

# SVM classification of heart disease dataset

Soundarya G

25/04/2021

- INTRODUCTION
  - 1. Load the necessary packages
  - 2. Load the dataset
- EXPLORATORY DATA ANALYSIS
  - 1. Columns and shape of dataset
  - 2. Viewing dataset
  - 3. Conversion factor variable
  - 4. Box plot for statistical distribution
  - 5. Correlation analysis between some variables
  - 6. Confusion matrix
- Model Building
  - 1. Data splitting
  - 2. Feature selections
  - 3. Building a SVM classifier
  - 4. ROC and AUC value

## INTRODUCTION

### Aim of analysis

In the following document, I will be using SVM classification technique to predict heart disease (angiographic disease status). From a set of 14 variables, the most important to predict heart failure are whether or not there is a reversible defect in Thallium stress test followed by whether or not there is an occurrence of asymptomatic chest pain.

## 1. Load the necessary packages

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(pROC) #to plot the ROC curves
```

```
## Loading required package: pROC
```

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

```
##  
## Attaching package: 'pROC'
```

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

```
require(caret)
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
# Attach Packages  
library(tidyverse)    # data manipulation and visualization
```

```
## — Attaching packages —————  
————— tidyverse 1.3.0 —
```

```
## ✓ tibble  3.0.3      ✓ dplyr    1.0.4  
## ✓ tidyr   1.1.1      ✓ stringr 1.4.0  
## ✓ readr   1.4.0      ✓ forcats 0.5.1  
## ✓ purrr   0.3.4
```

```
## — Conflicts —————  
————— tidyverse_conflicts() —  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()     masks stats::lag()  
## x purrr::lift()    masks caret::lift()
```

```
library(kernlab)      # SVM methodology
```

```
##  
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:purrr':  
##  
##      cross
```

```
## The following object is masked from 'package:ggplot2':  
##  
##      alpha
```

```
library(e1071)         # SVM methodology  
library(ISLR)          # contains example data set "Khan"  
library(RColorBrewer)  # customized coloring of plots
```

## 2. Load the dataset

```
The heart disease data are available at UCI and Kaggle.
```

```
heartdf <- read.csv("heart.csv")
```

# EXPLORATORY DATA ANALYSIS

## 1. Columns and shape of dataset

```
names(heartdf) <- c( "age", "sex", "cp", "trestbps", "chol","fbs", "restecg", "thalac  
h","exang", "oldpeak","slope", "ca", "thal", "num")  
attach(heartdf)
```

```
# dimensions of the dataset  
dim(heartdf)
```

```
## [1] 303  14
```

The variable we want to predict is num with Value 0: < 50% diameter narrowing and Value 1: > 50% diameter narrowing. We assume that every value with 0 means heart is okay, and 1,2,3,4 means heart disease.

From the possible values the variables can take, it is evident that the following need to be dummified because the distances in the values is random: cp,thal, restecg, slope

## 2. Viewing dataset

```
head(heartdf,5)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal num  
## 1  63  1  3    145   233   1        0    150     0     2.3    0  0    1    1  
## 2  37  1  2    130   250   0        1    187     0     3.5    0  0    2    1  
## 3  41  0  1    130   204   0        0    172     0     1.4    2  0    2    1  
## 4  56  1  1    120   236   0        1    178     0     0.8    2  0    2    1  
## 5  57  0  0    120   354   0        1    163     1     0.6    2  0    2    1
```

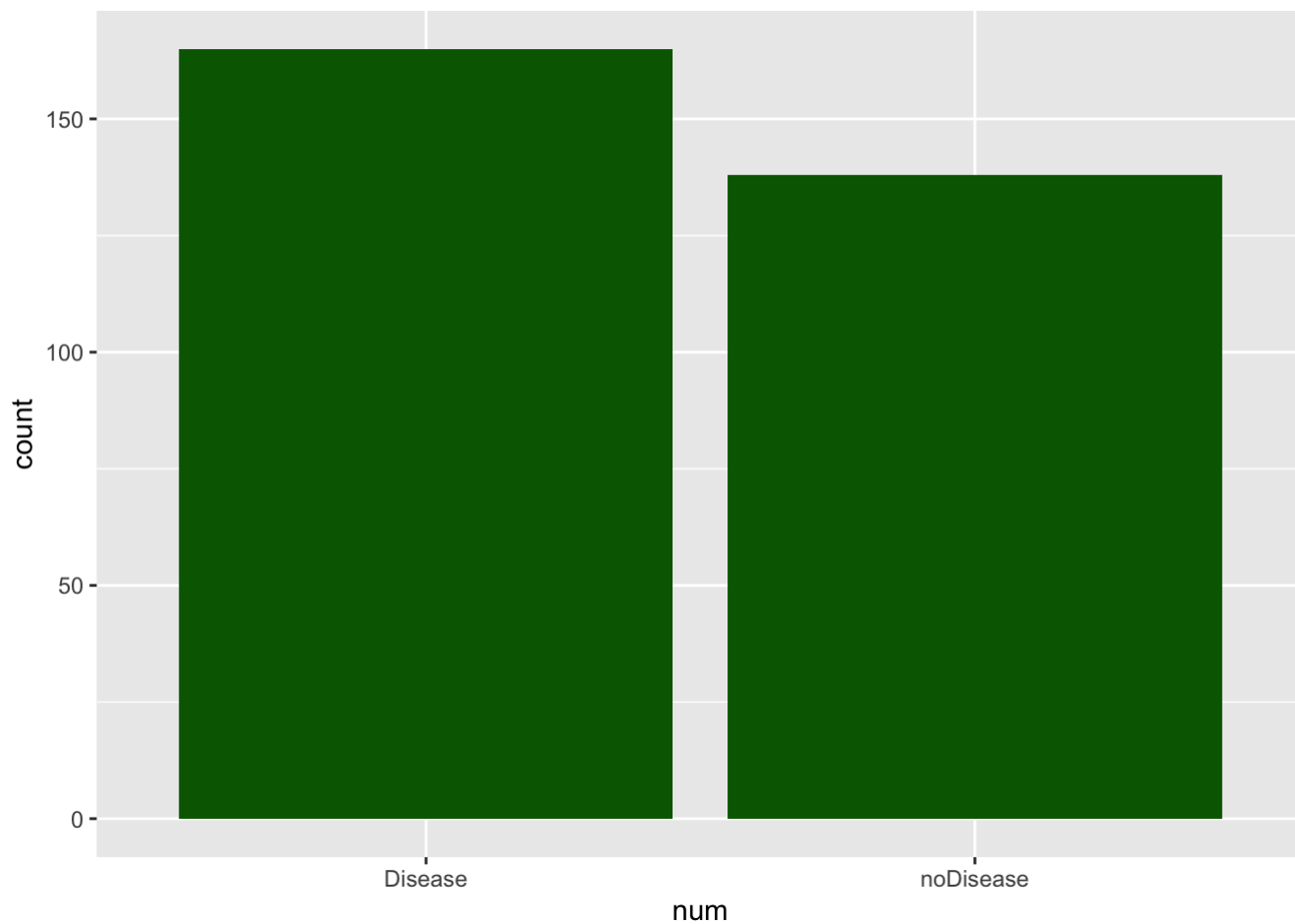
Explore the data and find how many had heart attacks, women or men have of a particular age?

```
#converting the num variable to binary class variable
```

```
heartdf$num<-ifelse(heartdf$num > 0,"Disease","noDisease")  
table(heartdf$num)
```

```
##  
## Disease noDisease  
##      165      138
```

```
#distribution of the target variable  
ggplot(heartdf,aes(x = num)) + geom_bar(fill="dark green")
```



### 3. Conversion factor variable

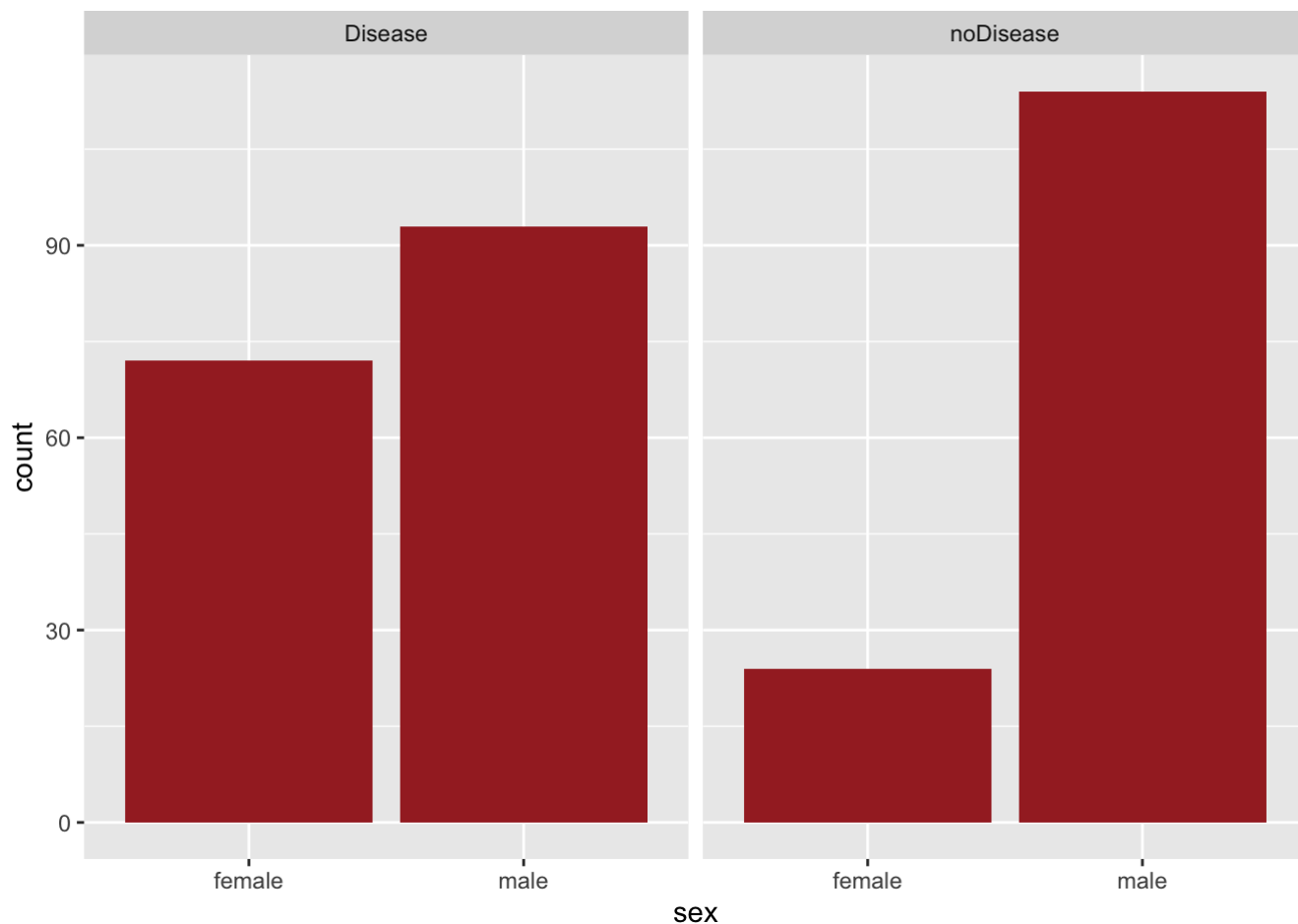
```
#converting to factor variable  
  
heartdf$sex<-ifelse(heartdf$sex==0,"female","male")  
  
table(heartdf$sex)
```

```
##  
## female   male  
##      96    207
```

```
table(sex=heartdf$sex,disease=heartdf$num)
```

```
##          disease
## sex      Disease noDisease
##  female      72      24
##   male      93     114
```

```
ggplot(heartdf,aes(x=sex)) + geom_bar(fill="brown") + facet_wrap(~num)
```



## 4. Box plot for statistical distribution

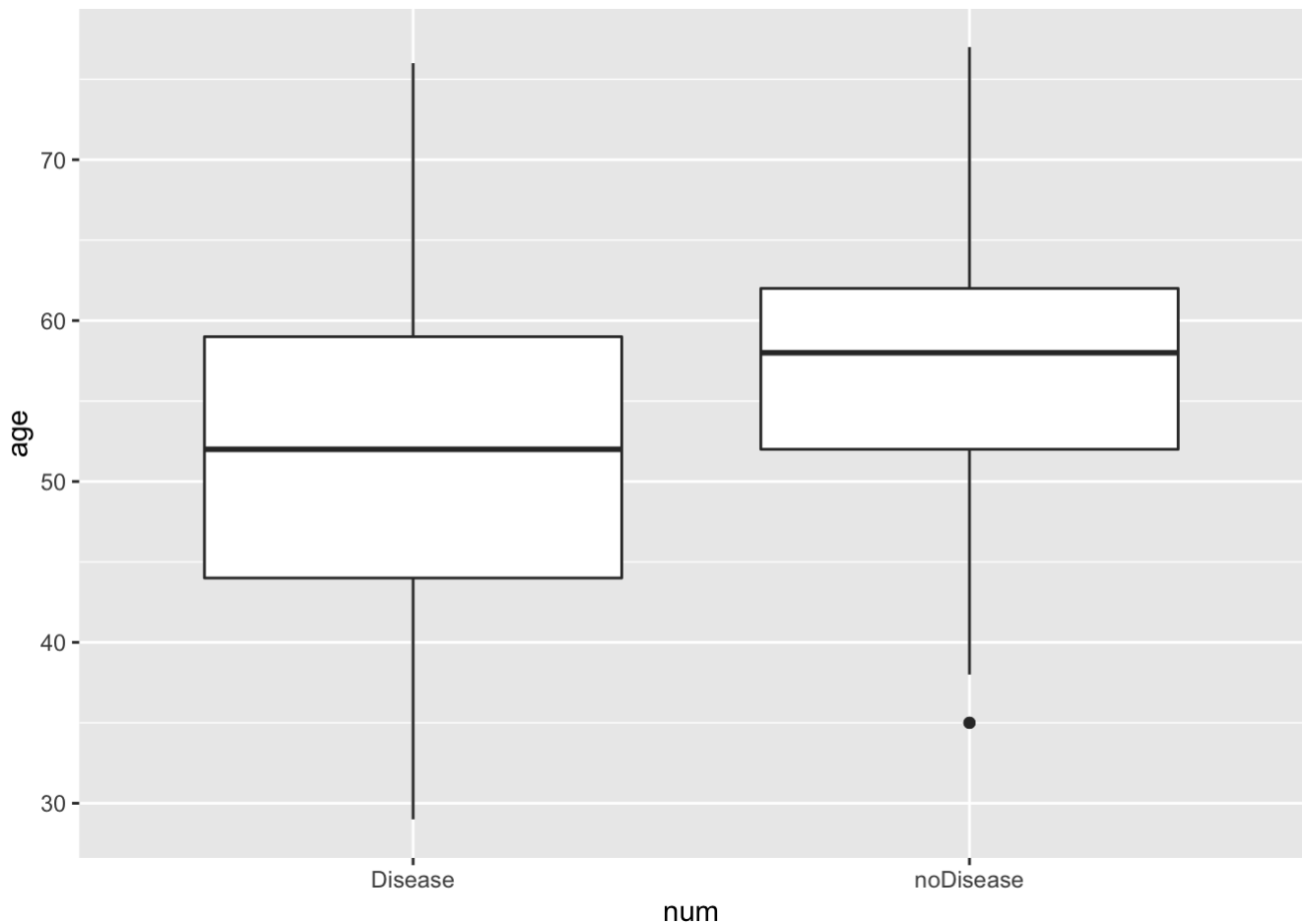
```
#heart disease and age
```

```
by(heartdf$age,heartdf$num,summary)
```

```
## heartdf$num: Disease
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   29.0   44.0   52.0   52.5   59.0   76.0
## -----
## heartdf$num: noDisease
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   35.0   52.0   58.0   56.6   62.0   77.0
```

-So people who had heart disease for them the mean age is 52.5

```
ggplot(heartdf,aes(x = num,y = age)) + geom_boxplot()
```



## 5. Correlation analysis between some variables

```
#very low correlation  
cor.test(age,chol)
```

```
##  
##  Pearson's product-moment correlation  
##  
## data:  age and chol  
## t = 3.7948, df = 301, p-value = 0.0001786  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
##  0.1034917 0.3186831  
## sample estimates:  
##      cor  
## 0.213678
```

## 6. Confusion matrix

```
#confusion matrix of chest pain and heart disease  
  
table(cp,num)
```

```
##      num
## cp    0    1
##    0 104  39
##    1   9  41
##    2  18  69
##    3   7  16
```

```
#confusuon matrix of exersice induced asthama and heart disease
```

```
table(exang,num)
```

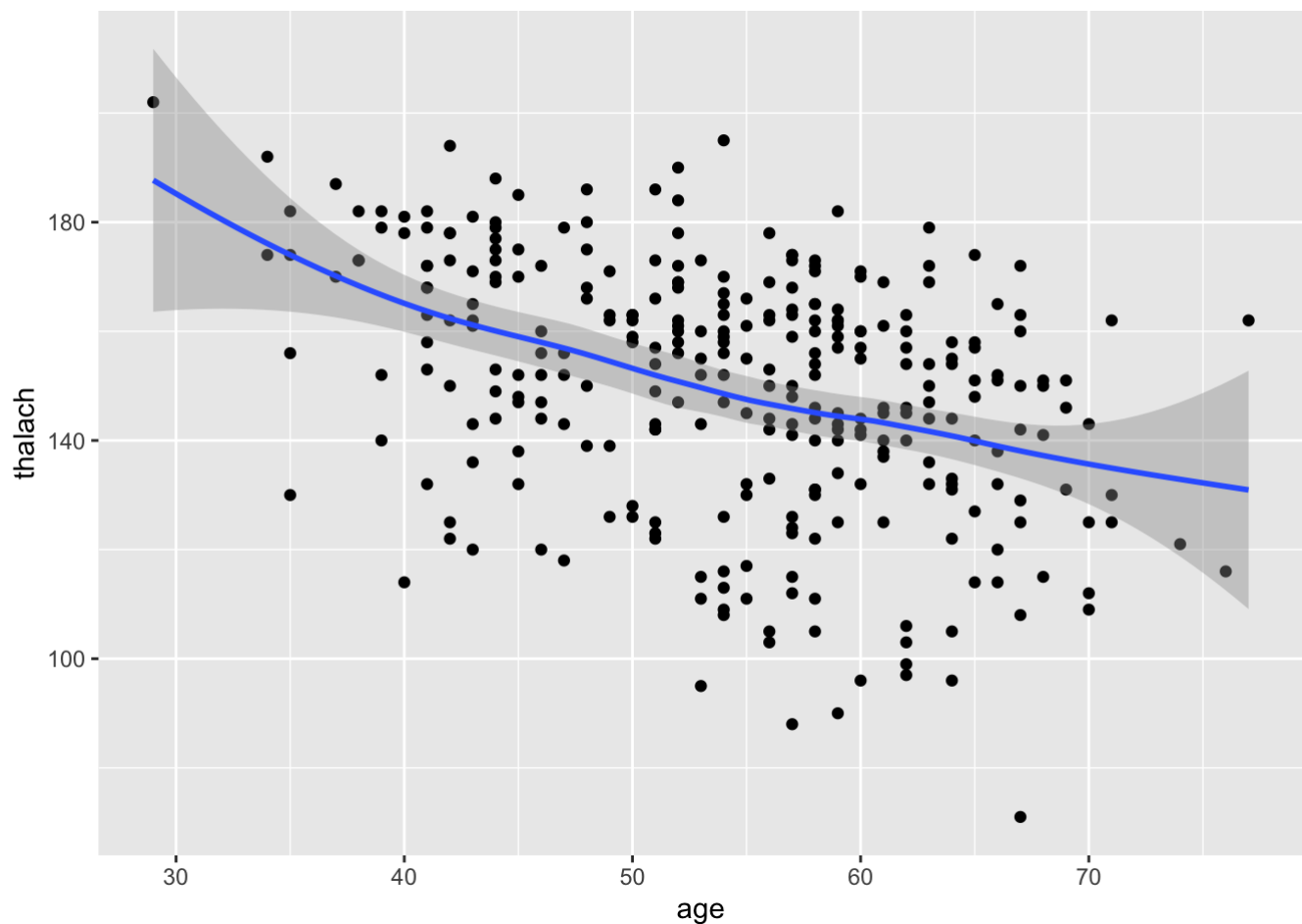
```
##      num
## exang  0    1
##    0  62 142
##    1  76  23
```

```
cor.test(age,thalach)
```

```
##
## Pearson's product-moment correlation
##
## data:  age and thalach
## t = -7.5386, df = 301, p-value = 5.628e-13
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## -0.4892312 -0.2992831
## sample estimates:
##          cor
## -0.3985219
```

```
ggplot(heartdf,aes(x = age,y = thalach )) + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



-Here we can notice as age increase maximum heart rate achieved decreases, as the correlation is negative.

# Model Building

## 1. Data splitting

```
set.seed(5)

inTrainRows <- createDataPartition(heartdf$num, p=0.8, list=FALSE)

trainData <- heartdf[inTrainRows,]
testData <- heartdf[-inTrainRows,]

nrow(trainData)/(nrow(testData)+nrow(trainData))
```

```
## [1] 0.8019802
```

## 2. Feature selections



```
# for this to work add names to all levels (numbers not allowed)
feature.names=names(heartdf)
for (f in feature.names) {
  if (class(heartdf[[f]])=="factor") {
    levels <- unique(c(heartdf[[f]]))
    heartdf[[f]] <- factor(heartdf[[f]],
      labels=make.names(levels))
  }
}
```

```
#converting to factor variable with 2 levels

heartdf$num<-as.factor(heartdf$num)
levels(heartdf$num) <- c("Notdisease","Disease")

table(heartdf$num)
```

```
##
## Notdisease      Disease
##           165          138
```

### 3. Building a SVM classifier

Now SVM classifier tends to generate hyperplanes which separate the classes with maximum margins i.e in simpler terms it aims to generate maximum marginal hyperplane.

```
set.seed(10)
inTrainRows <- createDataPartition(heartdf$num,p=0.7,list=FALSE)
trainData2 <- heartdf[inTrainRows,]
testData2 <- heartdf[-inTrainRows,]

#cross validation
fitControl <- trainControl(method = "repeatedcv",
  number = 10,
  repeats = 10,
  ## Estimate class probabilities
  classProbs = TRUE,
  ## Evaluate performance using
  ## the following function
  summaryFunction = twoClassSummary)
```

```
svmModel <- train(num ~ ., data = na.omit(trainData2),
  method = "svmRadial",
  trControl = fitControl,
  preProcess = c("center", "scale"),
  tuneLength = 8,
  metric = "ROC")

svmModel
```

```
## Support Vector Machines with Radial Basis Function Kernel
##
## 213 samples
## 13 predictor
## 2 classes: 'Notdisease', 'Disease'
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 191, 192, 192, 192, 191, 192, ...
## Resampling results across tuning parameters:
##
##      C          ROC          Sens          Spec
##      0.25  0.8934646  0.8569697  0.7500000
##      0.50  0.8929621  0.8697727  0.7461111
##      1.00  0.8941759  0.8534848  0.7697778
##      2.00  0.8891439  0.8408333  0.7674444
##      4.00  0.8805833  0.8246970  0.7380000
##      8.00  0.8785909  0.8303788  0.7130000
##     16.00  0.8678897  0.8192424  0.7103333
##     32.00  0.8589891  0.8083333  0.7010000
##
## Tuning parameter 'sigma' was held constant at a value of 0.05165118
## ROC was used to select the optimal model using the largest value.
## The final values used for the model were sigma = 0.05165118 and C = 1.
```

```
#prediction on test data-class labels
svmPrediction <- predict(svmModel, testData2)

#probability of no heart disease-finding probabilities value
svmPredictionprob <- predict(svmModel, testData2, type='prob')[2]

#generating a confusion matrix
ConfMatrixPrediction <- confusionMatrix(svmPrediction, na.omit(testData2)$num)
ConfMatrixPrediction$table
```

```
##              Reference
## Prediction  Notdisease Disease
## Notdisease      43      7
## Disease         6     34
```

-In the confusion matrix the diagonals represent the correctly classified examples, whereas the offdiagonals are incorrectly classifier examples.

-To find the ROC curver and the AUC value to better understand the accuracy and performance

-ROC curve is the plot of True positive rate vs the false positive rate.

## 4. ROC and AUC value

```
#ROC and AUC value
```

```
AUC<- roc(na.omit(testData2)$num,as.numeric(as.matrix((svmPredictionprob))))$auc
```

```
## Setting levels: control = Notdisease, case = Disease
```

```
## Setting direction: controls < cases
```

```
Accuracy<- ConfMatrixPrediction$overall['Accuracy']
```

```
svmPerformance<-cbind(AUC,Accuracy)  
svmPerformance
```

```
##                AUC  Accuracy  
## Accuracy 0.9133897 0.8555556
```

Hence we get an AUC value of 0.9133897 and overall prediction accuracy of 0.8555556.

```
auc_roc<-roc(na.omit(testData2)$num,as.numeric(as.matrix((svmPredictionprob))))
```

```
## Setting levels: control = Notdisease, case = Disease
```

```
## Setting direction: controls < cases
```

```
plot(auc_roc)
```

