CKME 136 Capstone

TODD BETHELL

July 7, 2019

# !!NOTE!! This will install ALL of the caret package - which will take several minutes - so comment the install out if you do not want to install the full caret package.

install.packages(“caret”, dependencies = c(“Depends”, “Suggests”)) install.packages(“PerformanceAnalytics”) install.packages(“lsr”) install.packages(“tidyverse”) install.packages(“dplyr”) install.packages(“class”) install.packages(“gmodels”) install.packages(“corrplot”) install.packages(“RCurl”) install.packages(“GGally”) install.packages(“FSelector”) install.packages(“FSelectorRcpp”) install.packages(“mlbench”) install.packages(“caretEnsemble”) library(GGally) library(knitr) library(RCurl) library(caret) library(PerformanceAnalytics) library(lsr) library(tidyverse) library(dplyr) library(class) library(gmodels) library(corrplot) library(GGally) library(FSelectorRcpp) library(caretEnsemble) —

library(knitr)  
library(RCurl)  
library(caret)  
library(PerformanceAnalytics)  
library(lsr)  
library(tidyverse)  
library(dplyr)  
library(plyr)  
library(class)  
library(gmodels)  
library(corrplot)  
library(GGally)  
library(FSelectorRcpp)  
library(caretEnsemble)

FullSetB4<-read.csv(file="https://raw.githubusercontent.com/ToddB11/TB136capstone/master/ZAlizadeh\_dataset.csv",header=T,sep=",")

# Review Data Set

head(FullSetB4)

## Ã¯..Age Weight Length Sex BMI DM HTN Smoker ExSmoker FH Obesity  
## 1 53 90 175 Male 29.38776 0 1 1 0 0 Y  
## 2 67 70 157 Fmale 28.39872 0 1 0 0 0 Y  
## 3 54 54 164 Male 20.07733 0 0 1 0 0 N  
## 4 66 67 158 Fmale 26.83865 0 1 0 0 0 Y  
## 5 50 87 153 Fmale 37.16519 0 1 0 0 0 Y  
## 6 50 75 175 Male 24.48980 0 0 1 0 0 N  
## CRF CVA AD TD CHF DLP BP PR Edema WPP LR SysM DiaM TCP Dyspnea Fclass  
## 1 N N N N N Y 110 80 0 N N N N 0 N 0  
## 2 N N N N N N 140 80 1 N N N N 1 N 0  
## 3 N N N N N N 100 100 0 N N N N 1 N 0  
## 4 N N N N N N 100 80 0 N N N Y 0 Y 3  
## 5 N N N N N N 110 80 0 N N Y N 0 Y 2  
## 6 N N N N N N 118 70 0 N N N N 1 N 3  
## ACP NCP ECP LTAng QWave STelev STdep Tinv LVH PoorR BBB FBS Cr TG LDL  
## 1 N N N N 0 0 1 1 N N N 90 0.7 250 155  
## 2 N N N N 0 0 1 1 N N N 80 1.0 309 121  
## 3 N N N N 0 0 0 0 N N N 85 1.0 103 70  
## 4 N Y N N 0 0 1 0 N N N 78 1.2 63 55  
## 5 N N N N 0 0 0 0 N N N 104 1.0 170 110  
## 6 N N N N 0 0 0 0 N N N 86 1.0 139 119  
## HDL BUN ESR Hb K Na WBC Lymph Neut PLT EF RWMA VHD CAD  
## 1 30 8 7 15.6 4.7 141 5700 39 52 261 50 0 N Cad  
## 2 36 30 26 13.9 4.7 156 7700 38 55 165 40 4 N Cad  
## 3 45 17 10 13.5 4.7 139 7400 38 60 230 40 2 mild Cad  
## 4 27 30 76 12.1 4.4 142 13000 18 72 742 55 0 Severe Normal  
## 5 50 16 27 13.2 4.0 140 9200 55 39 274 50 0 Severe Normal  
## 6 34 13 18 15.6 4.2 141 7300 26 66 194 50 0 N Cad

tail(FullSetB4)

## Ã¯..Age Weight Length Sex BMI DM HTN Smoker ExSmoker FH Obesity  
## 298 30 100 172 Male 33.80206 0 0 1 0 1 Y  
## 299 58 84 168 Male 29.76190 0 0 0 0 0 Y  
## 300 55 64 152 Fmale 27.70083 0 0 0 0 0 Y  
## 301 48 77 160 Fmale 30.07812 0 1 0 0 1 Y  
## 302 57 90 159 Fmale 35.59986 1 0 0 0 0 Y  
## 303 56 85 170 Fmale 29.41176 0 1 1 0 0 Y  
## CRF CVA AD TD CHF DLP BP PR Edema WPP LR SysM DiaM TCP Dyspnea Fclass  
## 298 N N N N N N 110 60 0 N N N N 0 N 0  
## 299 N N N N N N 100 76 0 N N N N 1 N 0  
## 300 N N N N N N 100 60 0 N N Y N 0 Y 0  
## 301 N N N N N N 130 70 0 N N N N 0 N 0  
## 302 N N N N N N 100 60 0 N N N N 0 Y 0  
## 303 N N N N N N 120 80 0 N N N N 1 N 0  
## ACP NCP ECP LTAng QWave STelev STdep Tinv LVH PoorR BBB FBS Cr TG  
## 298 Y N N N 0 0 0 0 N N N 83 1.0 205  
## 299 N N N N 0 0 0 0 N N N 92 1.0 112  
## 300 Y N N N 0 0 0 0 N N LBBB 86 0.9 111  
## 301 N Y N N 0 0 0 0 N N RBBB 83 1.0 93  
## 302 Y N N N 0 0 0 0 N N N 96 1.0 116  
## 303 N N N N 0 0 0 1 N N N 78 0.7 139  
## LDL HDL BUN ESR Hb K Na WBC Lymph Neut PLT EF RWMA VHD CAD  
## 298 97 53 20 16 13.1 4.0 143 9100 39 60 294 55 1 N Normal  
## 299 115 44 13 13 12.3 4.8 146 8500 34 58 251 45 0 N Cad  
## 300 40 23 23 3 12.4 4.0 139 11400 16 80 377 40 0 mild Normal  
## 301 112 42 13 20 12.8 4.0 140 9000 35 55 279 55 0 N Normal  
## 302 130 49 14 31 10.1 3.8 141 3800 48 40 208 55 0 N Normal  
## 303 124 34 16 13 14.7 4.4 147 6000 32 55 302 55 0 N Cad

str(FullSetB4)

## 'data.frame': 303 obs. of 56 variables:  
## $ Ã¯..Age : int 53 67 54 66 50 50 55 72 58 60 ...  
## $ Weight : int 90 70 54 67 87 75 80 80 84 71 ...  
## $ Length : int 175 157 164 158 153 175 165 175 163 170 ...  
## $ Sex : Factor w/ 2 levels "Fmale","Male": 2 1 2 1 1 2 2 2 1 2 ...  
## $ BMI : num 29.4 28.4 20.1 26.8 37.2 ...  
## $ DM : int 0 0 0 0 0 0 0 1 0 1 ...  
## $ HTN : int 1 1 0 1 1 0 0 0 0 0 ...  
## $ Smoker : int 1 0 1 0 0 1 0 1 0 0 ...  
## $ ExSmoker: int 0 0 0 0 0 0 1 0 0 0 ...  
## $ FH : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ Obesity : Factor w/ 2 levels "N","Y": 2 2 1 2 2 1 2 2 2 1 ...  
## $ CRF : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CVA : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ AD : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ TD : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CHF : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ DLP : Factor w/ 2 levels "N","Y": 2 1 1 1 1 1 1 2 1 1 ...  
## $ BP : int 110 140 100 100 110 118 110 130 90 130 ...  
## $ PR : int 80 80 100 80 80 70 80 70 50 70 ...  
## $ Edema : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ WPP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LR : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ SysM : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 1 1 ...  
## $ DiaM : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 1 1 ...  
## $ TCP : int 0 1 1 0 0 1 1 1 0 1 ...  
## $ Dyspnea : Factor w/ 2 levels "N","Y": 1 1 1 2 2 1 1 1 2 2 ...  
## $ Fclass : int 0 0 0 3 2 3 0 0 0 2 ...  
## $ ACP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ NCP : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 2 1 ...  
## $ ECP : Factor w/ 1 level "N": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LTAng : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ QWave : int 0 0 0 0 0 0 1 0 0 0 ...  
## $ STelev : int 0 0 0 0 0 0 1 0 0 0 ...  
## $ STdep : int 1 1 0 1 0 0 0 1 0 0 ...  
## $ Tinv : int 1 1 0 0 0 0 1 1 0 0 ...  
## $ LVH : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ PoorR : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ BBB : Factor w/ 3 levels "LBBB","N","RBBB": 2 2 2 2 2 2 2 2 2 1 ...  
## $ FBS : int 90 80 85 78 104 86 80 130 69 209 ...  
## $ Cr : num 0.7 1 1 1.2 1 1 0.8 0.9 0.6 1.3 ...  
## $ TG : int 250 309 103 63 170 139 83 80 79 80 ...  
## $ LDL : int 155 121 70 55 110 119 85 90 90 90 ...  
## $ HDL : num 30 36 45 27 50 34 34 55 59 44 ...  
## $ BUN : int 8 30 17 30 16 13 12 19 15 16 ...  
## $ ESR : int 7 26 10 76 27 18 38 4 5 8 ...  
## $ Hb : num 15.6 13.9 13.5 12.1 13.2 15.6 14.1 16.1 11.6 13.9 ...  
## $ K : num 4.7 4.7 4.7 4.4 4 4.2 4.8 4.3 3.4 4.6 ...  
## $ Na : int 141 156 139 142 140 141 139 142 139 140 ...  
## $ WBC : int 5700 7700 7400 13000 9200 7300 9400 12200 5100 4900 ...  
## $ Lymph : int 39 38 38 18 55 26 58 25 49 55 ...  
## $ Neut : int 52 55 60 72 39 66 33 74 50 42 ...  
## $ PLT : int 261 165 230 742 274 194 292 410 370 380 ...  
## $ EF : int 50 40 40 55 50 50 40 45 50 40 ...  
## $ RWMA : int 0 4 2 0 0 0 4 4 0 2 ...  
## $ VHD : Factor w/ 4 levels "mild","Moderate",..: 3 3 1 4 4 3 1 1 3 3 ...  
## $ CAD : Factor w/ 2 levels "Cad","Normal": 1 1 1 2 2 1 1 1 2 1 ...

# Fixing ‘Age’ column header (not displaying correctly):

names(FullSetB4)[1] <- "Age"

# Remove ‘Length’ attribute - since not used in final study; and captured via BMI (and ‘Obesity’ uses ‘BMI’):

FullSetB4 <- FullSetB4[-3]  
dim(FullSetB4)

## [1] 303 55

# CHECKING FOR MISSING DATA & ERRORS:

# Check number of observations and whether there is any missing data:

nrow(FullSetB4)

## [1] 303

sum(is.na(FullSetB4))

## [1] 0

# Double checking whether all observations are complete (no missing data): No incomplete cases found, but I am commenting this out for the html output - since the output is very long.

# FullSetB4[complete.cases(FullSetB4),]

# Checking for errors, starting with ‘Sex’ attribute (column 3):

SexF <- sum(FullSetB4$Sex == 'Fmale')  
SexM <- sum(FullSetB4$Sex == 'Male')  
sum(SexF + SexM)

## [1] 303

# Checking for errors in columns 5:9, 19, 24, 31:34 (must be ‘0’ or ‘1’)

#library(tidyverse)  
#library(dbplyr)  
  
Bins0 <- FullSetB4%>%  
 gather(x, value, 5:9,19,24,31:34)%>%  
 tally(value == 0)  
Bins1 <- FullSetB4%>%  
 gather(x, value, 5:9,19,24,31:34)%>%  
 tally(value == 1)  
 (Bins0 + Bins1)/11

## n  
## 1 303

# Checking for errors in columns 10:16, 20:23, 25, 27:28, 30, 35:36 (must be ‘Y’ or ‘N’)

BinsY <- FullSetB4%>%  
 gather(x, value, 10:16,20:23,25,27:28,30,35:36)%>%  
 tally(value == 'Y')  
BinsN <- FullSetB4%>%  
 gather(x, value, 10:16,20:23,25,27:28,30,35:36)%>%  
 tally(value == 'N')  
 (BinsY + BinsN)/17

## n  
## 1 303

# Checking for errors in the ‘BBB’(Bundle Branch Block) attribute (column 37):

nBBB <- sum(FullSetB4$BBB == 'N')  
LBBB <- sum(FullSetB4$BBB == 'LBBB')  
RBBB <- sum(FullSetB4$BBB == 'RBBB')  
sum(nBBB+LBBB+RBBB)

## [1] 303

# Checking for errors in the ‘VHD’(Valvular Heart Disease) attribute (column 54):

Vmild <- sum(FullSetB4$VHD == 'mild')  
VMod <- sum(FullSetB4$VHD == 'Moderate')  
VN <- sum(FullSetB4$VHD == 'N')  
VSev <- sum(FullSetB4$VHD == 'Severe')  
sum(Vmild+VMod+VN+VSev)

## [1] 303

# Converting int types to factor, where appropriate:

Allv1 <- FullSetB4  
Allv1$DM <- as.factor(Allv1$DM)  
Allv1$HTN <- as.factor(Allv1$HTN)  
Allv1$Smoker <- as.factor(Allv1$Smoker)  
Allv1$ExSmoker <- as.factor(Allv1$ExSmoker)  
Allv1$FH <- as.factor(Allv1$FH)  
Allv1$Edema <- as.factor(Allv1$Edema)  
Allv1$TCP <- as.factor(Allv1$TCP)  
Allv1$QWave <- as.factor(Allv1$QWave)  
Allv1$STelev <- as.factor(Allv1$STelev)  
Allv1$STdep <- as.factor(Allv1$STdep)  
Allv1$Tinv <- as.factor(Allv1$Tinv)  
#str(Allv1)

# DISCRETIZING:

# Change age attribute to factor: If male and age is <= to 45, OR if female and age is <= 55; categorize as “med” (for ‘medium’). If male and age is > 45 or female and age is > 55, categorize as “high”.

Allv1$Age <- factor(ifelse(Allv1$Age <= 45 & Allv1$Sex == 'Male' | Allv1$Age <= 55 & Allv1$Sex == 'Fmale', "Med", "High"))  
str(Allv1)

## 'data.frame': 303 obs. of 55 variables:  
## $ Age : Factor w/ 2 levels "High","Med": 1 1 1 1 2 1 1 1 1 1 ...  
## $ Weight : int 90 70 54 67 87 75 80 80 84 71 ...  
## $ Sex : Factor w/ 2 levels "Fmale","Male": 2 1 2 1 1 2 2 2 1 2 ...  
## $ BMI : num 29.4 28.4 20.1 26.8 37.2 ...  
## $ DM : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 2 ...  
## $ HTN : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 1 1 1 1 ...  
## $ Smoker : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 2 1 1 ...  
## $ ExSmoker: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ FH : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Obesity : Factor w/ 2 levels "N","Y": 2 2 1 2 2 1 2 2 2 1 ...  
## $ CRF : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CVA : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ AD : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ TD : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CHF : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ DLP : Factor w/ 2 levels "N","Y": 2 1 1 1 1 1 1 2 1 1 ...  
## $ BP : int 110 140 100 100 110 118 110 130 90 130 ...  
## $ PR : int 80 80 100 80 80 70 80 70 50 70 ...  
## $ Edema : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...  
## $ WPP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LR : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ SysM : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 1 1 ...  
## $ DiaM : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 1 1 ...  
## $ TCP : Factor w/ 2 levels "0","1": 1 2 2 1 1 2 2 2 1 2 ...  
## $ Dyspnea : Factor w/ 2 levels "N","Y": 1 1 1 2 2 1 1 1 2 2 ...  
## $ Fclass : int 0 0 0 3 2 3 0 0 0 2 ...  
## $ ACP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ NCP : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 2 1 ...  
## $ ECP : Factor w/ 1 level "N": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LTAng : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ QWave : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ STelev : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ STdep : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 1 2 1 1 ...  
## $ Tinv : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 2 2 1 1 ...  
## $ LVH : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ PoorR : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ BBB : Factor w/ 3 levels "LBBB","N","RBBB": 2 2 2 2 2 2 2 2 2 1 ...  
## $ FBS : int 90 80 85 78 104 86 80 130 69 209 ...  
## $ Cr : num 0.7 1 1 1.2 1 1 0.8 0.9 0.6 1.3 ...  
## $ TG : int 250 309 103 63 170 139 83 80 79 80 ...  
## $ LDL : int 155 121 70 55 110 119 85 90 90 90 ...  
## $ HDL : num 30 36 45 27 50 34 34 55 59 44 ...  
## $ BUN : int 8 30 17 30 16 13 12 19 15 16 ...  
## $ ESR : int 7 26 10 76 27 18 38 4 5 8 ...  
## $ Hb : num 15.6 13.9 13.5 12.1 13.2 15.6 14.1 16.1 11.6 13.9 ...  
## $ K : num 4.7 4.7 4.7 4.4 4 4.2 4.8 4.3 3.4 4.6 ...  
## $ Na : int 141 156 139 142 140 141 139 142 139 140 ...  
## $ WBC : int 5700 7700 7400 13000 9200 7300 9400 12200 5100 4900 ...  
## $ Lymph : int 39 38 38 18 55 26 58 25 49 55 ...  
## $ Neut : int 52 55 60 72 39 66 33 74 50 42 ...  
## $ PLT : int 261 165 230 742 274 194 292 410 370 380 ...  
## $ EF : int 50 40 40 55 50 50 40 45 50 40 ...  
## $ RWMA : int 0 4 2 0 0 0 4 4 0 2 ...  
## $ VHD : Factor w/ 4 levels "mild","Moderate",..: 3 3 1 4 4 3 1 1 3 3 ...  
## $ CAD : Factor w/ 2 levels "Cad","Normal": 1 1 1 2 2 1 1 1 2 1 ...

# Checking for any errors in Age factor values, and the proportion of ‘medium’ to ‘high’ ages (after being discretized):

MedAge <- sum(Allv1$Age == 'Med')  
HighAge <- sum(Allv1$Age == 'High')  
sum(MedAge+HighAge)

## [1] 303

(MedAge/(MedAge+HighAge))\*100

## [1] 21.45215

# Discretize Weight for males vs females. These discretized ranges were not specified in the original study, so I am using the mean and the factoring into ‘below’(“Ave”) or ‘above’ (“High”) the group average:

tapply(Allv1$Weight, Allv1$Sex, mean)

## Fmale Male   
## 70.52756 76.21591

Allv1$Weight <- factor(ifelse(Allv1$Weight <= 76.2 & Allv1$Sex == 'Male' | Allv1$Weight <= 70.5 & Allv1$Sex == 'Fmale', "Ave", "High"))

# Discretize Blood Pressure (BP):

max(Allv1$BP)

## [1] 190

Allv1$BP <- cut(Allv1$BP, breaks = c(0,89,140,200), labels = c("Low", "Med", "High"))  
  
BPlow <- sum(Allv1$BP == "Low")  
BPlow

## [1] 0

BPmed <- sum(Allv1$BP == "Med")  
BPmed

## [1] 255

BPhigh <- sum(Allv1$BP == "High")  
BPhigh

## [1] 48

BPcount <- sum(BPlow+BPmed+BPhigh)  
BPcount

## [1] 303

# Note that no patients had a ‘low’ blood pressure. Check the proportion of medium to high blood pressure results:

(BPmed/(BPmed+BPhigh))\*100

## [1] 84.15842

# Discretize PUlse Rate (PR). Note: Very low variation in pulse rate (98.3% was categorized with a ‘medium’ heart rate). THis would make the ‘Pulse rate’ attribute a good candidate for removal.

max(Allv1$PR)

## [1] 110

Allv1$PR <- cut(Allv1$PR, breaks = c(0,59,100,200), labels = c("Low", "Med", "High"))  
  
PRlow <- sum(Allv1$PR == "Low")  
PRlow

## [1] 2

PRmed <- sum(Allv1$PR == "Med")  
PRmed

## [1] 298

PRhigh <- sum(Allv1$PR == "High")  
PRhigh

## [1] 3

PRcount <- sum(PRlow+PRmed+PRhigh)  
PRcount

## [1] 303

# Discretize Heart Rate Functional Class (Fclass): Class 0 equals ‘Med’ and classes 1, 2, and 3 equal ‘High’. Note that 69.6% of the patients were evaluated to have a “Medium” Heart Failure Functional Class.

sum(Allv1$Fclass == 0)

## [1] 211

Allv1$Fclass <- factor(ifelse(Allv1$Fclass == 0, "Med", "High"))  
  
FCmed <- sum(Allv1$Fclass == "Med")  
FCmed

## [1] 211

FChigh <- sum(Allv1$Fclass == "High")  
FChigh

## [1] 92

sum(FCmed+FChigh)

## [1] 303

(FCmed/(FCmed+FChigh))\*100

## [1] 69.63696

# Count the levels and results of the ‘Exertional Chest Pain’ (ECP) attribute (yes/no entries).

nlevels(Allv1$ECP)

## [1] 1

sum(Allv1$ECP == 'N')

## [1] 303

# Deleting this attribute given that there is no variation in the results, and as such, does not improve predictive effectiveness:

Allv1$ECP <- NULL  
Allv2 <- Allv1

# Checking ‘Bundle Branch Block’ (BBB) for the numbers of LBBB and RBBB results there are in the total (93% of patients had no BBB).

sum(Allv2$BBB == 'N')

## [1] 282

sum(Allv2$BBB == 'LBBB')

## [1] 13

sum(Allv2$BBB == 'RBBB')

## [1] 8

# Discretizing the ‘Fasting Blood Sugar’ attribute into ‘low’, ‘med’ and ‘high’:

max(Allv2$FBS)

## [1] 400

Allv2$FBS <- cut(Allv2$FBS, breaks = c(0,69,105,500), labels = c("Low", "Med", "High"))  
  
FBSlow <- sum(Allv2$FBS == "Low")  
FBSlow

## [1] 3

FBSmed <- sum(Allv2$FBS == "Med")  
FBSmed

## [1] 179

FBShigh <- sum(Allv2$FBS == "High")  
FBShigh

## [1] 121

FBScount <- sum(FBSlow+FBSmed+FBShigh)  
FBScount

## [1] 303

# Discretizing the ‘Creatine’ attribute into ‘low’, ‘med’ and ‘high’:

max(Allv2$Cr)

## [1] 2.2

Allv2$Cr <- cut(Allv2$Cr, breaks = c(0,0.69,1.5,2.5), labels = c("Low", "Med", "High"))  
  
Crlow <- sum(Allv2$Cr == "Low")  
Crlow

## [1] 9

Crmed <- sum(Allv2$Cr == "Med")  
Crmed

## [1] 281

Crhigh <- sum(Allv2$Cr == "High")  
Crhigh

## [1] 13

CrCount <- sum(Crlow+Crmed+Crhigh)  
CrCount

## [1] 303

# Discretizing the Triglyceride (TG) variable into ‘med’ and ‘high’:

max(Allv2$TG)

## [1] 1050

Allv2$TG <- cut(Allv2$TG, breaks = c(0,200,1100), labels = c("Med", "High"))  
  
TGmed <- sum(Allv2$TG == "Med")  
TGmed

## [1] 241

TGhigh <- sum(Allv2$TG == "High")  
TGhigh

## [1] 62

TGcount <- sum(TGmed+TGhigh)  
TGcount

## [1] 303

# Discretizing Low Density Lipoprotein (LDL). It appears that the results of the cateorization of Triglycerides exactly matches that of LDL, hence, one of these two attributes can be removed.

max(Allv2$LDL)

## [1] 232

Allv2$LDL <- cut(Allv2$LDL, breaks = c(0,130,250), labels = c("Med", "High"))  
  
LDLmed <- sum(Allv2$LDL == "Med")  
LDLmed

## [1] 241

LDLhigh <- sum(Allv2$LDL == "High")  
LDLhigh

## [1] 62

LDLcount <- sum(LDLmed+LDLhigh)  
LDLcount

## [1] 303

# Discretizing High Density Lipoprotein (HDL).

max(Allv2$HDL)

## [1] 111

Allv2$HDL <- cut(Allv2$HDL, breaks = c(0,34,120), labels = c("Low", "Med"))  
  
HDLlow <- sum(Allv2$HDL == "Low")  
HDLlow

## [1] 87

HDLmed <- sum(Allv2$HDL == "Med")  
HDLmed

## [1] 216

HDLcount <- sum(HDLlow+HDLmed)  
HDLcount

## [1] 303

# Discretizing Blood Urea Nitrogen (BUN):

max(Allv2$BUN)

## [1] 52

Allv2$BUN <- cut(Allv2$BUN, breaks = c(0,6,20,55), labels = c("Low", "Med", "High"))  
  
BUNlow <- sum(Allv2$BUN == "Low")  
BUNlow

## [1] 1

BUNmed <- sum(Allv2$BUN == "Med")  
BUNmed

## [1] 229

BUNhigh <- sum(Allv2$BUN == "High")  
BUNhigh

## [1] 73

BUNcount <- sum(BUNlow+BUNmed+BUNhigh)  
BUNcount

## [1] 303

# Taking steps to discretize ‘Erythrocyte Sedimentation Rate’ (ESR). If male and ESR <= age/2 OR if female and ESR <= (age/2)+5 then ESR is considered “Mediium”. If male and ESR is > age/2 OR if female and ESR > (age/2)+5, then ESR is considered “High”:

tempAge <- data.frame(FullSetB4[,1:3, 44])  
  
tempAge$Age <- ifelse(tempAge$Sex == 'Male', tempAge$Age/2, tempAge$Age)  
tempAge2 <- tempAge  
  
tempAge2$Age <- ifelse(tempAge2$Sex == 'Fmale', (tempAge2$Age/2)+5, tempAge2$Age)  
  
tempAge2$ESR <- FullSetB4$ESR  
tempAge2$Weight <- NULL  
tempAge3 <- tempAge2  
  
tempAge3$ESR <- factor(ifelse(tempAge3$ESR <= tempAge3$Age & tempAge3$Sex == 'Male' | tempAge3$ESR <= tempAge3$Age & tempAge3$Sex == 'Fmale', "Med", "High"))  
str(tempAge3)

## 'data.frame': 303 obs. of 3 variables:  
## $ Age: num 26.5 38.5 27 38 30 25 27.5 36 34 30 ...  
## $ Sex: Factor w/ 2 levels "Fmale","Male": 2 1 2 1 1 2 2 2 1 2 ...  
## $ ESR: Factor w/ 2 levels "High","Med": 2 2 2 1 2 2 1 2 2 2 ...

Allv2$ESR <- tempAge3$ESR

# Discretizing Hemoglobin (Hb). If male & Hb < 14 or if female and Hb is < 12.5, then Hb is “Low”. If male & Hb is >= 14 but <= 17 OR if female and Hb is >= 12 but <= 15, then Hb is “Medium”. If male & Hb > 17 or if female and Hb is > 15, then Hb is “High”.

tempHb <- data.frame(Allv2[,3:44])  
tempHb[,2:41] <- NULL  
  
tempHb$Hb <- with(tempHb, ifelse(tempHb$Sex == 'Male' & tempHb$Hb < 14 | tempHb$Sex == 'Fmale' & tempHb$Hb < 12.5, "Low", ifelse(tempHb$Sex == 'Male' & tempHb$Hb > 17 | tempHb$Sex == 'Fmale' & tempHb$Hb > 15, "High", "Med")))  
  
Allv2$Hb <- as.factor(tempHb$Hb)  
#str(Allv2)

# Discretizing Potassium (K):

max(Allv2$K)

## [1] 6.6

Allv2$K <- cut(Allv2$K, breaks = c(0,3.7,5.6,6.8), labels = c("Low", "Med", "High"))  
  
Klow <- sum(Allv2$K == "Low")  
Klow

## [1] 36

Kmed <- sum(Allv2$K == "Med")  
Kmed

## [1] 266

Khigh <- sum(Allv2$K == "High")  
Khigh

## [1] 1

Kcount <- sum(Klow+Kmed+Khigh)  
Kcount

## [1] 303

# Discretizing Sodium (Na):

max(Allv2$Na)

## [1] 156

Allv2$Na <- cut(Allv2$Na, breaks = c(0,135,146,157), labels = c("Low", "Med", "High"))  
  
NaLow <- sum(Allv2$Na == "Low")  
NaLow

## [1] 18

NaMed <- sum(Allv2$Na == "Med")  
NaMed

## [1] 269

NaHigh <- sum(Allv2$Na == "High")  
NaHigh

## [1] 16

NaCount <- sum(NaLow+NaMed+NaHigh)  
NaCount

## [1] 303

# Discretizing White Blood Cell (WBC) count:

max(Allv2$WBC)

## [1] 18000

Allv2$WBC <- cut(Allv2$WBC, breaks = c(0,3999,11000,19000), labels = c("Low", "Med", "High"))  
  
WBClow <- sum(Allv2$WBC == "Low")  
WBClow

## [1] 3

WBCmed <- sum(Allv2$WBC == "Med")  
WBCmed

## [1] 276

WBChigh <- sum(Allv2$WBC == "High")  
WBChigh

## [1] 24

WBCcount <- sum(WBClow+WBCmed+WBChigh)  
WBCcount

## [1] 303

# Discretizing Lymphocyte (Lymph) percentage results (See reference 21 on Literature Review). Spreading the results into 3 sections: ‘Low’, ‘Med’ and ‘High’.

max(Allv2$Lymph)

## [1] 60

Allv2$Lymph <- cut(Allv2$Lymph, breaks = c(0,17,45,65), labels = c("Low", "Med", "High"))  
  
LymphLow <- sum(Allv2$Lymph == "Low")  
LymphLow

## [1] 25

LymphMed <- sum(Allv2$Lymph == "Med")  
LymphMed

## [1] 256

LymphHigh <- sum(Allv2$Lymph == "High")  
LymphHigh

## [1] 22

LymphCount <- sum(LymphLow+LymphMed+LymphHigh)  
LymphCount

## [1] 303

# Discretizing Neutrophil (Neut) percentage results (See reference 22 on Literature Review). Spreading the results into 3 sections: ‘Low’, ‘Med’ and ‘High’.

max(Allv2$Neut)

## [1] 89

Allv2$Neut <- cut(Allv2$Neut, breaks = c(0,44,75,90), labels = c("Low", "Med", "High"))  
  
NeutLow <- sum(Allv2$Neut == "Low")  
NeutLow

## [1] 16

NeutMed <- sum(Allv2$Neut == "Med")  
NeutMed

## [1] 264

NeutHigh <- sum(Allv2$Neut == "High")  
NeutHigh

## [1] 23

NeutCount <- sum(NeutLow+NeutMed+NeutHigh)  
NeutCount

## [1] 303

# Discretizing Ejection Fraction (EF):

max(Allv2$EF)

## [1] 60

Allv2$EF <- cut(Allv2$EF, breaks = c(0,50,120), labels = c("Low", "Med"))  
  
EFlow <- sum(Allv2$EF == "Low")  
EFlow

## [1] 197

EFmed <- sum(Allv2$EF == "Med")  
EFmed

## [1] 106

EFcount <- sum(EFlow+EFmed)  
EFcount

## [1] 303

# Discretizing Platelet (PLT) count (Note: 96% of patients had ‘Medium’ counts)

max(Allv2$PLT)

## [1] 742

Allv2$PLT <- cut(Allv2$PLT, breaks = c(0,149,450,750), labels = c("Low", "Med", "High"))  
  
PLTlow <- sum(Allv2$PLT == "Low")  
PLTlow

## [1] 11

PLTmed <- sum(Allv2$PLT == "Med")  
PLTmed

## [1] 291

PLThigh <- sum(Allv2$PLT == "High")  
PLThigh

## [1] 1

PLTcount <- sum(PLTlow+PLTmed+PLThigh)  
PLTcount

## [1] 303

# Discretizing Regional Wall Motion Abnormality (RWMA). RWMA = 0 is ‘Med’, RWMA != to 0 are ‘High’). Found that 71.6% were found to be ‘Medium’.

sum(Allv2$RWMA == 0)

## [1] 217

Allv2$RWMA <- factor(ifelse(Allv2$RWMA == 0, "Med", "High"))  
  
RWMAmed <- sum(Allv2$RWMA == "Med")  
RWMAmed

## [1] 217

RWMAhigh <- sum(Allv2$RWMA == "High")  
RWMAhigh

## [1] 86

sum(RWMAmed+RWMAhigh)

## [1] 303

(RWMAmed/(RWMAmed+RWMAhigh))\*100

## [1] 71.61716

# Removing the ‘BMI’ attribute since this is captured in the ‘Obesity’ attribute (given that a BMI result that is over 25 is considered ‘Obese’). Matching the removed attributes for the ‘before’ and ‘after’ (discretization) data sets:

Allv2$BMI <- NULL  
  
FullSetB4$BMI <- NULL  
FullSetB4$ECP <- NULL

#str(Allv2)

AllvFinal <- Allv2  
AllvFinal$Age <- factor(AllvFinal$Age, levels = c("Med", "High"), ordered = TRUE)  
AllvFinal$Weight <- factor(AllvFinal$Weight, levels = c("Ave", "High"), ordered = TRUE)  
AllvFinal$BP <- factor(AllvFinal$BP, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$PR <- factor(AllvFinal$PR, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$Fclass <- factor(AllvFinal$Fclass, levels = c("Med", "High"), ordered = TRUE)  
AllvFinal$FBS <- factor(AllvFinal$FBS, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$Cr <- factor(AllvFinal$Cr, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$TG <- factor(AllvFinal$TG, levels = c("Med", "High"), ordered = TRUE)  
AllvFinal$LDL <- factor(AllvFinal$LDL, levels = c("Med", "High"), ordered = TRUE)  
AllvFinal$HDL <- factor(AllvFinal$HDL, levels = c("Low", "Med"), ordered = TRUE)  
AllvFinal$BUN <- factor(AllvFinal$BUN, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$ESR <- factor(AllvFinal$ESR, levels = c("Med", "High"), ordered = TRUE)  
AllvFinal$Hb <- factor(AllvFinal$Hb, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$K <- factor(AllvFinal$K, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$Na <- factor(AllvFinal$Na, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$WBC <- factor(AllvFinal$WBC, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$Lymph <- factor(AllvFinal$Lymph, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$Neut <- factor(AllvFinal$Neut, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$PLT <- factor(AllvFinal$PLT, levels = c("Low", "Med", "High"), ordered = TRUE)  
AllvFinal$EF <- factor(AllvFinal$EF, levels = c("Low", "Med"), ordered = TRUE)  
AllvFinal$RWMA <- factor(AllvFinal$RWMA, levels = c("Med", "High"), ordered = TRUE)  
AllvFinal$VHD <- factor(AllvFinal$VHD, levels = c("N", "mild", "Moderate", "Severe"), ordered = TRUE)  
  
str(AllvFinal)

## 'data.frame': 303 obs. of 53 variables:  
## $ Age : Ord.factor w/ 2 levels "Med"<"High": 2 2 2 2 1 2 2 2 2 2 ...  
## $ Weight : Ord.factor w/ 2 levels "Ave"<"High": 2 1 1 1 2 1 2 2 2 1 ...  
## $ Sex : Factor w/ 2 levels "Fmale","Male": 2 1 2 1 1 2 2 2 1 2 ...  
## $ DM : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 1 2 ...  
## $ HTN : Factor w/ 2 levels "0","1": 2 2 1 2 2 1 1 1 1 1 ...  
## $ Smoker : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 2 1 1 ...  
## $ ExSmoker: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ FH : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Obesity : Factor w/ 2 levels "N","Y": 2 2 1 2 2 1 2 2 2 1 ...  
## $ CRF : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CVA : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ AD : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ TD : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CHF : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ DLP : Factor w/ 2 levels "N","Y": 2 1 1 1 1 1 1 2 1 1 ...  
## $ BP : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 2 2 2 2 2 2 ...  
## $ PR : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 2 2 2 2 1 2 ...  
## $ Edema : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 1 1 1 ...  
## $ WPP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ LR : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ SysM : Factor w/ 2 levels "N","Y": 1 1 1 1 2 1 2 1 1 1 ...  
## $ DiaM : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 1 1 ...  
## $ TCP : Factor w/ 2 levels "0","1": 1 2 2 1 1 2 2 2 1 2 ...  
## $ Dyspnea : Factor w/ 2 levels "N","Y": 1 1 1 2 2 1 1 1 2 2 ...  
## $ Fclass : Ord.factor w/ 2 levels "Med"<"High": 1 1 1 2 2 2 1 1 1 2 ...  
## $ ACP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ NCP : Factor w/ 2 levels "N","Y": 1 1 1 2 1 1 1 1 2 1 ...  
## $ LTAng : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ QWave : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ STelev : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 2 1 1 1 ...  
## $ STdep : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 1 2 1 1 ...  
## $ Tinv : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 2 2 1 1 ...  
## $ LVH : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ PoorR : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 1 1 1 ...  
## $ BBB : Factor w/ 3 levels "LBBB","N","RBBB": 2 2 2 2 2 2 2 2 2 1 ...  
## $ FBS : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 2 2 2 3 1 3 ...  
## $ Cr : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 2 2 2 2 1 2 ...  
## $ TG : Ord.factor w/ 2 levels "Med"<"High": 2 2 1 1 1 1 1 1 1 1 ...  
## $ LDL : Ord.factor w/ 2 levels "Med"<"High": 2 1 1 1 1 1 1 1 1 1 ...  
## $ HDL : Ord.factor w/ 2 levels "Low"<"Med": 1 2 2 1 2 1 1 2 2 2 ...  
## $ BUN : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 3 2 3 2 2 2 2 2 2 ...  
## $ ESR : Ord.factor w/ 2 levels "Med"<"High": 1 1 1 2 1 1 2 1 1 1 ...  
## $ Hb : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 1 1 2 2 2 2 1 1 ...  
## $ K : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 2 2 2 2 1 2 ...  
## $ Na : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 3 2 2 2 2 2 2 2 2 ...  
## $ WBC : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 3 2 2 2 3 2 2 ...  
## $ Lymph : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 3 2 3 2 3 3 ...  
## $ Neut : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 2 1 2 1 2 2 1 ...  
## $ PLT : Ord.factor w/ 3 levels "Low"<"Med"<"High": 2 2 2 3 2 2 2 2 2 2 ...  
## $ EF : Ord.factor w/ 2 levels "Low"<"Med": 1 1 1 2 1 1 1 1 1 1 ...  
## $ RWMA : Ord.factor w/ 2 levels "Med"<"High": 1 2 2 1 1 1 2 2 1 2 ...  
## $ VHD : Ord.factor w/ 4 levels "N"<"mild"<"Moderate"<..: 1 1 2 4 4 1 2 2 1 1 ...  
## $ CAD : Factor w/ 2 levels "Cad","Normal": 1 1 1 2 2 1 1 1 2 1 ...

# Taking the completed (and discretized) ‘Allv2’ data frame and saving it as 4 separate attribute groups (data frames); to be used for the rest of the predictions/modeling. Keeping the dependent result (CAD) at this stage:

Demo <- data.frame(Allv2[,1:15])  
Demo$CAD <- Allv2$CAD  
  
Symptom <- data.frame(Allv2[,16:28])  
Symptom$CAD <- Allv2$CAD  
  
ECG <- data.frame(Allv2[,29:35])  
ECG$CAD <- Allv2$CAD  
  
LAB <- data.frame(Allv2[,36:52])  
LAB$CAD <- Allv2$CAD

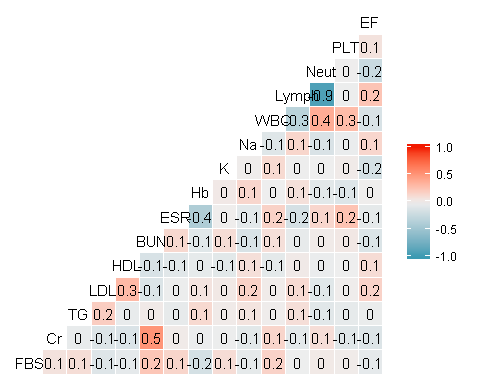
# Taking the data set before discretizing and factoring (FullSetB4) and saving it as 4 separate attribute groups (data frames); to be used for analysis. First removing the ‘ECP’ and ‘BMI’ attributes (to match the discretized data set):

FullSetB42 <- FullSetB4  
  
DemoB4 <- data.frame(FullSetB42[,1:15])  
DemoB4$CAD <- FullSetB42$CAD  
  
SymptomB4 <- data.frame(FullSetB42[,16:28])  
SymptomB4$CAD <- FullSetB42$CAD  
  
ECGB4 <- data.frame(FullSetB42[,29:35])  
ECGB4$CAD <- FullSetB42$CAD  
  
LABB4 <- data.frame(FullSetB42[,36:52])  
LABB4$CAD <- FullSetB42$CAD

UNIVARIATE ANALYSIS:

# Correlation coefficients for the ‘LAB’ results show no correlation (other than the negative correlation between Lymphocyte and Neutrophil results). With regards to this inverse relationship, an abnormal increase in one kind of white blood cell can cause a decrease in another kind. Both abnormal results can be due to the same underlying condition.

library(GGally)  
  
LABcor3 <- ggcorr(LABB4[1:15], palette = "RdY1Gn", label = TRUE)  
LABcor3

 # Looking into the Neutrophil to Lymphocyte relationship further. Calculating the NLR (neutrophil-lymphocyte ratio) below. Reference # 24 shows that they have identified that the normal NLR values in an adult, non-geriatric, population in good health are between 0.78 and 3.53.

NLR <- data.frame(FullSetB42[48:47])  
NLR$Ratio <- NLR$Neut/NLR$Lymph  
  
min(NLR$Ratio)

## [1] 0.5689655

max(NLR$Ratio)

## [1] 12.14286

mean(NLR$Ratio)

## [1] 2.244218

NLR$Result <- cut(NLR$Ratio, breaks = c(0,0.77,3.53,12.5), labels = c("Low", "Med", "High"))  
  
NLR$CAD <- FullSetB42$CAD  
  
NLRlow <- sum(NLR$Result == "Low")  
NLRlow

## [1] 10

NLRmed <- sum(NLR$Result == "Med")  
NLRmed

## [1] 257

NLRhigh <- sum(NLR$Result == "High")  
NLRhigh

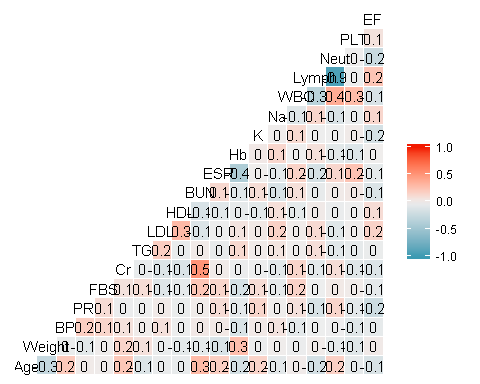
## [1] 36

NLRcount <- sum(NLRlow+NLRmed+NLRhigh)  
NLRcount

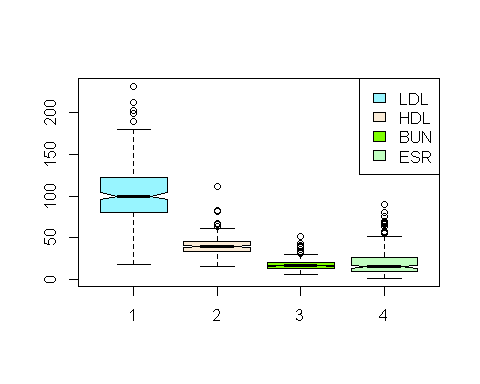
## [1] 303

# Include remaining numeric attributes for further correlation overview: Shows no significant correlation results, other than what was already found:

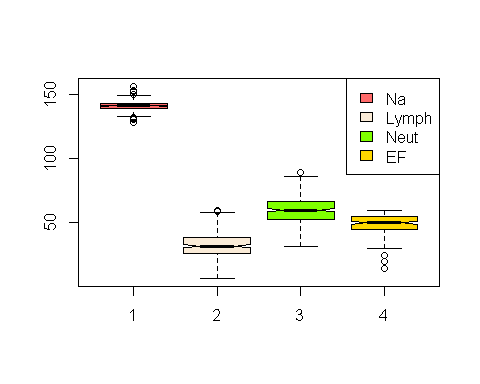
library(GGally)  
  
Allints1 <- ggcorr(FullSetB42[,c(1:2, 16:17, 36:50)], palette = "RdY1Gn", label = TRUE)  
Allints1

 # Boxplot of Low Density Lipoprotein (LDL), High Density Lipoprotein (HDL), Blood Urea Nitrogen (BUN), and Erythrocyte Sedimentation Rate (ESR):

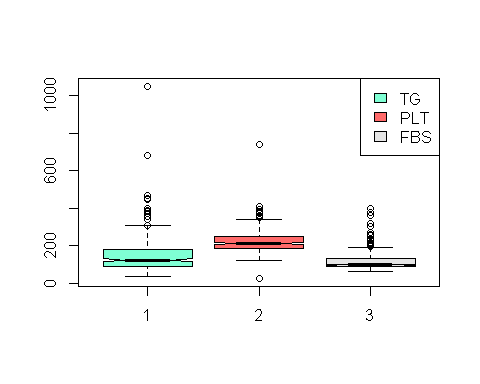
attach(LABB4)  
boxplot(LDL, HDL, BUN, ESR, col = c("cadetblue1", "antiquewhite", "chartreuse", "darkseagreen1"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)  
legend("topright", c("LDL", "HDL", "BUN", "ESR"),fill = c("cadetblue1", "antiquewhite", "chartreuse", "darkseagreen1"))

 # Boxplot of Sodium (Na), Lymphocyte % (Lymph), Neutrophil % (Neut), and Ejection Fraction % (EF):

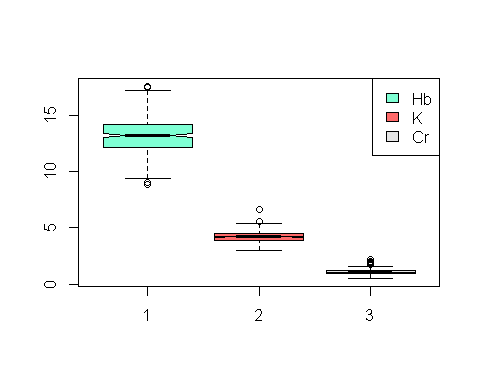
boxplot(Na, Lymph, Neut, EF, col = c("indianred1", "antiquewhite", "chartreuse", "gold"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)  
legend("topright", c("Na", "Lymph", "Neut", "EF"),fill = c("indianred1", "antiquewhite", "chartreuse", "gold"))

 # Boxplot of Triglyceride (TG), Platelet (PLT), and Fasting Blood Sugar (FBS):

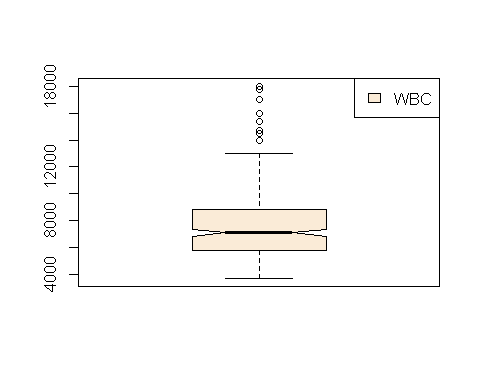
boxplot(TG, PLT, FBS, col = c("aquamarine","indianred1", "gray90"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)  
legend("topright", c("TG", "PLT", "FBS"),fill = c("aquamarine","indianred1", "gray90"))

 # Boxplot of Hemoglobin (Hb), Potassium (K), and Creatine (Cr):

boxplot(Hb, K, Cr, col = c("aquamarine","indianred1", "gray90"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)  
legend("topright", c("Hb", "K", "Cr"),fill = c("aquamarine","indianred1", "gray90"))

 # Boxplot of White Blood Cell (WBC) count:

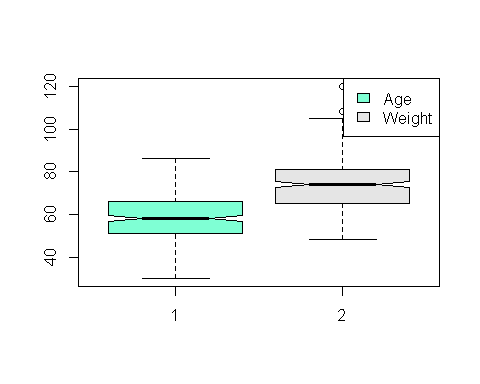
boxplot(WBC, col = c("antiquewhite"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)  
legend("topright", c("WBC"),fill = c("antiquewhite"))

 # Boxplot to show distribution of Age and Weight:

attach(DemoB4)

## The following object is masked from LABB4:  
##   
## CAD

boxplot(Age, Weight, col = c("aquamarine","gray90"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)  
legend("topright", c("Age", "Weight"),fill = c("aquamarine","gray90"))

 # Shapiro tests of normality (H0: Distribution is normal; Reject null if p-value is less than .05). Note: sensitive when n>80. This test shows that ‘Age’ is NOT normally distributed.

library(psych)

## Warning: package 'psych' was built under R version 3.5.3

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

describe(DemoB4$Age)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 303 58.9 10.39 58 58.68 11.86 30 86 56 0.13 -0.44  
## se  
## X1 0.6

AgeNorm <- shapiro.test((DemoB4$Age))  
AgeNorm

##   
## Shapiro-Wilk normality test  
##   
## data: (DemoB4$Age)  
## W = 0.98974, p-value = 0.03162

# Hence, weight is also not normally distributed in this study.

library(psych)  
  
describe(DemoB4$Weight)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 303 73.83 11.99 74 73.44 11.86 48 120 72 0.41 0.3  
## se  
## X1 0.69

WeightNorm <- shapiro.test((DemoB4$Weight))  
WeightNorm

##   
## Shapiro-Wilk normality test  
##   
## data: (DemoB4$Weight)  
## W = 0.98702, p-value = 0.007982

# And BMI is also not normally distributed.

describe(Allv1$BMI)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 303 27.25 4.1 26.78 27.07 3.77 18.12 40.9 22.79 0.43 0.11  
## se  
## X1 0.24

BMINorm <- shapiro.test((Allv1$BMI))  
BMINorm

##   
## Shapiro-Wilk normality test  
##   
## data: (Allv1$BMI)  
## W = 0.98564, p-value = 0.004061

# Can (carefully) use ‘describe’ for categorical variables too (since the psych package recodes categories as numbers):

library(psych)  
describe(FullSetB42[1:17])

## vars n mean sd median trimmed mad min max range skew  
## Age 1 303 58.90 10.39 58 58.68 11.86 30 86 56 0.13  
## Weight 2 303 73.83 11.99 74 73.44 11.86 48 120 72 0.41  
## Sex\* 3 303 1.58 0.49 2 1.60 0.00 1 2 1 -0.33  
## DM 4 303 0.30 0.46 0 0.25 0.00 0 1 1 0.88  
## HTN 5 303 0.59 0.49 1 0.61 0.00 0 1 1 -0.37  
## Smoker 6 303 0.21 0.41 0 0.14 0.00 0 1 1 1.43  
## ExSmoker 7 303 0.03 0.18 0 0.00 0.00 0 1 1 5.20  
## FH 8 303 0.16 0.37 0 0.07 0.00 0 1 1 1.86  
## Obesity\* 9 303 1.70 0.46 2 1.74 0.00 1 2 1 -0.85  
## CRF\* 10 303 1.02 0.14 1 1.00 0.00 1 2 1 6.86  
## CVA\* 11 303 1.02 0.13 1 1.00 0.00 1 2 1 7.55  
## AD\* 12 303 1.04 0.19 1 1.00 0.00 1 2 1 4.93  
## TD\* 13 303 1.02 0.15 1 1.00 0.00 1 2 1 6.32  
## CHF\* 14 303 1.00 0.06 1 1.00 0.00 1 2 1 17.23  
## DLP\* 15 303 1.37 0.48 1 1.34 0.00 1 2 1 0.54  
## BP 16 303 129.55 18.94 130 128.46 14.83 90 190 100 0.57  
## PR 17 303 75.14 8.91 70 74.22 7.41 50 110 60 1.07  
## kurtosis se  
## Age -0.44 0.60  
## Weight 0.30 0.69  
## Sex\* -1.90 0.03  
## DM -1.22 0.03  
## HTN -1.87 0.03  
## Smoker 0.05 0.02  
## ExSmoker 25.15 0.01  
## FH 1.47 0.02  
## Obesity\* -1.28 0.03  
## CRF\* 45.20 0.01  
## CVA\* 55.23 0.01  
## AD\* 22.41 0.01  
## TD\* 38.04 0.01  
## CHF\* 296.02 0.00  
## DLP\* -1.72 0.03  
## BP 0.48 1.09  
## PR 2.33 0.51

# Boxplot to show distribution of Blood Pressure and Pulse Rate

attach(SymptomB4)

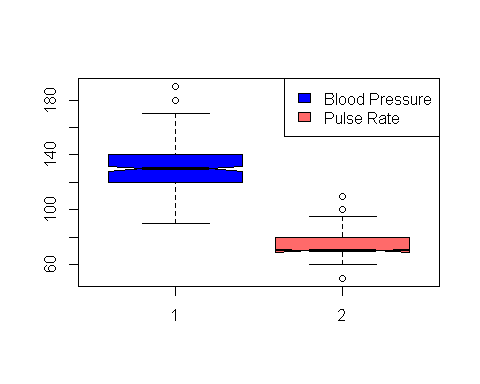
## The following object is masked from DemoB4:  
##   
## CAD

## The following object is masked from LABB4:  
##   
## CAD

boxplot(BP, PR, col = c("blue","indianred1"), notch = TRUE, horizontal = FALSE, outline = TRUE, plot = TRUE)

## Warning in bxp(list(stats = structure(c(90, 120, 130, 140, 170, 60, 70, :  
## some notches went outside hinges ('box'): maybe set notch=FALSE

legend("topright", c("Blood Pressure", "Pulse Rate"),fill = c("blue","indianred1"))

 # Blood Pressure (BP) is not normally distributed:

describe(SymptomB4$BP)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 303 129.55 18.94 130 128.46 14.83 90 190 100 0.57 0.48  
## se  
## X1 1.09

BPNorm <- shapiro.test((SymptomB4$BP))  
BPNorm

##   
## Shapiro-Wilk normality test  
##   
## data: (SymptomB4$BP)  
## W = 0.9543, p-value = 4.039e-08

# Pulse Rate is not normally distributed:

describe(SymptomB4$PR)

## vars n mean sd median trimmed mad min max range skew kurtosis  
## X1 1 303 75.14 8.91 70 74.22 7.41 50 110 60 1.07 2.33  
## se  
## X1 0.51

PRNorm <- shapiro.test((SymptomB4$PR))  
PRNorm

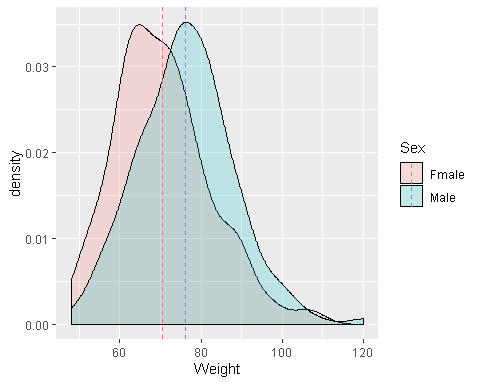
##   
## Shapiro-Wilk normality test  
##   
## data: (SymptomB4$PR)  
## W = 0.86016, p-value = 6.633e-16

# Density plot of weight and sex (before discretization):

library(ggplot2)  
library(dplyr)  
library(plyr)  
  
WeightMu <- ddply(DemoB4, "Sex", summarise, grp.mean=mean(Weight))  
head(WeightMu)

## Sex grp.mean  
## 1 Fmale 70.52756  
## 2 Male 76.21591

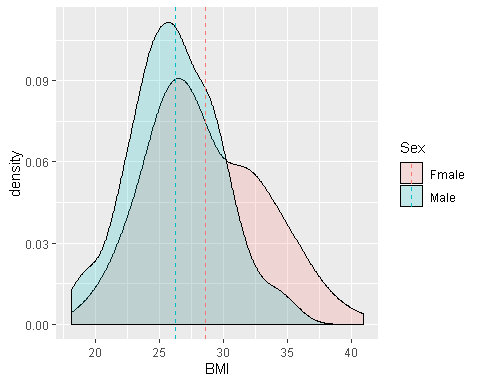
DDen1 <- ggplot(DemoB4, aes(x = Weight, fill = Sex)) +  
 geom\_density(alpha = 0.2)+  
 geom\_vline(data = WeightMu, aes(xintercept=grp.mean, colour = Sex),  
 linetype = "dashed")  
DDen1

 # Density plot of BMI and sex (before discretization):

BMIMu <- ddply(Allv1, "Sex", summarise, grp.mean=mean(BMI))  
head(BMIMu)

## Sex grp.mean  
## 1 Fmale 28.61687  
## 2 Male 26.26082

BMIden <- ggplot(Allv1, aes(x = BMI, fill = Sex)) +  
 geom\_density(alpha = 0.2)+  
 geom\_vline(data = BMIMu, aes(xintercept=grp.mean, colour = Sex),  
 linetype = "dashed")  
BMIden

 # Will be completing a number of ASSOCIATIONS between many categorical variables, starting with a few straight forward investigation into commonly accepted ‘associations’. Interpretation of results when K = 2: SMALL Association: 0.10 - < 0.30; MEDIUM Association: 0.30 - < 0.50; LARGE Association: ??? 0.50

# Starting with Diabetes (DM) and Hypertension (HTN): Shows small association

library(lsr)  
  
AllCatB4 <- Allv2[,c(3:15, 18:24,26:35)]  
cramersV(AllCatB4$DM, AllCatB4$HTN)

## [1] 0.2105191

# Ex-Smoker and Obesity shows no association:

cramersV(AllCatB4$ExSmoker,AllCatB4$Obesity)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.02154688

# Ex-Smoker and Obesity shows no association:

cramersV(AllCatB4$DM,AllCatB4$Obesity)

## [1] 0.01298559

# Hypertensin and current smoker has a small association:

cramersV(AllCatB4$HTN,AllCatB4$Smoker)

## [1] 0.1607305

# Hypertension and Stroke (CVA) has no association:

cramersV(AllCatB4$HTN,AllCatB4$CVA)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.02877846

# Congestive Heart Failure (CHF) and Obsesity has no association (see below):

cramersV(AllCatB4$CHF,AllCatB4$Obesity)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 1.5974e-15

# Determining whether there is any significant variation in the ‘NCP’ column: Very low variation (94.7% of NCP entries are ‘No’)

NCPn <- sum(FullSetB42$NCP == 'N')  
NCPy <- sum(FullSetB42$NCP == 'Y')  
(NCPn/(NCPn+NCPy))\*100

## [1] 94.71947

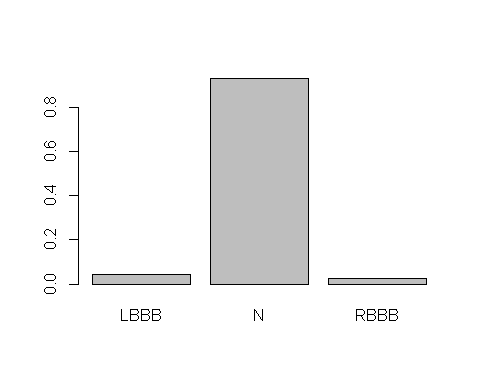
# Determining whether there is any significant variation in the ‘ACP’ column: 69.3% of these ‘ACP’ entries are ‘No’

ACPn <- sum(FullSetB42$ACP == 'N')  
ACPy <- sum(FullSetB42$ACP == 'Y')  
(ACPn/(ACPn+ACPy))\*100

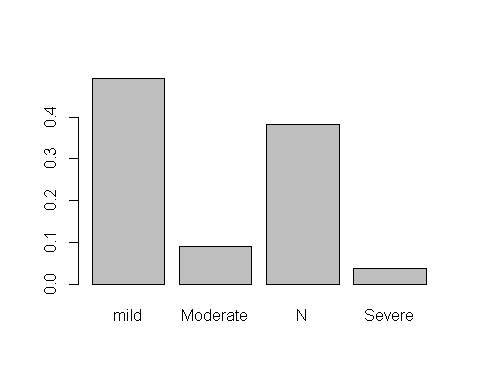
## [1] 69.30693

# Barplot of BBB attribute

barplot(prop.table(table(FullSetB42$BBB)))

 # Barplot of VHD attribute

barplot(prop.table(table(FullSetB42$VHD)))



# Feature selection on all the attributes combined, as well as on the 4 individual attribute groups - using the FSelector package:

# Starting with the full data set - selected: CAD ~ TCP + ACP + RWMA + EF + Age + HTN + NCP + DM + Tinv + VHD

library(FSelector)

## Warning: package 'FSelector' was built under R version 3.5.3

AllWeights <- chi.squared(CAD~.,AllvFinal)  
print(AllWeights)

## attr\_importance  
## Age 0.290317717  
## Weight 0.053693554  
## Sex 0.067041240  
## DM 0.252897001  
## HTN 0.287760932  
## Smoker 0.073503651  
## ExSmoker 0.035578171  
## FH 0.035605456  
## Obesity 0.022461339  
## CRF 0.090204905  
## CVA 0.024945620  
## AD 0.084178945  
## TD 0.048077279  
## CHF 0.036519875  
## DLP 0.012718268  
## BP 0.135498540  
## PR 0.073888082  
## Edema 0.054069268  
## WPP 0.082207155  
## LR 0.045178608  
## SysM 0.004856462  
## DiaM 0.146777311  
## TCP 0.542966708  
## Dyspnea 0.125210904  
## Fclass 0.085918550  
## ACP 0.415921817  
## NCP 0.274183618  
## LTAng 0.051732623  
## QWave 0.149848291  
## STelev 0.139684381  
## STdep 0.144425871  
## Tinv 0.236932878  
## LVH 0.051196374  
## PoorR 0.111040227  
## BBB 0.088749817  
## FBS 0.229347982  
## Cr 0.102491681  
## TG 0.086830465  
## LDL 0.014501583  
## HDL 0.015804598  
## BUN 0.091662224  
## ESR 0.065213016  
## Hb 0.048994402  
## K 0.069632280  
## Na 0.067555843  
## WBC 0.105345858  
## Lymph 0.129137722  
## Neut 0.106735676  
## PLT 0.100855982  
## EF 0.329842476  
## RWMA 0.334812718  
## VHD 0.231208043

subset <- cutoff.k(AllWeights, 10)  
FSall <- as.simple.formula(subset, "CAD")  
print(FSall)

## CAD ~ TCP + ACP + RWMA + EF + Age + HTN + NCP + DM + Tinv + VHD  
## <environment: 0x00000000253615c0>

# Next, using the ‘Demo’ (Demographic) attribute data set - results: CAD ~ Age + HTN + DM + CRF + AD

DemoWeights <- chi.squared(CAD~.,Demo)  
print(DemoWeights)

## attr\_importance  
## Age 0.29031772  
## Weight 0.05369355  
## Sex 0.06704124  
## DM 0.25289700  
## HTN 0.28776093  
## Smoker 0.07350365  
## ExSmoker 0.03557817  
## FH 0.03560546  
## Obesity 0.02246134  
## CRF 0.09020490  
## CVA 0.02494562  
## AD 0.08417894  
## TD 0.04807728  
## CHF 0.03651987  
## DLP 0.01271827

subset <- cutoff.k(DemoWeights, 5)  
FSdemo <- as.simple.formula(subset, "CAD")  
print(FSdemo)

## CAD ~ Age + HTN + DM + CRF + AD  
## <environment: 0x000000001e57f270>

# Next, using the ‘Symptom’ (Symptom & Examination) attribute data set - results: CAD ~ TCP + ACP + NCP + DiaM + BP

SymWeights <- chi.squared(CAD~.,Symptom)  
print(SymWeights)

## attr\_importance  
## BP 0.135498540  
## PR 0.073888082  
## Edema 0.054069268  
## WPP 0.082207155  
## LR 0.045178608  
## SysM 0.004856462  
## DiaM 0.146777311  
## TCP 0.542966708  
## Dyspnea 0.125210904  
## Fclass 0.085918550  
## ACP 0.415921817  
## NCP 0.274183618  
## LTAng 0.051732623

subset <- cutoff.k(SymWeights, 5)  
FSsym <- as.simple.formula(subset, "CAD")  
print(FSsym)

## CAD ~ TCP + ACP + NCP + DiaM + BP  
## <environment: 0x00000000206f1738>

# Next, using the ’ECG’attribute data set - results: CAD ~ Tinv + QWave + STdep + STelev + PoorR

ECGWeights <- chi.squared(CAD~.,ECG)  
print(ECGWeights)

## attr\_importance  
## QWave 0.14984829  
## STelev 0.13968438  
## STdep 0.14442587  
## Tinv 0.23693288  
## LVH 0.05119637  
## PoorR 0.11104023  
## BBB 0.08874982

subset <- cutoff.k(ECGWeights, 5)  
FSecg <- as.simple.formula(subset, "CAD")  
print(FSecg)

## CAD ~ Tinv + QWave + STdep + STelev + PoorR  
## <environment: 0x000000002155fd58>

# Lastly, using the ‘LAB’ (Laboratory and Echocardiographic) attribute data set - results: CAD ~ RWMA + EF + VHD + FBS + Lymph

LABWeights <- chi.squared(CAD~.,LAB)  
print(LABWeights)

## attr\_importance  
## FBS 0.22934798  
## Cr 0.10249168  
## TG 0.08683046  
## LDL 0.01450158  
## HDL 0.01580460  
## BUN 0.09166222  
## ESR 0.06521302  
## Hb 0.04899440  
## K 0.06963228  
## Na 0.06755584  
## WBC 0.10534586  
## Lymph 0.12913772  
## Neut 0.10673568  
## PLT 0.10085598  
## EF 0.32984248  
## RWMA 0.33481272  
## VHD 0.23120804

subset <- cutoff.k(LABWeights, 5)  
FSlab <- as.simple.formula(subset, "CAD")  
print(FSlab)

## CAD ~ RWMA + EF + VHD + FBS + Lymph  
## <environment: 0x0000000023820d20>

# Now, comparing these discretized results with those from before discretization:

# Using the full data set (FullSetB42) - results: CAD ~ TCP + ACP + RWMA + EF + Age + HTN + *BP* + NCP + DM + *FBS* (while the discretized results were: CAD ~ TCP + ACP + RWMA + EF + Age + HTN + NCP + DM + Tinv + VHD). NOTE: The top 6 were the same for both data sets (and 8 of the top 10 were shared across both data sets)

AllB4Weights <- chi.squared(CAD~.,FullSetB42)  
print(AllB4Weights)

## attr\_importance  
## Age 0.328281473  
## Weight 0.000000000  
## Sex 0.067041240  
## DM 0.252897001  
## HTN 0.287760932  
## Smoker 0.000000000  
## ExSmoker 0.000000000  
## FH 0.000000000  
## Obesity 0.022461339  
## CRF 0.090204905  
## CVA 0.024945620  
## AD 0.084178945  
## TD 0.048077279  
## CHF 0.036519875  
## DLP 0.012718268  
## BP 0.275203095  
## PR 0.000000000  
## Edema 0.000000000  
## WPP 0.082207155  
## LR 0.045178608  
## SysM 0.004856462  
## DiaM 0.146777311  
## TCP 0.542966708  
## Dyspnea 0.125210904  
## Fclass 0.000000000  
## ACP 0.415921817  
## NCP 0.274183618  
## LTAng 0.051732623  
## QWave 0.000000000  
## STelev 0.000000000  
## STdep 0.000000000  
## Tinv 0.236932878  
## LVH 0.051196374  
## PoorR 0.111040227  
## BBB 0.088749817  
## FBS 0.238635213  
## Cr 0.000000000  
## TG 0.000000000  
## LDL 0.000000000  
## HDL 0.000000000  
## BUN 0.000000000  
## ESR 0.000000000  
## Hb 0.000000000  
## K 0.000000000  
## Na 0.000000000  
## WBC 0.000000000  
## Lymph 0.000000000  
## Neut 0.000000000  
## PLT 0.000000000  
## EF 0.329842476  
## RWMA 0.334812718  
## VHD 0.231208043

subset <- cutoff.k(AllB4Weights, 10)  
FSB4all <- as.simple.formula(subset, "CAD")  
print(FSB4all)

## CAD ~ TCP + ACP + RWMA + EF + Age + HTN + BP + NCP + DM + FBS  
## <environment: 0x000000002020fa60>

# Comparing the discretized feature selection results found using the ‘FSelector’ package with the results using the ‘FSelectorRcpp’ package - results: CAD ~ TCP + ACP + RWMA + EF + HTN + Age + DM + NCP + Tinv + FBS

library(FSelectorRcpp)  
library(tidyverse)  
  
FS2all <- information\_gain(  
 formula = CAD~.,  
 data = AllvFinal,  
 type = "infogain",  
 threads = 2  
) %>%  
cut\_attrs(  
 k = 10  
) %>%  
to\_formula(  
 attrs = .,  
 class = "CAD"  
)  
print(FS2all)

## CAD ~ TCP + ACP + RWMA + EF + HTN + Age + DM + NCP + Tinv + FBS  
## <environment: 0x00000000207ac6f0>

# Lastly, comparing the output of the FSelectorRcpp package using the data set that was not discretized yet (FullSetB42 data set) - result: CAD ~ TCP + ACP + RWMA + Age + EF + HTN + DM + BP + NCP + Tinv

FS2B4all <- information\_gain(  
 formula = CAD~.,  
 data = FullSetB42,  
 type = "infogain",  
 threads = 2  
) %>%  
cut\_attrs(  
 k = 10  
) %>%  
to\_formula(  
 attrs = .,  
 class = "CAD"  
)  
print(FS2B4all)

## CAD ~ TCP + ACP + RWMA + Age + EF + HTN + DM + BP + NCP + Tinv  
## <environment: 0x0000000020b22bb0>

# Checking on strength of associations between top 12 attributes and the dependent variable, ranked in descending order:

library(lsr)  
  
cramersV(AllvFinal$TCP, AllvFinal$CAD)

## [1] 0.5356469

cramersV(AllvFinal$ACP, AllvFinal$CAD)

## [1] 0.4080137

cramersV(AllvFinal$RWMA, AllvFinal$CAD)

## [1] 0.3267227

cramersV(AllvFinal$EF, AllvFinal$CAD)

## [1] 0.3221946

cramersV(AllvFinal$Age, AllvFinal$CAD)

## [1] 0.2814322

cramersV(AllvFinal$HTN, AllvFinal$CAD)

## [1] 0.2803429

cramersV(AllvFinal$NCP, AllvFinal$CAD)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.2578747

cramersV(AllvFinal$DM, AllvFinal$CAD)

## [1] 0.2449149

cramersV(AllvFinal$VHD, AllvFinal$CAD)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.231208

cramersV(AllvFinal$FBS, AllvFinal$CAD)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.229348

cramersV(AllvFinal$Tinv, AllvFinal$CAD)

## [1] 0.2289508

cramersV(AllvFinal$BP, AllvFinal$CAD)

## [1] 0.1255092

# Checking on the strength of associations between the top 12 attributes and each other:

cramersV(AllvFinal$TCP, AllvFinal$ACP)

## [1] 0.7156668

cramersV(AllvFinal$HTN, AllvFinal$BP)

## [1] 0.3519147

cramersV(AllvFinal$RWMA, AllvFinal$EF)

## [1] 0.269923

cramersV(AllvFinal$TCP, AllvFinal$NCP)

## [1] 0.2416607

cramersV(AllvFinal$HTN, AllvFinal$DM)

## [1] 0.2105191

cramersV(AllvFinal$VHD, AllvFinal$Age)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.1636751

cramersV(AllvFinal$RWMA, AllvFinal$TCP)

## [1] 0.1608916

cramersV(AllvFinal$HTN, AllvFinal$VHD)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.1379553

cramersV(AllvFinal$Tinv, AllvFinal$EF)

## [1] 0.1360726

cramersV(AllvFinal$TCP, AllvFinal$Age)

## [1] 0.1078061

cramersV(AllvFinal$RWMA, AllvFinal$Age)

## [1] 0.1060843

cramersV(AllvFinal$HTN, AllvFinal$Age)

## [1] 0.09646362

# Checking associations between the final variables that do not have a statistically significant relationship with the dependent variable (per the logistic regression results later). That is, checkint ACP and NCP against each other, and the other variables, including TCP, EF, RWMA, HTN, Age and DM (as well as the dependent variable). NOTE: When DF=1, the interpretation of the Cramer’s V results (effect) would be as follows: (0.10 = small effect) (0.30 = medium effect) (0.50=large effect)

library(lsr)  
  
cramersV(AllvFinal$ACP, AllvFinal$TCP)

## [1] 0.7156668

cramersV(AllvFinal$ACP, AllvFinal$EF)

## [1] 0.1645235

cramersV(AllvFinal$ACP, AllvFinal$RWMA)

## [1] 0.1729343

cramersV(AllvFinal$ACP, AllvFinal$HTN)

## [1] 0.1373895

cramersV(AllvFinal$ACP, AllvFinal$Age)

## [1] 0.1141707

cramersV(AllvFinal$ACP, AllvFinal$DM)

## [1] 0.08023603

cramersV(AllvFinal$ACP, AllvFinal$CAD)

## [1] 0.4080137

cramersV(AllvFinal$NCP, AllvFinal$TCP)

## [1] 0.2416607

cramersV(AllvFinal$NCP, AllvFinal$EF)

## [1] 0.05887248

cramersV(AllvFinal$NCP, AllvFinal$RWMA)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.0668128

cramersV(AllvFinal$NCP, AllvFinal$HTN)

## [1] 0.001436249

cramersV(AllvFinal$NCP, AllvFinal$Age)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.03838205

cramersV(AllvFinal$NCP, AllvFinal$DM)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.07274291

cramersV(AllvFinal$NCP, AllvFinal$CAD)

## Warning in chisq.test(...): Chi-squared approximation may be incorrect

## [1] 0.2578747

# Creating the final variable data set using the FSelector package:

TenVars <- AllvFinal[, c("TCP", "ACP", "RWMA", "EF", "Age", "HTN", "NCP", "DM", "Tinv", "VHD", "CAD")]  
#TenVars

# Creating a second final variable data set using the FSelectorRcpp package:

TenVars2 <- AllvFinal[, c("TCP", "ACP", "RWMA", "EF", "HTN", "Age", "DM", "NCP", "Tinv", "FBS", "CAD")]  
#TenVars2

# Using createDataPartition() to split data into training and test sets (75% and 25%). Since the outcome variable (CAD) is categorical, this function will make sure that the distribution of outcome variable classes will be similar in both of the sets.

library(caret)  
  
index <- createDataPartition(AllvFinal$CAD, p = 0.75, list = FALSE)  
AllvFtrain <- AllvFinal[ index,]  
AllvFtest <- AllvFinal[-index,]

#str(AllvFtrain)

# Using ‘Recursive Feature Elimination’ to find the best subset of atttributes to use for modeling.

library(plyr)  
  
control <- rfeControl(functions = rfFuncs,  
 method = "repeatedcv",  
 repeats = 10,  
 verbose = FALSE)  
outcomeName <- 'CAD'  
predictors <- names(AllvFtrain)[!names(AllvFtrain) %in% outcomeName]  
CAD\_Profile <- rfe(AllvFtrain[,predictors], AllvFtrain[,outcomeName],  
 rfeControl = control)  
  
CAD\_Profile

##   
## Recursive feature selection  
##   
## Outer resampling method: Cross-Validated (10 fold, repeated 10 times)   
##   
## Resampling performance over subset size:  
##   
## Variables Accuracy Kappa AccuracySD KappaSD Selected  
## 4 0.8223 0.5882 0.06566 0.1576   
## 8 0.8133 0.5330 0.06509 0.1654   
## 16 0.8324 0.5906 0.07064 0.1704   
## 52 0.8683 0.6691 0.06552 0.1711 \*  
##   
## The top 5 variables (out of 52):  
## TCP, RWMA, ACP, EF, Tinv

CAD\_Profile$optVariables[1:8]

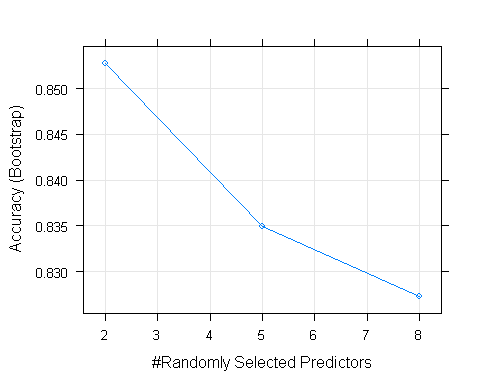
## [1] "TCP" "RWMA" "ACP" "EF" "Tinv" "NCP" "HTN" "Age"

# Taking the top 8 predictors:

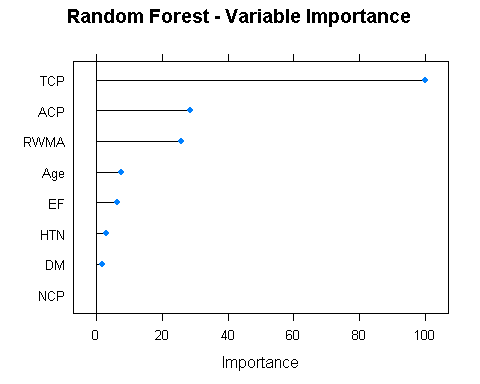
predictors <-c("TCP", "EF", "RWMA", "NCP", "ACP", "HTN", "Age", "DM")

# Random Forest Training

fitCntrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10, sampling = "smote")  
rfFit <- train(AllvFtrain[,predictors], AllvFtrain[,outcomeName], method = 'rf', returnResamp = "final")  
rfFit  
plot(rfFit)



varImp(object = rfFit)  
plot(varImp(object = rfFit),main="Random Forest - Variable Importance")



# Random Forest Predictions

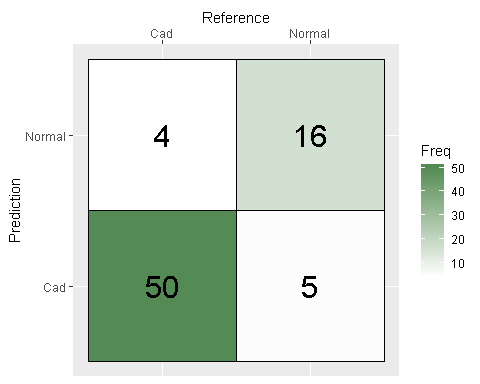
RFpredictions <- predict.train(object = rfFit, AllvFtest[,predictors], type = "raw")  
table(RFpredictions)

## RFpredictions  
## Cad Normal   
## 55 20

confusionMatrix(RFpredictions, AllvFtest[,outcomeName])

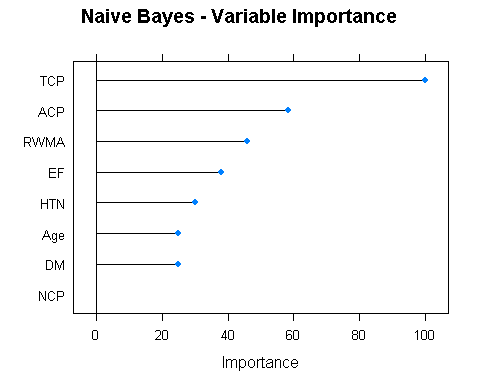
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Cad Normal  
## Cad 50 5  
## Normal 4 16  
##   
## Accuracy : 0.88   
## 95% CI : (0.7844, 0.9436)  
## No Information Rate : 0.72   
## P-Value [Acc > NIR] : 0.0007567   
##   
## Kappa : 0.698   
##   
## Mcnemar's Test P-Value : 1.0000000   
##   
## Sensitivity : 0.9259   
## Specificity : 0.7619   
## Pos Pred Value : 0.9091   
## Neg Pred Value : 0.8000   
## Prevalence : 0.7200   
## Detection Rate : 0.6667   
## Detection Prevalence : 0.7333   
## Balanced Accuracy : 0.8439   
##   
## 'Positive' Class : Cad   
##

RFcm <- confusionMatrix(RFpredictions, AllvFtest[,outcomeName])  
  
RFcm$table %>%  
 data.frame() %>%   
 mutate(Prediction = factor(Prediction, levels = c("Cad", "Normal"))) %>%  
 group\_by(Reference) %>%   
 mutate(total = sum(Freq)) %>%   
 ungroup() %>%   
 ggplot(aes(Reference, Prediction, fill = Freq)) +  
 geom\_tile() +  
 geom\_text(aes(label = Freq), size = 8) +  
 scale\_fill\_gradient(low = "white", high = "palegreen4") +  
 scale\_x\_discrete(position = "top") +  
 geom\_tile(color = "black", fill = "black", alpha = 0)



# Training for Naive Bayes:

library(klaR)  
  
NBfitCntrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10, sampling = "smote")  
NBfit <- train(AllvFtrain[,predictors], AllvFtrain[,outcomeName], method = 'nb', trControl= NBfitCntrl, returnResamp = "final")  
NBfit  
  
varImp(object = NBfit)  
plot(varImp(object = NBfit),main="Naive Bayes - Variable Importance")



# Naive Bayes Predictions:

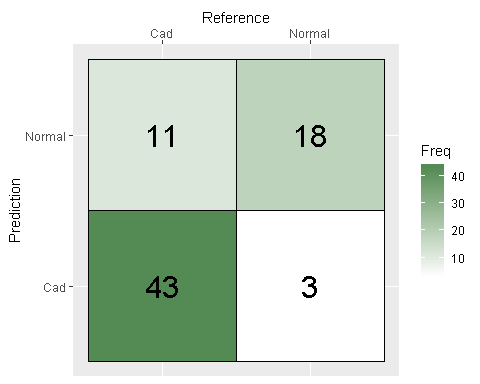
NBpredictions <- predict.train(object = NBfit, AllvFtest[,predictors])  
table(NBpredictions)

## NBpredictions  
## Cad Normal   
## 46 29

confusionMatrix(NBpredictions, AllvFtest[,outcomeName])

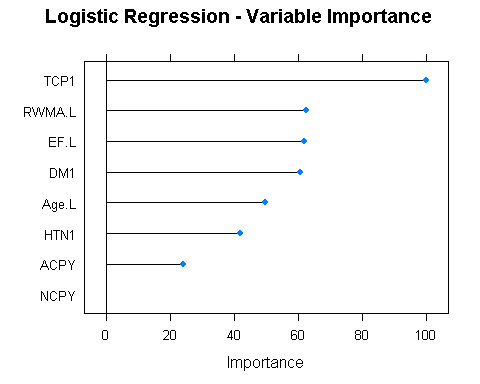
## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Cad Normal  
## Cad 43 3  
## Normal 11 18  
##   
## Accuracy : 0.8133   
## 95% CI : (0.7067, 0.894)  
## No Information Rate : 0.72   
## P-Value [Acc > NIR] : 0.04330   
##   
## Kappa : 0.5853   
##   
## Mcnemar's Test P-Value : 0.06137   
##   
## Sensitivity : 0.7963   
## Specificity : 0.8571   
## Pos Pred Value : 0.9348   
## Neg Pred Value : 0.6207   
## Prevalence : 0.7200   
## Detection Rate : 0.5733   
## Detection Prevalence : 0.6133   
## Balanced Accuracy : 0.8267   
##   
## 'Positive' Class : Cad   
##

NBcm <- confusionMatrix(NBpredictions, AllvFtest[,outcomeName])  
  
NBcm$table %>%  
 data.frame() %>%   
 mutate(Prediction = factor(Prediction, levels = c("Cad", "Normal"))) %>%  
 group\_by(Reference) %>%   
 mutate(total = sum(Freq)) %>%   
 ungroup() %>%   
 ggplot(aes(Reference, Prediction, fill = Freq)) +  
 geom\_tile() +  
 geom\_text(aes(label = Freq), size = 8) +  
 scale\_fill\_gradient(low = "white", high = "palegreen4") +  
 scale\_x\_discrete(position = "top") +  
 geom\_tile(color = "black", fill = "black", alpha = 0)



# Training for Logistic Regression:

LRfitCntrl <- trainControl(method = "repeatedcv", number = 10, repeats = 10, sampling = "smote")  
LRFit <- train(AllvFtrain[,predictors], AllvFtrain[,outcomeName], method = 'glm', family = "binomial", trControl= LRfitCntrl)  
LRFit  
  
varImp(object = LRFit)  
plot(varImp(object = LRFit), main="Logistic Regression - Variable Importance")



# Logistic Regression Predictions:

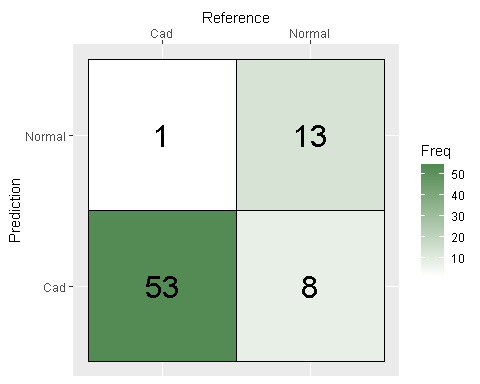
library(tidyverse)  
  
LRpredictions <- predict.train(object = LRFit, AllvFtest[,predictors])  
table(LRpredictions)

## LRpredictions  
## Cad Normal   
## 61 14

confusionMatrix(LRpredictions, AllvFtest[,outcomeName])

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Cad Normal  
## Cad 53 8  
## Normal 1 13  
##   
## Accuracy : 0.88   
## 95% CI : (0.7844, 0.9436)  
## No Information Rate : 0.72   
## P-Value [Acc > NIR] : 0.0007567   
##   
## Kappa : 0.6686   
##   
## Mcnemar's Test P-Value : 0.0455003   
##   
## Sensitivity : 0.9815   
## Specificity : 0.6190   
## Pos Pred Value : 0.8689   
## Neg Pred Value : 0.9286   
## Prevalence : 0.7200   
## Detection Rate : 0.7067   
## Detection Prevalence : 0.8133   
## Balanced Accuracy : 0.8003   
##   
## 'Positive' Class : Cad   
##

LRcm <- confusionMatrix(LRpredictions, AllvFtest[,outcomeName])  
  
LRcm$table %>%  
 data.frame() %>%   
 mutate(Prediction = factor(Prediction, levels = c("Cad", "Normal"))) %>%  
 group\_by(Reference) %>%   
 mutate(total = sum(Freq)) %>%   
 ungroup() %>%   
 ggplot(aes(Reference, Prediction, fill = Freq)) +  
 geom\_tile() +  
 geom\_text(aes(label = Freq), size = 8) +  
 scale\_fill\_gradient(low = "white", high = "palegreen4") +  
 scale\_x\_discrete(position = "top") +  
 geom\_tile(color = "black", fill = "black", alpha = 0)

 # Brier Score for Logistic Regression model:

# Ensamble evaluation:

#library(caretEnsemble)  
  
#ENCntrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3, search = "grid", savePredictions = "final", sampling = "smote")  
  
#ENFit <- caretList(AllvFtrain[,predictors], AllvFtrain[,outcomeName], trControl= ENCntrl, methodList = c("rf", "nb", "glm"))  
#ENFit  
  
#ENres <- resamples(ENFit)  
#summary(ENFit)

# Model comparisons:

#ModelCompare <- resamples(list(RF = rfFit, NB = NBfit, LR = LRFit))  
#ModelCompare$values  
#summary(ModelCompare)

# Checking on the direction of log odds for a one unit increase in the selected features (using logistic regression) to confirm the relationship between the independent variables and the dependent variable (CAD or Normal):

logiTrain <- data.frame(AllvFtrain[,c(23,51,27,26,50,1,5,4,53)])  
  
logiTrain$CAD <- relevel(logiTrain$CAD, ref = "Normal")  
str(logiTrain)

## 'data.frame': 228 obs. of 9 variables:  
## $ TCP : Factor w/ 2 levels "0","1": 2 1 1 2 2 2 2 1 1 1 ...  
## $ RWMA: Ord.factor w/ 2 levels "Med"<"High": 2 1 1 2 2 2 2 2 1 1 ...  
## $ NCP : Factor w/ 2 levels "N","Y": 1 2 1 1 1 1 1 1 1 1 ...  
## $ ACP : Factor w/ 2 levels "N","Y": 1 1 1 1 1 1 1 2 1 2 ...  
## $ EF : Ord.factor w/ 2 levels "Low"<"Med": 1 2 1 1 1 1 1 1 1 2 ...  
## $ Age : Ord.factor w/ 2 levels "Med"<"High": 2 2 1 2 2 2 2 2 1 2 ...  
## $ HTN : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 2 2 1 1 ...  
## $ DM : Factor w/ 2 levels "0","1": 1 1 1 1 2 2 1 2 1 1 ...  
## $ CAD : Factor w/ 2 levels "Normal","Cad": 2 1 1 2 2 2 2 2 1 1 ...

LRlogit <- glm(CAD~.,logiTrain, family = "binomial")  
summary(LRlogit)

##   
## Call:  
## glm(formula = CAD ~ ., family = "binomial", data = logiTrain)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.6803 -0.3087 0.1004 0.4065 2.1538   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.8431 0.7159 -1.178 0.238885   
## TCP1 3.3688 0.7652 4.403 1.07e-05 \*\*\*  
## RWMA.L 2.1698 0.6395 3.393 0.000691 \*\*\*  
## NCPY -0.8037 0.9566 -0.840 0.400847   
## ACPY 0.2305 0.6694 0.344 0.730545   
## EF.L -1.0135 0.3509 -2.888 0.003873 \*\*   
## Age.L 0.9157 0.3825 2.394 0.016658 \*   
## HTN1 1.2086 0.4692 2.576 0.010000 \*   
## DM1 1.7244 0.5743 3.003 0.002676 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 274.37 on 227 degrees of freedom  
## Residual deviance: 127.14 on 219 degrees of freedom  
## AIC: 145.14  
##   
## Number of Fisher Scoring iterations: 7

# Explanatin of the order of factor variables for the logistic regression output in R: R will use the alphabetical order of the variable results - and choose the letter that is higher in the alphabet as the reference variable (which is typically coded automatically with a ‘1’, versus ‘0’ - or a ‘2’ vs ‘1’ on the str() output). As such, ‘Normal’ would be the reference variable (equal to ‘2’ on the str() output.). However, ‘CAD’ should be the reference variable and so I have ‘re-leveled’ that attribute so that ‘CAD’ is used as the reference (and as such, is showing as a ‘2’ on the str() output).

# Other variables explained for the glm output:

# TCP: ‘Yes’, or ‘1’, is base case (hence shows ‘2’ on str() output.

# ACP: Same base case behaviour as TCP. This happens automatically because ‘Y’ comes after ‘N’ in the alphabet. ‘Y’ is showing with a ‘2’ on the str() output.

# RWMA: ‘High’ is the base case and is the ‘2’ on str() output. This factor has been ‘ordered’ so that even though the ‘H’ in ‘High’ comes before the ‘M’ in ‘Med’, High is treated as the base case (and is assigned the ‘2’ on the str() output.

# EF is also an ‘ordered factor’, and so it has ‘Med’ as the base case (and is the ‘2’ on the str() output).

# NCP: Is the same as ACP (for ‘Y’ and ‘N’, and ‘Y’ being the base case)

# Age: Is treated the same as RWMA - so, ‘High’ is the base case.

# HTN: Is treated the same as TCP

# DM: Is treated the same as TCP

# CAD: Releveled so that CAD is the reference (base case); as such the str() output shows ‘CAD’ as ‘2’, even though ‘C’ comes before ‘N’ (for ‘Normal’) in the alphabet.

# GLM Output interpretation:

# Note: The ACP result is not connected to the dependent variable in a statistically significant way. (This needs more investigation.)

# TCP: For a one unit increase in TCP, the log odds of being diagnosed with Coronary Artery Disease increases by 3.30238

# ACP: For a one unit increase in ACP, the log odds of being diagnosed with Coronary Artery Disease increases by 0.05199 (*Note: Not statistically significant relationship, per the p-value) # RWMA: For a one unit increase in RWMA, the log odds of being diagnosed with Coronary Artery Disease increases by 1.67934 # EF: For a one unit increase in EF, the log odds of being diagnosed with Coronary Artery Disease DECREASES by 1.02024 # NCP: For a one unit increase in NCP, the log odds of being diagnosed with Coronary Artery Disease DECREASES by 1.78913 (*Note: not statistically significant, just outside of significance range per the p-value)

# Age: For a one unit increase in Age, the log odds of being diagnosed with Coronary Artery Disease increases by 1.38648

# HTN: For a one unit increase in HTN, the log odds of being diagnosed with Coronary Artery Disease increases by 1.22778

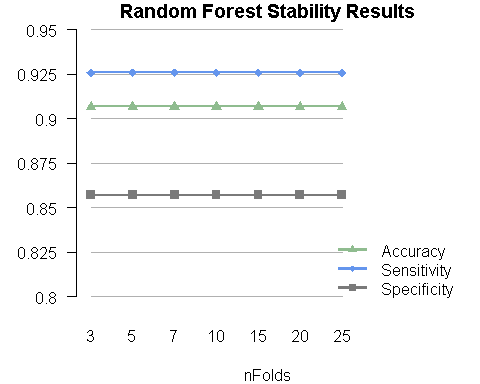
# DM: For a one unit increase in DM, the log odds of being diagnosed with Coronary Artery Disease increases by 1.54412

# Documenting and plotting the stability of the algorithms using different numbers of ‘folds’ during cross validation - starting with the Random Forest algorithm:

RFstability <- data.frame(Accuracy=rep(0.9067,7), Sensitivity=rep(0.9259, 7), Specificity=rep(0.8571,7))  
rownames(RFstability) <- c(3,5,7,10,15,20,25)  
RFstability

## Accuracy Sensitivity Specificity  
## 3 0.9067 0.9259 0.8571  
## 5 0.9067 0.9259 0.8571  
## 7 0.9067 0.9259 0.8571  
## 10 0.9067 0.9259 0.8571  
## 15 0.9067 0.9259 0.8571  
## 20 0.9067 0.9259 0.8571  
## 25 0.9067 0.9259 0.8571

cols <- c("darkseagreen","cornflowerblue","gray48")  
pch <- c(17,18,15)  
xmax <- nrow(RFstability) + 2.5  
par(mar=c(4,4,1,1))  
plot(1:nrow(RFstability), 1:nrow(RFstability), pch="",   
 xlab="nFolds", ylab=NA, xaxt="n", yaxt="n",   
 ylim=c(.8,.95), bty="n", xlim=c(1,xmax), main="Random Forest Stability Results")  
for (i in seq(.8,.95,by=.025)) {  
 lines(1:nrow(RFstability), rep(i,nrow(RFstability)), col="gray69")  
}  
for (i in 1:ncol(RFstability)) {  
  
 points(1:nrow(RFstability), RFstability[,i], pch=pch[i],   
 col=cols[i], cex=1.2)  
  
 lines(1:nrow(RFstability), RFstability[,i], col=cols[i],   
 lwd=2)  
}  
axis(side=1, at=1:nrow(RFstability), tick=FALSE,   
 labels=rownames(RFstability))  
  
axis(side=1, at=seq(-0.5,8.5,by=1),   
 tick=FALSE, labels=NA)  
  
axis(side=2, at=seq(.8,.95,by=.025), tick=TRUE,   
 las=TRUE, labels=paste(seq(.8,.95,by=.025)))  
  
legend('bottomright',legend=colnames(RFstability), pch=pch,   
 col=cols, cex=1.0, bty="n", lwd=3, lty=1)

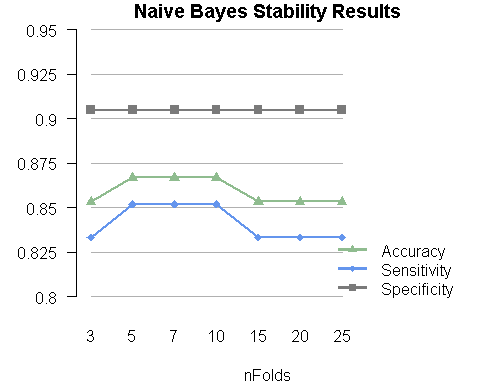


# Next, plotting the stability of the Naive Bayes algorithm:

RFstability <- data.frame(Accuracy=c(0.8533, 0.8667, 0.8667, 0.8667, 0.8533, 0.8533, 0.8533 ), Sensitivity=c(0.8333, 0.8519, 0.8519, 0.8519, 0.8333, 0.8333, 0.8333 ), Specificity=rep(0.9048, 7))  
rownames(RFstability) <- c(3,5,7,10,15,20,25)  
RFstability

## Accuracy Sensitivity Specificity  
## 3 0.8533 0.8333 0.9048  
## 5 0.8667 0.8519 0.9048  
## 7 0.8667 0.8519 0.9048  
## 10 0.8667 0.8519 0.9048  
## 15 0.8533 0.8333 0.9048  
## 20 0.8533 0.8333 0.9048  
## 25 0.8533 0.8333 0.9048

cols <- c("darkseagreen","cornflowerblue","gray48")  
pch <- c(17,18,15)  
xmax <- nrow(RFstability) + 2.5  
par(mar=c(4,4,1,1))  
plot(1:nrow(RFstability), 1:nrow(RFstability), pch="",   
 xlab="nFolds", ylab=NA, xaxt="n", yaxt="n",   
 ylim=c(.8,.95), bty="n", xlim=c(1,xmax), main="Naive Bayes Stability Results")  
for (i in seq(.8,.95,by=.025)) {  
 lines(1:nrow(RFstability), rep(i,nrow(RFstability)), col="gray69")  
}  
for (i in 1:ncol(RFstability)) {  
  
 points(1:nrow(RFstability), RFstability[,i], pch=pch[i],   
 col=cols[i], cex=1.2)  
  
 lines(1:nrow(RFstability), RFstability[,i], col=cols[i],   
 lwd=2)  
}  
axis(side=1, at=1:nrow(RFstability), tick=FALSE,   
 labels=rownames(RFstability))  
  
axis(side=1, at=seq(-0.5,8.5,by=1),   
 tick=FALSE, labels=NA)  
  
axis(side=2, at=seq(.8,.95,by=.025), tick=TRUE,   
 las=TRUE, labels=paste(seq(.8,.95,by=.025)))  
  
legend('bottomright',legend=colnames(RFstability), pch=pch,   
 col=cols, cex=1.0, bty="n", lwd=3, lty=1)



# Lastly, plotting the stability of the Logistic Regression algorithm:

RFstability <- data.frame(Accuracy=c(0.8533, 0.8800, 0.8933, 0.8933, 0.9067, 0.8933, 0.8533), Sensitivity=c(0.9074, 0.9074, 0.9074, 0.9074, 0.9259, 0.9259, 0.9074), Specificity=c(0.7143, 0.8059, 0.8571, 0.8571, 0.8571, 0.8095, 0.7143))  
rownames(RFstability) <- c(3,5,7,10,15,20,25)  
RFstability

## Accuracy Sensitivity Specificity  
## 3 0.8533 0.9074 0.7143  
## 5 0.8800 0.9074 0.8059  
## 7 0.8933 0.9074 0.8571  
## 10 0.8933 0.9074 0.8571  
## 15 0.9067 0.9259 0.8571  
## 20 0.8933 0.9259 0.8095  
## 25 0.8533 0.9074 0.7143

cols <- c("darkseagreen","cornflowerblue","gray48")  
pch <- c(17,18,15)  
xmax <- nrow(RFstability) + 2.5  
par(mar=c(4,4,1,1))  
plot(1:nrow(RFstability), 1:nrow(RFstability), pch="",   
 xlab="nFolds", ylab=NA, xaxt="n", yaxt="n",   
 ylim=c(.7,.95), bty="n", xlim=c(1,xmax), main="Logistic Regression Stability Results")  
for (i in seq(.5,.95,by=.025)) {  
 lines(1:nrow(RFstability), rep(i,nrow(RFstability)), col="gray69")  
}  
for (i in 1:ncol(RFstability)) {  
  
 points(1:nrow(RFstability), RFstability[,i], pch=pch[i],   
 col=cols[i], cex=1.2)  
  
 lines(1:nrow(RFstability), RFstability[,i], col=cols[i],   
 lwd=2)  
}  
axis(side=1, at=1:nrow(RFstability), tick=FALSE,   
 labels=rownames(RFstability))  
  
axis(side=1, at=seq(-0.5,8.5,by=1),   
 tick=FALSE, labels=NA)  
  
axis(side=2, at=seq(.7,.95,by=.025), tick=TRUE,   
 las=TRUE, labels=paste(seq(.7,.95,by=.025)))  
  
legend('bottomright',legend=colnames(RFstability), pch=pch,   
 col=cols, cex=1.0, bty="n", lwd=3, lty=1)

