STAT5703 Data Mining I – Final Project Report

Appendix

Winter 2018

Michael Armanious (100978616), Alex El-Hajj (100887389), Muneer Khan (100650973) and Enrique Reveron (101066270)

April 17th, 2018

Table of Contents

[1. Visualization 2](#_30j0zll)

[1.1 Calling Neccesary Libraries 2](#_3znysh7)

[1.2 Setting Up the Directory and Variables for Reproducibility 5](#_tyjcwt)

[1.3 Calling Neccesary Functions 6](#_1t3h5sf)

[1.4 Loading the Data 6](#_2s8eyo1)

[1.5 Clean the Data 7](#_3rdcrjn)

[1.6 Scaterplot Matrix 9](#_lnxbz9)

[1.7 Create a New Variable (Disease) 11](#_1ksv4uv)

[1.8 Create New Dataset with Factor Variables 12](#_44sinio)

[1.9 Create New Dataset with Factor Variables and Names 13](#_z337ya)

[1.10 Scaterplot Matrix Colored by Disease 15](#_1y810tw)

[1.11 Histogram Plots 16](#_2xcytpi)

[1.12 Mosaic Plots 20](#_3whwml4)

[2. Split the Dataset into Train and Data 24](#_qsh70q)

[3. Mining Association Rules (apriori) 26](#_1pxezwc)

[4. Logistic Regression (Quasibinomial) (glm) 84](#_2p2csry)

[5. Logistic Regression (Binomial) (glm) 91](#_3o7alnk)

[6. Logistic Regression (Binomial) with Factor Variables (glm) 97](#_ihv636)

[7. Linear Model with Stepwise Feature Selection (glmStepAIC) 103](#_1hmsyys)

[8. Kmeans (eclust) 163](#_2grqrue)

[8.1 PCA 163](#_3fwokq0)

[8.2 All Data Whithout PCA 176](#_4f1mdlm)

[9. Support Vector Machine (SVM) 181](#_19c6y18)

[9.1 SVM Radial (svmRadial) 181](#_28h4qwu)

[9.2 Linear SVM (svmLinear) 189](#_37m2jsg)

[10. Neural Networks (nnet) 194](#_46r0co2)

[11. K-Nearest Neighbors (knn) 198](#_111kx3o)

[12. Random Forest 203](#_206ipza)

[12.1 Random Forest with Tuning Parameters (rf) 203](#_2zbgiuw)

[12.2 Random Forest Boosted Tree (bstTree) 208](#_3ygebqi)

[12.3 Random Forest Boost with Tuning Parameters (gbm) 213](#_sqyw64)

[12.4 Random Forest with Stochastic Gradient Boost (gbm) 219](#_1rvwp1q)

[13. Classification Tree (rpart) 226](#_2r0uhxc)

[14. Fast-and-Frugal Decision Trees (FFTrees) 232](#_3q5sasy)

[15. Custom Fast-and-Frugal Decision Trees (FFTrees) 238](#_kgcv8k)

[16. Forest of Fast-and-Frugal Decision Trees (FFForest) 244](#_1jlao46)

[17. Minimun Cost 257](#_2iq8gzs)

[18. Results 259](#_3hv69ve)

[18.1 Mining Association Rules 259](#_4h042r0)

[Train Set 264](#_1baon6m)

[Test Set 269](#_2afmg28)

[17.2 All Prediction Models Results 273](#_39kk8xu)

[Confusion Matrix (Test Set) 275](#_48pi1tg)

[Models Accuracy 277](#_1302m92)

[Models Residuals 283](#_2250f4o)

# 1. Visualization

## 1.1 Calling Necessary Libraries

############################################  
### 1.1 Calling Necessary Libraries   
l\_packages = **c**("stats", "pls", "factoextra", "NbClust","ggplot2", "ggpubr","gridExtra",  
 "fastICA", "plot3D", "caret", "randomForest","bst","pROC",  
 "arules","arulesViz", "rpart", "FFTrees","rpart.plot",  
 "knitr","kableExtra","formatR", "xtable") *# used to create the Report (knitr)*  
  
**for**(p **in** l\_packages){  
 **if** (**!require**(p,character.only = TRUE)) **install.packages**(p)  
 **library**(p,character.only = TRUE)   
}

## Loading required package: pls

##   
## Attaching package: 'pls'

## The following object is masked from 'package:stats':  
##   
## loadings

## Loading required package: factoextra

## Loading required package: ggplot2

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

## Loading required package: NbClust

## Loading required package: ggpubr

## Loading required package: magrittr

## Loading required package: gridExtra

## Loading required package: fastICA

## Loading required package: plot3D

## Loading required package: caret

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:pls':  
##   
## R2

## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

## Loading required package: bst

## Loading required package: gbm

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: splines

## Loading required package: parallel

## Loaded gbm 2.1.3

## Loading required package: pROC

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

## Loading required package: arules

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

## Loading required package: arulesViz

## Loading required package: grid

## Loading required package: rpart

## Loading required package: FFTrees

## O

## / \

## F O

## / \

## F T

## FFTrees v1.3.5. Email: Nathaniel.D.Phillips.is@gmail.com

## FFTrees.guide() opens the package guide. Citation info at citation('FFTrees')

##   
## Attaching package: 'FFTrees'

## The following object is masked from 'package:pROC':  
##   
## auc

## Loading required package: rpart.plot

## Loading required package: kableExtra

## Loading required package: formatR

## Loading required package: xtable

## 1.2 Setting Up the Directory and Variables for Reproducibility

The **folder** variables needs to be updated in order to reproduce the work.

############################################  
### 1.2 Setting Up the Directory and Variables for Reproducibility  
*# Student and Assigment Information Variables*  
AS <- "FINAL-PROJECT"  
*# Folder Variables*  
drive="C:"  
path.upto <- **paste**("Users", "Enrique","Documents",   
 "Carleton", "Winter 2018", "STAT5703 Data Mining I", sep="/" )  
code.dir <- **paste**(drive, path.upto, AS, "Code", sep="/")  
data.dir <- **paste**(drive, path.upto, AS, "Data", sep="/")  
work.dir <- **paste**(drive, path.upto, AS, "Work", sep="/")  
report.dir <- **paste**(drive, path.upto, AS, "Report", sep="/")  
**setwd**(work.dir)  
**getwd**()

## [1] "C:/Users/Enrique/Documents/Carleton/Winter 2018/STAT5703 Data Mining I/FINAL-PROJECT/Work"

*# For reproducibility*  
**set.seed**(12345)

## 1.3 Calling Necessary Functions

############################################  
## 1.3 Calling Neccesary Functions  
**source**(**paste**(code.dir, **paste**(AS, "functions.r", sep=" "), sep="/"))

## [1] "successful!"

**lsf.str**()

## %w/o% : function (x, y)   
## convert\_to\_main.r : function ()   
## correct\_kmeans\_ids : function (class\_ids, kmeans\_ids)   
## count\_missclassifications : function (class\_ids, kmeans\_ids)   
## count\_rows : function (data, column, value)   
## create\_kmeans\_clusters : function (data, method, k\_max, type\_data, var1, var2)   
## elapsed\_time : function (tic1, tic2)   
## f.data.std : function (data)   
## find\_best\_seed\_kv2 : function (data, method, initial\_seed, final\_seed, k, real\_class)   
## find\_best\_seedv2 : function (data, method, min\_seed, max\_seed, min\_k, max\_k, real\_class)   
## find\_unique\_variables : function (data)   
## get.train : function (data.sz, train.sz)   
## has\_empty\_cluster : function (data)   
## panel.cor : function (x, y, digits = 2, prefix = "", cex.cor)   
## panel.hist : function (x, ...)   
## panel.smooth.asp : function (x, y, col = par("col"), bg = NA, pch = par("pch"), cex = 1,   
## col.smooth = "red", span = 2/3, iter = 3, asp, ...)   
## per\_rows : function (data, column, value)   
## plot\_clusters : function (clusters\_plots)   
## print\_kable : function (data, num\_lines = 0, latex\_options = 0, caption = NULL)   
## print\_kablev2 : function (data, num\_lines = 0, latex\_options = 0, caption = NULL)   
## print\_table : function (data, num\_lines = 0, latex\_options = 0, caption = NULL)   
## Sphere.Data : function (data)   
## std.to.orig : function (std.coef, mean.X, mean.Y, s.X, s.Y)

## 1.4 Loading the Data

############################################  
## 1.4 Loading the Data  
data.file <- **paste**(data.dir, "clev.csv", sep="/")  
heart.data<-**read.csv**(data.file, header=TRUE)  
heart.data**$**Ca <- **gsub**("?",NA,heart.data**$**Ca, fixed = TRUE)  
heart.data**$**Ca<-**as.numeric**(heart.data**$**Ca)  
heart.data**$**Thal<-**gsub**("?",NA,heart.data**$**Thal, fixed = TRUE)  
heart.data**$**Thal<-**as.numeric**(heart.data**$**Thal)  
heart.data<-**as.data.frame**(heart.data)  
**summary**(heart.data)

## Age Sex CP Trestbps Chol   
## Min. :29.0 Min. :0.00 Min. :1.00 Min. : 94 Min. :126   
## 1st Qu.:48.0 1st Qu.:0.00 1st Qu.:3.00 1st Qu.:120 1st Qu.:211   
## Median :56.0 Median :1.00 Median :3.00 Median :130 Median :241   
## Mean :54.4 Mean :0.68 Mean :3.16 Mean :132 Mean :247   
## 3rd Qu.:61.0 3rd Qu.:1.00 3rd Qu.:4.00 3rd Qu.:140 3rd Qu.:275   
## Max. :77.0 Max. :1.00 Max. :4.00 Max. :200 Max. :564   
##   
## Fbs Restecg Thalach Exang   
## Min. :0.000 Min. :0.00 Min. : 71 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.00 1st Qu.:134 1st Qu.:0.000   
## Median :0.000 Median :1.00 Median :153 Median :0.000   
## Mean :0.148 Mean :0.99 Mean :150 Mean :0.327   
## 3rd Qu.:0.000 3rd Qu.:2.00 3rd Qu.:166 3rd Qu.:1.000   
## Max. :1.000 Max. :2.00 Max. :202 Max. :1.000   
##   
## Oldpeak Slope Ca Thal   
## Min. :0.00 Min. :1.0 Min. :0.000 Min. :3.00   
## 1st Qu.:0.00 1st Qu.:1.0 1st Qu.:0.000 1st Qu.:3.00   
## Median :0.80 Median :2.0 Median :0.000 Median :3.00   
## Mean :1.04 Mean :1.6 Mean :0.672 Mean :4.73   
## 3rd Qu.:1.60 3rd Qu.:2.0 3rd Qu.:1.000 3rd Qu.:7.00   
## Max. :6.20 Max. :3.0 Max. :3.000 Max. :7.00   
## NA's :4 NA's :2   
## Num   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.937   
## 3rd Qu.:2.000   
## Max. :4.000   
##

## 1.5 Clean the Data

############################################  
## 1.5 Clean the Data   
*# We can notice 6 NA's on variables "Ca" and "Thal", let's see those rows*  
**dim**(heart.data)

## [1] 303 14

(heart.data.na <- **subset**(heart.data,(**is.na**(heart.data["Ca"]) **|** **is.na**(heart.data["Thal"]))))

## Age Sex CP Trestbps Chol Fbs Restecg Thalach Exang Oldpeak Slope Ca  
## 88 53 0 3 128 216 0 2 115 0 0.0 1 0  
## 167 52 1 3 138 223 0 0 169 0 0.0 1 NA  
## 193 43 1 4 132 247 1 2 143 1 0.1 2 NA  
## 267 52 1 4 128 204 1 0 156 1 1.0 2 0  
## 288 58 1 2 125 220 0 0 144 0 0.4 2 NA  
## 303 38 1 3 138 175 0 0 173 0 0.0 1 NA  
## Thal Num  
## 88 NA 0  
## 167 3 0  
## 193 7 1  
## 267 NA 2  
## 288 7 0  
## 303 3 0

*# Number of rows with NAs values*  
**nrow**(heart.data.na)

## [1] 6

*# Percentage*  
(**nrow**(heart.data.na)**/nrow**(heart.data) **\*** 100 )

## [1] 1.98

*# Because is only 2% I suggest to remove those rows*   
heart.data <- **subset**(heart.data, **!**(**is.na**(heart.data["Ca"]) **|** **is.na**(heart.data["Thal"])))  
  
*#Check the new data*  
**summary**(heart.data)

## Age Sex CP Trestbps Chol   
## Min. :29.0 Min. :0.000 Min. :1.00 Min. : 94 Min. :126   
## 1st Qu.:48.0 1st Qu.:0.000 1st Qu.:3.00 1st Qu.:120 1st Qu.:211   
## Median :56.0 Median :1.000 Median :3.00 Median :130 Median :243   
## Mean :54.5 Mean :0.677 Mean :3.16 Mean :132 Mean :247   
## 3rd Qu.:61.0 3rd Qu.:1.000 3rd Qu.:4.00 3rd Qu.:140 3rd Qu.:276   
## Max. :77.0 Max. :1.000 Max. :4.00 Max. :200 Max. :564   
## Fbs Restecg Thalach Exang   
## Min. :0.000 Min. :0.000 Min. : 71 Min. :0.000   
## 1st Qu.:0.000 1st Qu.:0.000 1st Qu.:133 1st Qu.:0.000   
## Median :0.000 Median :1.000 Median :153 Median :0.000   
## Mean :0.145 Mean :0.997 Mean :150 Mean :0.327   
## 3rd Qu.:0.000 3rd Qu.:2.000 3rd Qu.:166 3rd Qu.:1.000   
## Max. :1.000 Max. :2.000 Max. :202 Max. :1.000   
## Oldpeak Slope Ca Thal   
## Min. :0.00 Min. :1.0 Min. :0.000 Min. :3.00   
## 1st Qu.:0.00 1st Qu.:1.0 1st Qu.:0.000 1st Qu.:3.00   
## Median :0.80 Median :2.0 Median :0.000 Median :3.00   
## Mean :1.06 Mean :1.6 Mean :0.677 Mean :4.73   
## 3rd Qu.:1.60 3rd Qu.:2.0 3rd Qu.:1.000 3rd Qu.:7.00   
## Max. :6.20 Max. :3.0 Max. :3.000 Max. :7.00   
## Num   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.946   
## 3rd Qu.:2.000   
## Max. :4.000

**str**(heart.data)

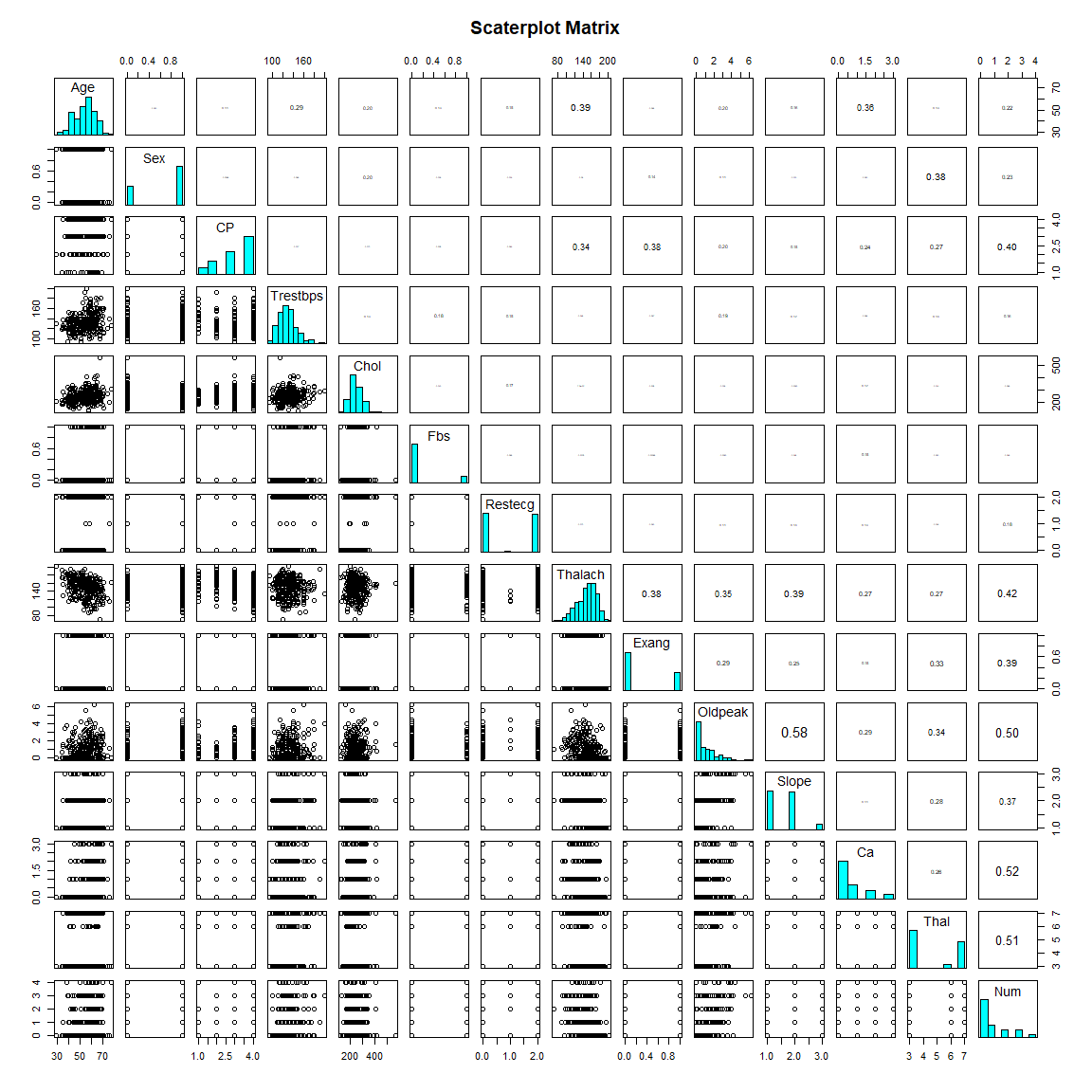
## 'data.frame': 297 obs. of 14 variables:  
## $ Age : int 63 67 67 37 41 56 62 57 63 53 ...  
## $ Sex : int 1 1 1 1 0 1 0 0 1 1 ...  
## $ CP : int 1 4 4 3 2 2 4 4 4 4 ...  
## $ Trestbps: int 145 160 120 130 130 120 140 120 130 140 ...  
## $ Chol : int 233 286 229 250 204 236 268 354 254 203 ...  
## $ Fbs : int 1 0 0 0 0 0 0 0 0 1 ...  
## $ Restecg : int 2 2 2 0 2 0 2 0 2 2 ...  
## $ Thalach : int 150 108 129 187 172 178 160 163 147 155 ...  
## $ Exang : int 0 1 1 0 0 0 0 1 0 1 ...  
## $ Oldpeak : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ Slope : int 3 2 2 3 1 1 3 1 2 3 ...  
## $ Ca : num 0 3 2 0 0 0 2 0 1 0 ...  
## $ Thal : num 6 3 7 3 3 3 3 3 7 7 ...  
## $ Num : int 0 2 1 0 0 0 3 0 2 1 ...

**dim**(heart.data)

## [1] 297 14

## 1.6 Scatterplot Matrix

############################################  
## 1.6 Some Plots  
**pairs**(heart.data, upper.panel=panel.cor, diag.panel=panel.hist,   
 main = "Scaterplot Matrix")



**kable**(**var**(heart.data))

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Age | Sex | CP | Trestbps | Chol | Fbs | Restecg | Thalach | Exang | Oldpeak | Slope | Ca | Thal | Num |
| Age | 81.8977 | -0.3918 | 0.9646 | 46.6937 | 95.3568 | 0.4213 | 1.3498 | -81.9172 | 0.4102 | 2.0803 | 0.8918 | 3.0778 | 2.2208 | 2.4820 |
| Sex | -0.3918 | 0.2195 | 0.0040 | -0.5521 | -4.8256 | 0.0064 | 0.0158 | -0.6502 | 0.0316 | 0.0582 | 0.0097 | 0.0404 | 0.3485 | 0.1312 |
| CP | 0.9646 | 0.0040 | 0.9310 | -0.6338 | 3.6167 | -0.0196 | 0.0613 | -7.5107 | 0.1711 | 0.2287 | 0.0901 | 0.2135 | 0.5022 | 0.4815 |
| Trestbps | 46.6937 | -0.5521 | -0.6338 | 315.5173 | 121.4894 | 1.1323 | 2.6375 | -20.0117 | 0.5565 | 3.9613 | 1.3306 | 1.6337 | 4.7584 | 3.5003 |
| Chol | 95.3568 | -4.8256 | 3.6167 | 121.4894 | 2703.7486 | 0.2329 | 8.5383 | -0.0890 | 1.4494 | 2.3403 | -0.2962 | 5.6609 | 1.0946 | 4.2655 |
| Fbs | 0.4213 | 0.0064 | -0.0196 | 1.1323 | 0.2329 | 0.1242 | 0.0241 | -0.0634 | -0.0001 | 0.0034 | 0.0104 | 0.0503 | 0.0425 | 0.0213 |
| Restecg | 1.3498 | 0.0158 | 0.0613 | 2.6375 | 8.5383 | 0.0241 | 0.9899 | -1.6500 | 0.0383 | 0.1319 | 0.0831 | 0.1205 | 0.0363 | 0.2262 |
| Thalach | -81.9172 | -0.6502 | -7.5107 | -20.0117 | -0.0890 | -0.0634 | -1.6500 | 526.3153 | -4.1423 | -9.3003 | -5.5212 | -5.7887 | -12.2232 | -11.9135 |
| Exang | 0.4102 | 0.0316 | 0.1711 | 0.5565 | 1.4494 | -0.0001 | 0.0383 | -4.1423 | 0.2207 | 0.1585 | 0.0728 | 0.0654 | 0.2977 | 0.2271 |
| Oldpeak | 2.0803 | 0.0582 | 0.2287 | 3.9613 | 2.3403 | 0.0034 | 0.1319 | -9.3003 | 0.1585 | 1.3598 | 0.4174 | 0.3224 | 0.7799 | 0.7219 |
| Slope | 0.8918 | 0.0097 | 0.0901 | 1.3306 | -0.2962 | 0.0104 | 0.0831 | -5.5212 | 0.0728 | 0.4174 | 0.3822 | 0.0637 | 0.3352 | 0.2860 |
| Ca | 3.0778 | 0.0404 | 0.2135 | 1.6337 | 5.6609 | 0.0503 | 0.1205 | -5.7887 | 0.0654 | 0.3224 | 0.0637 | 0.8817 | 0.4667 | 0.6041 |
| Thal | 2.2208 | 0.3485 | 0.5022 | 4.7584 | 1.0946 | 0.0425 | 0.0363 | -12.2232 | 0.2977 | 0.7799 | 0.3352 | 0.4667 | 3.7583 | 1.2287 |
| Num | 2.4820 | 0.1312 | 0.4815 | 3.5003 | 4.2655 | 0.0213 | 0.2262 | -11.9135 | 0.2271 | 0.7219 | 0.2860 | 0.6041 | 1.2287 | 1.5241 |

## 1.7 Create a New Variable (Disease)

############################################  
## 1.7 Create a New Variable (Disease)   
*# This variable will be used in the analysis, if Num == 0 it means heart is Ok (healthy) and*  
*# if Num > 0 means heart disease (non-healthy)*  
heart.data**$**Disease <- **ifelse**((heart.data**$**Num **!=** 0),1,0)  
heart.data**$**Disease <- **factor**(heart.data**$**Disease)  
**str**(heart.data**$**Disease)

## Factor w/ 2 levels "0","1": 1 2 2 1 1 1 2 1 2 2 ...

## 1.8 Create New Dataset with Factor Variables

############################################  
## 1.8 Create New Dataset with Factor Variables  
heart.factor <- heart.data  
heart.factor**$**Age <- **factor**(heart.factor**$**Age)  
heart.factor**$**Sex <- **factor**(heart.factor**$**Sex)  
heart.factor**$**CP <- **factor**(heart.factor**$**CP)  
heart.factor**$**Fbs <- **factor**(heart.factor**$**Fbs)  
heart.factor**$**Restecg <- **factor**(heart.factor**$**Restecg)  
heart.factor**$**Exang <- **factor**(heart.factor**$**Exang)  
heart.factor**$**Slope <- **factor**(heart.factor**$**Slope)  
heart.factor**$**Ca <- **factor**(heart.factor**$**Ca)  
heart.factor**$**Thal <- **factor**(heart.factor**$**Thal)  
heart.factor**$**Num <- **factor**(heart.factor**$**Num)  
*# Check the new Data*  
**summary**(heart.factor)

## Age Sex CP Trestbps Chol Fbs Restecg  
## 58 : 18 0: 96 1: 23 Min. : 94 Min. :126 0:254 0:147   
## 57 : 17 1:201 2: 49 1st Qu.:120 1st Qu.:211 1: 43 1: 4   
## 54 : 16 3: 83 Median :130 Median :243 2:146   
## 59 : 14 4:142 Mean :132 Mean :247   
## 51 : 12 3rd Qu.:140 3rd Qu.:276   
## 60 : 12 Max. :200 Max. :564   
## (Other):208   
## Thalach Exang Oldpeak Slope Ca Thal Num   
## Min. : 71 0:200 Min. :0.00 1:139 0:174 3:164 0:160   
## 1st Qu.:133 1: 97 1st Qu.:0.00 2:137 1: 65 6: 18 1: 54   
## Median :153 Median :0.80 3: 21 2: 38 7:115 2: 35   
## Mean :150 Mean :1.06 3: 20 3: 35   
## 3rd Qu.:166 3rd Qu.:1.60 4: 13   
## Max. :202 Max. :6.20   
##   
## Disease  
## 0:160   
## 1:137   
##   
##   
##   
##   
##

**str**(heart.factor)

## 'data.frame': 297 obs. of 15 variables:  
## $ Age : Factor w/ 41 levels "29","34","35",..: 30 34 34 4 8 23 29 24 30 20 ...  
## $ Sex : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 1 1 2 2 ...  
## $ CP : Factor w/ 4 levels "1","2","3","4": 1 4 4 3 2 2 4 4 4 4 ...  
## $ Trestbps: int 145 160 120 130 130 120 140 120 130 140 ...  
## $ Chol : int 233 286 229 250 204 236 268 354 254 203 ...  
## $ Fbs : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...  
## $ Restecg : Factor w/ 3 levels "0","1","2": 3 3 3 1 3 1 3 1 3 3 ...  
## $ Thalach : int 150 108 129 187 172 178 160 163 147 155 ...  
## $ Exang : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 2 1 2 ...  
## $ Oldpeak : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ Slope : Factor w/ 3 levels "1","2","3": 3 2 2 3 1 1 3 1 2 3 ...  
## $ Ca : Factor w/ 4 levels "0","1","2","3": 1 4 3 1 1 1 3 1 2 1 ...  
## $ Thal : Factor w/ 3 levels "3","6","7": 2 1 3 1 1 1 1 1 3 3 ...  
## $ Num : Factor w/ 5 levels "0","1","2","3",..: 1 3 2 1 1 1 4 1 3 2 ...  
## $ Disease : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 2 1 2 2 ...

**dim**(heart.factor)

## [1] 297 15

## 1.9 Create New Dataset with Factor Variables and Names

############################################  
## 1.9 Create New Dataset with Factor Variables  
heart.factor.names <- heart.factor  
  
**levels**(heart.factor.names**$**Sex) <- **c**("Female", "Male")  
  
**levels**(heart.factor.names**$**CP) <- **c**("Typical angina","Atypical angina",  
 "Non-anginal pain","Asymptomatic")  
  
**levels**(heart.factor.names**$**Fbs) <- **c**("< 120 mg/dl", "> 120 mg/dl")  
  
**levels**(heart.factor.names**$**Restecg) <- **c**("Normal", "Having ST-T wave abnormality",   
 "Showing left ventricular hypertrophy")  
  
**levels**(heart.factor.names**$**Exang) <- **c**("No", "Yes")  
  
**levels**(heart.factor.names**$**Slope) <- **c**("Upsloping", "Flat", "Downsloping")  
  
**levels**(heart.factor.names**$**Thal) <- **c**("Normal", "Fixed Defect",   
 "Reversible Defect")  
  
**levels**(heart.factor.names**$**Disease) <- **c**("Healthy", "Non-healthy")  
  
*# Check the new Data*  
**summary**(heart.factor.names)

## Age Sex CP Trestbps   
## 58 : 18 Female: 96 Typical angina : 23 Min. : 94   
## 57 : 17 Male :201 Atypical angina : 49 1st Qu.:120   
## 54 : 16 Non-anginal pain: 83 Median :130   
## 59 : 14 Asymptomatic :142 Mean :132   
## 51 : 12 3rd Qu.:140   
## 60 : 12 Max. :200   
## (Other):208   
## Chol Fbs   
## Min. :126 < 120 mg/dl:254   
## 1st Qu.:211 > 120 mg/dl: 43   
## Median :243   
## Mean :247   
## 3rd Qu.:276   
## Max. :564   
##   
## Restecg Thalach Exang   
## Normal :147 Min. : 71 No :200   
## Having ST-T wave abnormality : 4 1st Qu.:133 Yes: 97   
## Showing left ventricular hypertrophy:146 Median :153   
## Mean :150   
## 3rd Qu.:166   
## Max. :202   
##   
## Oldpeak Slope Ca Thal Num   
## Min. :0.00 Upsloping :139 0:174 Normal :164 0:160   
## 1st Qu.:0.00 Flat :137 1: 65 Fixed Defect : 18 1: 54   
## Median :0.80 Downsloping: 21 2: 38 Reversible Defect:115 2: 35   
## Mean :1.06 3: 20 3: 35   
## 3rd Qu.:1.60 4: 13   
## Max. :6.20   
##   
## Disease   
## Healthy :160   
## Non-healthy:137   
##   
##   
##   
##   
##

**str**(heart.factor.names)

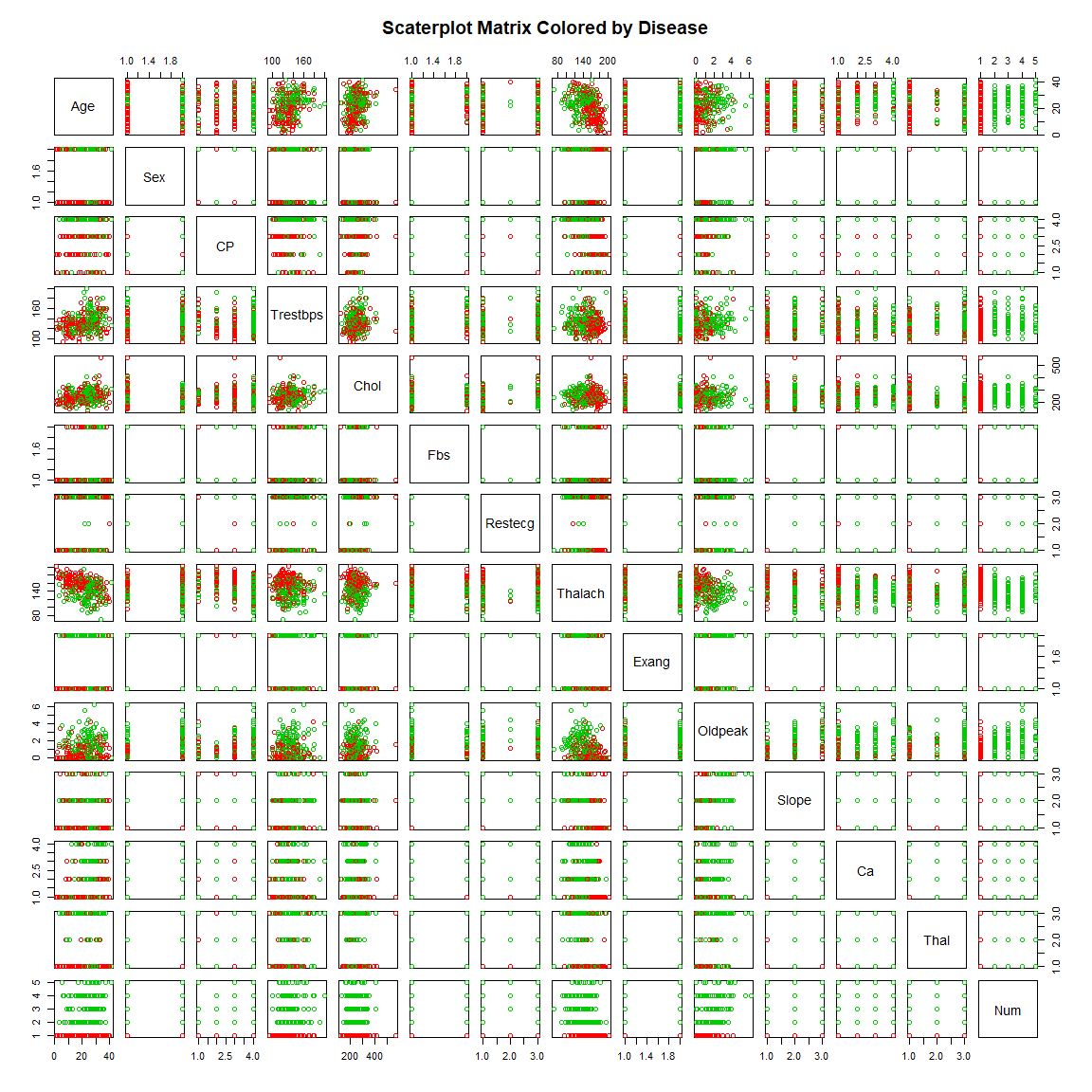
## 'data.frame': 297 obs. of 15 variables:  
## $ Age : Factor w/ 41 levels "29","34","35",..: 30 34 34 4 8 23 29 24 30 20 ...  
## $ Sex : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 1 2 2 ...  
## $ CP : Factor w/ 4 levels "Typical angina",..: 1 4 4 3 2 2 4 4 4 4 ...  
## $ Trestbps: int 145 160 120 130 130 120 140 120 130 140 ...  
## $ Chol : int 233 286 229 250 204 236 268 354 254 203 ...  
## $ Fbs : Factor w/ 2 levels "< 120 mg/dl",..: 2 1 1 1 1 1 1 1 1 2 ...  
## $ Restecg : Factor w/ 3 levels "Normal","Having ST-T wave abnormality",..: 3 3 3 1 3 1 3 1 3 3 ...  
## $ Thalach : int 150 108 129 187 172 178 160 163 147 155 ...  
## $ Exang : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 2 1 2 ...  
## $ Oldpeak : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ Slope : Factor w/ 3 levels "Upsloping","Flat",..: 3 2 2 3 1 1 3 1 2 3 ...  
## $ Ca : Factor w/ 4 levels "0","1","2","3": 1 4 3 1 1 1 3 1 2 1 ...  
## $ Thal : Factor w/ 3 levels "Normal","Fixed Defect",..: 2 1 3 1 1 1 1 1 3 3 ...  
## $ Num : Factor w/ 5 levels "0","1","2","3",..: 1 3 2 1 1 1 4 1 3 2 ...  
## $ Disease : Factor w/ 2 levels "Healthy","Non-healthy": 1 2 2 1 1 1 2 1 2 2 ...

**dim**(heart.factor.names)

## [1] 297 15

## 1.10 Scatterplot Matrix Colored by Disease

############################################  
## 1.10 Scatterplot Matrix Colored by Disease  
*# Plot colored by Disease*  
n.var <- **length**(heart.factor.names)  
**pairs**(heart.factor.names[,1**:**(n.var**-**1)],   
 col = **as.numeric**(heart.factor.names**$**Disease)**+**1,   
 main = "Scatterplot Matrix Colored by Disease")



## 1.11 Histogram Plots

############################################  
## 1.11 Histogram Plots  
*# Histogram Age by Disease*  
g.age <- **ggplot**(heart.factor.names, **aes**(x = Age, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**   
 **theme**(axis.text.x=**element\_text**(angle=90, hjust=1)) **+**  
 **ggtitle**("Histogram: Age by Disease") **+** **xlab**("Age (years)")

*# Histogram Sex by Disease*  
g.sex <- **ggplot**(heart.factor.names, **aes**(x = **factor**(Sex), fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Sex by Disease") **+** **xlab**("Sex")

*# Histogram CP by Disease*  
g.cp <- **ggplot**(heart.factor.names, **aes**(x = CP, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Type of Chest Pain by Disease") **+**  
 **xlab**("Type of Chest Pain")

*# Histogram TrestBps by Disease*  
g.trestbps <- **ggplot**(heart.factor.names, **aes**(x = Trestbps, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, bins = 30) **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Resting Blood Pressure by Disease") **+**  
 **xlab**("Resting Blood Pressure (mm Hg)")  
  
*# Histogram Chol by Disease*  
g.chol <- **ggplot**(heart.factor.names, **aes**(x = Chol, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, bins = 30) **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Serum Cholesterol by Disease") **+**  
 **xlab**("Serum Cholesterol (mg/dl)")  
  
*# Histogram Fbs by Disease*  
g.fbs <- **ggplot**(heart.factor.names, **aes**(x = Fbs, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Fasting Blood Sugar by Disease") **+**   
 **xlab**("Fasting Blood Sugar (mg/dl)")

*# Histogram Restecg by Disease*  
g.restegcg <- **ggplot**(heart.factor.names, **aes**(x = Restecg, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Resting Electrocardiogram by Disease") **+**   
 **xlab**("Resting ECG")

*# Histogram Thalach by Disease*  
g.thalach <- **ggplot**(heart.factor.names, **aes**(x = Thalach, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, bins = 30) **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Thalium Test Max Heart Rate by Disease") **+**   
 **xlab**("Thalium Test Max Heart Rate")  
  
*# Histogram Exang by Disease*  
g.exang <- **ggplot**(heart.factor.names, **aes**(x = Exang, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Exercise Induced Angina by Disease") **+**   
 **xlab**("Exercise Induced Angina")

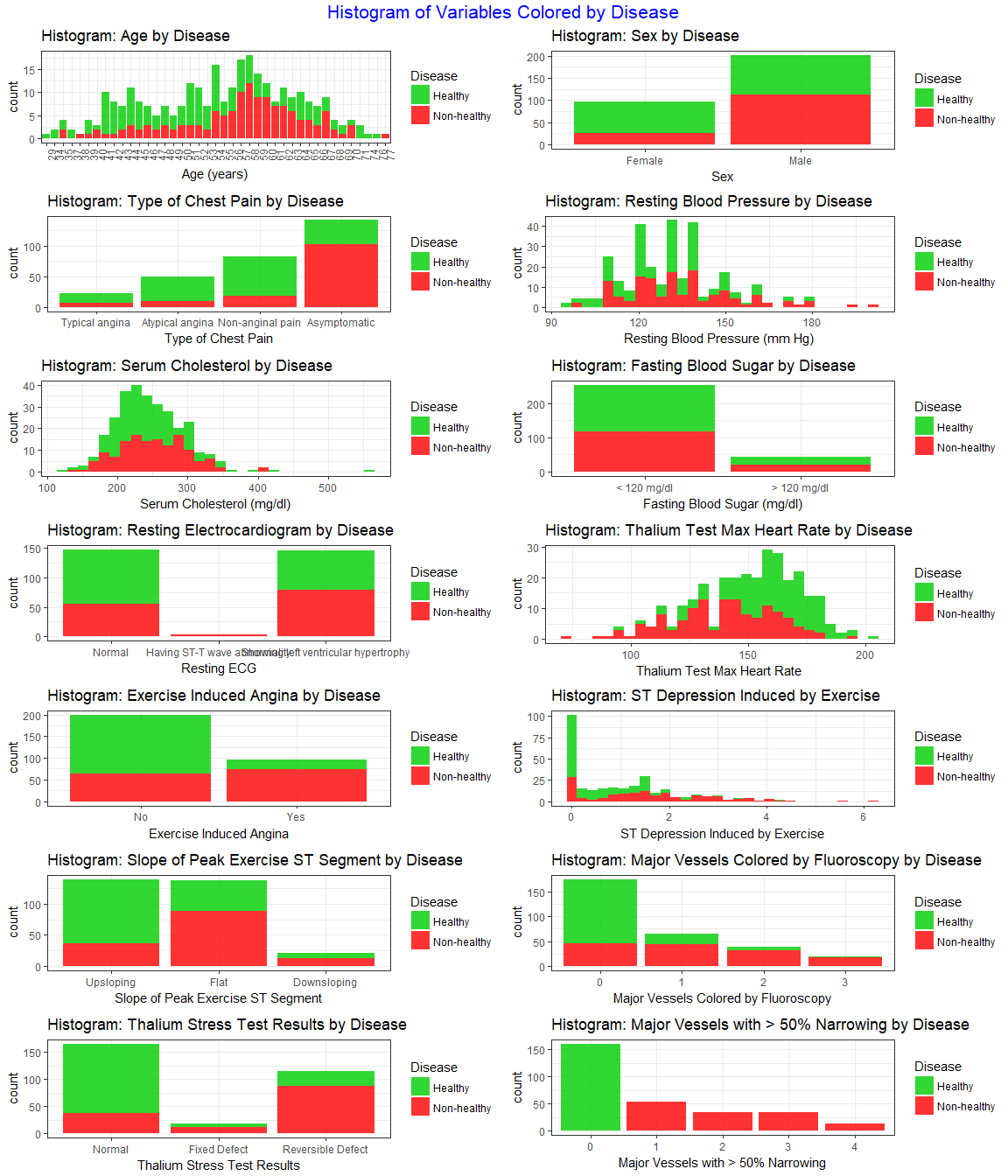
*# Histogram Oldpeak*  
g.oldpeak <- **ggplot**(heart.factor.names, **aes**(x = Oldpeak, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, bins = 30) **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: ST Depression Induced by Exercise") **+**   
 **xlab**("ST Depression Induced by Exercise")  
  
*# Histogram Slope by Disease*  
g.slope <- **ggplot**(heart.factor.names, **aes**(x = Slope, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Slope of Peak Exercise ST Segment by Disease") **+**   
 **xlab**("Slope of Peak Exercise ST Segment")

*# Histogram Ca by Disease*  
g.ca <- **ggplot**(heart.factor.names, **aes**(x = Ca, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Major Vessels Colored by Fluoroscopy by Disease") **+**   
 **xlab**("Major Vessels Colored by Fluoroscopy")

*# Histogram Thal by Disease*  
g.thal <- **ggplot**(heart.factor.names, **aes**(x = Thal, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Thalium Stress Test Results by Disease") **+**   
 **xlab**("Thalium Stress Test Results")

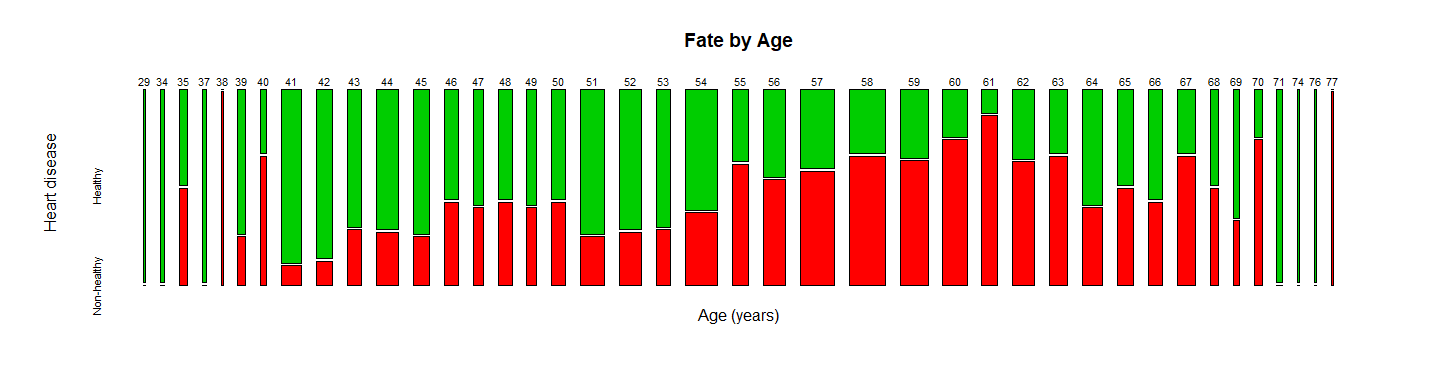
*# Histogram Num by Disease*  
g.num <- **ggplot**(heart.factor.names, **aes**(x = Num, fill = Disease)) **+**  
 **geom\_histogram**(alpha = .8, stat = "count") **+**   
 **scale\_fill\_manual**(values=**c**(3, 2)) **+** **theme\_bw**() **+**  
 **ggtitle**("Histogram: Major Vessels with > 50% Narrowing by Disease") **+**   
 **xlab**("Major Vessels with > 50% Narrowing")

*# Plot all the Histograms*  
g <- **grid.arrange**(  
 top = **text\_grob**(label = "Histogram of Variables Colored by Disease",   
 color = "blue", size=16 ),  
 g.age,  
 g.sex,  
 g.cp,  
 g.trestbps,  
 g.chol,  
 g.fbs,  
 g.restegcg,  
 g.thalach,  
 g.exang,  
 g.oldpeak,  
 g.slope,  
 g.ca,  
 g.thal,  
 g.num,  
 ncol = 2)

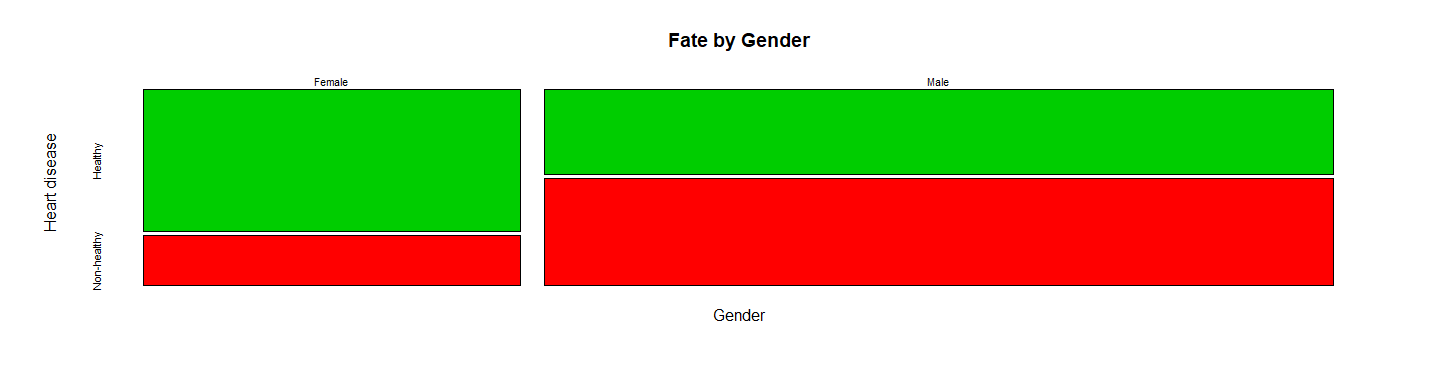


## 1.12 Mosaic Plots

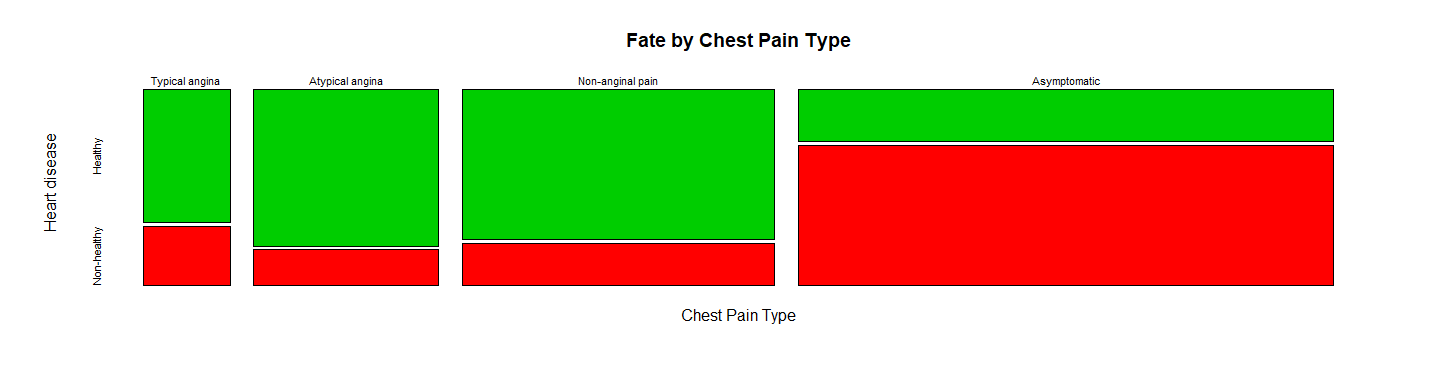
############################################  
## 1.12 Mosaic Plots  
**mosaicplot**(heart.factor.names**$**Age **~** heart.factor.names**$**Disease,  
 main="Fate by Age", shade=FALSE,color=**c**(3,2),  
 xlab="Age (years)", ylab="Heart disease")



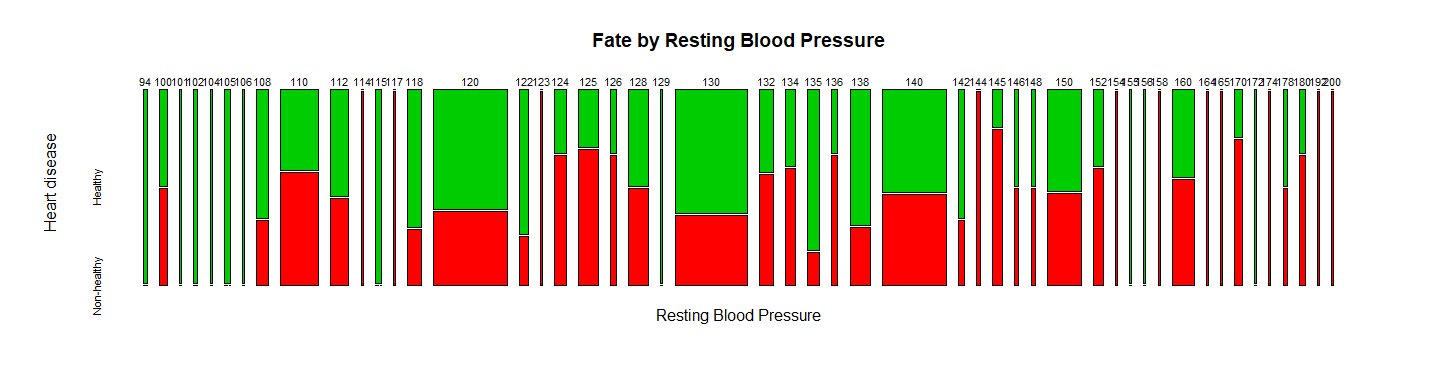
**mosaicplot**(heart.factor.names**$**Sex **~** heart.factor.names**$**Disease,  
 main="Fate by Gender", shade=FALSE,color=**c**(3,2),  
 xlab="Gender", ylab="Heart disease")



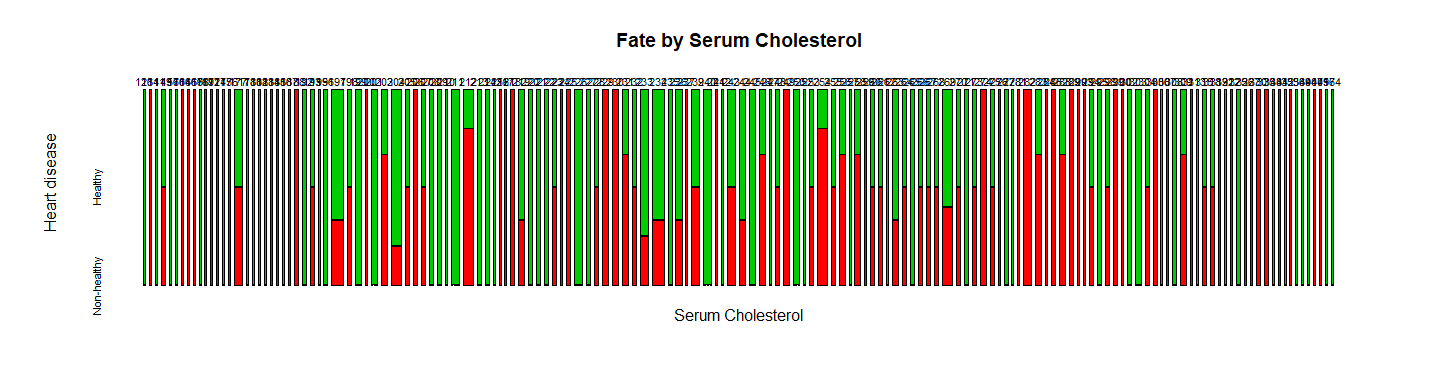
**mosaicplot**(heart.factor.names**$**CP **~** heart.factor.names**$**Disease,  
 main="Fate by Chest Pain Type", shade=FALSE,color=**c**(3,2),  
 xlab="Chest Pain Type", ylab="Heart disease")



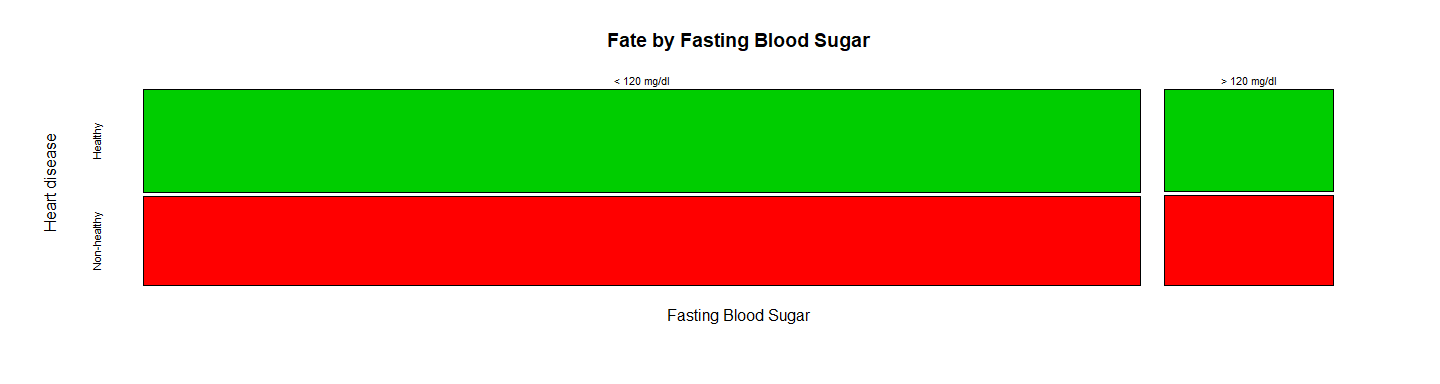
**mosaicplot**(heart.factor.names**$**Trestbps **~** heart.factor.names**$**Disease,  
 main="Fate by Resting Blood Pressure", shade=FALSE,color=**c**(3,2),  
 xlab="Resting Blood Pressure", ylab="Heart disease")



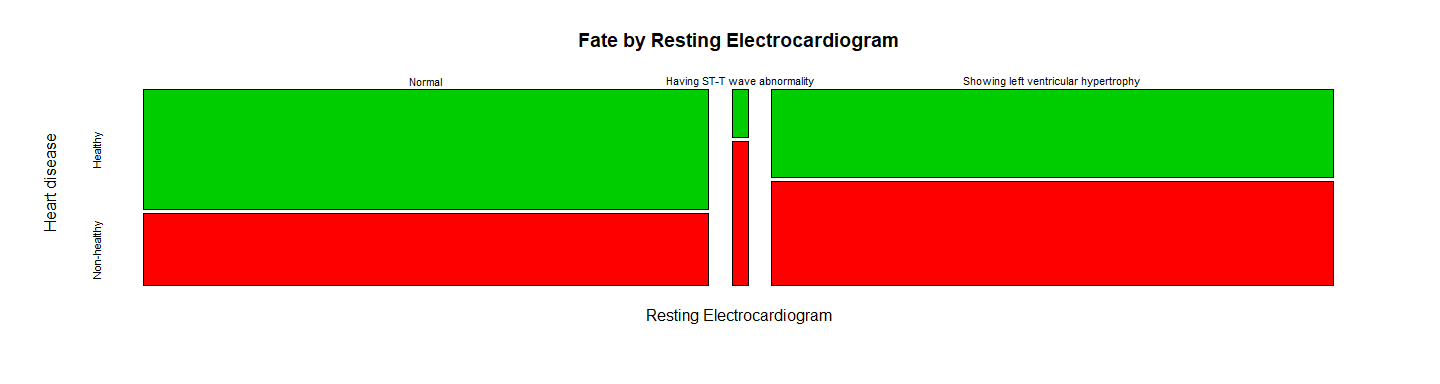
**mosaicplot**(heart.factor.names**$**Chol **~** heart.factor.names**$**Disease,  
 main="Fate by Serum Cholesterol", shade=FALSE,color=**c**(3,2),  
 xlab="Serum Cholesterol", ylab="Heart disease")



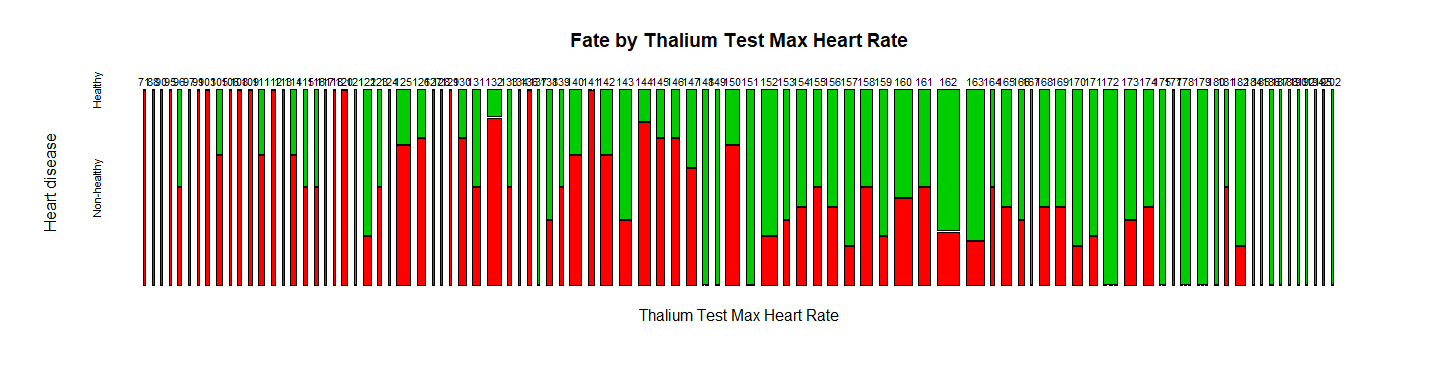
**mosaicplot**(heart.factor.names**$**Fbs **~** heart.factor.names**$**Disease,  
 main="Fate by Fasting Blood Sugar", shade=FALSE,color=**c**(3,2),  
 xlab="Fasting Blood Sugar", ylab="Heart disease")



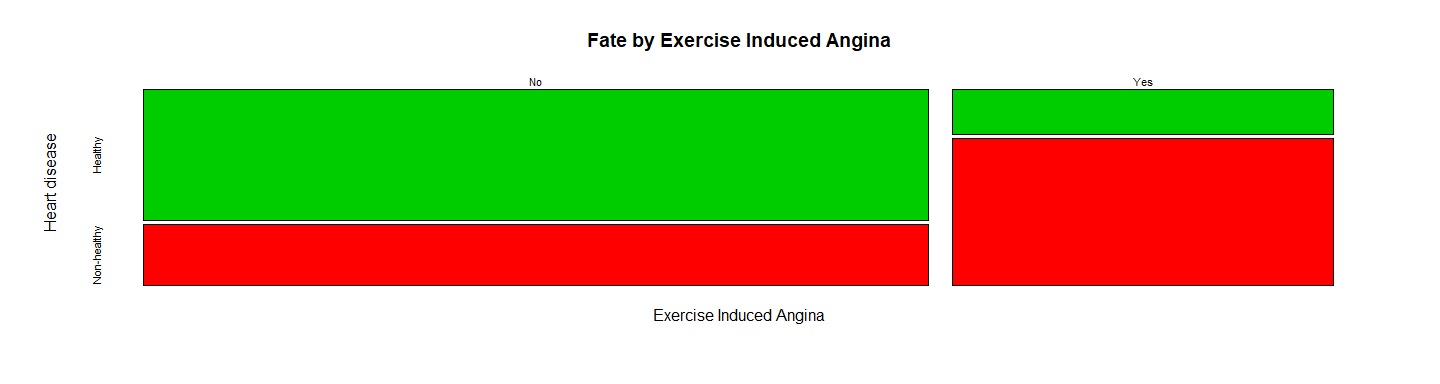
**mosaicplot**(heart.factor.names**$**Restecg **~** heart.factor.names**$**Disease,  
 main="Fate by Resting Electrocardiogram", shade=FALSE,color=**c**(3,2),  
 xlab="Resting Electrocardiogram", ylab="Heart disease")



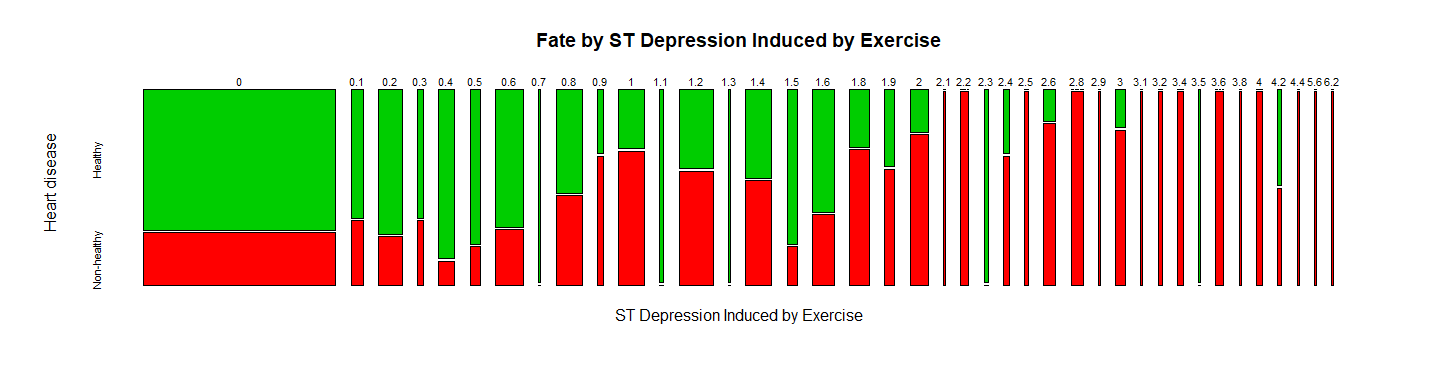
**mosaicplot**(heart.factor.names**$**Thalach **~** heart.factor.names**$**Disease,  
 main="Fate by Thalium Test Max Heart Rate", shade=FALSE,color=**c**(3,2),  
 xlab="Thalium Test Max Heart Rate", ylab="Heart disease")



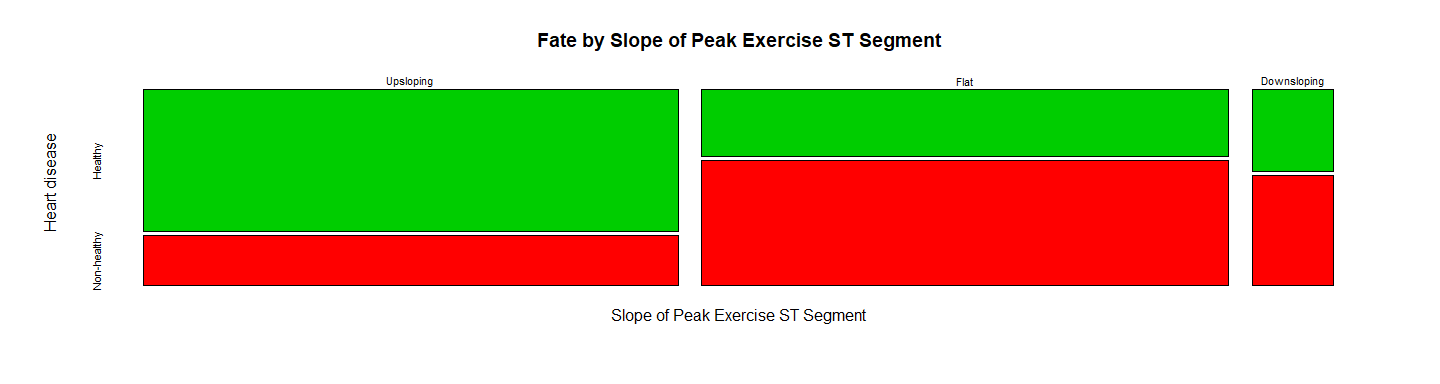
**mosaicplot**(heart.factor.names**$**Exang **~** heart.factor.names**$**Disease,  
 main="Fate by Exercise Induced Angina", shade=FALSE,color=**c**(3,2),  
 xlab="Exercise Induced Angina", ylab="Heart disease")



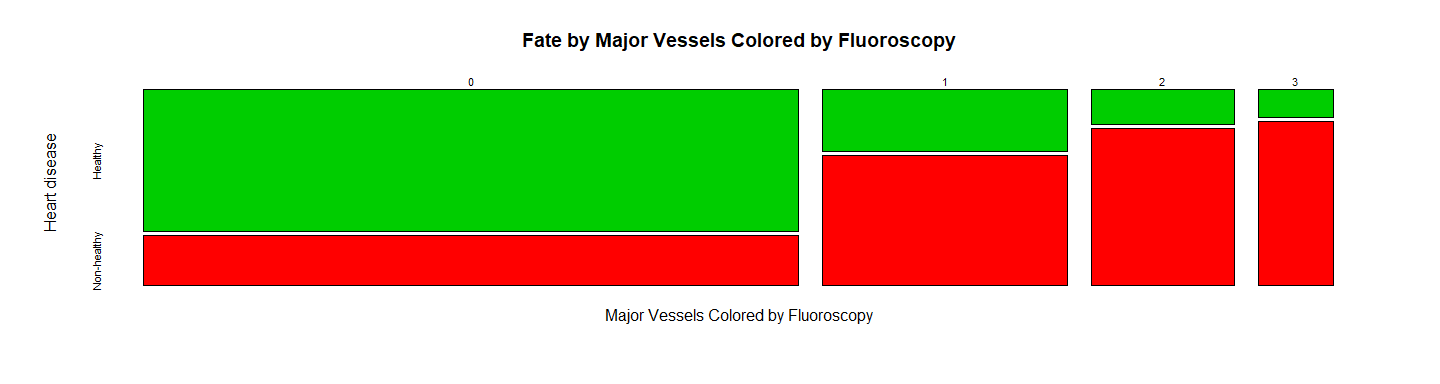
**mosaicplot**(heart.factor.names**$**Oldpeak **~** heart.factor.names**$**Disease,  
 main="Fate by ST Depression Induced by Exercise", shade=FALSE,color=**c**(3,2),  
 xlab="ST Depression Induced by Exercise", ylab="Heart disease")



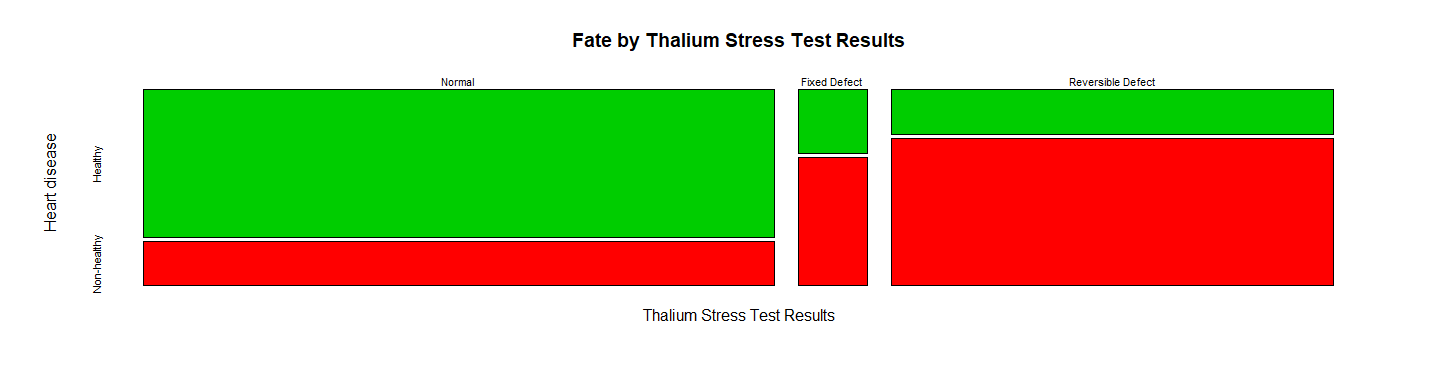
**mosaicplot**(heart.factor.names**$**Slope **~** heart.factor.names**$**Disease,  
 main="Fate by Slope of Peak Exercise ST Segment", shade=FALSE,color=**c**(3,2),  
 xlab="Slope of Peak Exercise ST Segment", ylab="Heart disease")



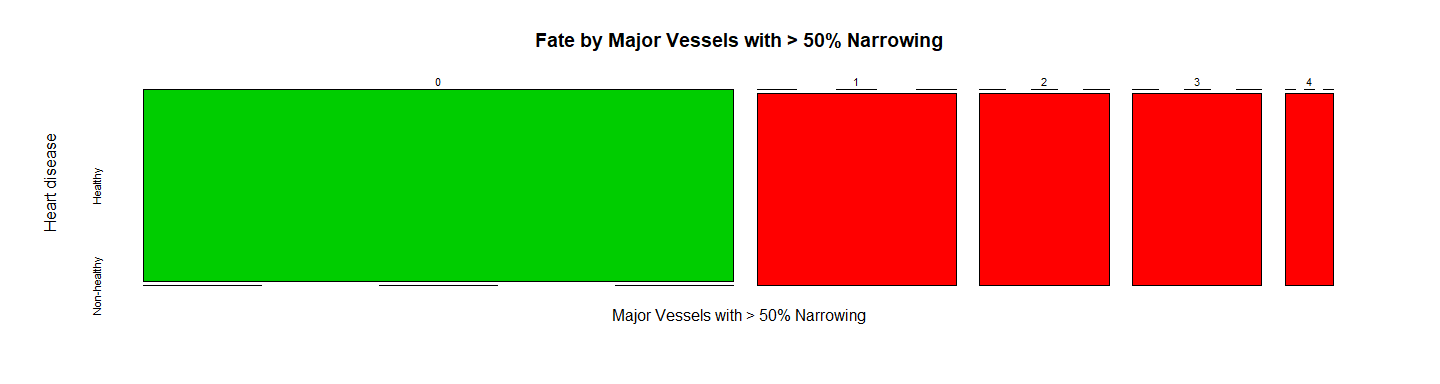
**mosaicplot**(heart.factor.names**$**Ca **~** heart.factor.names**$**Disease,  
 main="Fate by Major Vessels Colored by Fluoroscopy", shade=FALSE,color=**c**(3,2),  
 xlab="Major Vessels Colored by Fluoroscopy", ylab="Heart disease")



**mosaicplot**(heart.factor.names**$**Thal **~** heart.factor.names**$**Disease,  
 main="Fate by Thalium Stress Test Results", shade=FALSE,color=**c**(3,2),  
 xlab="Thalium Stress Test Results", ylab="Heart disease")



**mosaicplot**(heart.factor.names**$**Num **~** heart.factor.names**$**Disease,  
 main="Fate by Major Vessels with > 50% Narrowing", shade=FALSE,color=**c**(3,2),  
 xlab="Major Vessels with > 50% Narrowing", ylab="Heart disease")



# 2. Split the Dataset into Train and Data

We will split the dataset using 1/3 for test and 2/3 for training, for each Num type (0,1,2,3)

############################################  
*# 2. Split the Dataset into Train and Data*  
(n.var <- **length**(heart.data))

## [1] 15

*# The firsts 13 variables are the predictors and the 15 is the class*  
   
inTrainRows <- **createDataPartition**(heart.data**$**Num,p=2**/**3,list=FALSE)  
trainData <- heart.data[inTrainRows,]  
trainData.factor <- heart.factor[inTrainRows,]  
trainData.factor.names <- heart.factor.names[inTrainRows,]  
  
testData <- heart.data[**-**inTrainRows,]  
testData.factor <- heart.factor[**-**inTrainRows,]  
testData.factor.names <- heart.factor.names[**-**inTrainRows,]  
  
**nrow**(trainData)**/**(**nrow**(testData)**+nrow**(trainData)) *#checking whether really 2/3 -> OK*

## [1] 0.6667

*# Distribution of each class in the train and test set*  
**summary**(trainData.factor.names**$**Num)

## 0 1 2 3 4   
## 107 35 24 24 8

**summary**(testData.factor.names**$**Num)

## 0 1 2 3 4   
## 53 19 11 11 5

*# Create the Train Set*  
X.trainData <- trainData[,1**:**13]  
X.trainData.factor <- trainData.factor[,1**:**13]  
X.trainData.factor.names <- trainData.factor.names[,1**:**13]  
X.trainData.std <- **f.data.std**(X.trainData)  
Y.trainData <- trainData[,15]  
Y.trainData.factor.names <- trainData.factor.names[,15]  
**summary**(Y.trainData.factor.names)

## Healthy Non-healthy   
## 107 91

*# Create the Test Set*  
X.testData <- testData[,1**:**13]  
X.testData.factor <- testData.factor[,1**:**13]  
X.testData.factor.names <- testData.factor.names[,1**:**13]  
X.testData.std <- **f.data.std**(X.testData)  
Y.testData <- testData[,15]  
Y.testData.factor.names <- testData.factor.names[,15]  
**summary**(Y.testData.factor.names)

## Healthy Non-healthy   
## 53 46

# 3. Mining Association Rules (apriori)

############################################  
*# 3. Mining Association Rules (apriori)*  
data.file <- **paste**(data.dir, "allfactorsclev.csv", sep="/")  
*#this is a file where all the variables are discrete factors and every level of the factor*  
*# corresponds to something (shown in the document)*  
allfactors<-**read.csv**(data.file, header = TRUE)  
*#make all variables factors so we can do association mining*  
allfactors**$**Age.Range<-**as.factor**(allfactors**$**Age.Range)  
allfactors**$**Sex<-**as.factor**(allfactors**$**Sex)  
allfactors**$**CP<-**as.factor**(allfactors**$**CP)  
allfactors**$**Trestbps.range<-**as.factor**(allfactors**$**Trestbps.range)  
allfactors**$**Chol.range<-**as.factor**(allfactors**$**Chol.range)  
allfactors**$**Fbs<-**as.factor**(allfactors**$**Fbs)  
allfactors**$**Restecg<-**as.factor**(allfactors**$**Restecg)  
allfactors**$**Thalach.range<-**as.factor**(allfactors**$**Thalach.range)  
allfactors**$**Exang<-**as.factor**(allfactors**$**Exang)  
allfactors**$**Oldpeak.range<-**as.factor**(allfactors**$**Oldpeak.range)  
allfactors**$**Slope<-**as.factor**(allfactors**$**Slope)  
allfactors**$**Ca<-**as.factor**(allfactors**$**Ca)  
allfactors**$**Thal<-**as.factor**(allfactors**$**Thal)  
allfactors**$**Num<-**as.factor**(allfactors**$**Num)  
*#partition the classification factor so we can see the distinction*  
allfactors**$**Disease <- **ifelse**((allfactors**$**Num **!=** 0),1,0)  
allfactors**$**Disease <- **factor**(allfactors**$**Disease)  
allfactors**$**Num<-NULL  
  
*#create rules for each subset*  
  
rules<-**apriori**((allfactors), appearance=**list**(rhs= **c**("Disease=0"), default="lhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 30   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[44 item(s), 303 transaction(s)] done [0.00s].  
## sorting and recoding items ... [34 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].  
## writing ... [859 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

**summary**(rules)

## set of 859 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5 6 7 8   
## 1 43 200 314 222 71 8   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 4.00 5.00 5.12 6.00 8.00   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.102 Min. :0.800 Min. :1.48 Min. : 31.0   
## 1st Qu.:0.112 1st Qu.:0.854 1st Qu.:1.58 1st Qu.: 34.0   
## Median :0.129 Median :0.896 Median :1.66 Median : 39.0   
## Mean :0.144 Mean :0.895 Mean :1.65 Mean : 43.6   
## 3rd Qu.:0.160 3rd Qu.:0.939 3rd Qu.:1.73 3rd Qu.: 48.5   
## Max. :0.406 Max. :1.000 Max. :1.85 Max. :123.0   
##   
## mining info:  
## data ntransactions support confidence  
## (allfactors) 303 0.1 0.8

*#sort by count/support*  
rules\_count<-**sort**(rules, by='count', decreasing = T)  
**inspect**(rules\_count[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.4059 0.8146 1.505 123  
## [2] {Exang=0,   
## Thal=3} => {Disease=0} 0.3729 0.8370 1.546 113  
## [3] {Exang=0,   
## Ca=0} => {Disease=0} 0.3696 0.8485 1.568 112  
## [4] {Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.3564 0.8640 1.596 108  
## [5] {Fbs=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.3564 0.8182 1.512 108  
## [6] {Exang=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.3465 0.8607 1.590 105  
## [7] {Ca=0,   
## Thal=3} => {Disease=0} 0.3366 0.8870 1.639 102  
## [8] {Fbs=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.3267 0.8390 1.550 99  
## [9] {Fbs=0,   
## Exang=0,   
## Ca=0} => {Disease=0} 0.3234 0.8305 1.534 98  
## [10] {Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.3201 0.9065 1.675 97  
## [11] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.3135 0.8716 1.610 95  
## [12] {Exang=0,   
## Slope=1} => {Disease=0} 0.3069 0.8017 1.481 93  
## [13] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.3069 0.8455 1.562 93  
## [14] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1} => {Disease=0} 0.3036 0.8070 1.491 92  
## [15] {Fbs=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.3036 0.8762 1.619 92  
## [16] {Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2937 0.9082 1.678 89  
## [17] {Slope=1,   
## Thal=3} => {Disease=0} 0.2904 0.8627 1.594 88  
## [18] {Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2904 0.8627 1.594 88  
## [19] {Fbs=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2904 0.8980 1.659 88  
## [20] {Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2805 0.9239 1.707 85  
## [21] {Restecg=0,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.2673 0.8182 1.512 81  
## [22] {Slope=1,   
## Ca=0} => {Disease=0} 0.2640 0.8696 1.607 80  
## [23] {Fbs=0,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2640 0.8989 1.661 80  
## [24] {Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2607 0.8681 1.604 79  
## [25] {Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.2607 0.8681 1.604 79  
## [26] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2607 0.8681 1.604 79  
## [27] {Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.2541 0.8021 1.482 77  
## [28] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2541 0.9167 1.694 77  
## [29] {Fbs=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2508 0.8539 1.578 76  
## [30] {Fbs=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2508 0.8539 1.578 76  
## [31] {Thalach.range=3,   
## Ca=0} => {Disease=0} 0.2475 0.9036 1.669 75  
## [32] {Thalach.range=3,   
## Thal=3} => {Disease=0} 0.2442 0.8605 1.590 74  
## [33] {Restecg=0,   
## Thal=3} => {Disease=0} 0.2409 0.8588 1.587 73  
## [34] {Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.2376 0.9114 1.684 72  
## [35] {Thalach.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2343 0.8659 1.600 71  
## [36] {Exang=0,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.2343 0.8875 1.640 71  
## [37] {Fbs=0,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.2343 0.8554 1.580 71  
## [38] {Sex=0,   
## Oldpeak.range=1} => {Disease=0} 0.2310 0.8140 1.504 70  
## [39] {Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.2310 0.8140 1.504 70  
## [40] {Restecg=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2310 0.8750 1.617 70  
## [41] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.2310 0.8861 1.637 70  
## [42] {Fbs=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.2310 0.8537 1.577 70  
## [43] {Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.2310 0.8140 1.504 70  
## [44] {Sex=0,   
## Thal=3} => {Disease=0} 0.2277 0.8625 1.594 69  
## [45] {Thalach.range=3,   
## Slope=1} => {Disease=0} 0.2244 0.8000 1.478 68  
## [46] {Restecg=0,   
## Slope=1} => {Disease=0} 0.2244 0.8293 1.532 68  
## [47] {Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1} => {Disease=0} 0.2244 0.8095 1.496 68  
## [48] {Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.2244 0.8718 1.611 68  
## [49] {Restecg=0,   
## Oldpeak.range=1,   
## Slope=1} => {Disease=0} 0.2244 0.8293 1.532 68  
## [50] {Fbs=0,   
## Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2244 0.8608 1.590 68

*#sort by lift*  
rules\_lift<-**sort**(rules, by='lift', decreasing = T)  
**inspect**(rules\_lift[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Sex=0,   
## CP=3,   
## Thal=3} => {Disease=0} 0.1023 1.0000 1.848 31  
## [2] {CP=3,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1056 1.0000 1.848 32  
## [3] {Age.Range=2,   
## Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.1386 1.0000 1.848 42  
## [4] {Sex=0,   
## CP=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1023 1.0000 1.848 31  
## [5] {Age.Range=2,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1221 1.0000 1.848 37  
## [6] {Age.Range=2,   
## Trestbps.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1188 1.0000 1.848 36  
## [7] {Age.Range=2,   
## Trestbps.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1254 1.0000 1.848 38  
## [8] {Age.Range=2,   
## Trestbps.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1320 1.0000 1.848 40  
## [9] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Thal=3} => {Disease=0} 0.1221 1.0000 1.848 37  
## [10] {Restecg=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1287 1.0000 1.848 39  
## [11] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1155 1.0000 1.848 35  
## [12] {Age.Range=2,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1089 1.0000 1.848 33  
## [13] {Age.Range=2,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1155 1.0000 1.848 35  
## [14] {Age.Range=2,   
## Fbs=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1155 1.0000 1.848 35  
## [15] {Age.Range=2,   
## Trestbps.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1089 1.0000 1.848 33  
## [16] {Age.Range=2,   
## Trestbps.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1122 1.0000 1.848 34  
## [17] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1089 1.0000 1.848 33  
## [18] {Age.Range=2,   
## Trestbps.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1188 1.0000 1.848 36  
## [19] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1089 1.0000 1.848 33  
## [20] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1188 1.0000 1.848 36  
## [21] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1056 1.0000 1.848 32  
## [22] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1188 1.0000 1.848 36  
## [23] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1254 1.0000 1.848 38  
## [24] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1221 1.0000 1.848 37  
## [25] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1089 1.0000 1.848 33  
## [26] {Age.Range=2,   
## Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1023 1.0000 1.848 31  
## [27] {Age.Range=2,   
## Fbs=0,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1023 1.0000 1.848 31  
## [28] {Age.Range=2,   
## Fbs=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1122 1.0000 1.848 34  
## [29] {Age.Range=2,   
## Trestbps.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1023 1.0000 1.848 31  
## [30] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1056 1.0000 1.848 32  
## [31] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1056 1.0000 1.848 32  
## [32] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1056 1.0000 1.848 32  
## [33] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1155 1.0000 1.848 35  
## [34] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1122 1.0000 1.848 34  
## [35] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1188 1.0000 1.848 36  
## [36] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1089 1.0000 1.848 33  
## [37] {Age.Range=2,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1716 0.9811 1.813 52  
## [38] {Age.Range=2,   
## Fbs=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1650 0.9804 1.811 50  
## [39] {Age.Range=2,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1485 0.9783 1.807 45  
## [40] {Age.Range=2,   
## Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1419 0.9773 1.806 43  
## [41] {Trestbps.range=3,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1254 0.9744 1.800 38  
## [42] {Trestbps.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1254 0.9744 1.800 38  
## [43] {Sex=0,   
## Restecg=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1221 0.9737 1.799 37  
## [44] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0} => {Disease=0} 0.1188 0.9730 1.798 36  
## [45] {Trestbps.range=3,   
## Exang=0,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1188 0.9730 1.798 36  
## [46] {Trestbps.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1188 0.9730 1.798 36  
## [47] {Age.Range=2,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1155 0.9722 1.796 35  
## [48] {Sex=0,   
## Restecg=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1155 0.9722 1.796 35  
## [49] {Age.Range=2,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1155 0.9722 1.796 35  
## [50] {Sex=0,   
## CP=3} => {Disease=0} 0.1122 0.9714 1.795 34

rules1<-**apriori**((allfactors), appearance=**list**(rhs= **c**("Disease=1"), default="lhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 30   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[44 item(s), 303 transaction(s)] done [0.00s].  
## sorting and recoding items ... [34 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].  
## writing ... [174 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules\_1count <- **sort**(rules1, by='count', decreasing = TRUE)  
rules\_1lift <- **sort**(rules1, by='lift', decreasing = TRUE)  
**summary**(rules1)

## set of 174 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5 6   
## 2 24 79 65 4   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 4.00 4.00 4.26 5.00 6.00   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.102 Min. :0.800 Min. :1.74 Min. :31.0   
## 1st Qu.:0.109 1st Qu.:0.834 1st Qu.:1.82 1st Qu.:33.0   
## Median :0.122 Median :0.876 Median :1.91 Median :37.0   
## Mean :0.129 Mean :0.879 Mean :1.92 Mean :39.1   
## 3rd Qu.:0.139 3rd Qu.:0.914 3rd Qu.:1.99 3rd Qu.:42.0   
## Max. :0.234 Max. :1.000 Max. :2.18 Max. :71.0   
##   
## mining info:  
## data ntransactions support confidence  
## (allfactors) 303 0.1 0.8

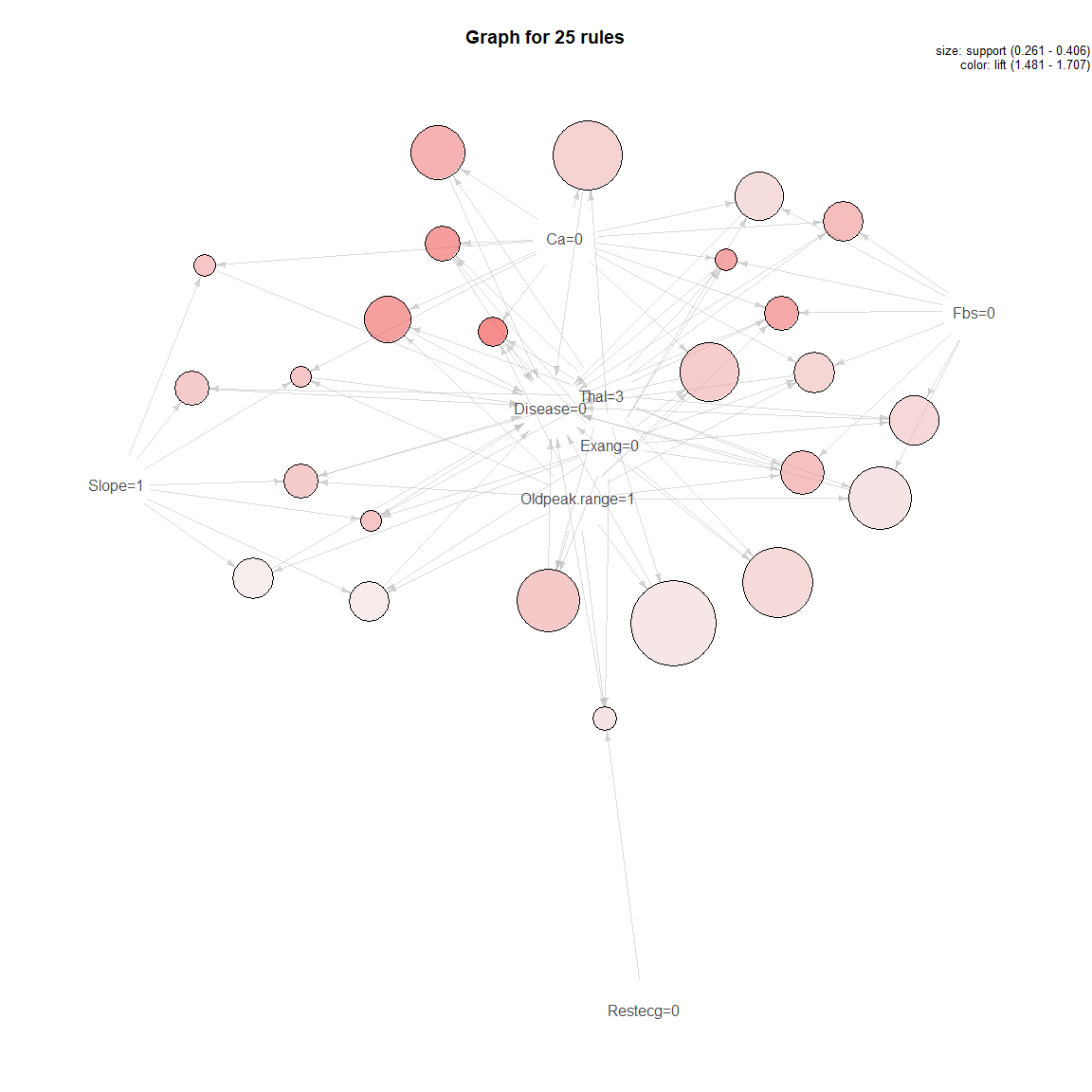
**inspect**(rules\_1count[1**:**50])

## lhs rhs support confidence  
## [1] {CP=4,Thal=7} => {Disease=1} 0.2343 0.9103   
## [2] {CP=4,Exang=1} => {Disease=1} 0.2310 0.8750   
## [3] {CP=4,Slope=2} => {Disease=1} 0.2244 0.8095   
## [4] {Age.Range=3,CP=4} => {Disease=1} 0.2112 0.8000   
## [5] {Sex=1,Exang=1} => {Disease=1} 0.2046 0.8052   
## [6] {Slope=2,Thal=7} => {Disease=1} 0.1980 0.8571   
## [7] {Sex=1,CP=4,Thal=7} => {Disease=1} 0.1947 0.8939   
## [8] {CP=4,Fbs=0,Thal=7} => {Disease=1} 0.1947 0.9077   
## [9] {CP=4,Fbs=0,Exang=1} => {Disease=1} 0.1914 0.8529   
## [10] {Thalach.range=2,Thal=7} => {Disease=1} 0.1848 0.8116   
## [11] {Sex=1,CP=4,Exang=1} => {Disease=1} 0.1848 0.9032   
## [12] {Exang=1,Slope=2} => {Disease=1} 0.1815 0.8594   
## [13] {Exang=1,Thal=7} => {Disease=1} 0.1749 0.8983   
## [14] {Fbs=0,Slope=2,Thal=7} => {Disease=1} 0.1749 0.8548   
## [15] {Chol.range=3,Thal=7} => {Disease=1} 0.1683 0.8226   
## [16] {Sex=1,CP=4,Slope=2} => {Disease=1} 0.1683 0.8793   
## [17] {Sex=1,CP=4,Restecg=2} => {Disease=1} 0.1650 0.8621   
## [18] {Sex=1,CP=4,Fbs=0,Thal=7} => {Disease=1} 0.1650 0.8929   
## [19] {Restecg=2,Thal=7} => {Disease=1} 0.1617 0.8448   
## [20] {CP=4,Exang=1,Thal=7} => {Disease=1} 0.1617 0.9423   
## [21] {CP=4,Exang=1,Slope=2} => {Disease=1} 0.1617 0.9074   
## [22] {Sex=1,Slope=2,Thal=7} => {Disease=1} 0.1617 0.8448   
## [23] {Chol.range=3,Exang=1} => {Disease=1} 0.1584 0.8276   
## [24] {Sex=1,CP=4,Thalach.range=2} => {Disease=1} 0.1584 0.8571   
## [25] {Sex=1,CP=4,Fbs=0,Exang=1} => {Disease=1} 0.1584 0.8889   
## [26] {Fbs=0,Exang=1,Slope=2} => {Disease=1} 0.1551 0.8545   
## [27] {CP=4,Thalach.range=2,Thal=7} => {Disease=1} 0.1551 0.9592   
## [28] {Fbs=0,Thalach.range=2,Thal=7} => {Disease=1} 0.1551 0.8103   
## [29] {CP=4,Thalach.range=2,Exang=1} => {Disease=1} 0.1518 0.9020   
## [30] {CP=4,Slope=2,Thal=7} => {Disease=1} 0.1518 0.9583   
## [31] {Sex=1,CP=4,Chol.range=3} => {Disease=1} 0.1518 0.8679   
## [32] {Age.Range=3,Exang=1} => {Disease=1} 0.1485 0.8182   
## [33] {Sex=1,Exang=1,Thal=7} => {Disease=1} 0.1485 0.8824   
## [34] {Age.Range=3,Sex=1,CP=4} => {Disease=1} 0.1485 0.8491   
## [35] {Sex=1,Fbs=0,Slope=2,Thal=7} => {Disease=1} 0.1452 0.8462   
## [36] {Sex=1,CP=4,Fbs=0,Slope=2} => {Disease=1} 0.1452 0.8627   
## [37] {Oldpeak.range=2} => {Disease=1} 0.1419 0.8431   
## [38] {Fbs=0,Exang=1,Thal=7} => {Disease=1} 0.1419 0.8776   
## [39] {Age.Range=3,CP=4,Exang=1} => {Disease=1} 0.1419 0.8958   
## [40] {CP=4,Chol.range=3,Exang=1} => {Disease=1} 0.1419 0.8958   
## [41] {Age.Range=3,CP=4,Thal=7} => {Disease=1} 0.1419 0.9149   
## [42] {Age.Range=3,CP=4,Slope=2} => {Disease=1} 0.1419 0.8431   
## [43] {CP=4,Fbs=0,Exang=1,Slope=2} => {Disease=1} 0.1419 0.8958   
## [44] {Sex=1,Exang=1,Slope=2} => {Disease=1} 0.1386 0.8571   
## [45] {CP=4,Oldpeak.range=1,Thal=7} => {Disease=1} 0.1386 0.8571   
## [46] {Sex=1,Chol.range=3,Thal=7} => {Disease=1} 0.1386 0.8235   
## [47] {Chol.range=3,Fbs=0,Thal=7} => {Disease=1} 0.1386 0.8235   
## [48] {CP=4,Chol.range=3,Slope=2} => {Disease=1} 0.1386 0.8235   
## [49] {Sex=1,CP=4,Fbs=0,Restecg=2} => {Disease=1} 0.1386 0.8400   
## [50] {Sex=1,Ca=1} => {Disease=1} 0.1353 0.8200   
## lift count  
## [1] 1.984 71   
## [2] 1.907 70   
## [3] 1.765 68   
## [4] 1.744 64   
## [5] 1.755 62   
## [6] 1.868 60   
## [7] 1.949 59   
## [8] 1.979 59   
## [9] 1.859 58   
## [10] 1.769 56   
## [11] 1.969 56   
## [12] 1.873 55   
## [13] 1.958 53   
## [14] 1.863 53   
## [15] 1.793 51   
## [16] 1.917 51   
## [17] 1.879 50   
## [18] 1.946 50   
## [19] 1.842 49   
## [20] 2.054 49   
## [21] 1.978 49   
## [22] 1.842 49   
## [23] 1.804 48   
## [24] 1.868 48   
## [25] 1.938 48   
## [26] 1.863 47   
## [27] 2.091 47   
## [28] 1.766 47   
## [29] 1.966 46   
## [30] 2.089 46   
## [31] 1.892 46   
## [32] 1.784 45   
## [33] 1.923 45   
## [34] 1.851 45   
## [35] 1.844 44   
## [36] 1.881 44   
## [37] 1.838 43   
## [38] 1.913 43   
## [39] 1.953 43   
## [40] 1.953 43   
## [41] 1.994 43   
## [42] 1.838 43   
## [43] 1.953 43   
## [44] 1.868 42   
## [45] 1.868 42   
## [46] 1.795 42   
## [47] 1.795 42   
## [48] 1.795 42   
## [49] 1.831 42   
## [50] 1.787 41

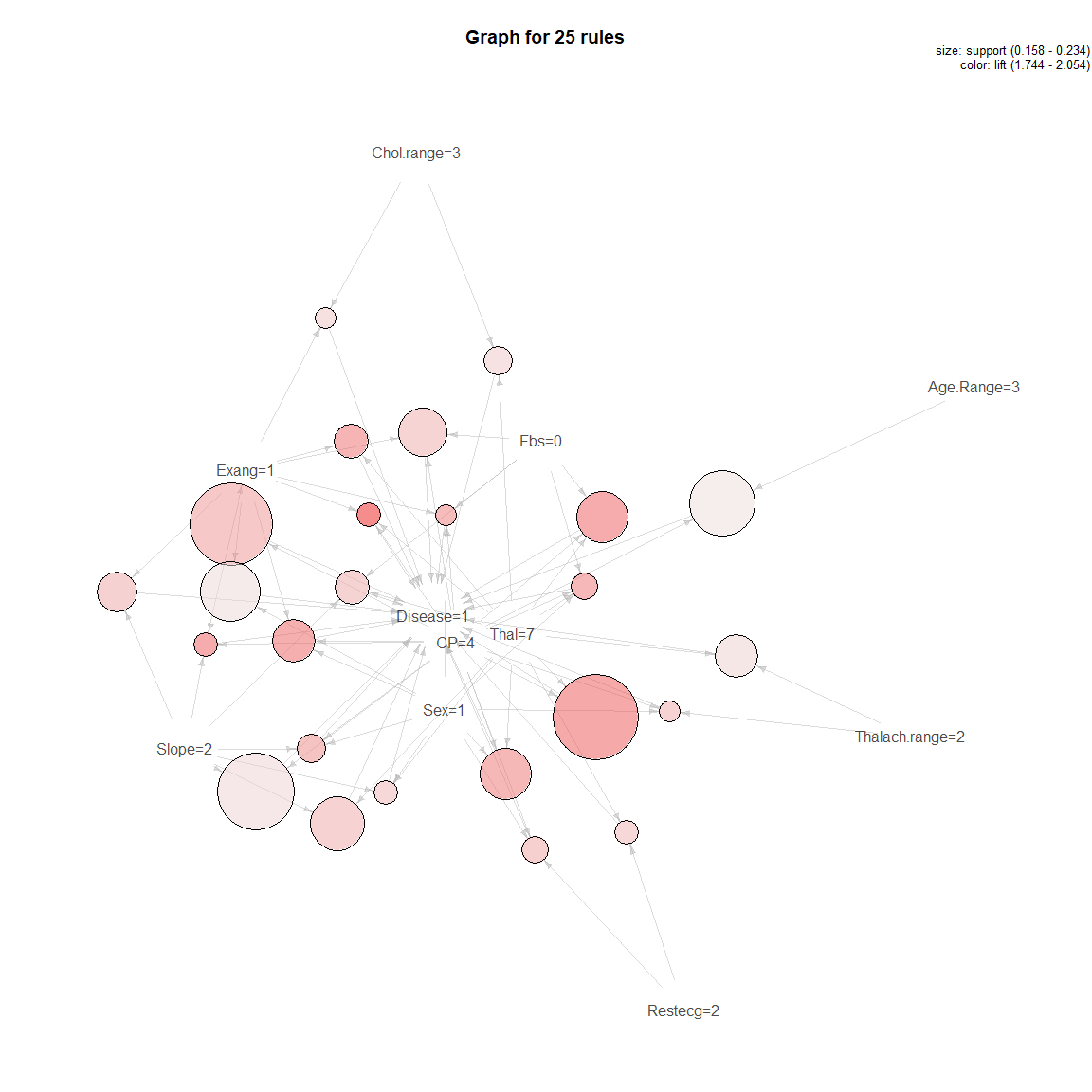
**inspect**(rules\_1lift[1**:**50])

## lhs rhs support confidence lift count  
## [1] {CP=4,   
## Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1155 1.0000 2.180 35  
## [2] {CP=4,   
## Thalach.range=2,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1089 1.0000 2.180 33  
## [3] {CP=4,   
## Restecg=2,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1056 1.0000 2.180 32  
## [4] {CP=4,   
## Chol.range=3,   
## Thal=7} => {Disease=1} 0.1353 0.9762 2.128 41  
## [5] {CP=4,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.1320 0.9756 2.127 40  
## [6] {CP=4,   
## Fbs=0,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1287 0.9750 2.125 39  
## [7] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Thal=7} => {Disease=1} 0.1089 0.9706 2.116 33  
## [8] {CP=4,   
## Chol.range=3,   
## Fbs=0,   
## Thal=7} => {Disease=1} 0.1089 0.9706 2.116 33  
## [9] {CP=4,   
## Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1056 0.9697 2.114 32  
## [10] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.1056 0.9697 2.114 32  
## [11] {CP=4,   
## Fbs=0,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.1056 0.9697 2.114 32  
## [12] {Sex=1,   
## CP=4,   
## Ca=1} => {Disease=1} 0.1023 0.9688 2.112 31  
## [13] {CP=4,   
## Trestbps.range=4,   
## Thal=7} => {Disease=1} 0.1023 0.9688 2.112 31  
## [14] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1023 0.9688 2.112 31  
## [15] {CP=4,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1551 0.9592 2.091 47  
## [16] {CP=4,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1518 0.9583 2.089 46  
## [17] {CP=4,   
## Fbs=0,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1320 0.9524 2.076 40  
## [18] {Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1221 0.9487 2.068 37  
## [19] {Sex=1,   
## CP=4,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1221 0.9487 2.068 37  
## [20] {Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1188 0.9474 2.065 36  
## [21] {Sex=1,   
## CP=4,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1188 0.9474 2.065 36  
## [22] {Sex=1,   
## CP=4,   
## Thalach.range=2,   
## Exang=1} => {Disease=1} 0.1155 0.9459 2.062 35  
## [23] {Chol.range=3,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1089 0.9429 2.055 33  
## [24] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Exang=1} => {Disease=1} 0.1089 0.9429 2.055 33  
## [25] {CP=4,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1617 0.9423 2.054 49  
## [26] {CP=4,   
## Ca=1} => {Disease=1} 0.1056 0.9412 2.052 32  
## [27] {Fbs=0,   
## Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1056 0.9412 2.052 32  
## [28] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Slope=2} => {Disease=1} 0.1056 0.9412 2.052 32  
## [29] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1056 0.9412 2.052 32  
## [30] {CP=4,   
## Fbs=0,   
## Ca=1} => {Disease=1} 0.1023 0.9394 2.048 31  
## [31] {Age.Range=3,   
## CP=4,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1023 0.9394 2.048 31  
## [32] {CP=4,   
## Chol.range=3,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1023 0.9394 2.048 31  
## [33] {Sex=1,   
## CP=4,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1353 0.9318 2.031 41  
## [34] {CP=4,   
## Fbs=0,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1320 0.9302 2.028 40  
## [35] {Sex=1,   
## CP=4,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1188 0.9231 2.012 36  
## [36] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Exang=1} => {Disease=1} 0.1155 0.9211 2.008 35  
## [37] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1155 0.9211 2.008 35  
## [38] {CP=4,   
## Oldpeak.range=2} => {Disease=1} 0.1089 0.9167 1.998 33  
## [39] {Age.Range=3,   
## Sex=1,   
## Ca=1} => {Disease=1} 0.1089 0.9167 1.998 33  
## [40] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1089 0.9167 1.998 33  
## [41] {Age.Range=3,   
## CP=4,   
## Thal=7} => {Disease=1} 0.1419 0.9149 1.994 43  
## [42] {Fbs=0,   
## Oldpeak.range=2,   
## Slope=2} => {Disease=1} 0.1056 0.9143 1.993 32  
## [43] {Age.Range=3,   
## Sex=1,   
## CP=4,   
## Exang=1} => {Disease=1} 0.1056 0.9143 1.993 32  
## [44] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Thalach.range=2} => {Disease=1} 0.1056 0.9143 1.993 32  
## [45] {Sex=1,   
## Oldpeak.range=2,   
## Slope=2} => {Disease=1} 0.1023 0.9118 1.988 31  
## [46] {Age.Range=3,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1023 0.9118 1.988 31  
## [47] {Chol.range=3,   
## Fbs=0,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1023 0.9118 1.988 31  
## [48] {Sex=1,   
## Chol.range=3,   
## Fbs=0,   
## Exang=1} => {Disease=1} 0.1023 0.9118 1.988 31  
## [49] {CP=4,   
## Thal=7} => {Disease=1} 0.2343 0.9103 1.984 71  
## [50] {Age.Range=3,   
## CP=4,   
## Restecg=2} => {Disease=1} 0.1320 0.9091 1.982 40

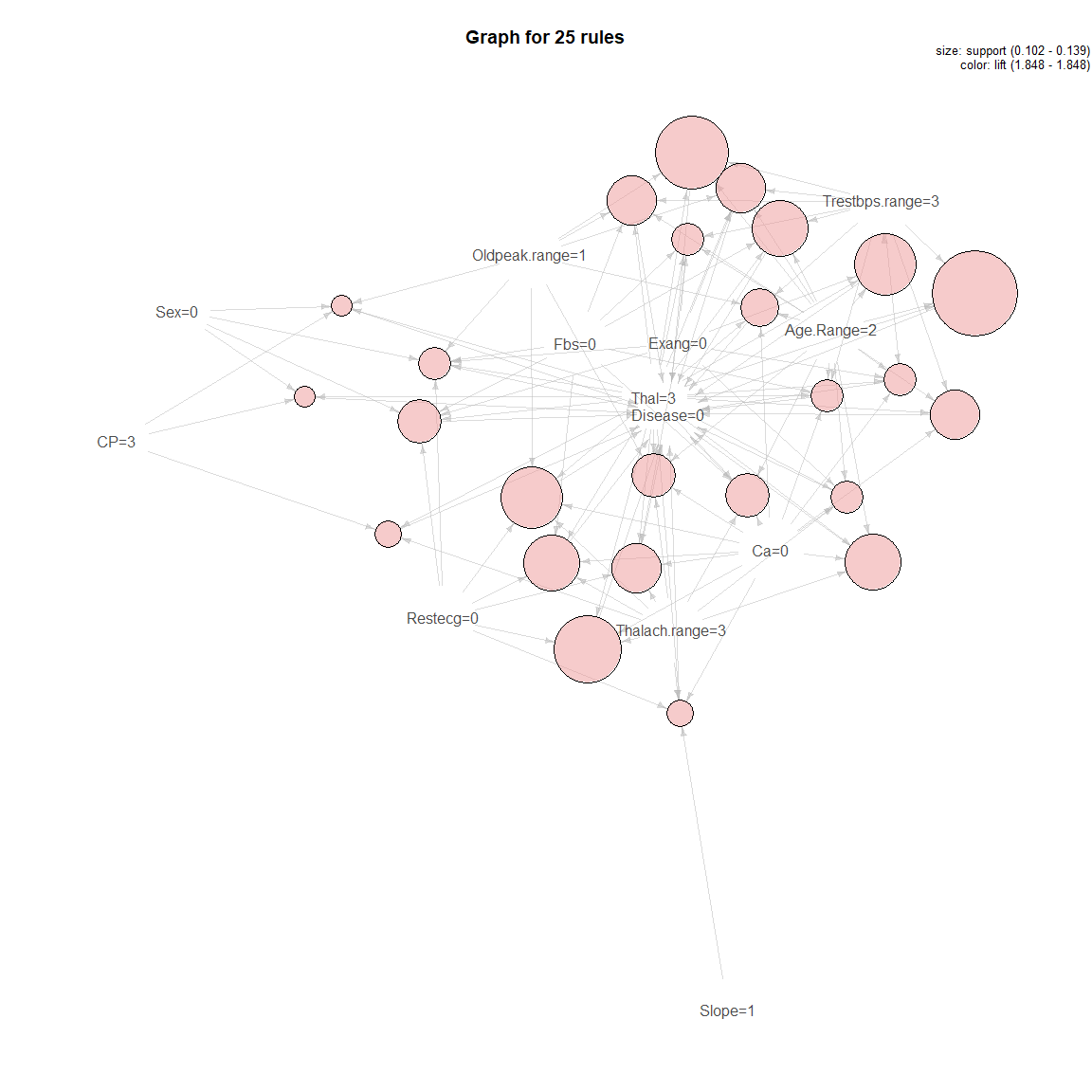
topRules\_count<-rules\_count[1**:**25]  
topRules\_1count<-rules\_1count[1**:**25]  
topRules\_lift<-rules\_lift[1**:**25]  
topRules\_1lift<-rules\_1lift[1**:**25]  
  
**plot**(topRules\_count, method="graph")



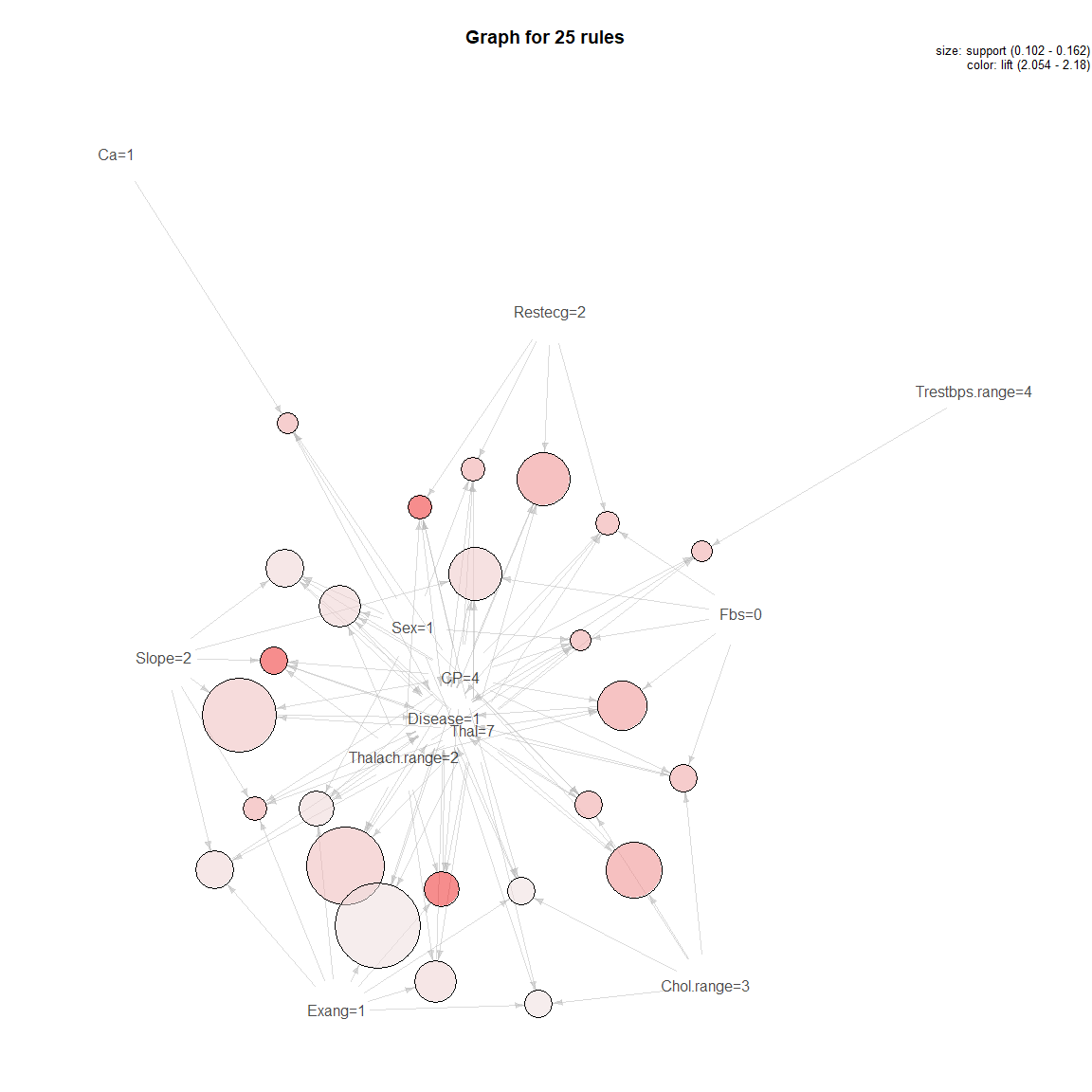
**plot**(topRules\_1count, method="graph")



**plot**(topRules\_lift, method="graph")



**plot**(topRules\_1lift, method="graph")



**plotly\_arules**(rules1)

*#Validating the results*  
*# Split the Dataset into Train and Data*  
(n.var <- **length**(allfactors))

## [1] 14

*# The firsts 13 variables are the predictors and the 15 is the class*  
inTrainRows\_m <- **createDataPartition**(allfactors**$**Disease,p=2**/**3,list=FALSE)  
trainData\_m <- allfactors[inTrainRows\_m,]  
testData\_m <- allfactors[**-**inTrainRows\_m,]  
  
*#For training data*  
rulestrain<-**apriori**((trainData\_m), appearance=**list**(rhs= **c**("Disease=0"), default="lhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 20   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[42 item(s), 203 transaction(s)] done [0.00s].  
## sorting and recoding items ... [33 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].  
## writing ... [1000 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules\_train\_count<-**sort**(rulestrain, by='count', decreasing = T)  
**inspect**(rules\_train\_count[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.3941 0.8000 1.476 80  
## [2] {Exang=0,   
## Ca=0} => {Disease=0} 0.3596 0.8588 1.585 73  
## [3] {Exang=0,   
## Thal=3} => {Disease=0} 0.3448 0.8434 1.556 70  
## [4] {Fbs=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.3399 0.8118 1.498 69  
## [5] {Exang=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.3399 0.8625 1.592 69  
## [6] {Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.3350 0.8608 1.588 68  
## [7] {Ca=0,   
## Thal=3} => {Disease=0} 0.3153 0.8889 1.640 64  
## [8] {Fbs=0,   
## Exang=0,   
## Ca=0} => {Disease=0} 0.3103 0.8400 1.550 63  
## [9] {Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.3054 0.8986 1.658 62  
## [10] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.3005 0.8472 1.564 61  
## [11] {Fbs=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.2956 0.8571 1.582 60  
## [12] {Restecg=0,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.2956 0.8333 1.538 60  
## [13] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2906 0.8806 1.625 59  
## [14] {Slope=1,   
## Thal=3} => {Disease=0} 0.2759 0.8485 1.566 56  
## [15] {Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2759 0.8485 1.566 56  
## [16] {Fbs=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2759 0.8750 1.615 56  
## [17] {Restecg=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.2709 0.8088 1.493 55  
## [18] {Fbs=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2709 0.8871 1.637 55  
## [19] {Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2660 0.9153 1.689 54  
## [20] {Restecg=0,   
## Thal=3} => {Disease=0} 0.2611 0.8833 1.630 53  
## [21] {Restecg=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2611 0.8833 1.630 53  
## [22] {Fbs=0,   
## Restecg=0,   
## Exang=0} => {Disease=0} 0.2611 0.8154 1.505 53  
## [23] {Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.2611 0.8413 1.553 53  
## [24] {Restecg=0,   
## Slope=1} => {Disease=0} 0.2562 0.8814 1.627 52  
## [25] {Slope=1,   
## Ca=0} => {Disease=0} 0.2562 0.9123 1.684 52  
## [26] {Restecg=0,   
## Oldpeak.range=1,   
## Slope=1} => {Disease=0} 0.2562 0.8814 1.627 52  
## [27] {Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2562 0.9286 1.714 52  
## [28] {Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.2512 0.9107 1.681 51  
## [29] {Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2463 0.8621 1.591 50  
## [30] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2463 0.8621 1.591 50  
## [31] {Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.2414 0.8448 1.559 49  
## [32] {Fbs=0,   
## Restecg=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.2414 0.8033 1.482 49  
## [33] {Thalach.range=3,   
## Ca=0} => {Disease=0} 0.2365 0.9057 1.671 48  
## [34] {Age.Range=2,   
## Ca=0} => {Disease=0} 0.2365 0.8136 1.501 48  
## [35] {Age.Range=2,   
## Exang=0} => {Disease=0} 0.2365 0.8136 1.501 48  
## [36] {Trestbps.range=3,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.2365 0.8276 1.527 48  
## [37] {Fbs=0,   
## Restecg=0,   
## Thal=3} => {Disease=0} 0.2365 0.8889 1.640 48  
## [38] {Fbs=0,   
## Restecg=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2365 0.8889 1.640 48  
## [39] {Sex=0,   
## Oldpeak.range=1} => {Disease=0} 0.2315 0.8103 1.495 47  
## [40] {Fbs=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2315 0.8393 1.549 47  
## [41] {Trestbps.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2315 0.8545 1.577 47  
## [42] {Restecg=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.2315 0.9400 1.735 47  
## [43] {Fbs=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.2315 0.8393 1.549 47  
## [44] {Restecg=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.2315 0.9400 1.735 47  
## [45] {Fbs=0,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.2315 0.9038 1.668 47  
## [46] {CP=3,   
## Oldpeak.range=1} => {Disease=0} 0.2266 0.8364 1.543 46  
## [47] {Thalach.range=3,   
## Thal=3} => {Disease=0} 0.2266 0.8679 1.602 46  
## [48] {Age.Range=2,   
## Thal=3} => {Disease=0} 0.2266 0.9020 1.665 46  
## [49] {Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.2266 0.9200 1.698 46  
## [50] {Age.Range=2,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.2266 0.8364 1.543 46

rules\_train\_lift<-**sort**(rulestrain, by='lift', decreasing = T)  
**inspect**(rules\_train\_lift[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Sex=0,   
## CP=3} => {Disease=0} 0.1084 1 1.845 22  
## [2] {Sex=0,   
## CP=3,   
## Exang=0} => {Disease=0} 0.1034 1 1.845 21  
## [3] {Sex=0,   
## CP=3,   
## Oldpeak.range=1} => {Disease=0} 0.1084 1 1.845 22  
## [4] {CP=3,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [5] {CP=3,   
## Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [6] {Chol.range=2,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1034 1 1.845 21  
## [7] {Chol.range=2,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [8] {Age.Range=2,   
## Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.1626 1 1.845 33  
## [9] {Sex=0,   
## CP=3,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.1034 1 1.845 21  
## [10] {CP=3,   
## Trestbps.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1034 1 1.845 21  
## [11] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0} => {Disease=0} 0.1133 1 1.845 23  
## [12] {Chol.range=2,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [13] {Age.Range=2,   
## Trestbps.range=3,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1182 1 1.845 24  
## [14] {Age.Range=2,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1232 1 1.845 25  
## [15] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1478 1 1.845 30  
## [16] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1330 1 1.845 27  
## [17] {Sex=1,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1133 1 1.845 23  
## [18] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1182 1 1.845 24  
## [19] {Restecg=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1281 1 1.845 26  
## [20] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1527 1 1.845 31  
## [21] {Age.Range=2,   
## Trestbps.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1034 1 1.845 21  
## [22] {Age.Range=2,   
## Trestbps.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1429 1 1.845 29  
## [23] {Age.Range=2,   
## Trestbps.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1478 1 1.845 30  
## [24] {Age.Range=2,   
## Trestbps.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1527 1 1.845 31  
## [25] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Thal=3} => {Disease=0} 0.1379 1 1.845 28  
## [26] {Trestbps.range=3,   
## Restecg=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1232 1 1.845 25  
## [27] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [28] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.1133 1 1.845 23  
## [29] {Age.Range=2,   
## Trestbps.range=3,   
## Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [30] {Age.Range=2,   
## Trestbps.range=3,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [31] {Age.Range=2,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [32] {Age.Range=2,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [33] {Age.Range=2,   
## Fbs=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [34] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1034 1 1.845 21  
## [35] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [36] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1330 1 1.845 27  
## [37] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1478 1 1.845 30  
## [38] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1379 1 1.845 28  
## [39] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1084 1 1.845 22  
## [40] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1330 1 1.845 27  
## [41] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1232 1 1.845 25  
## [42] {Sex=1,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1133 1 1.845 23  
## [43] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [44] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1182 1 1.845 24  
## [45] {Trestbps.range=3,   
## Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [46] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.1133 1 1.845 23  
## [47] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [48] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1281 1 1.845 26  
## [49] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1232 1 1.845 25  
## [50] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1527 1 1.845 31

**summary**(rulestrain)

## set of 1000 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 3 4 5 6 7 8   
## 45 205 359 283 98 10   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.00 4.75 5.00 5.21 6.00 8.00   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.103 Min. :0.800 Min. :1.48 Min. :21.0   
## 1st Qu.:0.113 1st Qu.:0.857 1st Qu.:1.58 1st Qu.:23.0   
## Median :0.128 Median :0.906 Median :1.67 Median :26.0   
## Mean :0.142 Mean :0.904 Mean :1.67 Mean :28.8   
## 3rd Qu.:0.158 3rd Qu.:0.955 3rd Qu.:1.76 3rd Qu.:32.0   
## Max. :0.394 Max. :1.000 Max. :1.85 Max. :80.0   
##   
## mining info:  
## data ntransactions support confidence  
## (trainData\_m) 203 0.1 0.8

rules1train<-**apriori**((trainData\_m), appearance=**list**(rhs= **c**("Disease=1"), default="lhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 20   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[42 item(s), 203 transaction(s)] done [0.00s].  
## sorting and recoding items ... [33 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].  
## writing ... [192 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules\_1train\_count <- **sort**(rules1train, by='count', decreasing = TRUE)  
**inspect**(rules\_1train\_count[1**:**50])

## lhs rhs support confidence  
## [1] {Sex=1,CP=4} => {Disease=1} 0.2562 0.8125   
## [2] {CP=4,Thal=7} => {Disease=1} 0.2315 0.8868   
## [3] {CP=4,Slope=2} => {Disease=1} 0.2315 0.8246   
## [4] {CP=4,Exang=1} => {Disease=1} 0.2167 0.8462   
## [5] {Slope=2,Thal=7} => {Disease=1} 0.2167 0.8462   
## [6] {Age.Range=3,CP=4} => {Disease=1} 0.2069 0.8077   
## [7] {CP=4,Chol.range=3} => {Disease=1} 0.2069 0.8235   
## [8] {Thalach.range=2,Thal=7} => {Disease=1} 0.2020 0.8367   
## [9] {CP=4,Restecg=2} => {Disease=1} 0.2020 0.8542   
## [10] {Exang=1,Slope=2} => {Disease=1} 0.1921 0.8478   
## [11] {CP=4,Fbs=0,Thal=7} => {Disease=1} 0.1921 0.8667   
## [12] {Sex=1,CP=4,Thal=7} => {Disease=1} 0.1872 0.8636   
## [13] {Fbs=0,Slope=2,Thal=7} => {Disease=1} 0.1872 0.8444   
## [14] {Sex=1,Slope=2,Thal=7} => {Disease=1} 0.1823 0.8409   
## [15] {Chol.range=3,Thal=7} => {Disease=1} 0.1773 0.8182   
## [16] {CP=4,Fbs=0,Exang=1} => {Disease=1} 0.1773 0.8182   
## [17] {Exang=1,Thal=7} => {Disease=1} 0.1675 0.8718   
## [18] {Restecg=2,Thal=7} => {Disease=1} 0.1675 0.8500   
## [19] {CP=4,Thalach.range=2,Exang=1} => {Disease=1} 0.1675 0.8947   
## [20] {Sex=1,CP=4,Exang=1} => {Disease=1} 0.1675 0.8947   
## [21] {CP=4,Thalach.range=2,Thal=7} => {Disease=1} 0.1675 0.9714   
## [22] {Sex=1,CP=4,Slope=2} => {Disease=1} 0.1675 0.8718   
## [23] {Sex=1,CP=4,Thalach.range=2} => {Disease=1} 0.1675 0.8718   
## [24] {Chol.range=3,Exang=1} => {Disease=1} 0.1626 0.8250   
## [25] {CP=4,Exang=1,Slope=2} => {Disease=1} 0.1626 0.8919   
## [26] {CP=4,Slope=2,Thal=7} => {Disease=1} 0.1626 0.9429   
## [27] {Sex=1,Thalach.range=2,Thal=7} => {Disease=1} 0.1626 0.8250   
## [28] {Fbs=0,Thalach.range=2,Thal=7} => {Disease=1} 0.1626 0.8049   
## [29] {CP=4,Fbs=0,Restecg=2} => {Disease=1} 0.1626 0.8250   
## [30] {Fbs=0,Exang=1,Slope=2} => {Disease=1} 0.1576 0.8421   
## [31] {CP=4,Chol.range=3,Slope=2} => {Disease=1} 0.1576 0.8889   
## [32] {CP=4,Thalach.range=2,Slope=2} => {Disease=1} 0.1576 0.8205   
## [33] {Sex=1,CP=4,Fbs=0,Thal=7} => {Disease=1} 0.1576 0.8421   
## [34] {Sex=1,Fbs=0,Slope=2,Thal=7} => {Disease=1} 0.1576 0.8421   
## [35] {Age.Range=3,CP=4,Thal=7} => {Disease=1} 0.1527 0.8857   
## [36] {Sex=1,CP=4,Restecg=2} => {Disease=1} 0.1527 0.9394   
## [37] {Sex=1,Chol.range=3,Slope=2} => {Disease=1} 0.1527 0.8158   
## [38] {Sex=1,Ca=1} => {Disease=1} 0.1478 0.8571   
## [39] {CP=4,Exang=1,Thal=7} => {Disease=1} 0.1478 0.9091   
## [40] {Sex=1,Exang=1,Slope=2} => {Disease=1} 0.1478 0.8571   
## [41] {Thalach.range=2,Slope=2,Thal=7} => {Disease=1} 0.1478 0.9091   
## [42] {Age.Range=3,CP=4,Slope=2} => {Disease=1} 0.1478 0.8571   
## [43] {Sex=1,CP=4,Chol.range=3} => {Disease=1} 0.1478 0.9375   
## [44] {Sex=1,Exang=1,Thal=7} => {Disease=1} 0.1429 0.8529   
## [45] {Thalach.range=2,Exang=1,Slope=2} => {Disease=1} 0.1429 0.8286   
## [46] {Sex=1,Thalach.range=2,Exang=1} => {Disease=1} 0.1429 0.8529   
## [47] {CP=4,Chol.range=3,Thal=7} => {Disease=1} 0.1429 0.9667   
## [48] {Sex=1,Chol.range=3,Thal=7} => {Disease=1} 0.1429 0.8286   
## [49] {CP=4,Restecg=2,Thalach.range=2} => {Disease=1} 0.1429 0.8286   
## [50] {Age.Range=3,CP=4,Thalach.range=2} => {Disease=1} 0.1429 0.8286   
## lift count  
## [1] 1.774 52   
## [2] 1.936 47   
## [3] 1.800 47   
## [4] 1.847 44   
## [5] 1.847 44   
## [6] 1.763 42   
## [7] 1.798 42   
## [8] 1.826 41   
## [9] 1.864 41   
## [10] 1.851 39   
## [11] 1.892 39   
## [12] 1.885 38   
## [13] 1.843 38   
## [14] 1.836 37   
## [15] 1.786 36   
## [16] 1.786 36   
## [17] 1.903 34   
## [18] 1.855 34   
## [19] 1.953 34   
## [20] 1.953 34   
## [21] 2.120 34   
## [22] 1.903 34   
## [23] 1.903 34   
## [24] 1.801 33   
## [25] 1.947 33   
## [26] 2.058 33   
## [27] 1.801 33   
## [28] 1.757 33   
## [29] 1.801 33   
## [30] 1.838 32   
## [31] 1.940 32   
## [32] 1.791 32   
## [33] 1.838 32   
## [34] 1.838 32   
## [35] 1.933 31   
## [36] 2.051 31   
## [37] 1.781 31   
## [38] 1.871 30   
## [39] 1.984 30   
## [40] 1.871 30   
## [41] 1.984 30   
## [42] 1.871 30   
## [43] 2.046 30   
## [44] 1.862 29   
## [45] 1.809 29   
## [46] 1.862 29   
## [47] 2.110 29   
## [48] 1.809 29   
## [49] 1.809 29   
## [50] 1.809 29

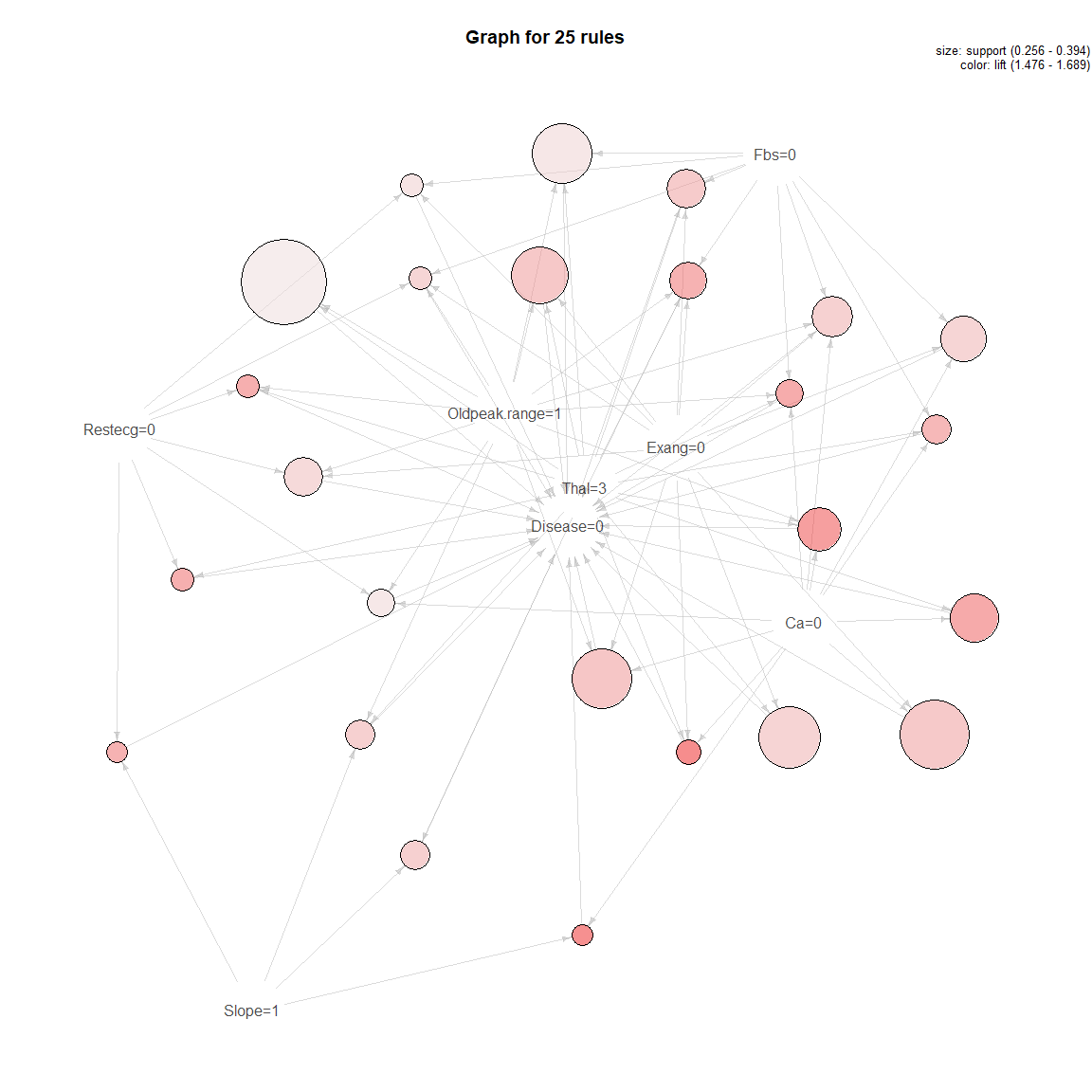
rules\_1train\_lift <- **sort**(rules1train, by='lift', decreasing = TRUE)  
**inspect**(rules\_1train\_lift[1**:**50])

## lhs rhs support confidence lift count  
## [1] {CP=4,   
## Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1232 1.0000 2.183 25  
## [2] {CP=4,   
## Restecg=2,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1133 1.0000 2.183 23  
## [3] {CP=4,   
## Thalach.range=2,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1182 1.0000 2.183 24  
## [4] {CP=4,   
## Chol.range=3,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1034 1.0000 2.183 21  
## [5] {CP=4,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1675 0.9714 2.120 34  
## [6] {CP=4,   
## Chol.range=3,   
## Thal=7} => {Disease=1} 0.1429 0.9667 2.110 29  
## [7] {CP=4,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.1379 0.9655 2.108 28  
## [8] {Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1330 0.9643 2.105 27  
## [9] {CP=4,   
## Fbs=0,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1330 0.9643 2.105 27  
## [10] {Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1281 0.9630 2.102 26  
## [11] {Age.Range=3,   
## CP=4,   
## Restecg=2} => {Disease=1} 0.1281 0.9630 2.102 26  
## [12] {Sex=1,   
## CP=4,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1281 0.9630 2.102 26  
## [13] {Sex=1,   
## CP=4,   
## Thalach.range=2,   
## Exang=1} => {Disease=1} 0.1232 0.9615 2.099 25  
## [14] {Chol.range=3,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1133 0.9583 2.092 23  
## [15] {Fbs=0,   
## Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1133 0.9583 2.092 23  
## [16] {CP=4,   
## Chol.range=3,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1133 0.9583 2.092 23  
## [17] {CP=4,   
## Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [18] {Sex=1,   
## Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [19] {Sex=1,   
## Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [20] {CP=4,   
## Fbs=0,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [21] {Age.Range=3,   
## CP=4,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [22] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [23] {CP=4,   
## Chol.range=3,   
## Fbs=0,   
## Thal=7} => {Disease=1} 0.1084 0.9565 2.088 22  
## [24] {Sex=1,   
## Slope=2,   
## Ca=1} => {Disease=1} 0.1034 0.9545 2.084 21  
## [25] {Fbs=0,   
## Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.1034 0.9545 2.084 21  
## [26] {Age.Range=3,   
## CP=4,   
## Fbs=0,   
## Restecg=2} => {Disease=1} 0.1034 0.9545 2.084 21  
## [27] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Thalach.range=2} => {Disease=1} 0.1034 0.9545 2.084 21  
## [28] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Oldpeak.range=1} => {Disease=1} 0.1034 0.9545 2.084 21  
## [29] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1034 0.9545 2.084 21  
## [30] {CP=4,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1626 0.9429 2.058 33  
## [31] {Sex=1,   
## CP=4,   
## Restecg=2} => {Disease=1} 0.1527 0.9394 2.051 31  
## [32] {Sex=1,   
## CP=4,   
## Chol.range=3} => {Disease=1} 0.1478 0.9375 2.046 30  
## [33] {CP=4,   
## Fbs=0,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1379 0.9333 2.037 28  
## [34] {Restecg=2,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.1281 0.9286 2.027 26  
## [35] {Sex=1,   
## CP=4,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1281 0.9286 2.027 26  
## [36] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Restecg=2} => {Disease=1} 0.1281 0.9286 2.027 26  
## [37] {Sex=1,   
## Chol.range=3,   
## Exang=1} => {Disease=1} 0.1232 0.9259 2.021 25  
## [38] {Oldpeak.range=2,   
## Slope=2} => {Disease=1} 0.1182 0.9231 2.015 24  
## [39] {Age.Range=3,   
## Sex=1,   
## Ca=1} => {Disease=1} 0.1182 0.9231 2.015 24  
## [40] {Sex=1,   
## CP=4,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1182 0.9231 2.015 24  
## [41] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Fbs=0} => {Disease=1} 0.1182 0.9231 2.015 24  
## [42] {CP=4,   
## Chol.range=3,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1133 0.9200 2.008 23  
## [43] {Chol.range=3,   
## Fbs=0,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.1133 0.9200 2.008 23  
## [44] {Fbs=0,   
## Oldpeak.range=2} => {Disease=1} 0.1084 0.9167 2.001 22  
## [45] {Sex=1,   
## Restecg=2,   
## Exang=1} => {Disease=1} 0.1084 0.9167 2.001 22  
## [46] {Chol.range=3,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.1084 0.9167 2.001 22  
## [47] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Slope=2} => {Disease=1} 0.1084 0.9167 2.001 22  
## [48] {Sex=1,   
## CP=4,   
## Thalach.range=2,   
## Slope=2} => {Disease=1} 0.1084 0.9167 2.001 22  
## [49] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Oldpeak.range=1} => {Disease=1} 0.1084 0.9167 2.001 22  
## [50] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.1084 0.9167 2.001 22

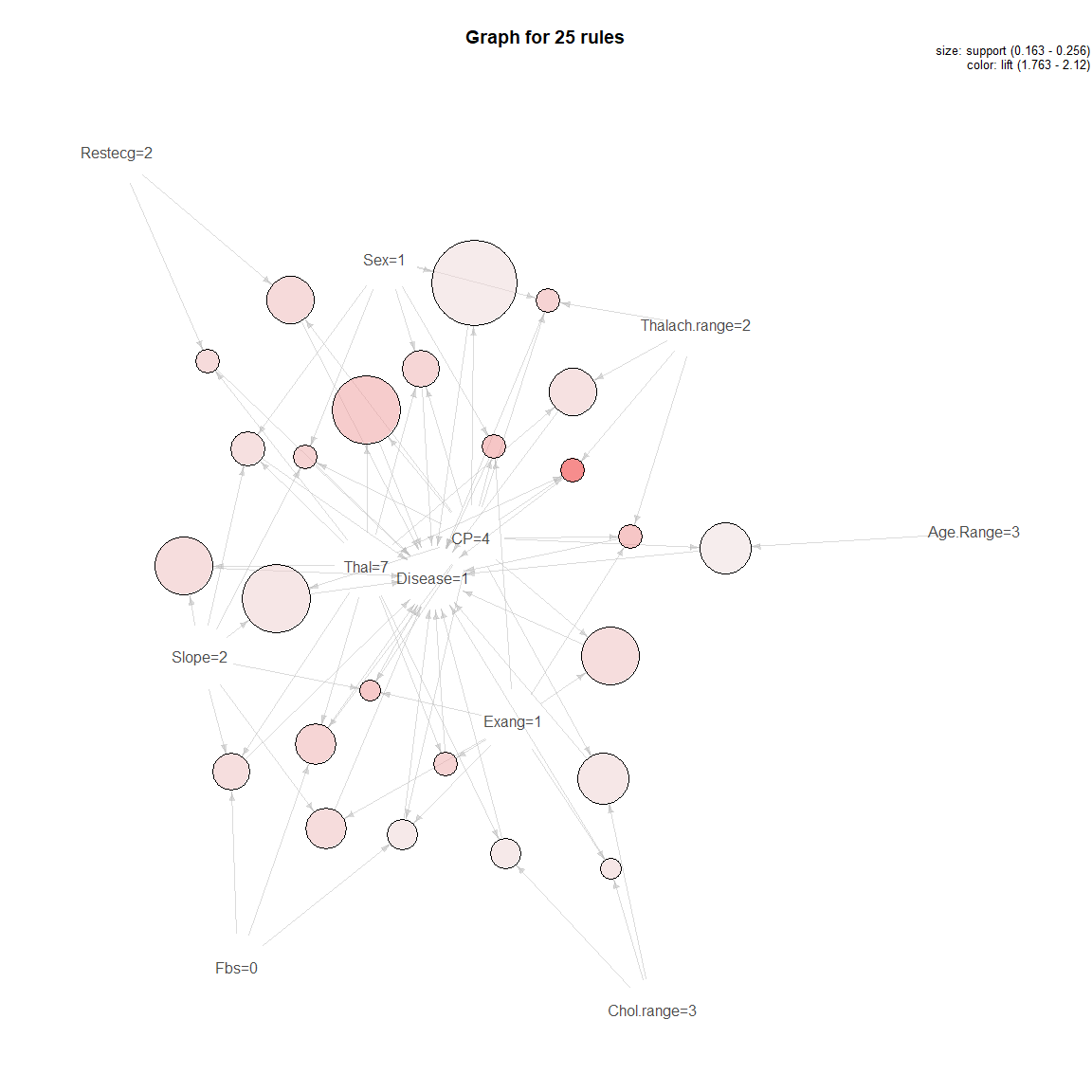
**summary**(rules1train)

## set of 192 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5 6   
## 1 22 90 74 5   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 4.00 4.00 4.31 5.00 6.00   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.103 Min. :0.800 Min. :1.75 Min. :21.0   
## 1st Qu.:0.108 1st Qu.:0.838 1st Qu.:1.83 1st Qu.:22.0   
## Median :0.123 Median :0.867 Median :1.89 Median :25.0   
## Mean :0.131 Mean :0.876 Mean :1.91 Mean :26.6   
## 3rd Qu.:0.143 3rd Qu.:0.917 3rd Qu.:2.00 3rd Qu.:29.0   
## Max. :0.256 Max. :1.000 Max. :2.18 Max. :52.0   
##   
## mining info:  
## data ntransactions support confidence  
## (trainData\_m) 203 0.1 0.8

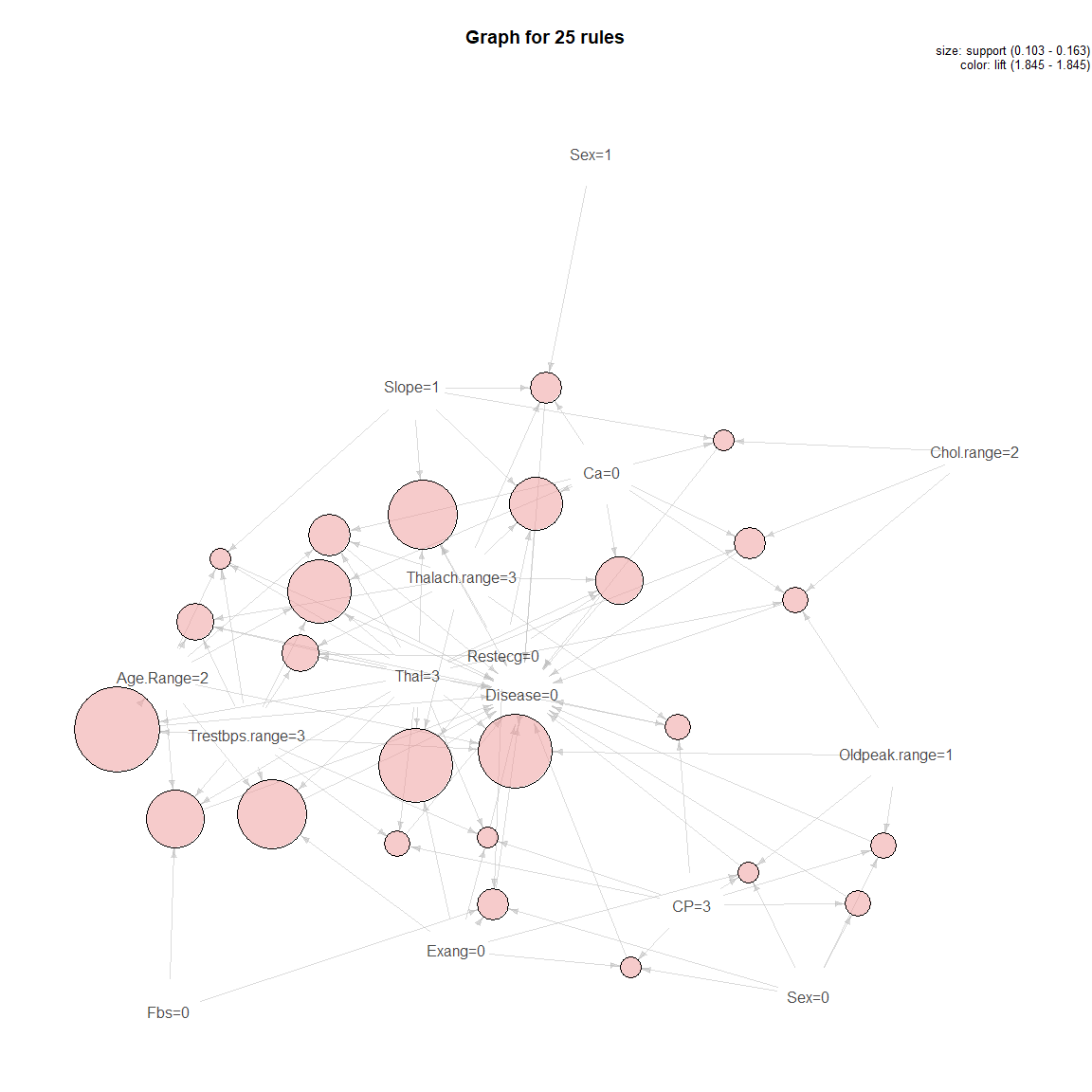
topRules\_train\_count<-rules\_train\_count[1**:**25]  
topRules\_1train\_count<-rules\_1train\_count[1**:**25]  
topRules\_train\_lift<-rules\_train\_lift[1**:**25]  
topRules\_1train\_lift<-rules\_1train\_lift[1**:**25]  
  
**plot**(topRules\_train\_count, method="graph")



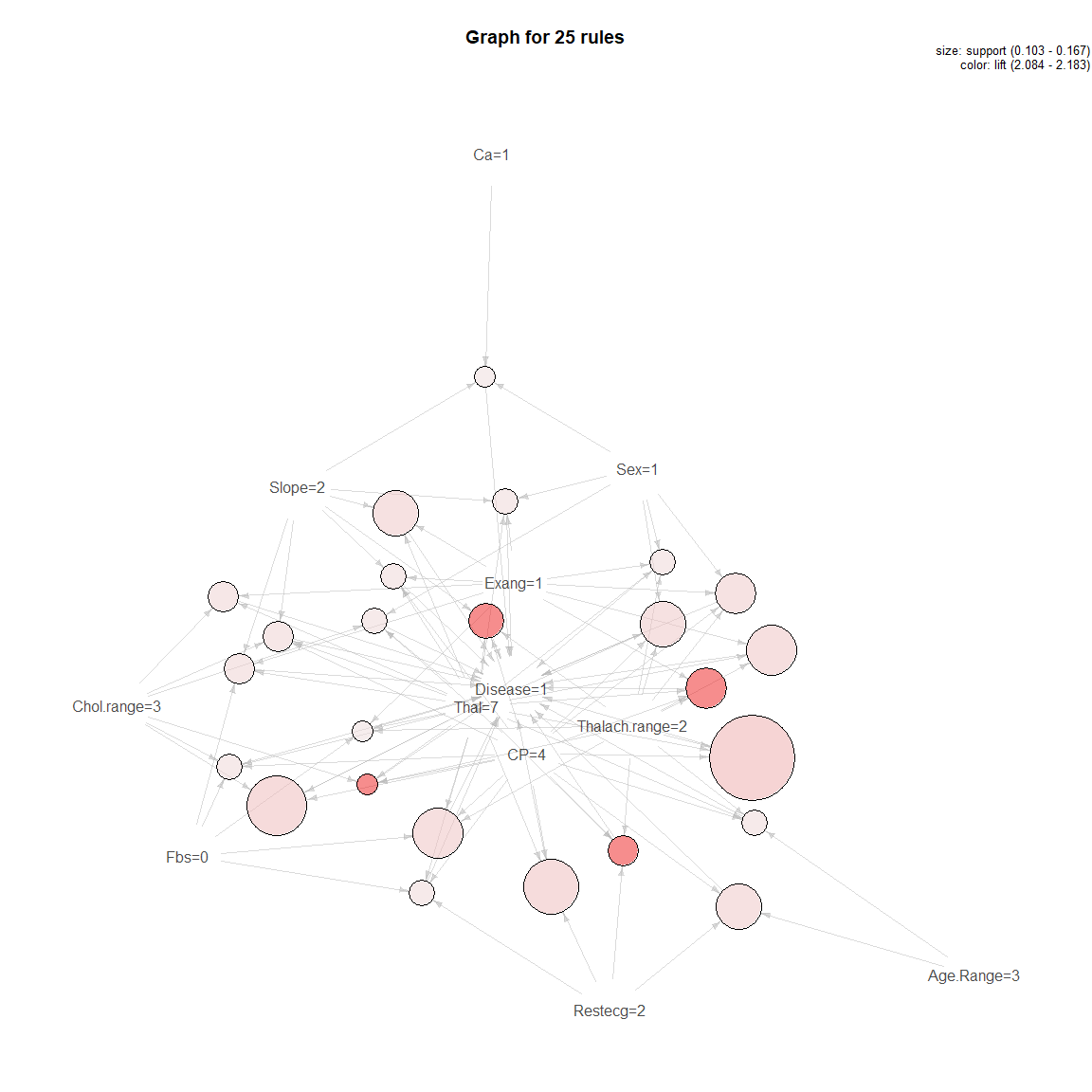
**plot**(topRules\_1train\_count, method="graph")



**plot**(topRules\_train\_lift, method="graph")



**plot**(topRules\_1train\_lift, method="graph")



*#For test Data*  
  
rulestest<-**apriori**((testData\_m), appearance=**list**(rhs= **c**("Disease=0"), default="lhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 10   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[43 item(s), 100 transaction(s)] done [0.00s].  
## sorting and recoding items ... [36 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 9 done [0.00s].  
## writing ... [1193 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules\_test\_count<-**sort**(rulestest, by='count', decreasing = T)  
**inspect**(rules\_test\_count[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Exang=0,   
## Thal=3} => {Disease=0} 0.43 0.8269 1.531 43  
## [2] {Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.43 0.8431 1.561 43  
## [3] {Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.40 0.8696 1.610 40  
## [4] {Exang=0,   
## Ca=0} => {Disease=0} 0.39 0.8298 1.537 39  
## [5] {Fbs=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.39 0.8125 1.505 39  
## [6] {Fbs=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.39 0.8298 1.537 39  
## [7] {Ca=0,   
## Thal=3} => {Disease=0} 0.38 0.8837 1.637 38  
## [8] {Fbs=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.36 0.8780 1.626 36  
## [9] {Exang=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.36 0.8571 1.587 36  
## [10] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.36 0.8571 1.587 36  
## [11] {Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.35 0.8974 1.662 35  
## [12] {Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.35 0.9211 1.706 35  
## [13] {Fbs=0,   
## Exang=0,   
## Ca=0} => {Disease=0} 0.35 0.8140 1.507 35  
## [14] {Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.33 0.9167 1.698 33  
## [15] {Fbs=0,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.33 0.8919 1.652 33  
## [16] {Fbs=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.33 0.9167 1.698 33  
## [17] {Slope=1,   
## Thal=3} => {Disease=0} 0.32 0.8889 1.646 32  
## [18] {Exang=0,   
## Slope=1} => {Disease=0} 0.32 0.8205 1.519 32  
## [19] {Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.32 0.8889 1.646 32  
## [20] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1} => {Disease=0} 0.32 0.8421 1.559 32  
## [21] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.32 0.8421 1.559 32  
## [22] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.31 0.9118 1.688 31  
## [23] {Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.29 0.8788 1.627 29  
## [24] {Fbs=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.29 0.8788 1.627 29  
## [25] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.29 0.8788 1.627 29  
## [26] {Fbs=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.29 0.8788 1.627 29  
## [27] {Thalach.range=3,   
## Thal=3} => {Disease=0} 0.28 0.8485 1.571 28  
## [28] {Thalach.range=3,   
## Exang=0} => {Disease=0} 0.28 0.8000 1.481 28  
## [29] {Slope=1,   
## Ca=0} => {Disease=0} 0.28 0.8000 1.481 28  
## [30] {Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.28 0.8000 1.481 28  
## [31] {Thalach.range=3,   
## Ca=0} => {Disease=0} 0.27 0.9000 1.667 27  
## [32] {Thalach.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.27 0.8710 1.613 27  
## [33] {Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.27 0.8438 1.562 27  
## [34] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Slope=1} => {Disease=0} 0.27 0.8182 1.515 27  
## [35] {Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.26 0.8966 1.660 26  
## [36] {Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.26 0.8387 1.553 26  
## [37] {Fbs=0,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.26 0.8387 1.553 26  
## [38] {Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.26 0.9286 1.720 26  
## [39] {Exang=0,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.26 0.8667 1.605 26  
## [40] {Oldpeak.range=1,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.26 0.9286 1.720 26  
## [41] {Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.26 0.8667 1.605 26  
## [42] {Fbs=0,   
## Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.26 0.8667 1.605 26  
## [43] {Fbs=0,   
## Exang=0,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.26 0.8667 1.605 26  
## [44] {Sex=0,   
## Thal=3} => {Disease=0} 0.25 0.8621 1.596 25  
## [45] {Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.25 0.8929 1.653 25  
## [46] {Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.25 0.9615 1.781 25  
## [47] {Thalach.range=3,   
## Exang=0,   
## Ca=0} => {Disease=0} 0.25 0.9259 1.715 25  
## [48] {Restecg=2,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.25 0.8333 1.543 25  
## [49] {Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.25 0.8929 1.653 25  
## [50] {Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.25 0.8621 1.596 25

rules\_test\_lift<-**sort**(rulestrain, by='lift', decreasing = T)  
**inspect**(rules\_test\_lift[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Sex=0,   
## CP=3} => {Disease=0} 0.1084 1 1.845 22  
## [2] {Sex=0,   
## CP=3,   
## Exang=0} => {Disease=0} 0.1034 1 1.845 21  
## [3] {Sex=0,   
## CP=3,   
## Oldpeak.range=1} => {Disease=0} 0.1084 1 1.845 22  
## [4] {CP=3,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [5] {CP=3,   
## Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [6] {Chol.range=2,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1034 1 1.845 21  
## [7] {Chol.range=2,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [8] {Age.Range=2,   
## Trestbps.range=3,   
## Thal=3} => {Disease=0} 0.1626 1 1.845 33  
## [9] {Sex=0,   
## CP=3,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.1034 1 1.845 21  
## [10] {CP=3,   
## Trestbps.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1034 1 1.845 21  
## [11] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0} => {Disease=0} 0.1133 1 1.845 23  
## [12] {Chol.range=2,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [13] {Age.Range=2,   
## Trestbps.range=3,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1182 1 1.845 24  
## [14] {Age.Range=2,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1232 1 1.845 25  
## [15] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1478 1 1.845 30  
## [16] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1330 1 1.845 27  
## [17] {Sex=1,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1133 1 1.845 23  
## [18] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1182 1 1.845 24  
## [19] {Restecg=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1281 1 1.845 26  
## [20] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1527 1 1.845 31  
## [21] {Age.Range=2,   
## Trestbps.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1034 1 1.845 21  
## [22] {Age.Range=2,   
## Trestbps.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1429 1 1.845 29  
## [23] {Age.Range=2,   
## Trestbps.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1478 1 1.845 30  
## [24] {Age.Range=2,   
## Trestbps.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1527 1 1.845 31  
## [25] {Age.Range=2,   
## Trestbps.range=3,   
## Fbs=0,   
## Thal=3} => {Disease=0} 0.1379 1 1.845 28  
## [26] {Trestbps.range=3,   
## Restecg=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1232 1 1.845 25  
## [27] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [28] {Sex=0,   
## Fbs=0,   
## Restecg=0,   
## Exang=0,   
## Oldpeak.range=1} => {Disease=0} 0.1133 1 1.845 23  
## [29] {Age.Range=2,   
## Trestbps.range=3,   
## Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [30] {Age.Range=2,   
## Trestbps.range=3,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [31] {Age.Range=2,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [32] {Age.Range=2,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [33] {Age.Range=2,   
## Fbs=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [34] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1034 1 1.845 21  
## [35] {Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [36] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1330 1 1.845 27  
## [37] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1478 1 1.845 30  
## [38] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Thal=3} => {Disease=0} 0.1379 1 1.845 28  
## [39] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1084 1 1.845 22  
## [40] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1330 1 1.845 27  
## [41] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1232 1 1.845 25  
## [42] {Sex=1,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Slope=1,   
## Ca=0} => {Disease=0} 0.1133 1 1.845 23  
## [43] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [44] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1182 1 1.845 24  
## [45] {Trestbps.range=3,   
## Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Thal=3} => {Disease=0} 0.1084 1 1.845 22  
## [46] {Trestbps.range=3,   
## Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0} => {Disease=0} 0.1133 1 1.845 23  
## [47] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1133 1 1.845 23  
## [48] {Restecg=0,   
## Thalach.range=3,   
## Oldpeak.range=1,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1281 1 1.845 26  
## [49] {Fbs=0,   
## Restecg=0,   
## Thalach.range=3,   
## Ca=0,   
## Thal=3} => {Disease=0} 0.1232 1 1.845 25  
## [50] {Restecg=0,   
## Thalach.range=3,   
## Exang=0,   
## Oldpeak.range=1,   
## Thal=3} => {Disease=0} 0.1527 1 1.845 31

**summary**(rulestest)

## set of 1193 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5 6 7 8 9   
## 2 42 192 372 362 180 41 2   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 5.00 5.00 5.48 6.00 9.00   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.100 Min. :0.800 Min. :1.48 Min. :10.0   
## 1st Qu.:0.110 1st Qu.:0.857 1st Qu.:1.59 1st Qu.:11.0   
## Median :0.120 Median :0.909 Median :1.68 Median :12.0   
## Mean :0.137 Mean :0.901 Mean :1.67 Mean :13.7   
## 3rd Qu.:0.140 3rd Qu.:0.929 3rd Qu.:1.72 3rd Qu.:14.0   
## Max. :0.430 Max. :1.000 Max. :1.85 Max. :43.0   
##   
## mining info:  
## data ntransactions support confidence  
## (testData\_m) 100 0.1 0.8

rules1test<-**apriori**((testData\_m), appearance=**list**(rhs= **c**("Disease=1"), default="lhs"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 10 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 10   
##   
## set item appearances ...[1 item(s)] done [0.00s].  
## set transactions ...[43 item(s), 100 transaction(s)] done [0.00s].  
## sorting and recoding items ... [36 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 9 done [0.00s].  
## writing ... [252 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

rules\_1test\_count<- **sort**(rules1test, by='count', decreasing = TRUE)  
**inspect**(rules\_1test\_count[1**:**50])

## lhs rhs support confidence  
## [1] {Thal=7} => {Disease=1} 0.28 0.8000   
## [2] {Exang=1} => {Disease=1} 0.26 0.8387   
## [3] {CP=4,Exang=1} => {Disease=1} 0.26 0.9286   
## [4] {CP=4,Thal=7} => {Disease=1} 0.24 0.9600   
## [5] {Fbs=0,Thal=7} => {Disease=1} 0.24 0.8571   
## [6] {Sex=1,Exang=1} => {Disease=1} 0.22 0.8462   
## [7] {Fbs=0,Exang=1} => {Disease=1} 0.22 0.8148   
## [8] {Sex=1,CP=4,Exang=1} => {Disease=1} 0.22 0.9167   
## [9] {CP=4,Fbs=0,Exang=1} => {Disease=1} 0.22 0.9167   
## [10] {Sex=1,Slope=2} => {Disease=1} 0.21 0.8077   
## [11] {Sex=1,CP=4,Thal=7} => {Disease=1} 0.21 0.9545   
## [12] {Sex=1,Fbs=0,Thal=7} => {Disease=1} 0.21 0.8400   
## [13] {CP=4,Fbs=0,Thal=7} => {Disease=1} 0.20 1.0000   
## [14] {Sex=1,Fbs=0,Slope=2} => {Disease=1} 0.20 0.8333   
## [15] {Exang=1,Thal=7} => {Disease=1} 0.19 0.9500   
## [16] {CP=4,Exang=1,Thal=7} => {Disease=1} 0.19 1.0000   
## [17] {Sex=1,Fbs=0,Exang=1} => {Disease=1} 0.19 0.8261   
## [18] {Sex=1,CP=4,Fbs=0,Exang=1} => {Disease=1} 0.19 0.9048   
## [19] {Sex=1,CP=4,Fbs=0,Thal=7} => {Disease=1} 0.18 1.0000   
## [20] {Oldpeak.range=2} => {Disease=1} 0.17 0.8095   
## [21] {Sex=1,CP=4,Slope=2} => {Disease=1} 0.17 0.8947   
## [22] {Age.Range=3,Sex=1,CP=4} => {Disease=1} 0.17 0.8947   
## [23] {Fbs=0,Oldpeak.range=2} => {Disease=1} 0.16 0.8000   
## [24] {Exang=1,Slope=2} => {Disease=1} 0.16 0.8889   
## [25] {Age.Range=3,Exang=1} => {Disease=1} 0.16 0.9412   
## [26] {Slope=2,Thal=7} => {Disease=1} 0.16 0.8889   
## [27] {CP=4,Trestbps.range=4} => {Disease=1} 0.16 0.8000   
## [28] {Sex=1,Exang=1,Thal=7} => {Disease=1} 0.16 0.9412   
## [29] {CP=4,Exang=1,Slope=2} => {Disease=1} 0.16 0.9412   
## [30] {Age.Range=3,CP=4,Exang=1} => {Disease=1} 0.16 1.0000   
## [31] {CP=4,Oldpeak.range=1,Thal=7} => {Disease=1} 0.16 0.9412   
## [32] {Sex=1,CP=4,Exang=1,Thal=7} => {Disease=1} 0.16 1.0000   
## [33] {Sex=1,CP=4,Fbs=0,Slope=2} => {Disease=1} 0.16 0.8889   
## [34] {Chol.range=3,Exang=1} => {Disease=1} 0.15 0.8333   
## [35] {Chol.range=3,Thal=7} => {Disease=1} 0.15 0.8333   
## [36] {Restecg=2,Thal=7} => {Disease=1} 0.15 0.8333   
## [37] {Fbs=0,Exang=1,Thal=7} => {Disease=1} 0.15 0.9375   
## [38] {Fbs=0,Exang=1,Slope=2} => {Disease=1} 0.15 0.8824   
## [39] {CP=4,Chol.range=3,Exang=1} => {Disease=1} 0.15 0.9375   
## [40] {CP=4,Restecg=2,Exang=1} => {Disease=1} 0.15 0.8824   
## [41] {CP=4,Exang=1,Oldpeak.range=1} => {Disease=1} 0.15 0.9375   
## [42] {Fbs=0,Slope=2,Thal=7} => {Disease=1} 0.15 0.8824   
## [43] {Sex=1,Restecg=2,Thal=7} => {Disease=1} 0.15 0.8333   
## [44] {Fbs=0,Oldpeak.range=1,Thal=7} => {Disease=1} 0.15 0.8333   
## [45] {CP=4,Fbs=0,Exang=1,Thal=7} => {Disease=1} 0.15 1.0000   
## [46] {CP=4,Fbs=0,Exang=1,Slope=2} => {Disease=1} 0.15 0.9375   
## [47] {Sex=1,CP=4,Restecg=2,Exang=1} => {Disease=1} 0.15 0.8824   
## [48] {Sex=1,Oldpeak.range=2} => {Disease=1} 0.14 0.8750   
## [49] {Age.Range=3,Fbs=0,Exang=1} => {Disease=1} 0.14 0.9333   
## [50] {Chol.range=3,Fbs=0,Exang=1} => {Disease=1} 0.14 0.8235   
## lift count  
## [1] 1.739 28   
## [2] 1.823 26   
## [3] 2.019 26   
## [4] 2.087 24   
## [5] 1.863 24   
## [6] 1.839 22   
## [7] 1.771 22   
## [8] 1.993 22   
## [9] 1.993 22   
## [10] 1.756 21   
## [11] 2.075 21   
## [12] 1.826 21   
## [13] 2.174 20   
## [14] 1.812 20   
## [15] 2.065 19   
## [16] 2.174 19   
## [17] 1.796 19   
## [18] 1.967 19   
## [19] 2.174 18   
## [20] 1.760 17   
## [21] 1.945 17   
## [22] 1.945 17   
## [23] 1.739 16   
## [24] 1.932 16   
## [25] 2.046 16   
## [26] 1.932 16   
## [27] 1.739 16   
## [28] 2.046 16   
## [29] 2.046 16   
## [30] 2.174 16   
## [31] 2.046 16   
## [32] 2.174 16   
## [33] 1.932 16   
## [34] 1.812 15   
## [35] 1.812 15   
## [36] 1.812 15   
## [37] 2.038 15   
## [38] 1.918 15   
## [39] 2.038 15   
## [40] 1.918 15   
## [41] 2.038 15   
## [42] 1.918 15   
## [43] 1.812 15   
## [44] 1.812 15   
## [45] 2.174 15   
## [46] 2.038 15   
## [47] 1.918 15   
## [48] 1.902 14   
## [49] 2.029 14   
## [50] 1.790 14

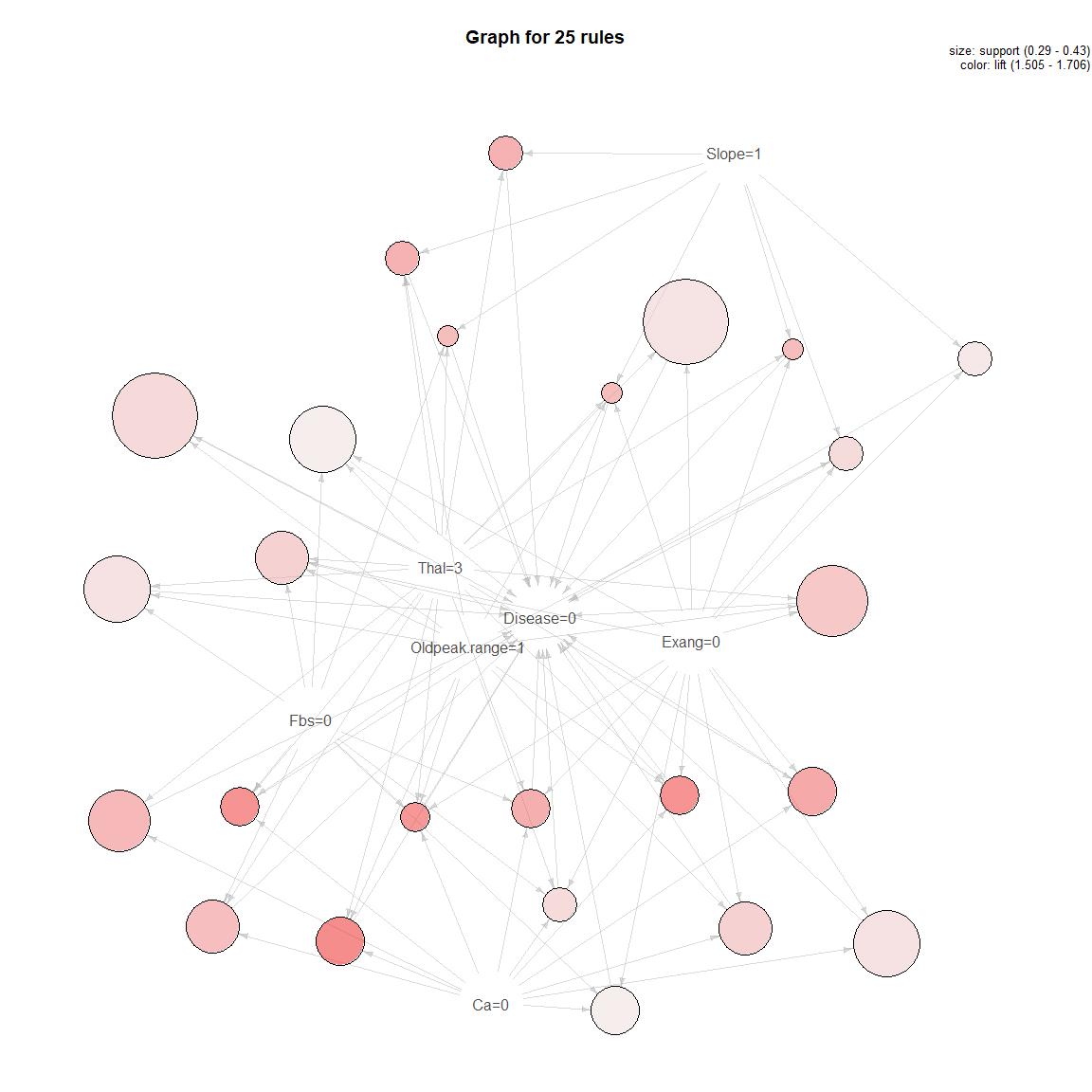
rules\_1test\_lift<- **sort**(rules1test, by='lift', decreasing = TRUE)  
**inspect**(rules\_1test\_lift[1**:**50])

## lhs rhs support confidence lift count  
## [1] {Age.Range=3,   
## Thalach.range=1} => {Disease=1} 0.11 1 2.174 11  
## [2] {CP=4,   
## Thalach.range=1} => {Disease=1} 0.12 1 2.174 12  
## [3] {Sex=1,   
## Thalach.range=1} => {Disease=1} 0.11 1 2.174 11  
## [4] {Oldpeak.range=2,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [5] {CP=4,   
## Ca=1} => {Disease=1} 0.12 1 2.174 12  
## [6] {Age.Range=3,   
## CP=4,   
## Thalach.range=1} => {Disease=1} 0.10 1 2.174 10  
## [7] {Sex=1,   
## CP=4,   
## Thalach.range=1} => {Disease=1} 0.11 1 2.174 11  
## [8] {CP=4,   
## Fbs=0,   
## Thalach.range=1} => {Disease=1} 0.10 1 2.174 10  
## [9] {Age.Range=3,   
## Slope=2,   
## Ca=1} => {Disease=1} 0.10 1 2.174 10  
## [10] {Fbs=0,   
## Slope=2,   
## Ca=1} => {Disease=1} 0.10 1 2.174 10  
## [11] {Age.Range=3,   
## CP=4,   
## Ca=1} => {Disease=1} 0.10 1 2.174 10  
## [12] {Sex=1,   
## CP=4,   
## Ca=1} => {Disease=1} 0.11 1 2.174 11  
## [13] {CP=4,   
## Fbs=0,   
## Ca=1} => {Disease=1} 0.12 1 2.174 12  
## [14] {CP=4,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.19 1 2.174 19  
## [15] {CP=4,   
## Trestbps.range=3,   
## Exang=1} => {Disease=1} 0.11 1 2.174 11  
## [16] {Age.Range=3,   
## CP=4,   
## Exang=1} => {Disease=1} 0.16 1 2.174 16  
## [17] {CP=4,   
## Trestbps.range=4,   
## Thal=7} => {Disease=1} 0.11 1 2.174 11  
## [18] {CP=4,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.13 1 2.174 13  
## [19] {CP=4,   
## Trestbps.range=3,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [20] {Age.Range=3,   
## CP=4,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12  
## [21] {CP=4,   
## Chol.range=3,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12  
## [22] {CP=4,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12  
## [23] {CP=4,   
## Ca=0,   
## Thal=7} => {Disease=1} 0.13 1 2.174 13  
## [24] {CP=4,   
## Fbs=0,   
## Thal=7} => {Disease=1} 0.20 1 2.174 20  
## [25] {Age.Range=3,   
## Fbs=0,   
## Slope=2,   
## Ca=1} => {Disease=1} 0.10 1 2.174 10  
## [26] {Age.Range=3,   
## CP=4,   
## Fbs=0,   
## Ca=1} => {Disease=1} 0.10 1 2.174 10  
## [27] {Sex=1,   
## CP=4,   
## Fbs=0,   
## Ca=1} => {Disease=1} 0.11 1 2.174 11  
## [28] {CP=4,   
## Exang=1,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [29] {CP=4,   
## Thalach.range=2,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [30] {Age.Range=3,   
## CP=4,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [31] {CP=4,   
## Chol.range=3,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [32] {CP=4,   
## Exang=1,   
## Ca=0,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [33] {Sex=1,   
## CP=4,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.16 1 2.174 16  
## [34] {CP=4,   
## Exang=1,   
## Oldpeak.range=1,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12  
## [35] {CP=4,   
## Fbs=0,   
## Exang=1,   
## Thal=7} => {Disease=1} 0.15 1 2.174 15  
## [36] {Age.Range=3,   
## CP=4,   
## Exang=1,   
## Slope=2} => {Disease=1} 0.10 1 2.174 10  
## [37] {Sex=1,   
## CP=4,   
## Trestbps.range=3,   
## Exang=1} => {Disease=1} 0.10 1 2.174 10  
## [38] {CP=4,   
## Trestbps.range=3,   
## Fbs=0,   
## Exang=1} => {Disease=1} 0.10 1 2.174 10  
## [39] {Age.Range=3,   
## CP=4,   
## Chol.range=3,   
## Exang=1} => {Disease=1} 0.10 1 2.174 10  
## [40] {Age.Range=3,   
## CP=4,   
## Restecg=2,   
## Exang=1} => {Disease=1} 0.10 1 2.174 10  
## [41] {Age.Range=3,   
## Sex=1,   
## CP=4,   
## Exang=1} => {Disease=1} 0.13 1 2.174 13  
## [42] {Age.Range=3,   
## CP=4,   
## Fbs=0,   
## Exang=1} => {Disease=1} 0.14 1 2.174 14  
## [43] {Sex=1,   
## CP=4,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [44] {CP=4,   
## Fbs=0,   
## Slope=2,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12  
## [45] {CP=4,   
## Fbs=0,   
## Thalach.range=2,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12  
## [46] {Age.Range=3,   
## Sex=1,   
## CP=4,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [47] {Age.Range=3,   
## CP=4,   
## Fbs=0,   
## Thal=7} => {Disease=1} 0.10 1 2.174 10  
## [48] {Sex=1,   
## CP=4,   
## Chol.range=3,   
## Thal=7} => {Disease=1} 0.11 1 2.174 11  
## [49] {CP=4,   
## Chol.range=3,   
## Fbs=0,   
## Thal=7} => {Disease=1} 0.11 1 2.174 11  
## [50] {Sex=1,   
## CP=4,   
## Restecg=2,   
## Thal=7} => {Disease=1} 0.12 1 2.174 12

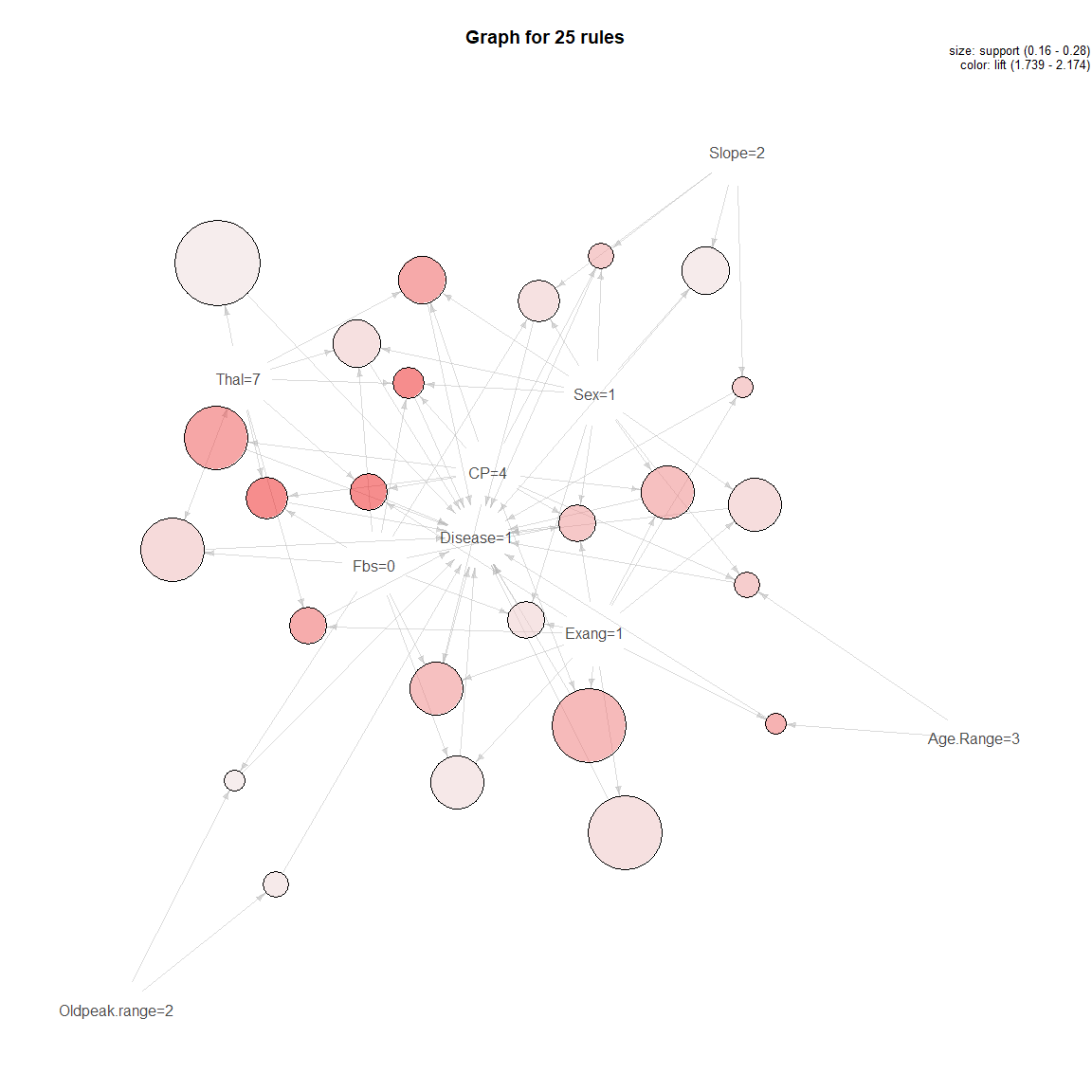
**summary**(rules1test)

## set of 252 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 2 3 4 5 6   
## 5 35 100 91 21   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.00 4.00 4.00 4.35 5.00 6.00   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.100 Min. :0.800 Min. :1.74 Min. :10.0   
## 1st Qu.:0.100 1st Qu.:0.846 1st Qu.:1.84 1st Qu.:10.0   
## Median :0.115 Median :0.909 Median :1.98 Median :11.5   
## Mean :0.125 Mean :0.908 Mean :1.97 Mean :12.5   
## 3rd Qu.:0.130 3rd Qu.:1.000 3rd Qu.:2.17 3rd Qu.:13.0   
## Max. :0.280 Max. :1.000 Max. :2.17 Max. :28.0   
##   
## mining info:  
## data ntransactions support confidence  
## (testData\_m) 100 0.1 0.8

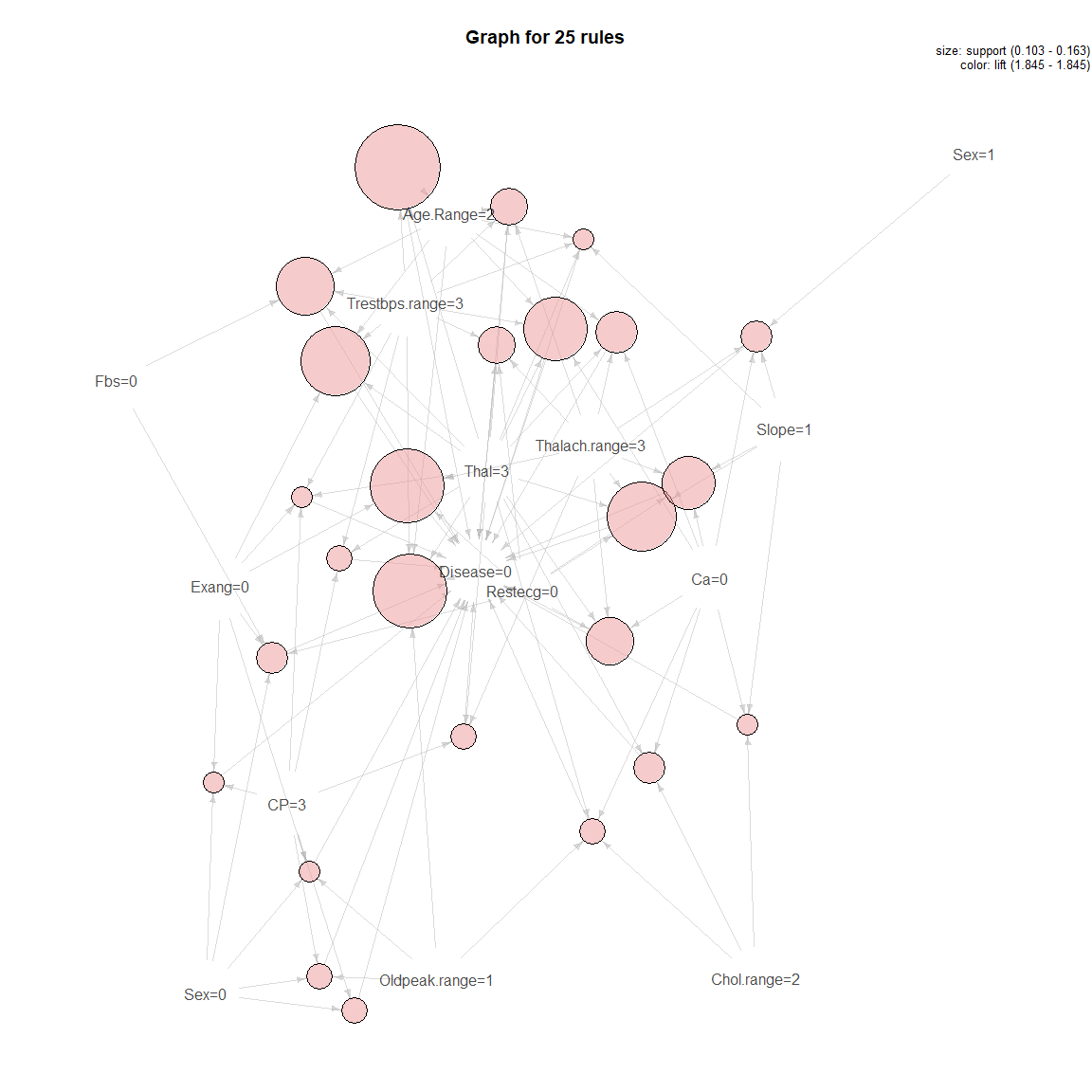
topRules\_test\_count<-rules\_test\_count[1**:**25]  
topRules\_1test\_count<-rules\_1test\_count[1**:**25]  
topRules\_test\_lift<-rules\_test\_lift[1**:**25]  
topRules\_1test\_lift<-rules\_1test\_lift[1**:**25]  
  
**plot**(topRules\_test\_count, method="graph")



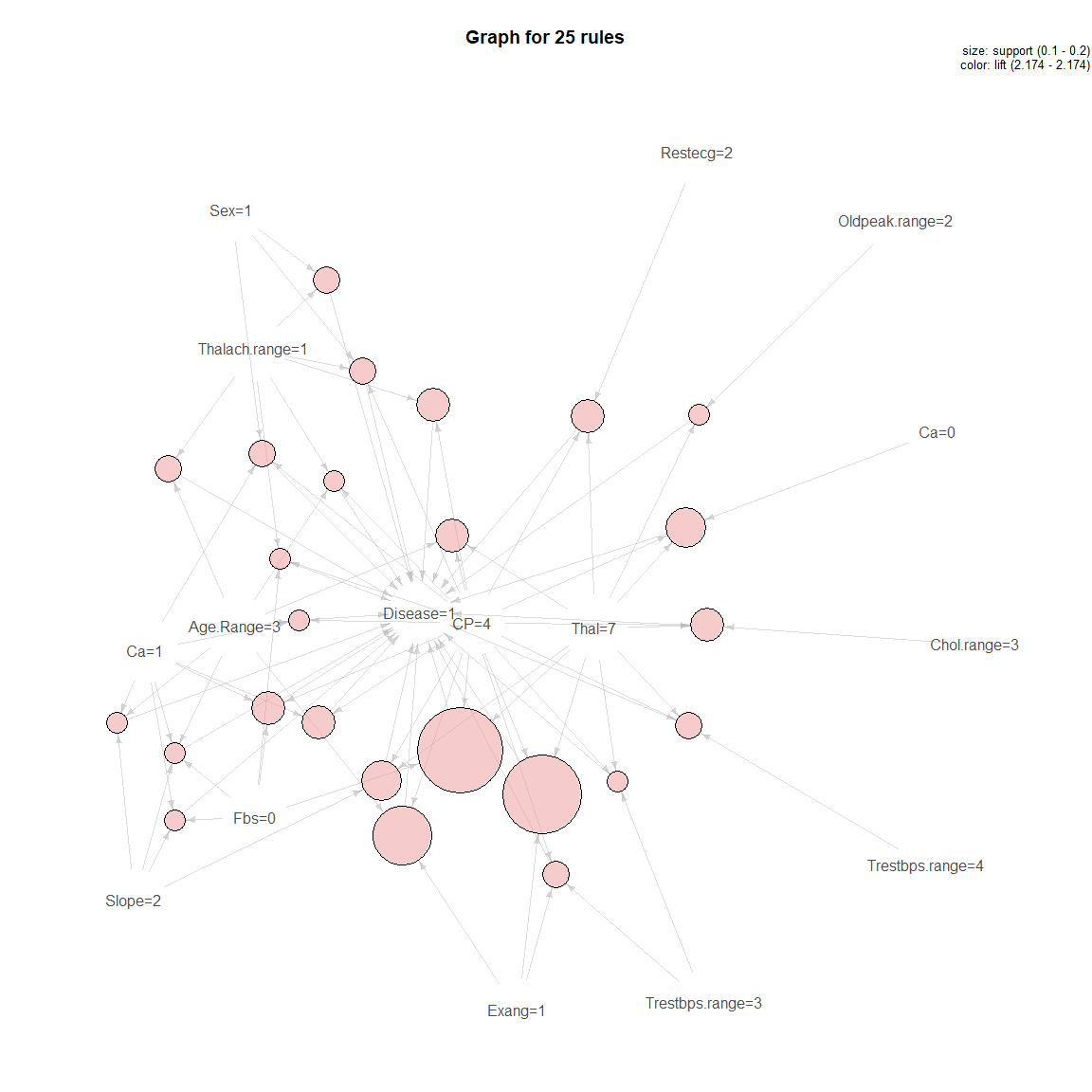
**plot**(topRules\_1test\_count, method="graph")



**plot**(topRules\_test\_lift, method="graph")



**plot**(topRules\_1test\_lift, method="graph")



# 4. Logistic Regression (Quasibinomial) (glm)

############################################  
*# 4. Logistic Regression (Quasibinomial)*  
model.name <- "Logistic Regression (Quasibinomial) (glm)"  
model.name1 <- model.name  
file.name <- **paste**(work.dir, "logistic.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "logistic-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
 (reg.logistic <- **train**(Disease **~** Age **+** Sex **+** CP **+** Trestbps **+** Chol **+** Fbs **+** Restecg   
 **+** Thalach **+** Exang **+** Oldpeak **+** Slope **+** Ca **+** Thal,   
 data = trainData,  
 method = "glm",   
 family = "quasibinomial"))  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(reg.logistic, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Logistic Regression (Quasibinomial) (glm) Executed ... time:1.3"

Let’s see the results:

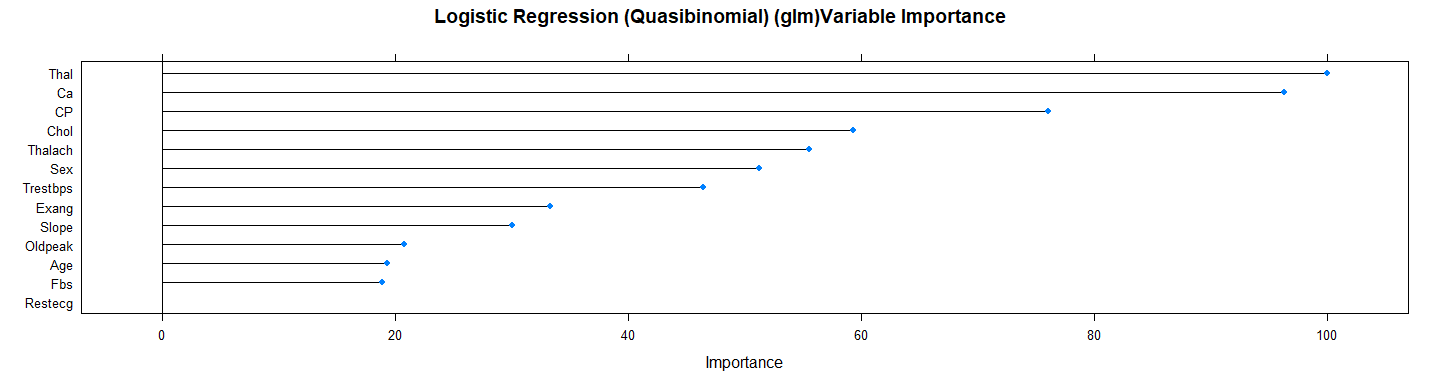
reg.logistic

## Generalized Linear Model   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 198, 198, 198, 198, 198, 198, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8132 0.6247

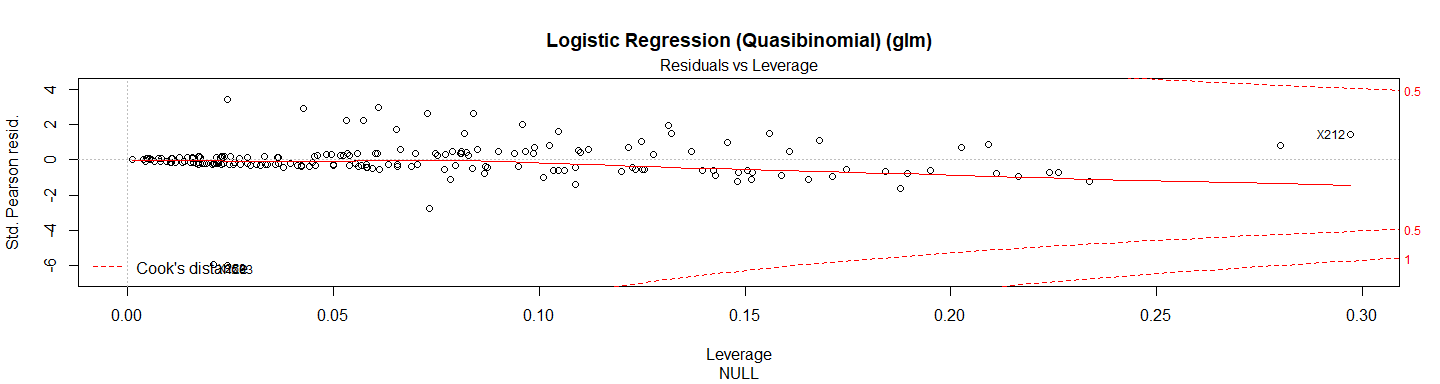
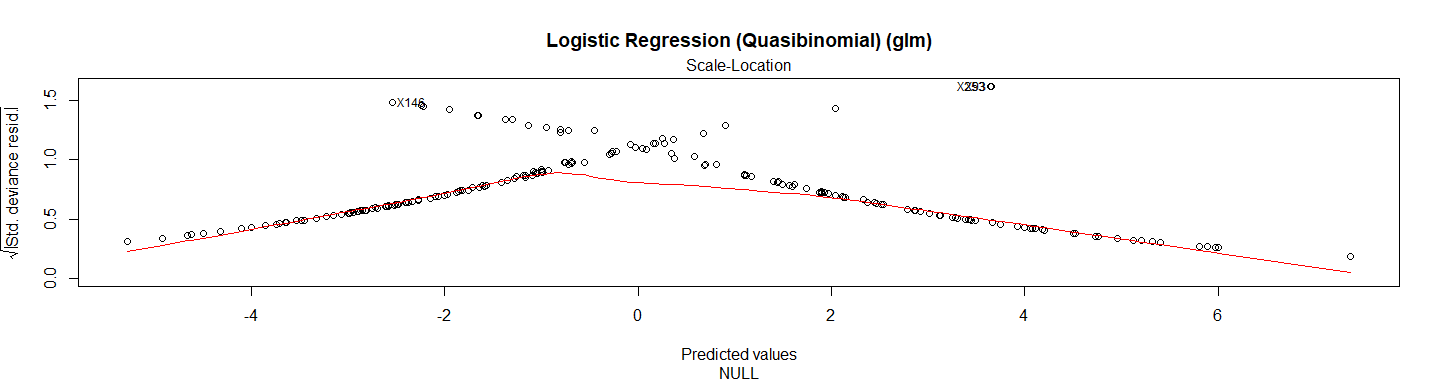
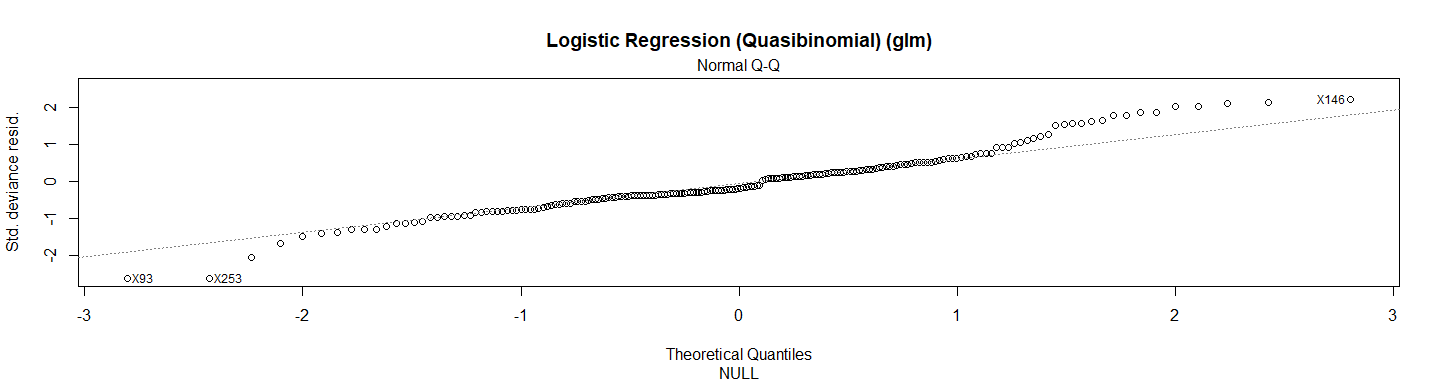
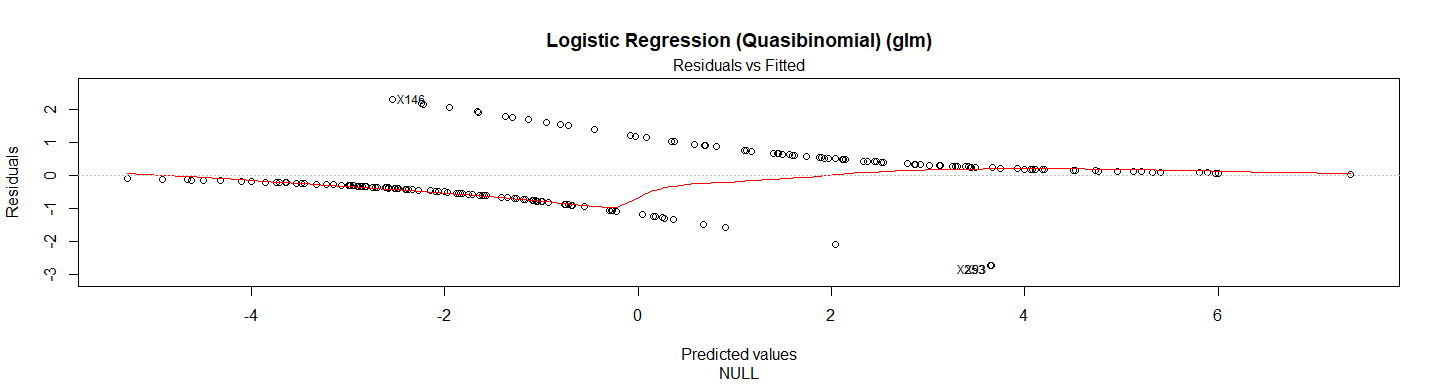
**summary**(reg.logistic)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.713 -0.508 -0.198 0.402 2.288   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -7.64475 3.83052 -2.00 0.0474 \*   
## Age -0.02474 0.03346 -0.74 0.4606   
## Sex 1.09851 0.62850 1.75 0.0822 .   
## CP 0.66565 0.26339 2.53 0.0123 \*   
## Trestbps 0.02224 0.01394 1.59 0.1125   
## Chol 0.01113 0.00557 2.00 0.0470 \*   
## Fbs -0.50169 0.69070 -0.73 0.4685   
## Restecg 0.03192 0.24187 0.13 0.8952   
## Thalach -0.02483 0.01320 -1.88 0.0616 .   
## Exang 0.66244 0.56059 1.18 0.2389   
## Oldpeak 0.22920 0.29119 0.79 0.4322   
## Slope 0.51804 0.48082 1.08 0.2827   
## Ca 1.12548 0.35527 3.17 0.0018 \*\*  
## Thal 0.43944 0.13386 3.28 0.0012 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasibinomial family taken to be 1.106)  
##   
## Null deviance: 273.19 on 197 degrees of freedom  
## Residual deviance: 134.08 on 184 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 6

**plot**(**varImp**(reg.logistic), main = **paste0**(model.name,"Variable Importance"))



**plot**(reg.logistic**$**finalModel, main = model.name)



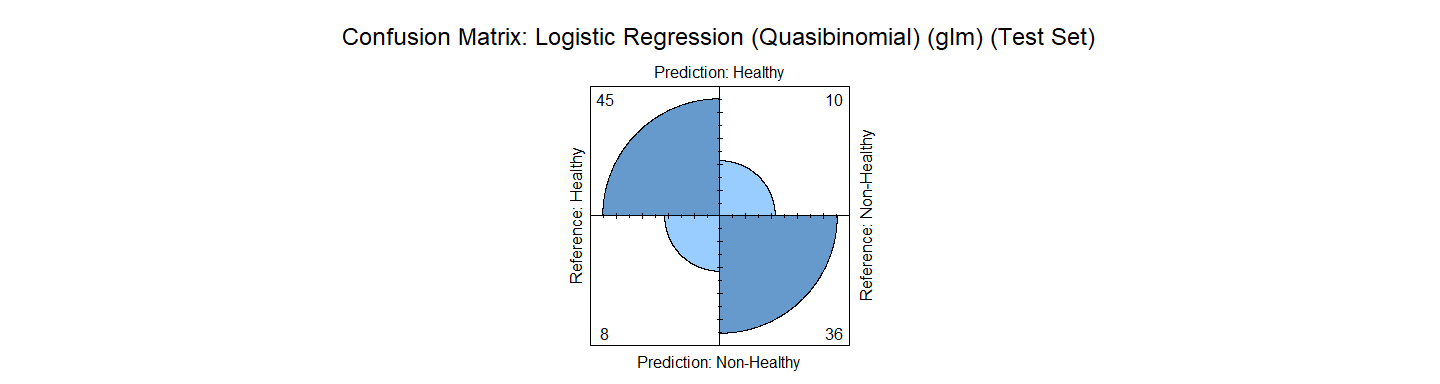
test\_pred\_logistic <- **predict**(reg.logistic, newdata = X.testData)  
test\_pred\_logistic

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 1 0 0 0  
## Levels: 0 1

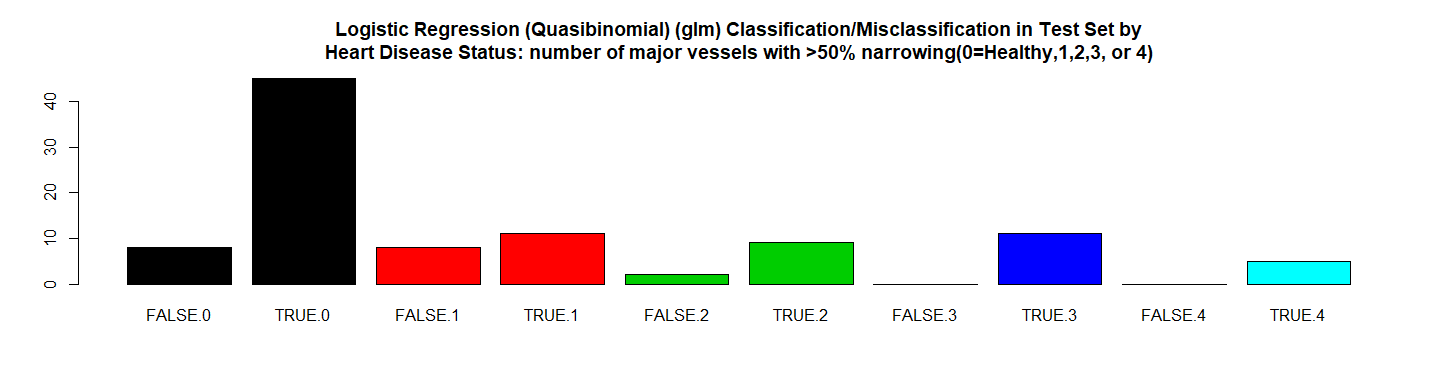
res.logistic <- **confusionMatrix**(test\_pred\_logistic, Y.testData)   
res.logistic

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 45 10  
## 1 8 36  
##   
## Accuracy : 0.818   
## 95% CI : (0.728, 0.889)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 3.57e-09   
##   
## Kappa : 0.633   
## Mcnemar's Test P-Value : 0.814   
##   
## Sensitivity : 0.849   
## Specificity : 0.783   
## Pos Pred Value : 0.818   
## Neg Pred Value : 0.818   
## Prevalence : 0.535   
## Detection Rate : 0.455   
## Detection Prevalence : 0.556   
## Balanced Accuracy : 0.816   
##   
## 'Positive' Class : 0   
##

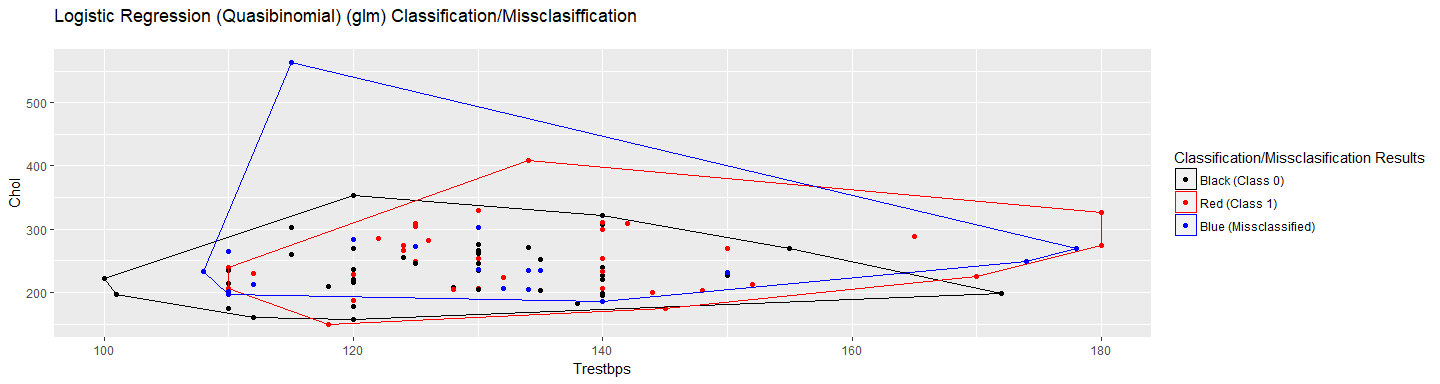
*# This function is used to plot the confusion matrix*  
print\_confusionm <- **function**(confmatrix, main, labels) {  
 temp <- confmatrix**$**table  
 **colnames**(temp) <- labels  
 **rownames**(temp) <- labels  
   
 **fourfoldplot**(temp, main = main, conf.level = 0)   
}  
  
**print\_confusionm**(res.logistic,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



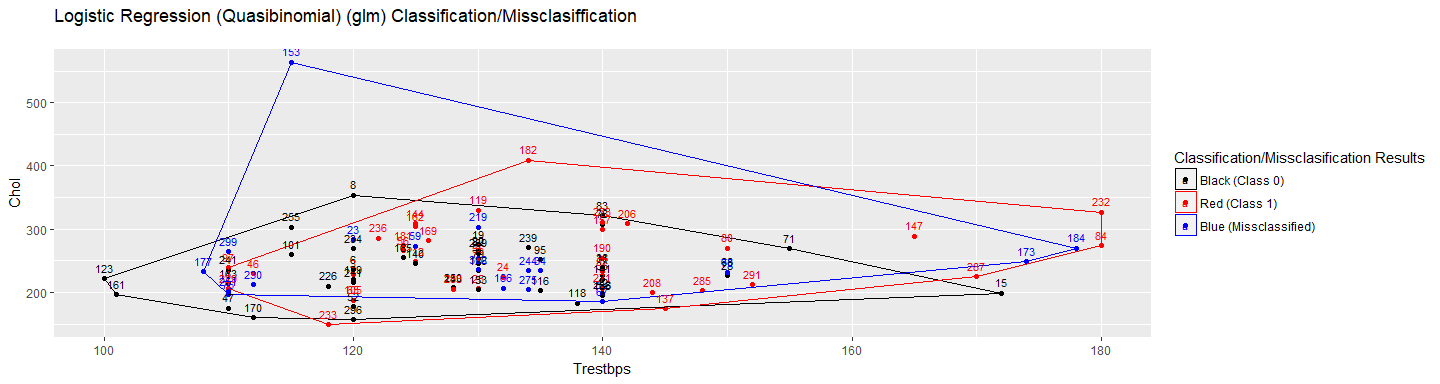
*# Let's see the Complete Classification/Missclassification considering the Type the next*  
*# function will calculate that*  
  
plot\_results\_by\_type <- **function**(predict, real.class, main, testData) {  
   
 new.set <- **data.frame**(predict)   
 new.set**$**classification <- predict **==** real.class   
 new.set**$**type <- testData**$**Num  
 new.set**$**factorC <- **with**(new.set,   
 **interaction**(**factor**(classification), **factor**(type)))  
 **plot**(new.set**$**factorC, main = main, col = **c**(1,1,2,2,3,3,4,4,5,5))  
}  
  
main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_logistic,  
 Y.testData,  
 main, testData)



*#This function will be used to plot the results*   
  
plot\_results <- **function**(X.data, Y.data, Y.predict, title,   
 var1 = NULL, var2 = NULL, labels = TRUE) {  
  
 res <- **data.frame**(X.data)  
 res**$**class <- Y.data  
 res**$**ID <- **row.names**(res)  
 res**$**predict\_class <- Y.predict  
 res**$**col <- **ifelse**(res**$**class **==** res**$**predict\_class, res**$**class,3)  
   
 **if** (**is.null**(var1)) {  
 var1 <- **colnames**(res)[1]  
 }  
   
 **if** (**is.null**(var2)) {  
 var2 <- **colnames**(res)[2]  
 }  
   
 g <- **ggplot**(res,  
 **aes**(res[[var1]], res[[var2]],  
 color = **as.factor**(res**$**col))) **+**   
 **scale\_color\_manual**(values = **c**(1,2,4),   
 name ="Classification/Missclasification Results",  
 labels =**c**("Black (Class 0)","Red (Class 1)","Blue (Missclassified)")) **+**  
 **geom\_point**() **+**   
 **labs**(title = title) **+** **xlab**(var1) **+** **ylab**(var2) **+**   
 **stat\_chull**(**aes**(group = **as.factor**(res**$**col)),  
 geom = "polygon", fill = NA)  
   
 **if** (labels) {  
 *# Add the labels to the graph*  
 g <- g **+** **geom\_text**(**aes**(label = res**$**ID), size = 3, vjust = **-**0.7)  
 }  
 g  
}  
  
**plot\_results**(X.testData, Y.testData,test\_pred\_logistic ,   
 **paste0**(model.name," Classification/Missclasiffication\n"),   
 "Trestbps","Chol",labels = FALSE)



**plot\_results**(X.testData, Y.testData, test\_pred\_logistic,   
 **paste0**(model.name," Classification/Missclasiffication\n"),  
 "Trestbps","Chol",labels = TRUE)



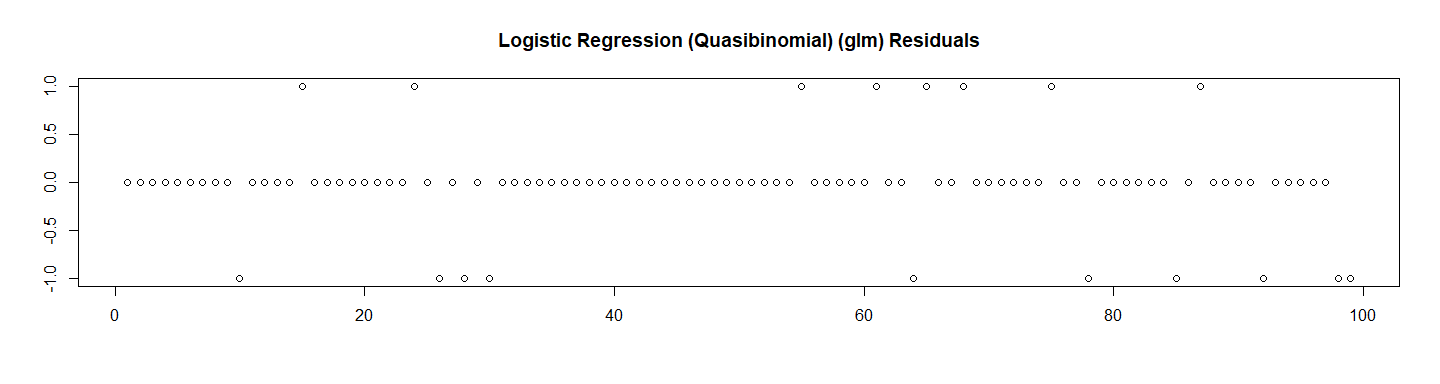
*# RMSE Resulting*  
(rmse\_logistic <- **sqrt**(**mean**((**as.numeric**(test\_pred\_logistic)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4264

*# Residuals Plot*  
(r\_logistic <- (**as.numeric**(test\_pred\_logistic)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0 0 0 0  
## [24] 1 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 0 0 -1 -1

**plot**(r\_logistic,   
 main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_logistic\_prob <- **predict**(reg.logistic, newdata = X.testData, type ="prob")[2]  
  
res.logistic.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**((test\_pred\_logistic\_prob))))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 reg.logistic**$**results**$**Accuracy,  
 "Prediction Accuracy in Test Set" = res.logistic**$**overall[1],  
 "RMSE Test"= rmse\_logistic,  
 "ROC" = res.logistic.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE)

# 5. Logistic Regression (Binomial) (glm)

############################################  
*# 5. Logistic Regression (Binomial) (glm)*  
model.name <- "Logistic Regression (Binomial) (glm)"  
model.name2 <- model.name  
file.name <- **paste**(work.dir, "logistic2.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "logistic2-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
 (reg.logistic2 <- **train**(**factor**(Disease) **~** Age **+** Sex **+** CP **+** Trestbps **+** Chol **+** Fbs **+** Restecg   
 **+** Thalach **+** Exang **+** Oldpeak **+** Slope **+** Ca **+** Thal,   
 data = trainData,  
 method = "glm",   
 family = "binomial"))  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(reg.logistic2, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Logistic Regression (Binomial) (glm) Executed ... time:1.4"

Let’s see the results:

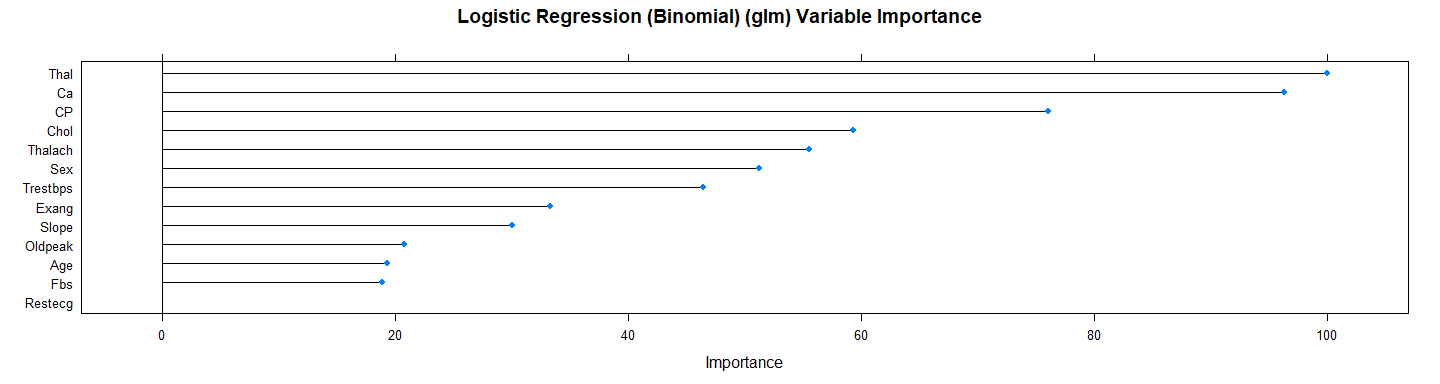
reg.logistic2

## Generalized Linear Model   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 198, 198, 198, 198, 198, 198, ...   
## Resampling results:  
##   
## Accuracy Kappa  
## 0.8021 0.598

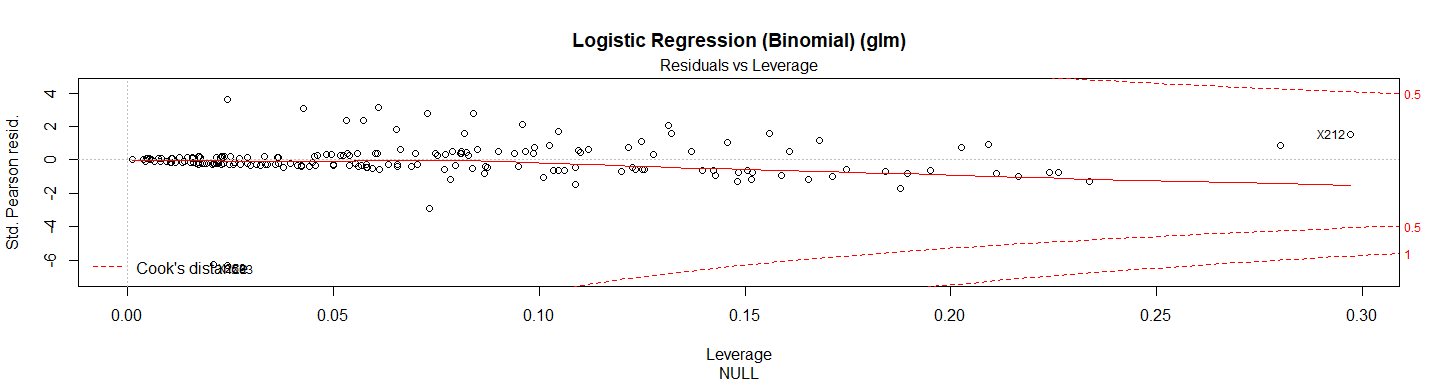
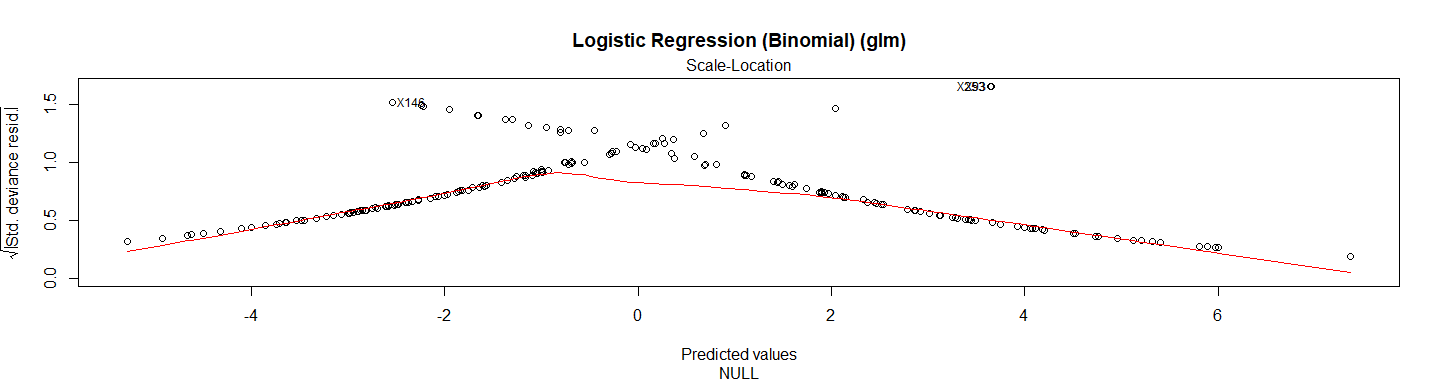
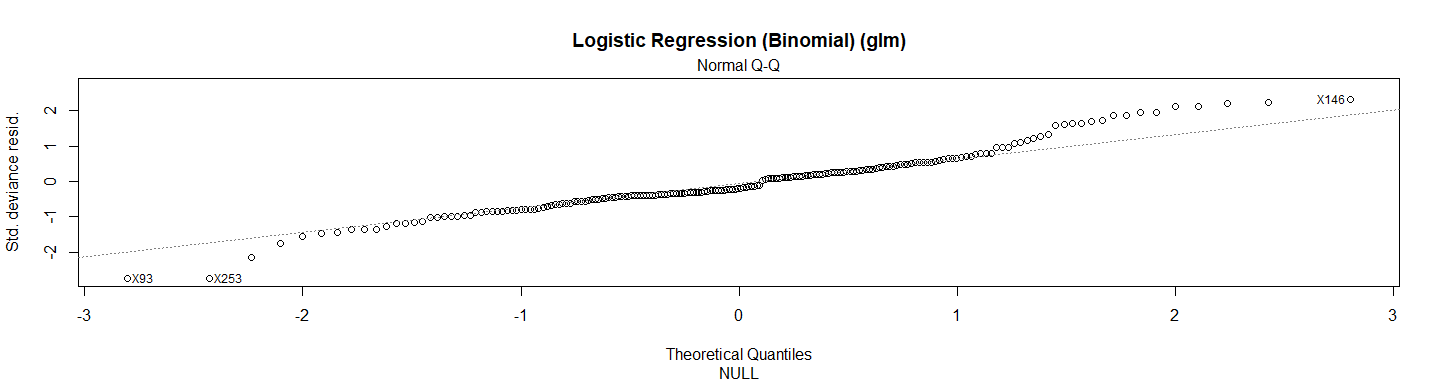
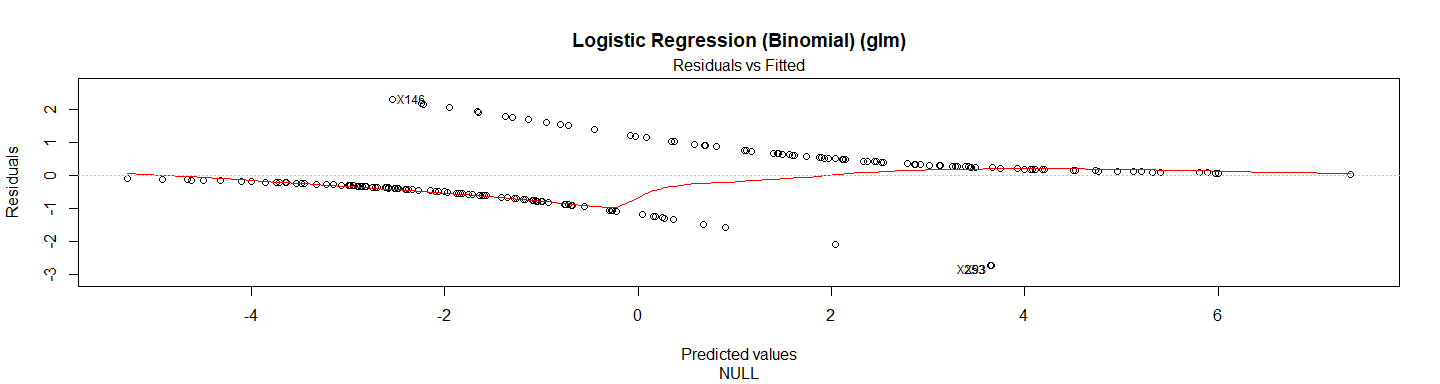
**summary**(reg.logistic2)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.713 -0.508 -0.198 0.402 2.288   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.64475 3.64255 -2.10 0.03584 \*   
## Age -0.02474 0.03182 -0.78 0.43678   
## Sex 1.09851 0.59766 1.84 0.06606 .   
## CP 0.66565 0.25046 2.66 0.00787 \*\*   
## Trestbps 0.02224 0.01326 1.68 0.09351 .   
## Chol 0.01113 0.00529 2.10 0.03547 \*   
## Fbs -0.50169 0.65680 -0.76 0.44496   
## Restecg 0.03192 0.23001 0.14 0.88963   
## Thalach -0.02483 0.01255 -1.98 0.04796 \*   
## Exang 0.66244 0.53309 1.24 0.21400   
## Oldpeak 0.22920 0.27691 0.83 0.40784   
## Slope 0.51804 0.45723 1.13 0.25721   
## Ca 1.12548 0.33784 3.33 0.00086 \*\*\*  
## Thal 0.43944 0.12729 3.45 0.00056 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 273.19 on 197 degrees of freedom  
## Residual deviance: 134.08 on 184 degrees of freedom  
## AIC: 162.1  
##   
## Number of Fisher Scoring iterations: 6

**plot**(**varImp**(reg.logistic2),   
 main = **paste0**(model.name," Variable Importance"))



**plot**(reg.logistic2**$**finalModel, main = model.name)



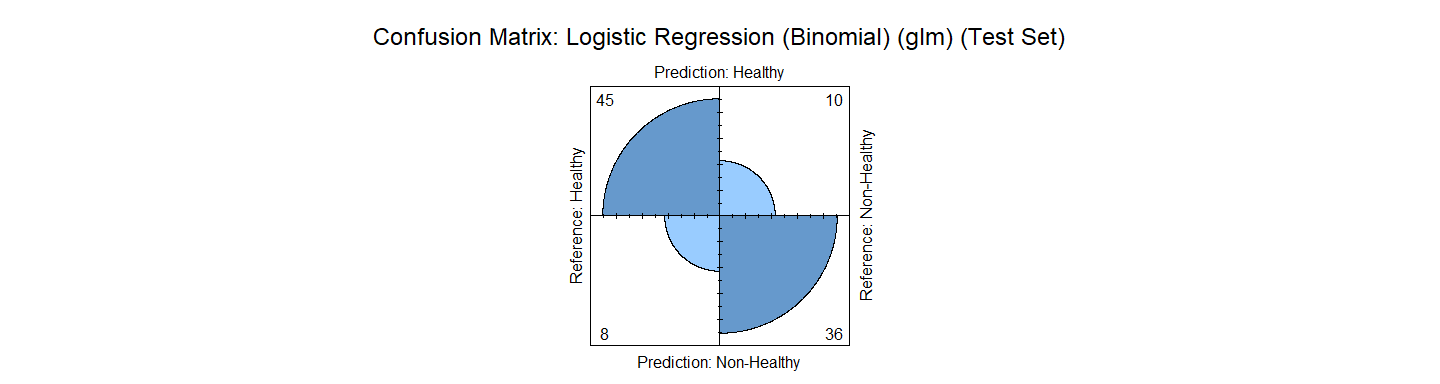
test\_pred\_logistic2 <- **predict**(reg.logistic2, newdata = X.testData)  
test\_pred\_logistic2

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 1 0 0 0  
## Levels: 0 1

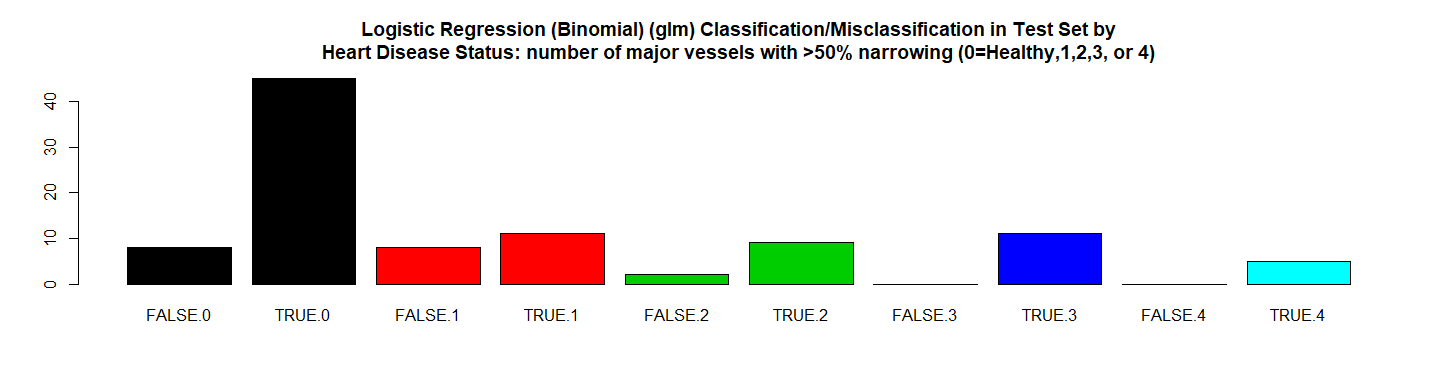
res.logistic2 <- **confusionMatrix**(test\_pred\_logistic2, Y.testData)   
res.logistic2

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 45 10  
## 1 8 36  
##   
## Accuracy : 0.818   
## 95% CI : (0.728, 0.889)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 3.57e-09   
##   
## Kappa : 0.633   
## Mcnemar's Test P-Value : 0.814   
##   
## Sensitivity : 0.849   
## Specificity : 0.783   
## Pos Pred Value : 0.818   
## Neg Pred Value : 0.818   
## Prevalence : 0.535   
## Detection Rate : 0.455   
## Detection Prevalence : 0.556   
## Balanced Accuracy : 0.816   
##   
## 'Positive' Class : 0   
##

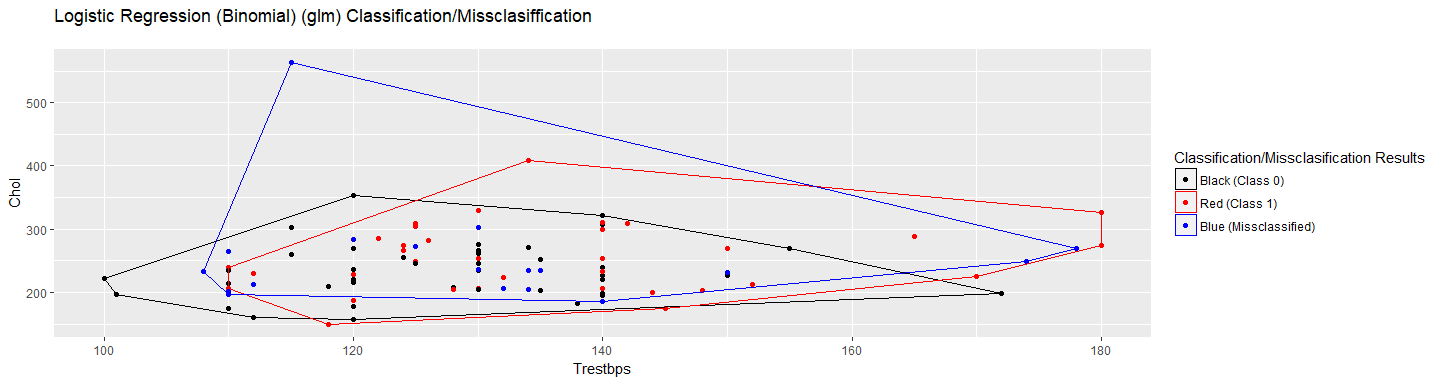
**print\_confusionm**(res.logistic2,   
 main = **paste0**("Confusion Matrix: ", model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



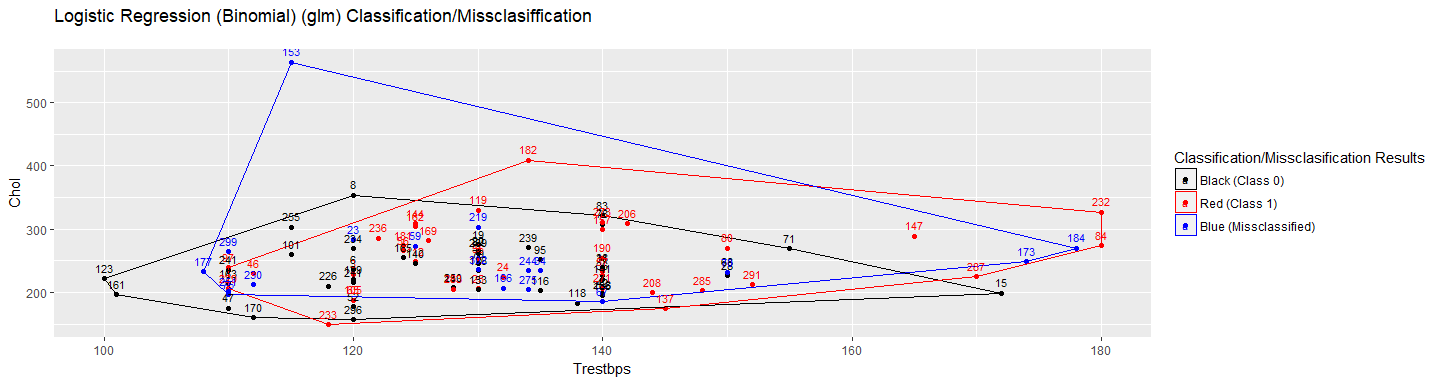
main <- **paste0**(model.name, " Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_logistic2,  
 Y.testData,  
 main, testData)



**plot\_results**(X.testData, Y.testData,test\_pred\_logistic2 ,   
 **paste0**(model.name," Classification/Missclasiffication\n"),   
 "Trestbps","Chol",labels = FALSE)



**plot\_results**(X.testData, Y.testData, test\_pred\_logistic2,   
 **paste0**(model.name," Classification/Missclasiffication\n"),  
 "Trestbps","Chol",labels = TRUE)



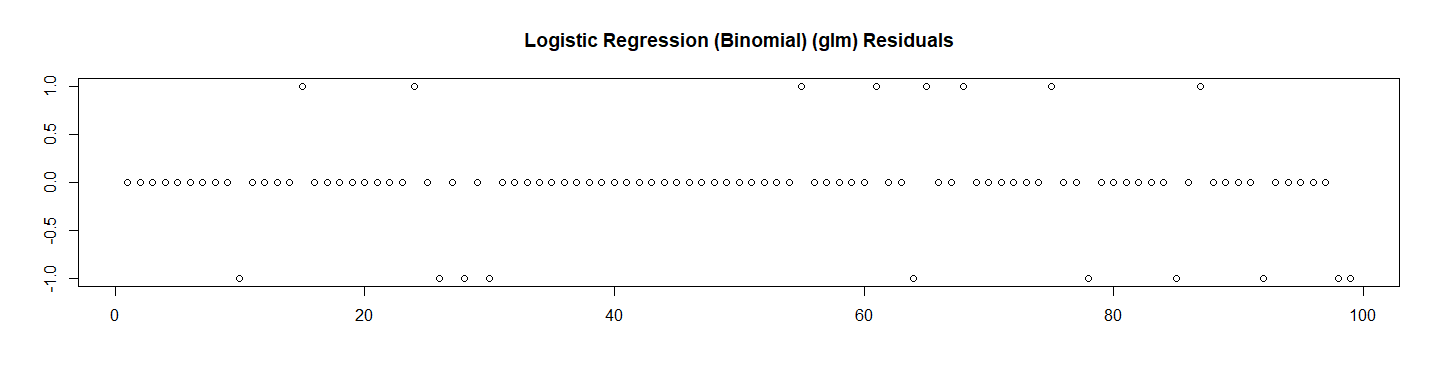
*# RMSE Resulting*  
(rmse\_logistic2 <- **sqrt**(**mean**((**as.numeric**(test\_pred\_logistic2)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4264

*# Residuals Plot*  
(r\_logistic2 <- (**as.numeric**(test\_pred\_logistic2)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0 0 0 0  
## [24] 1 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 0 0 -1 -1

**plot**(r\_logistic2,   
 main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_logistic2\_prob <- **predict**(reg.logistic2, newdata = X.testData, type ="prob")[2]  
  
res.logistic2.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**((test\_pred\_logistic2\_prob))))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 reg.logistic2**$**results**$**Accuracy,  
 "Prediction Accuracy in Test Set" =   
 res.logistic2**$**overall[1],  
 "RMSE Test"= rmse\_logistic2,  
 "ROC" = res.logistic2.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 6. Logistic Regression (Binomial) with Factor Variables (glm)

############################################  
*# 6. Logistic Regression (Binomial) with Factor Variables (glm)*   
model.name <- "Logistic Regression (Binomial) with Factor Variables (glm)"   
model.name4 <- model.name  
file.name <- **paste**(work.dir, "logistic4.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "logistic4-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
 (reg.logistic4 <- **train**(  
 **factor**(Disease) **~** **as.numeric**(Age) **+** **factor**(Sex) **+** **factor**(CP) **+**   
 **as.numeric**(Trestbps) **+** **as.numeric**(Chol) **+** **factor**(Fbs) **+**   
 **factor**(Restecg) **+** **as.numeric**(Thalach) **+** **factor**(Exang) **+**   
 **as.numeric**(Oldpeak) **+** **factor**(Slope) **+** **factor**(Ca) **+** **factor**(Thal),   
 data = trainData,  
 method = "glm",   
 family = "binomial"))  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(reg.logistic4, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Logistic Regression (Binomial) with Factor Variables (glm) Executed ... time:1.2"

Let’s see the results:

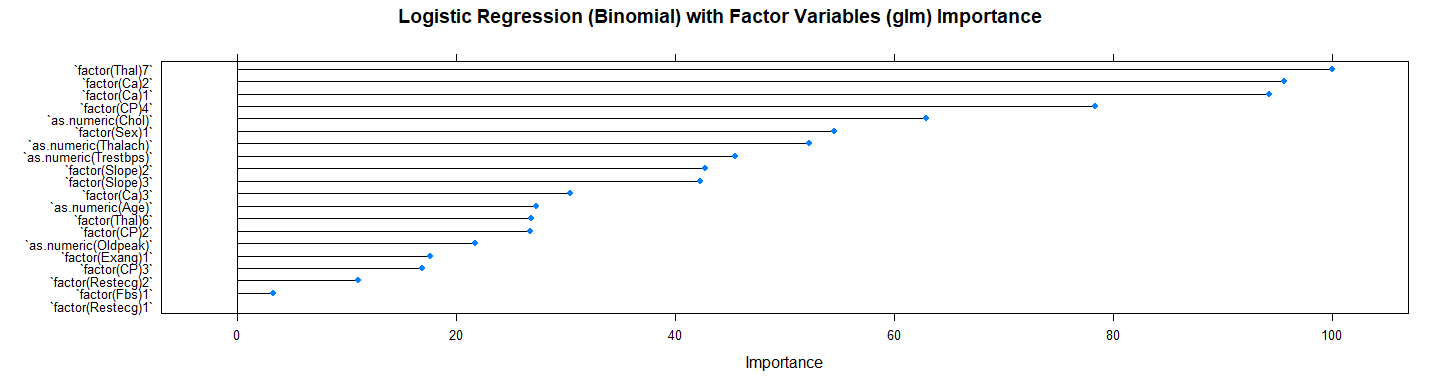
reg.logistic4

## Generalized Linear Model   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 198, 198, 198, 198, 198, 198, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8009 0.5977

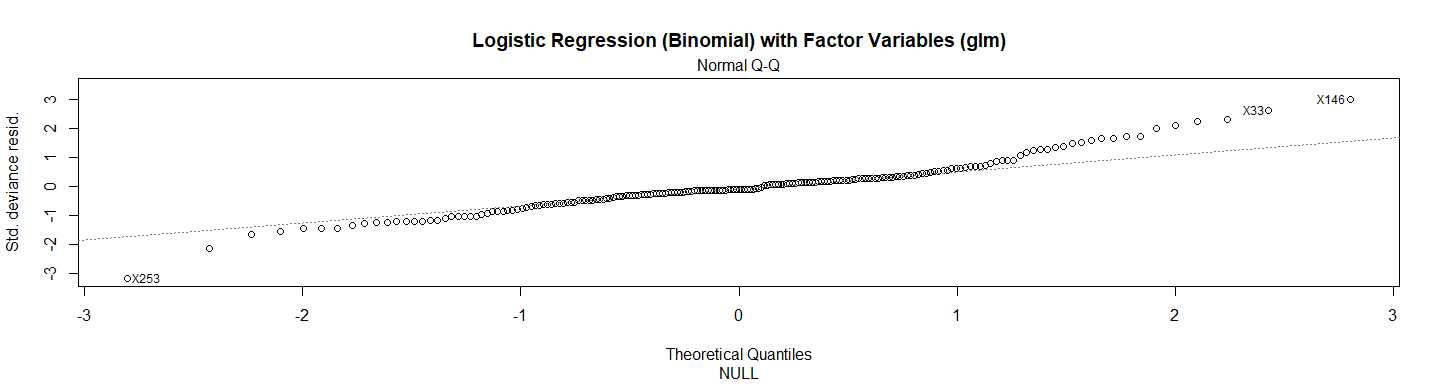
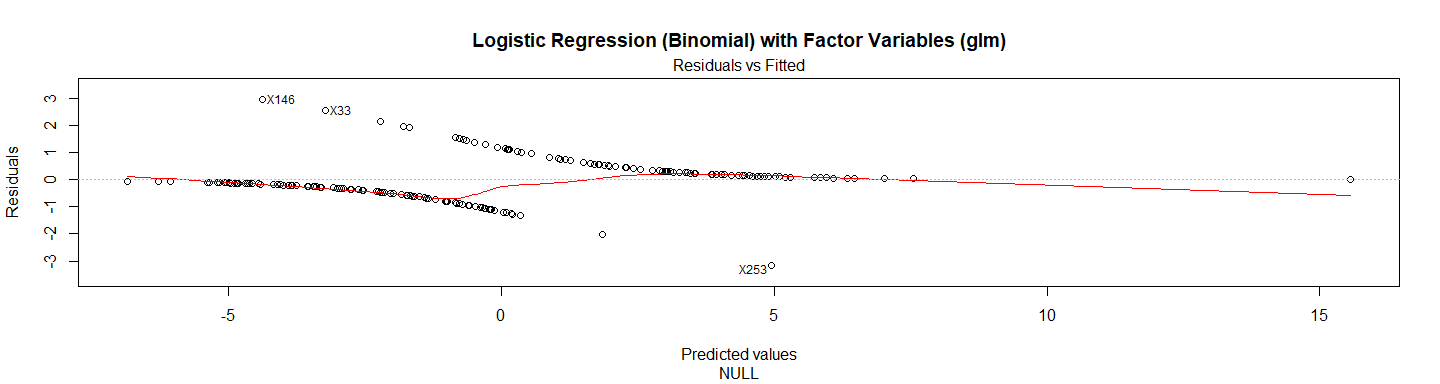
**summary**(reg.logistic4)

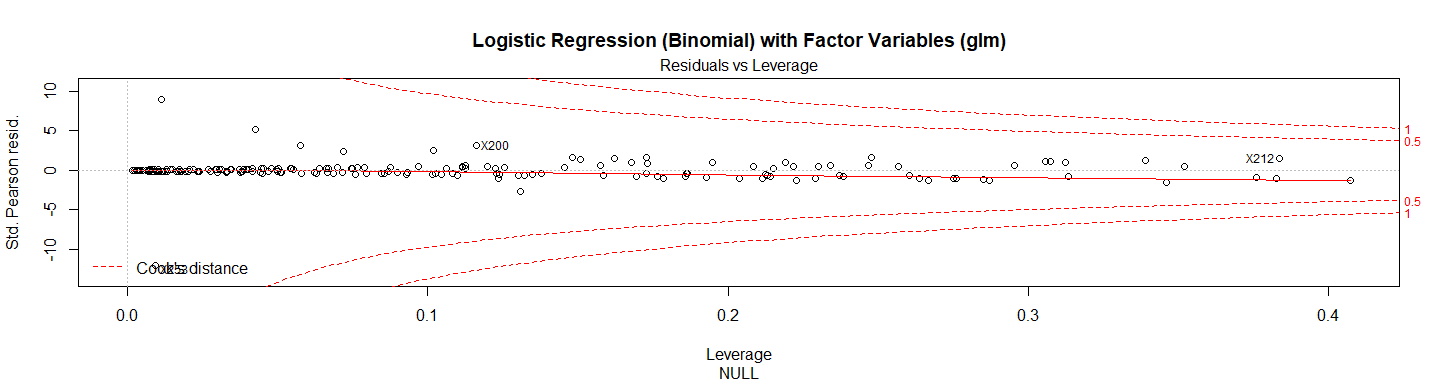
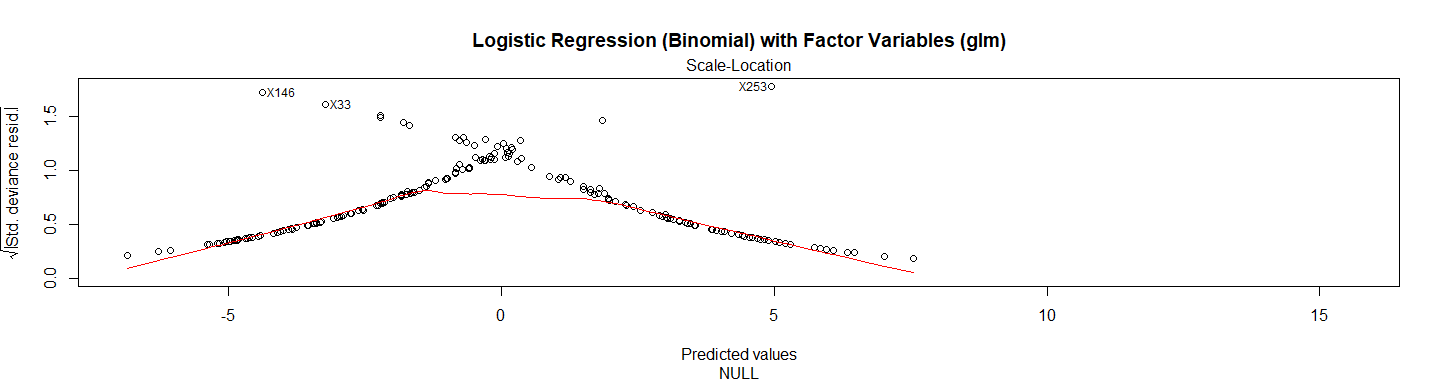
##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.151 -0.452 -0.113 0.296 2.962   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.14e+00 3.83e+00 -1.34 0.1799   
## `as.numeric(Age)` -3.12e-02 3.52e-02 -0.89 0.3761   
## `factor(Sex)1` 1.20e+00 6.83e-01 1.76 0.0780 .   
## `factor(CP)2` 8.91e-01 1.03e+00 0.87 0.3855   
## `factor(CP)3` -5.37e-01 9.76e-01 -0.55 0.5824   
## `factor(CP)4` 2.32e+00 9.17e-01 2.53 0.0115 \*   
## `as.numeric(Trestbps)` 2.20e-02 1.50e-02 1.47 0.1418   
## `as.numeric(Chol)` 1.22e-02 6.03e-03 2.03 0.0424 \*   
## `factor(Fbs)1` 8.63e-02 7.60e-01 0.11 0.9096   
## `factor(Restecg)1` 1.14e+01 1.46e+03 0.01 0.9937   
## `factor(Restecg)2` -1.86e-01 5.11e-01 -0.36 0.7162   
## `as.numeric(Thalach)` -2.49e-02 1.47e-02 -1.69 0.0914 .   
## `factor(Exang)1` 3.40e-01 5.91e-01 0.57 0.5656   
## `as.numeric(Oldpeak)` 2.12e-01 3.00e-01 0.71 0.4796   
## `factor(Slope)2` 9.02e-01 6.53e-01 1.38 0.1672   
## `factor(Slope)3` 1.60e+00 1.17e+00 1.37 0.1719   
## `factor(Ca)1` 2.02e+00 6.64e-01 3.04 0.0024 \*\*  
## `factor(Ca)2` 2.98e+00 9.67e-01 3.08 0.0021 \*\*  
## `factor(Ca)3` 1.27e+00 1.29e+00 0.99 0.3243   
## `factor(Thal)6` 8.48e-01 9.74e-01 0.87 0.3836   
## `factor(Thal)7` 1.85e+00 5.74e-01 3.22 0.0013 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 273.19 on 197 degrees of freedom  
## Residual deviance: 116.26 on 177 degrees of freedom  
## AIC: 158.3  
##   
## Number of Fisher Scoring iterations: 14

**plot**(**varImp**(reg.logistic4),   
 main = **paste0**(model.name," Importance"))



**plot**(reg.logistic4**$**finalModel,   
 main = model.name)





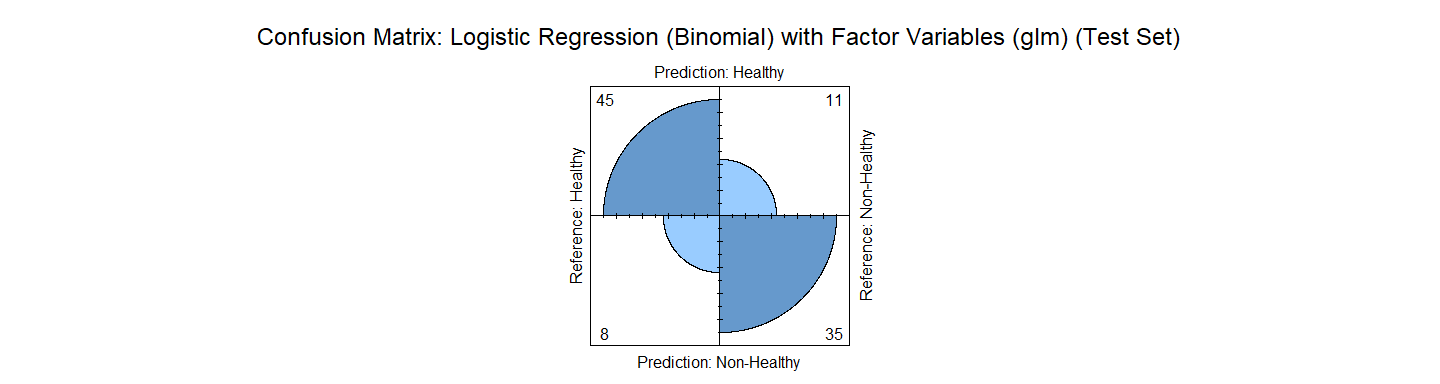
test\_pred\_logistic4 <- **predict**(reg.logistic4, newdata = X.testData)  
test\_pred\_logistic4

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 1 1 0 0 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 0 1 0  
## Levels: 0 1

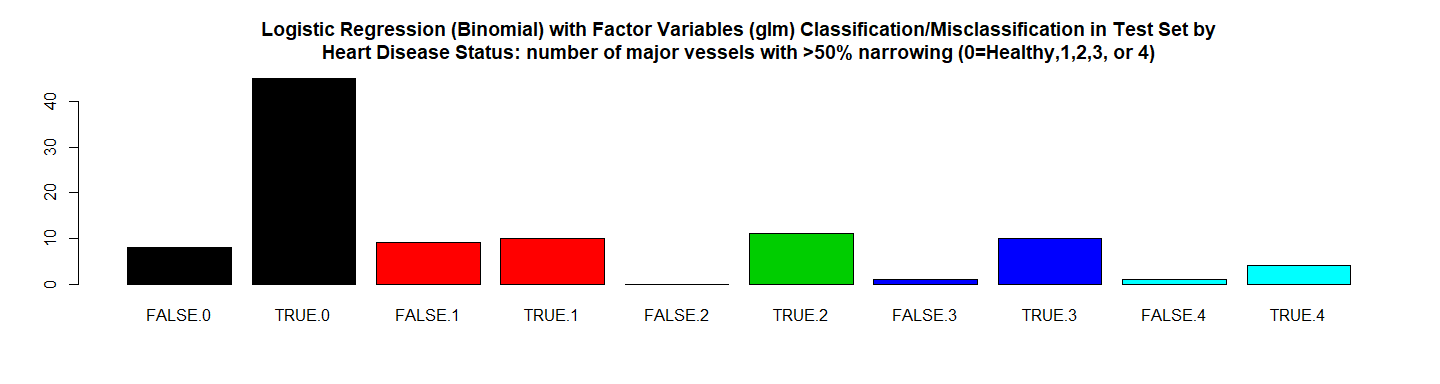
res.logistic4 <- **confusionMatrix**(test\_pred\_logistic4, Y.testData)   
res.logistic4

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 45 11  
## 1 8 35  
##   
## Accuracy : 0.808   
## 95% CI : (0.717, 0.88)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 1.35e-08   
##   
## Kappa : 0.613   
## Mcnemar's Test P-Value : 0.646   
##   
## Sensitivity : 0.849   
## Specificity : 0.761   
## Pos Pred Value : 0.804   
## Neg Pred Value : 0.814   
## Prevalence : 0.535   
## Detection Rate : 0.455   
## Detection Prevalence : 0.566   
## Balanced Accuracy : 0.805   
##   
## 'Positive' Class : 0   
##

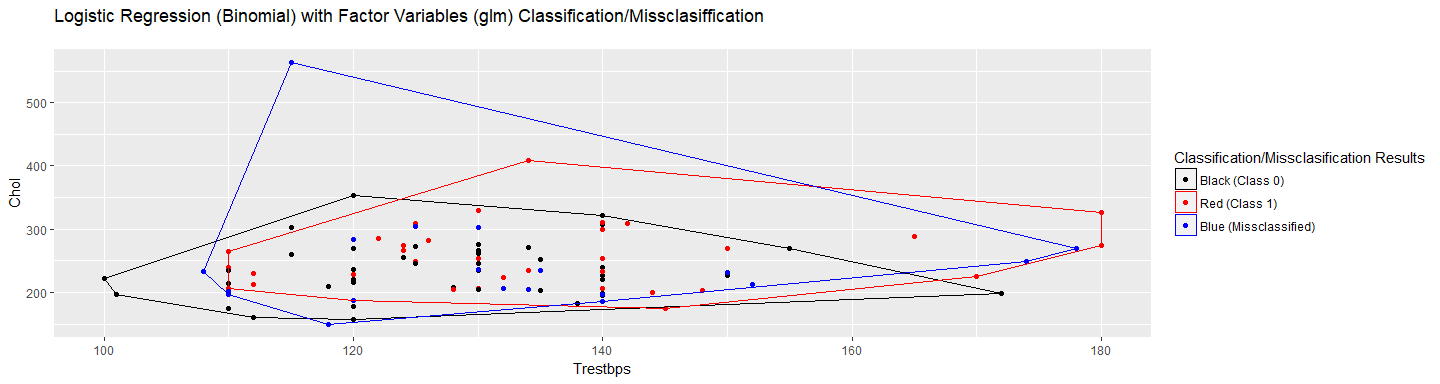
**print\_confusionm**(res.logistic4,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),  
 labels = **c**("Healthy", "Non-Healthy"))



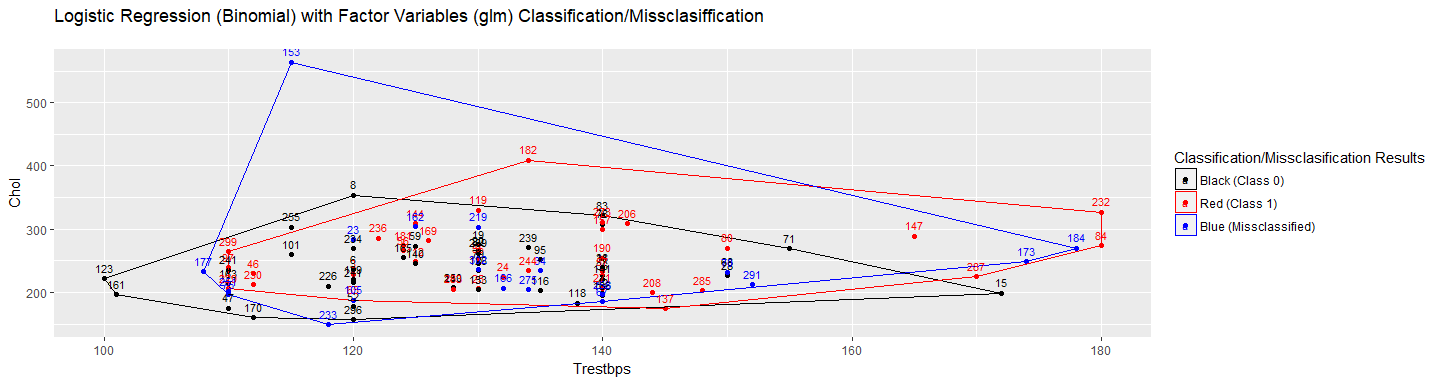
main <- **paste0**(model.name, " Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_logistic4,  
 Y.testData,  
 main, testData)



*#This function will be used to plot the results*   
  
**plot\_results**(X.testData, Y.testData,test\_pred\_logistic4,   
 **paste0**(model.name," Classification/Missclasiffication\n"),   
 "Trestbps","Chol",labels = FALSE)



**plot\_results**(X.testData, Y.testData, test\_pred\_logistic4,   
 **paste0**(model.name, " Classification/Missclasiffication\n"),  
 "Trestbps","Chol",labels = TRUE)



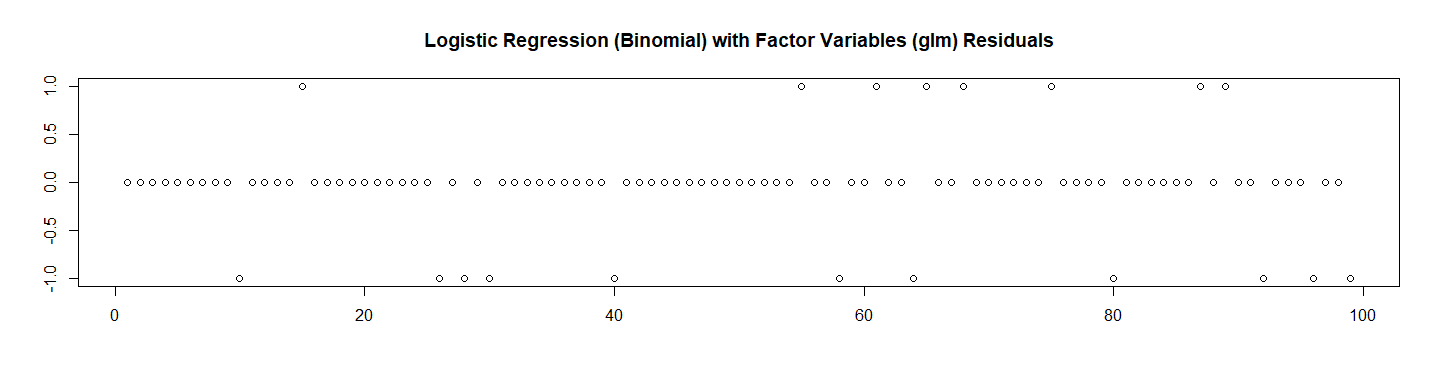
*# RMSE Resulting*  
(rmse\_logistic4 <- **sqrt**(**mean**((**as.numeric**(test\_pred\_logistic4)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4381

*# Residuals Plot*  
(r\_logistic4 <- (**as.numeric**(test\_pred\_logistic4)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0 0 0 0  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 -1 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 -1 0 0 0 0 0 0 1 0 1 0 0 -1  
## [93] 0 0 0 -1 0 0 -1

**plot**(r\_logistic4,   
 main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_logistic4\_prob <- **predict**(reg.logistic4, newdata = X.testData, type ="prob")[2]  
  
res.logistic4.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**((test\_pred\_logistic4\_prob))))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 reg.logistic4**$**results**$**Accuracy,  
 "Prediction Accuracy in Test Set" =   
 res.logistic4**$**overall[1],  
 "RMSE Test"= rmse\_logistic4,  
 "ROC" = res.logistic4.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 7. Linear Model with Stepwise Feature Selection (glmStepAIC)

############################################  
*# 7. Linear Model with Stepwise Feature Selection (glmStepAIC)*  
model.name <- "Linear Model with Stepwise Feature Selection (glmStepAIC)"  
model.name5 <- model.name  
file.name <- **paste**(work.dir, "glmStepAIC.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "glmStepAIC-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
 (glmStepAIC <- **train**(**factor**(Disease) **~** Age **+** Sex **+** CP **+** Trestbps **+** Chol **+** Fbs **+** Restecg   
 **+** Thalach **+** Exang **+** Oldpeak **+** Slope **+** Ca **+** Thal, data = trainData,  
 method = "glmStepAIC",  
 preProcess = **c**("center", "scale")))  
   
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name, " Executed ... time:", et))  
 **save**(glmStepAIC, file = file.name)  
 **save**(et, file = file.name.et)  
}

## Start: AIC=184.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 157 183  
## - Slope 1 157 183  
## - Sex 1 157 183  
## - Exang 1 158 184  
## - Chol 1 158 184  
## - Fbs 1 158 184  
## - Thalach 1 158 184  
## <none> 157 185  
## - Restecg 1 160 186  
## - Trestbps 1 162 188  
## - Age 1 163 189  
## - Thal 1 166 192  
## - CP 1 167 193  
## - Ca 1 170 196  
##   
## Step: AIC=182.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 157 181  
## - Slope 1 158 182  
## - Exang 1 158 182  
## - Chol 1 158 182  
## - Fbs 1 158 182  
## <none> 157 183  
## - Thalach 1 159 183  
## - Restecg 1 160 184  
## - Trestbps 1 162 186  
## - Age 1 163 187  
## - Thal 1 167 191  
## - CP 1 168 192  
## - Ca 1 171 195  
##   
## Step: AIC=181.2  
## .outcome ~ Age + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 158 180  
## - Chol 1 158 180  
## - Exang 1 159 181  
## - Fbs 1 159 181  
## <none> 157 181  
## - Thalach 1 159 181  
## - Restecg 1 161 183  
## - Trestbps 1 163 185  
## - Age 1 165 187  
## - CP 1 168 190  
## - Thal 1 170 192  
## - Ca 1 172 194  
##   
## Step: AIC=179.8  
## .outcome ~ Age + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 158 178  
## - Exang 1 160 180  
## <none> 158 180  
## - Fbs 1 161 181  
## - Thalach 1 161 181  
## - Restecg 1 162 182  
## - Trestbps 1 164 184  
## - Age 1 166 186  
## - CP 1 168 188  
## - Thal 1 172 192  
## - Ca 1 173 193  
##   
## Step: AIC=178.5  
## .outcome ~ Age + CP + Trestbps + Fbs + Restecg + Thalach + Exang +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 160 178  
## <none> 158 178  
## - Fbs 1 161 179  
## - Thalach 1 162 180  
## - Restecg 1 162 180  
## - Trestbps 1 165 183  
## - Age 1 166 184  
## - CP 1 170 188  
## - Thal 1 172 190  
## - Ca 1 174 192  
##   
## Step: AIC=178.2  
## .outcome ~ Age + CP + Trestbps + Fbs + Restecg + Thalach + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 160 178  
## - Fbs 1 163 179  
## - Restecg 1 164 180  
## - Thalach 1 166 182  
## - Trestbps 1 168 184  
## - Age 1 169 185  
## - Thal 1 174 190  
## - Ca 1 175 191  
## - CP 1 175 191  
## Start: AIC=144.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 116 142  
## - Fbs 1 116 142  
## - Slope 1 116 142  
## - Sex 1 117 143  
## - Trestbps 1 117 143  
## - Chol 1 118 144  
## <none> 116 144  
## - CP 1 120 146  
## - Exang 1 121 147  
## - Oldpeak 1 122 148  
## - Age 1 122 148  
## - Thal 1 126 152  
## - Ca 1 128 154  
## - Thalach 1 129 155  
##   
## Step: AIC=142.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 116 140  
## - Slope 1 116 140  
## - Sex 1 117 141  
## - Trestbps 1 117 141  
## <none> 116 142  
## - Chol 1 118 142  
## - CP 1 120 144  
## - Exang 1 121 145  
## - Oldpeak 1 122 146  
## - Age 1 122 146  
## - Thal 1 126 150  
## - Ca 1 128 152  
## - Thalach 1 129 153  
##   
## Step: AIC=140.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 116 138  
## - Sex 1 117 139  
## - Trestbps 1 117 139  
## <none> 116 140  
## - Chol 1 118 140  
## - CP 1 120 142  
## - Exang 1 121 143  
## - Oldpeak 1 122 144  
## - Age 1 122 144  
## - Thal 1 126 148  
## - Ca 1 128 150  
## - Thalach 1 130 152  
##   
## Step: AIC=138.4  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 117 137  
## - Trestbps 1 118 138  
## <none> 116 138  
## - Chol 1 118 138  
## - CP 1 120 140  
## - Exang 1 121 141  
## - Age 1 122 142  
## - Oldpeak 1 122 142  
## - Thal 1 126 146  
## - Ca 1 129 149  
## - Thalach 1 130 150  
##   
## Step: AIC=137.1  
## .outcome ~ Age + CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 119 137  
## - Trestbps 1 119 137  
## <none> 117 137  
## - CP 1 120 138  
## - Exang 1 122 140  
## - Oldpeak 1 123 141  
## - Age 1 124 142  
## - Ca 1 131 149  
## - Thalach 1 132 150  
## - Thal 1 132 150  
##   
## Step: AIC=136.6  
## .outcome ~ Age + CP + Trestbps + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 119 137  
## - Trestbps 1 121 137  
## - CP 1 122 138  
## - Oldpeak 1 124 140  
## - Exang 1 124 140  
## - Age 1 124 140  
## - Thalach 1 133 149  
## - Ca 1 133 149  
## - Thal 1 133 149  
## Start: AIC=137.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 110 136  
## - Age 1 110 136  
## - Exang 1 110 136  
## - Trestbps 1 110 136  
## - Fbs 1 111 137  
## - Thalach 1 111 137  
## <none> 110 138  
## - Restecg 1 112 138  
## - CP 1 113 139  
## - Chol 1 114 140  
## - Slope 1 115 141  
## - Thal 1 117 143  
## - Sex 1 118 144  
## - Ca 1 125 151  
##   
## Step: AIC=135.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 110 134  
## - Exang 1 110 134  
## - Trestbps 1 111 135  
## - Fbs 1 111 135  
## - Thalach 1 111 135  
## <none> 110 136  
## - Restecg 1 112 136  
## - Chol 1 114 138  
## - CP 1 114 138  
## - Thal 1 117 141  
## - Slope 1 118 142  
## - Sex 1 119 143  
## - Ca 1 127 151  
##   
## Step: AIC=133.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 110 132  
## - Trestbps 1 111 133  
## - Fbs 1 111 133  
## - Thalach 1 111 133  
## <none> 110 134  
## - Restecg 1 112 134  
## - Chol 1 114 136  
## - CP 1 114 136  
## - Slope 1 118 140  
## - Thal 1 118 140  
## - Sex 1 120 142  
## - Ca 1 128 150  
##   
## Step: AIC=132.5  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 112 132  
## - Fbs 1 112 132  
## - Thalach 1 112 132  
## <none> 110 132  
## - Restecg 1 113 133  
## - Chol 1 114 134  
## - CP 1 117 137  
## - Slope 1 119 139  
## - Thal 1 119 139  
## - Sex 1 120 140  
## - Ca 1 128 148  
##   
## Step: AIC=131.5  
## .outcome ~ Sex + CP + Chol + Fbs + Restecg + Thalach + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 112 130  
## - Restecg 1 113 131  
## - Thalach 1 113 131  
## <none> 112 132  
## - Chol 1 116 134  
## - CP 1 117 135  
## - Slope 1 120 138  
## - Sex 1 121 139  
## - Thal 1 121 139  
## - Ca 1 128 146  
##   
## Step: AIC=130.4  
## .outcome ~ Sex + CP + Chol + Restecg + Thalach + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## - Restecg 1 114 130  
## - Thalach 1 114 130  
## <none> 112 130  
## - Chol 1 117 133  
## - CP 1 118 134  
## - Slope 1 120 136  
## - Sex 1 121 137  
## - Thal 1 123 139  
## - Ca 1 129 145  
##   
## Step: AIC=129.7  
## .outcome ~ Sex + CP + Chol + Thalach + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 115 129  
## <none> 114 130  
## - Chol 1 117 131  
## - Slope 1 120 134  
## - CP 1 120 134  
## - Sex 1 121 135  
## - Thal 1 124 138  
## - Ca 1 131 145  
##   
## Step: AIC=129  
## .outcome ~ Sex + CP + Chol + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 115 129  
## - Chol 1 118 130  
## - CP 1 123 135  
## - Sex 1 123 135  
## - Slope 1 124 136  
## - Thal 1 130 142  
## - Ca 1 135 147  
## Start: AIC=155.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 128 154  
## - Age 1 128 154  
## - Fbs 1 128 154  
## - Restecg 1 128 154  
## - Exang 1 129 155  
## <none> 128 156  
## - Thal 1 130 156  
## - Sex 1 131 157  
## - Trestbps 1 132 158  
## - Slope 1 132 158  
## - Thalach 1 136 162  
## - Chol 1 137 163  
## - CP 1 137 163  
## - Ca 1 149 175  
##   
## Step: AIC=153.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 128 152  
## - Fbs 1 128 152  
## - Restecg 1 128 152  
## - Exang 1 129 153  
## <none> 128 154  
## - Thal 1 130 154  
## - Sex 1 131 155  
## - Trestbps 1 132 156  
## - Slope 1 133 157  
## - Thalach 1 136 160  
## - Chol 1 137 161  
## - CP 1 138 162  
## - Ca 1 152 176  
##   
## Step: AIC=151.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 128 150  
## - Restecg 1 128 150  
## - Exang 1 129 151  
## <none> 128 152  
## - Thal 1 130 152  
## - Trestbps 1 132 154  
## - Sex 1 132 154  
## - Slope 1 133 155  
## - Thalach 1 136 158  
## - Chol 1 137 159  
## - CP 1 138 160  
## - Ca 1 153 175  
##   
## Step: AIC=150.4  
## .outcome ~ Sex + CP + Trestbps + Chol + Restecg + Thalach + Exang +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 129 149  
## - Exang 1 130 150  
## <none> 128 150  
## - Thal 1 131 151  
## - Sex 1 132 152  
## - Trestbps 1 132 152  
## - Slope 1 134 154  
## - Thalach 1 137 157  
## - Chol 1 137 157  
## - CP 1 141 161  
## - Ca 1 154 174  
##   
## Step: AIC=149  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Exang + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 130 148  
## <none> 129 149  
## - Thal 1 132 150  
## - Sex 1 133 151  
## - Trestbps 1 134 152  
## - Slope 1 135 153  
## - Thalach 1 138 156  
## - Chol 1 139 157  
## - CP 1 141 159  
## - Ca 1 156 174  
##   
## Step: AIC=148.5  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 130 148  
## - Sex 1 134 150  
## - Thal 1 135 151  
## - Trestbps 1 135 151  
## - Slope 1 136 152  
## - Thalach 1 140 156  
## - Chol 1 141 157  
## - CP 1 147 163  
## - Ca 1 157 173  
## Start: AIC=137.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 110 136  
## - Restecg 1 110 136  
## - Oldpeak 1 110 136  
## - Thalach 1 111 137  
## - Sex 1 112 138  
## <none> 110 138  
## - Fbs 1 112 138  
## - Trestbps 1 112 138  
## - Slope 1 112 138  
## - Exang 1 113 139  
## - CP 1 114 140  
## - Chol 1 118 144  
## - Thal 1 120 146  
## - Ca 1 128 154  
##   
## Step: AIC=135.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 110 134  
## - Oldpeak 1 110 134  
## - Thalach 1 111 135  
## - Sex 1 112 136  
## <none> 110 136  
## - Fbs 1 112 136  
## - Trestbps 1 112 136  
## - Slope 1 112 136  
## - Exang 1 114 138  
## - CP 1 114 138  
## - Chol 1 118 142  
## - Thal 1 120 144  
## - Ca 1 128 152  
##   
## Step: AIC=134.4  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Thalach + Exang +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 111 133  
## - Thalach 1 111 133  
## - Sex 1 112 134  
## <none> 110 134  
## - Trestbps 1 113 135  
## - Fbs 1 113 135  
## - Slope 1 113 135  
## - Exang 1 114 136  
## - CP 1 115 137  
## - Chol 1 118 140  
## - Thal 1 122 144  
## - Ca 1 128 150  
##   
## Step: AIC=132.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Thalach + Exang +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 112 132  
## <none> 111 133  
## - Fbs 1 113 133  
## - Sex 1 114 134  
## - Trestbps 1 114 134  
## - Exang 1 115 135  
## - CP 1 117 137  
## - Slope 1 117 137  
## - Chol 1 118 138  
## - Thal 1 122 142  
## - Ca 1 130 150  
##   
## Step: AIC=132.2  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Exang + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 112 132  
## - Sex 1 114 132  
## - Fbs 1 115 133  
## - Trestbps 1 115 133  
## - Exang 1 118 136  
## - Slope 1 119 137  
## - CP 1 120 138  
## - Chol 1 120 138  
## - Thal 1 125 143  
## - Ca 1 137 155  
## Start: AIC=182.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 155 181  
## - Restecg 1 155 181  
## - Sex 1 155 181  
## - CP 1 155 181  
## - Oldpeak 1 155 181  
## - Slope 1 156 182  
## - Age 1 156 182  
## <none> 155 183  
## - Thalach 1 157 183  
## - Thal 1 158 184  
## - Chol 1 158 184  
## - Exang 1 161 187  
## - Trestbps 1 162 188  
## - Ca 1 169 195  
##   
## Step: AIC=180.9  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Restecg + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 155 179  
## - Sex 1 155 179  
## - Oldpeak 1 156 180  
## - Slope 1 156 180  
## - CP 1 156 180  
## - Age 1 156 180  
## <none> 155 181  
## - Thalach 1 157 181  
## - Thal 1 158 182  
## - Chol 1 158 182  
## - Exang 1 161 185  
## - Trestbps 1 162 186  
## - Ca 1 169 193  
##   
## Step: AIC=179.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 156 178  
## - Oldpeak 1 156 178  
## - CP 1 156 178  
## - Slope 1 156 178  
## - Age 1 156 178  
## <none> 155 179  
## - Thalach 1 157 179  
## - Thal 1 158 180  
## - Chol 1 158 180  
## - Exang 1 161 183  
## - Trestbps 1 163 185  
## - Ca 1 169 191  
##   
## Step: AIC=177.7  
## .outcome ~ Age + CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 156 176  
## - CP 1 156 176  
## - Oldpeak 1 157 177  
## - Age 1 157 177  
## <none> 156 178  
## - Thalach 1 158 178  
## - Chol 1 159 179  
## - Thal 1 161 181  
## - Exang 1 162 182  
## - Trestbps 1 163 183  
## - Ca 1 169 189  
##   
## Step: AIC=176.3  
## .outcome ~ Age + CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - CP 1 157 175  
## - Age 1 158 176  
## <none> 156 176  
## - Oldpeak 1 159 177  
## - Chol 1 159 177  
## - Thalach 1 160 178  
## - Thal 1 162 180  
## - Exang 1 163 181  
## - Trestbps 1 164 182  
## - Ca 1 170 188  
##   
## Step: AIC=174.8  
## .outcome ~ Age + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 158 174  
## <none> 157 175  
## - Chol 1 160 176  
## - Oldpeak 1 160 176  
## - Thalach 1 161 177  
## - Thal 1 162 178  
## - Trestbps 1 164 180  
## - Exang 1 166 182  
## - Ca 1 170 186  
##   
## Step: AIC=174.4  
## .outcome ~ Trestbps + Chol + Thalach + Exang + Oldpeak + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 158 174  
## - Chol 1 161 175  
## - Oldpeak 1 161 175  
## - Thalach 1 161 175  
## - Thal 1 164 178  
## - Trestbps 1 164 178  
## - Exang 1 168 182  
## - Ca 1 170 184  
## Start: AIC=107.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 79.3 105  
## - Thalach 1 79.3 105  
## - Restecg 1 79.7 106  
## - Age 1 79.9 106  
## - Trestbps 1 80.3 106  
## <none> 79.3 107  
## - Sex 1 82.5 108  
## - Slope 1 83.8 110  
## - Chol 1 86.7 113  
## - Fbs 1 87.6 114  
## - Ca 1 87.7 114  
## - Exang 1 90.7 117  
## - CP 1 100.5 126  
## - Thal 1 102.7 129  
##   
## Step: AIC=105.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 79.3 103  
## - Restecg 1 79.8 104  
## - Age 1 79.9 104  
## - Trestbps 1 80.4 104  
## <none> 79.3 105  
## - Sex 1 82.7 107  
## - Slope 1 84.1 108  
## - Chol 1 86.9 111  
## - Fbs 1 87.9 112  
## - Ca 1 87.9 112  
## - Exang 1 92.3 116  
## - CP 1 100.5 124  
## - Thal 1 103.9 128  
##   
## Step: AIC=103.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 79.8 102  
## - Age 1 80.1 102  
## - Trestbps 1 80.4 102  
## <none> 79.3 103  
## - Sex 1 82.7 105  
## - Slope 1 84.2 106  
## - Chol 1 87.1 109  
## - Ca 1 87.9 110  
## - Fbs 1 88.2 110  
## - Exang 1 92.8 115  
## - CP 1 103.2 125  
## - Thal 1 103.9 126  
##   
## Step: AIC=101.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Exang + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 80.6 101  
## - Trestbps 1 80.6 101  
## <none> 79.8 102  
## - Sex 1 83.2 103  
## - Slope 1 84.4 104  
## - Chol 1 87.3 107  
## - Fbs 1 88.3 108  
## - Ca 1 88.9 109  
## - Exang 1 93.3 113  
## - CP 1 103.3 123  
## - Thal 1 104.5 124  
##   
## Step: AIC=100.6  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Exang + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 81.1 99.1  
## <none> 80.6 100.6  
## - Sex 1 84.0 102.0  
## - Slope 1 84.9 102.9  
## - Chol 1 87.3 105.3  
## - Fbs 1 89.5 107.5  
## - Ca 1 89.7 107.7  
## - Exang 1 96.3 114.3  
## - CP 1 103.3 121.3  
## - Thal 1 105.2 123.2  
##   
## Step: AIC=99.1  
## .outcome ~ Sex + CP + Chol + Fbs + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 81.1 99.1  
## - Sex 1 84.4 100.4  
## - Slope 1 85.0 101.0  
## - Chol 1 88.0 104.0  
## - Fbs 1 89.6 105.6  
## - Ca 1 90.6 106.6  
## - Exang 1 96.9 112.9  
## - CP 1 103.5 119.5  
## - Thal 1 105.8 121.8  
## Start: AIC=131.9  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 104 130  
## - Oldpeak 1 105 131  
## - Restecg 1 106 132  
## <none> 104 132  
## - Chol 1 106 132  
## - Trestbps 1 107 133  
## - Slope 1 107 133  
## - Age 1 107 133  
## - Fbs 1 108 134  
## - Exang 1 108 134  
## - Sex 1 111 137  
## - Ca 1 117 143  
## - CP 1 126 152  
## - Thal 1 126 152  
##   
## Step: AIC=129.9  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 105 129  
## <none> 104 130  
## - Restecg 1 106 130  
## - Chol 1 106 130  
## - Trestbps 1 107 131  
## - Slope 1 107 131  
## - Fbs 1 108 132  
## - Exang 1 108 132  
## - Age 1 109 133  
## - Sex 1 112 136  
## - Ca 1 118 142  
## - Thal 1 128 152  
## - CP 1 130 154  
##   
## Step: AIC=129.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 105 129  
## - Restecg 1 107 129  
## - Trestbps 1 108 130  
## - Chol 1 108 130  
## - Age 1 110 132  
## - Fbs 1 110 132  
## - Exang 1 111 133  
## - Slope 1 113 135  
## - Sex 1 114 136  
## - Ca 1 121 143  
## - Thal 1 129 151  
## - CP 1 130 152  
## Start: AIC=182  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 154 180  
## - Trestbps 1 154 180  
## - Slope 1 154 180  
## - Fbs 1 155 181  
## <none> 154 182  
## - Sex 1 156 182  
## - Thalach 1 156 182  
## - Oldpeak 1 158 184  
## - Thal 1 159 185  
## - CP 1 160 186  
## - Restecg 1 161 187  
## - Age 1 161 187  
## - Chol 1 163 189  
## - Ca 1 184 210  
##   
## Step: AIC=180.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 154 178  
## - Trestbps 1 154 178  
## - Fbs 1 155 179  
## <none> 154 180  
## - Sex 1 156 180  
## - Thalach 1 156 180  
## - Oldpeak 1 158 182  
## - Thal 1 159 183  
## - CP 1 160 184  
## - Age 1 161 185  
## - Restecg 1 161 185  
## - Chol 1 163 187  
## - Ca 1 184 208  
##   
## Step: AIC=178.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 154 176  
## - Fbs 1 156 178  
## <none> 154 178  
## - Sex 1 156 178  
## - Thalach 1 157 179  
## - CP 1 160 182  
## - Thal 1 160 182  
## - Oldpeak 1 161 183  
## - Age 1 161 183  
## - Restecg 1 161 183  
## - Chol 1 163 185  
## - Ca 1 184 206  
##   
## Step: AIC=176.5  
## .outcome ~ Age + Sex + CP + Chol + Fbs + Restecg + Thalach +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 156 176  
## <none> 154 176  
## - Sex 1 156 176  
## - Thalach 1 157 177  
## - CP 1 160 180  
## - Age 1 161 181  
## - Oldpeak 1 161 181  
## - Thal 1 161 181  
## - Restecg 1 162 182  
## - Chol 1 164 184  
## - Ca 1 184 204  
##   
## Step: AIC=175.7  
## .outcome ~ Age + Sex + CP + Chol + Restecg + Thalach + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 158 176  
## <none> 156 176  
## - Sex 1 158 176  
## - CP 1 161 179  
## - Age 1 162 180  
## - Thal 1 163 181  
## - Restecg 1 163 181  
## - Oldpeak 1 164 182  
## - Chol 1 166 184  
## - Ca 1 186 204  
##   
## Step: AIC=175.5  
## .outcome ~ Age + Sex + CP + Chol + Restecg + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 158 176  
## - Sex 1 160 176  
## - Age 1 162 178  
## - Restecg 1 165 181  
## - Thal 1 166 182  
## - Chol 1 167 183  
## - Oldpeak 1 167 183  
## - CP 1 168 184  
## - Ca 1 187 203  
## Start: AIC=139.9  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 112 138  
## - Age 1 112 138  
## - Exang 1 113 139  
## - Fbs 1 113 139  
## - Thal 1 113 139  
## - Restecg 1 113 139  
## <none> 112 140  
## - Trestbps 1 114 140  
## - Slope 1 117 143  
## - Chol 1 117 143  
## - Sex 1 117 143  
## - Oldpeak 1 119 145  
## - CP 1 126 152  
## - Ca 1 130 156  
##   
## Step: AIC=138  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 112 136  
## - Exang 1 113 137  
## - Fbs 1 113 137  
## - Thal 1 113 137  
## - Restecg 1 113 137  
## <none> 112 138  
## - Trestbps 1 114 138  
## - Chol 1 117 141  
## - Slope 1 117 141  
## - Sex 1 117 141  
## - Oldpeak 1 120 144  
## - CP 1 128 152  
## - Ca 1 130 154  
##   
## Step: AIC=136.2  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Exang +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 113 135  
## - Fbs 1 113 135  
## - Thal 1 113 135  
## - Restecg 1 114 136  
## - Trestbps 1 114 136  
## <none> 112 136  
## - Chol 1 117 139  
## - Slope 1 117 139  
## - Sex 1 118 140  
## - Oldpeak 1 120 142  
## - CP 1 128 150  
## - Ca 1 131 153  
##   
## Step: AIC=135  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Oldpeak +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 114 134  
## - Restecg 1 114 134  
## - Thal 1 115 135  
## <none> 113 135  
## - Trestbps 1 116 136  
## - Chol 1 118 138  
## - Sex 1 118 138  
## - Slope 1 119 139  
## - Oldpeak 1 121 141  
## - Ca 1 132 152  
## - CP 1 135 155  
##   
## Step: AIC=133.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Restecg + Oldpeak + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 115 133  
## <none> 114 134  
## - Trestbps 1 116 134  
## - Thal 1 117 135  
## - Chol 1 118 136  
## - Sex 1 118 136  
## - Slope 1 119 137  
## - Oldpeak 1 121 139  
## - Ca 1 132 150  
## - CP 1 140 158  
##   
## Step: AIC=133.2  
## .outcome ~ Sex + CP + Trestbps + Chol + Oldpeak + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 115 133  
## - Trestbps 1 118 134  
## - Thal 1 118 134  
## - Chol 1 119 135  
## - Sex 1 119 135  
## - Slope 1 120 136  
## - Oldpeak 1 122 138  
## - Ca 1 135 151  
## - CP 1 144 160  
## Start: AIC=151.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 124 150  
## - Slope 1 124 150  
## - Sex 1 124 150  
## - Fbs 1 124 150  
## - Restecg 1 124 150  
## - Ca 1 125 151  
## <none> 124 152  
## - Trestbps 1 126 152  
## - Exang 1 126 152  
## - Thalach 1 127 153  
## - Oldpeak 1 129 155  
## - Chol 1 129 155  
## - CP 1 134 160  
## - Thal 1 148 174  
##   
## Step: AIC=149.6  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 124 148  
## - Sex 1 124 148  
## - Fbs 1 124 148  
## - Restecg 1 124 148  
## - Ca 1 125 149  
## <none> 124 150  
## - Exang 1 126 150  
## - Trestbps 1 126 150  
## - Thalach 1 128 152  
## - Oldpeak 1 129 153  
## - Chol 1 129 153  
## - CP 1 134 158  
## - Thal 1 149 173  
##   
## Step: AIC=147.8  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 124 146  
## - Fbs 1 124 146  
## - Restecg 1 124 146  
## - Ca 1 125 147  
## <none> 124 148  
## - Trestbps 1 126 148  
## - Exang 1 126 148  
## - Thalach 1 128 150  
## - Oldpeak 1 130 152  
## - Chol 1 131 153  
## - CP 1 134 156  
## - Thal 1 149 171  
##   
## Step: AIC=146.3  
## .outcome ~ CP + Trestbps + Chol + Fbs + Restecg + Thalach + Exang +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 125 145  
## - Restecg 1 125 145  
## - Ca 1 126 146  
## <none> 124 146  
## - Trestbps 1 126 146  
## - Exang 1 127 147  
## - Thalach 1 128 148  
## - Chol 1 131 151  
## - Oldpeak 1 131 151  
## - CP 1 134 154  
## - Thal 1 158 178  
##   
## Step: AIC=144.8  
## .outcome ~ CP + Trestbps + Chol + Restecg + Thalach + Exang +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 126 144  
## - Ca 1 126 144  
## <none> 125 145  
## - Trestbps 1 127 145  
## - Exang 1 128 146  
## - Thalach 1 129 147  
## - Chol 1 131 149  
## - Oldpeak 1 132 150  
## - CP 1 134 152  
## - Thal 1 159 177  
##   
## Step: AIC=143.5  
## .outcome ~ CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Ca 1 127 143  
## <none> 126 144  
## - Exang 1 128 144  
## - Trestbps 1 128 144  
## - Thalach 1 131 147  
## - Chol 1 132 148  
## - Oldpeak 1 134 150  
## - CP 1 135 151  
## - Thal 1 159 175  
##   
## Step: AIC=142.7  
## .outcome ~ CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Thal  
##   
## Df Deviance AIC  
## <none> 127 143  
## - Exang 1 129 143  
## - Trestbps 1 130 144  
## - Thalach 1 134 148  
## - Chol 1 134 148  
## - CP 1 138 152  
## - Oldpeak 1 139 153  
## - Thal 1 165 179  
## Start: AIC=138.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 111 137  
## - Thalach 1 111 137  
## - Age 1 111 137  
## - Restecg 1 112 138  
## - Chol 1 113 139  
## - Oldpeak 1 113 139  
## <none> 111 139  
## - Trestbps 1 113 139  
## - Sex 1 113 139  
## - Fbs 1 116 142  
## - Slope 1 118 144  
## - Thal 1 121 147  
## - CP 1 124 150  
## - Ca 1 126 152  
##   
## Step: AIC=136.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 111 135  
## - Age 1 111 135  
## - Restecg 1 112 136  
## - Chol 1 113 137  
## <none> 111 137  
## - Oldpeak 1 113 137  
## - Trestbps 1 113 137  
## - Sex 1 113 137  
## - Fbs 1 116 140  
## - Slope 1 118 142  
## - Thal 1 122 146  
## - CP 1 126 150  
## - Ca 1 126 150  
##   
## Step: AIC=134.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 111 133  
## - Restecg 1 112 134  
## - Chol 1 113 135  
## <none> 111 135  
## - Oldpeak 1 113 135  
## - Trestbps 1 113 135  
## - Sex 1 113 135  
## - Fbs 1 116 138  
## - Slope 1 119 141  
## - Thal 1 122 144  
## - Ca 1 126 148  
## - CP 1 130 152  
##   
## Step: AIC=132.8  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Oldpeak +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 112 132  
## - Chol 1 113 133  
## <none> 111 133  
## - Oldpeak 1 113 133  
## - Trestbps 1 113 133  
## - Sex 1 113 133  
## - Fbs 1 116 136  
## - Slope 1 119 139  
## - Thal 1 122 142  
## - Ca 1 127 147  
## - CP 1 130 150  
##   
## Step: AIC=132.1  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Oldpeak + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 113 131  
## <none> 112 132  
## - Trestbps 1 114 132  
## - Sex 1 114 132  
## - Oldpeak 1 114 132  
## - Fbs 1 117 135  
## - Slope 1 120 138  
## - Thal 1 123 141  
## - Ca 1 128 146  
## - CP 1 131 149  
##   
## Step: AIC=131.4  
## .outcome ~ Sex + CP + Trestbps + Fbs + Oldpeak + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## - Sex 1 115 131  
## <none> 113 131  
## - Trestbps 1 116 132  
## - Oldpeak 1 116 132  
## - Fbs 1 119 135  
## - Slope 1 120 136  
## - Thal 1 124 140  
## - Ca 1 130 146  
## - CP 1 132 148  
##   
## Step: AIC=130.8  
## .outcome ~ CP + Trestbps + Fbs + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 116 130  
## <none> 115 131  
## - Oldpeak 1 119 133  
## - Fbs 1 119 133  
## - Slope 1 121 135  
## - Ca 1 131 145  
## - CP 1 132 146  
## - Thal 1 133 147  
##   
## Step: AIC=130.3  
## .outcome ~ CP + Fbs + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 116 130  
## - Fbs 1 120 132  
## - Oldpeak 1 120 132  
## - Slope 1 121 133  
## - CP 1 133 145  
## - Ca 1 133 145  
## - Thal 1 134 146  
## Start: AIC=156  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 128 154  
## - Restecg 1 128 154  
## - Fbs 1 128 154  
## - Exang 1 128 154  
## - Age 1 128 154  
## - Slope 1 129 155  
## - Thalach 1 129 155  
## - Trestbps 1 129 155  
## <none> 128 156  
## - Chol 1 132 158  
## - Thal 1 134 160  
## - Sex 1 137 163  
## - Ca 1 147 173  
## - CP 1 150 176  
##   
## Step: AIC=154.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 128 152  
## - Fbs 1 128 152  
## - Exang 1 128 152  
## - Age 1 128 152  
## - Slope 1 129 153  
## - Thalach 1 129 153  
## - Trestbps 1 129 153  
## <none> 128 154  
## - Chol 1 132 156  
## - Thal 1 134 158  
## - Sex 1 138 162  
## - Ca 1 148 172  
## - CP 1 151 175  
##   
## Step: AIC=152.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 128 150  
## - Exang 1 128 150  
## - Age 1 128 150  
## - Slope 1 129 151  
## - Thalach 1 129 151  
## - Trestbps 1 129 151  
## <none> 128 152  
## - Chol 1 132 154  
## - Thal 1 134 156  
## - Sex 1 138 160  
## - Ca 1 148 170  
## - CP 1 151 173  
##   
## Step: AIC=150.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 128 148  
## - Age 1 128 148  
## - Thalach 1 129 149  
## - Trestbps 1 129 149  
## - Slope 1 129 149  
## <none> 128 150  
## - Chol 1 132 152  
## - Thal 1 134 154  
## - Sex 1 138 158  
## - Ca 1 148 168  
## - CP 1 151 171  
##   
## Step: AIC=148.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 129 147  
## - Trestbps 1 129 147  
## - Thalach 1 129 147  
## - Slope 1 130 148  
## <none> 128 148  
## - Chol 1 132 150  
## - Thal 1 134 152  
## - Sex 1 139 157  
## - Ca 1 148 166  
## - CP 1 154 172  
##   
## Step: AIC=146.6  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 129 145  
## - Slope 1 130 146  
## - Thalach 1 130 146  
## <none> 129 147  
## - Chol 1 134 150  
## - Thal 1 134 150  
## - Sex 1 139 155  
## - Ca 1 152 168  
## - CP 1 154 170  
##   
## Step: AIC=145.2  
## .outcome ~ Sex + CP + Chol + Thalach + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 131 145  
## - Thalach 1 131 145  
## <none> 129 145  
## - Chol 1 134 148  
## - Thal 1 135 149  
## - Sex 1 140 154  
## - Ca 1 152 166  
## - CP 1 155 169  
##   
## Step: AIC=144.6  
## .outcome ~ Sex + CP + Chol + Thalach + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 131 145  
## - Thalach 1 134 146  
## - Chol 1 134 146  
## - Thal 1 138 150  
## - Sex 1 141 153  
## - Ca 1 154 166  
## - CP 1 155 167  
## Start: AIC=150.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 122 148  
## - Thalach 1 122 148  
## - Chol 1 122 148  
## - Restecg 1 122 148  
## - Fbs 1 123 149  
## - Trestbps 1 123 149  
## - Sex 1 124 150  
## <none> 122 150  
## - Oldpeak 1 125 151  
## - Exang 1 125 151  
## - Age 1 126 152  
## - Thal 1 130 156  
## - CP 1 144 170  
## - Ca 1 148 174  
##   
## Step: AIC=148.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 122 146  
## - Restecg 1 122 146  
## - Chol 1 122 146  
## - Fbs 1 123 147  
## - Trestbps 1 123 147  
## - Sex 1 124 148  
## <none> 122 148  
## - Oldpeak 1 125 149  
## - Exang 1 125 149  
## - Age 1 126 150  
## - Thal 1 130 154  
## - CP 1 144 168  
## - Ca 1 148 172  
##   
## Step: AIC=146.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 122 144  
## - Chol 1 122 144  
## - Fbs 1 123 145  
## - Trestbps 1 123 145  
## - Sex 1 124 146  
## <none> 122 146  
## - Oldpeak 1 126 148  
## - Exang 1 126 148  
## - Age 1 126 148  
## - Thal 1 131 153  
## - CP 1 147 169  
## - Ca 1 151 173  
##   
## Step: AIC=144.4  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 122 142  
## - Fbs 1 123 143  
## - Trestbps 1 123 143  
## <none> 122 144  
## - Sex 1 124 144  
## - Oldpeak 1 126 146  
## - Exang 1 126 146  
## - Age 1 127 147  
## - Thal 1 131 151  
## - CP 1 147 167  
## - Ca 1 152 172  
##   
## Step: AIC=142.5  
## .outcome ~ Age + Sex + CP + Trestbps + Fbs + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 123 141  
## - Trestbps 1 123 141  
## <none> 122 142  
## - Sex 1 125 143  
## - Oldpeak 1 126 144  
## - Exang 1 126 144  
## - Age 1 127 145  
## - Thal 1 131 149  
## - CP 1 148 166  
## - Ca 1 154 172  
##   
## Step: AIC=140.8  
## .outcome ~ Age + Sex + CP + Trestbps + Exang + Oldpeak + Ca +   
## Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 123 139  
## <none> 123 141  
## - Sex 1 126 142  
## - Oldpeak 1 126 142  
## - Exang 1 127 143  
## - Age 1 127 143  
## - Thal 1 132 148  
## - CP 1 148 164  
## - Ca 1 157 173  
##   
## Step: AIC=139.2  
## .outcome ~ Age + Sex + CP + Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 123 139  
## - Sex 1 126 140  
## - Oldpeak 1 126 140  
## - Exang 1 127 141  
## - Age 1 129 143  
## - Thal 1 133 147  
## - CP 1 149 163  
## - Ca 1 157 171  
## Start: AIC=107.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 79.8 106  
## - Age 1 80.3 106  
## - Restecg 1 80.7 107  
## - Oldpeak 1 81.1 107  
## - Thalach 1 81.1 107  
## <none> 79.7 108  
## - Chol 1 82.8 109  
## - Slope 1 83.1 109  
## - Sex 1 85.0 111  
## - Trestbps 1 85.3 111  
## - Fbs 1 88.8 115  
## - CP 1 93.1 119  
## - Ca 1 104.4 130  
## - Thal 1 107.6 134  
##   
## Step: AIC=105.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 80.4 104  
## - Restecg 1 81.1 105  
## - Oldpeak 1 81.1 105  
## - Thalach 1 81.1 105  
## <none> 79.8 106  
## - Slope 1 83.1 107  
## - Chol 1 83.2 107  
## - Sex 1 85.2 109  
## - Trestbps 1 85.5 110  
## - Fbs 1 88.8 113  
## - CP 1 94.3 118  
## - Ca 1 104.8 129  
## - Thal 1 107.8 132  
##   
## Step: AIC=104.4  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 81.3 103  
## - Restecg 1 82.0 104  
## <none> 80.4 104  
## - Thalach 1 83.6 106  
## - Chol 1 84.3 106  
## - Slope 1 84.4 106  
## - Sex 1 85.4 107  
## - Trestbps 1 88.0 110  
## - Fbs 1 89.2 111  
## - CP 1 94.5 116  
## - Thal 1 107.8 130  
## - Ca 1 112.9 135  
##   
## Step: AIC=103.3  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Restecg + Thalach +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 83.1 103  
## <none> 81.3 103  
## - Thalach 1 84.8 105  
## - Chol 1 85.2 105  
## - Sex 1 87.6 108  
## - Trestbps 1 89.7 110  
## - Fbs 1 89.8 110  
## - Slope 1 92.5 112  
## - CP 1 97.2 117  
## - Thal 1 107.9 128  
## - Ca 1 115.1 135  
##   
## Step: AIC=103.1  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Thalach + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 83.1 103  
## - Thalach 1 85.8 104  
## - Chol 1 86.1 104  
## - Sex 1 88.6 107  
## - Trestbps 1 90.2 108  
## - Fbs 1 90.6 109  
## - Slope 1 93.7 112  
## - CP 1 99.5 118  
## - Thal 1 110.6 129  
## - Ca 1 118.0 136  
## Start: AIC=144.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 116 142  
## - Exang 1 116 142  
## - Oldpeak 1 116 142  
## - Sex 1 118 144  
## - Age 1 118 144  
## <none> 116 144  
## - Slope 1 119 145  
## - Fbs 1 120 146  
## - Trestbps 1 120 146  
## - Thalach 1 122 148  
## - CP 1 124 150  
## - Chol 1 125 151  
## - Thal 1 133 159  
## - Ca 1 138 164  
##   
## Step: AIC=142.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 116 140  
## - Oldpeak 1 116 140  
## - Sex 1 118 142  
## - Age 1 118 142  
## <none> 116 142  
## - Slope 1 119 143  
## - Fbs 1 120 144  
## - Trestbps 1 120 144  
## - Thalach 1 122 146  
## - CP 1 124 148  
## - Chol 1 125 149  
## - Thal 1 133 157  
## - Ca 1 138 162  
##   
## Step: AIC=140.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 116 138  
## - Sex 1 118 140  
## - Age 1 118 140  
## <none> 116 140  
## - Slope 1 119 141  
## - Fbs 1 120 142  
## - Trestbps 1 120 142  
## - Thalach 1 122 144  
## - CP 1 125 147  
## - Chol 1 126 148  
## - Thal 1 136 158  
## - Ca 1 139 161  
##   
## Step: AIC=138.5  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 118 138  
## <none> 116 138  
## - Age 1 119 139  
## - Fbs 1 120 140  
## - Trestbps 1 121 141  
## - Slope 1 123 143  
## - Thalach 1 123 143  
## - Chol 1 126 146  
## - CP 1 127 147  
## - Thal 1 136 156  
## - Ca 1 142 162  
##   
## Step: AIC=138.4  
## .outcome ~ Age + CP + Trestbps + Chol + Fbs + Thalach + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 118 138  
## - Age 1 122 140  
## - Fbs 1 122 140  
## - Trestbps 1 123 141  
## - Slope 1 124 142  
## - Thalach 1 126 144  
## - Chol 1 126 144  
## - CP 1 127 145  
## - Ca 1 145 163  
## - Thal 1 148 166  
## Start: AIC=177.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 150 176  
## - Exang 1 150 176  
## - Thal 1 150 176  
## - Oldpeak 1 150 176  
## - Age 1 151 177  
## - Fbs 1 151 177  
## <none> 150 178  
## - Slope 1 152 178  
## - Trestbps 1 153 179  
## - Chol 1 157 183  
## - Sex 1 158 184  
## - Thalach 1 158 184  
## - Ca 1 158 184  
## - CP 1 164 190  
##   
## Step: AIC=175.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 150 174  
## - Thal 1 150 174  
## - Oldpeak 1 150 174  
## - Age 1 151 175  
## - Fbs 1 151 175  
## <none> 150 176  
## - Slope 1 152 176  
## - Trestbps 1 153 177  
## - Chol 1 158 182  
## - Sex 1 158 182  
## - Thalach 1 158 182  
## - Ca 1 158 182  
## - CP 1 164 188  
##   
## Step: AIC=173.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thal 1 150 172  
## - Oldpeak 1 151 173  
## - Age 1 151 173  
## - Fbs 1 151 173  
## <none> 150 174  
## - Slope 1 153 175  
## - Trestbps 1 153 175  
## - Chol 1 158 180  
## - Sex 1 158 180  
## - Thalach 1 158 180  
## - Ca 1 159 181  
## - CP 1 165 187  
##   
## Step: AIC=172.5  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Oldpeak + Slope + Ca  
##   
## Df Deviance AIC  
## - Oldpeak 1 152 172  
## - Age 1 152 172  
## - Fbs 1 152 172  
## <none> 150 172  
## - Slope 1 154 174  
## - Trestbps 1 154 174  
## - Chol 1 158 178  
## - Ca 1 160 180  
## - Thalach 1 161 181  
## - Sex 1 164 184  
## - CP 1 171 191  
##   
## Step: AIC=171.5  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Slope + Ca  
##   
## Df Deviance AIC  
## - Age 1 153 171  
## - Fbs 1 153 171  
## <none> 152 172  
## - Trestbps 1 156 174  
## - Slope 1 159 177  
## - Chol 1 160 178  
## - Thalach 1 163 181  
## - Ca 1 165 183  
## - Sex 1 169 187  
## - CP 1 177 195  
##   
## Step: AIC=170.6  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Thalach + Slope +   
## Ca  
##   
## Df Deviance AIC  
## - Fbs 1 154 170  
## <none> 153 171  
## - Trestbps 1 156 172  
## - Chol 1 160 176  
## - Slope 1 161 177  
## - Thalach 1 163 179  
## - Ca 1 165 181  
## - Sex 1 172 188  
## - CP 1 178 194  
##   
## Step: AIC=169.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Slope + Ca  
##   
## Df Deviance AIC  
## <none> 154 170  
## - Trestbps 1 159 173  
## - Chol 1 162 176  
## - Slope 1 163 177  
## - Thalach 1 164 178  
## - Ca 1 168 182  
## - Sex 1 174 188  
## - CP 1 178 192  
## Start: AIC=129.9  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 102 128  
## - Oldpeak 1 102 128  
## - Fbs 1 103 129  
## <none> 102 130  
## - Slope 1 104 130  
## - CP 1 104 130  
## - Exang 1 105 131  
## - Sex 1 105 131  
## - Trestbps 1 107 133  
## - Age 1 107 133  
## - Chol 1 108 134  
## - Thalach 1 116 142  
## - Thal 1 120 146  
## - Ca 1 124 150  
##   
## Step: AIC=128  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 102 126  
## - Fbs 1 103 127  
## <none> 102 128  
## - Slope 1 104 128  
## - CP 1 104 128  
## - Exang 1 105 129  
## - Sex 1 105 129  
## - Trestbps 1 107 131  
## - Age 1 107 131  
## - Chol 1 108 132  
## - Thalach 1 116 140  
## - Thal 1 121 145  
## - Ca 1 124 148  
##   
## Step: AIC=126.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 103 125  
## <none> 102 126  
## - CP 1 105 127  
## - Slope 1 106 128  
## - Sex 1 107 129  
## - Exang 1 107 129  
## - Trestbps 1 108 130  
## - Age 1 108 130  
## - Chol 1 109 131  
## - Thalach 1 117 139  
## - Thal 1 121 143  
## - Ca 1 127 149  
##   
## Step: AIC=125.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 103 125  
## - Slope 1 107 127  
## - CP 1 107 127  
## - Exang 1 107 127  
## - Sex 1 108 128  
## - Trestbps 1 108 128  
## - Age 1 109 129  
## - Chol 1 110 130  
## - Thalach 1 118 138  
## - Thal 1 122 142  
## - Ca 1 128 148  
## Start: AIC=153.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 125 151  
## - Fbs 1 126 152  
## - Exang 1 126 152  
## - Sex 1 127 153  
## <none> 125 153  
## - Restecg 1 128 154  
## - Trestbps 1 128 154  
## - Age 1 128 154  
## - Oldpeak 1 129 155  
## - CP 1 133 159  
## - Chol 1 134 160  
## - Thal 1 136 162  
## - Thalach 1 136 162  
## - Ca 1 139 165  
##   
## Step: AIC=151.4  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 126 150  
## - Exang 1 126 150  
## - Sex 1 127 151  
## <none> 125 151  
## - Restecg 1 128 152  
## - Trestbps 1 128 152  
## - Age 1 128 152  
## - Oldpeak 1 130 154  
## - CP 1 133 157  
## - Chol 1 134 158  
## - Thal 1 136 160  
## - Thalach 1 138 162  
## - Ca 1 139 163  
##   
## Step: AIC=150  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Restecg + Thalach +   
## Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 127 149  
## - Sex 1 127 149  
## <none> 126 150  
## - Restecg 1 128 150  
## - Trestbps 1 128 150  
## - Age 1 129 151  
## - Oldpeak 1 131 153  
## - CP 1 133 155  
## - Chol 1 135 157  
## - Thal 1 136 158  
## - Thalach 1 138 160  
## - Ca 1 141 163  
##   
## Step: AIC=149  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Restecg + Thalach +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 128 148  
## - Restecg 1 129 149  
## <none> 127 149  
## - Trestbps 1 129 149  
## - Age 1 130 150  
## - Oldpeak 1 131 151  
## - CP 1 134 154  
## - Chol 1 136 156  
## - Thal 1 137 157  
## - Thalach 1 138 158  
## - Ca 1 141 161  
##   
## Step: AIC=148.4  
## .outcome ~ Age + CP + Trestbps + Chol + Restecg + Thalach + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 130 148  
## <none> 128 148  
## - Restecg 1 131 149  
## - Age 1 132 150  
## - CP 1 134 152  
## - Oldpeak 1 134 152  
## - Chol 1 136 154  
## - Thalach 1 140 158  
## - Thal 1 142 160  
## - Ca 1 145 163  
##   
## Step: AIC=148.3  
## .outcome ~ Age + CP + Chol + Restecg + Thalach + Oldpeak + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 130 148  
## - Restecg 1 133 149  
## - Age 1 133 149  
## - CP 1 136 152  
## - Oldpeak 1 137 153  
## - Chol 1 138 154  
## - Thalach 1 140 156  
## - Thal 1 145 161  
## - Ca 1 147 163  
## Start: AIC=153.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 125 151  
## - Fbs 1 126 152  
## - Oldpeak 1 126 152  
## - Exang 1 126 152  
## - Slope 1 126 152  
## - Age 1 127 153  
## <none> 125 153  
## - Thal 1 128 154  
## - Trestbps 1 129 155  
## - Ca 1 129 155  
## - Chol 1 130 156  
## - Thalach 1 130 156  
## - Sex 1 144 170  
## - CP 1 145 171  
##   
## Step: AIC=151.4  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 126 150  
## - Oldpeak 1 126 150  
## - Slope 1 126 150  
## - Exang 1 126 150  
## - Age 1 127 151  
## <none> 125 151  
## - Trestbps 1 129 153  
## - Ca 1 129 153  
## - Thal 1 129 153  
## - Chol 1 130 154  
## - Thalach 1 130 154  
## - Sex 1 144 168  
## - CP 1 146 170  
##   
## Step: AIC=149.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 126 148  
## - Slope 1 126 148  
## - Exang 1 126 148  
## - Age 1 127 149  
## <none> 126 150  
## - Trestbps 1 129 151  
## - Thal 1 129 151  
## - Ca 1 130 152  
## - Chol 1 130 152  
## - Thalach 1 130 152  
## - Sex 1 144 166  
## - CP 1 146 168  
##   
## Step: AIC=148.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 127 147  
## - Age 1 127 147  
## <none> 126 148  
## - Slope 1 129 149  
## - Trestbps 1 130 150  
## - Thal 1 130 150  
## - Thalach 1 131 151  
## - Chol 1 132 152  
## - Ca 1 132 152  
## - Sex 1 147 167  
## - CP 1 147 167  
##   
## Step: AIC=147.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 129 147  
## <none> 127 147  
## - Slope 1 131 149  
## - Trestbps 1 131 149  
## - Thal 1 132 150  
## - Chol 1 132 150  
## - Thalach 1 132 150  
## - Ca 1 133 151  
## - Sex 1 147 165  
## - CP 1 152 170  
##   
## Step: AIC=146.7  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 129 147  
## - Trestbps 1 132 148  
## - Slope 1 132 148  
## - Thalach 1 133 149  
## - Chol 1 133 149  
## - Ca 1 133 149  
## - Thal 1 134 150  
## - Sex 1 152 168  
## - CP 1 160 176  
## Start: AIC=141.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 114 140  
## - Restecg 1 114 140  
## - Slope 1 114 140  
## - Fbs 1 114 140  
## - CP 1 115 141  
## <none> 114 142  
## - Sex 1 117 143  
## - Trestbps 1 117 143  
## - Age 1 120 146  
## - Chol 1 121 147  
## - Thal 1 125 151  
## - Exang 1 125 151  
## - Thalach 1 129 155  
## - Ca 1 130 156  
##   
## Step: AIC=139.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 114 138  
## - Slope 1 114 138  
## - Fbs 1 114 138  
## - CP 1 115 139  
## <none> 114 140  
## - Sex 1 117 141  
## - Trestbps 1 117 141  
## - Age 1 120 144  
## - Chol 1 121 145  
## - Thal 1 125 149  
## - Exang 1 126 150  
## - Thalach 1 129 153  
## - Ca 1 130 154  
##   
## Step: AIC=137.9  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 114 136  
## - Fbs 1 114 136  
## - CP 1 115 137  
## <none> 114 138  
## - Sex 1 118 140  
## - Trestbps 1 118 140  
## - Age 1 120 142  
## - Chol 1 122 144  
## - Thal 1 126 148  
## - Exang 1 127 149  
## - Thalach 1 129 151  
## - Ca 1 131 153  
##   
## Step: AIC=136.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 115 135  
## - CP 1 115 135  
## <none> 114 136  
## - Sex 1 118 138  
## - Trestbps 1 118 138  
## - Age 1 120 140  
## - Chol 1 123 143  
## - Thal 1 127 147  
## - Exang 1 130 150  
## - Ca 1 131 151  
## - Thalach 1 134 154  
##   
## Step: AIC=134.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 115 135  
## - CP 1 117 135  
## - Sex 1 118 136  
## - Trestbps 1 118 136  
## - Age 1 121 139  
## - Chol 1 123 141  
## - Thal 1 130 148  
## - Exang 1 130 148  
## - Ca 1 131 149  
## - Thalach 1 135 153  
## Start: AIC=146.4  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 118 144  
## - Oldpeak 1 119 145  
## - Restecg 1 119 145  
## - Exang 1 120 146  
## - Slope 1 120 146  
## <none> 118 146  
## - Age 1 120 146  
## - Thal 1 122 148  
## - Trestbps 1 122 148  
## - CP 1 123 149  
## - Chol 1 128 154  
## - Sex 1 128 154  
## - Thalach 1 131 157  
## - Ca 1 136 162  
##   
## Step: AIC=144.4  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Restecg + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 119 143  
## - Restecg 1 119 143  
## - Exang 1 120 144  
## - Slope 1 120 144  
## <none> 118 144  
## - Age 1 121 145  
## - Thal 1 122 146  
## - Trestbps 1 122 146  
## - CP 1 123 147  
## - Chol 1 128 152  
## - Sex 1 128 152  
## - Thalach 1 131 155  
## - Ca 1 137 161  
##   
## Step: AIC=142.8  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Restecg + Thalach +   
## Exang + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 119 141  
## - Exang 1 120 142  
## <none> 119 143  
## - Age 1 121 143  
## - Thal 1 122 144  
## - Trestbps 1 123 145  
## - Slope 1 124 146  
## - CP 1 124 146  
## - Chol 1 128 150  
## - Sex 1 130 152  
## - Thalach 1 132 154  
## - Ca 1 139 161  
##   
## Step: AIC=141.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 121 141  
## <none> 119 141  
## - Age 1 122 142  
## - Thal 1 122 142  
## - CP 1 124 144  
## - Trestbps 1 124 144  
## - Slope 1 125 145  
## - Chol 1 129 149  
## - Sex 1 130 150  
## - Thalach 1 133 153  
## - Ca 1 140 160  
##   
## Step: AIC=141.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 121 141  
## - Age 1 124 142  
## - Thal 1 125 143  
## - Trestbps 1 126 144  
## - CP 1 127 145  
## - Slope 1 127 145  
## - Chol 1 131 149  
## - Sex 1 131 149  
## - Thalach 1 138 156  
## - Ca 1 142 160  
## Start: AIC=142  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 114 140  
## - Slope 1 114 140  
## - Fbs 1 115 141  
## - Sex 1 115 141  
## - Age 1 116 142  
## <none> 114 142  
## - Exang 1 117 143  
## - Chol 1 117 143  
## - Thalach 1 118 144  
## - Oldpeak 1 120 146  
## - Trestbps 1 124 150  
## - Ca 1 127 153  
## - CP 1 134 160  
## - Thal 1 135 161  
##   
## Step: AIC=140  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 114 138  
## - Fbs 1 115 139  
## - Sex 1 115 139  
## - Age 1 116 140  
## <none> 114 140  
## - Exang 1 117 141  
## - Chol 1 118 142  
## - Thalach 1 118 142  
## - Oldpeak 1 120 144  
## - Trestbps 1 125 149  
## - Ca 1 127 151  
## - CP 1 134 158  
## - Thal 1 136 160  
##   
## Step: AIC=138.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 115 137  
## - Sex 1 116 138  
## - Age 1 116 138  
## <none> 114 138  
## - Exang 1 117 139  
## - Thalach 1 118 140  
## - Chol 1 118 140  
## - Oldpeak 1 120 142  
## - Trestbps 1 125 147  
## - Ca 1 129 151  
## - CP 1 135 157  
## - Thal 1 137 159  
##   
## Step: AIC=137.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Sex 1 117 137  
## <none> 115 137  
## - Age 1 118 138  
## - Exang 1 118 138  
## - Chol 1 119 139  
## - Thalach 1 119 139  
## - Oldpeak 1 121 141  
## - Trestbps 1 126 146  
## - Ca 1 129 149  
## - Thal 1 137 157  
## - CP 1 138 158  
##   
## Step: AIC=136.8  
## .outcome ~ Age + CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## <none> 117 137  
## - Age 1 119 137  
## - Chol 1 119 137  
## - Exang 1 120 138  
## - Thalach 1 121 139  
## - Oldpeak 1 123 141  
## - Trestbps 1 127 145  
## - Ca 1 132 150  
## - CP 1 138 156  
## - Thal 1 149 167  
## Start: AIC=142.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Thalach 1 115 141  
## - Slope 1 115 141  
## - Restecg 1 115 141  
## - Age 1 115 141  
## - Chol 1 115 141  
## - Thal 1 116 142  
## - Trestbps 1 116 142  
## - Fbs 1 116 142  
## <none> 115 143  
## - Exang 1 119 145  
## - Sex 1 120 146  
## - CP 1 120 146  
## - Oldpeak 1 122 148  
## - Ca 1 136 162  
##   
## Step: AIC=140.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 115 139  
## - Slope 1 115 139  
## - Age 1 115 139  
## - Chol 1 115 139  
## - Thal 1 116 140  
## - Trestbps 1 116 140  
## - Fbs 1 116 140  
## <none> 115 141  
## - Exang 1 119 143  
## - Sex 1 120 144  
## - CP 1 121 145  
## - Oldpeak 1 122 146  
## - Ca 1 136 160  
##   
## Step: AIC=138.6  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Exang + Oldpeak +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 115 137  
## - Age 1 115 137  
## - Chol 1 115 137  
## - Thal 1 116 138  
## - Trestbps 1 116 138  
## - Fbs 1 116 138  
## <none> 115 139  
## - Exang 1 119 141  
## - Sex 1 120 142  
## - CP 1 121 143  
## - Oldpeak 1 122 144  
## - Ca 1 137 159  
##   
## Step: AIC=136.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 115 135  
## - Chol 1 115 135  
## - Trestbps 1 116 136  
## - Thal 1 116 136  
## - Fbs 1 116 136  
## <none> 115 137  
## - Exang 1 119 139  
## - Sex 1 120 140  
## - CP 1 121 141  
## - Oldpeak 1 126 146  
## - Ca 1 137 157  
##   
## Step: AIC=134.9  
## .outcome ~ Sex + CP + Trestbps + Chol + Fbs + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 115 133  
## - Trestbps 1 116 134  
## - Thal 1 116 134  
## <none> 115 135  
## - Fbs 1 117 135  
## - Exang 1 120 138  
## - CP 1 121 139  
## - Sex 1 122 140  
## - Oldpeak 1 126 144  
## - Ca 1 138 156  
##   
## Step: AIC=133.4  
## .outcome ~ Sex + CP + Trestbps + Fbs + Exang + Oldpeak + Ca +   
## Thal  
##   
## Df Deviance AIC  
## - Trestbps 1 117 133  
## - Thal 1 117 133  
## - Fbs 1 117 133  
## <none> 115 133  
## - Exang 1 120 136  
## - CP 1 121 137  
## - Sex 1 122 138  
## - Oldpeak 1 127 143  
## - Ca 1 140 156  
##   
## Step: AIC=132.7  
## .outcome ~ Sex + CP + Fbs + Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Thal 1 118 132  
## - Fbs 1 118 132  
## <none> 117 133  
## - CP 1 122 136  
## - Exang 1 122 136  
## - Sex 1 123 137  
## - Oldpeak 1 128 142  
## - Ca 1 146 160  
##   
## Step: AIC=132.2  
## .outcome ~ Sex + CP + Fbs + Exang + Oldpeak + Ca  
##   
## Df Deviance AIC  
## - Fbs 1 120 132  
## <none> 118 132  
## - Exang 1 125 137  
## - CP 1 125 137  
## - Sex 1 131 143  
## - Oldpeak 1 136 148  
## - Ca 1 156 168  
##   
## Step: AIC=132.1  
## .outcome ~ Sex + CP + Exang + Oldpeak + Ca  
##   
## Df Deviance AIC  
## <none> 120 132  
## - Exang 1 126 136  
## - CP 1 132 142  
## - Sex 1 133 143  
## - Oldpeak 1 138 148  
## - Ca 1 156 166  
## Start: AIC=144.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 117 143  
## - Restecg 1 117 143  
## - Oldpeak 1 118 144  
## - Chol 1 118 144  
## - Fbs 1 118 144  
## - CP 1 118 144  
## - Age 1 118 144  
## <none> 117 145  
## - Sex 1 120 146  
## - Trestbps 1 121 147  
## - Exang 1 125 151  
## - Ca 1 127 153  
## - Thal 1 129 155  
## - Thalach 1 129 155  
##   
## Step: AIC=142.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 117 141  
## - Chol 1 118 142  
## - Fbs 1 118 142  
## - CP 1 118 142  
## - Age 1 118 142  
## - Oldpeak 1 118 142  
## <none> 117 143  
## - Sex 1 121 145  
## - Trestbps 1 121 145  
## - Exang 1 126 150  
## - Ca 1 127 151  
## - Thal 1 129 153  
## - Thalach 1 129 153  
##   
## Step: AIC=141.3  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 118 140  
## - CP 1 118 140  
## - Chol 1 118 140  
## - Oldpeak 1 119 141  
## - Age 1 119 141  
## <none> 117 141  
## - Sex 1 122 144  
## - Trestbps 1 123 145  
## - Exang 1 126 148  
## - Ca 1 128 150  
## - Thal 1 129 151  
## - Thalach 1 131 153  
##   
## Step: AIC=140.2  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 119 139  
## - CP 1 120 140  
## - Oldpeak 1 120 140  
## <none> 118 140  
## - Age 1 120 140  
## - Sex 1 122 142  
## - Trestbps 1 124 144  
## - Exang 1 126 146  
## - Ca 1 128 148  
## - Thal 1 130 150  
## - Thalach 1 132 152  
##   
## Step: AIC=139.1  
## .outcome ~ Age + Sex + CP + Trestbps + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - CP 1 120 138  
## - Oldpeak 1 121 139  
## - Age 1 121 139  
## <none> 119 139  
## - Sex 1 123 141  
## - Trestbps 1 126 144  
## - Exang 1 127 145  
## - Ca 1 130 148  
## - Thal 1 131 149  
## - Thalach 1 133 151  
##   
## Step: AIC=138.4  
## .outcome ~ Age + Sex + Trestbps + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Oldpeak 1 122 138  
## <none> 120 138  
## - Age 1 122 138  
## - Sex 1 123 139  
## - Trestbps 1 126 142  
## - Exang 1 130 146  
## - Ca 1 133 149  
## - Thal 1 134 150  
## - Thalach 1 137 153  
##   
## Step: AIC=138  
## .outcome ~ Age + Sex + Trestbps + Thalach + Exang + Ca + Thal  
##   
## Df Deviance AIC  
## <none> 122 138  
## - Age 1 124 138  
## - Sex 1 125 139  
## - Trestbps 1 128 142  
## - Exang 1 134 148  
## - Thal 1 138 152  
## - Ca 1 139 153  
## - Thalach 1 140 154  
## Start: AIC=162.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Restecg +   
## Thalach + Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Restecg 1 134 160  
## - Fbs 1 135 161  
## - Age 1 135 161  
## - Oldpeak 1 135 161  
## - Slope 1 135 161  
## - Exang 1 136 162  
## <none> 134 162  
## - Trestbps 1 137 163  
## - Sex 1 138 164  
## - Thalach 1 138 164  
## - Chol 1 139 165  
## - CP 1 142 168  
## - Thal 1 147 173  
## - Ca 1 147 173  
##   
## Step: AIC=160.1  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Fbs + Thalach +   
## Exang + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 135 159  
## - Age 1 135 159  
## - Oldpeak 1 135 159  
## - Slope 1 135 159  
## - Exang 1 136 160  
## <none> 134 160  
## - Trestbps 1 137 161  
## - Sex 1 138 162  
## - Thalach 1 138 162  
## - Chol 1 139 163  
## - CP 1 142 166  
## - Thal 1 147 171  
## - Ca 1 147 171  
##   
## Step: AIC=158.7  
## .outcome ~ Age + Sex + CP + Trestbps + Chol + Thalach + Exang +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 135 157  
## - Oldpeak 1 136 158  
## - Slope 1 136 158  
## - Exang 1 136 158  
## <none> 135 159  
## - Trestbps 1 138 160  
## - Sex 1 138 160  
## - Thalach 1 139 161  
## - Chol 1 140 162  
## - CP 1 143 165  
## - Ca 1 147 169  
## - Thal 1 148 170  
##   
## Step: AIC=157.4  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Slope 1 136 156  
## - Oldpeak 1 136 156  
## - Exang 1 137 157  
## <none> 135 157  
## - Trestbps 1 138 158  
## - Thalach 1 139 159  
## - Sex 1 139 159  
## - Chol 1 140 160  
## - CP 1 144 164  
## - Ca 1 147 167  
## - Thal 1 148 168  
##   
## Step: AIC=156.3  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Exang + Oldpeak +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - Exang 1 138 156  
## <none> 136 156  
## - Trestbps 1 138 156  
## - Oldpeak 1 140 158  
## - Sex 1 140 158  
## - Chol 1 140 158  
## - Thalach 1 141 159  
## - CP 1 145 163  
## - Ca 1 148 166  
## - Thal 1 151 169  
##   
## Step: AIC=156.3  
## .outcome ~ Sex + CP + Trestbps + Chol + Thalach + Oldpeak + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 138 156  
## - Trestbps 1 140 156  
## - Sex 1 142 158  
## - Chol 1 142 158  
## - Oldpeak 1 143 159  
## - Thalach 1 144 160  
## - CP 1 149 165  
## - Ca 1 149 165  
## - Thal 1 154 170  
## [1] "Linear Model with Stepwise Feature Selection (glmStepAIC) Executed ... time:5.3"

Let’s see the results:

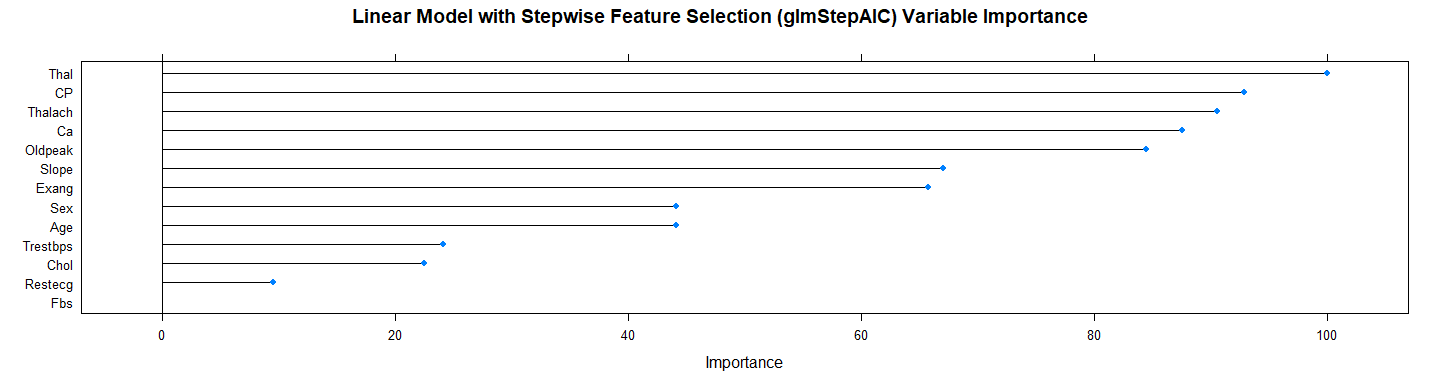
glmStepAIC

## Generalized Linear Model with Stepwise Feature Selection   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 198, 198, 198, 198, 198, 198, ...   
## Resampling results:  
##   
## Accuracy Kappa   
## 0.8076 0.6117

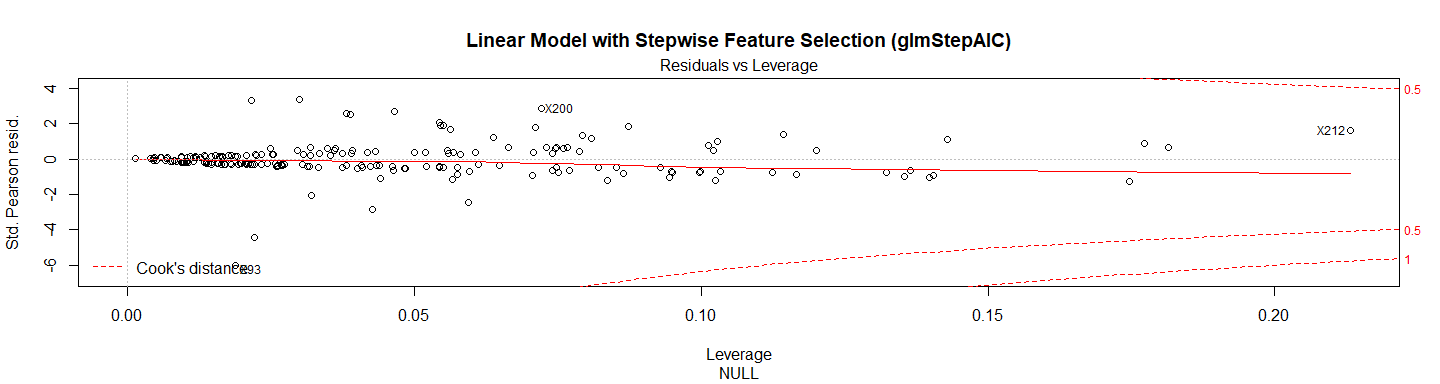
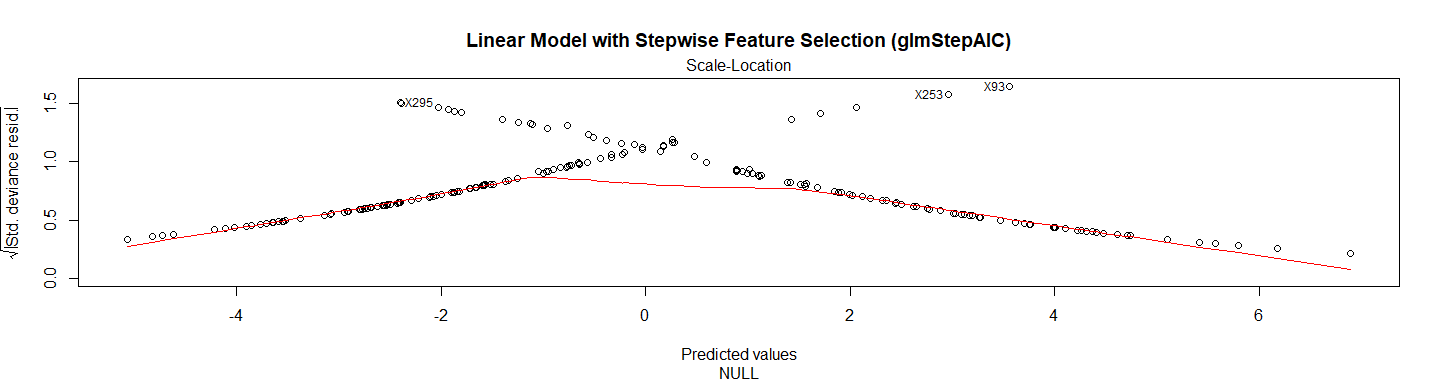
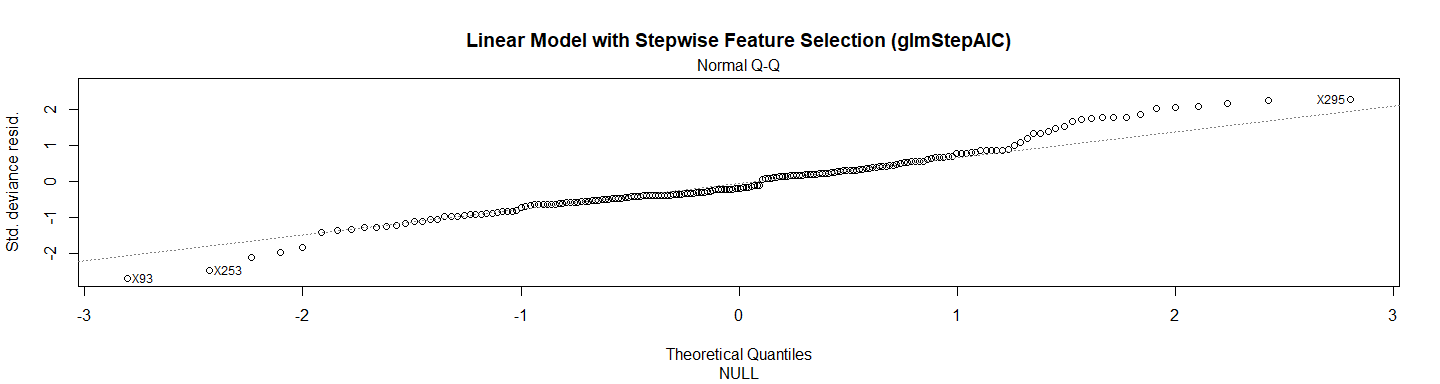
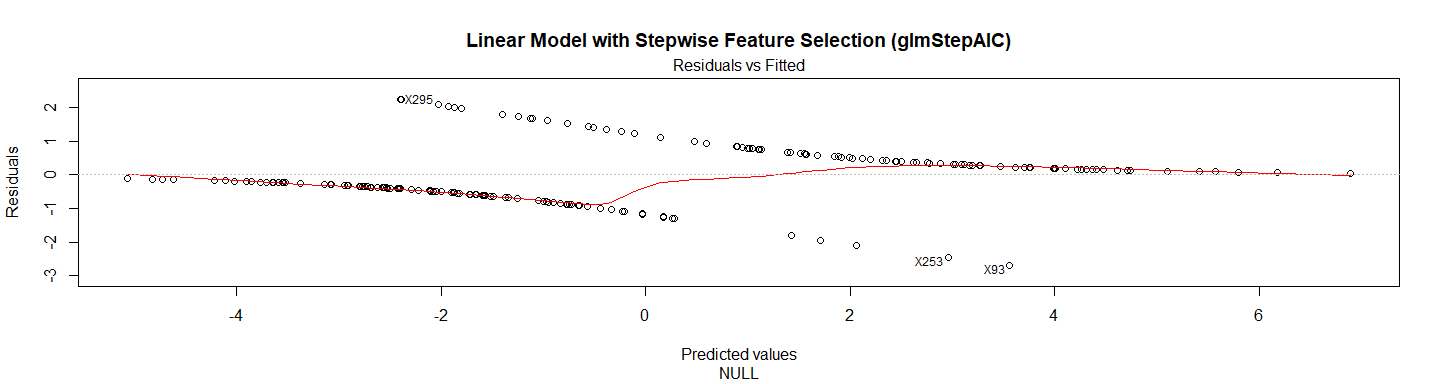
**summary**(glmStepAIC)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.680 -0.535 -0.202 0.407 2.229   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.0729 0.2284 -0.32 0.74960   
## Sex 0.4871 0.2654 1.84 0.06648 .   
## CP 0.7260 0.2328 3.12 0.00181 \*\*   
## Trestbps 0.3264 0.2211 1.48 0.13977   
## Chol 0.4631 0.2372 1.95 0.05089 .   
## Thalach -0.5805 0.2555 -2.27 0.02311 \*   
## Oldpeak 0.6107 0.2803 2.18 0.02936 \*   
## Ca 0.8384 0.2663 3.15 0.00164 \*\*   
## Thal 0.9094 0.2346 3.88 0.00011 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 273.19 on 197 degrees of freedom  
## Residual deviance: 138.29 on 189 degrees of freedom  
## AIC: 156.3  
##   
## Number of Fisher Scoring iterations: 6

**plot**(**varImp**(glmStepAIC),   
 main = **paste0**(model.name," Variable Importance"))



**plot**(glmStepAIC**$**finalModel,   
 main = model.name)



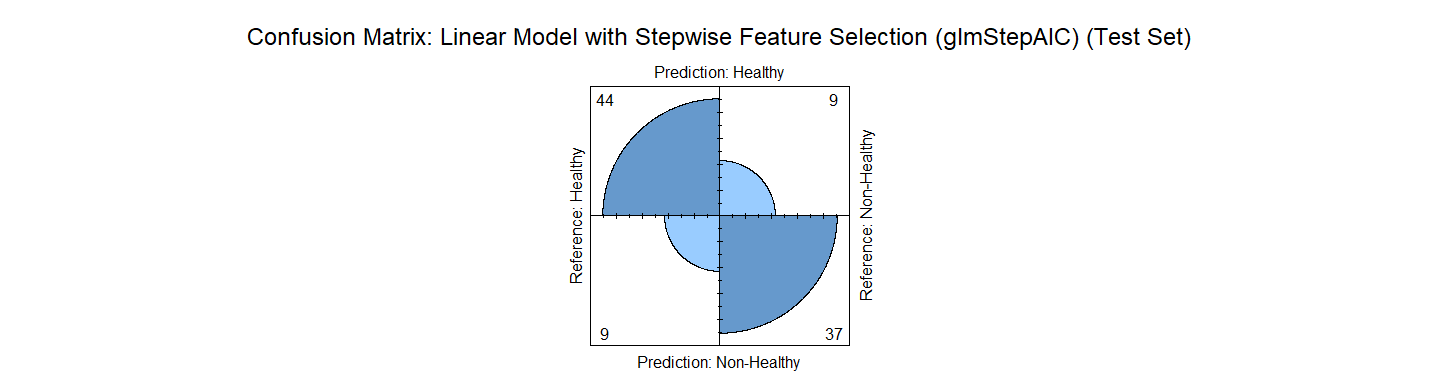
test\_pred\_glmStepAIC <- **predict**(glmStepAIC, newdata = X.testData)  
test\_pred\_glmStepAIC

## [1] 1 0 0 0 1 1 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 0 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 0 1 1 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 1 0 0 0  
## Levels: 0 1

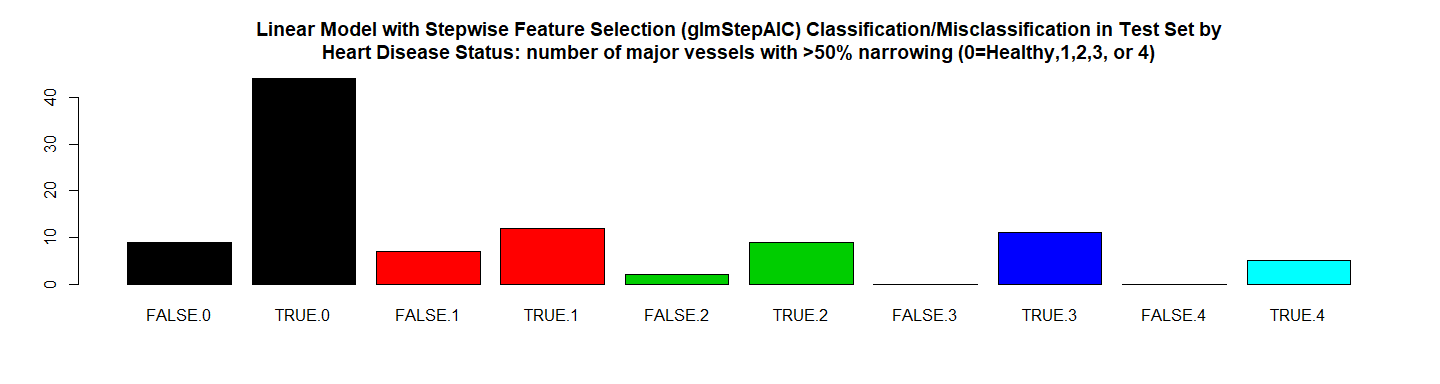
res.glmStepAIC <- **confusionMatrix**(test\_pred\_glmStepAIC, Y.testData)   
res.glmStepAIC

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 44 9  
## 1 9 37  
##   
## Accuracy : 0.818   
## 95% CI : (0.728, 0.889)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 3.57e-09   
##   
## Kappa : 0.635   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.830   
## Specificity : 0.804   
## Pos Pred Value : 0.830   
## Neg Pred Value : 0.804   
## Prevalence : 0.535   
## Detection Rate : 0.444   
## Detection Prevalence : 0.535   
## Balanced Accuracy : 0.817   
##   
## 'Positive' Class : 0   
##

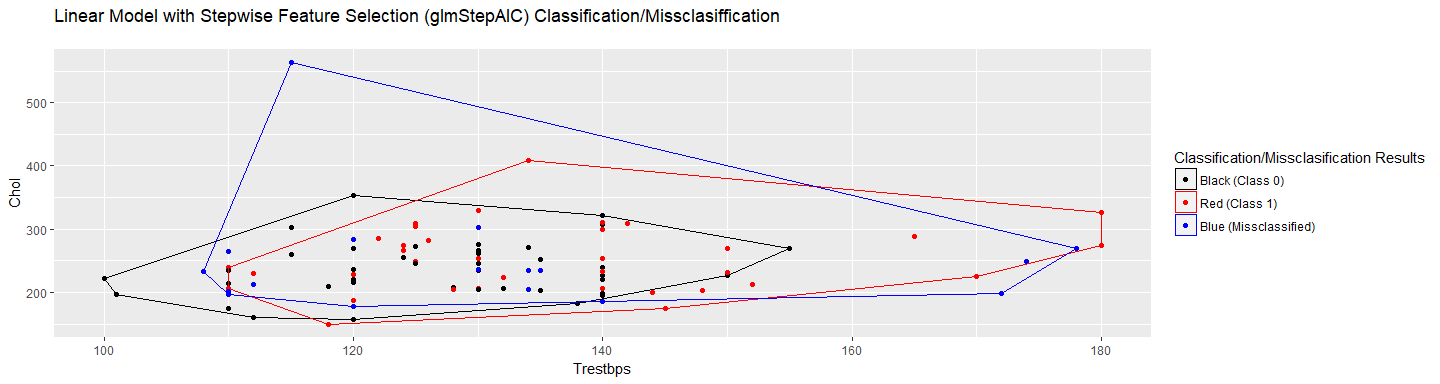
**print\_confusionm**(res.glmStepAIC,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



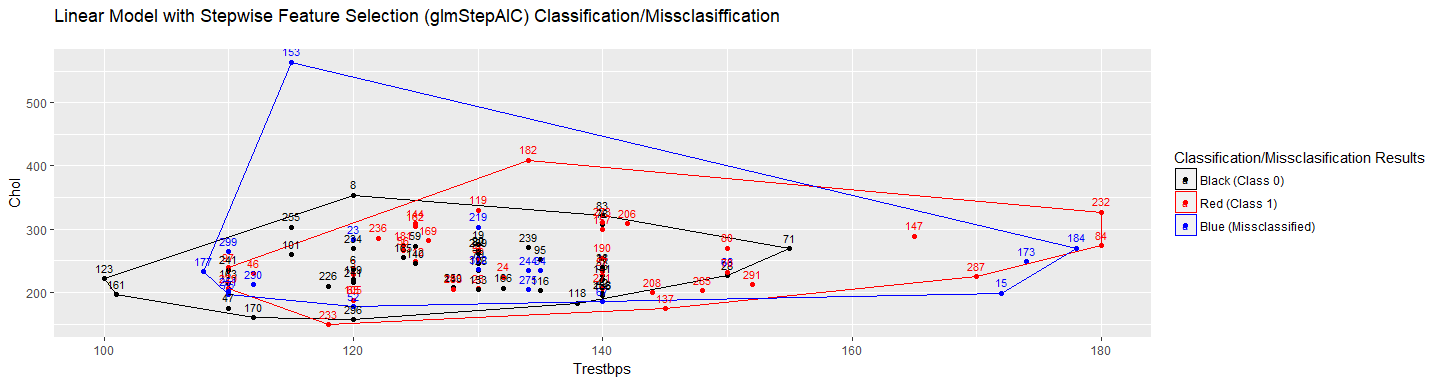
main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% ",  
 "narrowing (0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_glmStepAIC,  
 Y.testData,  
 main, testData)



**plot\_results**(X.testData, Y.testData,test\_pred\_glmStepAIC ,   
 **paste0**(model.name, " Classification/Missclasiffication\n"),   
 "Trestbps","Chol",labels = FALSE)



**plot\_results**(X.testData, Y.testData, test\_pred\_glmStepAIC,   
 **paste0**(model.name," Classification/Missclasiffication\n"),  
 "Trestbps","Chol",labels = TRUE)



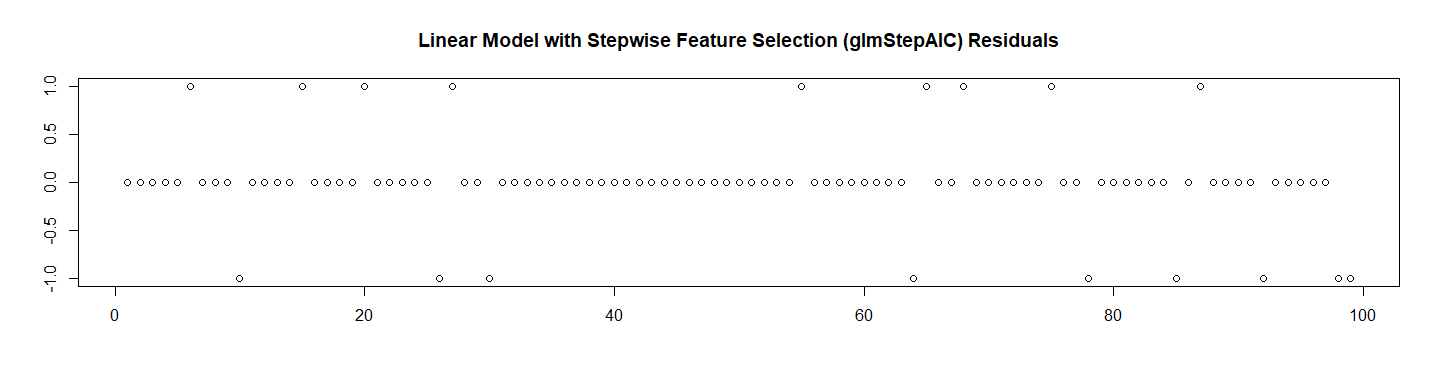
*# RMSE Resulting*  
(rmse\_glmStepAIC <- **sqrt**(**mean**((**as.numeric**(test\_pred\_glmStepAIC)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4264

*# Residuals Plot*  
(r\_glmStepAIC <- (**as.numeric**(test\_pred\_glmStepAIC)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 1 0 0 0 -1 0 0 0 0 1 0 0 0 0 1 0 0 0  
## [24] 0 0 -1 1 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 -1 0 0 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 0 0 -1 -1

**plot**(r\_glmStepAIC,   
 main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_glmStepAIC\_prob <- **predict**(glmStepAIC, newdata = X.testData, type ="prob")[2]  
  
res.glmStepAIC.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**((test\_pred\_glmStepAIC\_prob))))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 glmStepAIC**$**results**$**Accuracy,  
 "Prediction Accuracy in Test Set" =   
 res.glmStepAIC**$**overall[1],  
 "RMSE Test"= rmse\_glmStepAIC,  
 "ROC" = res.glmStepAIC.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 8. Kmeans (eclust)

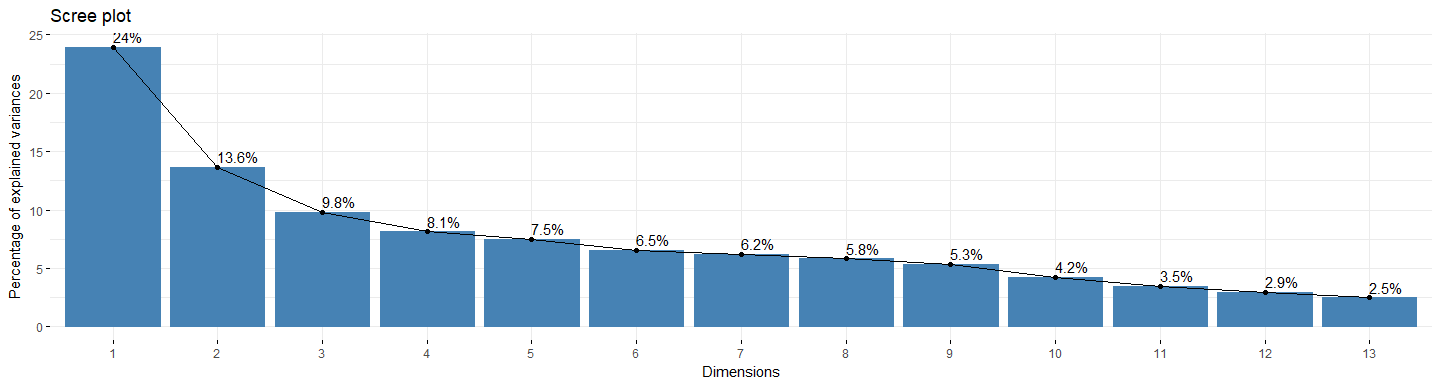
## 8.1 PCA

Because the dataset don’t have many variables (only 13) let’s validate if make sense to do PCA for Kmeans:

############################################  
*# 8. Kmeans (eclust)*  
############################################  
## 8.1 PCA  
pc.train <- **prcomp**(**as.matrix**(X.trainData.std), scale = TRUE, center = TRUE)  
  
*# Eigenvalues*   
eig.val <- **get\_eig**(pc.train)  
  
*# Let's see the PC*  
**kable**(eig.val)

|  |  |  |  |
| --- | --- | --- | --- |
|  | eigenvalue | variance.percent | cumulative.variance.percent |
| Dim.1 | 3.1153 | 23.964 | 23.96 |
| Dim.2 | 1.7717 | 13.629 | 37.59 |
| Dim.3 | 1.2738 | 9.798 | 47.39 |
| Dim.4 | 1.0564 | 8.126 | 55.52 |
| Dim.5 | 0.9726 | 7.482 | 63.00 |
| Dim.6 | 0.8483 | 6.525 | 69.52 |
| Dim.7 | 0.8070 | 6.208 | 75.73 |
| Dim.8 | 0.7554 | 5.810 | 81.54 |
| Dim.9 | 0.6910 | 5.316 | 86.86 |
| Dim.10 | 0.5504 | 4.234 | 91.09 |
| Dim.11 | 0.4538 | 3.491 | 94.58 |
| Dim.12 | 0.3814 | 2.933 | 97.52 |
| Dim.13 | 0.3229 | 2.484 | 100.00 |

**fviz\_screeplot**(pc.train, addlabels = TRUE, ncp=13)



*# Extract the results for variables*  
var <- **get\_pca\_var**(pc.train)  
  
**print**(var)

## Principal Component Analysis Results for variables  
## ===================================================  
## Name Description   
## 1 "$coord" "Coordinates for the variables"   
## 2 "$cor" "Correlations between variables and dimensions"  
## 3 "$cos2" "Cos2 for the variables"   
## 4 "$contrib" "contributions of the variables"

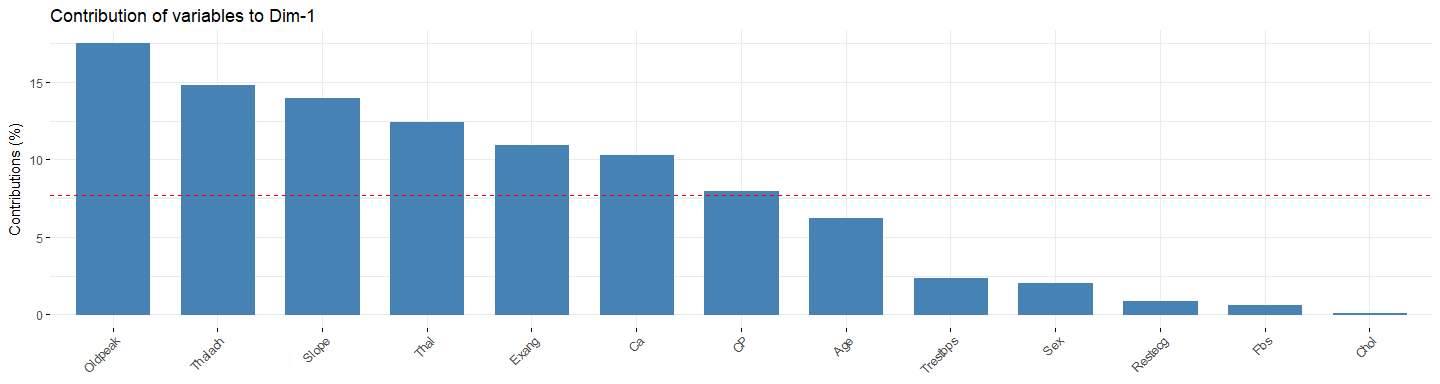
*# Let see the model generated (first five dimension and 10 variables)*  
**kable**(var**$**coord)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 | Dim.6 | Dim.7 | Dim.8 | Dim.9 | Dim.10 | Dim.11 | Dim.12 | Dim.13 |
| Age | -0.4409 | 0.6325 | -0.1050 | 0.2379 | -0.1118 | 0.2376 | -0.1447 | 0.2133 | -0.2020 | -0.0254 | -0.0479 | -0.3927 | -0.0682 |
| Sex | -0.2522 | -0.4320 | 0.5609 | 0.3546 | 0.3395 | 0.1583 | 0.0215 | -0.0185 | -0.1595 | 0.0181 | 0.3597 | -0.0924 | 0.0678 |
| CP | -0.4974 | -0.1913 | -0.5640 | 0.0947 | 0.0137 | 0.0506 | 0.2237 | -0.0384 | 0.4171 | -0.3093 | 0.2300 | -0.1046 | 0.0347 |
| Trestbps | -0.2701 | 0.5492 | 0.3464 | -0.0745 | 0.0750 | -0.2852 | -0.0143 | 0.5297 | 0.3006 | 0.0020 | 0.1435 | 0.1328 | 0.0602 |
| Chol | -0.0430 | 0.5534 | -0.2786 | 0.0726 | 0.4079 | -0.4655 | 0.1009 | -0.2812 | -0.3321 | -0.0921 | 0.1189 | 0.0473 | 0.0395 |
| Fbs | -0.1366 | 0.3705 | 0.4597 | 0.0161 | -0.5246 | 0.0087 | 0.4990 | -0.2995 | -0.0080 | -0.1022 | 0.0641 | 0.0183 | -0.0597 |
| Restecg | -0.1608 | 0.3461 | 0.1831 | -0.4790 | 0.5438 | 0.4658 | 0.1455 | -0.1470 | 0.1277 | -0.0713 | -0.1157 | 0.0087 | -0.0334 |
| Thalach | 0.6798 | 0.0305 | 0.2720 | -0.0230 | 0.1160 | -0.3255 | -0.0881 | -0.2364 | 0.3678 | 0.0579 | -0.0590 | -0.3689 | 0.0357 |
| Exang | -0.5834 | -0.2921 | -0.1220 | -0.1168 | 0.1247 | -0.1928 | 0.5047 | 0.1541 | -0.0464 | 0.4208 | -0.1254 | -0.1400 | -0.0084 |
| Oldpeak | -0.7385 | -0.1044 | 0.0689 | -0.2714 | -0.0513 | -0.1795 | -0.3361 | -0.1972 | 0.0629 | 0.0974 | 0.1534 | 0.0010 | -0.3762 |
| Slope | -0.6591 | -0.1411 | 0.1019 | -0.5060 | -0.2162 | -0.0994 | -0.2119 | -0.0973 | -0.1453 | -0.0921 | 0.0058 | -0.0988 | 0.3591 |
| Ca | -0.5668 | 0.3080 | -0.0488 | 0.4279 | -0.0095 | 0.1319 | -0.1857 | -0.3301 | 0.2666 | 0.3323 | -0.0843 | 0.1447 | 0.1672 |
| Thal | -0.6222 | -0.2675 | 0.2850 | 0.3123 | 0.1528 | -0.2159 | -0.0099 | 0.0154 | 0.0172 | -0.3478 | -0.4088 | 0.0319 | -0.0532 |

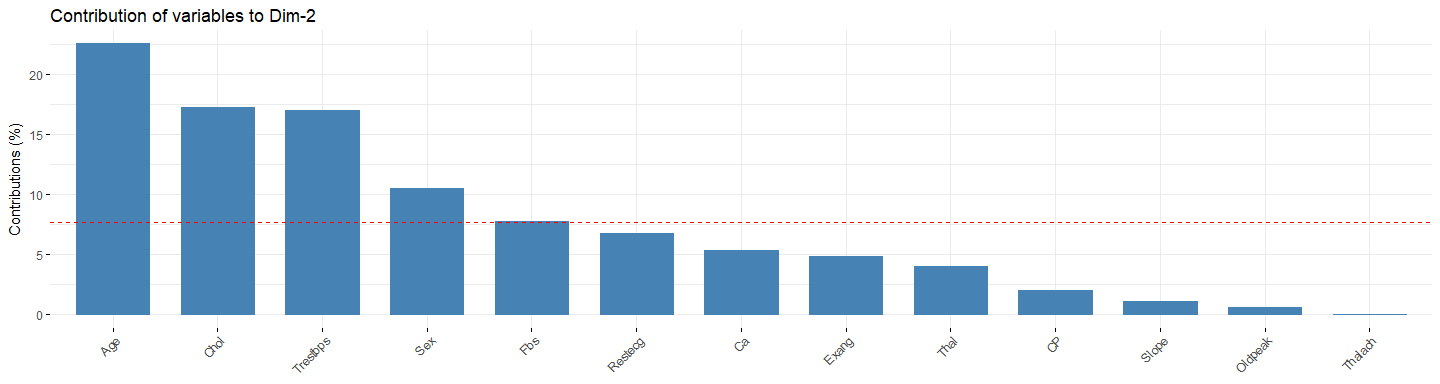
*# Let see the contribution of each variable to each PC*  
**kable**(var**$**contrib)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 | Dim.6 | Dim.7 | Dim.8 | Dim.9 | Dim.10 | Dim.11 | Dim.12 | Dim.13 |
| Age | 6.2389 | 22.5788 | 0.8662 | 5.3594 | 1.2844 | 6.6542 | 2.5938 | 6.0213 | 5.9060 | 0.1172 | 0.5056 | 40.4320 | 1.4421 |
| Sex | 2.0417 | 10.5325 | 24.7012 | 11.9018 | 11.8519 | 2.9527 | 0.0570 | 0.0451 | 3.6800 | 0.0598 | 28.5105 | 2.2407 | 1.4250 |
| CP | 7.9422 | 2.0658 | 24.9694 | 0.8497 | 0.0193 | 0.3024 | 6.2006 | 0.1949 | 25.1741 | 17.3819 | 11.6590 | 2.8671 | 0.3737 |
| Trestbps | 2.3413 | 17.0249 | 9.4184 | 0.5258 | 0.5784 | 9.5861 | 0.0254 | 37.1428 | 13.0740 | 0.0008 | 4.5351 | 4.6231 | 1.1240 |
| Chol | 0.0593 | 17.2864 | 6.0925 | 0.4983 | 17.1058 | 25.5424 | 1.2624 | 10.4658 | 15.9627 | 1.5403 | 3.1149 | 0.5864 | 0.4827 |
| Fbs | 0.5988 | 7.7467 | 16.5937 | 0.0244 | 28.2902 | 0.0089 | 30.8579 | 11.8731 | 0.0092 | 1.8966 | 0.9067 | 0.0883 | 1.1054 |
| Restecg | 0.8301 | 6.7619 | 2.6325 | 21.7173 | 30.4014 | 25.5740 | 2.6246 | 2.8600 | 2.3581 | 0.9225 | 2.9520 | 0.0198 | 0.3457 |
| Thalach | 14.8335 | 0.0523 | 5.8074 | 0.0499 | 1.3834 | 12.4875 | 0.9625 | 7.3959 | 19.5709 | 0.6083 | 0.7675 | 35.6856 | 0.3952 |
| Exang | 10.9238 | 4.8161 | 1.1694 | 1.2921 | 1.5986 | 4.3822 | 31.5694 | 3.1424 | 0.3115 | 32.1672 | 3.4677 | 5.1374 | 0.0221 |
| Oldpeak | 17.5084 | 0.6152 | 0.3727 | 6.9730 | 0.2711 | 3.7983 | 14.0005 | 5.1493 | 0.5729 | 1.7252 | 5.1884 | 0.0003 | 43.8249 |
| Slope | 13.9426 | 1.1240 | 0.8146 | 24.2395 | 4.8048 | 1.1653 | 5.5613 | 1.2522 | 3.0541 | 1.5427 | 0.0075 | 2.5606 | 39.9307 |
| Ca | 10.3116 | 5.3551 | 0.1872 | 17.3336 | 0.0092 | 2.0505 | 4.2723 | 14.4257 | 10.2834 | 20.0610 | 1.5665 | 5.4912 | 8.6526 |
| Thal | 12.4278 | 4.0402 | 6.3747 | 9.2350 | 2.4016 | 5.4954 | 0.0122 | 0.0315 | 0.0430 | 21.9764 | 36.8186 | 0.2676 | 0.8760 |

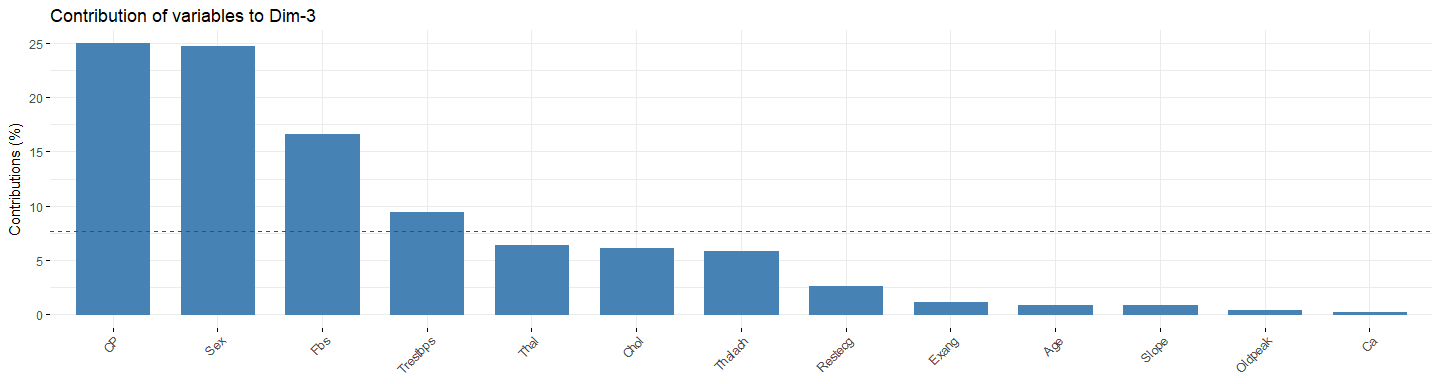
*# Contributions of variables to PC1*  
**fviz\_contrib**(pc.train, choice = "var", axes = 1, top = 13)



*# Contributions of variables to PC2*  
**fviz\_contrib**(pc.train, choice = "var", axes = 2, top = 13)



*# Contributions of variables to PC3*  
**fviz\_contrib**(pc.train, choice = "var", axes = 3, top = 13)



*# Extract the results for individuals*  
ind <- **get\_pca\_ind**(pc.train)  
  
*# Let's use the first 8 variables for the model*  
xx.pc.train <- **as.matrix**(ind**$**coord[,1**:**8])

Now let’s do clustering with K-means

min\_k <- 2  
max\_k <- 4  
min\_seed <- 1  
max\_seed <- 100  
method <- "euclidean"  
type\_data <- " PCA"  
  
*# Find Best Seed and K Using the PCA Data on training*   
file.name <- **paste**(work.dir, "best\_seeds\_std\_pca.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "best\_seeds\_std\_pca-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
  
 best\_seeds\_std\_pca <- **find\_best\_seedv2**(xx.pc.train,  
 method,  
 min\_seed,  
 max\_seed,  
 min\_k,  
 max\_k,  
 Y.trainData)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**("Best Seed for Kmeans PCA Finded ... time:",et))  
 **save**(best\_seeds\_std\_pca, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Results for k=2 PCA"  
## [1] "======================="  
## [1] "Best Seed find: k=2 seed=1 errors=91 sum(whithinss)=1625.42940877136"  
## [1] ""  
## [1] "Results for k=3 PCA"  
## [1] "======================="  
## [1] "Best Seed find: k=3 seed=1 errors=140 sum(whithinss)=1455.45826953285"  
## [1] "Best Seed find: k=3 seed=3 errors=138 sum(whithinss)=1457.43620803959"  
## [1] "Best Seed find: k=3 seed=5 errors=128 sum(whithinss)=1461.48346218718"  
## [1] ""  
## [1] "Results for k=4 PCA"  
## [1] "======================="  
## [1] "Best Seed find: k=4 seed=1 errors=147 sum(whithinss)=1312.9987279498"  
## [1] "Best Seed find: k=4 seed=26 errors=132 sum(whithinss)=1338.25742471506"  
## [1] ""  
## [1] ""  
## [1] "Final Results"  
## [1] "=============="  
## [1] "Best Seed find: k=2 seed=1 errors=91 sum(whithinss)=1625.42940877136"  
## [1] "Best Seed for Kmeans PCA Finded ... time:5.8"

(k <- best\_seeds\_std\_pca**$**best\_k)

## [1] 2

(best\_seed <- best\_seeds\_std\_pca**$**best\_seed)

## [1] 1

res.train.kmeans\_pca <- **confusionMatrix**(best\_seeds\_std\_pca**$**best\_km**$**cluster,   
 Y.trainData)  
res.train.kmeans\_pca

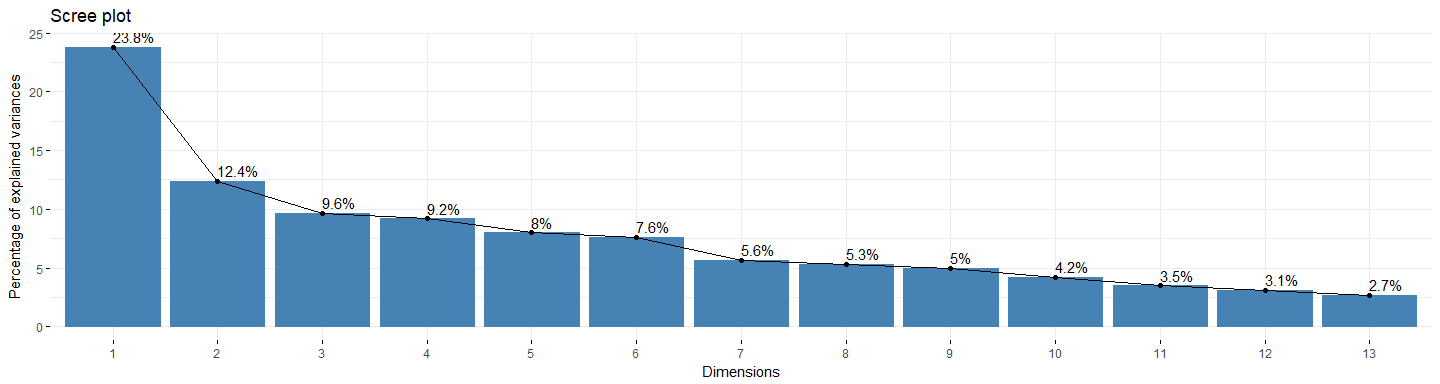
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 99 25  
## 1 8 66  
##   
## Accuracy : 0.833   
## 95% CI : (0.774, 0.882)  
## No Information Rate : 0.54   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.66   
## Mcnemar's Test P-Value : 0.00535   
##   
## Sensitivity : 0.925   
## Specificity : 0.725   
## Pos Pred Value : 0.798   
## Neg Pred Value : 0.892   
## Prevalence : 0.540   
## Detection Rate : 0.500   
## Detection Prevalence : 0.626   
## Balanced Accuracy : 0.825   
##   
## 'Positive' Class : 0   
##

Let’s do PCA on the test set and create the cluster with the k and best\_seed:

pc.test <- **prcomp**(**as.matrix**(X.testData.std), scale = TRUE, center = TRUE)  
  
*# Eigenvalues*   
  
eig.val <- **get\_eig**(pc.test)  
  
*# Let's see the PC*  
**kable**(eig.val)

|  |  |  |  |
| --- | --- | --- | --- |
|  | eigenvalue | variance.percent | cumulative.variance.percent |
| Dim.1 | 3.0958 | 23.814 | 23.81 |
| Dim.2 | 1.6122 | 12.402 | 36.22 |
| Dim.3 | 1.2515 | 9.627 | 45.84 |
| Dim.4 | 1.1936 | 9.182 | 55.02 |
| Dim.5 | 1.0383 | 7.987 | 63.01 |
| Dim.6 | 0.9871 | 7.593 | 70.60 |
| Dim.7 | 0.7340 | 5.646 | 76.25 |
| Dim.8 | 0.6911 | 5.316 | 81.57 |
| Dim.9 | 0.6449 | 4.961 | 86.53 |
| Dim.10 | 0.5447 | 4.190 | 90.72 |
| Dim.11 | 0.4523 | 3.479 | 94.20 |
| Dim.12 | 0.4070 | 3.130 | 97.33 |
| Dim.13 | 0.3475 | 2.673 | 100.00 |

**fviz\_screeplot**(pc.test, addlabels = TRUE, ncp=13)



*# Extract the results for variables*  
var <- **get\_pca\_var**(pc.test)  
  
**print**(var)

## Principal Component Analysis Results for variables  
## ===================================================  
## Name Description   
## 1 "$coord" "Coordinates for the variables"   
## 2 "$cor" "Correlations between variables and dimensions"  
## 3 "$cos2" "Cos2 for the variables"   
## 4 "$contrib" "contributions of the variables"

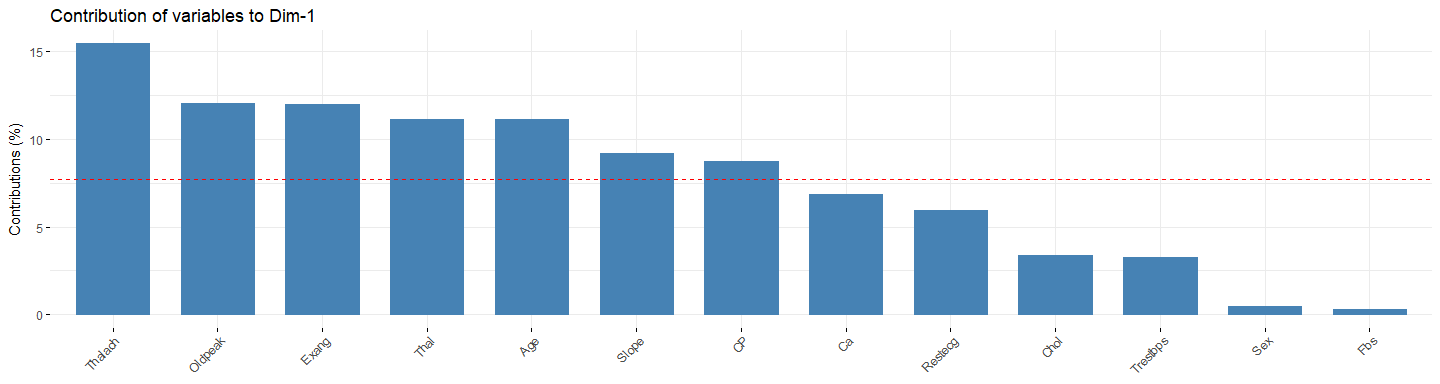
*# Let see the model generated (first five dimension and 10 variables)*  
**kable**(var**$**coord)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 | Dim.6 | Dim.7 | Dim.8 | Dim.9 | Dim.10 | Dim.11 | Dim.12 | Dim.13 |
| Age | -0.5869 | 0.2269 | -0.0365 | 0.1200 | -0.4239 | 0.1927 | -0.0062 | -0.2683 | 0.4604 | 0.1575 | 0.1856 | -0.0848 | -0.1453 |
| Sex | -0.1198 | -0.6788 | 0.4605 | 0.0055 | 0.1767 | -0.2421 | -0.0006 | -0.1202 | 0.2742 | -0.1062 | 0.1887 | 0.2883 | 0.0575 |
| CP | -0.5210 | -0.4129 | -0.3372 | -0.2575 | 0.1561 | 0.1755 | 0.1244 | 0.1000 | -0.2765 | 0.3159 | 0.3474 | 0.0043 | -0.0217 |
| Trestbps | -0.3171 | 0.3480 | 0.1073 | 0.5072 | 0.3614 | 0.4352 | 0.1328 | -0.2685 | -0.1584 | 0.0555 | -0.0523 | 0.2553 | 0.0618 |
| Chol | -0.3243 | 0.1973 | -0.6482 | 0.1198 | 0.2716 | -0.1660 | -0.4542 | 0.1507 | 0.2241 | -0.0617 | 0.0372 | 0.1486 | 0.1164 |
| Fbs | -0.0955 | -0.3561 | 0.0600 | 0.7299 | -0.1208 | 0.1730 | 0.0677 | 0.5076 | 0.0613 | -0.0583 | 0.0372 | -0.1082 | -0.0243 |
| Restecg | -0.4297 | -0.0763 | -0.3599 | 0.2365 | 0.0830 | -0.5343 | 0.5194 | -0.1588 | 0.0545 | 0.0116 | -0.1266 | -0.1016 | 0.0851 |
| Thalach | 0.6918 | -0.1116 | -0.1824 | 0.3027 | 0.3369 | -0.1641 | -0.1013 | -0.2201 | -0.0506 | -0.0231 | 0.1295 | -0.0872 | -0.3967 |
| Exang | -0.6090 | -0.2413 | -0.2197 | -0.3055 | 0.1131 | 0.3412 | 0.1405 | 0.0190 | 0.0566 | -0.4549 | -0.1550 | 0.0170 | -0.2133 |
| Oldpeak | -0.6112 | 0.3781 | 0.3183 | 0.0715 | 0.1174 | -0.1963 | -0.1145 | -0.0462 | -0.2166 | -0.3211 | 0.3109 | -0.2402 | 0.0713 |
| Slope | -0.5337 | 0.4541 | 0.3153 | -0.1174 | 0.1212 | -0.3163 | 0.0717 | 0.3505 | 0.0047 | 0.1781 | -0.0778 | 0.1839 | -0.2852 |
| Ca | -0.4610 | -0.2838 | -0.0921 | 0.2204 | -0.5519 | -0.2124 | -0.2706 | -0.1743 | -0.3804 | -0.0361 | -0.1111 | 0.1733 | -0.0898 |
| Thal | -0.5879 | -0.3693 | 0.2624 | 0.0084 | 0.3431 | 0.0319 | -0.3143 | -0.1131 | 0.0426 | 0.2342 | -0.2901 | -0.2792 | 0.0061 |

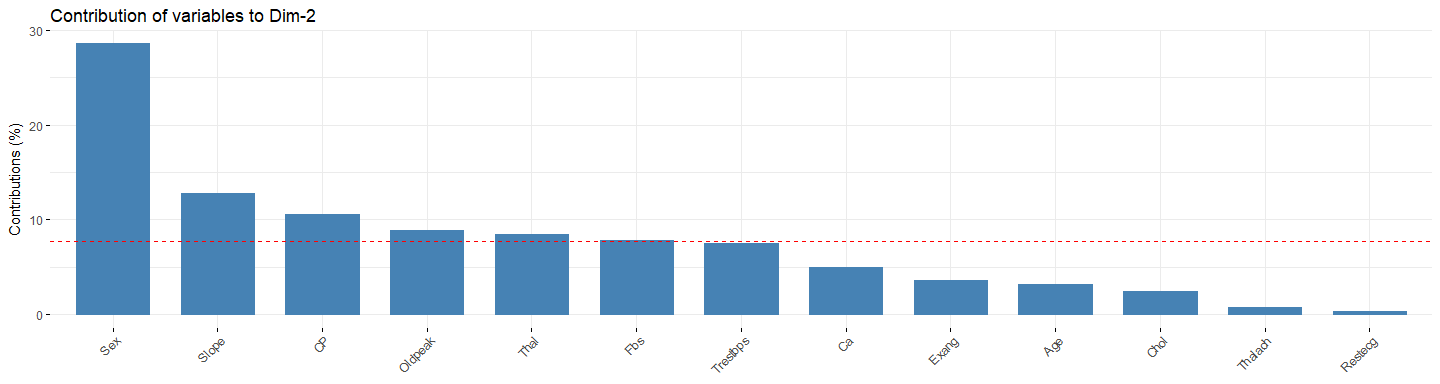
*# Let see the contribution of each variable to each PC*  
**kable**(var**$**contrib)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 | Dim.6 | Dim.7 | Dim.8 | Dim.9 | Dim.10 | Dim.11 | Dim.12 | Dim.13 |
| Age | 11.1255 | 3.1925 | 0.1067 | 1.2074 | 17.3048 | 3.7615 | 0.0052 | 10.4188 | 32.8661 | 4.5535 | 7.6188 | 1.7650 | 6.0743 |
| Sex | 0.4638 | 28.5818 | 16.9435 | 0.0025 | 3.0072 | 5.9379 | 0.0000 | 2.0902 | 11.6619 | 2.0707 | 7.8699 | 20.4189 | 0.9515 |
| CP | 8.7689 | 10.5740 | 9.0850 | 5.5572 | 2.3457 | 3.1211 | 2.1073 | 1.4481 | 11.8518 | 18.3152 | 26.6858 | 0.0045 | 0.1354 |
| Trestbps | 3.2484 | 7.5096 | 0.9199 | 21.5501 | 12.5806 | 19.1887 | 2.4040 | 10.4290 | 3.8900 | 0.5647 | 0.6049 | 16.0124 | 1.0978 |
| Chol | 3.3962 | 2.4154 | 33.5721 | 1.2015 | 7.1063 | 2.7918 | 28.1036 | 3.2881 | 7.7905 | 0.6998 | 0.3060 | 5.4272 | 3.9015 |
| Fbs | 0.2945 | 7.8640 | 0.2876 | 44.6390 | 1.4064 | 3.0329 | 0.6250 | 37.2891 | 0.5835 | 0.6236 | 0.3058 | 2.8785 | 0.1702 |
| Restecg | 5.9643 | 0.3607 | 10.3522 | 4.6851 | 0.6632 | 28.9266 | 36.7502 | 3.6491 | 0.4604 | 0.0248 | 3.5456 | 2.5354 | 2.0825 |
| Thalach | 15.4603 | 0.7725 | 2.6580 | 7.6783 | 10.9325 | 2.7283 | 1.3968 | 7.0077 | 0.3972 | 0.0982 | 3.7072 | 1.8698 | 45.2933 |
| Exang | 11.9796 | 3.6130 | 3.8558 | 7.8192 | 1.2311 | 11.7969 | 2.6906 | 0.0521 | 0.4975 | 37.9921 | 5.3117 | 0.0713 | 13.0890 |
| Oldpeak | 12.0675 | 8.8681 | 8.0938 | 0.4288 | 1.3273 | 3.9030 | 1.7863 | 0.3087 | 7.2748 | 18.9305 | 21.3757 | 14.1742 | 1.4615 |
| Slope | 9.2020 | 12.7917 | 7.9458 | 1.1556 | 1.4151 | 10.1363 | 0.6997 | 17.7718 | 0.0034 | 5.8221 | 1.3383 | 8.3076 | 23.4108 |
| Ca | 6.8661 | 4.9975 | 0.6776 | 4.0694 | 29.3396 | 4.5718 | 9.9748 | 4.3979 | 22.4415 | 0.2396 | 2.7268 | 7.3761 | 2.3215 |
| Thal | 11.1628 | 8.4593 | 5.5021 | 0.0059 | 11.3403 | 0.1033 | 13.4564 | 1.8496 | 0.2816 | 10.0652 | 18.6036 | 19.1592 | 0.0108 |

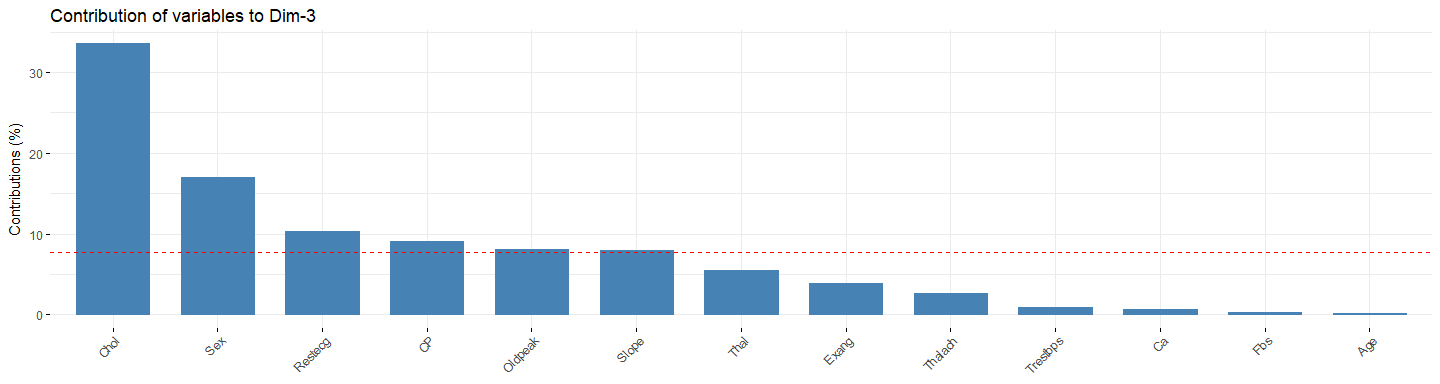
*# Contributions of variables to PC1*  
**fviz\_contrib**(pc.test, choice = "var", axes = 1, top = 13)



*# Contributions of variables to PC2*  
**fviz\_contrib**(pc.test, choice = "var", axes = 2, top = 13)



*# Contributions of variables to PC3*  
**fviz\_contrib**(pc.test, choice = "var", axes = 3, top = 13)



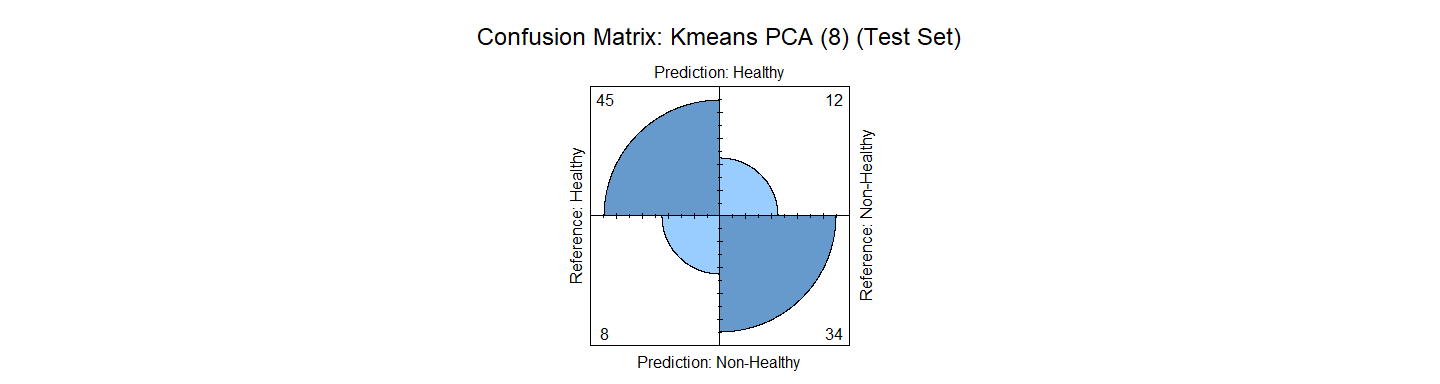
*# Extract the results for individuals*  
ind <- **get\_pca\_ind**(pc.test)  
  
*# Let's use the first 8 variables for the model*  
xx.pc.test <- **as.matrix**(ind**$**coord[,1**:**8])

Let’s check the K-means output

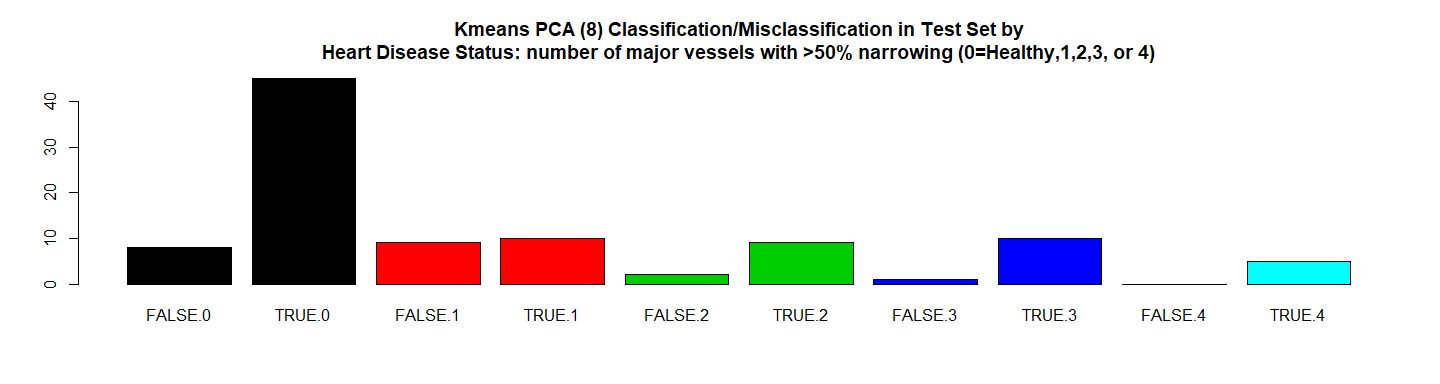
model.name <- "Kmeans PCA (8)"  
model.name6 <- model.name  
xx.km <- **eclust**(xx.pc.test,k=k,hc\_method=method, nstart=10, graph = FALSE, seed = best\_seed)  
kmeans\_class\_pca <- **correct\_kmeans\_ids**(Y.testData,xx.km**$**cluster)  
kmeans\_class\_pca <- **factor**(kmeans\_class\_pca)  
  
  
res.kmeans\_pca <- **confusionMatrix**(kmeans\_class\_pca, Y.testData)  
res.kmeans\_pca

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 45 12  
## 1 8 34  
##   
## Accuracy : 0.798   
## 95% CI : (0.705, 0.872)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 4.8e-08   
##   
## Kappa : 0.592   
## Mcnemar's Test P-Value : 0.502   
##   
## Sensitivity : 0.849   
## Specificity : 0.739   
## Pos Pred Value : 0.789   
## Neg Pred Value : 0.810   
## Prevalence : 0.535   
## Detection Rate : 0.455   
## Detection Prevalence : 0.576   
## Balanced Accuracy : 0.794   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.kmeans\_pca,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(kmeans\_class\_pca,  
 Y.testData,  
 main, testData)



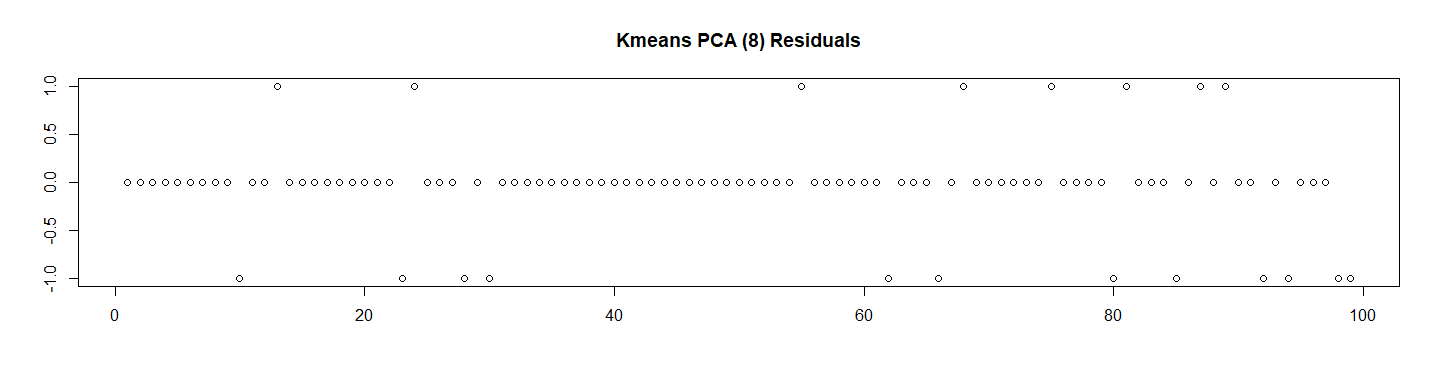
*# RMSE*  
(rmse\_kmeans\_pca <- **sqrt**(**mean**((**as.numeric**(kmeans\_class\_pca)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4495

*# Residuals Plot*  
(r\_kmeans\_pca <- (**as.numeric**(kmeans\_class\_pca)**-as.numeric**(Y.testData)))

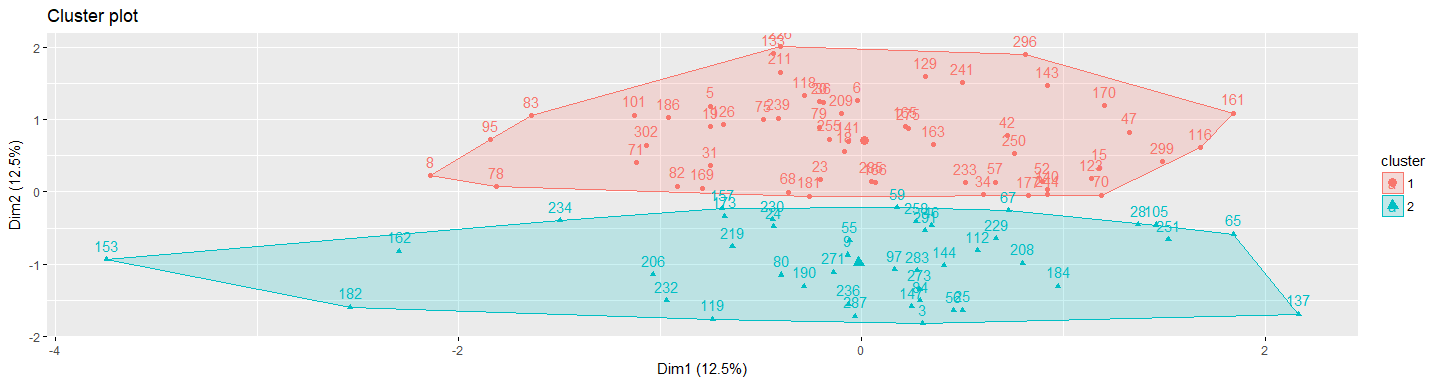
## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 -1  
## [24] 1 0 0 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0 0 0 -1 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 -1 1 0 0 0 -1 0 1 0 1 0 0 -1  
## [93] 0 -1 0 0 0 -1 -1

**plot**(r\_kmeans\_pca,   
 main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")

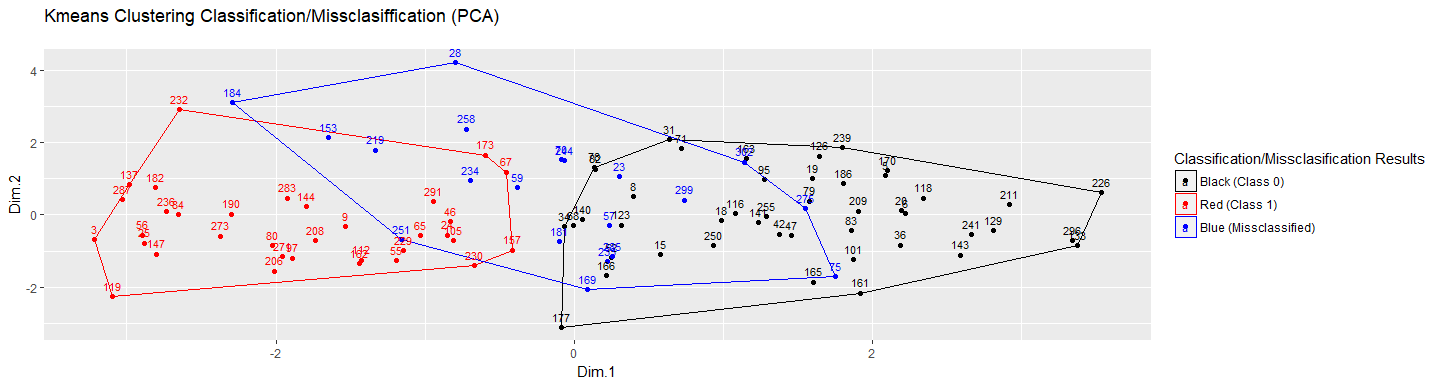


*# Let's Calculate the ROC*  
res.kmeans\_pca.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(kmeans\_class\_pca)))**$**auc  
  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 res.train.kmeans\_pca**$**overall[1] ,  
 "Prediction Accuracy in Test Set" =   
 res.kmeans\_pca**$**overall[1],  
 "RMSE Test"= rmse\_kmeans\_pca,  
 "ROC" = res.kmeans\_pca.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

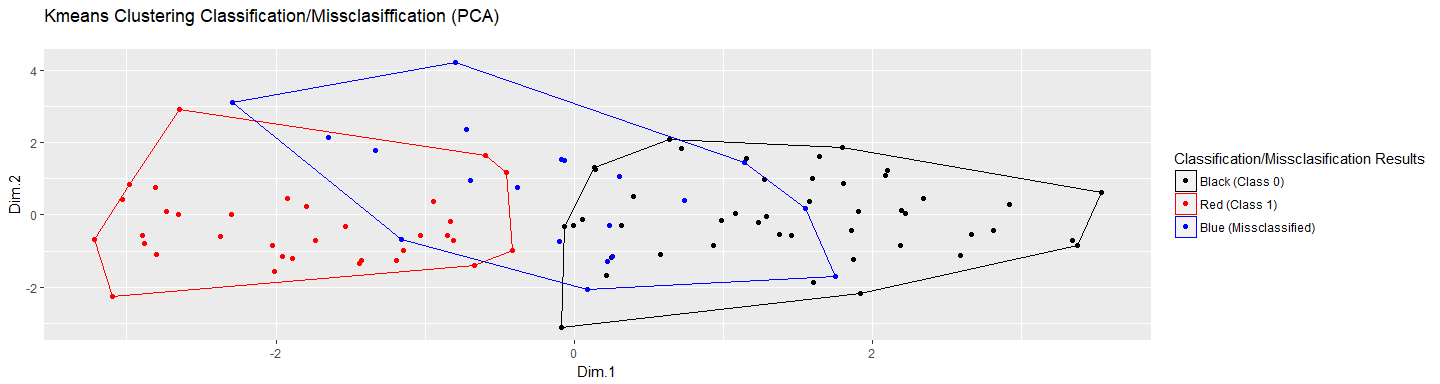
**fviz\_cluster**(xx.km)



**plot\_results**(xx.pc.test, Y.testData, kmeans\_class\_pca,   
 **paste0**("Kmeans Clustering Classification/Missclasiffication (PCA)\n"),  
 "Dim.1","Dim.2", labels = TRUE)



**plot\_results**(xx.pc.test, Y.testData, kmeans\_class\_pca,   
 **paste0**("Kmeans Clustering Classification/Missclasiffication (PCA)\n"),  
 "Dim.1","Dim.2", labels = FALSE)



## 8.2 All Data Whithout PCA

############################################  
## 8.2 All Data Whithout PCA  
model.name <- "Kmeans (13)"  
model.name7 <- model.name  
min\_k <- 2  
max\_k <- 4  
min\_seed <- 1  
max\_seed <- 100  
method <- "euclidean"  
  
*# Find Best Seed and K and Kmeans for the all Std Data*  
file.name <- **paste**(work.dir, "best\_seeds\_std\_m.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "best\_seeds\_std\_m-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)   
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
 t1 <- **proc.time**()  
 best\_seeds\_std\_m <- **find\_best\_seedv2**(X.trainData.std,  
 method,  
 min\_seed,  
 max\_seed,  
 min\_k,  
 max\_k,  
 Y.trainData)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**("Best Seed for Kmeans Finded ... time:", et))  
 **save**(best\_seeds\_std\_m, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Results for k=2 PCA"  
## [1] "======================="  
## [1] "Best Seed find: k=2 seed=1 errors=91 sum(whithinss)=2097.7357405401"  
## [1] ""  
## [1] "Results for k=3 PCA"  
## [1] "======================="  
## [1] "Best Seed find: k=3 seed=1 errors=139 sum(whithinss)=1920.4509302518"  
## [1] "Best Seed find: k=3 seed=2 errors=129 sum(whithinss)=1930.44287212597"  
## [1] "Best Seed find: k=3 seed=27 errors=127 sum(whithinss)=1930.80688693482"  
## [1] ""  
## [1] "Results for k=4 PCA"  
## [1] "======================="  
## [1] "Best Seed find: k=4 seed=1 errors=150 sum(whithinss)=1788.77207944515"  
## [1] "Best Seed find: k=4 seed=2 errors=147 sum(whithinss)=1778.61798103626"  
## [1] "Best Seed find: k=4 seed=12 errors=145 sum(whithinss)=1793.30071387004"  
## [1] "Best Seed find: k=4 seed=26 errors=143 sum(whithinss)=1796.41444368493"  
## [1] "Best Seed find: k=4 seed=61 errors=141 sum(whithinss)=1789.63176228369"  
## [1] ""  
## [1] ""  
## [1] "Final Results"  
## [1] "=============="  
## [1] "Best Seed find: k=2 seed=1 errors=91 sum(whithinss)=2097.7357405401"  
## [1] "Best Seed for Kmeans Finded ... time:4.5"

(k <- best\_seeds\_std\_m**$**best\_k)

## [1] 2

(best\_seed <- best\_seeds\_std\_m**$**best\_seed)

## [1] 1

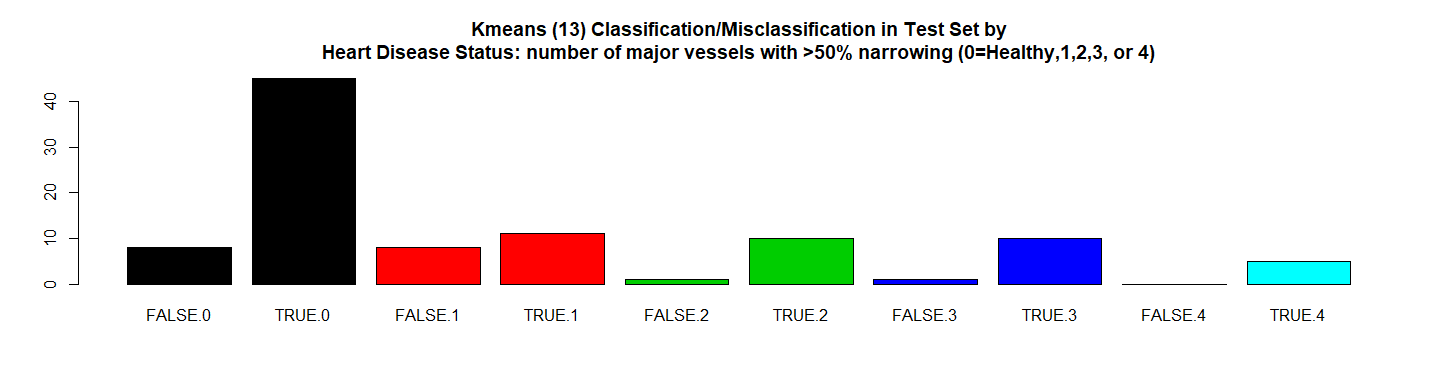
res.train.kmeans <- **confusionMatrix**(best\_seeds\_std\_m**$**best\_km**$**cluster, Y.trainData)  
res.train.kmeans

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 99 25  
## 1 8 66  
##   
## Accuracy : 0.833   
## 95% CI : (0.774, 0.882)  
## No Information Rate : 0.54   
## P-Value [Acc > NIR] : < 2e-16   
##   
## Kappa : 0.66   
## Mcnemar's Test P-Value : 0.00535   
##   
## Sensitivity : 0.925   
## Specificity : 0.725   
## Pos Pred Value : 0.798   
## Neg Pred Value : 0.892   
## Prevalence : 0.540   
## Detection Rate : 0.500   
## Detection Prevalence : 0.626   
## Balanced Accuracy : 0.825   
##   
## 'Positive' Class : 0   
##

*# Use the k and best seeds in the test set*  
  
xx.km2 <- **eclust**(X.testData.std,k=k,hc\_method=method, nstart=10,   
 graph = FALSE, seed = best\_seed)  
  
kmeans\_class <- **correct\_kmeans\_ids**(Y.testData,xx.km2**$**cluster)  
kmeans\_class <- **factor**(kmeans\_class)  
  
res.kmeans <- **confusionMatrix**(kmeans\_class, Y.testData)  
res.kmeans

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 45 10  
## 1 8 36  
##   
## Accuracy : 0.818   
## 95% CI : (0.728, 0.889)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 3.57e-09   
##   
## Kappa : 0.633   
## Mcnemar's Test P-Value : 0.814   
##   
## Sensitivity : 0.849   
## Specificity : 0.783   
## Pos Pred Value : 0.818   
## Neg Pred Value : 0.818   
## Prevalence : 0.535   
## Detection Rate : 0.455   
## Detection Prevalence : 0.556   
## Balanced Accuracy : 0.816   
##   
## 'Positive' Class : 0   
##

main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(kmeans\_class,  
 Y.testData,  
 main, testData)



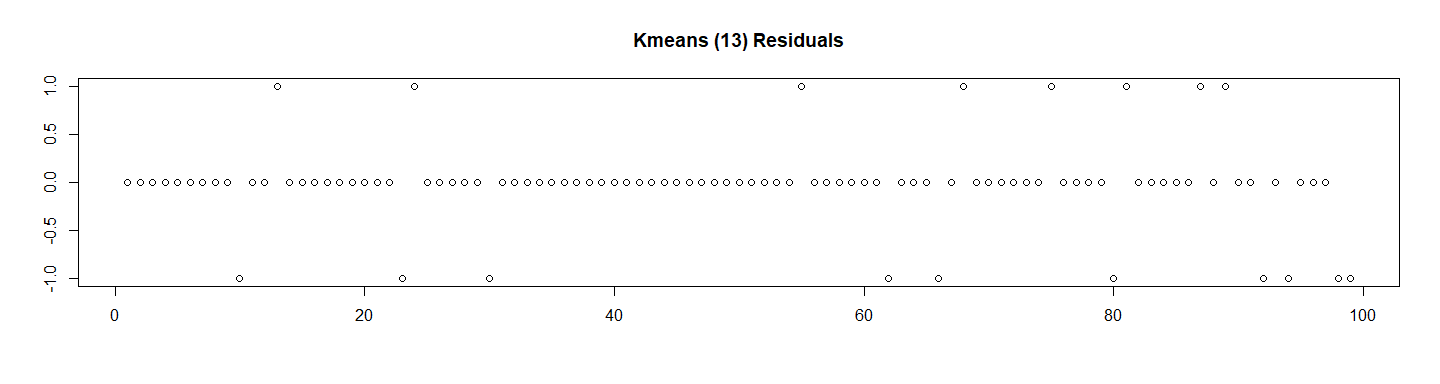
*# RMSE*  
(rmse\_kmeans <- **sqrt**(**mean**((**as.numeric**(kmeans\_class)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4264

*# Residuals Plot*  
(r\_kmeans <- (**as.numeric**(kmeans\_class)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 -1  
## [24] 1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 -1 0 0 0 -1 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 -1 1 0 0 0 0 0 1 0 1 0 0 -1  
## [93] 0 -1 0 0 0 -1 -1

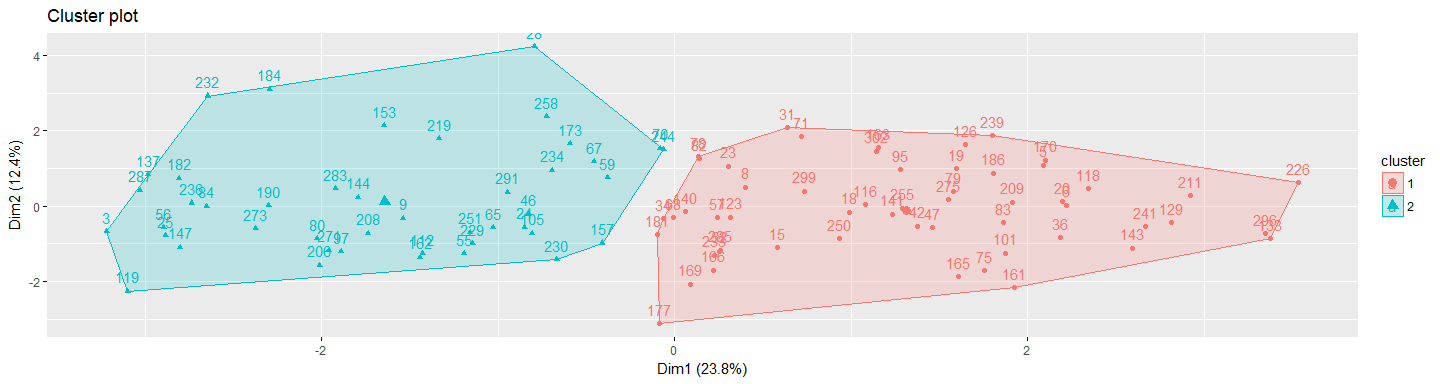
**plot**(r\_kmeans, main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



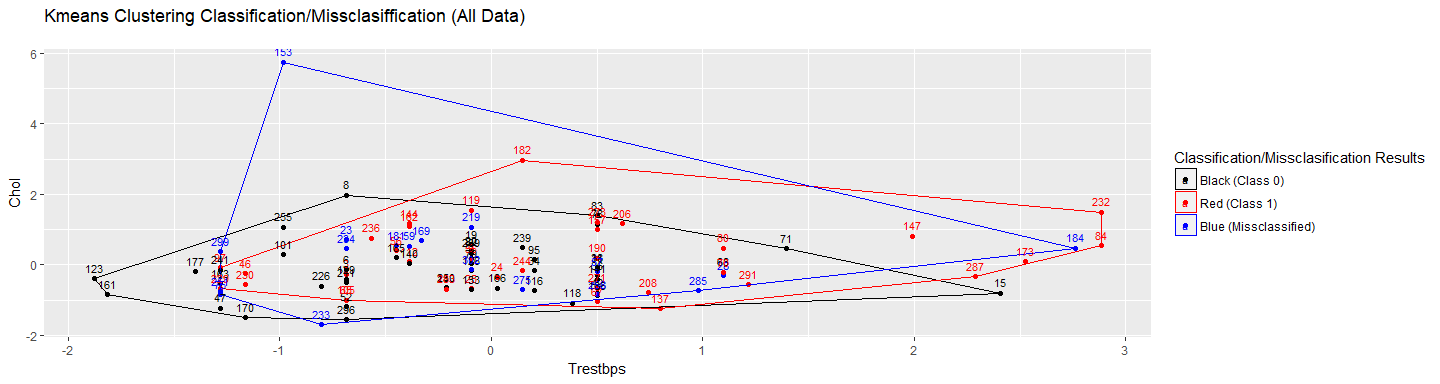
*# Let's Calculate the ROC*  
res.kmeans.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(kmeans\_class)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =  
 res.train.kmeans**$**overall[1] ,  
 "Prediction Accuracy in Test Set" =   
 res.kmeans**$**overall[1],  
 "RMSE Test"= rmse\_kmeans,  
 "ROC" = res.kmeans.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

Let’s see the results

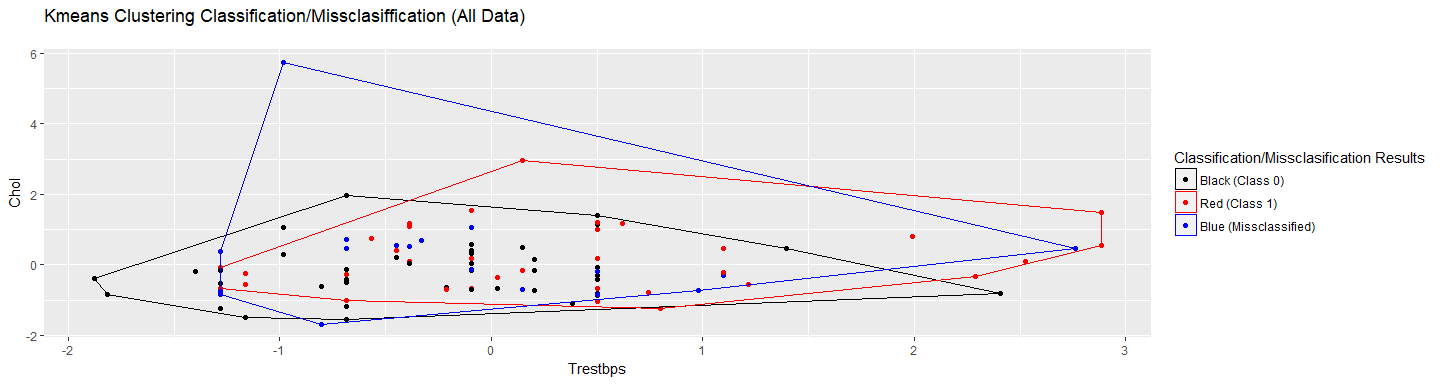
**fviz\_cluster**(xx.km2)



**plot\_results**(X.testData.std, Y.testData, kmeans\_class,   
 **paste0**("Kmeans Clustering Classification/Missclasiffication (All Data)\n"),  
 "Trestbps","Chol", labels = TRUE)



**plot\_results**(X.testData.std, Y.testData, kmeans\_class,   
 **paste0**("Kmeans Clustering Classification/Missclasiffication (All Data)\n"),  
 "Trestbps","Chol", labels = FALSE)



# 9. Support Vector Machine (SVM)

## 9.1 SVM Radial (svmRadial)

############################################  
*# 9. Support Vector Machine (SVM)*  
############################################  
## 9.1 SVM Radial (svmRadial)  
model.name <- "SVM Radial"  
model.name8 <- model.name  
(d <- **log2**(13))

## [1] 3.7

(sigma\_values <- 2**^**(**c**(**-**d**-**3, **-**d**-**2, **-**d**-**1, **-**d, **-**d **+** 1, **-**d **+**2, **-**d**+**3)))

## [1] 0.009615 0.019231 0.038462 0.076923 0.153846 0.307692 0.615385

(c\_values <- **c**(0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000))

## [1] 1e-03 1e-02 1e-01 1e+00 1e+01 1e+02 1e+03 1e+04

kernel\_method\_type <- "svmRadial"  
*# Grid Parameters*  
(grid\_radial <- **expand.grid**(C = c\_values, sigma = sigma\_values))

## C sigma  
## 1 1e-03 0.009615  
## 2 1e-02 0.009615  
## 3 1e-01 0.009615  
## 4 1e+00 0.009615  
## 5 1e+01 0.009615  
## 6 1e+02 0.009615  
## 7 1e+03 0.009615  
## 8 1e+04 0.009615  
## 9 1e-03 0.019231  
## 10 1e-02 0.019231  
## 11 1e-01 0.019231  
## 12 1e+00 0.019231  
## 13 1e+01 0.019231  
## 14 1e+02 0.019231  
## 15 1e+03 0.019231  
## 16 1e+04 0.019231  
## 17 1e-03 0.038462  
## 18 1e-02 0.038462  
## 19 1e-01 0.038462  
## 20 1e+00 0.038462  
## 21 1e+01 0.038462  
## 22 1e+02 0.038462  
## 23 1e+03 0.038462  
## 24 1e+04 0.038462  
## 25 1e-03 0.076923  
## 26 1e-02 0.076923  
## 27 1e-01 0.076923  
## 28 1e+00 0.076923  
## 29 1e+01 0.076923  
## 30 1e+02 0.076923  
## 31 1e+03 0.076923  
## 32 1e+04 0.076923  
## 33 1e-03 0.153846  
## 34 1e-02 0.153846  
## 35 1e-01 0.153846  
## 36 1e+00 0.153846  
## 37 1e+01 0.153846  
## 38 1e+02 0.153846  
## 39 1e+03 0.153846  
## 40 1e+04 0.153846  
## 41 1e-03 0.307692  
## 42 1e-02 0.307692  
## 43 1e-01 0.307692  
## 44 1e+00 0.307692  
## 45 1e+01 0.307692  
## 46 1e+02 0.307692  
## 47 1e+03 0.307692  
## 48 1e+04 0.307692  
## 49 1e-03 0.615385  
## 50 1e-02 0.615385  
## 51 1e-01 0.615385  
## 52 1e+00 0.615385  
## 53 1e+01 0.615385  
## 54 1e+02 0.615385  
## 55 1e+03 0.615385  
## 56 1e+04 0.615385

*# Cross Validation Parameters*  
trctrl <- **trainControl**(method = "repeatedcv",   
 number = 10,  
 repeats = 3)  
  
file.name <- **paste**(work.dir, "svm\_radial.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "svm\_radial-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)   
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
 t1 <- **proc.time**()  
 svm\_Radial\_Grid <- **train**(y = **factor**(Y.trainData),  
 x = X.trainData,  
 method = kernel\_method\_type,  
 trControl=trctrl,  
 preProcess = **c**("center", "scale"),   
 trace = FALSE,  
 tuneGrid = grid\_radial)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:",et))  
 **save**(svm\_Radial\_Grid, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "SVM Radial Executed ... time:58"

We can see the main results here:

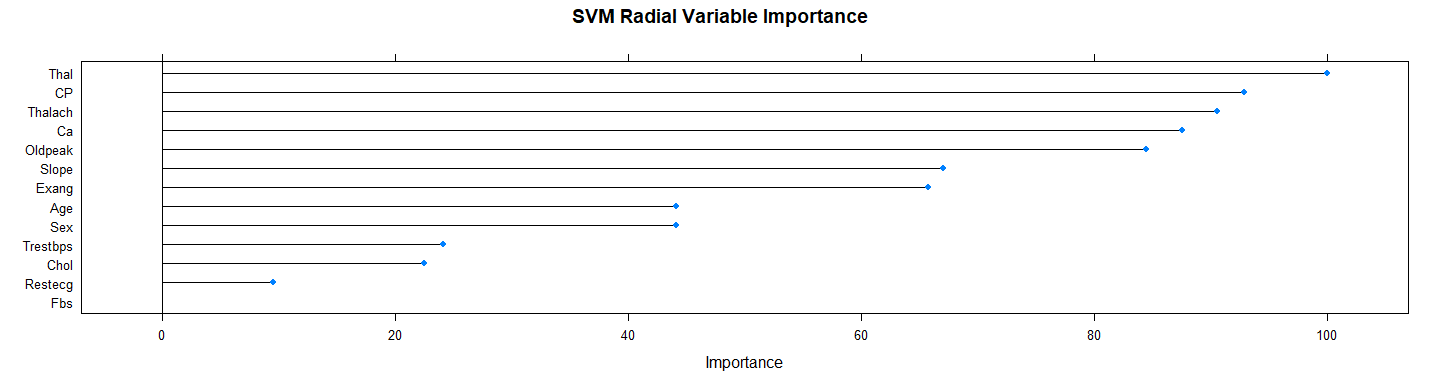
*# Let's see some details about the model*  
svm\_Radial\_Grid

## Support Vector Machines with Radial Basis Function Kernel   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 178, 179, 179, 178, 178, 178, ...   
## Resampling results across tuning parameters:  
##   
## C sigma Accuracy Kappa   
## 1e-03 0.009615 0.5403 0.0000  
## 1e-03 0.019231 0.5403 0.0000  
## 1e-03 0.038462 0.5403 0.0000  
## 1e-03 0.076923 0.5403 0.0000  
## 1e-03 0.153846 0.5403 0.0000  
## 1e-03 0.307692 0.5403 0.0000  
## 1e-03 0.615385 0.5403 0.0000  
## 1e-02 0.009615 0.5403 0.0000  
## 1e-02 0.019231 0.5403 0.0000  
## 1e-02 0.038462 0.5403 0.0000  
## 1e-02 0.076923 0.5403 0.0000  
## 1e-02 0.153846 0.5403 0.0000  
## 1e-02 0.307692 0.5403 0.0000  
## 1e-02 0.615385 0.5403 0.0000  
## 1e-01 0.009615 0.7528 0.4821  
## 1e-01 0.019231 0.8448 0.6813  
## 1e-01 0.038462 0.8533 0.7003  
## 1e-01 0.076923 0.8516 0.6967  
## 1e-01 0.153846 0.6609 0.2801  
## 1e-01 0.307692 0.5403 0.0000  
## 1e-01 0.615385 0.5403 0.0000  
## 1e+00 0.009615 0.8501 0.6955  
## 1e+00 0.019231 0.8467 0.6888  
## 1e+00 0.038462 0.8332 0.6622  
## 1e+00 0.076923 0.8314 0.6593  
## 1e+00 0.153846 0.8164 0.6299  
## 1e+00 0.307692 0.7912 0.5830  
## 1e+00 0.615385 0.6657 0.3007  
## 1e+01 0.009615 0.8350 0.6651  
## 1e+01 0.019231 0.8168 0.6295  
## 1e+01 0.038462 0.8085 0.6127  
## 1e+01 0.076923 0.7785 0.5542  
## 1e+01 0.153846 0.7661 0.5308  
## 1e+01 0.307692 0.7625 0.5262  
## 1e+01 0.615385 0.6694 0.3157  
## 1e+02 0.009615 0.7988 0.5923  
## 1e+02 0.019231 0.7783 0.5537  
## 1e+02 0.038462 0.7411 0.4803  
## 1e+02 0.076923 0.7614 0.5205  
## 1e+02 0.153846 0.7644 0.5275  
## 1e+02 0.307692 0.7625 0.5262  
## 1e+02 0.615385 0.6694 0.3157  
## 1e+03 0.009615 0.7376 0.4729  
## 1e+03 0.019231 0.7159 0.4288  
## 1e+03 0.038462 0.7328 0.4635  
## 1e+03 0.076923 0.7614 0.5205  
## 1e+03 0.153846 0.7644 0.5275  
## 1e+03 0.307692 0.7625 0.5262  
## 1e+03 0.615385 0.6694 0.3157  
## 1e+04 0.009615 0.7088 0.4137  
## 1e+04 0.019231 0.7159 0.4288  
## 1e+04 0.038462 0.7328 0.4635  
## 1e+04 0.076923 0.7614 0.5205  
## 1e+04 0.153846 0.7644 0.5275  
## 1e+04 0.307692 0.7625 0.5262  
## 1e+04 0.615385 0.6694 0.3157  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were sigma = 0.03846 and C = 0.1.

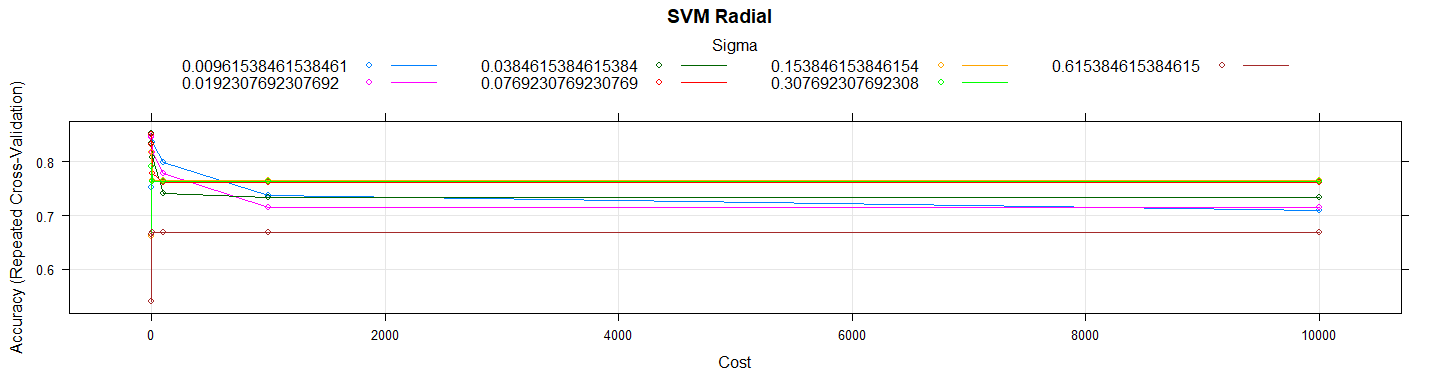
**summary**(svm\_Radial\_Grid)

## Length Class Mode   
## 1 ksvm S4

**plot**(**varImp**(svm\_Radial\_Grid),   
 main = **paste0**(model.name," Variable Importance"))



**plot**(svm\_Radial\_Grid, main = model.name)



*# Let's check the detail in the Training Set*  
res.svmradial.training <- **confusionMatrix**(svm\_Radial\_Grid**$**finalModel**@**fitted, Y.trainData)  
res.svmradial.training

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 99 21  
## 1 8 70  
##   
## Accuracy : 0.854   
## 95% CI : (0.796, 0.9)  
## No Information Rate : 0.54   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.702   
## Mcnemar's Test P-Value : 0.0259   
##   
## Sensitivity : 0.925   
## Specificity : 0.769   
## Pos Pred Value : 0.825   
## Neg Pred Value : 0.897   
## Prevalence : 0.540   
## Detection Rate : 0.500   
## Detection Prevalence : 0.606   
## Balanced Accuracy : 0.847   
##   
## 'Positive' Class : 0   
##

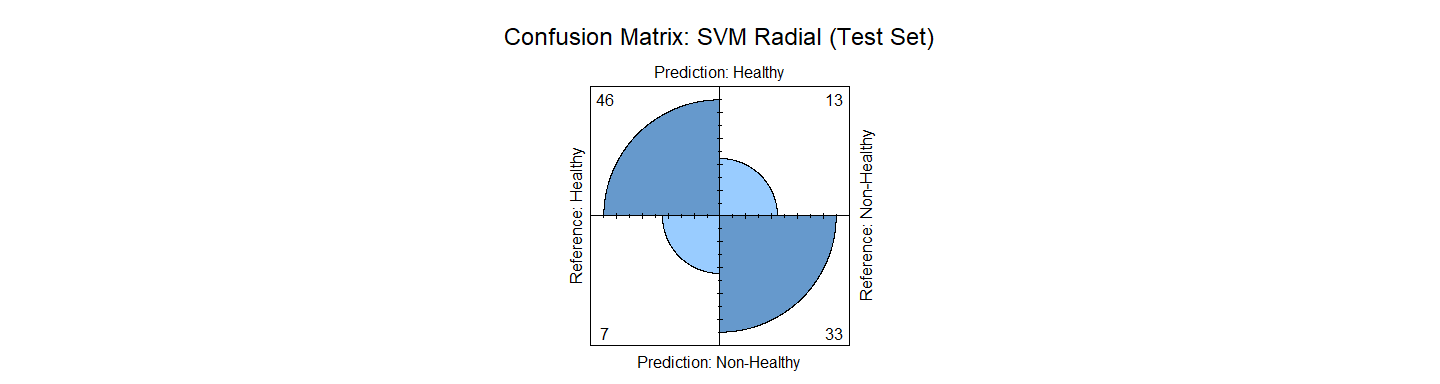
*# Let's use the model in test set*  
test\_pred\_rgrid <- **predict**(svm\_Radial\_Grid, newdata = X.testData)  
test\_pred\_rgrid

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 1 0 0 1 0 1 1 0 1  
## [71] 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 1 0 0 1 1 0 1 0 1 1 0 0 0  
## Levels: 0 1

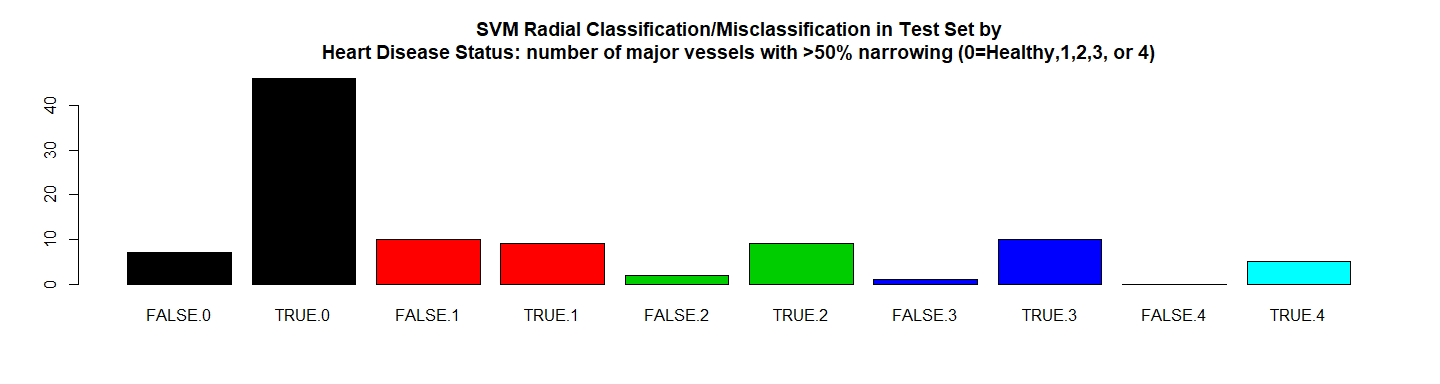
res.svmradial <- **confusionMatrix**(test\_pred\_rgrid, Y.testData)  
  
res.svmradial

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 46 13  
## 1 7 33  
##   
## Accuracy : 0.798   
## 95% CI : (0.705, 0.872)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 4.8e-08   
##   
## Kappa : 0.59   
## Mcnemar's Test P-Value : 0.264   
##   
## Sensitivity : 0.868   
## Specificity : 0.717   
## Pos Pred Value : 0.780   
## Neg Pred Value : 0.825   
## Prevalence : 0.535   
## Detection Rate : 0.465   
## Detection Prevalence : 0.596   
## Balanced Accuracy : 0.793   
##   
## 'Positive' Class : 0   
##

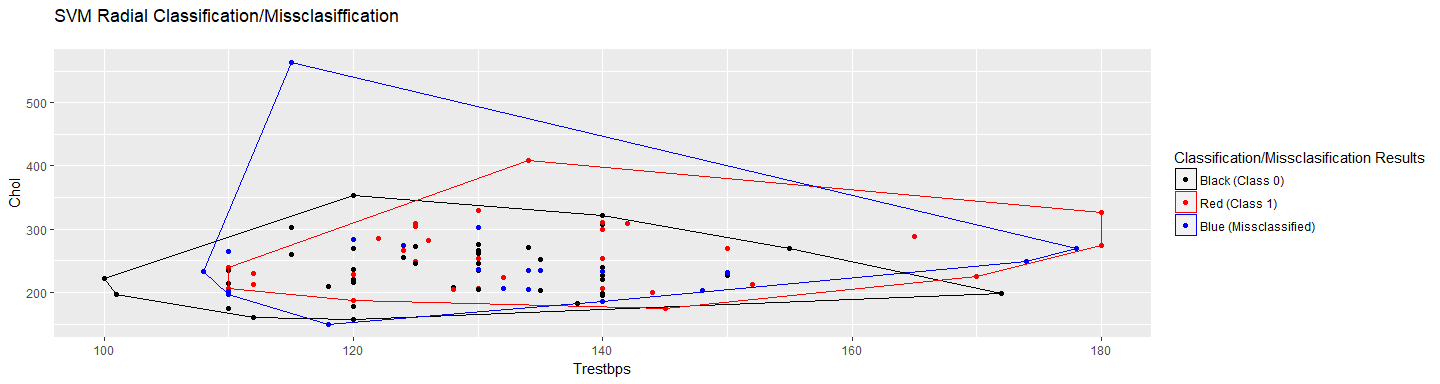
**print\_confusionm**(res.svmradial,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



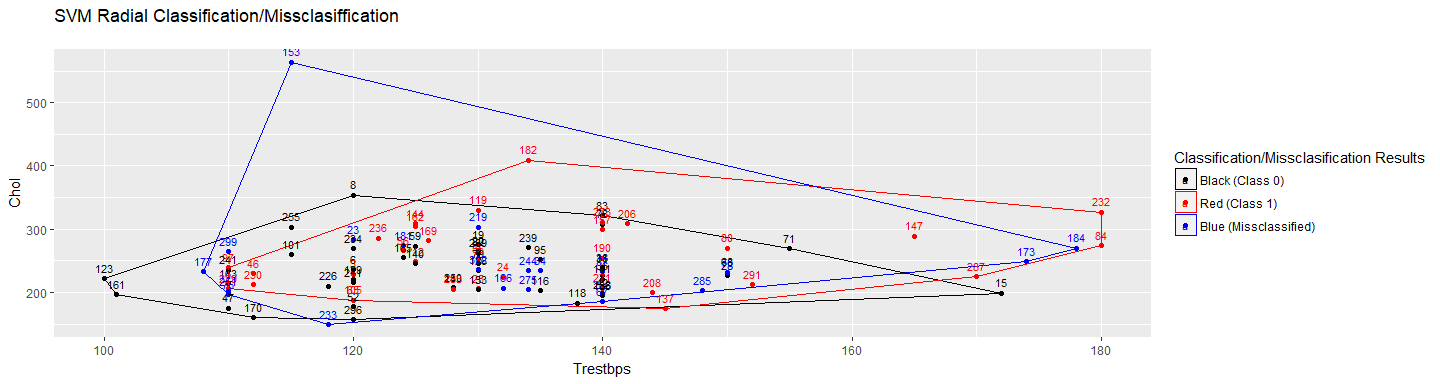
main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_rgrid,  
 Y.testData,  
 main, testData)



**plot\_results**(X.testData, Y.testData,test\_pred\_rgrid ,   
 **paste0**(model.name," Classification/Missclasiffication\n"),   
 "Trestbps","Chol",labels = FALSE)



**plot\_results**(X.testData, Y.testData,test\_pred\_rgrid,   
 **paste0**(model.name," Classification/Missclasiffication\n"),  
 "Trestbps","Chol",labels = TRUE)



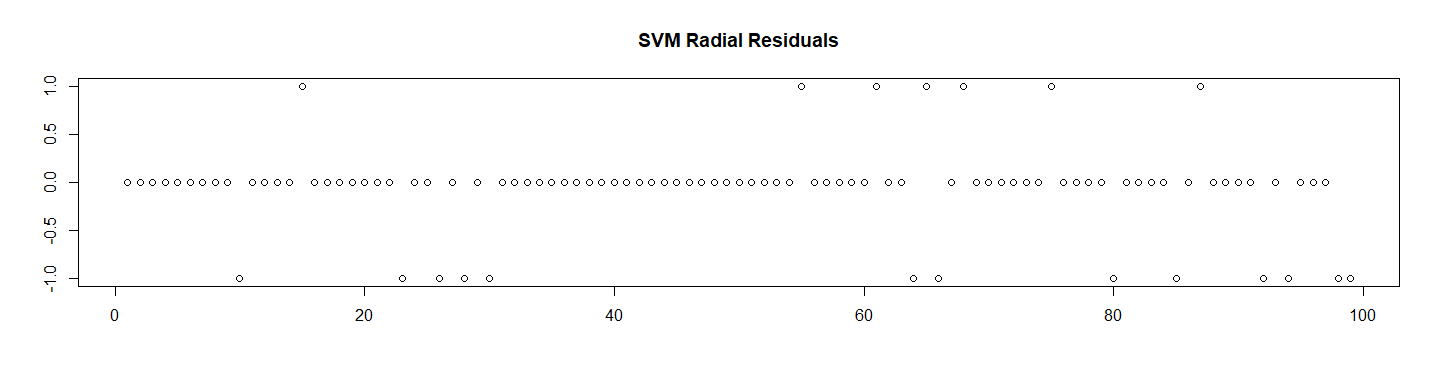
*# RMSE Resulting*  
(rmse\_svmradial <- **sqrt**(**mean**((**as.numeric**(test\_pred\_rgrid)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4495

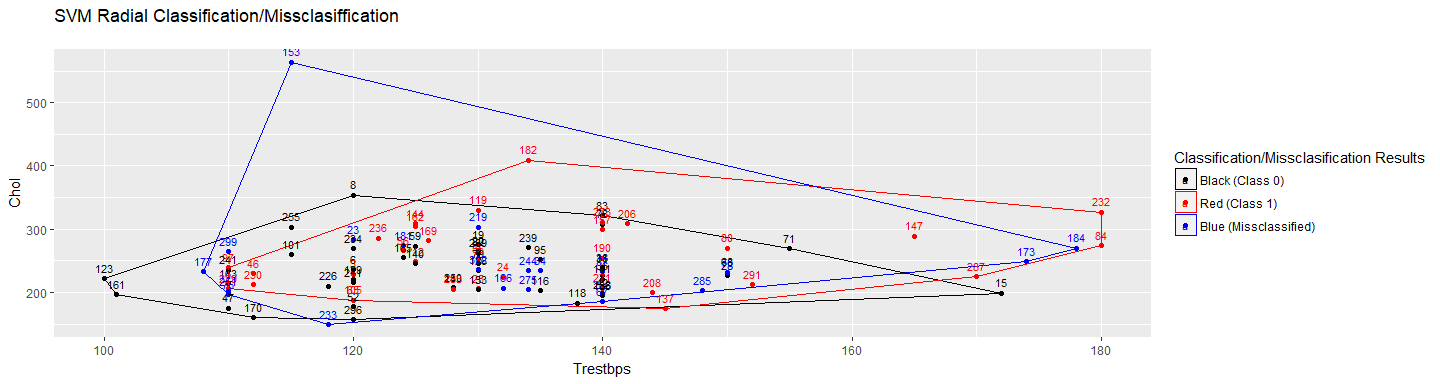
*# Residuals Plot*  
(r\_svmradial <- (**as.numeric**(test\_pred\_rgrid)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0 0 0 -1  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 -1 1 -1 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 -1 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 -1 0 0 0 -1 -1

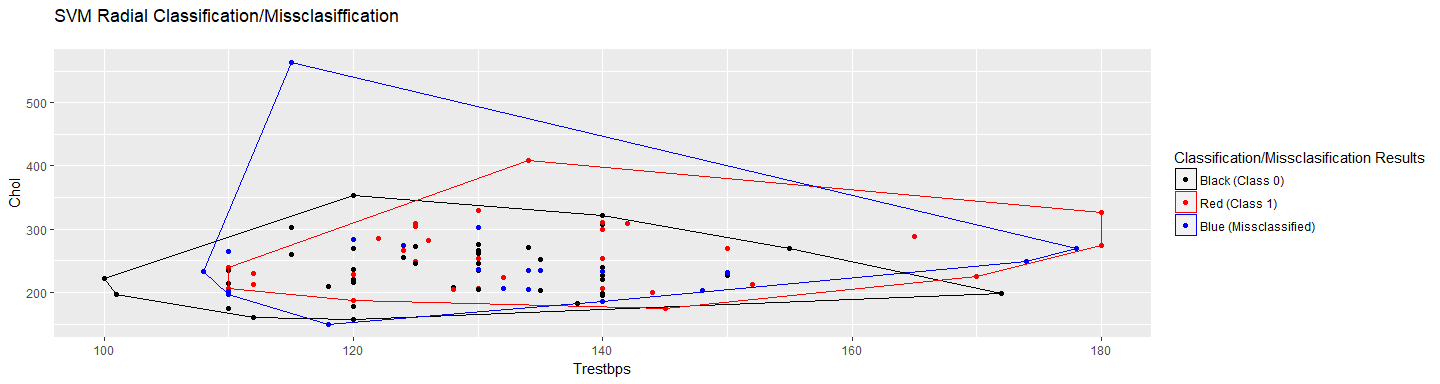
**plot**(r\_svmradial,   
 main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 res.svmradial.training**$**overall[1] ,  
 "Prediction Accuracy in Test Set" =   
 res.svmradial**$**overall[1],  
 "RMSE Test"= rmse\_svmradial,  
 "ROC" = NA,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))  
  
**plot\_results**(X.testData, Y.testData, test\_pred\_rgrid ,   
 **paste0**(model.name," Classification/Missclasiffication\n"),  
 "Trestbps","Chol",labels = TRUE)



**plot\_results**(X.testData, Y.testData, test\_pred\_rgrid ,   
 **paste0**(model.name," Classification/Missclasiffication\n"),  
 "Trestbps","Chol", labels = FALSE)



## 9.2 Linear SVM (svmLinear)

############################################  
## 9.2 Linear SVM (svmLinear)  
*# Setting Hyperparameters According with Paper*  
model.name <- "SVM Linear"  
model.name9 <- model.name  
kernel\_method\_type <- "svmLinear"  
  
grid\_Linear <- **expand.grid**(C = c\_values)  
  
file.name <- **paste**(work.dir, "svm\_linear.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "svm\_linear-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)   
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
 t1 <- **proc.time**()  
 svm\_Linear\_Grid <- **train**(y = **factor**(Y.trainData),   
 x = X.trainData,  
 method = kernel\_method\_type,  
 trControl=trctrl,  
 preProcess = **c**("center", "scale"),  
 tuneGrid = grid\_Linear,  
 trace = FALSE,  
 tuneLength = 10)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(svm\_Linear\_Grid, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "SVM Linear Executed ... time:169"

We can see the main results here:

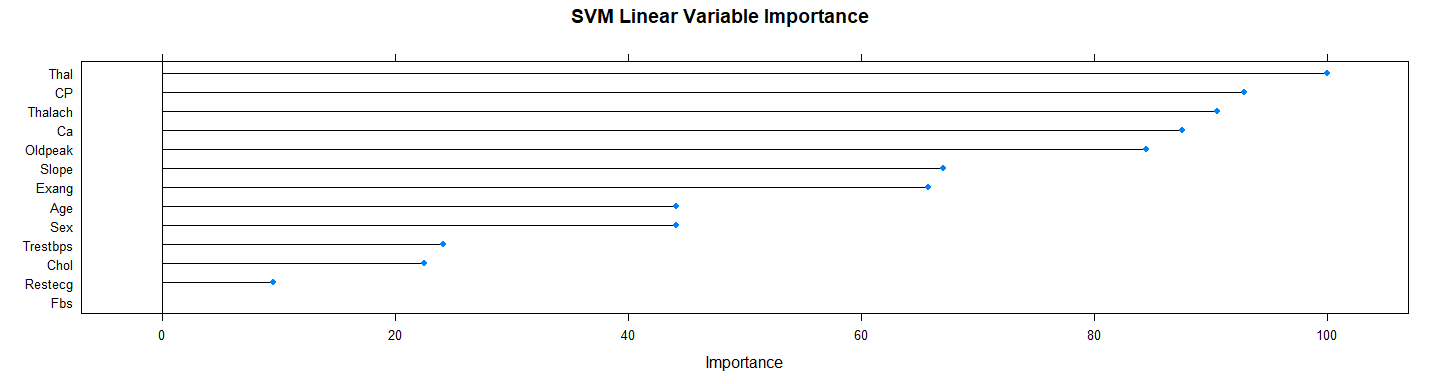
*# Let's see some details about the model*  
svm\_Linear\_Grid

## Support Vector Machines with Linear Kernel   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 178, 178, 178, 178, 178, 179, ...   
## Resampling results across tuning parameters:  
##   
## C Accuracy Kappa   
## 1e-03 0.5637 0.05547  
## 1e-02 0.8542 0.70437  
## 1e-01 0.8389 0.67409  
## 1e+00 0.8373 0.67076  
## 1e+01 0.8406 0.67737  
## 1e+02 0.8424 0.68079  
## 1e+03 0.8424 0.68079  
## 1e+04 0.8374 0.67086  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was C = 0.01.

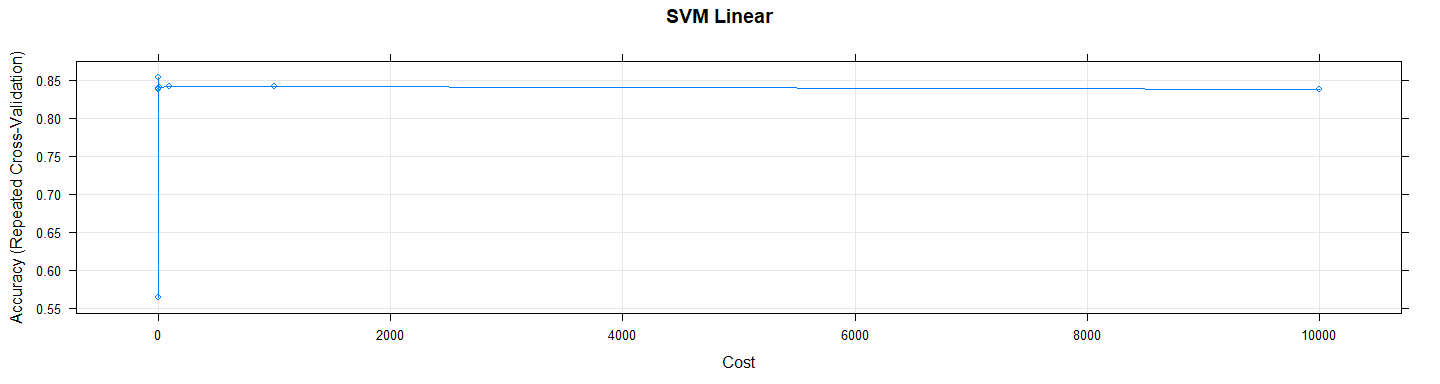
**summary**(svm\_Linear\_Grid)

## Length Class Mode   
## 1 ksvm S4

**plot**(**varImp**(svm\_Linear\_Grid),   
 main = **paste0**(model.name," Variable Importance"))



**plot**(svm\_Linear\_Grid, main = model.name)



*# Let's check the detail in the Training Set*  
res.svmlinear.training <- **confusionMatrix**(svm\_Linear\_Grid**$**finalModel**@**fitted, Y.trainData)  
res.svmlinear.training

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 98 19  
## 1 9 72  
##   
## Accuracy : 0.859   
## 95% CI : (0.802, 0.904)  
## No Information Rate : 0.54   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.713   
## Mcnemar's Test P-Value : 0.089   
##   
## Sensitivity : 0.916   
## Specificity : 0.791   
## Pos Pred Value : 0.838   
## Neg Pred Value : 0.889   
## Prevalence : 0.540   
## Detection Rate : 0.495   
## Detection Prevalence : 0.591   
## Balanced Accuracy : 0.854   
##   
## 'Positive' Class : 0   
##

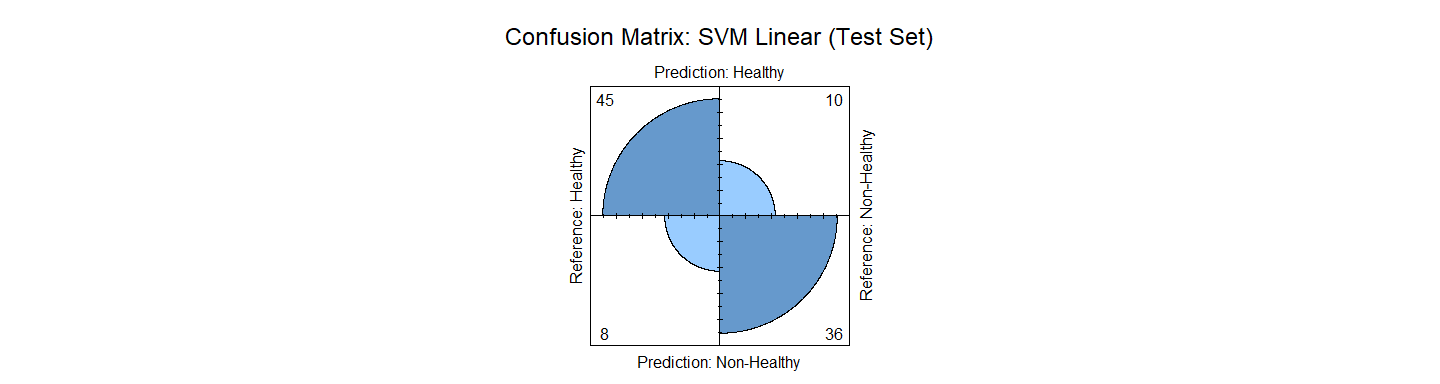
*# Let's use the model in test set*  
  
test\_pred\_lgrid <- **predict**(svm\_Linear\_Grid, newdata = X.testData)  
test\_pred\_lgrid

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0  
## Levels: 0 1

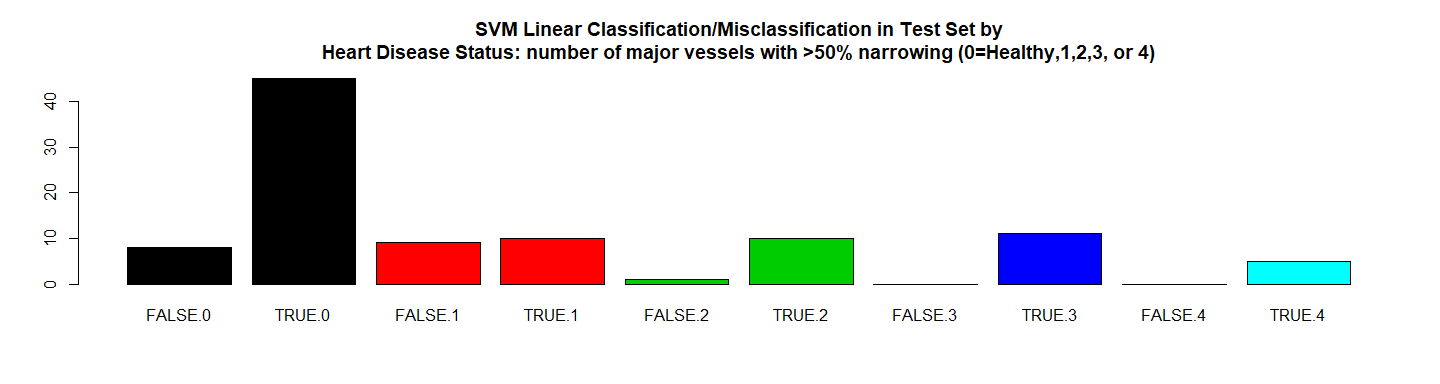
res.svmlinear <- **confusionMatrix**(test\_pred\_lgrid, Y.testData)  
  
res.svmlinear

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 45 10  
## 1 8 36  
##   
## Accuracy : 0.818   
## 95% CI : (0.728, 0.889)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 3.57e-09   
##   
## Kappa : 0.633   
## Mcnemar's Test P-Value : 0.814   
##   
## Sensitivity : 0.849   
## Specificity : 0.783   
## Pos Pred Value : 0.818   
## Neg Pred Value : 0.818   
## Prevalence : 0.535   
## Detection Rate : 0.455   
## Detection Prevalence : 0.556   
## Balanced Accuracy : 0.816   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.svmlinear,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_lgrid,  
 Y.testData,  
 main, testData)



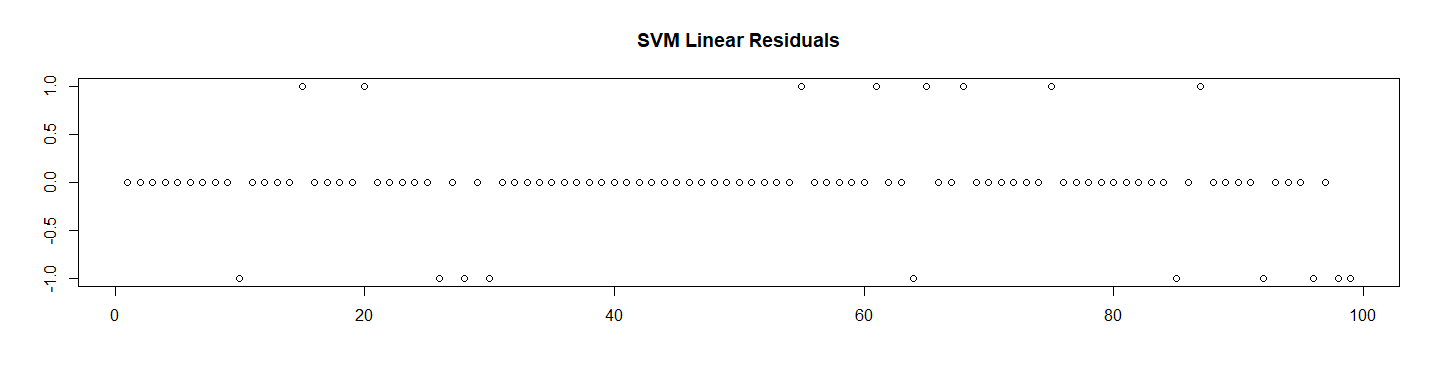
*# RMSE Resulting*  
(rmse\_svmlinear <- **sqrt**(**mean**((**as.numeric**(test\_pred\_lgrid)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4264

*# Residuals Plot*  
(r\_svmlinear <- (**as.numeric**(test\_pred\_lgrid)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 1 0 0 0  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 -1 -1

**plot**(r\_svmlinear, main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 res.svmlinear.training**$**overall[1] ,  
 "Prediction Accuracy in Test Set" =   
 res.svmlinear**$**overall[1],  
 "RMSE Test"= rmse\_svmlinear,  
 "ROC" = NA,  
   
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 10. Neural Networks (nnet)

############################################  
*# 10. Neural Networks*  
model.name <- "Neural Networks (nnet)"  
model.name10 <- model.name  
file.name <- **paste**(work.dir, "nnet.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "nnet-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 grid\_n\_net <- **expand.grid**(decay = **c**(0.0005, 0.005, 0.05), size = **c**(0,1,2,3))  
 t1 <- **proc.time**()  
 n\_net <- **train**(y = **factor**(Y.trainData),   
 x = X.trainData,  
 method="nnet",  
 preProcess = **c**("center", "scale"),  
 tuneGrid = grid\_n\_net,  
 trace = FALSE,  
 maxit = 500)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(n\_net, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Neural Networks (nnet) Executed ... time:9.5"

Let’s see the results:

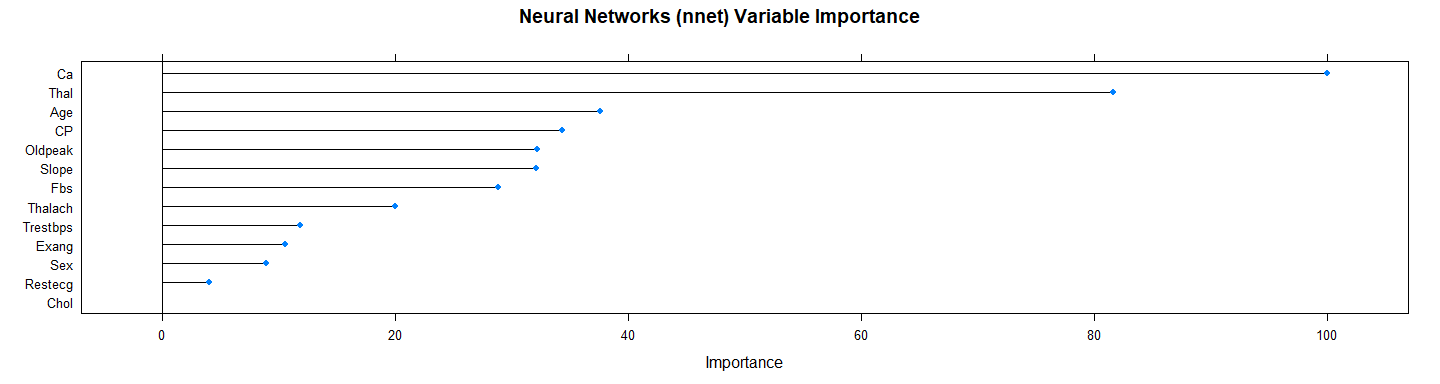
*# Let's see some details about the model*  
n\_net

## Neural Network   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 198, 198, 198, 198, 198, 198, ...   
## Resampling results across tuning parameters:  
##   
## decay size Accuracy Kappa   
## 5e-04 0 NaN NaN  
## 5e-04 1 0.7953 0.5853  
## 5e-04 2 0.7851 0.5672  
## 5e-04 3 0.7681 0.5324  
## 5e-03 0 NaN NaN  
## 5e-03 1 0.7939 0.5821  
## 5e-03 2 0.7723 0.5430  
## 5e-03 3 0.7815 0.5608  
## 5e-02 0 NaN NaN  
## 5e-02 1 0.7918 0.5789  
## 5e-02 2 0.7715 0.5391  
## 5e-02 3 0.7714 0.5407  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were size = 1 and decay = 5e-04.

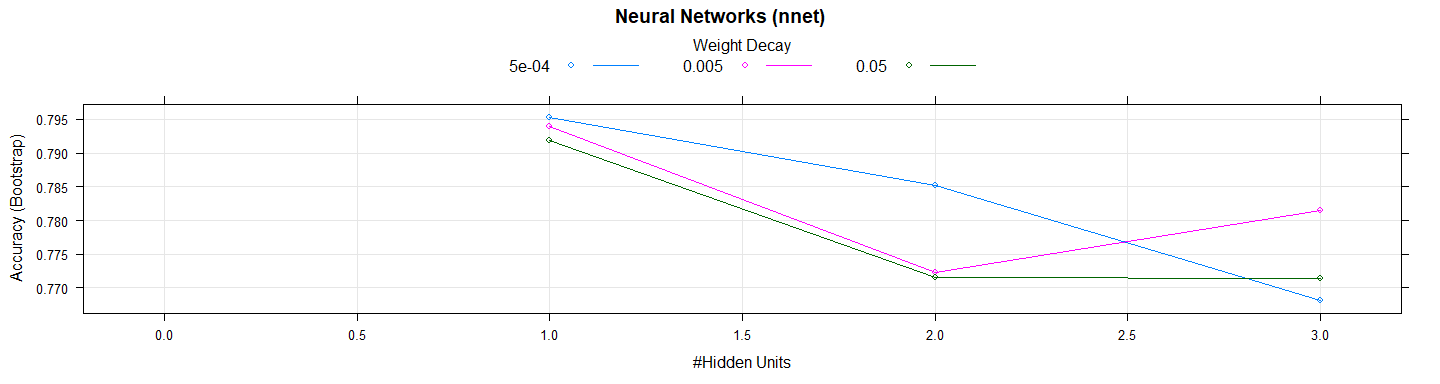
**summary**(n\_net)

## a 13-1-1 network with 16 weights  
## options were - entropy fitting decay=5e-04  
## b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1 i6->h1 i7->h1 i8->h1   
## -10.64 -9.53 -2.59 8.73 3.29 -0.43 -7.41 1.40 -5.27   
## i9->h1 i10->h1 i11->h1 i12->h1 i13->h1   
## -2.98 8.22 8.19 24.62 20.17   
## b->o h1->o   
## -1.89 5.57

**plot**(**varImp**(n\_net), main = **paste0**(model.name, " Variable Importance"))



**plot**(n\_net, main = model.name)



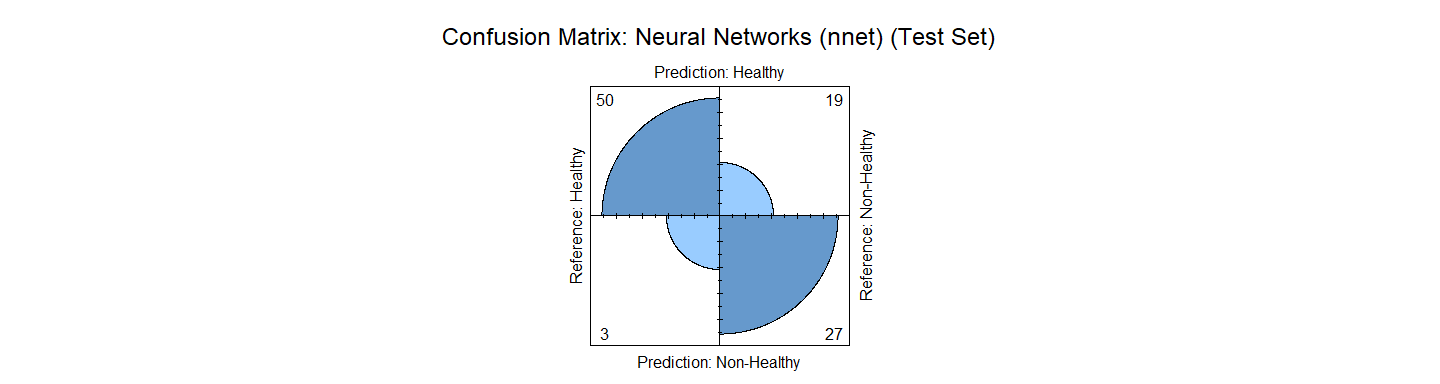
*# Let's use the model in test set*  
test\_pred\_n\_net <- **predict**(n\_net, newdata = X.testData)  
test\_pred\_n\_net

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0  
## [36] 0 0 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 0 0  
## Levels: 0 1

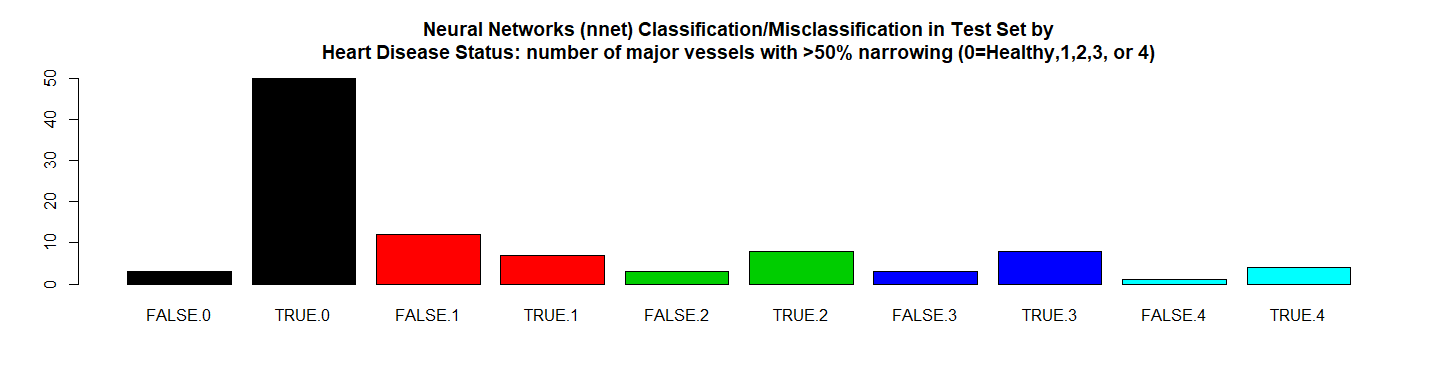
res.n\_net <- **confusionMatrix**(test\_pred\_n\_net, Y.testData)  
res.n\_net

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 50 19  
## 1 3 27  
##   
## Accuracy : 0.778   
## 95% CI : (0.683, 0.855)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 5.07e-07   
##   
## Kappa : 0.543   
## Mcnemar's Test P-Value : 0.00138   
##   
## Sensitivity : 0.943   
## Specificity : 0.587   
## Pos Pred Value : 0.725   
## Neg Pred Value : 0.900   
## Prevalence : 0.535   
## Detection Rate : 0.505   
## Detection Prevalence : 0.697   
## Balanced Accuracy : 0.765   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.n\_net,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
  
**plot\_results\_by\_type**(test\_pred\_n\_net,  
 Y.testData,  
 main, testData)



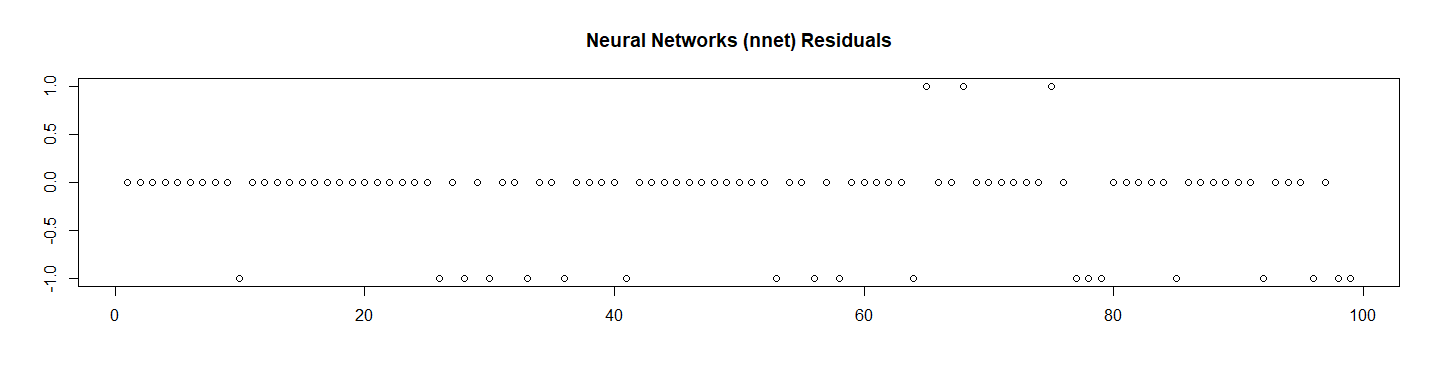
*# RMSE Resulting*  
(rmse\_n\_net <- **sqrt**(**mean**((**as.numeric**(test\_pred\_n\_net)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4714

*# Residuals Plot*  
(r\_n\_net <- (**as.numeric**(test\_pred\_n\_net)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [24] 0 0 -1 0 -1 0 -1 0 0 -1 0 0 -1 0 0 0 0 -1 0 0 0 0 0  
## [47] 0 0 0 0 0 0 -1 0 0 -1 0 -1 0 0 0 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 -1 -1 -1 0 0 0 0 0 -1 0 0 0 0 0 0 -1  
## [93] 0 0 0 -1 0 -1 -1

**plot**(r\_n\_net, main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_n\_net\_prob <- **predict**(n\_net, newdata = X.testData, type ="prob")[2]  
  
res.n\_net.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_n\_net\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(n\_net**$**results**$**Accuracy, na.rm = TRUE),  
 "Prediction Accuracy in Test Set" = res.n\_net**$**overall[1],  
 "RMSE Test"= rmse\_n\_net,  
 "ROC" = res.n\_net.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 11. K-Nearest Neighbors (knn)

############################################  
*# 11. K-Nearest Neighbors*  
model.name <- "K-Nearest Neighbors (knn)"  
model.name11 <- model.name  
file.name <- **paste**(work.dir, "knn.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "knn-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 grid\_knn <- **expand.grid**(k = **c**(1,2,3,4,5,6,7,8,9,10,11,12,13))  
 t1 <- **proc.time**()  
 knn <- **train**(y = **factor**(Y.trainData),   
 x = X.trainData,  
 method="knn",  
 preProcess = **c**("center", "scale"),  
 trControl=trctrl,  
 tuneGrid = grid\_knn)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(knn, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "K-Nearest Neighbors (knn) Executed ... time:6.9"

Let’s see the results:

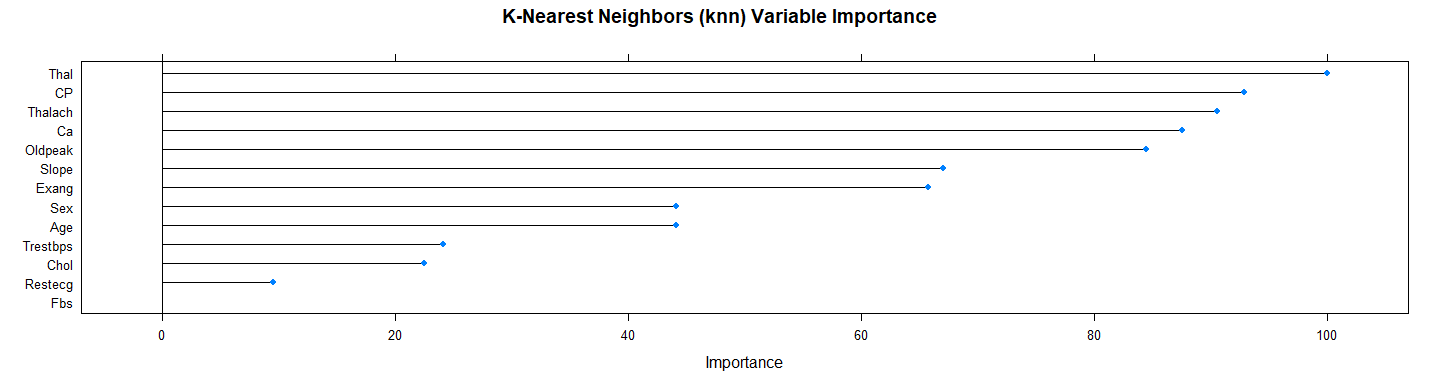
*# Let's see some details about the model*  
knn

## k-Nearest Neighbors   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 178, 179, 178, 178, 178, 178, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 1 0.7643 0.5284  
## 2 0.7860 0.5702  
## 3 0.8070 0.6115  
## 4 0.8102 0.6159  
## 5 0.8201 0.6358  
## 6 0.8118 0.6181  
## 7 0.8218 0.6377  
## 8 0.8166 0.6282  
## 9 0.8135 0.6206  
## 10 0.8133 0.6194  
## 11 0.8251 0.6442  
## 12 0.8299 0.6545  
## 13 0.8165 0.6264  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 12.

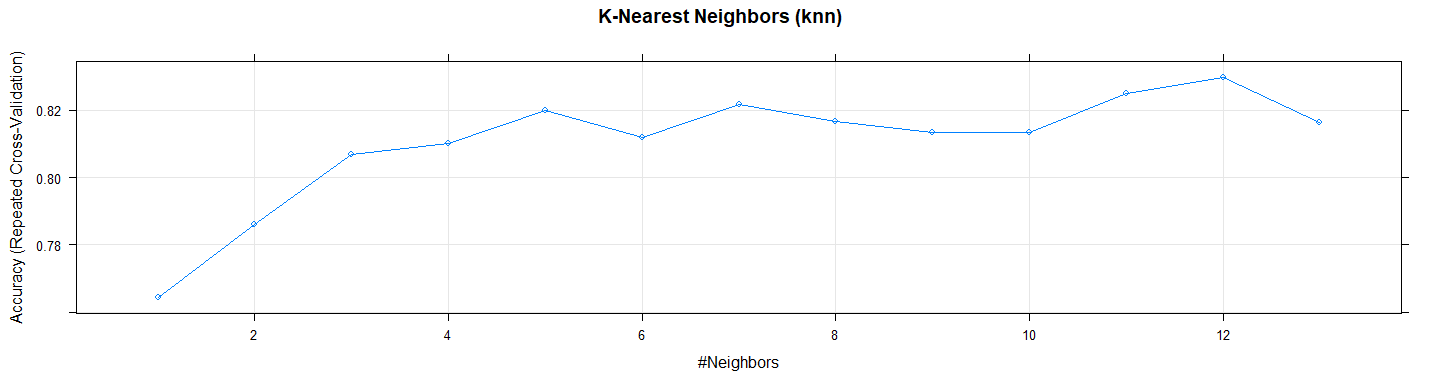
**summary**(knn)

## Length Class Mode   
## learn 2 -none- list   
## k 1 -none- numeric   
## theDots 0 -none- list   
## xNames 13 -none- character  
## problemType 1 -none- character  
## tuneValue 1 data.frame list   
## obsLevels 2 -none- character  
## param 0 -none- list

**plot**(**varImp**(knn),   
 main = **paste0**(model.name," Variable Importance"))



**plot**(knn, main = model.name)



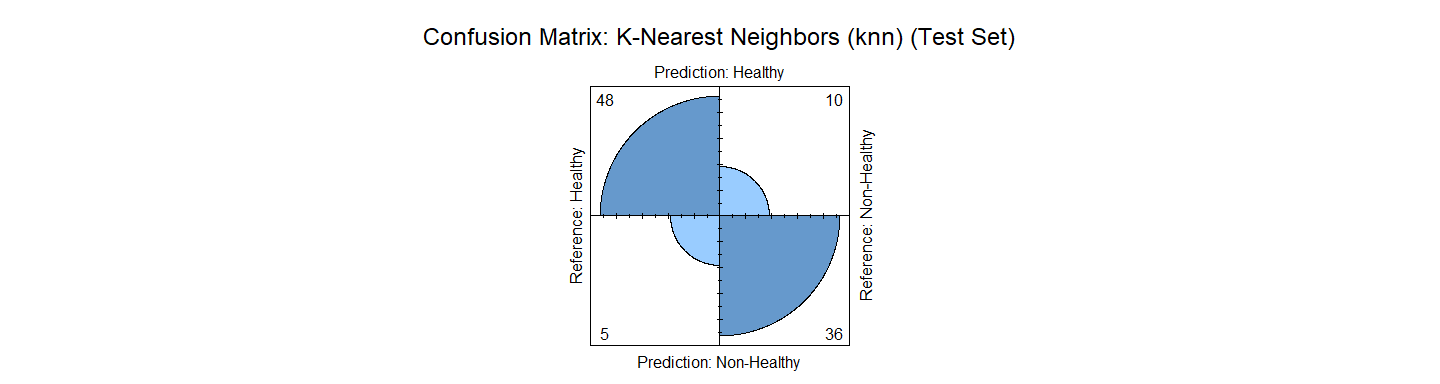
*# Let's use the model in test set*  
test\_pred\_knn <- **predict**(knn, newdata = X.testData)  
test\_pred\_knn

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1  
## [71] 1 1 0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0  
## Levels: 0 1

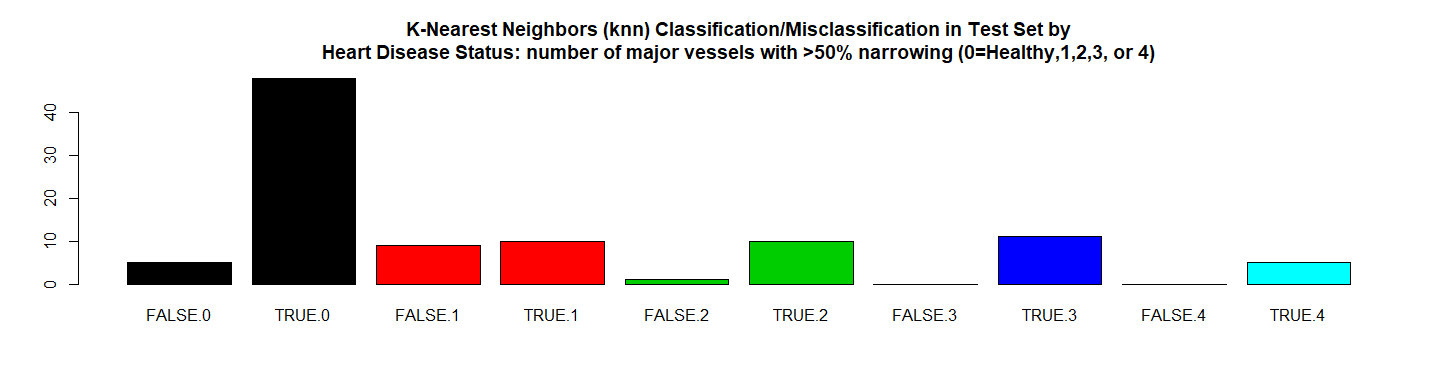
res.knn <- **confusionMatrix**(test\_pred\_knn, Y.testData)  
res.knn

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 48 10  
## 1 5 36  
##   
## Accuracy : 0.848   
## 95% CI : (0.762, 0.913)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 4.39e-11   
##   
## Kappa : 0.693   
## Mcnemar's Test P-Value : 0.302   
##   
## Sensitivity : 0.906   
## Specificity : 0.783   
## Pos Pred Value : 0.828   
## Neg Pred Value : 0.878   
## Prevalence : 0.535   
## Detection Rate : 0.485   
## Detection Prevalence : 0.586   
## Balanced Accuracy : 0.844   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.knn,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
  
**plot\_results\_by\_type**(test\_pred\_knn,  
 Y.testData,  
 main, testData)



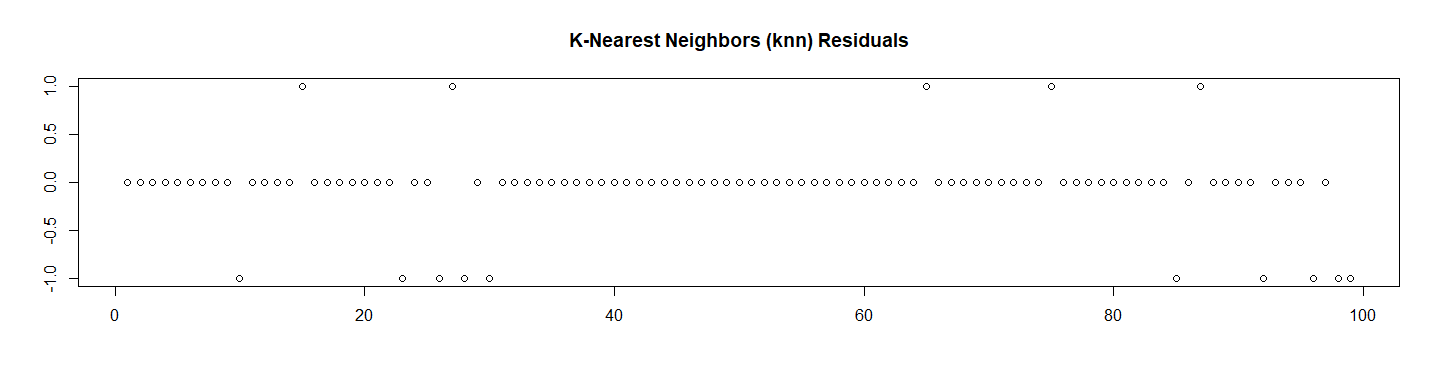
*# RMSE Resulting*  
(rmse\_knn <- **sqrt**(**mean**((**as.numeric**(test\_pred\_knn)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.3892

*# Residuals Plot*  
(r\_knn <- (**as.numeric**(test\_pred\_knn)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0 0 0 -1  
## [24] 0 0 -1 1 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [70] 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 -1 -1

**plot**(r\_knn, main = **paste0**(model.name," Residuals"),xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_knn\_prob <- **predict**(knn, newdata = X.testData, type ="prob")[2]  
  
res.knn.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_knn\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(knn**$**results**$**Accuracy, na.rm = TRUE),  
 "Prediction Accuracy in Test Set" = res.knn**$**overall[1],  
 "RMSE Test"= rmse\_knn,  
 "ROC" = res.knn.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 12. Random Forest

## 12.1 Random Forest with Tuning Parameters (rf)

############################################  
*# 12. Random Forest*  
############################################  
## 12.1 Random Forest with Tuning Parameters (rf)  
model.name <- "Random Forest with Tuning Parameters (rf)"  
model.name12 <- model.name  
file.name <- **paste**(work.dir, "ramdonf.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "randomf-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 grid\_randomf <- **expand.grid**(mtry = **c**(1,2,3,4,5,6,7,8,9,10,11,12,13))  
 t1 <- **proc.time**()  
 randomf <- **train**(y = **factor**(Y.trainData),   
 x = X.trainData,  
 method="rf",  
 preProcess = **c**("center", "scale"),  
 trControl=trctrl,   
 importance = TRUE,  
 tuneGrid = grid\_randomf)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(randomf, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Random Forest with Tuning Parameters (rf) Executed ... time:52"

Let’s see the results:

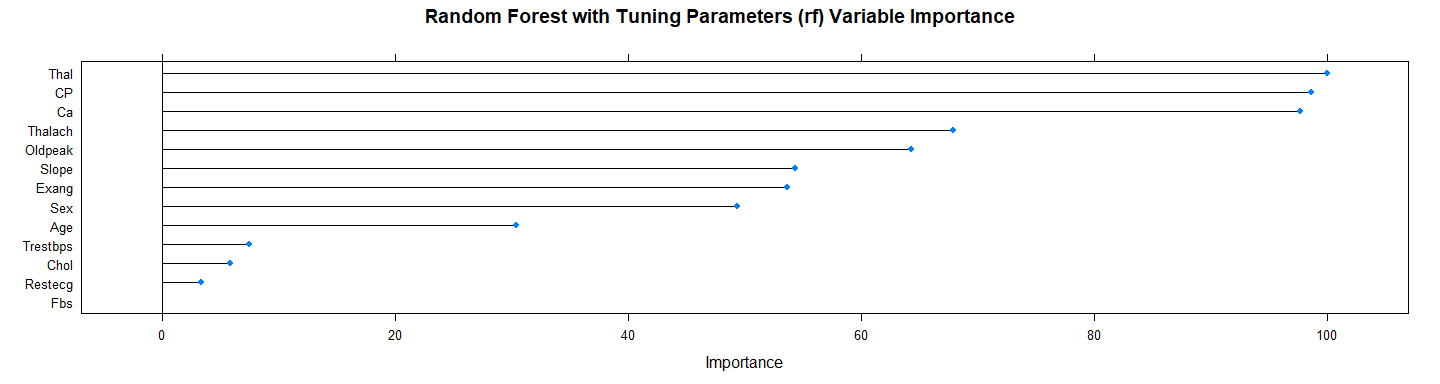
*# Let's see some details about the model*  
randomf

## Random Forest   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 178, 178, 178, 178, 179, 178, ...   
## Resampling results across tuning parameters:  
##   
## mtry Accuracy Kappa   
## 1 0.8449 0.6824  
## 2 0.8314 0.6567  
## 3 0.8298 0.6536  
## 4 0.8177 0.6299  
## 5 0.8110 0.6168  
## 6 0.8179 0.6302  
## 7 0.8126 0.6196  
## 8 0.8178 0.6302  
## 9 0.8109 0.6167  
## 10 0.8109 0.6164  
## 11 0.8059 0.6066  
## 12 0.8093 0.6129  
## 13 0.8143 0.6228  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was mtry = 1.

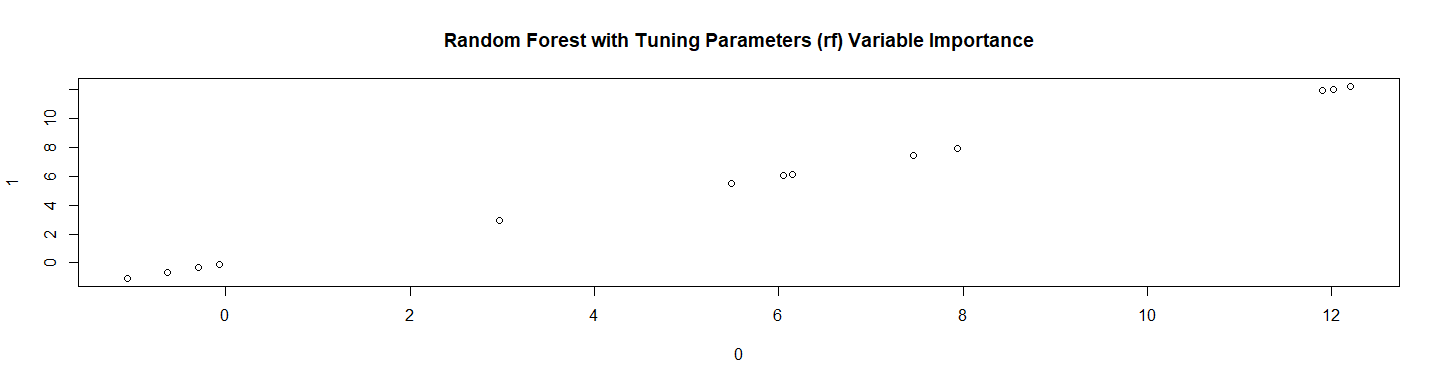
**summary**(randomf)

## Length Class Mode   
## call 5 -none- call   
## type 1 -none- character  
## predicted 198 factor numeric   
## err.rate 1500 -none- numeric   
## confusion 6 -none- numeric   
## votes 396 matrix numeric   
## oob.times 198 -none- numeric   
## classes 2 -none- character  
## importance 52 -none- numeric   
## importanceSD 39 -none- numeric   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 14 -none- list   
## y 198 factor numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## xNames 13 -none- character  
## problemType 1 -none- character  
## tuneValue 1 data.frame list   
## obsLevels 2 -none- character  
## param 1 -none- list

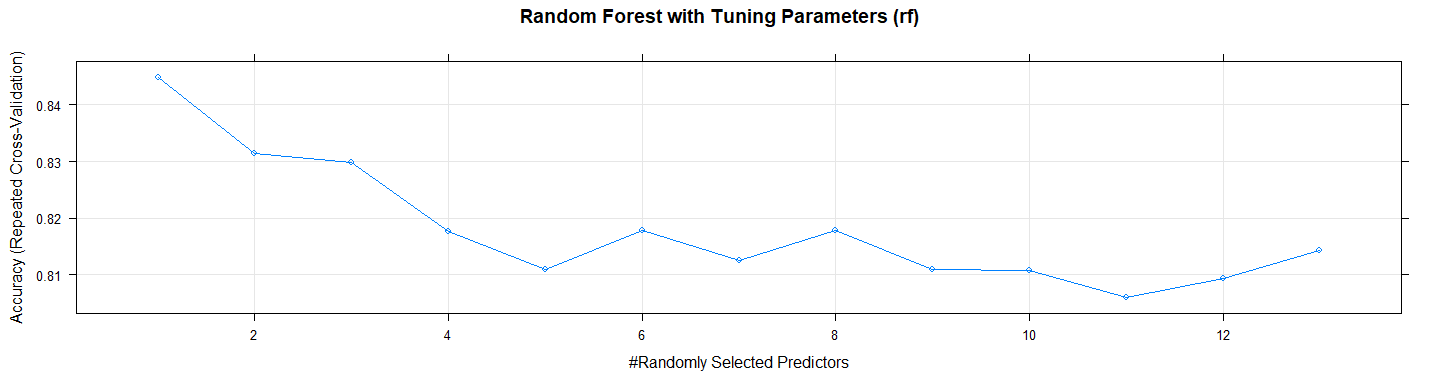
**plot**(**varImp**(randomf), main = **paste0**(model.name," Variable Importance"))



**plot**(**varImp**(randomf**$**finalModel), main = **paste0**(model.name," Variable Importance"))



**plot**(randomf, main = model.name)



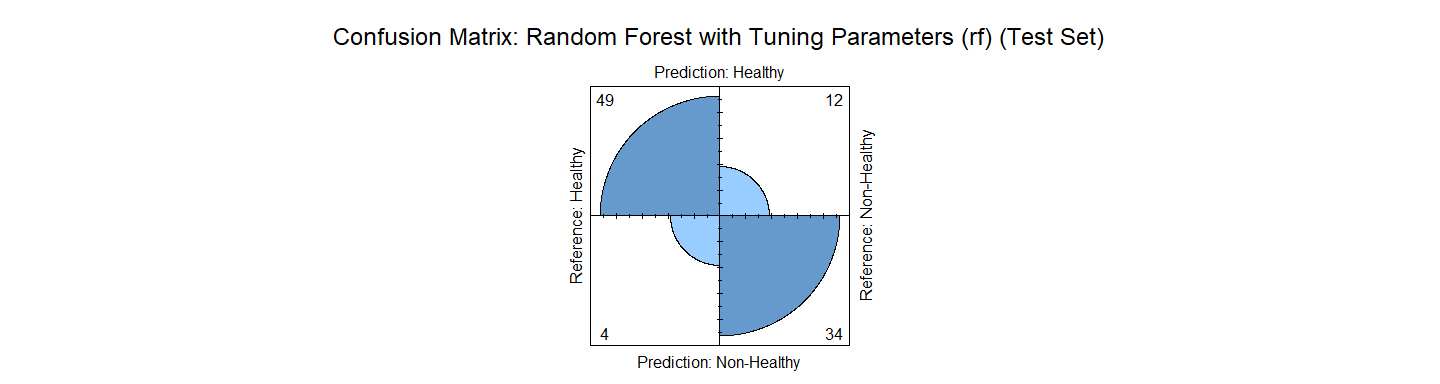
*# Let's use the model in test set*  
test\_pred\_randomf <- **predict**(randomf, newdata = X.testData)  
test\_pred\_randomf

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0  
## [36] 1 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 0 1 0 0 1 0 1 1 0 1  
## [71] 1 1 0 0 1 0 1 1 1 0 0 1 0 0 1 0 1 0 0 1 1 0 1 1 1 0 0 0 0  
## Levels: 0 1

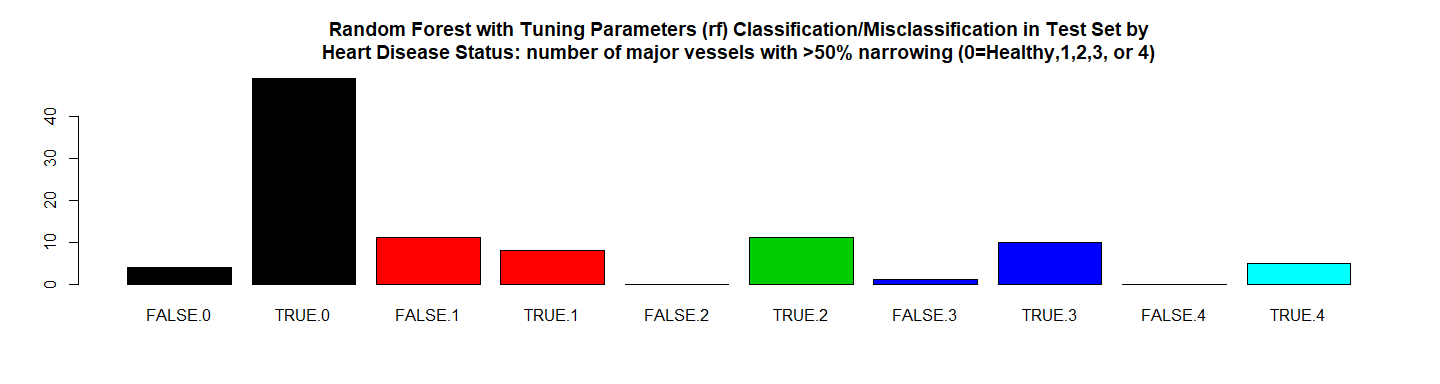
res.randomf <- **confusionMatrix**(test\_pred\_randomf, Y.testData)  
res.randomf

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 49 12  
## 1 4 34  
##   
## Accuracy : 0.838   
## 95% CI : (0.751, 0.905)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 2.04e-10   
##   
## Kappa : 0.671   
## Mcnemar's Test P-Value : 0.0801   
##   
## Sensitivity : 0.925   
## Specificity : 0.739   
## Pos Pred Value : 0.803   
## Neg Pred Value : 0.895   
## Prevalence : 0.535   
## Detection Rate : 0.495   
## Detection Prevalence : 0.616   
## Balanced Accuracy : 0.832   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.randomf,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
  
**plot\_results\_by\_type**(test\_pred\_randomf,  
 Y.testData,  
 main, testData)



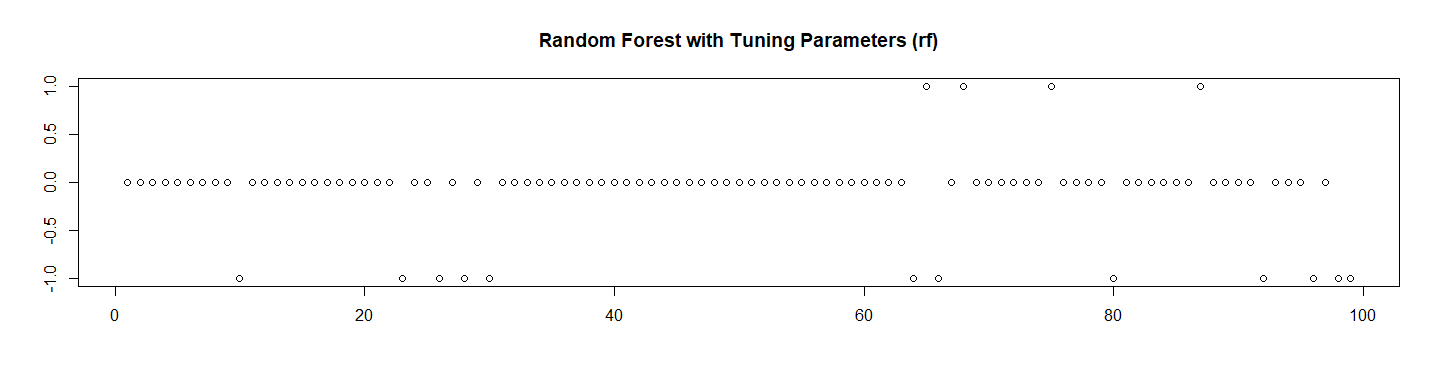
*# RMSE Resulting*  
(rmse\_randomf <- **sqrt**(**mean**((**as.numeric**(test\_pred\_randomf)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.402

*# Residuals Plot*  
(r\_randomf <- (**as.numeric**(test\_pred\_randomf)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 -1  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 1 -1 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 -1 0 0 0 0 0 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 -1 -1

**plot**(r\_randomf, main = model.name,xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_randomf\_prob <- **predict**(randomf, newdata = X.testData, type ="prob")[2]  
  
res.randomf.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_randomf\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(randomf**$**results**$**Accuracy, na.rm = TRUE),  
 "Prediction Accuracy in Test Set" = res.randomf**$**overall[1],  
 "RMSE Test"= rmse\_randomf,  
 "ROC" = res.randomf.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

## 12.2 Random Forest Boosted Tree (bstTree)

############################################  
## 12.2 Random Forest Boosted Tree (bstTree)  
model.name <- "Random Forest Boosted Tree (bstTree)"  
model.name13 <- model.name  
file.name <- **paste**(work.dir, "ramdonf2.Rds", sep="/")   
file.name.et <- **paste**(work.dir, "randomf2-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
 grid\_randomf2 <- **expand.grid**(maxdepth = **c**(1,2,3), mstop = **c**(50,100,150,200), nu = **c**(0.1,0.2,0.3))  
 randomf2 <- **train**(y = **factor**(Y.trainData),   
 x = X.trainData,  
 method="bstTree",  
 preProcess = **c**("center", "scale"),  
 tuneGrid = grid\_randomf2,  
 trControl=trctrl)   
   
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(randomf2, file = file.name)  
 **save**(et, file = file.name.et)  
}

## [1] "Random Forest Boosted Tree (bstTree) Executed ... time:425"

Let’s see the results:

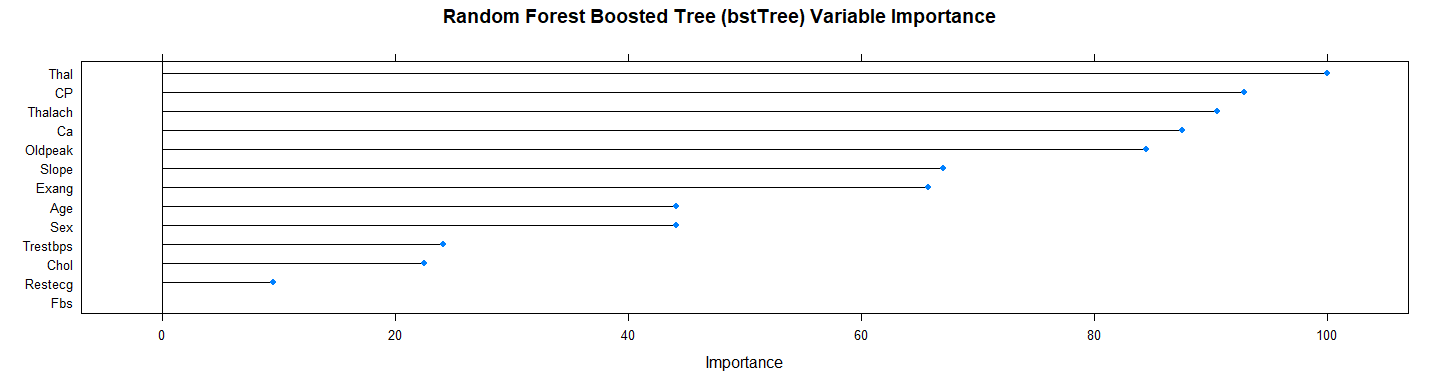
*# Let's see some details about the model*  
randomf2

## Boosted Tree   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## Pre-processing: centered (13), scaled (13)   
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 178, 179, 178, 179, 178, 178, ...   
## Resampling results across tuning parameters:  
##   
## maxdepth nu mstop Accuracy Kappa   
## 1 0.1 50 0.7339 0.4646  
## 1 0.1 100 0.7339 0.4646  
## 1 0.1 150 0.7339 0.4646  
## 1 0.1 200 0.7372 0.4709  
## 1 0.2 50 0.7339 0.4646  
## 1 0.2 100 0.7339 0.4646  
## 1 0.2 150 0.7645 0.5238  
## 1 0.2 200 0.8096 0.6148  
## 1 0.3 50 0.7355 0.4678  
## 1 0.3 100 0.7610 0.5173  
## 1 0.3 150 0.8165 0.6288  
## 1 0.3 200 0.8283 0.6530  
## 2 0.1 50 0.7976 0.5866  
## 2 0.1 100 0.8094 0.6121  
## 2 0.1 150 0.8178 0.6294  
## 2 0.1 200 0.8126 0.6194  
## 2 0.2 50 0.8128 0.6199  
## 2 0.2 100 0.8178 0.6301  
## 2 0.2 150 0.8078 0.6100  
## 2 0.2 200 0.8047 0.6036  
## 2 0.3 50 0.8062 0.6064  
## 2 0.3 100 0.8146 0.6235  
## 2 0.3 150 0.8014 0.5973  
## 2 0.3 200 0.7997 0.5943  
## 3 0.1 50 0.8230 0.6419  
## 3 0.1 100 0.8248 0.6455  
## 3 0.1 150 0.8162 0.6288  
## 3 0.1 200 0.8079 0.6121  
## 3 0.2 50 0.8247 0.6453  
## 3 0.2 100 0.8063 0.6090  
## 3 0.2 150 0.8048 0.6059  
## 3 0.2 200 0.8014 0.5995  
## 3 0.3 50 0.8282 0.6523  
## 3 0.3 100 0.8145 0.6255  
## 3 0.3 150 0.7927 0.5816  
## 3 0.3 200 0.7880 0.5727  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were mstop = 200, maxdepth = 1 and  
## nu = 0.3.

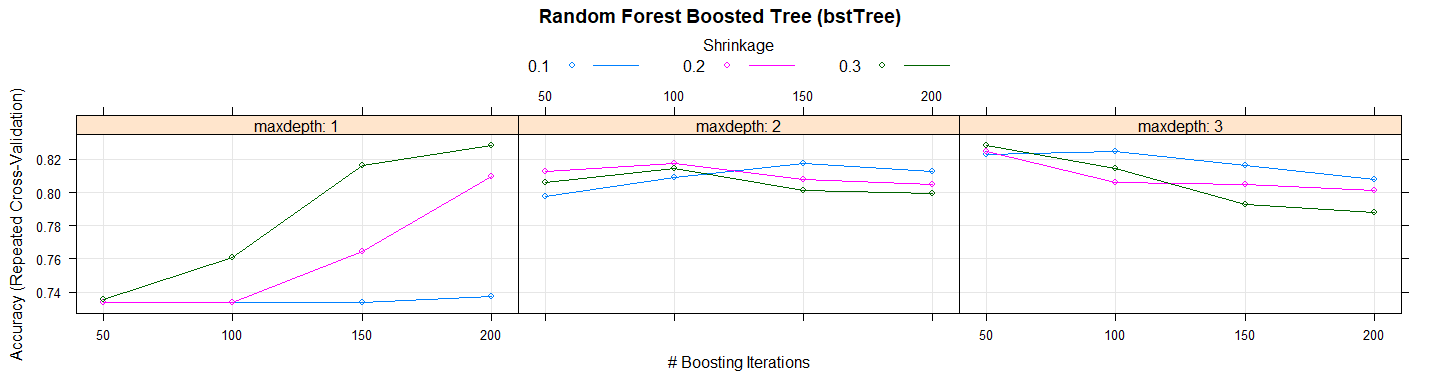
**summary**(randomf2)

## Length Class Mode   
## y 198 -none- numeric   
## x 13 data.frame list   
## cost 1 -none- numeric   
## family 1 -none- character  
## learner 1 -none- character  
## yhat 198 -none- numeric   
## offset 1 -none- numeric   
## ens 200 -none- list   
## control.tree 1 -none- list   
## risk 200 -none- numeric   
## ctrl 19 bst\_control list   
## maxdepth 1 -none- numeric   
## xselect 9 -none- numeric   
## coef 200 -none- logical   
## ensemble 200 -none- list   
## ml.fit 14 rpart list   
## meanx 198 -none- numeric   
## int 0 -none- NULL   
## call 7 -none- call   
## xNames 13 -none- character  
## problemType 1 -none- character  
## tuneValue 3 data.frame list   
## obsLevels 2 -none- character  
## param 0 -none- list

**plot**(**varImp**(randomf2), main = **paste0**(model.name," Variable Importance"))



**plot**(randomf2, main = model.name)



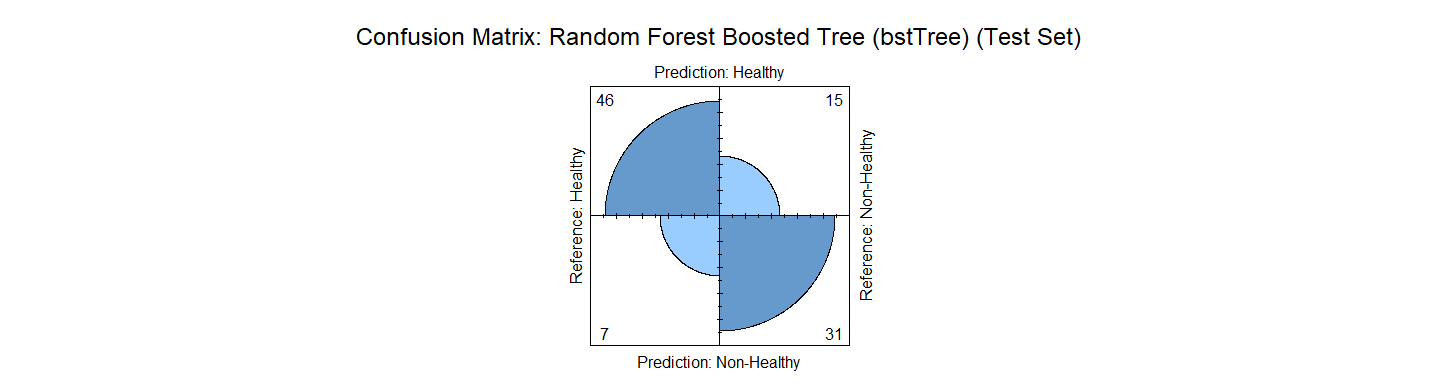
*# Let's use the model in test set*  
test\_pred\_randomf2 <- **predict**(randomf2, newdata = X.testData)  
test\_pred\_randomf2

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0  
## [36] 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0  
## Levels: 0 1

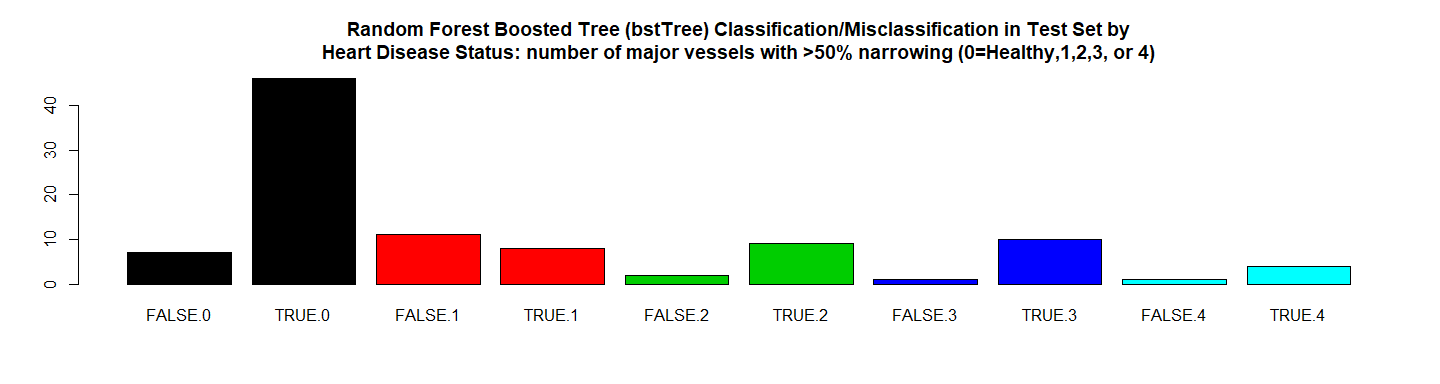
res.randomf2 <- **confusionMatrix**(test\_pred\_randomf2, Y.testData)  
res.randomf2

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 46 15  
## 1 7 31  
##   
## Accuracy : 0.778   
## 95% CI : (0.683, 0.855)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 5.07e-07   
##   
## Kappa : 0.548   
## Mcnemar's Test P-Value : 0.136   
##   
## Sensitivity : 0.868   
## Specificity : 0.674   
## Pos Pred Value : 0.754   
## Neg Pred Value : 0.816   
## Prevalence : 0.535   
## Detection Rate : 0.465   
## Detection Prevalence : 0.616   
## Balanced Accuracy : 0.771   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.randomf2,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
  
**plot\_results\_by\_type**(test\_pred\_randomf2,   
 Y.testData,  
 main, testData)



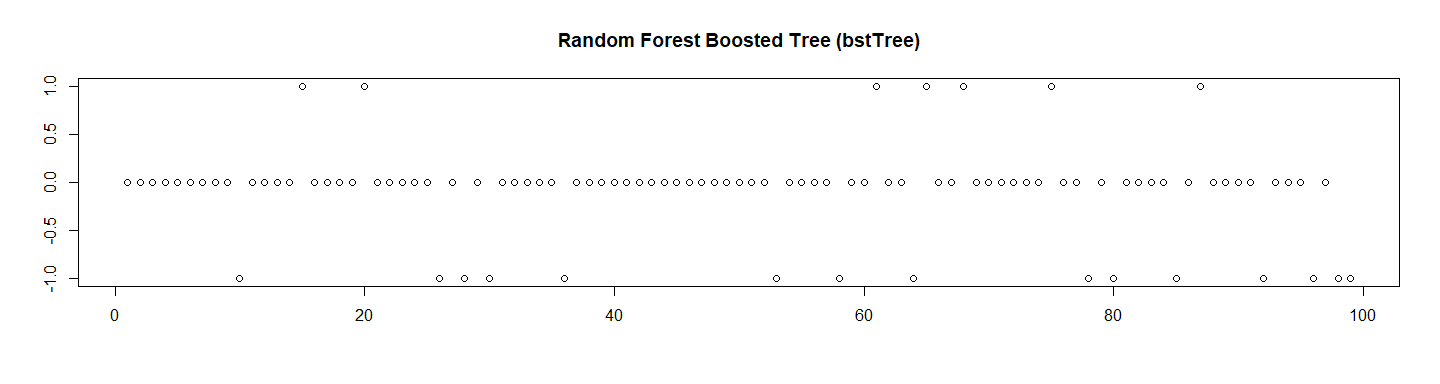
*# RMSE Resulting*  
(rmse\_randomf2 <- **sqrt**(**mean**((**as.numeric**(test\_pred\_randomf2)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4714

*# Residuals Plot*  
(r\_randomf2 <- (**as.numeric**(test\_pred\_randomf2)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 1 0 0 0  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 -1 0 0 0 0 -1 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 -1 0 -1 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 -1 -1

**plot**(r\_randomf2, main = model.name,xlab = "", ylab = "")



*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(randomf2**$**results**$**Accuracy, na.rm = TRUE),  
 "Prediction Accuracy in Test Set" = res.randomf2**$**overall[1],  
 "RMSE Test"= rmse\_randomf2,  
 "ROC" = NA,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

## 12.3 Random Forest Boost with Tuning Parameters (gbm)

############################################  
## 12.3 Random Forest Boost with Tuning Parameters (gbm)  
model.name <- "Random Forest Boost with Tuning Parameters (gbm)"  
model.name14 <- model.name  
file.name <- **paste**(work.dir, "ramdonf3.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "randomf3-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
 objControl <- **trainControl**(method='cv', number=10, repeats = 10)  
 grid\_randomf3 <- **expand.grid**(interaction.depth = **c**(1, 5, 9),  
 n.trees = (1**:**30)**\***50,  
 shrinkage = 0.1,  
 n.minobsinnode =10)  
   
 t1 <- **proc.time**()  
 randomf3 <- **train**(y = **factor**(Y.trainData),   
 x = X.trainData,  
 method="gbm",  
 trControl=trctrl,   
 tuneGrid = grid\_randomf3,  
 verbose=FALSE)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name, " Executed ... time:", et))  
 **save**(randomf3, file = file.name)  
 **save**(et, file = file.name.et)  
}

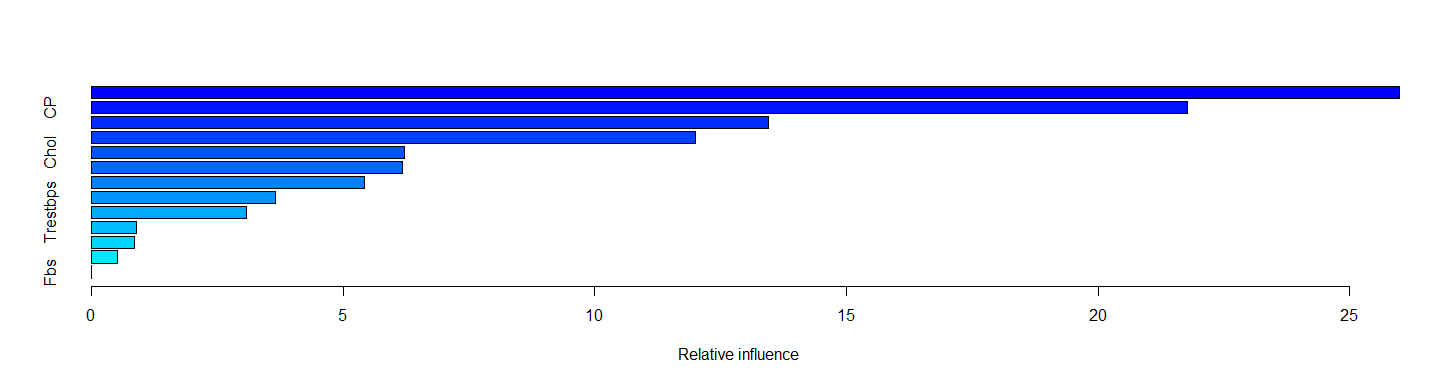
## [1] "Random Forest Boost with Tuning Parameters (gbm) Executed ... time:29"

Let’s see the results:

*# Let's see some details about the model*  
randomf3

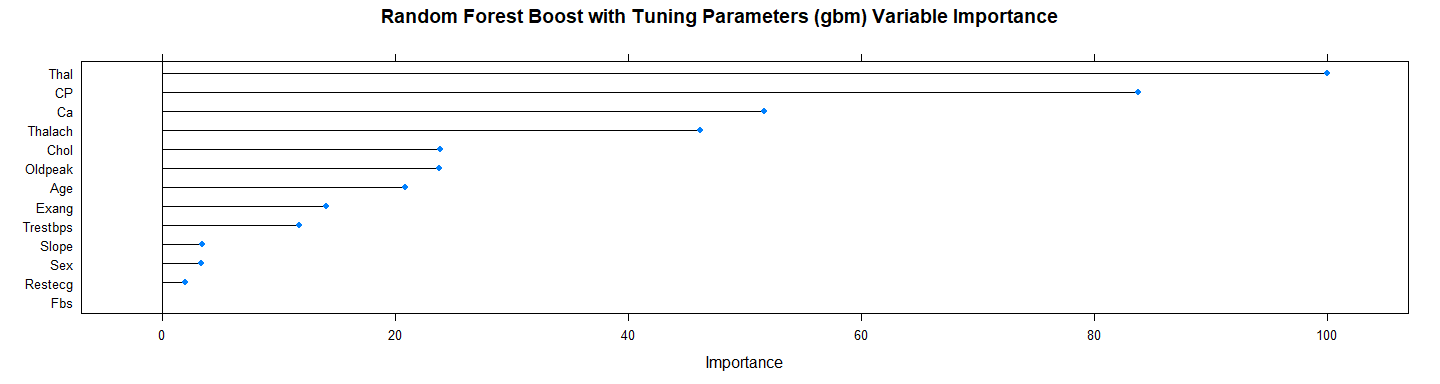
## Stochastic Gradient Boosting   
##   
## 198 samples  
## 13 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 178, 179, 178, 179, 177, 178, ...   
## Resampling results across tuning parameters:  
##   
## interaction.depth n.trees Accuracy Kappa   
## 1 50 0.8386 0.6739  
## 1 100 0.8469 0.6908  
## 1 150 0.8282 0.6531  
## 1 200 0.8114 0.6206  
## 1 250 0.8011 0.5998  
## 1 300 0.8011 0.6000  
## 1 350 0.8077 0.6128  
## 1 400 0.8045 0.6072  
## 1 450 0.8043 0.6069  
## 1 500 0.7977 0.5931  
## 1 550 0.7995 0.5970  
## 1 600 0.7879 0.5750  
## 1 650 0.7912 0.5808  
## 1 700 0.7912 0.5807  
## 1 750 0.7811 0.5603  
## 1 800 0.7794 0.5572  
## 1 850 0.7828 0.5641  
## 1 900 0.7760 0.5506  
## 1 950 0.7745 0.5473  
## 1 1000 0.7675 0.5336  
## 1 1050 0.7710 0.5400  
## 1 1100 0.7710 0.5398  
## 1 1150 0.7691 0.5360  
## 1 1200 0.7675 0.5325  
## 1 1250 0.7759 0.5493  
## 1 1300 0.7709 0.5397  
## 1 1350 0.7710 0.5398  
## 1 1400 0.7742 0.5462  
## 1 1450 0.7741 0.5463  
## 1 1500 0.7692 0.5364  
## 5 50 0.8197 0.6372  
## 5 100 0.7945 0.5860  
## 5 150 0.7778 0.5531  
## 5 200 0.7758 0.5490  
## 5 250 0.7842 0.5656  
## 5 300 0.7842 0.5651  
## 5 350 0.7743 0.5456  
## 5 400 0.7726 0.5424  
## 5 450 0.7760 0.5491  
## 5 500 0.7827 0.5627  
## 5 550 0.7810 0.5592  
## 5 600 0.7695 0.5362  
## 5 650 0.7693 0.5362  
## 5 700 0.7744 0.5454  
## 5 750 0.7676 0.5325  
## 5 800 0.7675 0.5330  
## 5 850 0.7727 0.5435  
## 5 900 0.7676 0.5327  
## 5 950 0.7643 0.5260  
## 5 1000 0.7693 0.5360  
## 5 1050 0.7676 0.5326  
## 5 1100 0.7709 0.5393  
## 5 1150 0.7709 0.5392  
## 5 1200 0.7742 0.5456  
## 5 1250 0.7743 0.5466  
## 5 1300 0.7793 0.5571  
## 5 1350 0.7776 0.5531  
## 5 1400 0.7760 0.5497  
## 5 1450 0.7742 0.5458  
## 5 1500 0.7761 0.5495  
## 9 50 0.8095 0.6159  
## 9 100 0.7943 0.5853  
## 9 150 0.7944 0.5856  
## 9 200 0.7959 0.5888  
## 9 250 0.7892 0.5750  
## 9 300 0.7891 0.5752  
## 9 350 0.7839 0.5646  
## 9 400 0.7872 0.5711  
## 9 450 0.7922 0.5818  
## 9 500 0.7856 0.5688  
## 9 550 0.7840 0.5649  
## 9 600 0.7824 0.5622  
## 9 650 0.7891 0.5761  
## 9 700 0.7841 0.5656  
## 9 750 0.7857 0.5690  
## 9 800 0.7891 0.5762  
## 9 850 0.7927 0.5829  
## 9 900 0.7942 0.5862  
## 9 950 0.7909 0.5795  
## 9 1000 0.7959 0.5897  
## 9 1050 0.7892 0.5760  
## 9 1100 0.7909 0.5794  
## 9 1150 0.7927 0.5830  
## 9 1200 0.7927 0.5830  
## 9 1250 0.7960 0.5897  
## 9 1300 0.7961 0.5899  
## 9 1350 0.7945 0.5865  
## 9 1400 0.7977 0.5933  
## 9 1450 0.7977 0.5933  
## 9 1500 0.7944 0.5862  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were n.trees = 100,  
## interaction.depth = 1, shrinkage = 0.1 and n.minobsinnode = 10.

**summary**(randomf3)

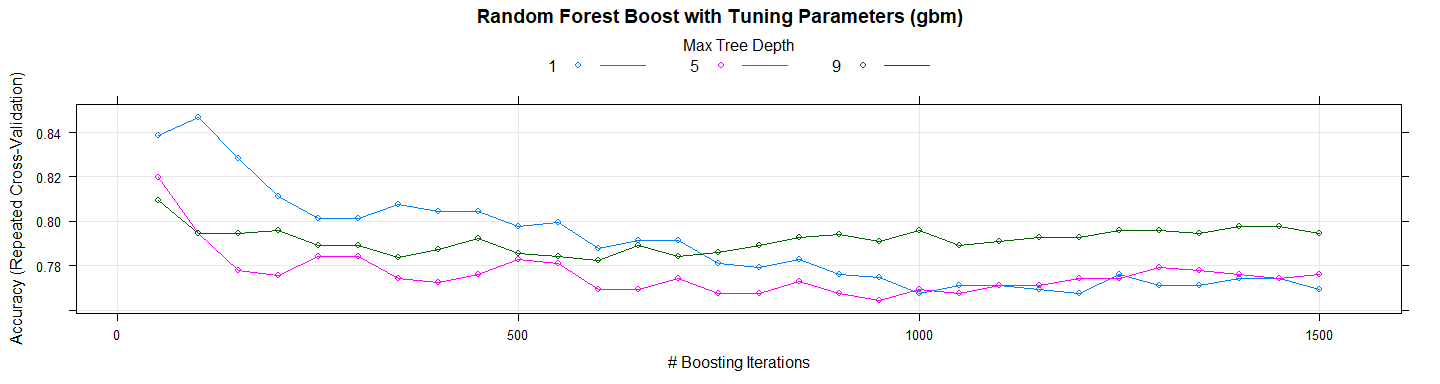


## var rel.inf  
## Thal Thal 25.9931  
## CP CP 21.7730  
## Ca Ca 13.4401  
## Thalach Thalach 12.0016  
## Chol Chol 6.2083  
## Oldpeak Oldpeak 6.1773  
## Age Age 5.4220  
## Exang Exang 3.6567  
## Trestbps Trestbps 3.0648  
## Slope Slope 0.8892  
## Sex Sex 0.8579  
## Restecg Restecg 0.5160  
## Fbs Fbs 0.0000

**plot**(**varImp**(randomf3),   
 main = **paste0**(model.name," Variable Importance"))



**plot**(randomf3, main = model.name)



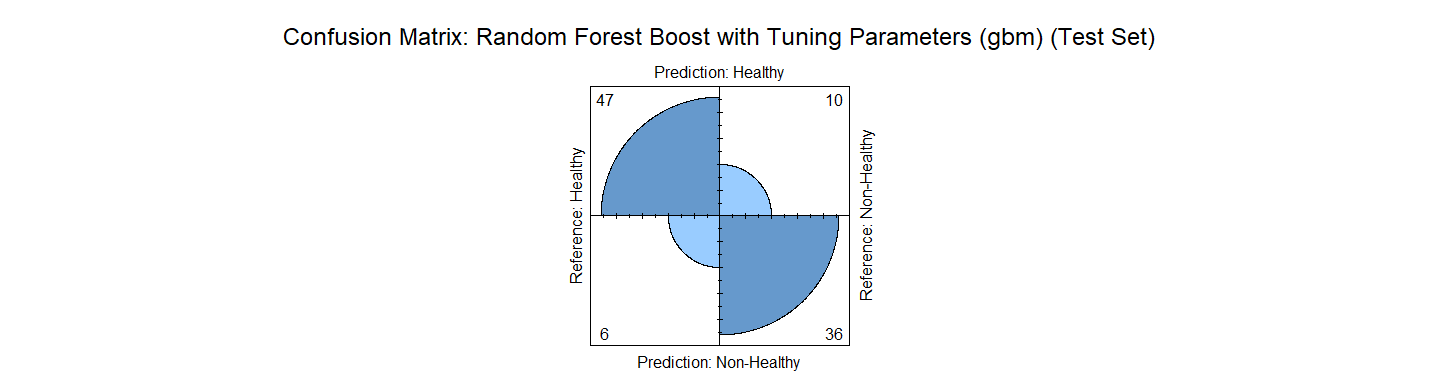
*# Let's use the model in test set*  
test\_pred\_randomf3 <- **predict**(randomf3, newdata = X.testData)  
test\_pred\_randomf3

## [1] 1 0 0 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1 0 0  
## [36] 0 0 1 0 1 1 0 0 1 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 1 1 0 1 1 1 1 1 0 1  
## [71] 1 1 0 0 1 0 1 0 1 0 0 1 0 0 1 0 1 0 0 1 1 0 1 1 1 0 0 1 0  
## Levels: 0 1

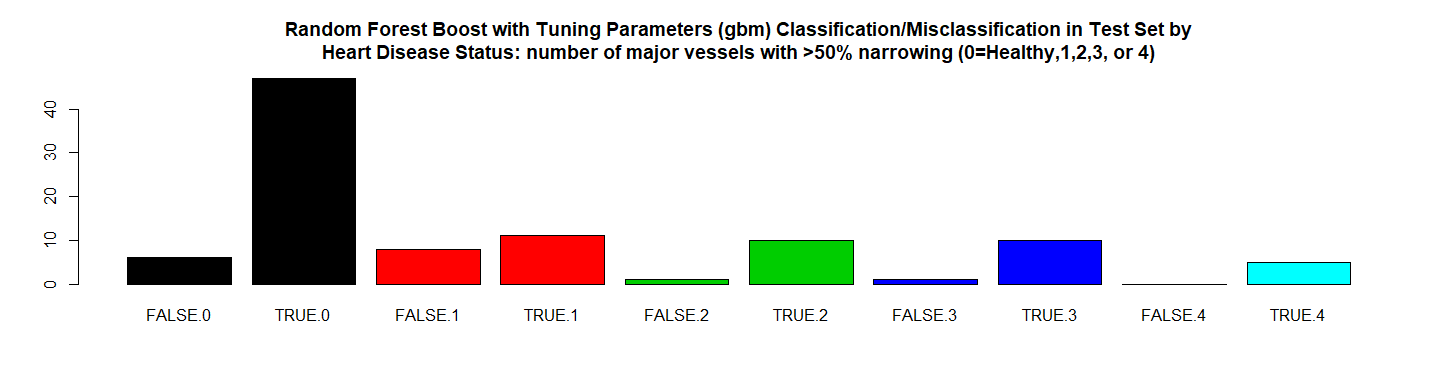
res.randomf3 <- **confusionMatrix**(test\_pred\_randomf3, Y.testData)  
res.randomf3

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 47 10  
## 1 6 36  
##   
## Accuracy : 0.838   
## 95% CI : (0.751, 0.905)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 2.04e-10   
##   
## Kappa : 0.673   
## Mcnemar's Test P-Value : 0.453   
##   
## Sensitivity : 0.887   
## Specificity : 0.783   
## Pos Pred Value : 0.825   
## Neg Pred Value : 0.857   
## Prevalence : 0.535   
## Detection Rate : 0.475   
## Detection Prevalence : 0.576   
## Balanced Accuracy : 0.835   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.randomf3,   
 main = **paste0**("Confusion Matrix: ",model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
  
**plot\_results\_by\_type**(test\_pred\_randomf3,  
 Y.testData,  
 main, testData)



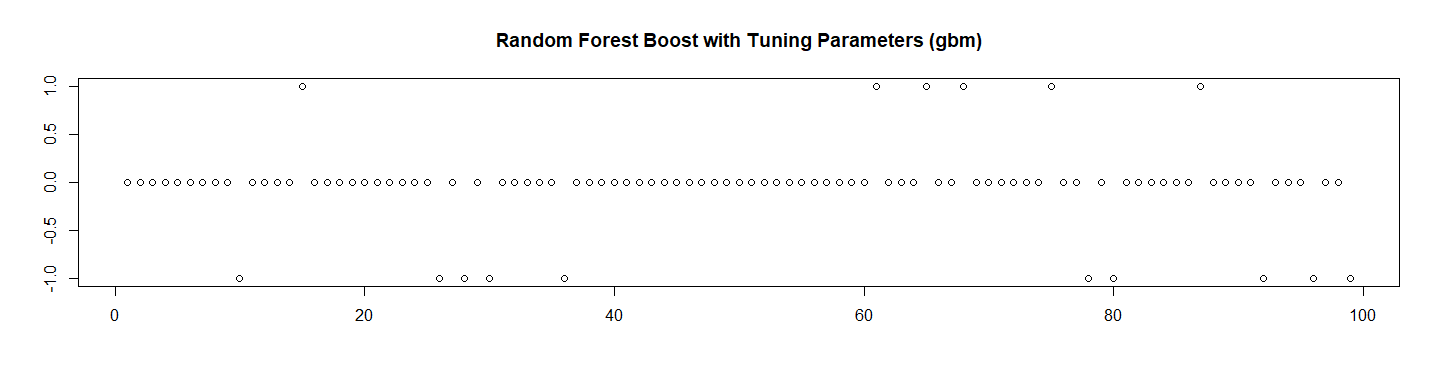
*# RMSE Resulting*  
(rmse\_randomf3 <- **sqrt**(**mean**((**as.numeric**(test\_pred\_randomf3)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.402

*# Residuals Plot*  
(r\_randomf3 <- (**as.numeric**(test\_pred\_randomf3)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 0 0 0 0  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 -1 0 -1 0 0 0 0 0 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 0 -1

**plot**(r\_randomf3, main = model.name,xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_randomf3\_prob <- **predict**(randomf3, newdata = X.testData, type ="prob")[2]  
  
res.randomf3.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_randomf3\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(randomf3**$**results**$**Accuracy, na.rm = TRUE),  
 "Prediction Accuracy in Test Set" = res.randomf3**$**overall[1],  
 "RMSE Test"= rmse\_randomf3,  
 "ROC" = res.randomf3.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

## 12.4 Random Forest with Stochastic Gradient Boost (gbm)

############################################  
## 12.4 Random Forest with Stochastic Gradient Boost (gbm)  
model.name <- "Random Forest with Stochastic Gradient Boost (gbm)"  
model.name15 <- model.name  
file.name <- **paste**(work.dir, "ramdonf4.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "randomf4-et.Rds", sep="/")  
Y.trainData.names <- **factor**(**make.names**(Y.trainData))  
Y.testData.names <- **factor**(**make.names**(Y.testData))  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
 grid\_randomf4 <- **expand.grid**(interaction.depth = **c**(1, 5, 9),  
 n.trees = (1**:**30)**\***50,  
 shrinkage = 0.1,  
 n.minobsinnode =10)  
 trctrl2 <- **trainControl**(method = "repeatedcv",  
 number = 10,  
 repeats = 10,  
 ## Estimate class probabilities  
 classProbs = TRUE,  
 ## Evaluate performance using  
 ## the following function  
 summaryFunction = twoClassSummary)  
   
 t1 <- **proc.time**()  
 randomf4 <- **train**(y = **factor**(Y.trainData.names),   
 x = X.trainData,  
 method="gbm",  
 trControl=trctrl2,   
 tuneGrid = grid\_randomf4,  
 verbose=FALSE,  
 metric = "ROC")  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(randomf4, file = file.name)  
 **save**(et, file = file.name.et)  
}

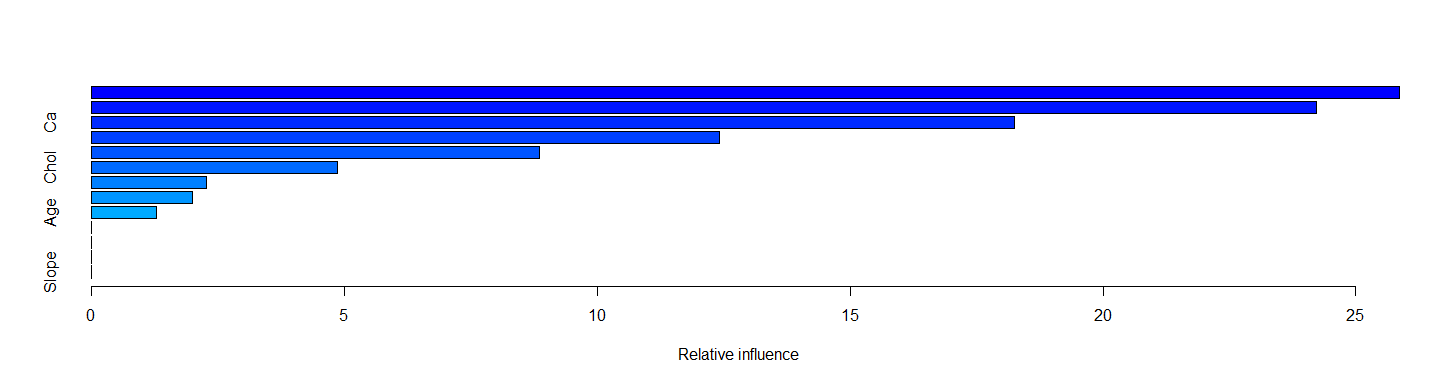
## [1] "Random Forest with Stochastic Gradient Boost (gbm) Executed ... time:96"

Let’s see the results:

*# Let's see some details about the model*  
randomf4

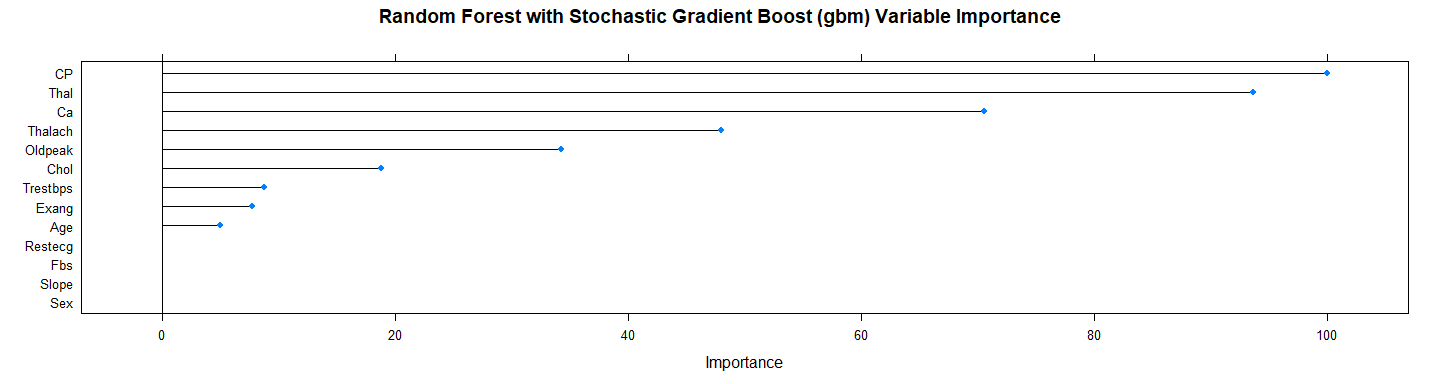
## Stochastic Gradient Boosting   
##   
## 198 samples  
## 13 predictor  
## 2 classes: 'X0', 'X1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 10 times)   
## Summary of sample sizes: 178, 178, 177, 178, 179, 178, ...   
## Resampling results across tuning parameters:  
##   
## interaction.depth n.trees ROC Sens Spec   
## 1 50 0.9053 0.8954 0.7850  
## 1 100 0.9032 0.8891 0.7886  
## 1 150 0.9032 0.8697 0.7930  
## 1 200 0.8978 0.8614 0.7864  
## 1 250 0.8941 0.8462 0.7920  
## 1 300 0.8894 0.8366 0.7919  
## 1 350 0.8873 0.8293 0.7890  
## 1 400 0.8804 0.8273 0.7811  
## 1 450 0.8764 0.8170 0.7778  
## 1 500 0.8739 0.8132 0.7767  
## 1 550 0.8736 0.8107 0.7733  
## 1 600 0.8688 0.8079 0.7712  
## 1 650 0.8653 0.8095 0.7681  
## 1 700 0.8646 0.8134 0.7659  
## 1 750 0.8611 0.8050 0.7616  
## 1 800 0.8582 0.8051 0.7604  
## 1 850 0.8561 0.8096 0.7614  
## 1 900 0.8538 0.8024 0.7498  
## 1 950 0.8529 0.8051 0.7494  
## 1 1000 0.8514 0.8041 0.7541  
## 1 1050 0.8516 0.8041 0.7452  
## 1 1100 0.8501 0.7976 0.7441  
## 1 1150 0.8497 0.8022 0.7508  
## 1 1200 0.8485 0.8005 0.7463  
## 1 1250 0.8454 0.7995 0.7397  
## 1 1300 0.8454 0.7985 0.7440  
## 1 1350 0.8441 0.8004 0.7374  
## 1 1400 0.8412 0.7968 0.7398  
## 1 1450 0.8409 0.7966 0.7386  
## 1 1500 0.8388 0.7938 0.7410  
## 5 50 0.8897 0.8551 0.7643  
## 5 100 0.8825 0.8281 0.7621  
## 5 150 0.8763 0.8311 0.7611  
## 5 200 0.8723 0.8225 0.7536  
## 5 250 0.8683 0.8185 0.7469  
## 5 300 0.8671 0.8133 0.7459  
## 5 350 0.8649 0.8197 0.7417  
## 5 400 0.8634 0.8123 0.7450  
## 5 450 0.8622 0.8122 0.7417  
## 5 500 0.8609 0.8104 0.7417  
## 5 550 0.8602 0.8059 0.7437  
## 5 600 0.8606 0.8050 0.7427  
## 5 650 0.8590 0.8049 0.7394  
## 5 700 0.8583 0.8021 0.7472  
## 5 750 0.8572 0.8069 0.7461  
## 5 800 0.8560 0.8050 0.7397  
## 5 850 0.8558 0.8052 0.7427  
## 5 900 0.8552 0.8051 0.7439  
## 5 950 0.8556 0.8114 0.7426  
## 5 1000 0.8553 0.8067 0.7396  
## 5 1050 0.8566 0.8078 0.7406  
## 5 1100 0.8563 0.8067 0.7416  
## 5 1150 0.8552 0.8028 0.7427  
## 5 1200 0.8544 0.8029 0.7404  
## 5 1250 0.8539 0.8076 0.7372  
## 5 1300 0.8542 0.8085 0.7372  
## 5 1350 0.8554 0.8029 0.7404  
## 5 1400 0.8530 0.8029 0.7406  
## 5 1450 0.8528 0.8067 0.7426  
## 5 1500 0.8516 0.8020 0.7437  
## 9 50 0.8906 0.8618 0.7676  
## 9 100 0.8792 0.8336 0.7511  
## 9 150 0.8710 0.8262 0.7511  
## 9 200 0.8662 0.8234 0.7447  
## 9 250 0.8656 0.8274 0.7402  
## 9 300 0.8632 0.8198 0.7436  
## 9 350 0.8638 0.8144 0.7401  
## 9 400 0.8620 0.8227 0.7413  
## 9 450 0.8628 0.8144 0.7447  
## 9 500 0.8620 0.8108 0.7446  
## 9 550 0.8616 0.8116 0.7478  
## 9 600 0.8613 0.8125 0.7424  
## 9 650 0.8596 0.8116 0.7402  
## 9 700 0.8604 0.8096 0.7392  
## 9 750 0.8613 0.8050 0.7404  
## 9 800 0.8606 0.8069 0.7393  
## 9 850 0.8606 0.8095 0.7414  
## 9 900 0.8611 0.8088 0.7437  
## 9 950 0.8607 0.8080 0.7403  
## 9 1000 0.8606 0.8134 0.7470  
## 9 1050 0.8602 0.8070 0.7459  
## 9 1100 0.8588 0.8100 0.7437  
## 9 1150 0.8593 0.8051 0.7417  
## 9 1200 0.8590 0.8144 0.7394  
## 9 1250 0.8572 0.8095 0.7417  
## 9 1300 0.8582 0.8135 0.7407  
## 9 1350 0.8588 0.8162 0.7449  
## 9 1400 0.8577 0.8171 0.7503  
## 9 1450 0.8584 0.8189 0.7504  
## 9 1500 0.8585 0.8162 0.7481  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## ROC was used to select the optimal model using the largest value.  
## The final values used for the model were n.trees = 50, interaction.depth  
## = 1, shrinkage = 0.1 and n.minobsinnode = 10.

**summary**(randomf4)

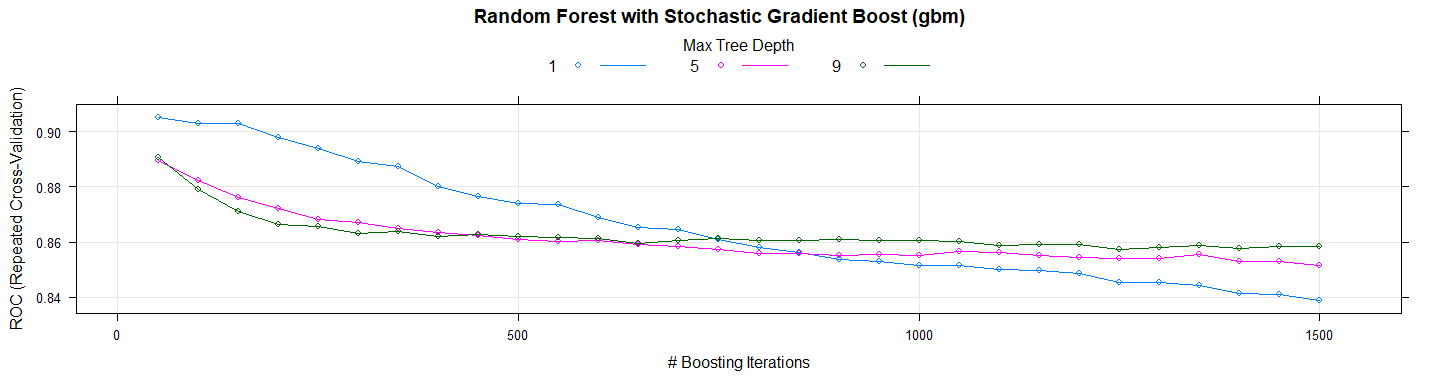


## var rel.inf  
## CP CP 25.864  
## Thal Thal 24.216  
## Ca Ca 18.252  
## Thalach Thalach 12.409  
## Oldpeak Oldpeak 8.851  
## Chol Chol 4.859  
## Trestbps Trestbps 2.271  
## Exang Exang 2.000  
## Age Age 1.279  
## Sex Sex 0.000  
## Fbs Fbs 0.000  
## Restecg Restecg 0.000  
## Slope Slope 0.000

**plot**(**varImp**(randomf4), main = **paste0**(model.name," Variable Importance"))



**plot**(randomf4, main = model.name)



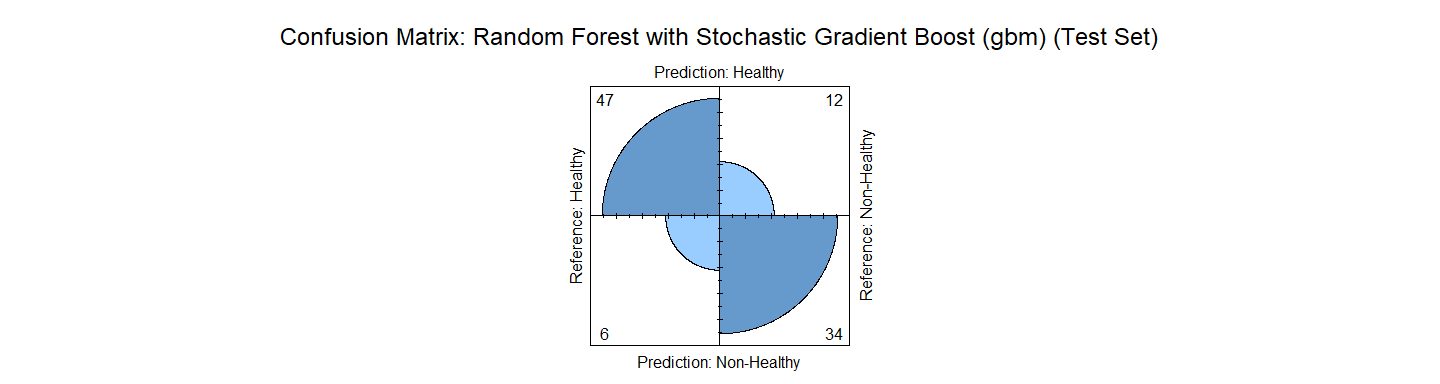
*# Let's use the model in test set*  
test\_pred\_randomf4 <- **predict**(randomf4, newdata = X.testData)  
test\_pred\_randomf4

## [1] X1 X0 X0 X0 X1 X0 X0 X0 X0 X0 X1 X1 X0 X0 X1 X0 X0 X1 X0 X1 X1 X1 X0  
## [24] X0 X1 X0 X0 X0 X0 X0 X0 X0 X1 X0 X0 X1 X0 X1 X0 X1 X1 X0 X0 X1 X0 X0  
## [47] X0 X0 X1 X0 X0 X0 X1 X1 X0 X1 X0 X1 X0 X0 X0 X1 X0 X0 X1 X1 X1 X1 X0  
## [70] X1 X1 X1 X0 X0 X1 X0 X1 X1 X1 X0 X0 X1 X0 X0 X0 X0 X1 X0 X0 X1 X1 X0  
## [93] X1 X1 X1 X0 X0 X0 X0  
## Levels: X0 X1

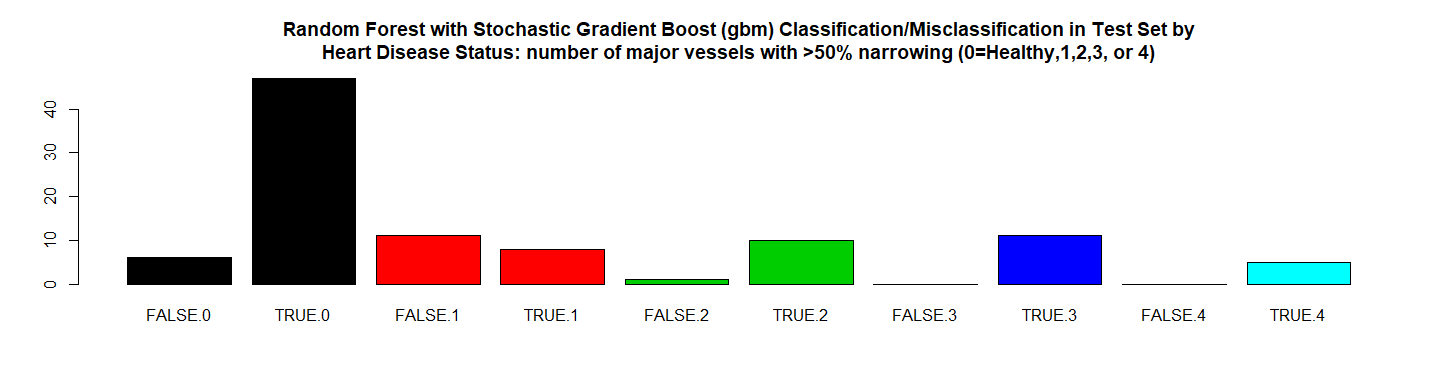
res.randomf4 <- **confusionMatrix**(test\_pred\_randomf4, Y.testData.names)  
res.randomf4

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction X0 X1  
## X0 47 12  
## X1 6 34  
##   
## Accuracy : 0.818   
## 95% CI : (0.728, 0.889)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 3.57e-09   
##   
## Kappa : 0.631   
## Mcnemar's Test P-Value : 0.239   
##   
## Sensitivity : 0.887   
## Specificity : 0.739   
## Pos Pred Value : 0.797   
## Neg Pred Value : 0.850   
## Prevalence : 0.535   
## Detection Rate : 0.475   
## Detection Prevalence : 0.596   
## Balanced Accuracy : 0.813   
##   
## 'Positive' Class : X0   
##

**print\_confusionm**(res.randomf4,   
 main = **paste0**("Confusion Matrix: ", model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_randomf4,  
 **factor**(Y.testData.names),  
 main, testData)



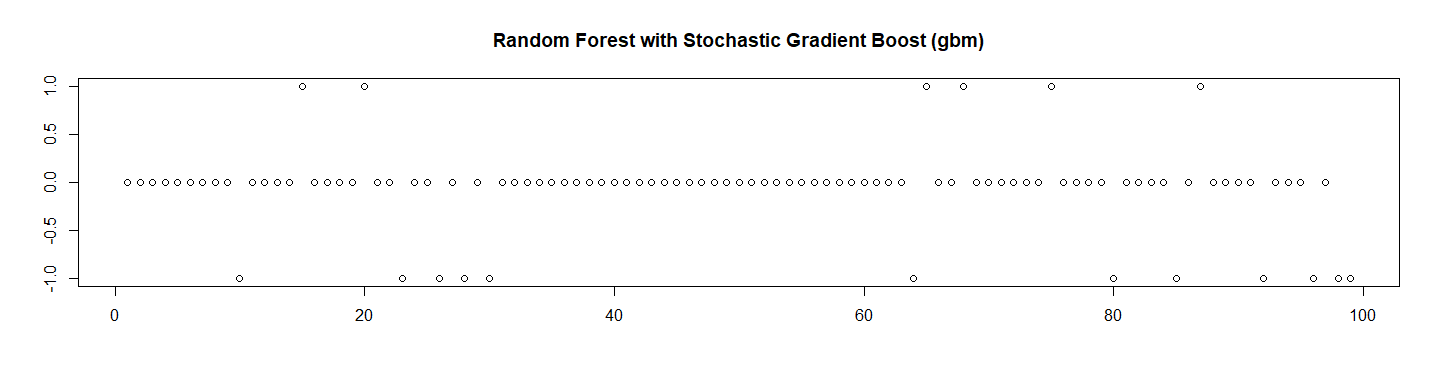
*# RMSE Resulting*  
(rmse\_randomf4 <- **sqrt**(**mean**((**as.numeric**(test\_pred\_randomf4)**-as.numeric**(Y.testData.names))**^**2)))

## [1] 0.4264

*# Residuals Plot*  
(r\_randomf4 <- (**as.numeric**(test\_pred\_randomf4)**-as.numeric**(Y.testData.names)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 1 0 0 0 0 1 0 0 -1  
## [24] 0 0 -1 0 -1 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 0 -1 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 -1 -1

**plot**(r\_randomf4, main = model.name,xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_randomf4\_prob <- **predict**(randomf4, newdata = X.testData, type ="prob")[2]  
  
res.randomf4.roc <- **roc**(**as.numeric**(Y.testData.names),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_randomf4\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(randomf4**$**results**$**ROC, na.rm = TRUE),  
 "Prediction Accuracy in Test Set" = res.randomf4**$**overall[1],  
 "RMSE Test"= rmse\_randomf4,  
 "ROC" = res.randomf4.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 13. Classification Tree (rpart)

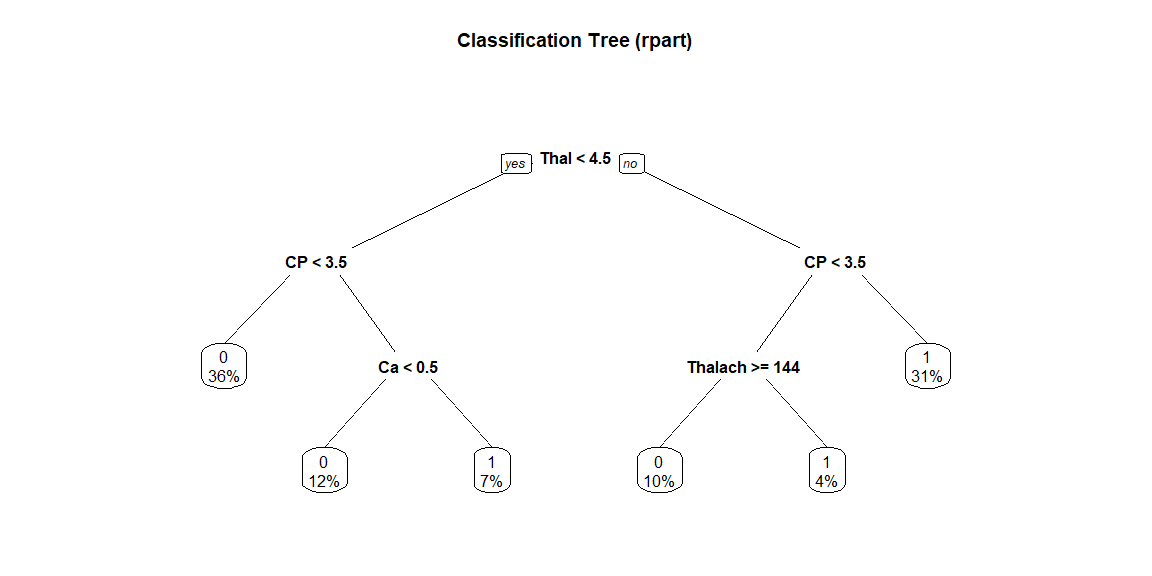
############################################  
*# 13. Classification Tree (rpart)*  
model.name <- "Classification Tree (rpart)"  
model.name16 <- model.name  
heart.tree <- **rpart**(**factor**(Disease) **~** Age **+** Sex **+** CP **+** Trestbps **+** Chol **+** Fbs **+** Restecg **+**   
 Thalach **+** Exang **+** Oldpeak **+** Slope **+** Ca **+** Thal, method = "class",   
 data = trainData)  
  
heart.tree

## n= 198   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 198 91 0 (0.54040 0.45960)   
## 2) Thal< 4.5 108 22 0 (0.79630 0.20370)   
## 4) CP< 3.5 71 6 0 (0.91549 0.08451) \*  
## 5) CP>=3.5 37 16 0 (0.56757 0.43243)   
## 10) Ca< 0.5 24 5 0 (0.79167 0.20833) \*  
## 11) Ca>=0.5 13 2 1 (0.15385 0.84615) \*  
## 3) Thal>=4.5 90 21 1 (0.23333 0.76667)   
## 6) CP< 3.5 28 12 0 (0.57143 0.42857)   
## 12) Thalach>=144 20 5 0 (0.75000 0.25000) \*  
## 13) Thalach< 144 8 1 1 (0.12500 0.87500) \*  
## 7) CP>=3.5 62 5 1 (0.08065 0.91935) \*

**summary**(heart.tree)

## Call:  
## rpart(formula = factor(Disease) ~ Age + Sex + CP + Trestbps +   
## Chol + Fbs + Restecg + Thalach + Exang + Oldpeak + Slope +   
## Ca + Thal, data = trainData, method = "class")  
## n= 198   
##   
## CP nsplit rel error xerror xstd  
## 1 0.52747 0 1.0000 1.0000 0.07706  
## 2 0.05495 1 0.4725 0.5714 0.06805  
## 3 0.04945 3 0.3626 0.4945 0.06480  
## 4 0.01000 5 0.2637 0.4176 0.06089  
##   
## Variable importance  
## Thal CP Thalach Oldpeak Sex Slope Ca Age   
## 26 20 17 10 9 8 6 2   
## Chol Exang Trestbps   
## 1 1 1   
##   
## Node number 1: 198 observations, complexity param=0.5275  
## predicted class=0 expected loss=0.4596 P(node) =1  
## class counts: 107 91  
## probabilities: 0.540 0.460   
## left son=2 (108 obs) right son=3 (90 obs)  
## Primary splits:  
## Thal < 4.5 to the left, improve=31.12, (0 missing)  
## CP < 3.5 to the left, improve=30.56, (0 missing)  
## Thalach < 145.5 to the right, improve=21.79, (0 missing)  
## Ca < 0.5 to the left, improve=20.74, (0 missing)  
## Oldpeak < 1.95 to the left, improve=17.30, (0 missing)  
## Surrogate splits:  
## Thalach < 147.5 to the right, agree=0.697, adj=0.333, (0 split)  
## Sex < 0.5 to the left, agree=0.682, adj=0.300, (0 split)  
## Oldpeak < 1.7 to the left, agree=0.677, adj=0.289, (0 split)  
## CP < 3.5 to the left, agree=0.672, adj=0.278, (0 split)  
## Slope < 1.5 to the left, agree=0.672, adj=0.278, (0 split)  
##   
## Node number 2: 108 observations, complexity param=0.04945  
## predicted class=0 expected loss=0.2037 P(node) =0.5455  
## class counts: 86 22  
## probabilities: 0.796 0.204   
## left son=4 (71 obs) right son=5 (37 obs)  
## Primary splits:  
## CP < 3.5 to the left, improve=5.889, (0 missing)  
## Age < 58.5 to the left, improve=5.539, (0 missing)  
## Thalach < 127.5 to the right, improve=5.471, (0 missing)  
## Ca < 0.5 to the left, improve=4.971, (0 missing)  
## Chol < 267.5 to the left, improve=2.943, (0 missing)  
## Surrogate splits:  
## Thalach < 127.5 to the right, agree=0.722, adj=0.189, (0 split)  
## Exang < 0.5 to the left, agree=0.713, adj=0.162, (0 split)  
## Age < 60.5 to the left, agree=0.704, adj=0.135, (0 split)  
## Oldpeak < 2.1 to the left, agree=0.694, adj=0.108, (0 split)  
## Ca < 1.5 to the left, agree=0.685, adj=0.081, (0 split)  
##   
## Node number 3: 90 observations, complexity param=0.05495  
## predicted class=1 expected loss=0.2333 P(node) =0.4545  
## class counts: 21 69  
## probabilities: 0.233 0.767   
## left son=6 (28 obs) right son=7 (62 obs)  
## Primary splits:  
## CP < 3.5 to the left, improve=9.292, (0 missing)  
## Thalach < 145.5 to the right, improve=5.648, (0 missing)  
## Oldpeak < 0.9 to the left, improve=4.721, (0 missing)  
## Ca < 0.5 to the left, improve=4.635, (0 missing)  
## Exang < 0.5 to the left, improve=3.492, (0 missing)  
## Surrogate splits:  
## Thalach < 172 to the right, agree=0.756, adj=0.214, (0 split)  
## Trestbps < 109 to the left, agree=0.711, adj=0.071, (0 split)  
## Age < 67.5 to the right, agree=0.700, adj=0.036, (0 split)  
## Oldpeak < 0.7 to the left, agree=0.700, adj=0.036, (0 split)  
##   
## Node number 4: 71 observations  
## predicted class=0 expected loss=0.08451 P(node) =0.3586  
## class counts: 65 6  
## probabilities: 0.915 0.085   
##   
## Node number 5: 37 observations, complexity param=0.04945  
## predicted class=0 expected loss=0.4324 P(node) =0.1869  
## class counts: 21 16  
## probabilities: 0.568 0.432   
## left son=10 (24 obs) right son=11 (13 obs)  
## Primary splits:  
## Ca < 0.5 to the left, improve=6.861, (0 missing)  
## Age < 59.5 to the left, improve=4.952, (0 missing)  
## Oldpeak < 0.75 to the left, improve=3.668, (0 missing)  
## Thalach < 128 to the right, improve=2.722, (0 missing)  
## Chol < 266.5 to the left, improve=2.091, (0 missing)  
## Surrogate splits:  
## Oldpeak < 1.7 to the left, agree=0.784, adj=0.385, (0 split)  
## Thalach < 120 to the right, agree=0.757, adj=0.308, (0 split)  
## Age < 66.5 to the left, agree=0.703, adj=0.154, (0 split)  
## Chol < 266.5 to the left, agree=0.703, adj=0.154, (0 split)  
## Slope < 2.5 to the left, agree=0.703, adj=0.154, (0 split)  
##   
## Node number 6: 28 observations, complexity param=0.05495  
## predicted class=0 expected loss=0.4286 P(node) =0.1414  
## class counts: 16 12  
## probabilities: 0.571 0.429   
## left son=12 (20 obs) right son=13 (8 obs)  
## Primary splits:  
## Thalach < 144 to the right, improve=4.464, (0 missing)  
## Ca < 0.5 to the left, improve=2.381, (0 missing)  
## Chol < 228 to the left, improve=1.524, (0 missing)  
## Trestbps < 122 to the left, improve=1.339, (0 missing)  
## Slope < 1.5 to the left, improve=1.129, (0 missing)  
## Surrogate splits:  
## Sex < 0.5 to the right, agree=0.786, adj=0.250, (0 split)  
## Age < 58.5 to the left, agree=0.750, adj=0.125, (0 split)  
## Thal < 6.5 to the right, agree=0.750, adj=0.125, (0 split)  
##   
## Node number 7: 62 observations  
## predicted class=1 expected loss=0.08065 P(node) =0.3131  
## class counts: 5 57  
## probabilities: 0.081 0.919   
##   
## Node number 10: 24 observations  
## predicted class=0 expected loss=0.2083 P(node) =0.1212  
## class counts: 19 5  
## probabilities: 0.792 0.208   
##   
## Node number 11: 13 observations  
## predicted class=1 expected loss=0.1538 P(node) =0.06566  
## class counts: 2 11  
## probabilities: 0.154 0.846   
##   
## Node number 12: 20 observations  
## predicted class=0 expected loss=0.25 P(node) =0.101  
## class counts: 15 5  
## probabilities: 0.750 0.250   
##   
## Node number 13: 8 observations  
## predicted class=1 expected loss=0.125 P(node) =0.0404  
## class counts: 1 7  
## probabilities: 0.125 0.875

**prp**(heart.tree, extra = 100, main = model.name)



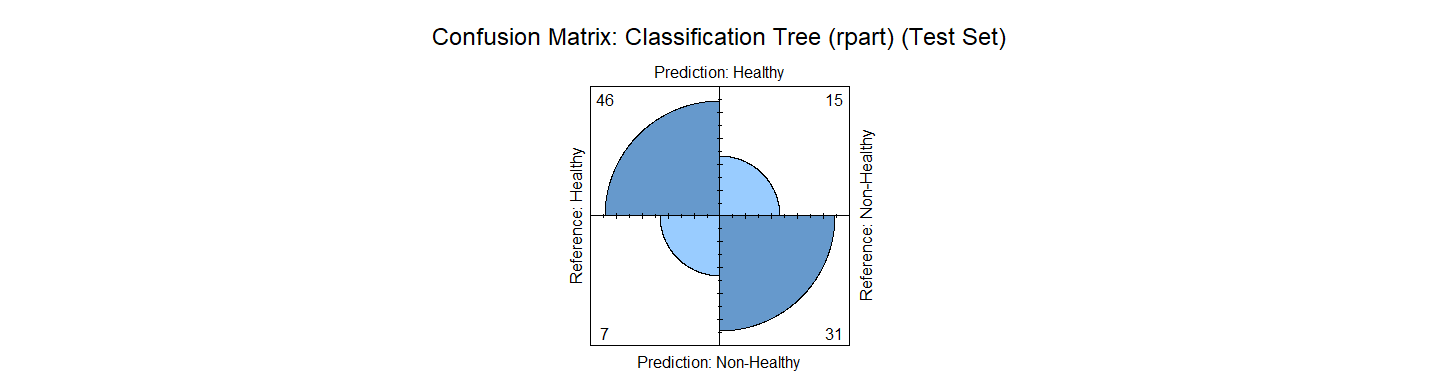
*# Let's use the model in test set*  
test\_pred\_classtree <- **predict**(heart.tree,newdata = testData, type = "class")  
test\_pred\_classtree

## 3 5 6 8 9 15 18 19 20 23 24 25 28 31 34 36 42 46   
## 1 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0   
## 47 52 55 56 57 59 65 67 68 70 71 75 78 79 80 82 83 84   
## 0 1 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0 0   
## 95 97 101 105 112 116 118 119 123 126 129 133 137 140 141 143 144 147   
## 0 1 0 1 1 1 0 1 0 0 0 0 1 0 0 0 1 1   
## 153 157 161 162 163 165 166 169 170 173 177 181 182 184 186 190 206 208   
## 0 1 0 1 0 0 1 1 0 0 1 1 1 0 0 0 1 1   
## 209 211 219 226 229 230 232 233 234 236 239 241 244 250 251 255 258 271   
## 0 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1   
## 273 275 283 285 287 291 296 299 302   
## 1 0 1 1 1 0 0 1 0   
## Levels: 0 1

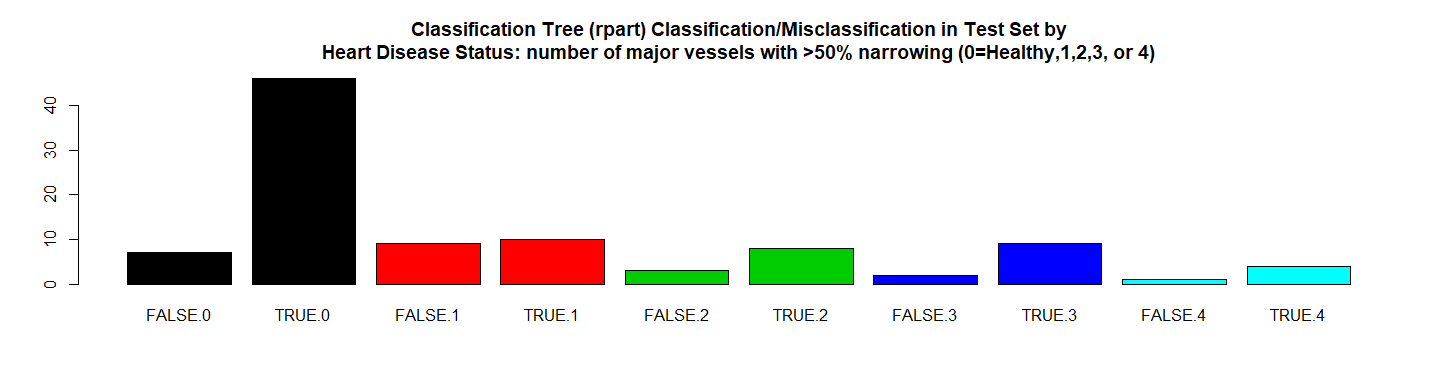
res.classtree <- **confusionMatrix**(test\_pred\_classtree, Y.testData)  
res.classtree

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 46 15  
## 1 7 31  
##   
## Accuracy : 0.778   
## 95% CI : (0.683, 0.855)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 5.07e-07   
##   
## Kappa : 0.548   
## Mcnemar's Test P-Value : 0.136   
##   
## Sensitivity : 0.868   
## Specificity : 0.674   
## Pos Pred Value : 0.754   
## Neg Pred Value : 0.816   
## Prevalence : 0.535   
## Detection Rate : 0.465   
## Detection Prevalence : 0.616   
## Balanced Accuracy : 0.771   
##   
## 'Positive' Class : 0   
##

**print\_confusionm**(res.classtree,   
 main = **paste0**("Confusion Matrix: ", model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_classtree,  
 Y.testData,  
 main, testData)



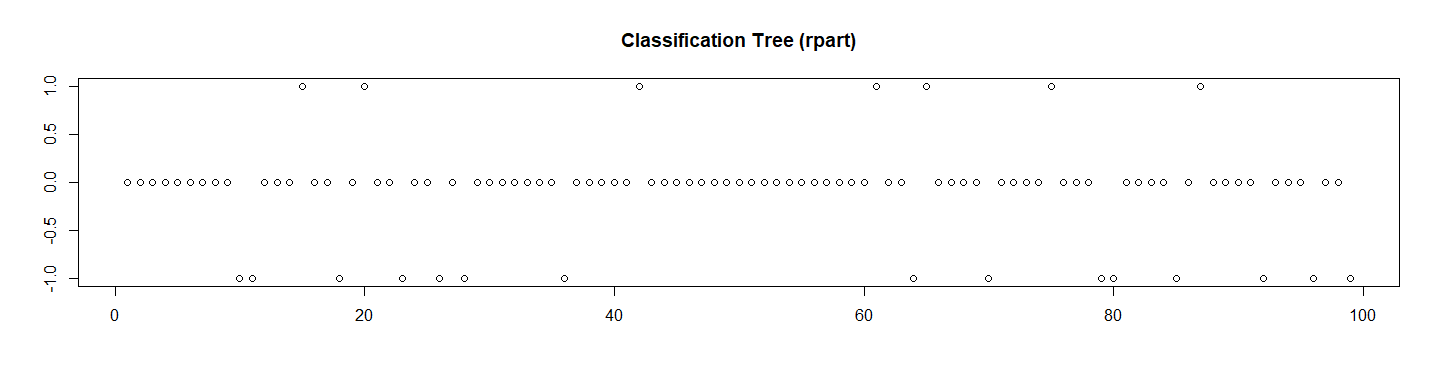
*# RMSE Resulting*  
(rmse\_classtree <- **sqrt**(**mean**((**as.numeric**(test\_pred\_classtree)**-as.numeric**(Y.testData))**^**2)))

## [1] 0.4714

*# Residuals Plot*  
(r\_classtree <- (**as.numeric**(test\_pred\_classtree)**-as.numeric**(Y.testData)))

## [1] 0 0 0 0 0 0 0 0 0 -1 -1 0 0 0 1 0 0 -1 0 1 0 0 -1  
## [24] 0 0 -1 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 1 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 -1 1 0 0 0 0  
## [70] -1 0 0 0 0 1 0 0 0 -1 -1 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 -1 0 0 -1

**plot**(r\_classtree, main = model.name,xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_classtree\_prob <- **predict**(heart.tree, newdata = X.testData, type ="prob")[,2]  
  
res.classtree.roc <- **roc**(**as.numeric**(Y.testData),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_classtree\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 NA,  
 "Prediction Accuracy in Test Set" =   
 res.classtree**$**overall[1],  
 "RMSE Test"= rmse\_classtree,  
 "ROC" = res.classtree.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 14. Fast-and-Frugal Decision Trees (FFTrees)

############################################  
*# 14. Fast-and-Frugal Decision Trees (FFTrees)*  
model.name <- "Fast-and-Frugal Decision Trees (FFTrees)"  
model.name17 <- model.name  
  
trainDataF <- trainData  
testDataF <- testData  
  
*# Make Disease Boolean*  
trainDataF**$**Disease <- **ifelse**(trainDataF**$**Disease **==** 1,TRUE,FALSE)   
testDataF**$**Disease <- **ifelse**(testDataF**$**Disease **==** 1,TRUE,FALSE)  
  
*# Create an FFTrees object called `heart\_FFT`*  
heart\_FFT <- **FFTrees**(formula = Disease **~** Age **+** Sex **+** CP **+** Trestbps **+**   
 Chol **+** Fbs **+** Restecg **+** Thalach **+** Exang **+**   
 Oldpeak **+** Slope **+** Ca **+** Thal,  
 data = trainDataF,  
 data.test = testDataF,  
 main = "ER Decisions",  
 decision.labels = **c**("Stable", "H Attack"))

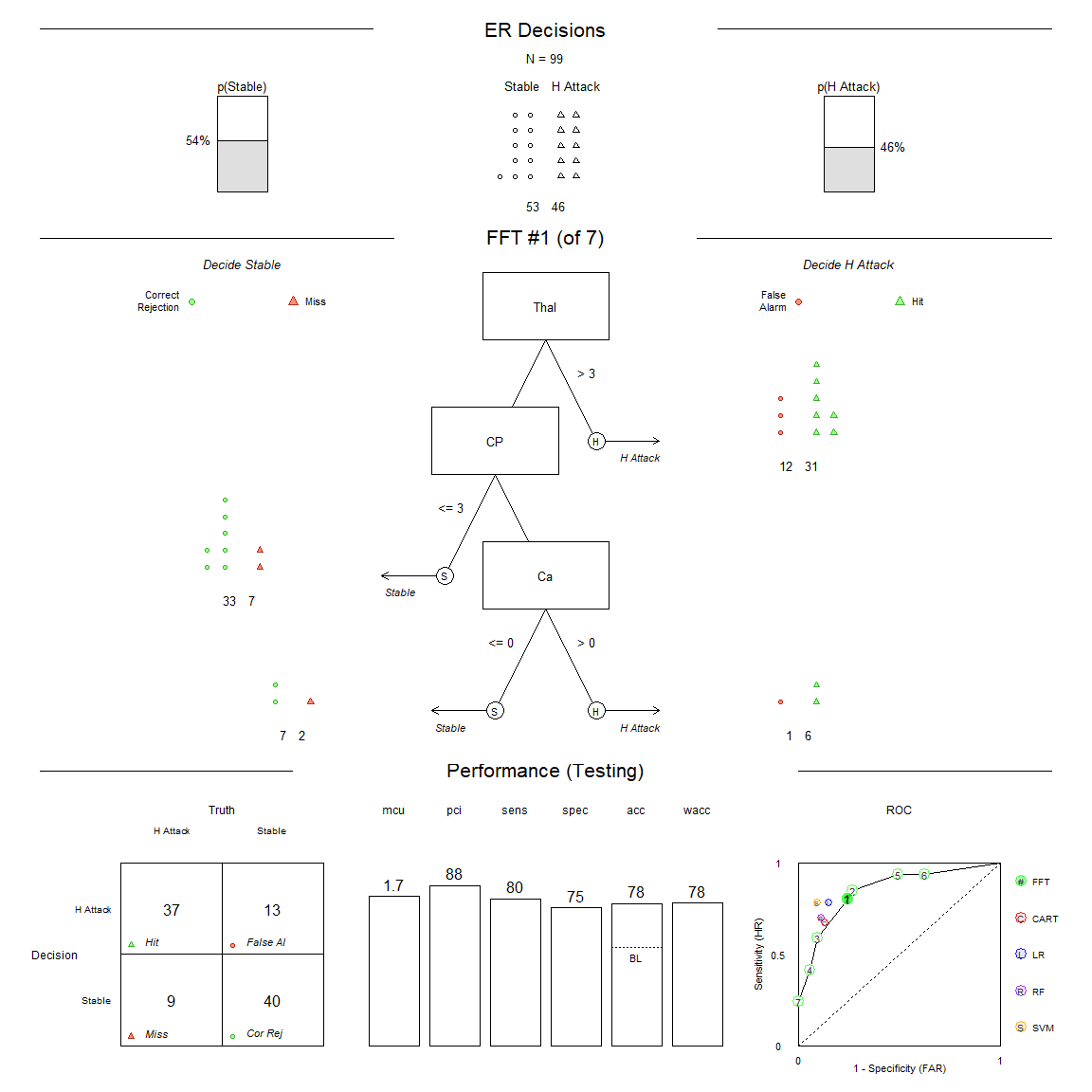
## Growing FFTs with ifan

## Fitting non-FFTrees algorithms for comparison (you can turn this off with do.comp = FALSE) ...

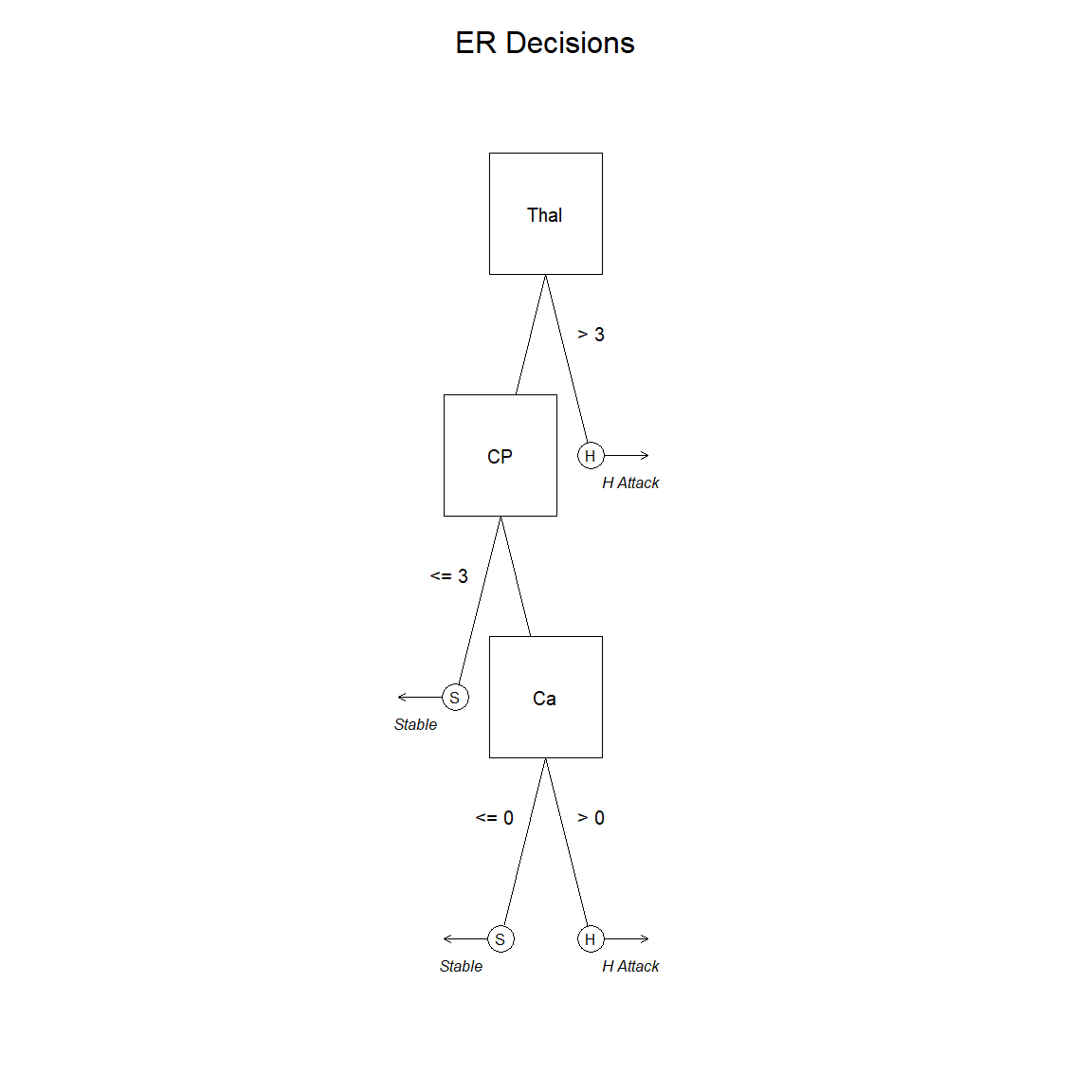
*# Print the summary statistics*  
heart\_FFT

## ER Decisions  
## FFT #1 predicts Disease using 3 cues: {Thal,CP,Ca}  
##   
## [1] If Thal > 3, predict H Attack.  
## [2] If CP <= 3, predict Stable.  
## [3] If Ca <= 0, predict Stable, otherwise, predict H Attack.  
##   
## train test  
## cases :n 198.00 99.00  
## speed :mcu 1.73 1.73  
## frugality :pci 0.88 0.88  
## accuracy :acc 0.83 0.78  
## weighted :wacc 0.83 0.78  
## sensitivity :sens 0.88 0.80  
## specificity :spec 0.79 0.75  
##   
## pars: algorithm = 'ifan', goal = 'wacc', goal.chase = 'bacc', sens.w = 0.5, max.levels = 4

*# Visualise the tree applied to the test data heart.test*  
**plot**(heart\_FFT, data = "test")



**plot**(heart\_FFT, stats = FALSE)



Let’s see the results:

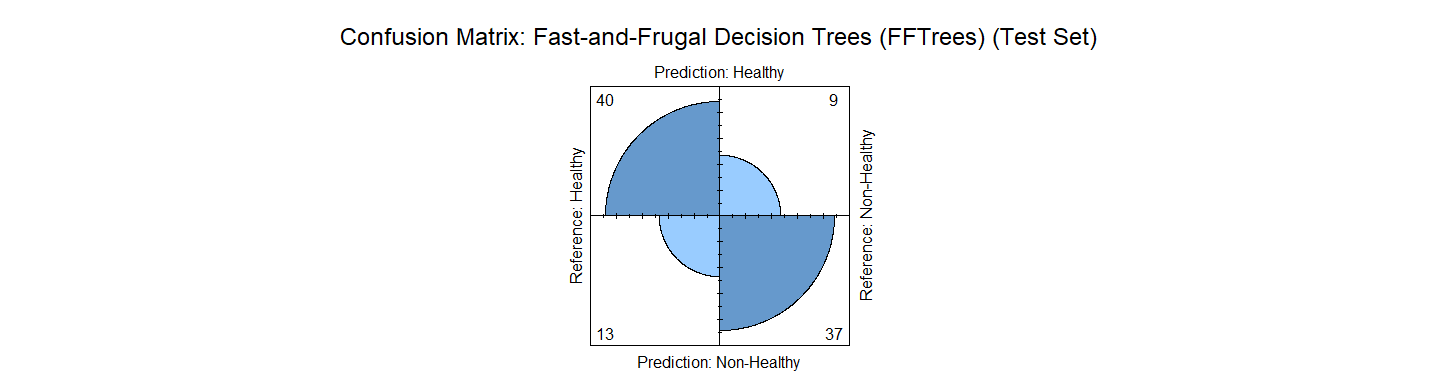
*# Let's use the model in test set*  
test\_pred\_fft <- **predict**(heart\_FFT,newdata = testDataF, type = "class")  
test\_pred\_fft

## [1] TRUE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE  
## [12] TRUE FALSE FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE  
## [23] TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE  
## [34] FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE TRUE  
## [45] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE  
## [56] TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE TRUE TRUE  
## [67] TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE  
## [78] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE  
## [89] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE

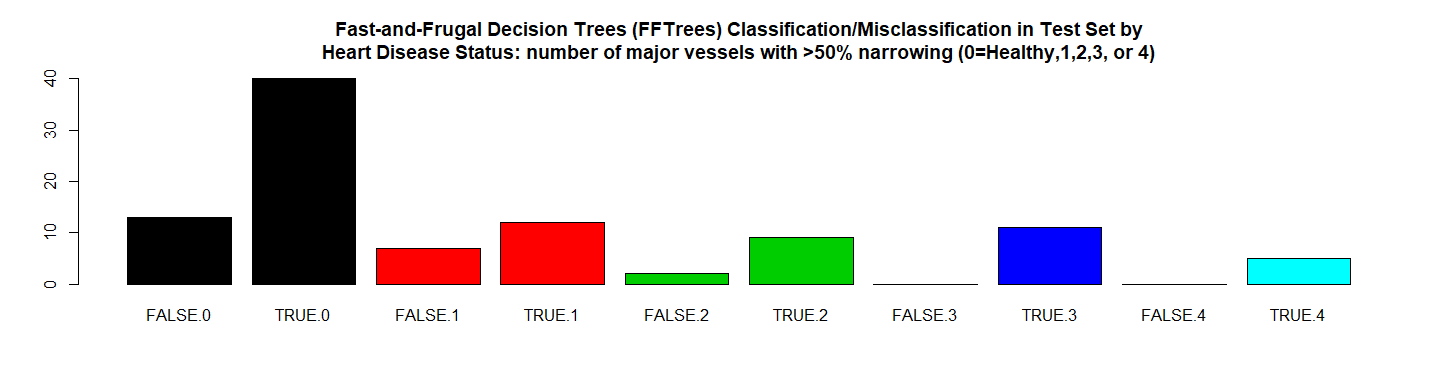
res.fft <- **confusionMatrix**(test\_pred\_fft, testDataF**$**Disease)  
res.fft

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 40 9  
## TRUE 13 37  
##   
## Accuracy : 0.778   
## 95% CI : (0.683, 0.855)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 5.07e-07   
##   
## Kappa : 0.556   
## Mcnemar's Test P-Value : 0.522   
##   
## Sensitivity : 0.755   
## Specificity : 0.804   
## Pos Pred Value : 0.816   
## Neg Pred Value : 0.740   
## Prevalence : 0.535   
## Detection Rate : 0.404   
## Detection Prevalence : 0.495   
## Balanced Accuracy : 0.780   
##   
## 'Positive' Class : FALSE   
##

**print\_confusionm**(res.fft,   
 main = **paste0**("Confusion Matrix: ", model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_fft,  
 testDataF**$**Disease,  
 main, testDataF)



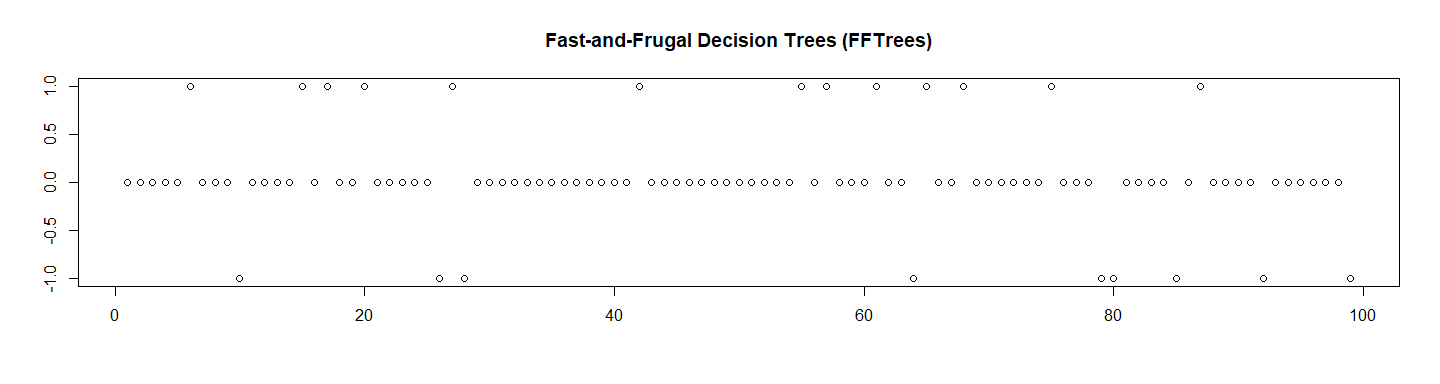
*# RMSE Resulting*  
(rmse\_fft <- **sqrt**(**mean**((**as.numeric**(test\_pred\_fft)**-as.numeric**(testDataF**$**Disease))**^**2)))

## [1] 0.4714

*# Residuals Plot*  
(r\_fft <- (**as.numeric**(test\_pred\_fft)**-as.numeric**(testDataF**$**Disease)))

## [1] 0 0 0 0 0 1 0 0 0 -1 0 0 0 0 1 0 1 0 0 1 0 0 0  
## [24] 0 0 -1 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 -1 -1 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 0 0 0 -1

**plot**(r\_fft, main = model.name,xlab = "", ylab = "")



*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_fft\_prob <- **predict**(heart\_FFT, newdata = testDataF, type ="prob")[,2]  
  
res.fft.roc <- **roc**(**as.numeric**(testDataF**$**Disease),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_fft\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(heart\_FFT**$**tree.stats**$**train**$**acc,na.rm = TRUE),  
 "Prediction Accuracy in Test Set" =   
 res.fft**$**overall[1],  
 "RMSE Test"= rmse\_fft,  
 "ROC" = res.fft.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 15. Custom Fast-and-Frugal Decision Trees (FFTrees)

############################################  
*# 15. Custom Fast-and-Frugal Decision Trees (FFTrees)*  
model.name <- "Custom Fast-and-Frugal Decision Trees (FFTrees)"  
model.name18 <- model.name  
  
*# Create a Custom an FFTrees object called `custom\_\_FFT`*  
  
custom\_FFT <- **FFTrees**(formula = Disease **~** Age **+** Sex **+** CP **+** Trestbps **+**   
 Chol **+** Fbs **+** Restecg **+** Thalach **+** Exang **+**   
 Oldpeak **+** Slope **+** Ca **+** Thal,   
 data = trainDataF,   
 data.test = testDataF,   
 main = "Custom's Tree",   
 decision.labels = **c**("Stable", "Attack",   
 sens.w = 0.99,  
 goal = "acc"))

## Growing FFTs with ifan

## Fitting non-FFTrees algorithms for comparison (you can turn this off with do.comp = FALSE) ...

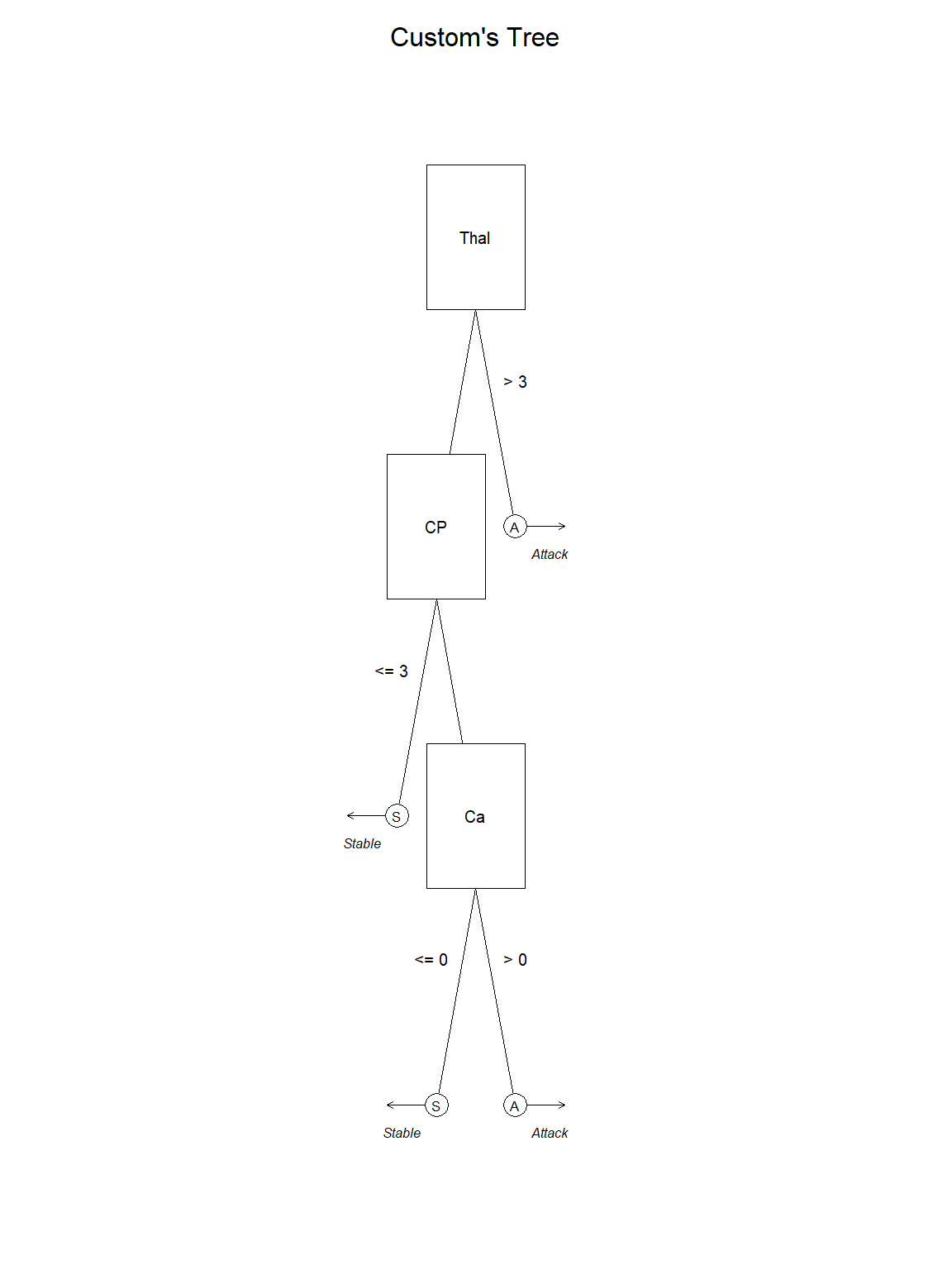
*# Print the summary statistics*  
custom\_FFT

## Custom's Tree  
## FFT #1 predicts Disease using 3 cues: {Thal,CP,Ca}  
##   
## [1] If Thal > 3, predict Attack.  
## [2] If CP <= 3, predict Stable.  
## [3] If Ca <= 0, predict Stable, otherwise, predict Attack.  
##   
## train test  
## cases :n 198.00 99.00  
## speed :mcu 1.73 1.73  
## frugality :pci 0.88 0.88  
## accuracy :acc 0.83 0.78  
## weighted :wacc 0.83 0.78  
## sensitivity :sens 0.88 0.80  
## specificity :spec 0.79 0.75  
##   
## pars: algorithm = 'ifan', goal = 'wacc', goal.chase = 'bacc', sens.w = 0.5, max.levels = 4

*# Visualise the tree applied to the test data heart.test*  
**plot**(custom\_FFT, data = "test")



**plot**(custom\_FFT, stats = FALSE)



Let’s see the results:

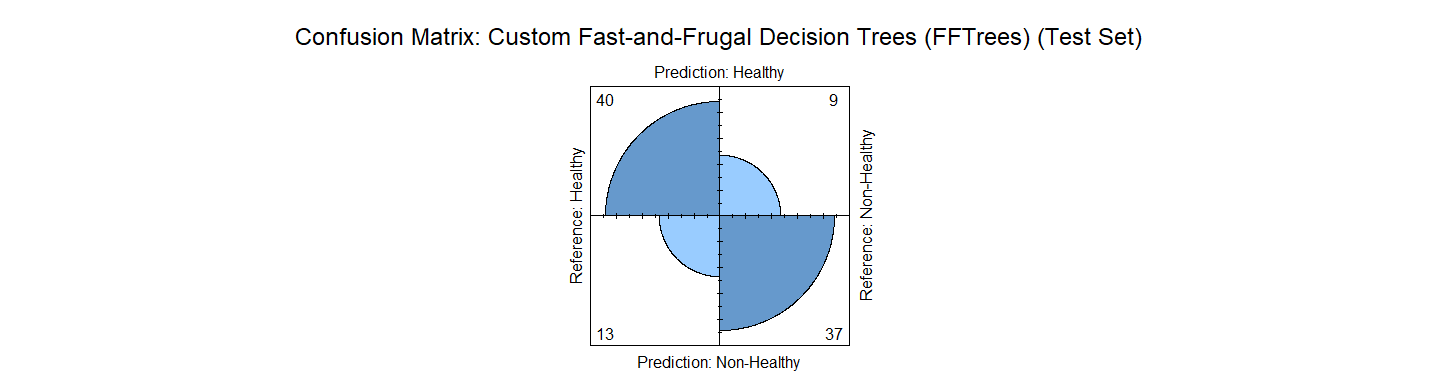
*# Let's use the model in test set*  
test\_pred\_cfft <- **predict**(custom\_FFT,newdata = testDataF, type = "class")  
test\_pred\_cfft

## [1] TRUE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE  
## [12] TRUE FALSE FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE  
## [23] TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE  
## [34] FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE TRUE  
## [45] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE TRUE  
## [56] TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE FALSE TRUE TRUE  
## [67] TRUE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE  
## [78] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE  
## [89] FALSE TRUE TRUE FALSE TRUE TRUE TRUE TRUE FALSE TRUE FALSE

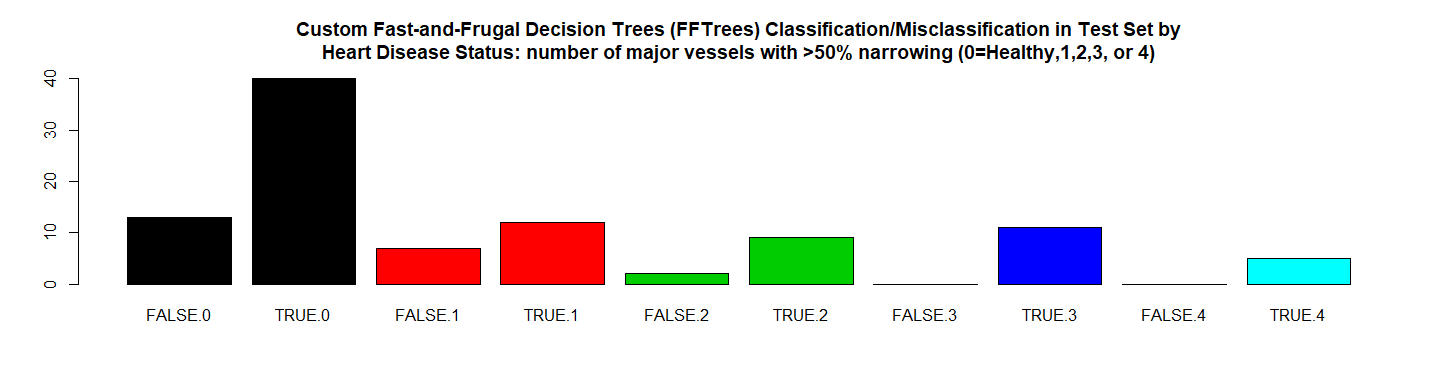
res.cfft <- **confusionMatrix**(test\_pred\_cfft, testDataF**$**Disease)  
res.cfft

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 40 9  
## TRUE 13 37  
##   
## Accuracy : 0.778   
## 95% CI : (0.683, 0.855)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 5.07e-07   
##   
## Kappa : 0.556   
## Mcnemar's Test P-Value : 0.522   
##   
## Sensitivity : 0.755   
## Specificity : 0.804   
## Pos Pred Value : 0.816   
## Neg Pred Value : 0.740   
## Prevalence : 0.535   
## Detection Rate : 0.404   
## Detection Prevalence : 0.495   
## Balanced Accuracy : 0.780   
##   
## 'Positive' Class : FALSE   
##

**print\_confusionm**(res.cfft,   
 main = **paste0**("Confusion Matrix: ", model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))



main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_cfft,  
 testDataF**$**Disease,  
 main, testDataF)



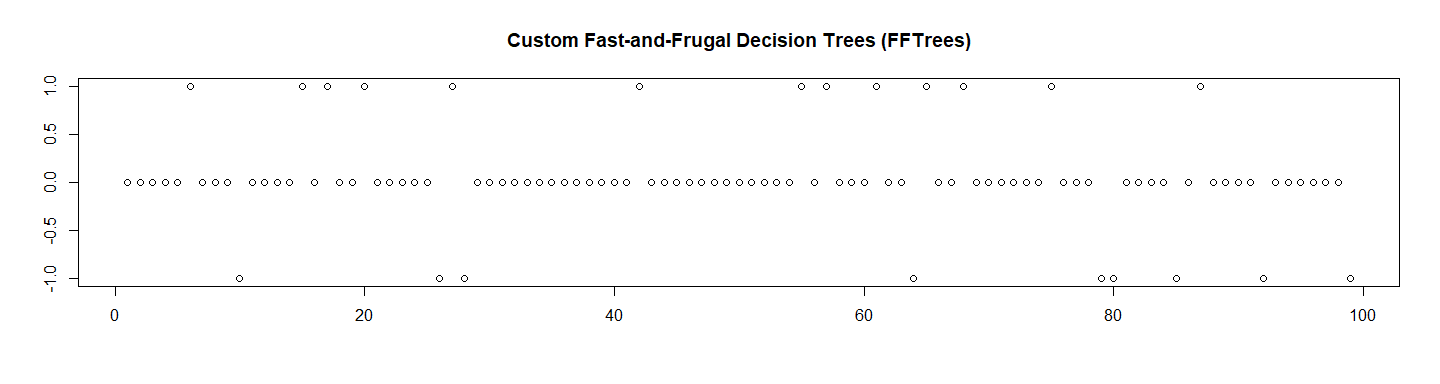
*# RMSE Resulting*  
(rmse\_cfft <- **sqrt**(**mean**((**as.numeric**(test\_pred\_cfft)**-as.numeric**(testDataF**$**Disease))**^**2)))

## [1] 0.4714

*# Residuals Plot*  
(r\_cfft <- (**as.numeric**(test\_pred\_cfft)**-as.numeric**(testDataF**$**Disease)))

## [1] 0 0 0 0 0 1 0 0 0 -1 0 0 0 0 1 0 1 0 0 1 0 0 0  
## [24] 0 0 -1 1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 -1 1 0 0 1 0  
## [70] 0 0 0 0 0 1 0 0 0 -1 -1 0 0 0 0 -1 0 1 0 0 0 0 -1  
## [93] 0 0 0 0 0 0 -1

**plot**(r\_cfft, main = model.name,xlab = "", ylab = "")



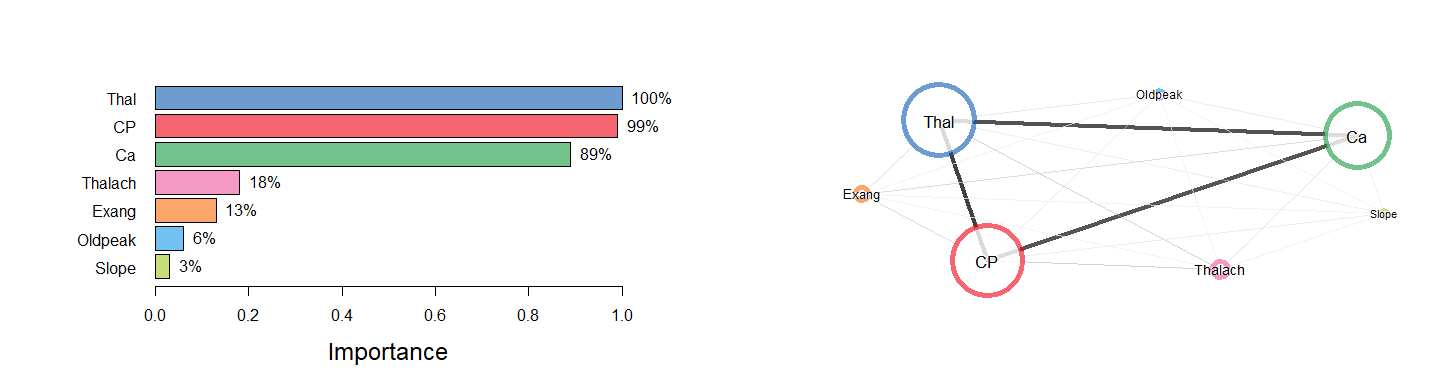
*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_cfft\_prob <- **predict**(custom\_FFT, newdata = testDataF, type ="prob")[,2]  
  
res.cfft.roc <- **roc**(**as.numeric**(testDataF**$**Disease),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_cfft\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(custom\_FFT**$**tree.stats**$**train**$**acc,na.rm = TRUE),  
 "Prediction Accuracy in Test Set" =   
 res.cfft**$**overall[1],  
 "RMSE Test"= rmse\_cfft,  
 "ROC" = res.cfft.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))

# 16. Forest of Fast-and-Frugal Decision Trees (FFForest)

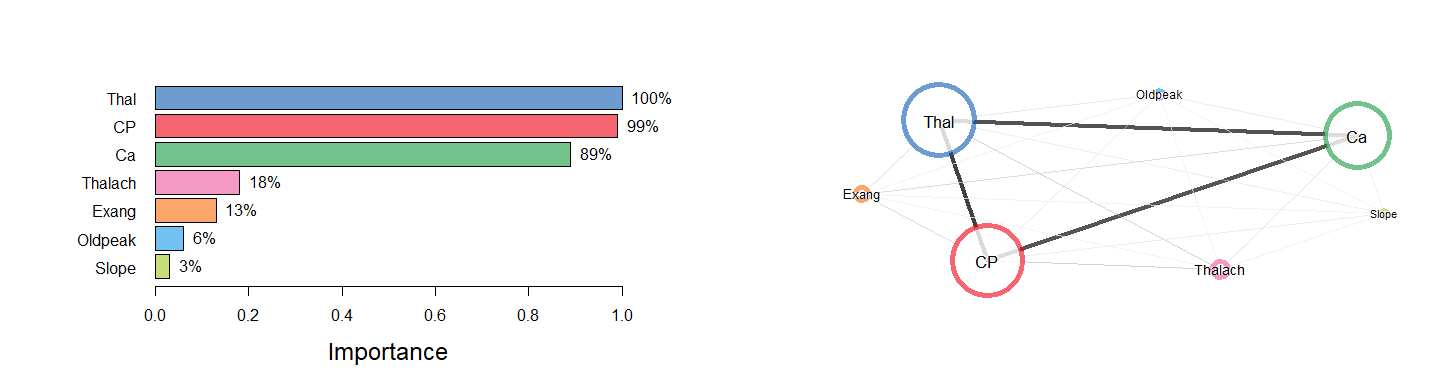
############################################  
*# 16. Forest of Fast-and-Frugal Decision Trees (FFFOrest)*   
model.name <- "Forest of Fast-and-Frugal Decision Trees (FFForest)"  
model.name19 <- model.name  
num\_trees <- 100  
  
file.name <- **paste**(work.dir, "heart\_fff.Rds", sep="/")  
file.name.et <- **paste**(work.dir, "heart\_fff-et.Rds", sep="/")  
  
**if** (**file.exists**(file.name)) {  
 **load**(file.name,.GlobalEnv)  
 **load**(file.name.et,.GlobalEnv)  
} **else** {  
   
 t1 <- **proc.time**()  
 *# Create an FFForest object (can take a few minutes)*  
 heart\_fff <- **FFForest**(formula = Disease **~** Age **+** Sex **+** CP **+** Trestbps **+**   
 Chol **+** Fbs **+** Restecg **+** Thalach **+** Exang **+**   
 Oldpeak **+** Slope **+** Ca **+** Thal,  
 data = **rbind**(trainDataF,testDataF),  
 ntree = num\_trees,   
 train.p = 2**/**3,  
 verbose = FALSE)  
 t2 <- **proc.time**()  
 et <- **elapsed\_time**(t1,t2)  
 **print**(**paste0**(model.name," Executed ... time:", et))  
 **save**(heart\_fff, file = file.name)  
 **save**(et, file = file.name.et)  
}  
  
## [1] "Forest of Fast-and-Frugal Decision Trees (FFForest) Executed ... time:127"

Let’s see the results:

*# Visualise the tree applied to the test data heart.test*  
**plot**(heart\_fff, data = "test")



**plot**(heart\_fff, stats = FALSE)



**plot**(heart\_fff)  
  
*# Let's choose the best tree*  
  
best <- 0  
tree\_id <- 0  
  
**print**("Accuracy Results for Trees")

## [1] "Accuracy Results for Trees"

**for** (i **in** 1**:**num\_trees) {  
   
 fff\_tree <- (heart\_fff**$**fft.models[i])[[1]]  
   
 test\_pred\_fff <- **predict**(fff\_tree, data = testDataF)  
 test\_pred\_fff  
   
 res.fff <- **confusionMatrix**(test\_pred\_fff, testDataF**$**Disease)  
   
 fff\_tre\_acc <- res.fff**$**overall[1]  
 **print**(**paste0**("Tree:",i," Accuracy:",fff\_tre\_acc))  
   
   
 **if** ( (best**==**0) **||** (fff\_tre\_acc **>** best) ) {  
   
 best <- fff\_tre\_acc  
 tree\_id <- i  
 }  
}

## [1] "Tree:1 Accuracy:0.777777777777778"  
## [1] "Tree:2 Accuracy:0.747474747474748"  
## [1] "Tree:3 Accuracy:0.757575757575758"  
## [1] "Tree:4 Accuracy:0.757575757575758"  
## [1] "Tree:5 Accuracy:0.777777777777778"  
## [1] "Tree:6 Accuracy:0.787878787878788"  
## [1] "Tree:7 Accuracy:0.797979797979798"  
## [1] "Tree:8 Accuracy:0.777777777777778"  
## [1] "Tree:9 Accuracy:0.767676767676768"  
## [1] "Tree:10 Accuracy:0.787878787878788"  
## [1] "Tree:11 Accuracy:0.757575757575758"  
## [1] "Tree:12 Accuracy:0.777777777777778"  
## [1] "Tree:13 Accuracy:0.777777777777778"  
## [1] "Tree:14 Accuracy:0.777777777777778"  
## [1] "Tree:15 Accuracy:0.767676767676768"  
## [1] "Tree:16 Accuracy:0.777777777777778"  
## [1] "Tree:17 Accuracy:0.767676767676768"  
## [1] "Tree:18 Accuracy:0.777777777777778"  
## [1] "Tree:19 Accuracy:0.767676767676768"  
## [1] "Tree:20 Accuracy:0.777777777777778"  
## [1] "Tree:21 Accuracy:0.777777777777778"  
## [1] "Tree:22 Accuracy:0.777777777777778"  
## [1] "Tree:23 Accuracy:0.757575757575758"  
## [1] "Tree:24 Accuracy:0.777777777777778"  
## [1] "Tree:25 Accuracy:0.757575757575758"  
## [1] "Tree:26 Accuracy:0.777777777777778"  
## [1] "Tree:27 Accuracy:0.777777777777778"  
## [1] "Tree:28 Accuracy:0.777777777777778"  
## [1] "Tree:29 Accuracy:0.767676767676768"  
## [1] "Tree:30 Accuracy:0.777777777777778"  
## [1] "Tree:31 Accuracy:0.787878787878788"  
## [1] "Tree:32 Accuracy:0.757575757575758"  
## [1] "Tree:33 Accuracy:0.777777777777778"  
## [1] "Tree:34 Accuracy:0.757575757575758"  
## [1] "Tree:35 Accuracy:0.767676767676768"  
## [1] "Tree:36 Accuracy:0.777777777777778"  
## [1] "Tree:37 Accuracy:0.777777777777778"  
## [1] "Tree:38 Accuracy:0.757575757575758"  
## [1] "Tree:39 Accuracy:0.767676767676768"  
## [1] "Tree:40 Accuracy:0.787878787878788"  
## [1] "Tree:41 Accuracy:0.777777777777778"  
## [1] "Tree:42 Accuracy:0.747474747474748"  
## [1] "Tree:43 Accuracy:0.777777777777778"  
## [1] "Tree:44 Accuracy:0.767676767676768"  
## [1] "Tree:45 Accuracy:0.777777777777778"  
## [1] "Tree:46 Accuracy:0.777777777777778"  
## [1] "Tree:47 Accuracy:0.747474747474748"  
## [1] "Tree:48 Accuracy:0.767676767676768"  
## [1] "Tree:49 Accuracy:0.777777777777778"  
## [1] "Tree:50 Accuracy:0.777777777777778"  
## [1] "Tree:51 Accuracy:0.777777777777778"  
## [1] "Tree:52 Accuracy:0.777777777777778"  
## [1] "Tree:53 Accuracy:0.757575757575758"  
## [1] "Tree:54 Accuracy:0.777777777777778"  
## [1] "Tree:55 Accuracy:0.767676767676768"  
## [1] "Tree:56 Accuracy:0.787878787878788"  
## [1] "Tree:57 Accuracy:0.787878787878788"  
## [1] "Tree:58 Accuracy:0.767676767676768"  
## [1] "Tree:59 Accuracy:0.757575757575758"  
## [1] "Tree:60 Accuracy:0.777777777777778"  
## [1] "Tree:61 Accuracy:0.767676767676768"  
## [1] "Tree:62 Accuracy:0.757575757575758"  
## [1] "Tree:63 Accuracy:0.757575757575758"  
## [1] "Tree:64 Accuracy:0.787878787878788"  
## [1] "Tree:65 Accuracy:0.777777777777778"  
## [1] "Tree:66 Accuracy:0.767676767676768"  
## [1] "Tree:67 Accuracy:0.777777777777778"  
## [1] "Tree:68 Accuracy:0.767676767676768"  
## [1] "Tree:69 Accuracy:0.787878787878788"  
## [1] "Tree:70 Accuracy:0.777777777777778"  
## [1] "Tree:71 Accuracy:0.777777777777778"  
## [1] "Tree:72 Accuracy:0.747474747474748"  
## [1] "Tree:73 Accuracy:0.787878787878788"  
## [1] "Tree:74 Accuracy:0.777777777777778"  
## [1] "Tree:75 Accuracy:0.757575757575758"  
## [1] "Tree:76 Accuracy:0.757575757575758"  
## [1] "Tree:77 Accuracy:0.727272727272727"  
## [1] "Tree:78 Accuracy:0.767676767676768"  
## [1] "Tree:79 Accuracy:0.777777777777778"  
## [1] "Tree:80 Accuracy:0.727272727272727"  
## [1] "Tree:81 Accuracy:0.767676767676768"  
## [1] "Tree:82 Accuracy:0.777777777777778"  
## [1] "Tree:83 Accuracy:0.757575757575758"  
## [1] "Tree:84 Accuracy:0.777777777777778"  
## [1] "Tree:85 Accuracy:0.787878787878788"  
## [1] "Tree:86 Accuracy:0.777777777777778"  
## [1] "Tree:87 Accuracy:0.777777777777778"  
## [1] "Tree:88 Accuracy:0.777777777777778"  
## [1] "Tree:89 Accuracy:0.777777777777778"  
## [1] "Tree:90 Accuracy:0.787878787878788"  
## [1] "Tree:91 Accuracy:0.777777777777778"  
## [1] "Tree:92 Accuracy:0.777777777777778"  
## [1] "Tree:93 Accuracy:0.777777777777778"  
## [1] "Tree:94 Accuracy:0.777777777777778"  
## [1] "Tree:95 Accuracy:0.787878787878788"  
## [1] "Tree:96 Accuracy:0.787878787878788"  
## [1] "Tree:97 Accuracy:0.767676767676768"  
## [1] "Tree:98 Accuracy:0.757575757575758"  
## [1] "Tree:99 Accuracy:0.787878787878788"  
## [1] "Tree:100 Accuracy:0.747474747474748"

**print**(**paste0**("Best Tree id:",tree\_id, " accuracy:", fff\_tre\_acc))

## [1] "Best Tree id:7 accuracy:0.747474747474748"

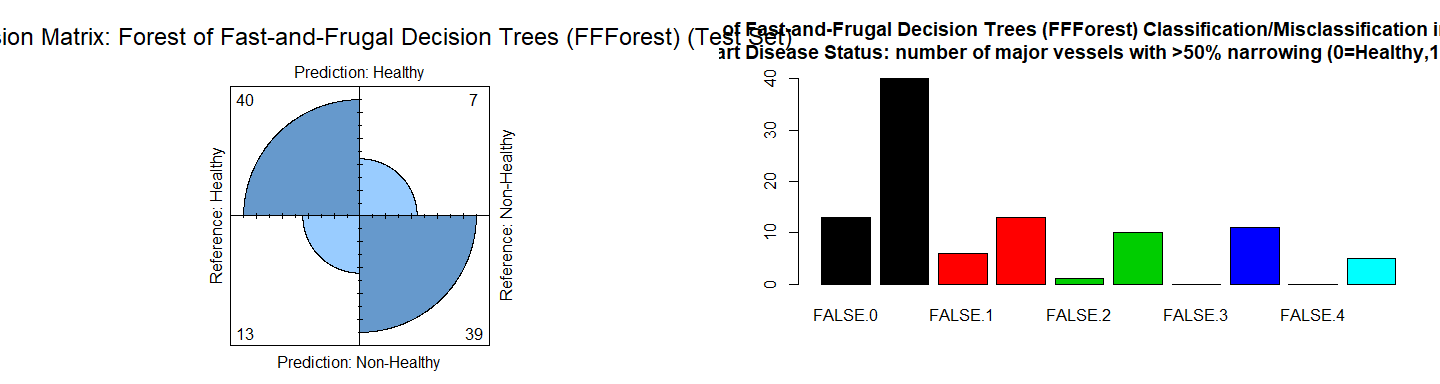
*# Select the best Tree for calculations*  
fff\_tree <- (heart\_fff**$**fft.models[tree\_id])[[1]]  
  
test\_pred\_fff <- **predict**(fff\_tree, data = testDataF)  
test\_pred\_fff

## [1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE  
## [12] TRUE FALSE TRUE TRUE FALSE TRUE TRUE FALSE TRUE TRUE TRUE  
## [23] TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE  
## [34] FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE FALSE TRUE  
## [45] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE FALSE  
## [56] TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE TRUE  
## [67] TRUE FALSE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE  
## [78] TRUE FALSE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE  
## [89] FALSE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE

res.fff <- **confusionMatrix**(test\_pred\_fff, testDataF**$**Disease)  
res.fff

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction FALSE TRUE  
## FALSE 40 7  
## TRUE 13 39  
##   
## Accuracy : 0.798   
## 95% CI : (0.705, 0.872)  
## No Information Rate : 0.535   
## P-Value [Acc > NIR] : 4.8e-08   
##   
## Kappa : 0.597   
## Mcnemar's Test P-Value : 0.264   
##   
## Sensitivity : 0.755   
## Specificity : 0.848   
## Pos Pred Value : 0.851   
## Neg Pred Value : 0.750   
## Prevalence : 0.535   
## Detection Rate : 0.404   
## Detection Prevalence : 0.475   
## Balanced Accuracy : 0.801   
##   
## 'Positive' Class : FALSE   
##

**print\_confusionm**(res.fff,   
 main = **paste0**("Confusion Matrix: ", model.name," (Test Set)"),   
 labels = **c**("Healthy", "Non-Healthy"))  
  
main <- **paste0**(model.name," Classification/Misclassification in Test Set by\n",  
 "Heart Disease Status: number of major vessels with >50% narrowing ",  
 "(0=Healthy,1,2,3, or 4)")  
  
**plot\_results\_by\_type**(test\_pred\_fff,  
 testDataF**$**Disease,  
 main, testDataF)



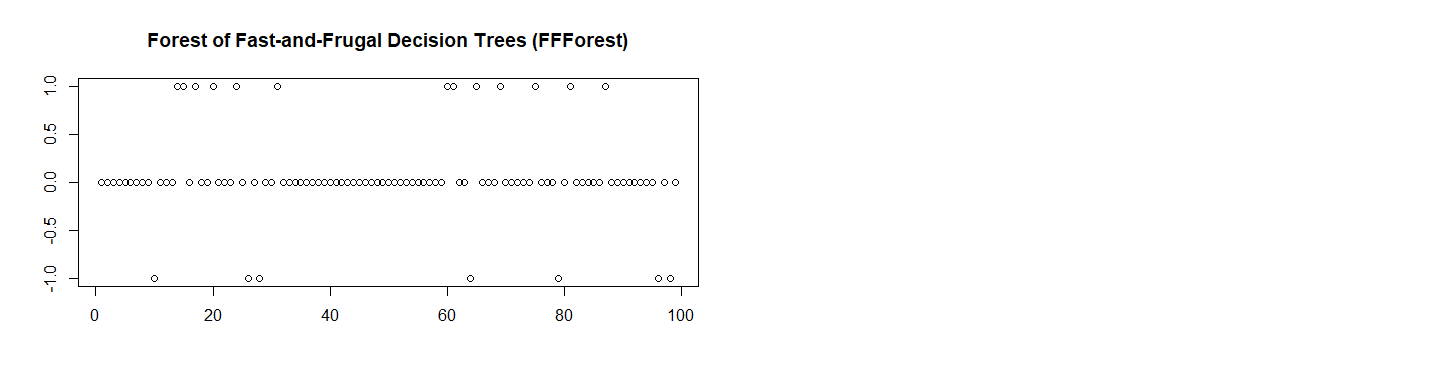
*# RMSE Resulting*  
(rmse\_fff <- **sqrt**(**mean**((**as.numeric**(test\_pred\_fff)**-as.numeric**(testDataF**$**Disease))**^**2)))

## [1] 0.4495

*# Residuals Plot*  
(r\_fff <- (**as.numeric**(test\_pred\_fff)**-as.numeric**(testDataF**$**Disease)))

## [1] 0 0 0 0 0 0 0 0 0 -1 0 0 0 1 1 0 1 0 0 1 0 0 0  
## [24] 1 0 -1 0 -1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [47] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 -1 1 0 0 0 1  
## [70] 0 0 0 0 0 1 0 0 0 -1 0 1 0 0 0 0 0 1 0 0 0 0 0  
## [93] 0 0 0 -1 0 -1 0

**plot**(r\_fff, main = model.name,xlab = "", ylab = "")  
  
*# Let's check the probabilities and Calculate the ROC*  
test\_pred\_fff\_prob <- **predict**(fff\_tree, data = testDataF, type ="prob")[,2]  
  
res.fff.roc <- **roc**(**as.numeric**(testDataF**$**Disease),  
 **as.numeric**(  
 **as.matrix**(test\_pred\_fff\_prob)))**$**auc  
  
*# Store the Values for Report*  
results\_all <- **rbind**(results\_all,  
 **data.frame**("Method"= model.name,  
 "Prediction Accuracy in Training Set" =   
 **max**(fff\_tree**$**tree.stats**$**train**$**acc,na.rm = TRUE),  
 "Prediction Accuracy in Test Set" =   
 res.fff**$**overall[1],  
 "RMSE Test"= rmse\_fff,  
 "ROC" = res.fff.roc,  
 "Time Elapsed" = et,  
 stringsAsFactors = FALSE))



# 17. Optimizing Cost

############################################  
*# 17. Optimizing Cost*  
model.name <- "Optimizing Cost"  
  
## Specifying Cost  
heart.cue.cost <- **data.frame**("cue" =   
 **c**("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg",  
 "thalach", "exang", "oldpeak", "slope", "ca", "thal"),  
 "cost" = **c**(1, 1, 1, 1, 7.27, 5.2, 15.5, 102.9, 87.3, 87.3,   
 87.3, 100.9, 102.9))  
*# Specify the following costs for heart disease diagnosis:*  
*# cost(Hit) = 0, cost(False Alarm) = 100, cost(Miss) = 200, cost(correct rejection) = 0*  
heart.cost.outcomes <- **c**(0, 500, 1000, 0)  
  
heart.costA.fft <- **FFTrees**(formula = Disease **~**.,  
 data = **rbind**(trainDataF,testDataF),  
 cost.outcomes = heart.cost.outcomes,  
 cost.cues = heart.cue.cost,  
 goal = "bacc",  
 goal.chase = "bacc")

## Growing FFTs with ifan

## Fitting non-FFTrees algorithms for comparison (you can turn this off with do.comp = FALSE) ...

**summary**(heart.costA.fft)**$**train[1,]

## tree n hi mi fa cr sens spec ppv npv far acc bacc wacc  
## 1 1 297 137 0 10 150 1 0.9375 0.932 1 0.0625 0.9663 0.8145 0.9688  
## bpv dprime cost pci mcu  
## 1 0.966 4.198 248.8519 0.89 1.65

heart.costB.fft <- **FFTrees**(formula = Disease **~**.,  
 data = **rbind**(trainDataF,testDataF),  
 cost.outcomes = heart.cost.outcomes,  
 cost.cues = heart.cue.cost,  
 goal = "cost",  
 goal.chase = "cost")

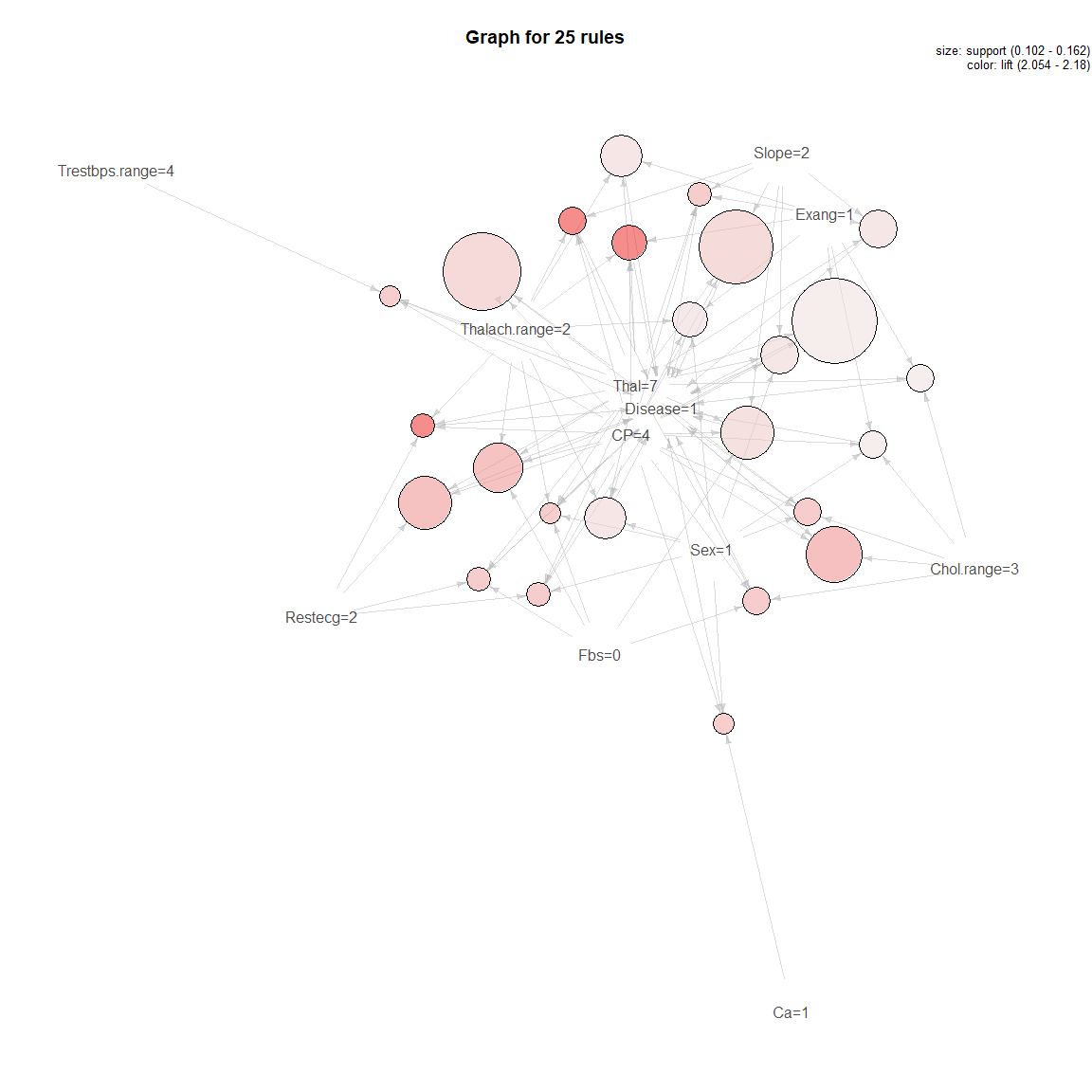
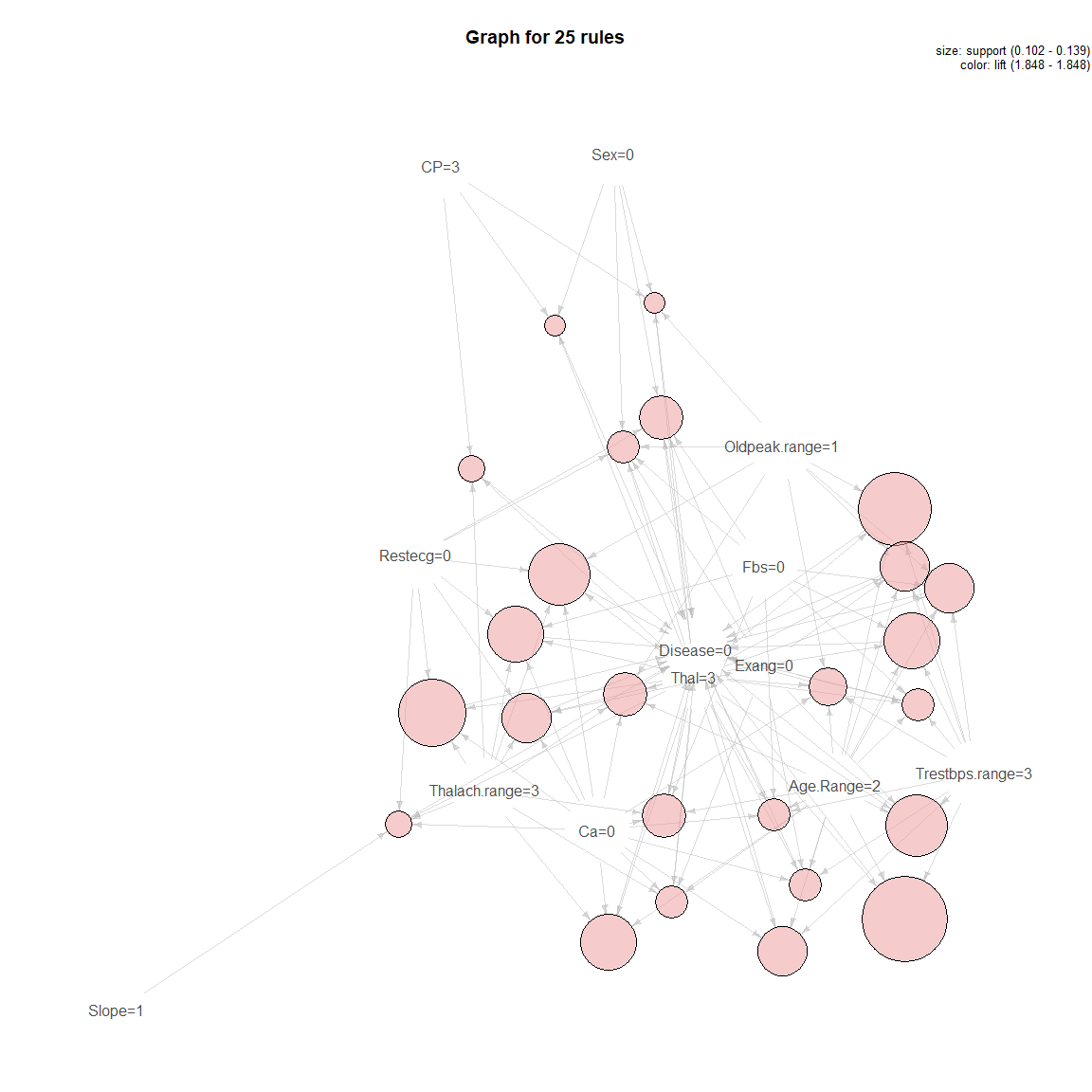
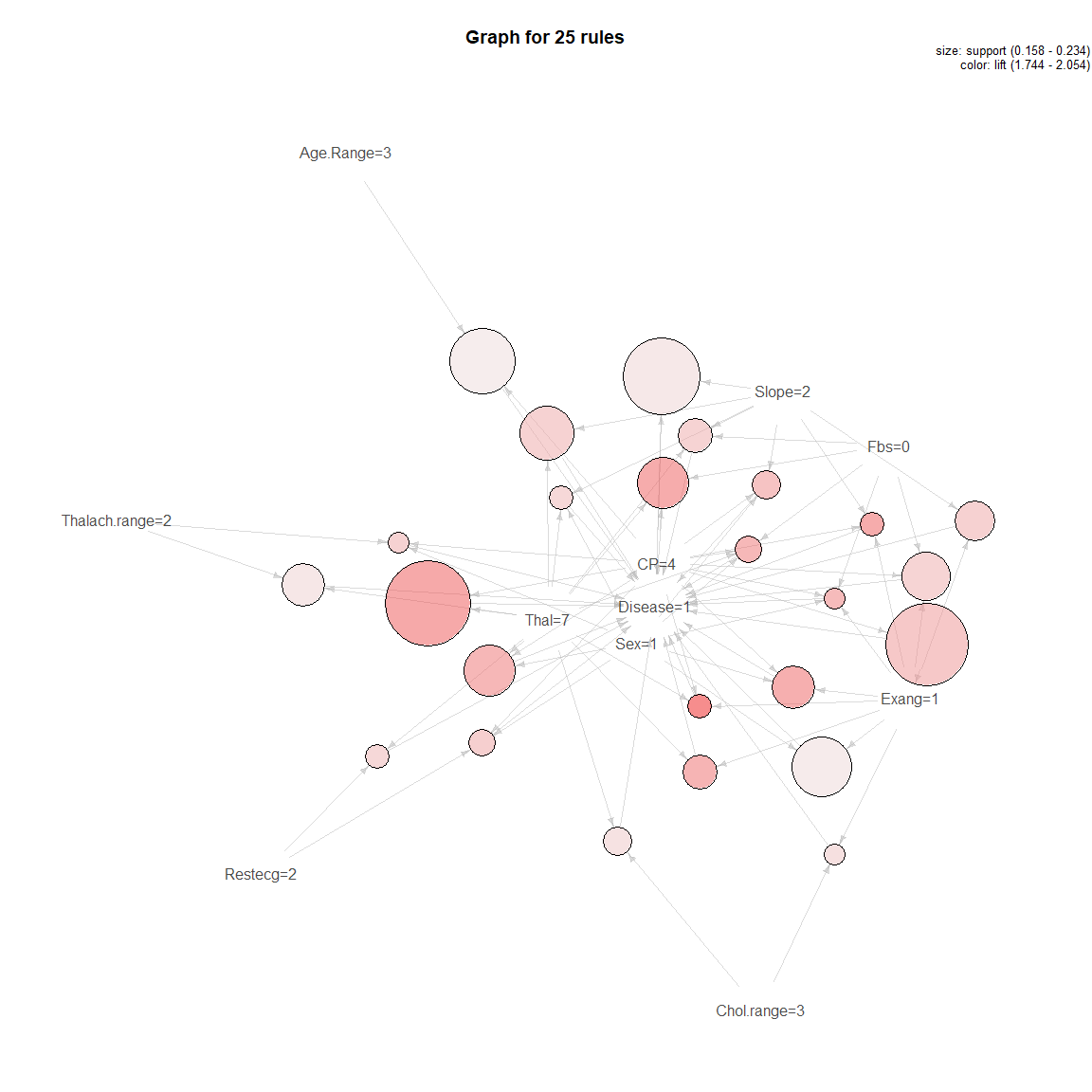
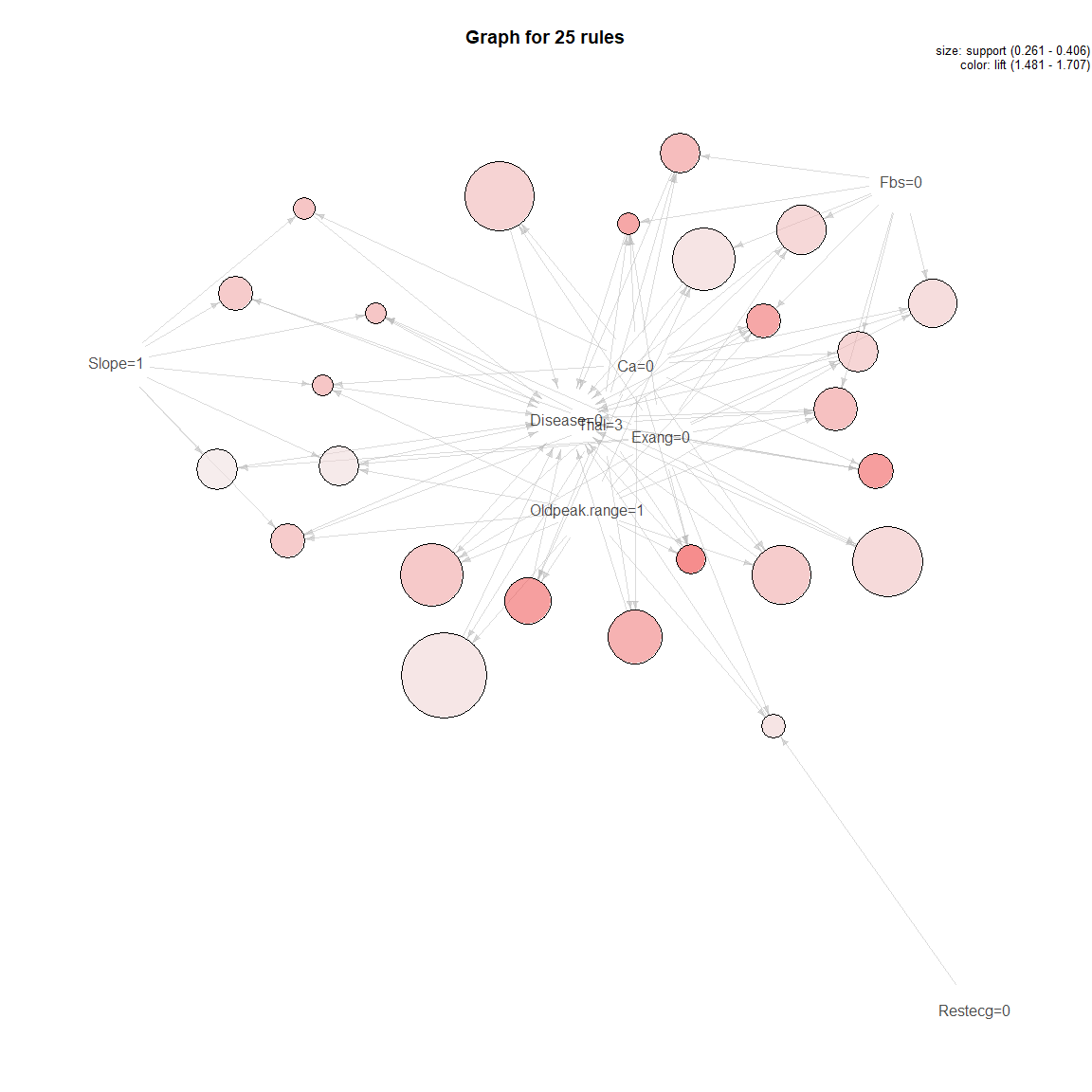
## Growing FFTs with ifan  
## Fitting non-FFTrees algorithms for comparison (you can turn this off with do.comp = FALSE) ...

**summary**(heart.costB.fft)**$**train[1,]

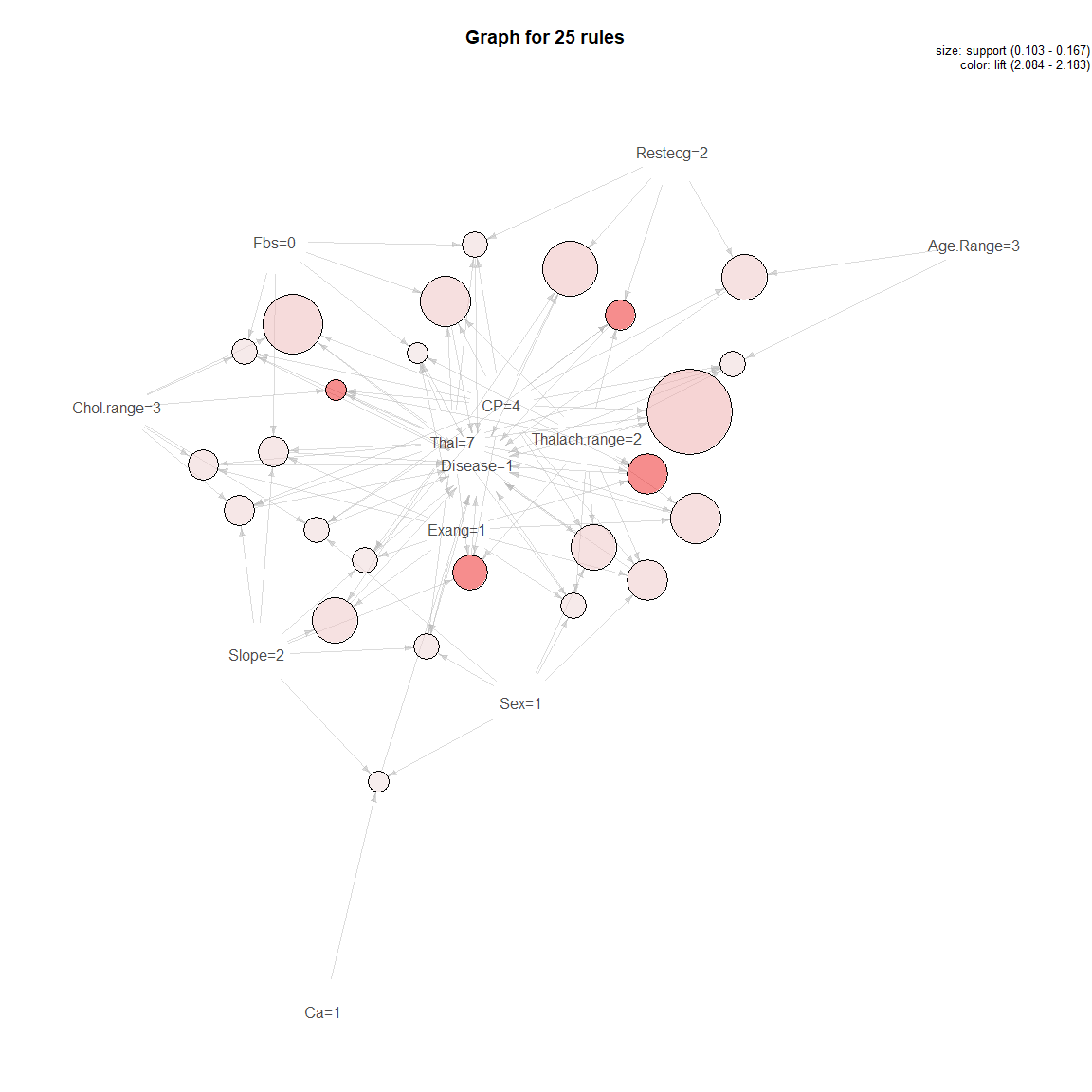
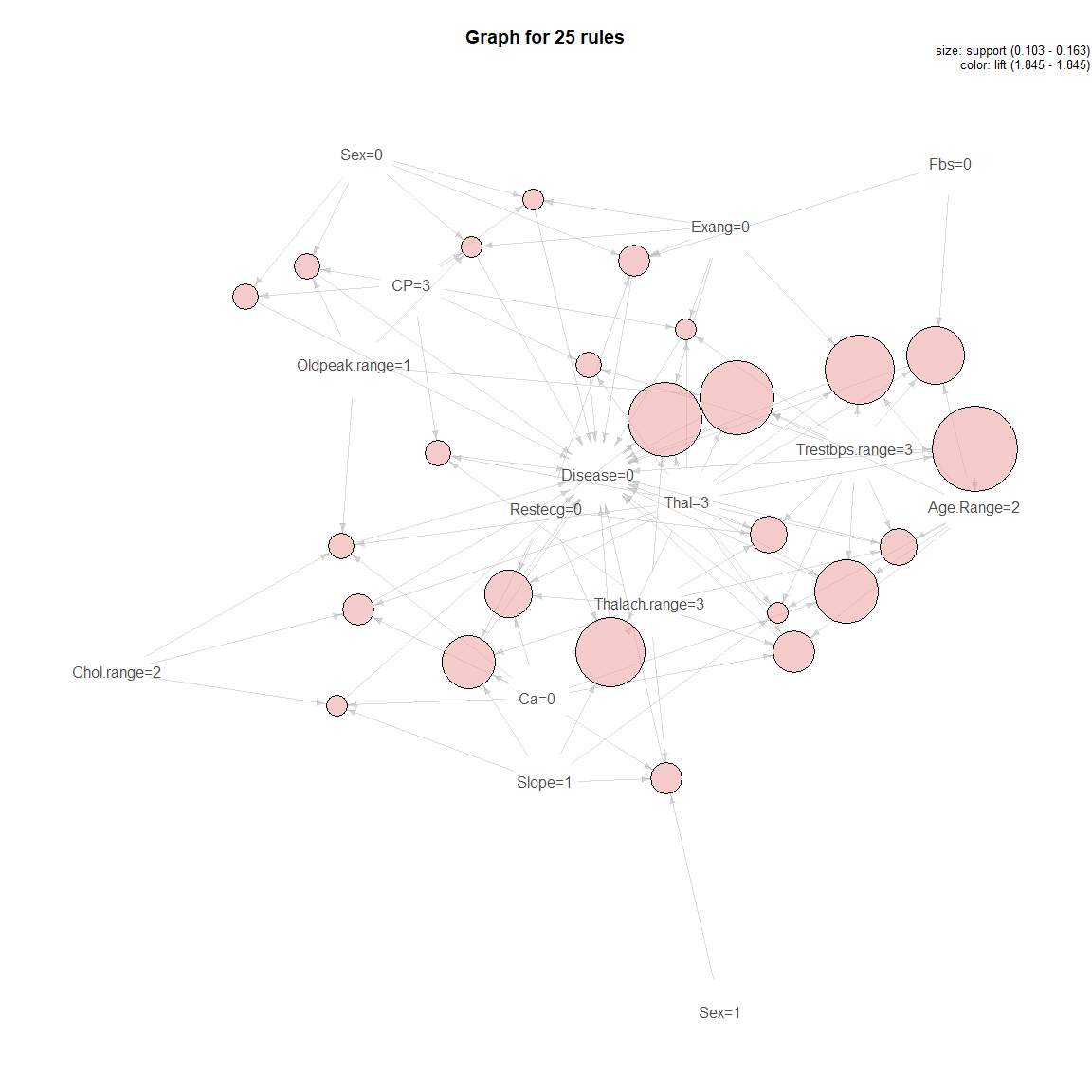
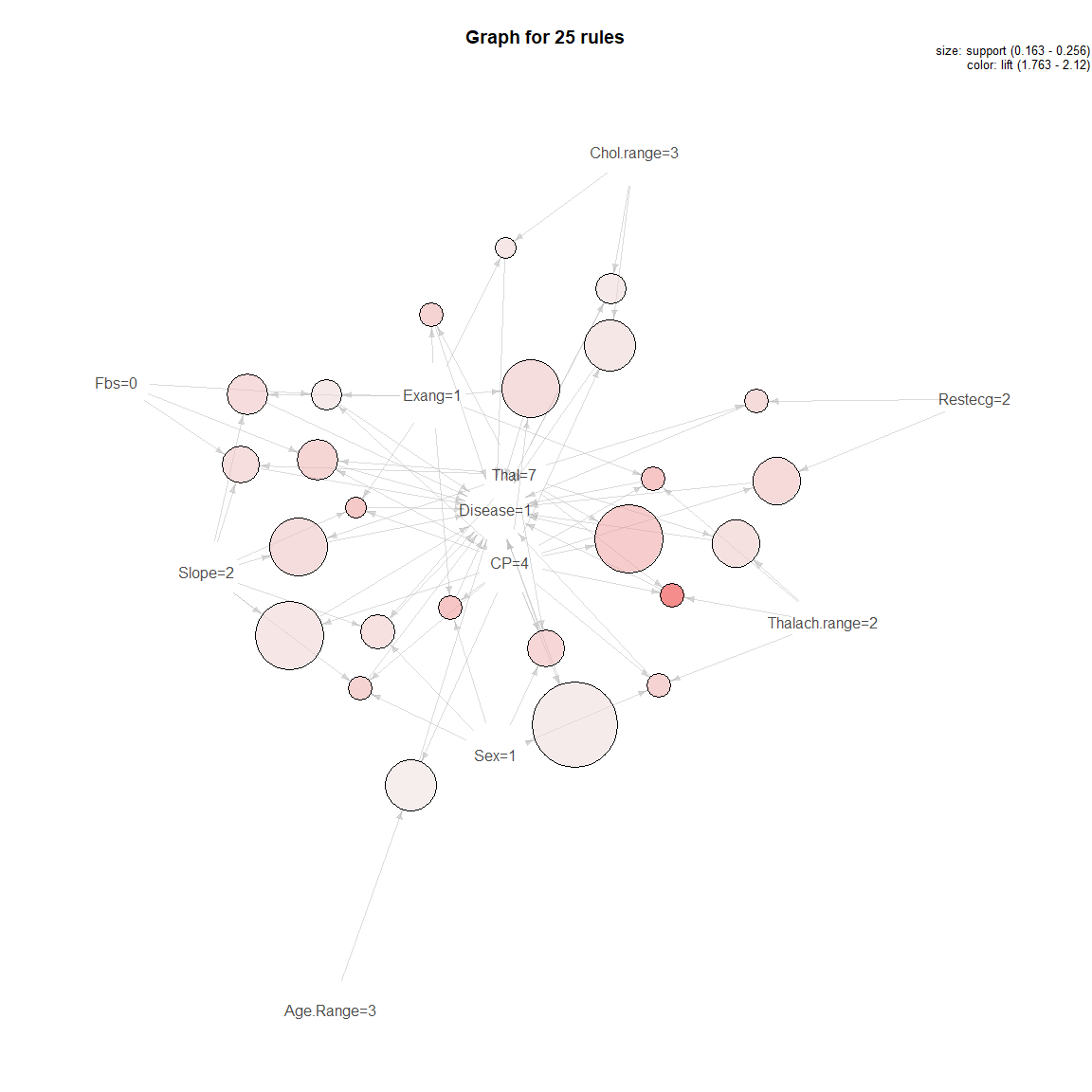
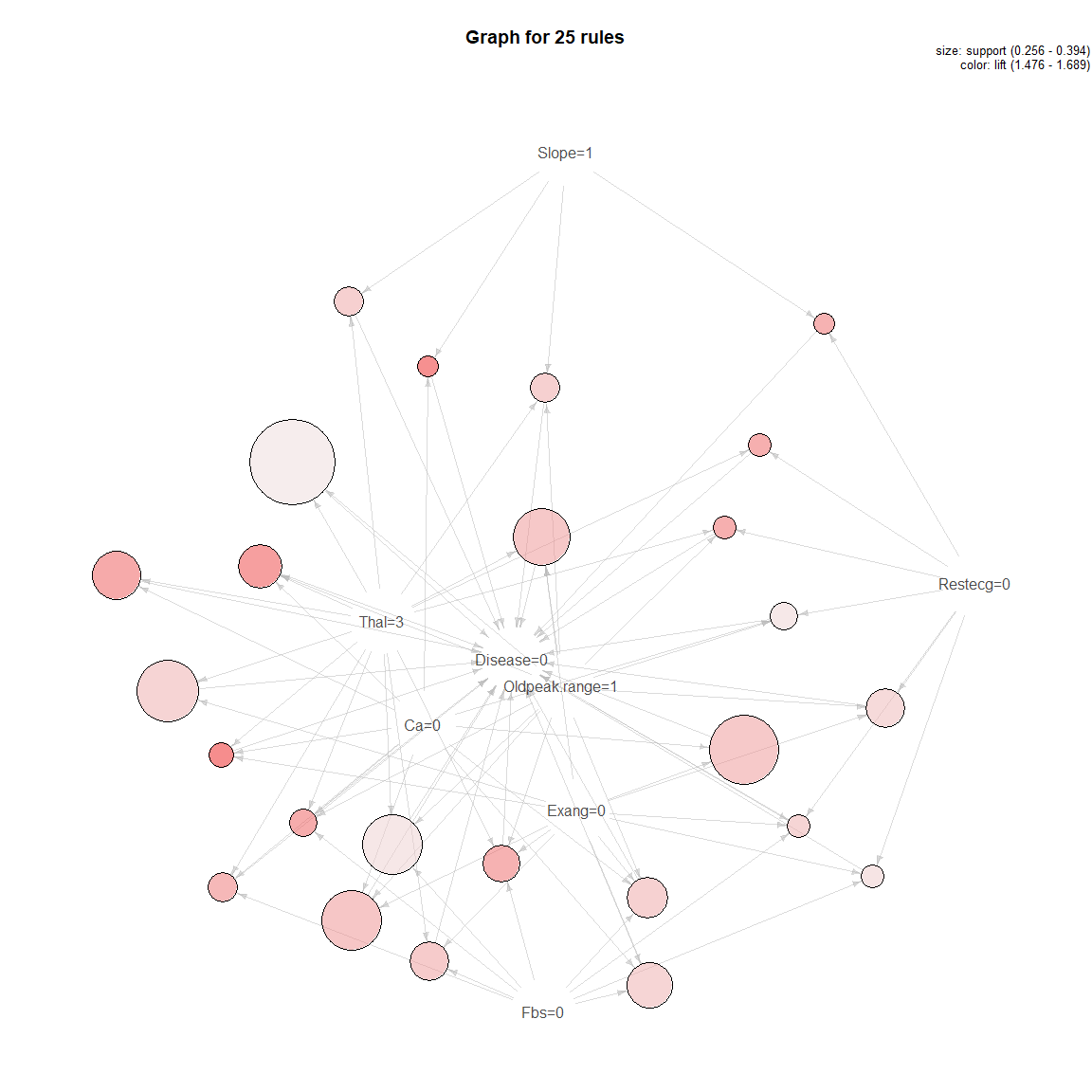
## tree n hi mi fa cr sens spec ppv npv far acc bacc wacc  
## 1 1 297 137 0 10 150 1 0.9375 0.932 1 0.0625 0.9663 0.7318 0.9688  
## bpv dprime cost pci mcu  
## 1 0.966 4.198 151.5152 0.8887 1.67

# 18. Results

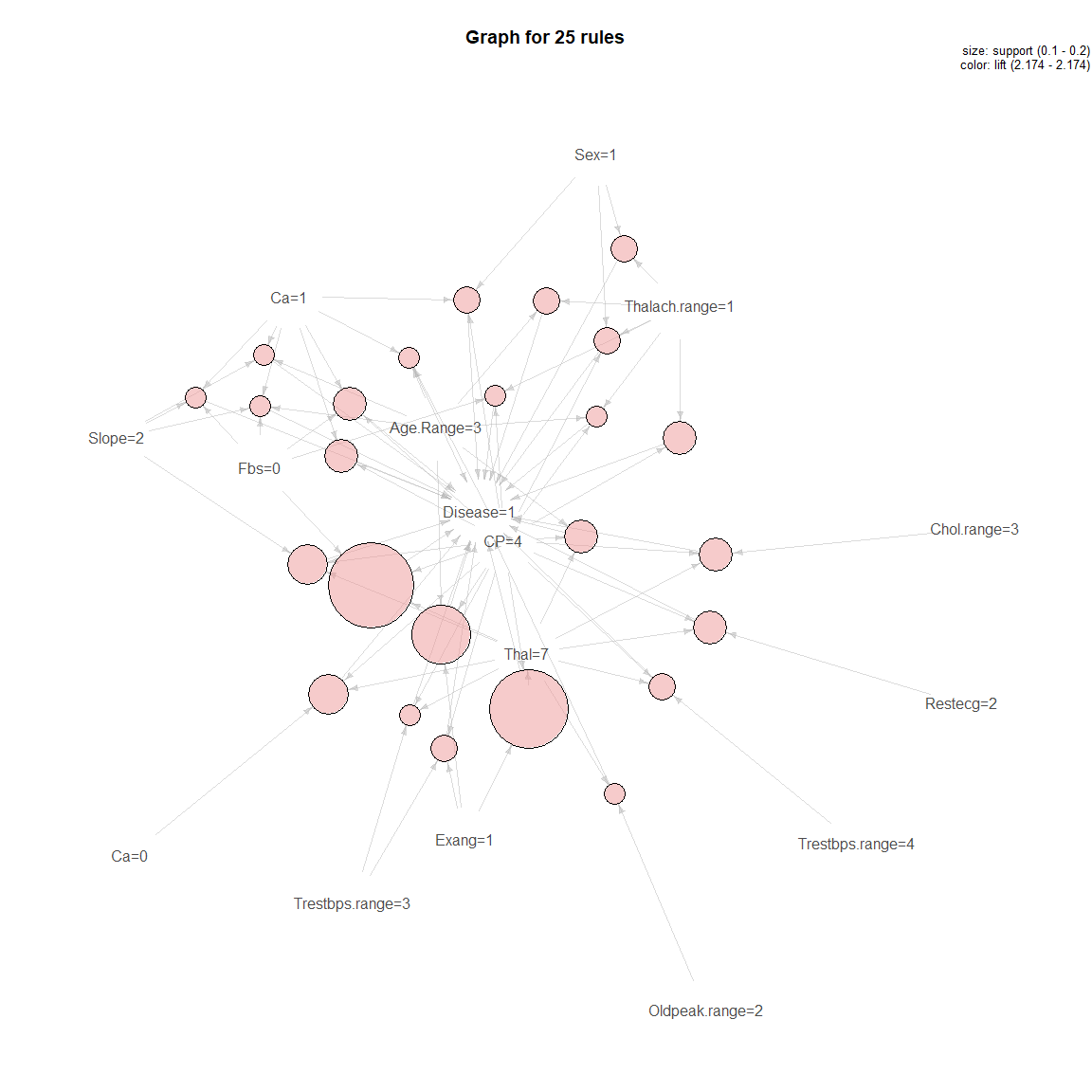
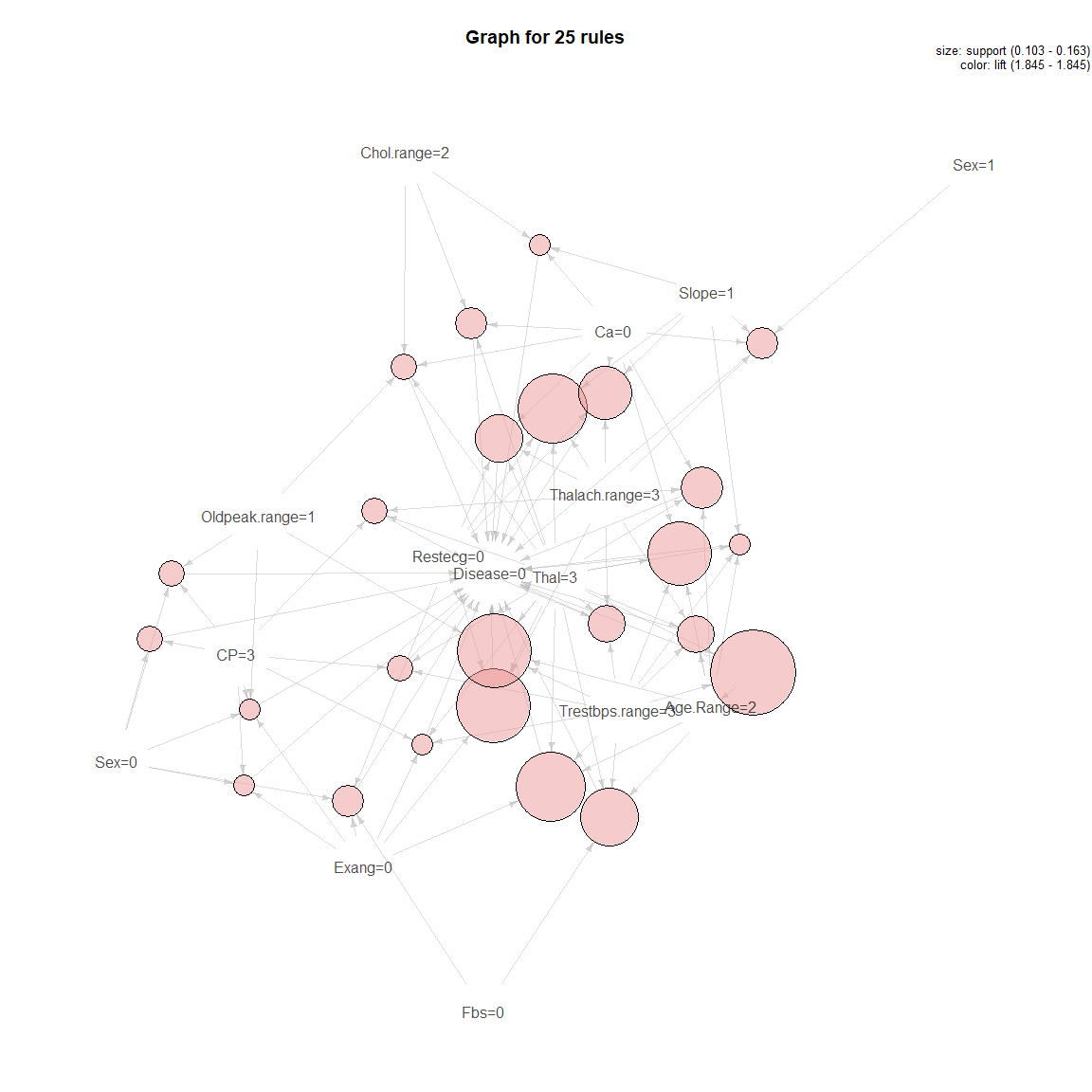
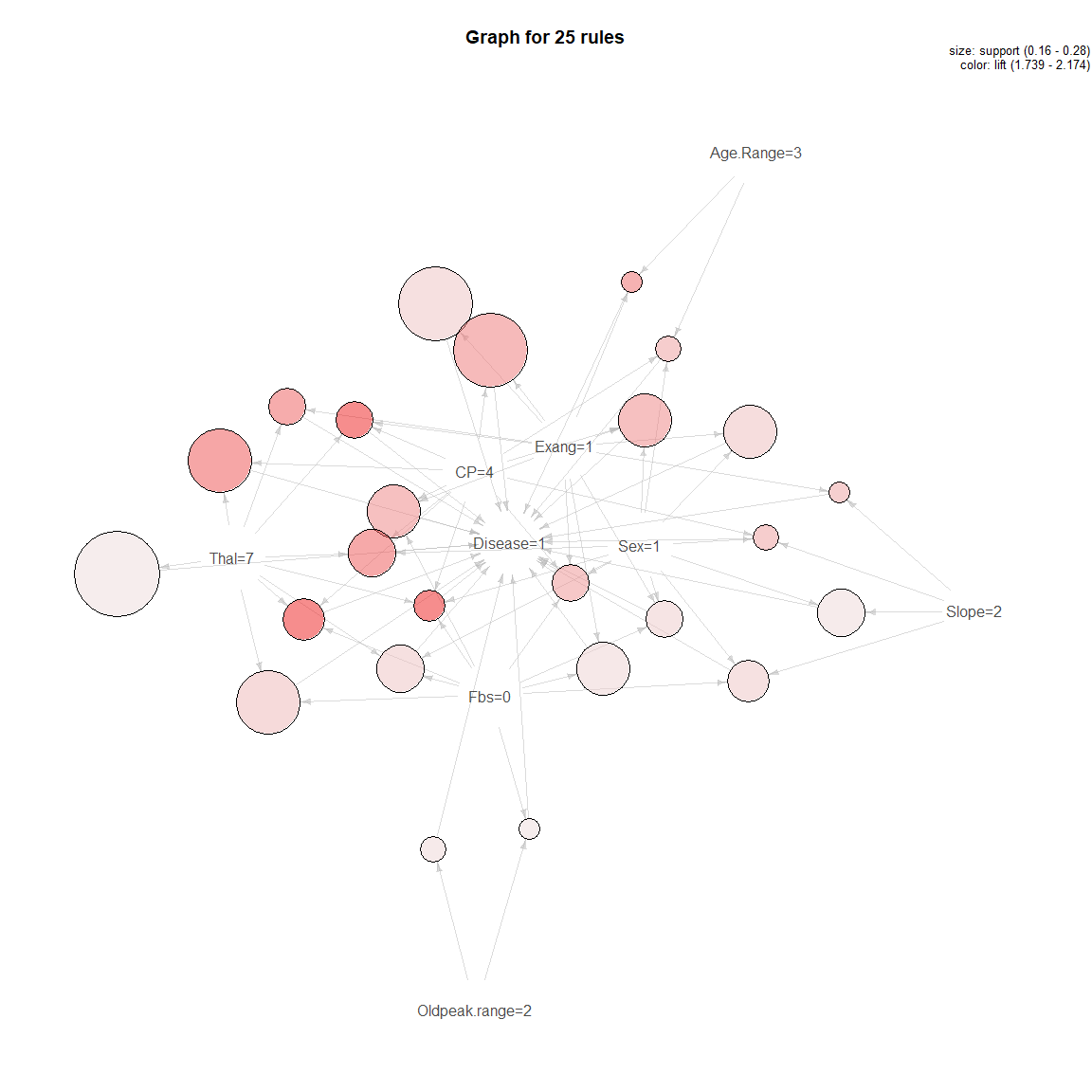
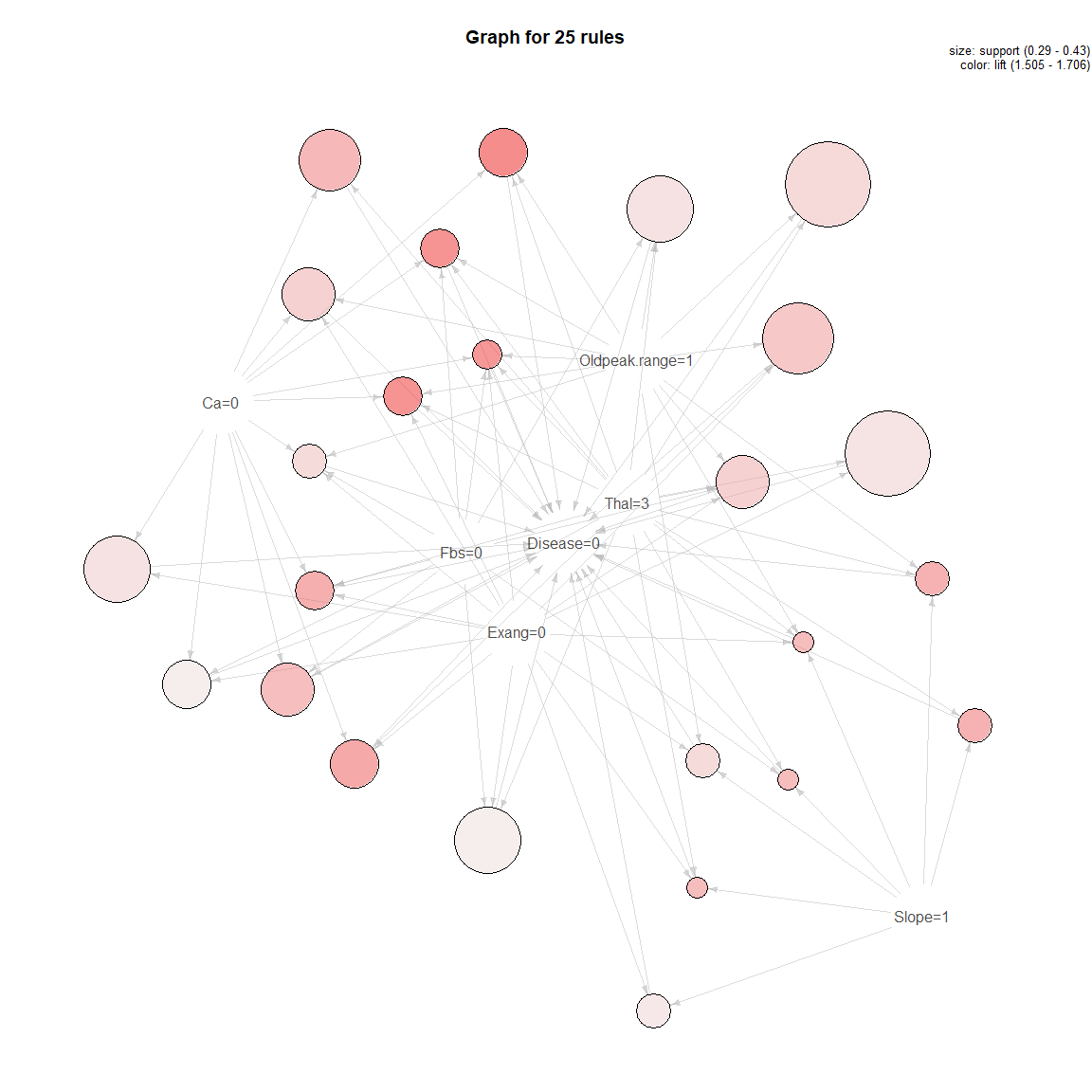
## 18.1 Mining Association Rules



### Train Set



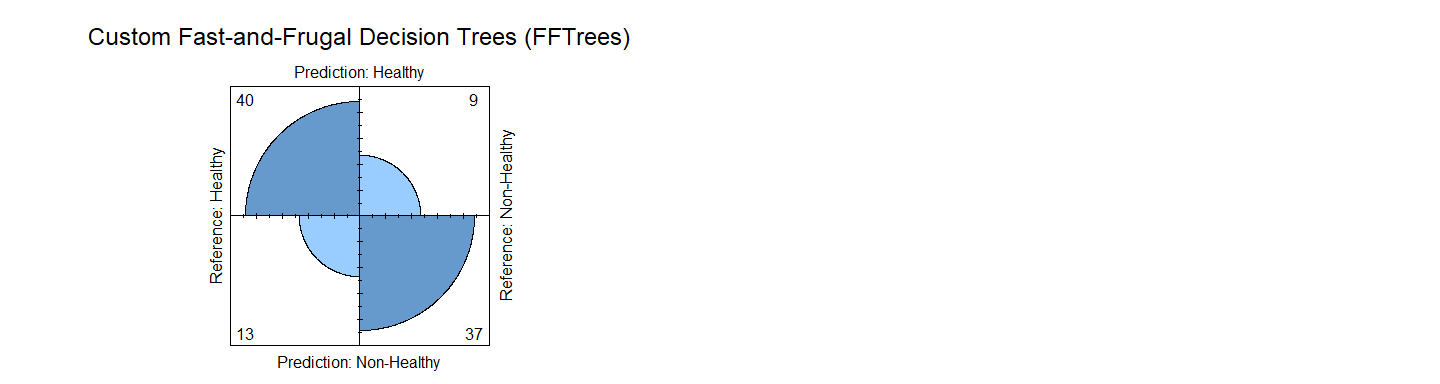
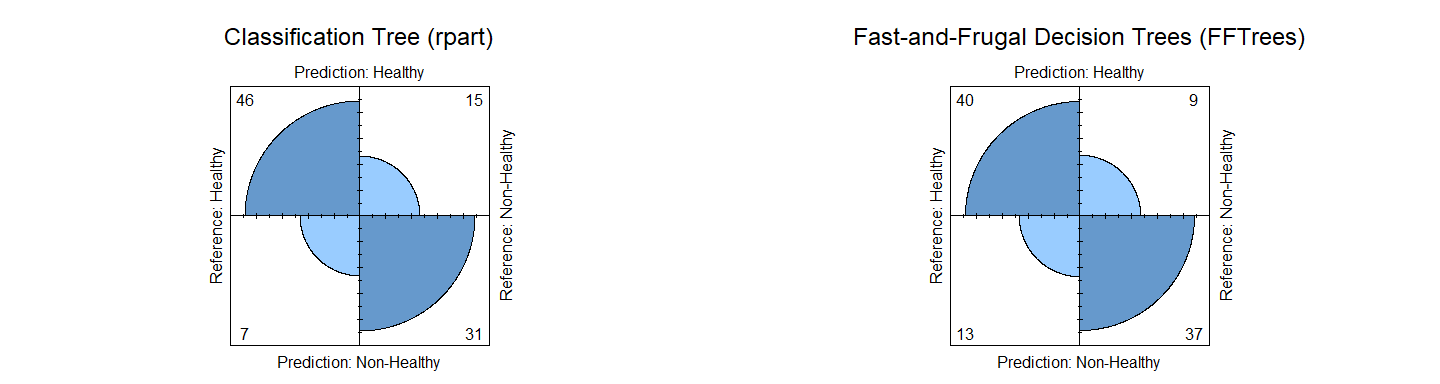
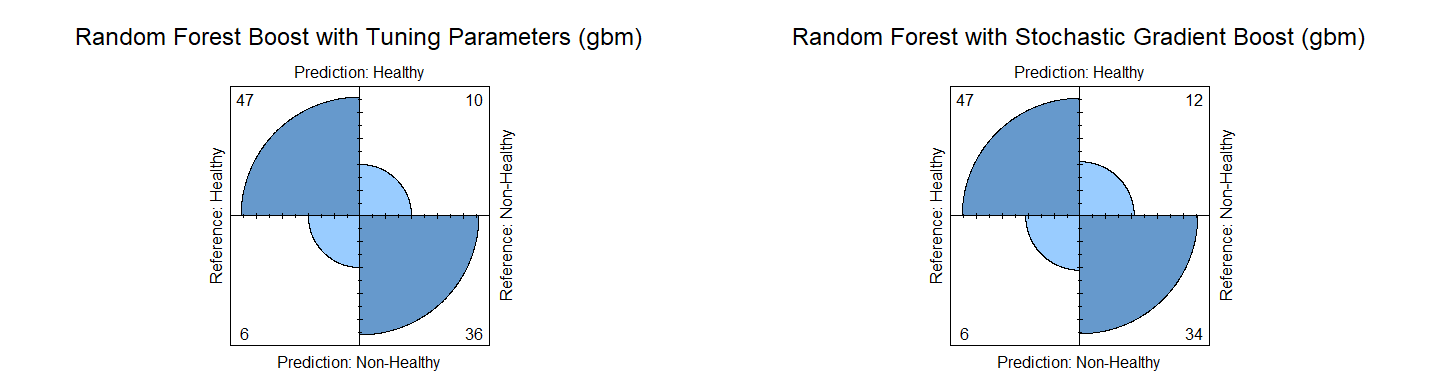
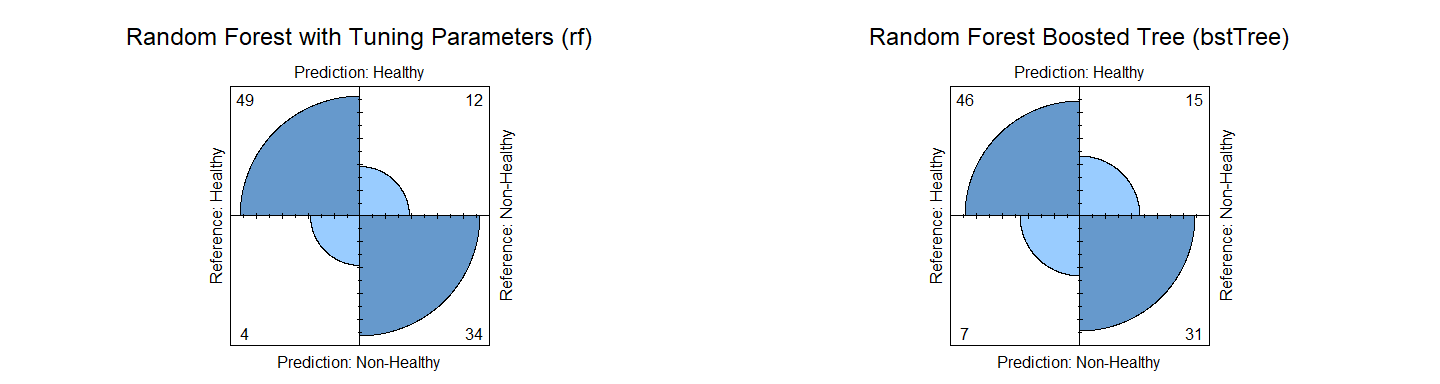
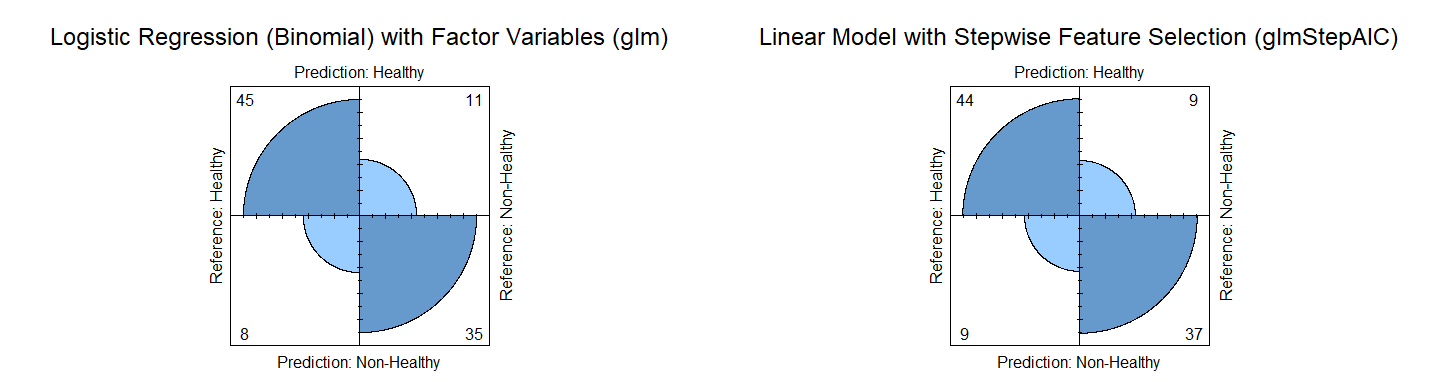
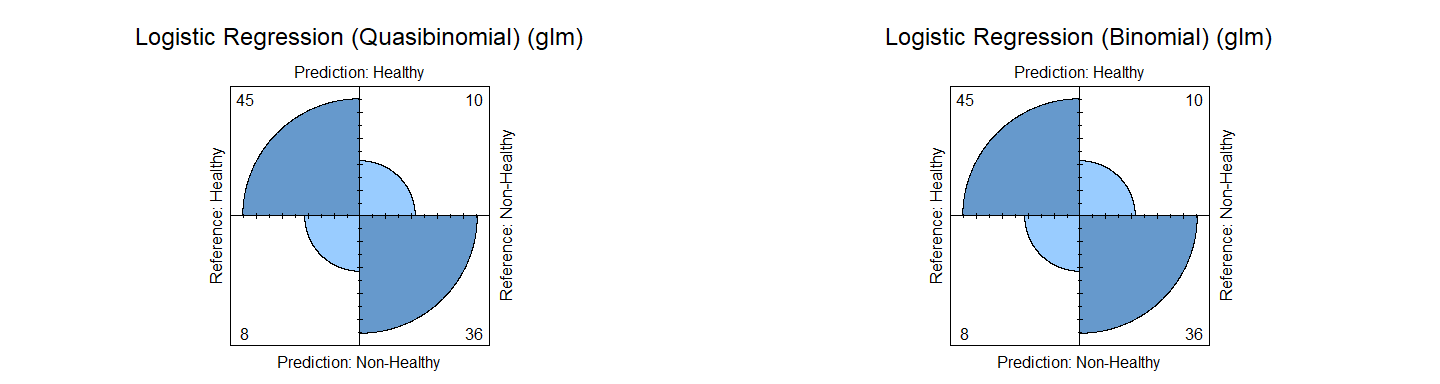
### Test Set



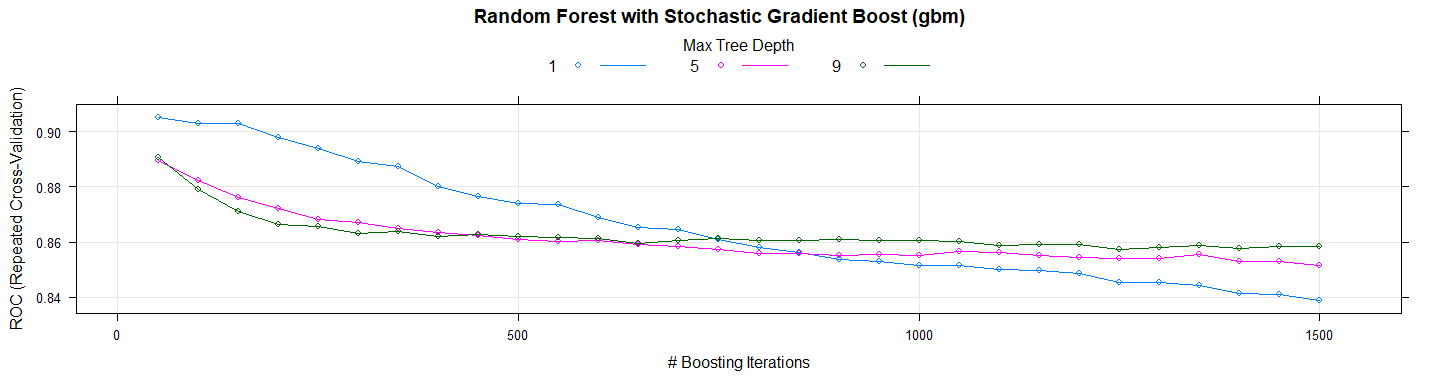
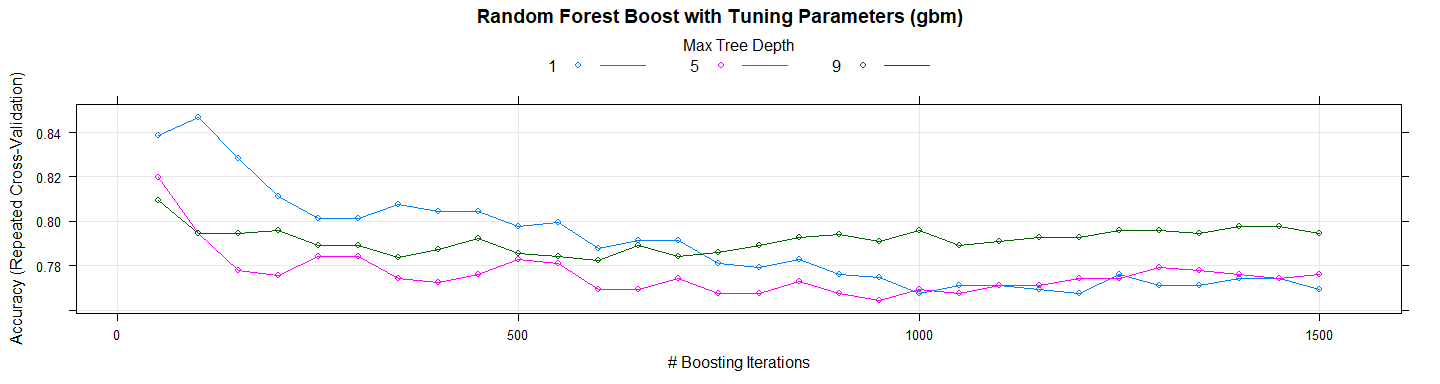
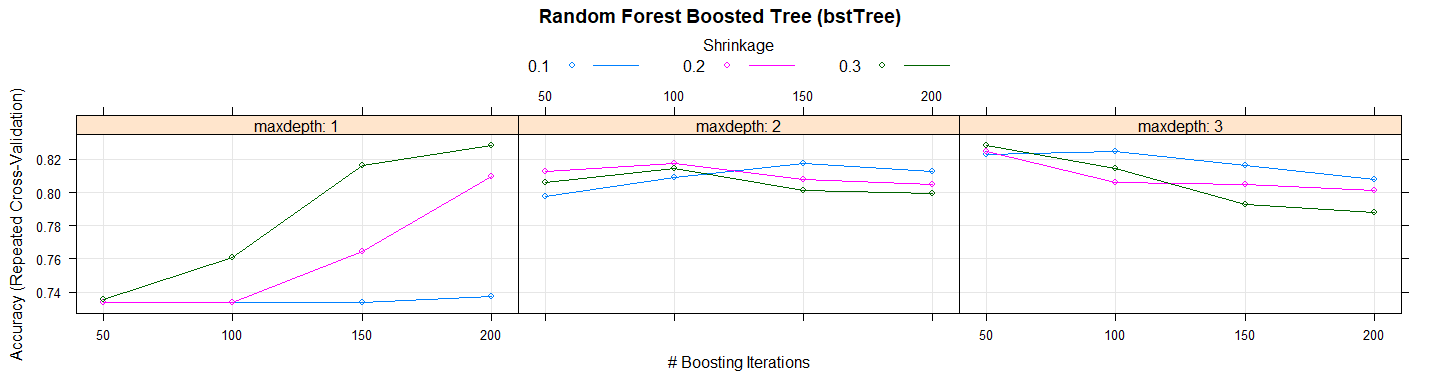
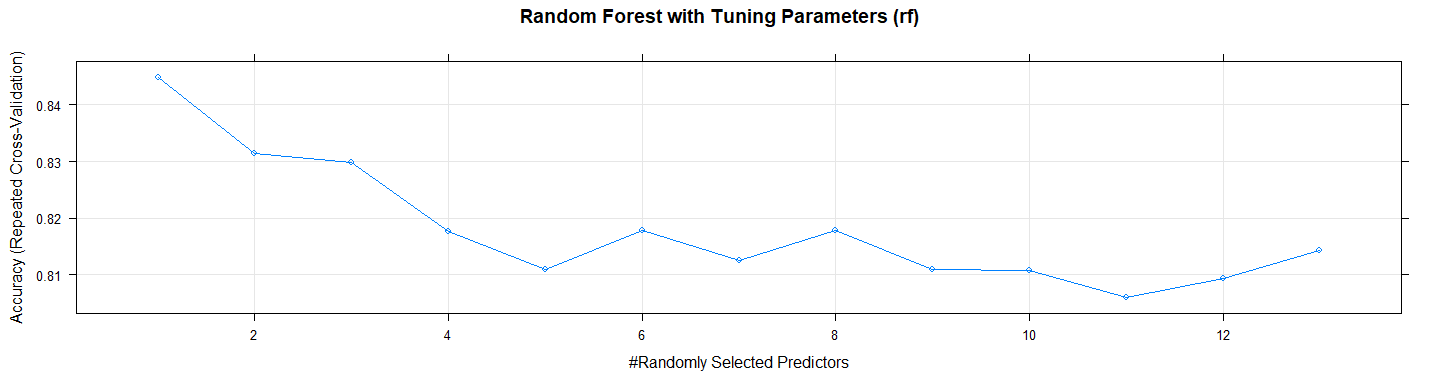
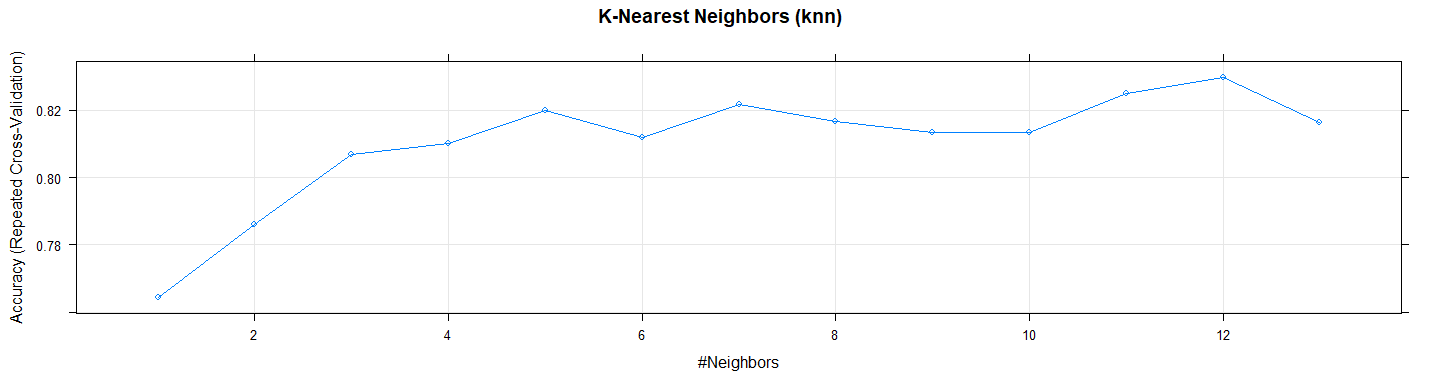
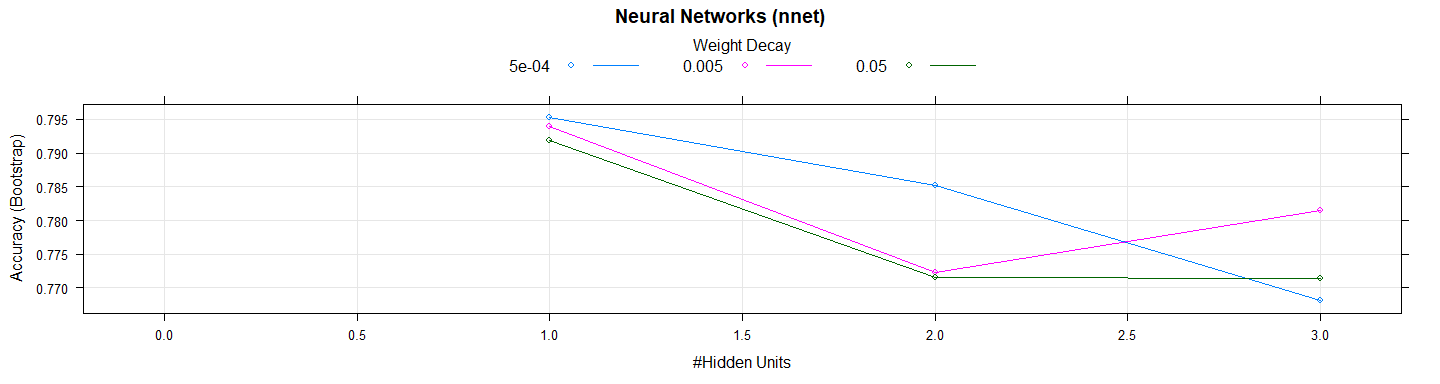
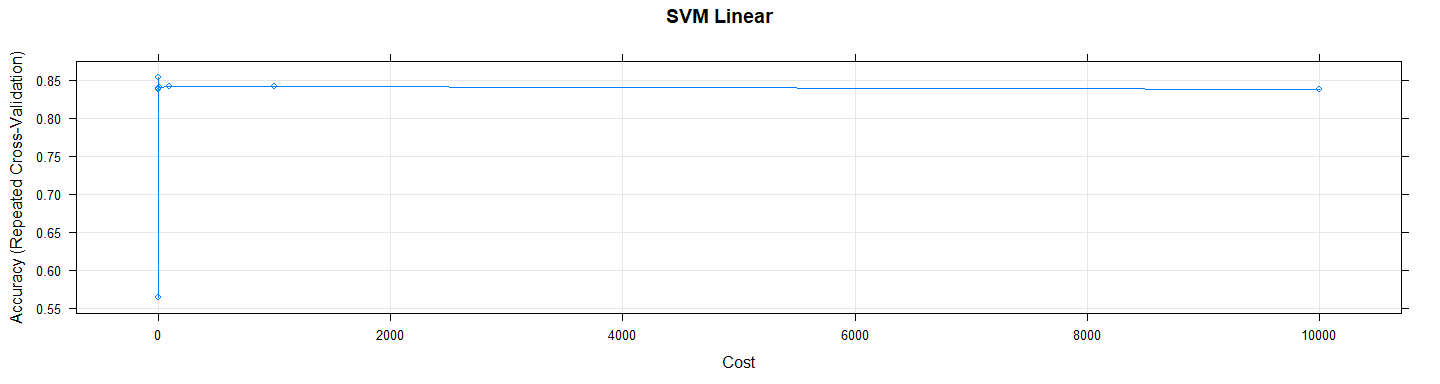
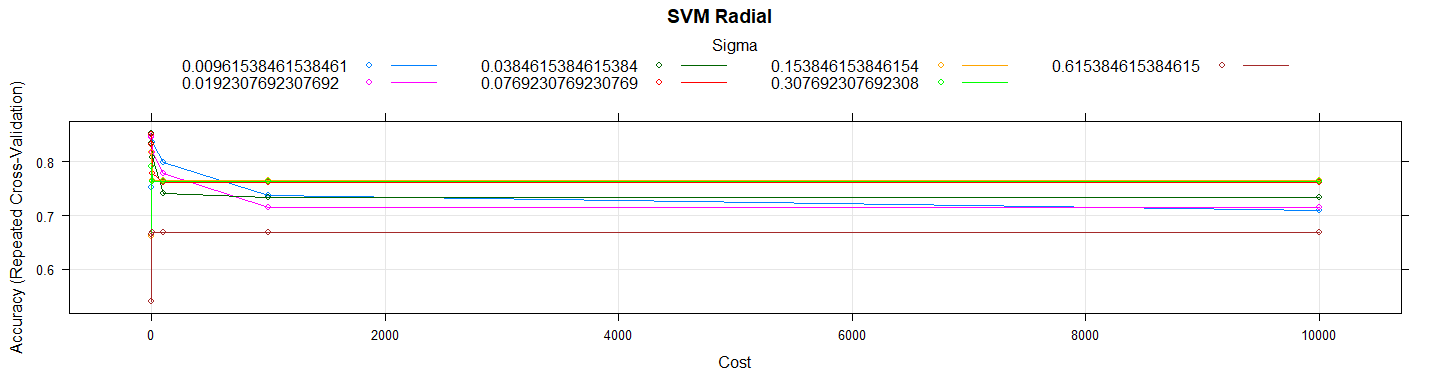
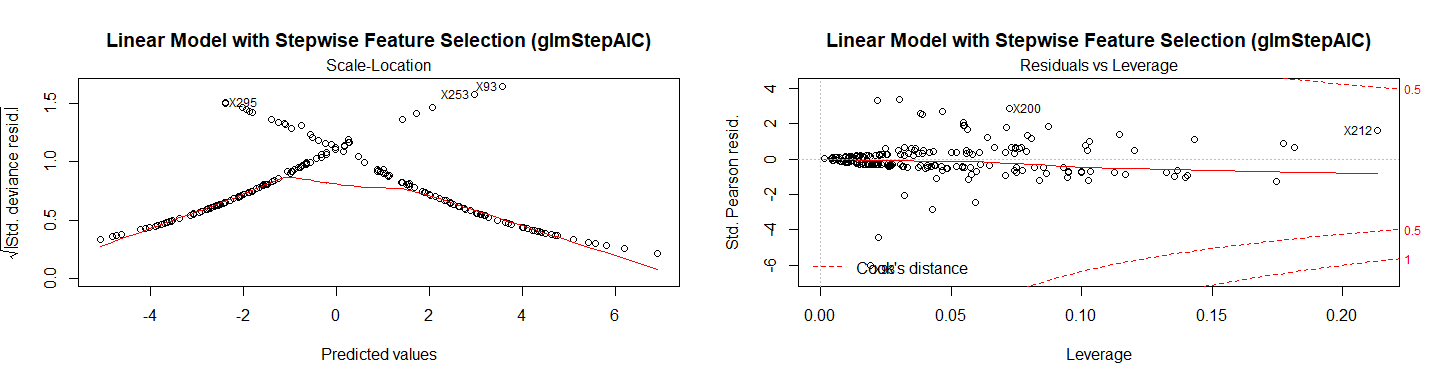
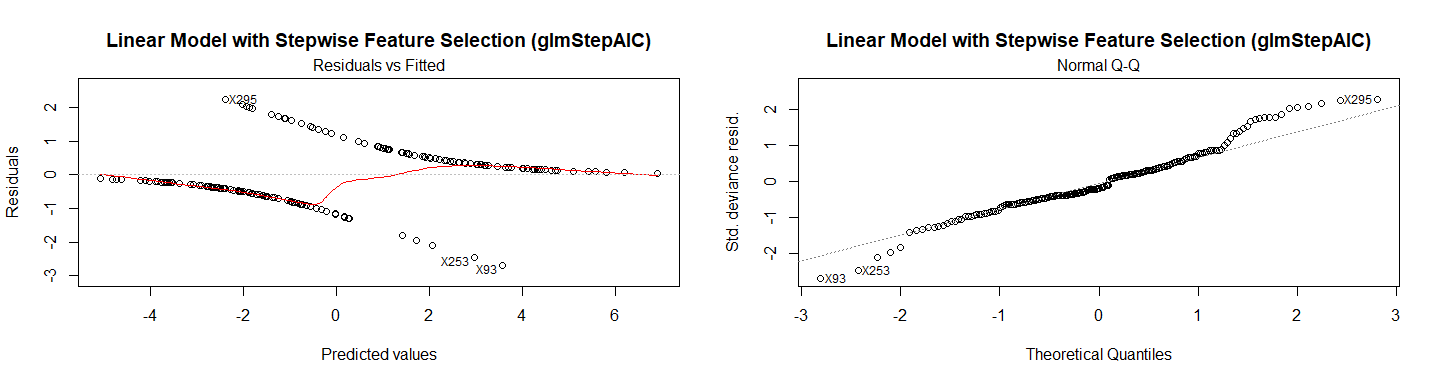
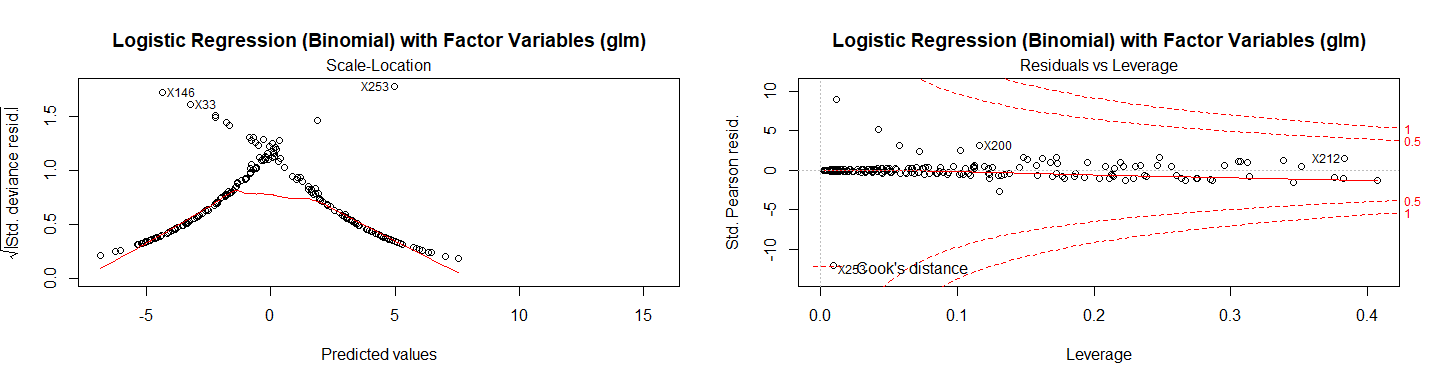
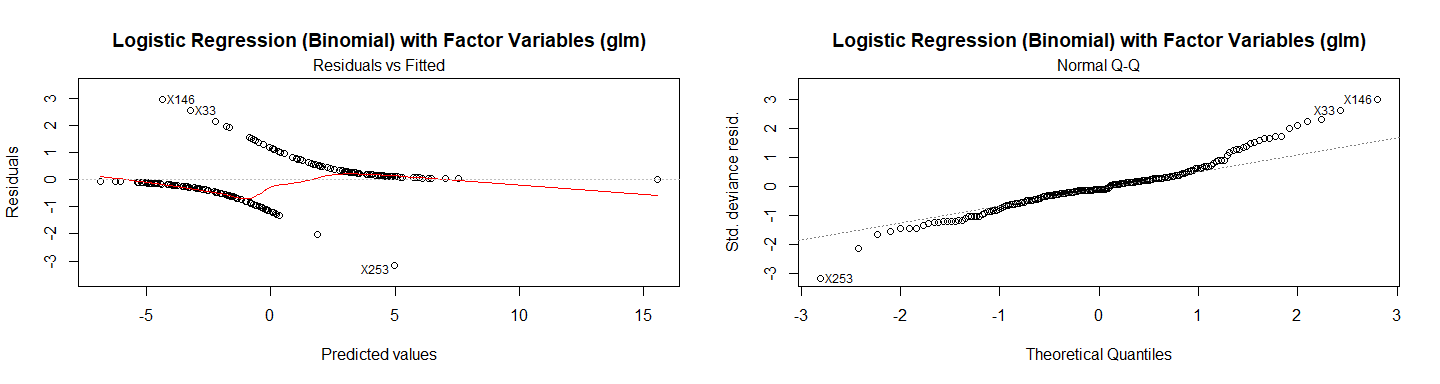
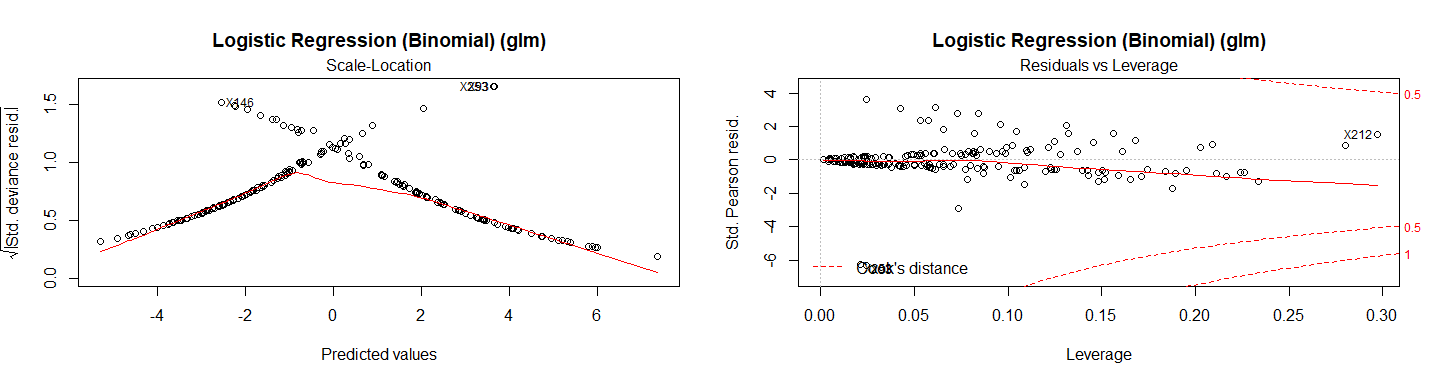
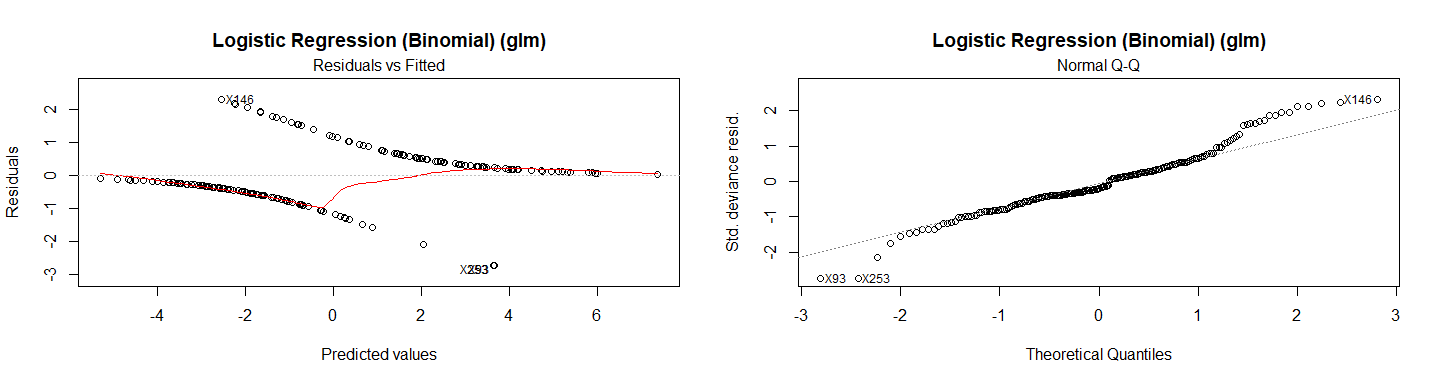
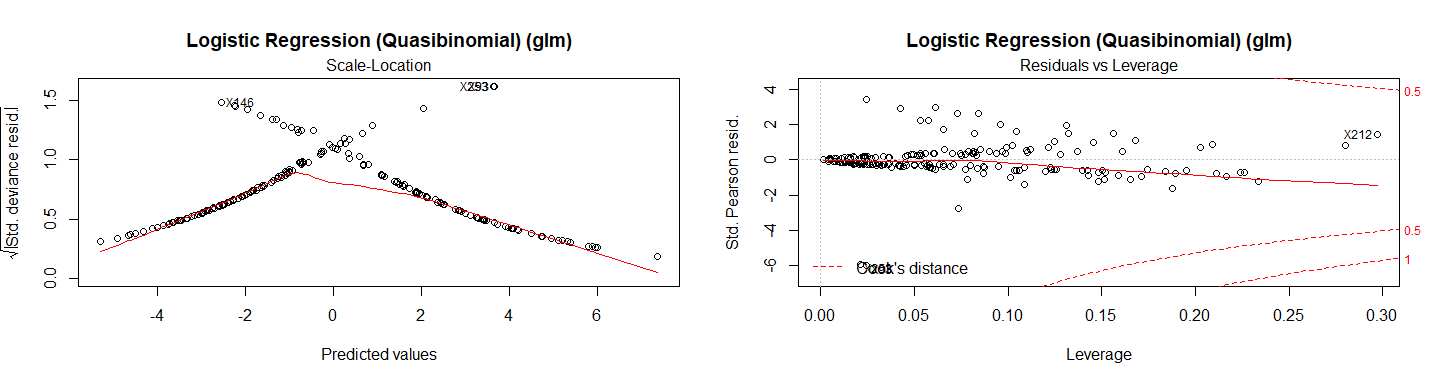
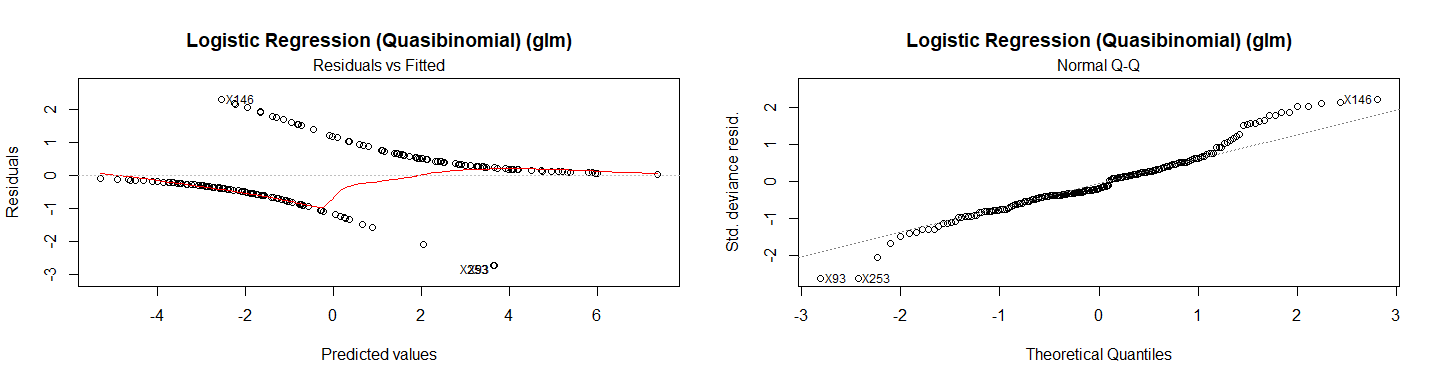
## 17.2 All Prediction Models Results

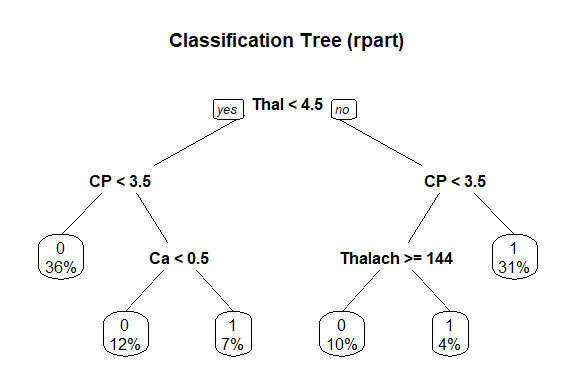
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Prediction Accuracy in Training Set | Prediction Accuracy in Test Set | RMSE Test | ROC | Time Elapsed |
| Logistic Regression (Quasibinomial) (glm) | 0.8132 | 0.8182 | 0.4264 | 0.9036 | 1.3 |
| Logistic Regression (Binomial) (glm) | 0.8021 | 0.8182 | 0.4264 | 0.9036 | 1.4 |
| Logistic Regression (Binomial) with Factor Variables (glm) | 0.8009 | 0.8081 | 0.4381 | 0.8720 | 1.2 |
| Linear Model with Stepwise Feature Selection (glmStepAIC) | 0.8076 | 0.8182 | 0.4264 | 0.8925 | 5.3 |
| Kmeans PCA (8) | 0.8333 | 0.7980 | 0.4495 | 0.7941 | 5.8 |
| Kmeans (13) | 0.8333 | 0.8182 | 0.4264 | 0.8158 | 4.5 |
| SVM Radial | 0.8535 | 0.7980 | 0.4495 |  | 58 |
| SVM Linear | 0.8586 | 0.8182 | 0.4264 |  | 169 |
| Neural Networks (nnet) | 0.7953 | 0.7778 | 0.4714 | 0.8376 | 9.5 |
| K-Nearest Neighbors (knn) | 0.8299 | 0.8485 | 0.3892 | 0.9100 | 6.9 |
| Random Forest with Tuning Parameters (rf) | 0.8449 | 0.8384 | 0.4020 | 0.9182 | 52 |
| Random Forest Boosted Tree (bstTree) | 0.8283 | 0.7778 | 0.4714 |  | 425 |
| Random Forest Boost with Tuning Parameters (gbm) | 0.8469 | 0.8384 | 0.4020 | 0.8897 | 29 |
| Random Forest with Stochastic Gradient Boost (gbm) | 0.9053 | 0.8182 | 0.4264 | 0.8819 | 96 |
| Classification Tree (rpart) |  | 0.7778 | 0.4714 | 0.8117 | 96 |
| Fast-and-Frugal Decision Trees (FFTrees) | 0.8283 | 0.7778 | 0.4714 | 0.7914 | 96 |
| Custom Fast-and-Frugal Decision Trees (FFTrees) | 0.8283 | 0.7778 | 0.4714 | 0.7914 | 96 |
| Forest of Fast-and-Frugal Decision Trees (FFForest) | 0.8283 | 0.7980 | 0.4495 | 0.8273 | 127 |

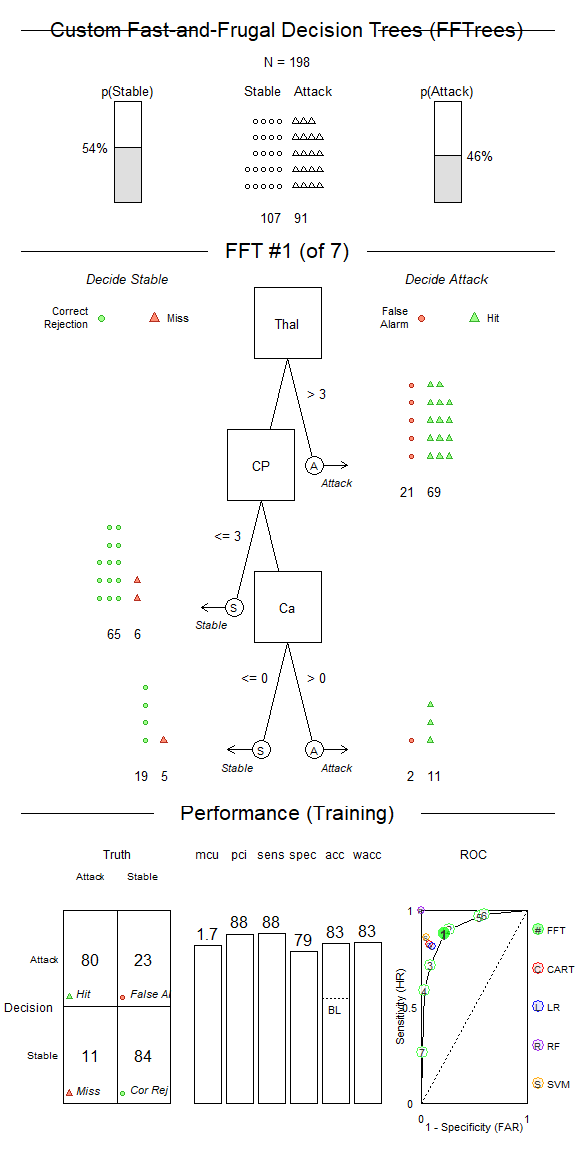
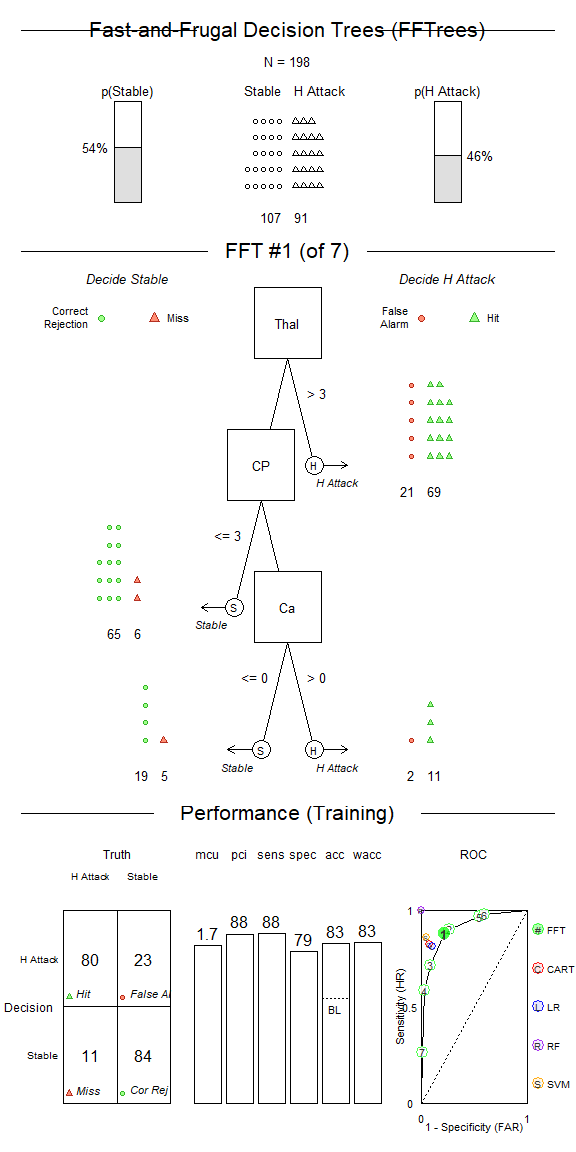
### Confusion Matrix (Test Set)



### Models Accuracy







### Models Residuals

