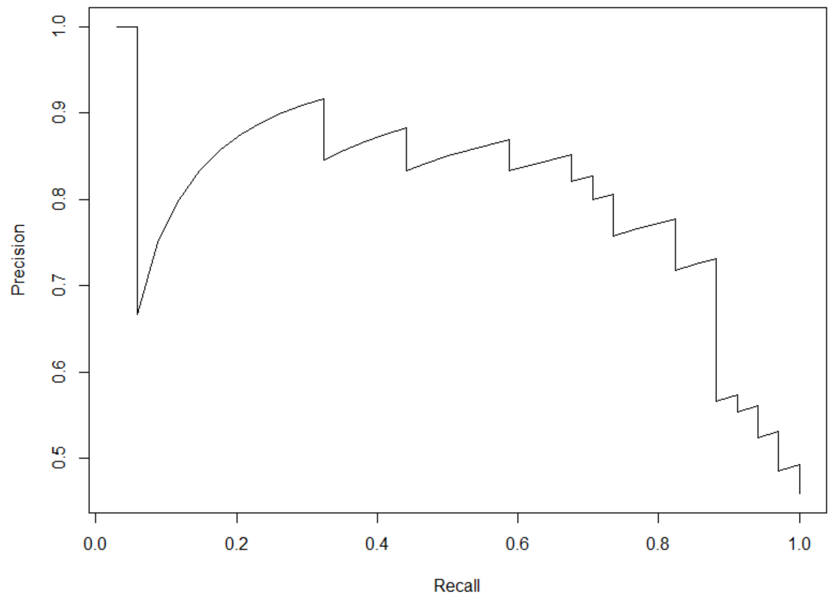
RAUC: 0.6606618

Training time: 5.3seconds (For all 3 algorithms - 'rpart', 'knn', 'nb')

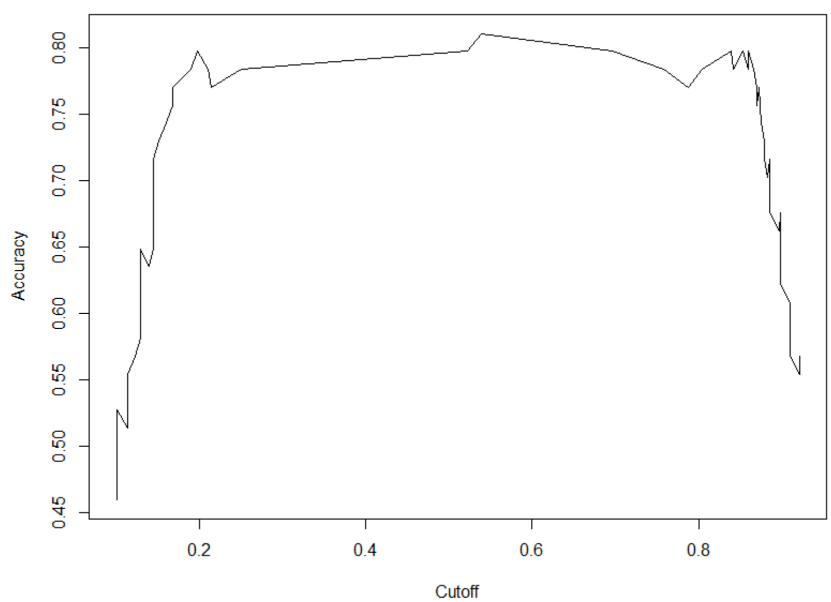
```
start_time <- Sys.time()
models <- caretList(target~., data=training,
trControl=stacking_control, methodList=stacking_algorithms)
end_time <- Sys.time()
end_time - start_time
Time difference of 5.342518 secs
```

Combined predictors model

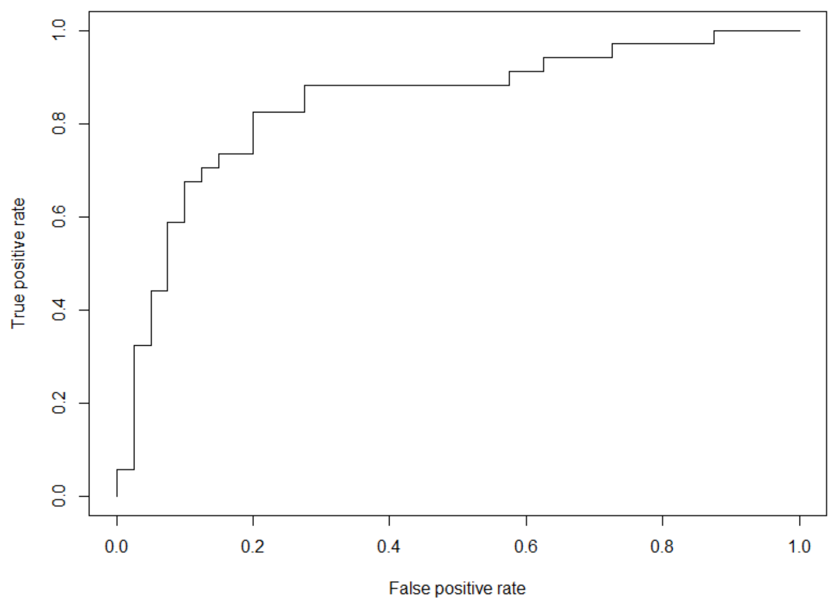
Precision Vs. Recall



Accuracy



Receiver Operating Characteristic Curve



RAUC: 0.8433824

Training time: 1.3 seconds

```
start_time <- Sys.time()
stack_glm <- caretStack(models, method="glm", metric="Accuracy",
trControl=stacking_control)
end_time <- Sys.time()
end_time - start_time
Time difference of 1.267734 secs
```