Heart\_Disease

Yewande Ogunbowale

2022-08-19

Importe cleveland data into R. and change column names

Cleveland <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data", na.strings="?", sep = "," )  
colnames(Cleveland) <- c("age","sex","cp","trestbps","chol","fbs","restecg","thalach","exang","oldpeak","slope","ca","thal","pred")  
  
  
attach(Cleveland)

Data Exploration. This is the initial process of performing investigations on data so as to discover patterns, spot anomalies and check assumptions with the help of summary statistics and graphical representations

library(DataExplorer)  
  
#This gives a brief account(mean, median...) of the data  
summary(Cleveland)

## age sex cp trestbps   
## Min. :29.00 Min. :0.0000 Min. :1.000 Min. : 94.0   
## 1st Qu.:48.00 1st Qu.:0.0000 1st Qu.:3.000 1st Qu.:120.0   
## Median :56.00 Median :1.0000 Median :3.000 Median :130.0   
## Mean :54.44 Mean :0.6799 Mean :3.158 Mean :131.7   
## 3rd Qu.:61.00 3rd Qu.:1.0000 3rd Qu.:4.000 3rd Qu.:140.0   
## Max. :77.00 Max. :1.0000 Max. :4.000 Max. :200.0   
##   
## chol fbs restecg thalach   
## Min. :126.0 Min. :0.0000 Min. :0.0000 Min. : 71.0   
## 1st Qu.:211.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:133.5   
## Median :241.0 Median :0.0000 Median :1.0000 Median :153.0   
## Mean :246.7 Mean :0.1485 Mean :0.9901 Mean :149.6   
## 3rd Qu.:275.0 3rd Qu.:0.0000 3rd Qu.:2.0000 3rd Qu.:166.0   
## Max. :564.0 Max. :1.0000 Max. :2.0000 Max. :202.0   
##   
## exang oldpeak slope ca   
## Min. :0.0000 Min. :0.00 Min. :1.000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:0.00 1st Qu.:1.000 1st Qu.:0.0000   
## Median :0.0000 Median :0.80 Median :2.000 Median :0.0000   
## Mean :0.3267 Mean :1.04 Mean :1.601 Mean :0.6722   
## 3rd Qu.:1.0000 3rd Qu.:1.60 3rd Qu.:2.000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :6.20 Max. :3.000 Max. :3.0000   
## NA's :4   
## thal pred   
## Min. :3.000 Min. :0.0000   
## 1st Qu.:3.000 1st Qu.:0.0000   
## Median :3.000 Median :0.0000   
## Mean :4.734 Mean :0.9373   
## 3rd Qu.:7.000 3rd Qu.:2.0000   
## Max. :7.000 Max. :4.0000   
## NA's :2

#Head displays the first six data  
head(Cleveland)

## age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal pred  
## 1 63 1 1 145 233 1 2 150 0 2.3 3 0 6 0  
## 2 67 1 4 160 286 0 2 108 1 1.5 2 3 3 2  
## 3 67 1 4 120 229 0 2 129 1 2.6 2 2 7 1  
## 4 37 1 3 130 250 0 0 187 0 3.5 3 0 3 0  
## 5 41 0 2 130 204 0 2 172 0 1.4 1 0 3 0  
## 6 56 1 2 120 236 0 0 178 0 0.8 1 0 3 0

#Tail shows the last six data  
tail(Cleveland)

## age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal  
## 298 57 0 4 140 241 0 0 123 1 0.2 2 0 7  
## 299 45 1 1 110 264 0 0 132 0 1.2 2 0 7  
## 300 68 1 4 144 193 1 0 141 0 3.4 2 2 7  
## 301 57 1 4 130 131 0 0 115 1 1.2 2 1 7  
## 302 57 0 2 130 236 0 2 174 0 0.0 2 1 3  
## 303 38 1 3 138 175 0 0 173 0 0.0 1 NA 3  
## pred  
## 298 1  
## 299 1  
## 300 2  
## 301 3  
## 302 1  
## 303 0

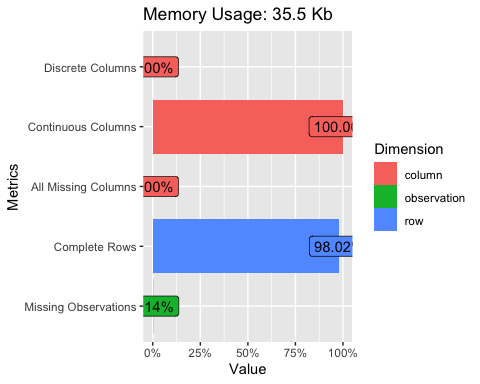
str(Cleveland)

## 'data.frame': 303 obs. of 14 variables:  
## $ age : num 63 67 67 37 41 56 62 57 63 53 ...  
## $ sex : num 1 1 1 1 0 1 0 0 1 1 ...  
## $ cp : num 1 4 4 3 2 2 4 4 4 4 ...  
## $ trestbps: num 145 160 120 130 130 120 140 120 130 140 ...  
## $ chol : num 233 286 229 250 204 236 268 354 254 203 ...  
## $ fbs : num 1 0 0 0 0 0 0 0 0 1 ...  
## $ restecg : num 2 2 2 0 2 0 2 0 2 2 ...  
## $ thalach : num 150 108 129 187 172 178 160 163 147 155 ...  
## $ exang : num 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 : num 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 ...  
## $ pred : int 0 2 1 0 0 0 3 0 2 1 ...

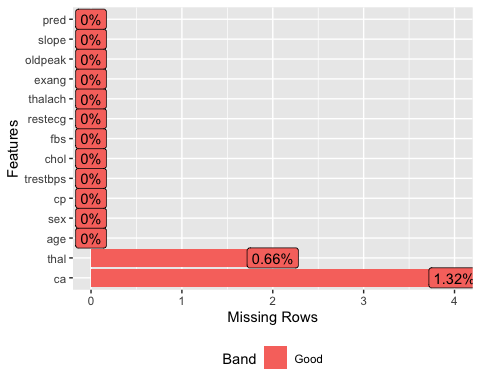
introduce(Cleveland)

## rows columns discrete\_columns continuous\_columns all\_missing\_columns  
## 1 303 14 0 14 0  
## total\_missing\_values complete\_rows total\_observations memory\_usage  
## 1 6 297 4242 36336

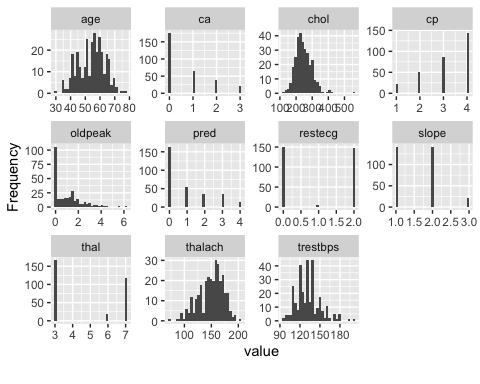
plot\_intro(Cleveland)



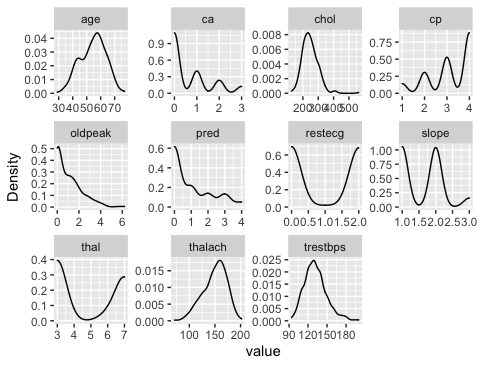
#Missing data poses problem in every data. This function displays missing data in cleveland.  
plot\_missing(Cleveland)



plot\_histogram(Cleveland)



plot\_density(Cleveland)



Data explorer() displays the content of Cleveland data. There are 6 missing datas. 2 in thal and 4 ca. Missing data or Na.s if not taken care of might affect our conclusion negatively.

Data Cleaning

anyNA(Cleveland)

## [1] TRUE

#True means there are missing data in cleveland. Some cleaning up is there for needed.  
  
  
sum(is.na(Cleveland))

## [1] 6

#summed up all the missing data in my dataset.  
  
  
colSums(is.na(Cleveland))

## age sex cp trestbps chol fbs restecg thalach   
## 0 0 0 0 0 0 0 0   
## exang oldpeak slope ca thal pred   
## 0 0 0 4 2 0

#shows the location of the missing value.  
  
  
Cleveland.new <- na.omit(Cleveland)  
nrow(Cleveland.new)

## [1] 297

#After cleaning, cleveland data reduced to 297 from 303 observations. Cleveland. new now has 297 observations.  
  
  
  
Cleveland$thal[which(is.na(Cleveland$thal))]<- mean(Cleveland$thal,na.rm = TRUE)  
Cleveland$ca[which(is.na(Cleveland$ca))]<- mean(Cleveland$ca,na.rm = TRUE)  
anyNA(Cleveland)

## [1] FALSE

There are 4 missing data in thal and 2 in ca. Cleveland data reduced to 297 after cleaning up.

After replacing missing values in thal and ca with mean. I checked for Na’s and it returned False.

Normalization minimizes data redundancy(Repetiton of similar data in multiple places)

normalize <- function(x) (x - min(x))/max(x)-min(x)  
cleveland\_norm <- as.data.frame(lapply(Cleveland.new[,c(1,4,5,8,10)],normalize))   
cleveland\_norm\_scaled <- scale(cleveland\_norm, center = TRUE, scale = TRUE)  
str(cleveland\_norm)

## 'data.frame': 297 obs. of 5 variables:  
## $ age : num -28.6 -28.5 -28.5 -28.9 -28.8 ...  
## $ trestbps: num -93.7 -93.7 -93.9 -93.8 -93.8 ...  
## $ chol : num -126 -126 -126 -126 -126 ...  
## $ thalach : num -70.6 -70.8 -70.7 -70.4 -70.5 ...  
## $ oldpeak : num 0.371 0.242 0.419 0.565 0.226 ...

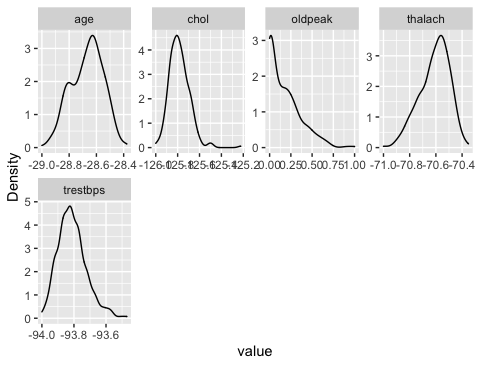
summary(cleveland\_norm)

## age trestbps chol thalach   
## Min. :-29.00 Min. :-94.00 Min. :-126.0 Min. :-71.00   
## 1st Qu.:-28.75 1st Qu.:-93.87 1st Qu.:-125.8 1st Qu.:-70.69   
## Median :-28.65 Median :-93.82 Median :-125.8 Median :-70.59   
## Mean :-28.67 Mean :-93.81 Mean :-125.8 Mean :-70.61   
## 3rd Qu.:-28.58 3rd Qu.:-93.77 3rd Qu.:-125.7 3rd Qu.:-70.53   
## Max. :-28.38 Max. :-93.47 Max. :-125.2 Max. :-70.35   
## oldpeak   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.1290   
## Mean :0.1703   
## 3rd Qu.:0.2581   
## Max. :1.0000

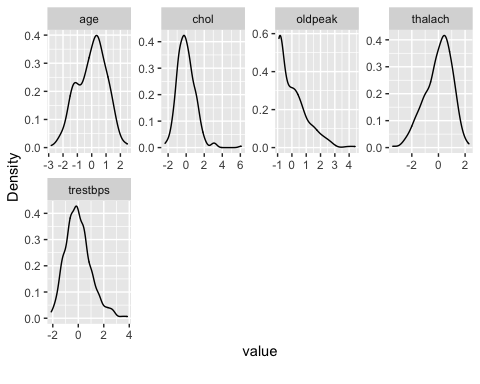
summary(cleveland\_norm\_scaled)

## age trestbps chol thalach   
## Min. :-2.8224 Min. :-2.12205 Min. :-2.33377 Min. :-3.4261   
## 1st Qu.:-0.7229 1st Qu.:-0.65832 1st Qu.:-0.69907 1st Qu.:-0.7235   
## Median : 0.1611 Median :-0.09534 Median :-0.08366 Median : 0.1482   
## Mean : 0.0000 Mean : 0.00000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.: 0.7136 3rd Qu.: 0.46763 3rd Qu.: 0.55098 3rd Qu.: 0.7149   
## Max. : 2.4816 Max. : 3.84547 Max. : 6.08970 Max. : 2.2841   
## oldpeak   
## Min. :-0.9052   
## 1st Qu.:-0.9052   
## Median :-0.2192   
## Mean : 0.0000   
## 3rd Qu.: 0.4669   
## Max. : 4.4116

plot\_density(cleveland\_norm)



plot\_density(cleveland\_norm\_scaled)



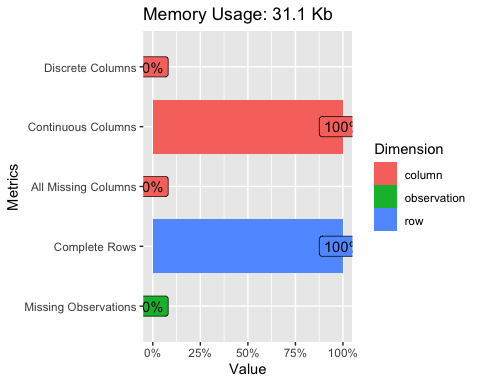
Convertion of factor variable to dummy Variable. Transforms a categorical variable with more than two modalities in several dummy variables with only two modalities, Yes(=1) or No(0)

library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

cleveland\_dum <- as.data.frame(lapply(Cleveland.new[,c(3,7,11:13)],as.factor))  
cleveland\_dumm <- dummyVars(~.,data=cleveland\_dum,fullRank = TRUE)  
cleveland\_dummy <- as.data.frame(predict(cleveland\_dumm,newdata=cleveland\_dum))  
plot\_intro(cleveland\_dummy)



str(cleveland\_dummy)

## 'data.frame': 297 obs. of 12 variables:  
## $ cp.2 : num 0 0 0 0 1 1 0 0 0 0 ...  
## $ cp.3 : num 0 0 0 1 0 0 0 0 0 0 ...  
## $ cp.4 : num 0 1 1 0 0 0 1 1 1 1 ...  
## $ restecg.1: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ restecg.2: num 1 1 1 0 1 0 1 0 1 1 ...  
## $ slope.2 : num 0 1 1 0 0 0 0 0 1 0 ...  
## $ slope.3 : num 1 0 0 1 0 0 1 0 0 1 ...  
## $ ca.1 : num 0 0 0 0 0 0 0 0 1 0 ...  
## $ ca.2 : num 0 0 1 0 0 0 1 0 0 0 ...  
## $ ca.3 : num 0 1 0 0 0 0 0 0 0 0 ...  
## $ thal.6 : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ thal.7 : num 0 0 1 0 0 0 0 0 1 1 ...

Cleveland.new$pred <- with(Cleveland.new, ifelse(pred >=1, 1, 0))  
Cleveland.new$pred <- as.factor(Cleveland.new$pred) #change predicted variable to factor.

cleveland\_we <- cbind(Cleveland.new$sex,Cleveland.new$fbs,cleveland\_norm,cleveland\_dummy,Cleveland.new$pred) # combine features with cbind()  
str(cleveland\_we)

## 'data.frame': 297 obs. of 20 variables:  
## $ Cleveland.new$sex : num 1 1 1 1 0 1 0 0 1 1 ...  
## $ Cleveland.new$fbs : num 1 0 0 0 0 0 0 0 0 1 ...  
## $ age : num -28.6 -28.5 -28.5 -28.9 -28.8 ...  
## $ trestbps : num -93.7 -93.7 -93.9 -93.8 -93.8 ...  
## $ chol : num -126 -126 -126 -126 -126 ...  
## $ thalach : num -70.6 -70.8 -70.7 -70.4 -70.5 ...  
## $ oldpeak : num 0.371 0.242 0.419 0.565 0.226 ...  
## $ cp.2 : num 0 0 0 0 1 1 0 0 0 0 ...  
## $ cp.3 : num 0 0 0 1 0 0 0 0 0 0 ...  
## $ cp.4 : num 0 1 1 0 0 0 1 1 1 1 ...  
## $ restecg.1 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ restecg.2 : num 1 1 1 0 1 0 1 0 1 1 ...  
## $ slope.2 : num 0 1 1 0 0 0 0 0 1 0 ...  
## $ slope.3 : num 1 0 0 1 0 0 1 0 0 1 ...  
## $ ca.1 : num 0 0 0 0 0 0 0 0 1 0 ...  
## $ ca.2 : num 0 0 1 0 0 0 1 0 0 0 ...  
## $ ca.3 : num 0 1 0 0 0 0 0 0 0 0 ...  
## $ thal.6 : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ thal.7 : num 0 0 1 0 0 0 0 0 1 1 ...  
## $ Cleveland.new$pred: Factor w/ 2 levels "0","1": 1 2 2 1 1 1 2 1 2 2 ...

DATA PARTITION this is normally done for manageability, performance and load balancing

set.seed(789) # To get the same random sample  
cle <- sample(2, nrow(Cleveland), replace = T, prob = c(0.7, 0.3)) #create samples with ratio 70:30  
  
  
cleveland\_train <- Cleveland[cle==1, 1:5] # 70% training data  
cleveland\_test <- Cleveland[cle==2, 1:5] # remainng 30% test data  
  
  
#creating seperate data frame   
cleveland\_train\_labels <- Cleveland[cle==1, 6]  
cleveland\_test\_labels <- Cleveland[cle==2, 6]

set.seed(789) # To get the same random sample  
cle <- sample(2, nrow(Cleveland), replace = T, prob = c(0.7, 0.3)) #create samples with ratio 70:30  
  
  
cleveland\_train <- Cleveland[cle==1, 1:5] # 70% training data  
cleveland\_test <- Cleveland[cle==2, 1:5] # remainng 30% test data  
  
  
#creating seperate data frame   
cleveland\_train\_labels <- Cleveland[cle==1, 6]  
cleveland\_test\_labels <- Cleveland[cle==2, 6]

library(class) # To call packages as it carries KNN function  
  
NROW(cleveland\_train\_labels) #to find the number of observation

## [1] 209

sqrt(209) #square root of 209 to identify the optimum value.

## [1] 14.45683

#Initiate the value of k for k = 1, 10, 11, 14, 15 and 12  
  
cleveland\_pred <- knn(train = cleveland\_train, test = cleveland\_test, cl = cleveland\_train\_labels, k = 1)  
cleveland\_pred10 <- knn(train = cleveland\_train, test = cleveland\_test, cl = cleveland\_train\_labels, k = 10)  
cleveland\_pred11 <- knn(train = cleveland\_train, test = cleveland\_test, cl = cleveland\_train\_labels, k = 11)  
cleveland\_pred15 <- knn(train = cleveland\_train, test = cleveland\_test, cl = cleveland\_train\_labels, k = 15)  
cleveland\_pred14 <- knn(train = cleveland\_train, test = cleveland\_test, cl = cleveland\_train\_labels, k = 14)  
cleveland\_pred12 <- knn(train = cleveland\_train, test = cleveland\_test, cl = cleveland\_train\_labels, k = 12)  
  
  
#Accuracy value  
#Proportion of correct classification for k = 1,10,11, 14, 12 and 15  
Cleveland.1 <- 100 \* sum(cleveland\_test\_labels == cleveland\_pred)/NROW(cleveland\_test\_labels)  
Cleveland.1

## [1] 71.2766

summary(Cleveland.1)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 71.28 71.28 71.28 71.28 71.28 71.28

cleveland.10 <- 100 \* sum(cleveland\_test\_labels == cleveland\_pred10)/NROW(cleveland\_test\_labels)  
cleveland.10

## [1] 82.97872

summary(cleveland.10)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 82.98 82.98 82.98 82.98 82.98 82.98

cleveland.11 <- 100 \* sum(cleveland\_test\_labels == cleveland\_pred11)/NROW(cleveland\_test\_labels)  
cleveland.11

## [1] 82.97872

summary(cleveland.11)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 82.98 82.98 82.98 82.98 82.98 82.98

cleveland.14 <- 100 \* sum(cleveland\_test\_labels == cleveland\_pred14)/NROW(cleveland\_test\_labels)  
cleveland.14

## [1] 82.97872

summary(cleveland.14)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 82.98 82.98 82.98 82.98 82.98 82.98

cleveland.15 <- 100 \* sum(cleveland\_test\_labels == cleveland\_pred15)/NROW(cleveland\_test\_labels)  
cleveland.15

## [1] 82.97872

summary(cleveland.15)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 82.98 82.98 82.98 82.98 82.98 82.98

cleveland.12 <- 100 \* sum(cleveland\_test\_labels == cleveland\_pred12)/NROW(cleveland\_test\_labels)  
cleveland.12

## [1] 82.97872

summary(12)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 12 12 12 12 12 12

table(cleveland\_pred15 ,cleveland\_test\_labels)

## cleveland\_test\_labels  
## cleveland\_pred15 0 1  
## 0 78 16  
## 1 0 0

#out of 94 observations 78 people does not have heart disease, out of no observation just one person has heart disease.  
cleveland.15

## [1] 82.97872

# It "0" shows no heartdisease. Accuracy value = 82.97872

Confusion matrix

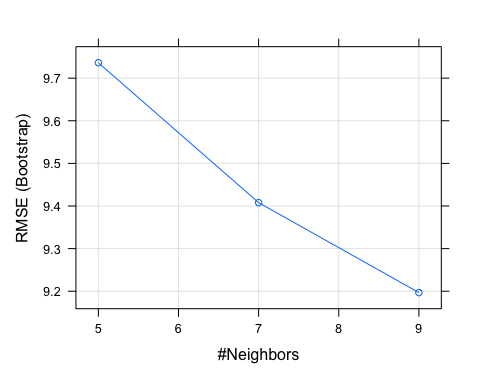
library(caret) #It load the confusion matrix function  
  
  
confusionMatrix(table(cleveland\_pred15 ,cleveland\_test\_labels))

## Confusion Matrix and Statistics  
##   
## cleveland\_test\_labels  
## cleveland\_pred15 0 1  
## 0 78 16  
## 1 0 0  
##   
## Accuracy : 0.8298   
## 95% CI : (0.7384, 0.8995)  
## No Information Rate : 0.8298   
## P-Value [Acc > NIR] : 0.5662505   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : 0.0001768   
##   
## Sensitivity : 1.0000   
## Specificity : 0.0000   
## Pos Pred Value : 0.8298   
## Neg Pred Value : NaN   
## Prevalence : 0.8298   
## Detection Rate : 0.8298   
## Detection Prevalence : 1.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : 0   
##

#Because my training data and testing data is not huge, i got an accuracy value of 82%, but the more data I feed the machine, the more accurate my algorithm.   
 #95% confident interval is between 73% and 89%, no information rate. If there was heartdisease, the model would have been correct 82%  
  
  
  
trcontrol <- trainControl(method = "LOOCV",  
 verboseIter = T)  
Cleveland\_fit <- train(age ~.,  
 data = cleveland\_train,  
 method ="knn",  
 trcontrol = trcontrol,  
 preProcess = c("center","scale"),  
 tuneLenght = 15)  
Cleveland\_fit

## k-Nearest Neighbors   
##   
## 209 samples  
## 4 predictor  
##   
## Pre-processing: centered (4), scaled (4)   
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 209, 209, 209, 209, 209, 209, ...   
## Resampling results across tuning parameters:  
##   
## k RMSE Rsquared MAE   
## 5 9.736095 0.01920921 7.878851  
## 7 9.407884 0.02166241 7.656981  
## 9 9.196737 0.02243345 7.507003  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was k = 9.

plot(Cleveland\_fit)



varImp(Cleveland\_fit)

## loess r-squared variable importance  
##   
## Overall  
## trestbps 100.00  
## chol 51.69  
## sex 14.96  
## cp 0.00

cleveland\_predict <- predict(Cleveland\_fit, newdata = cleveland\_test)  
cleveland\_predict

## [1] 56.22222 54.77778 53.77778 50.33333 59.66667 52.11111 54.11111 50.66667  
## [9] 50.77778 54.77778 50.66667 53.00000 49.55556 55.11111 53.11111 56.77778  
## [17] 56.33333 54.77778 56.11111 56.11111 61.11111 54.88889 51.22222 55.11111  
## [25] 54.88889 54.66667 51.11111 51.88889 51.11111 57.77778 48.77778 52.00000  
## [33] 57.00000 59.22222 48.55556 56.44444 57.77778 60.44444 54.66667 59.22222  
## [41] 52.66667 50.33333 52.44444 49.22222 50.77778 54.55556 58.00000 56.11111  
## [49] 50.44444 53.66667 54.55556 56.44444 58.33333 50.55556 63.22222 61.77778  
## [57] 50.66667 50.66667 51.77778 53.22222 55.66667 58.88889 54.88889 51.88889  
## [65] 52.11111 61.22222 49.00000 51.11111 50.33333 60.55556 53.44444 57.00000  
## [73] 54.00000 54.33333 53.66667 54.88889 48.11111 55.00000 54.11111 53.88889  
## [81] 53.55556 55.33333 50.33333 53.88889 59.55556 51.88889 49.22222 56.88889  
## [89] 60.44444 54.11111 52.88889 59.22222 61.00000 57.77778

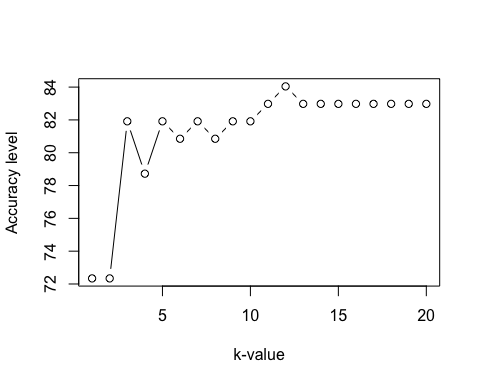
# Here, K optm = 9.

K OPTIMAL

i=1 #declaration to initiate for loop  
k.optm=1 #declaration to initiate for loop  
for (i in 1:20){  
 cleveland.model <- knn(train=cleveland\_train, test=cleveland\_test, cl= cleveland\_train\_labels, k=i)  
 k.optm[i] <- 100 \* sum(cleveland\_test\_labels == cleveland.model)/NROW(cleveland\_test\_labels)  
 k=i  
 cat(k, '=', k.optm[i],'\n') #to print % accuracy  
}

## 1 = 72.34043   
## 2 = 72.34043   
## 3 = 81.91489   
## 4 = 78.7234   
## 5 = 81.91489   
## 6 = 80.85106   
## 7 = 81.91489   
## 8 = 80.85106   
## 9 = 81.91489   
## 10 = 81.91489   
## 11 = 82.97872   
## 12 = 84.04255   
## 13 = 82.97872   
## 14 = 82.97872   
## 15 = 82.97872   
## 16 = 82.97872   
## 17 = 82.97872   
## 18 = 82.97872   
## 19 = 82.97872   
## 20 = 82.97872

plot(k.optm, type="b", xlab="k-value",ylab="Accuracy level") #to plot % accuracy wrt to k-value



#This gives the best value of K  
# From the graph, I can choose any value between 10 and 20 for k optm.

CONCLUSION

Heart disease is one of the most common health conditions in developed countries, involving the narrowing or blocking of heart vessels that leads to increased risk of heart atttack. The condition can be diagonised conclusively with imagging scan, but the costs of these scans prohibit most people from undergoing them regularly.

A more systemic approach to diagonising underlying ailments might be to use algorithms trained on entire medical data to generate more accurate predictions.