***Drazen Zack, Ankur Patel, Gulkhan Anassova, Max Gemma***

*RISK FACTORS IN HEART DISEASE*

**TABLE OF CONTENTS**

1. **Introduction ………………………………………………………………………………..3**
2. **Read, View, Structure, Summary ……………………………………………………....4**
3. **Clean Data………………………………………………………………………………....5**
4. **Exploratory Data Analysis ……………………………………………………………….8**
5. **Train & Test Data Sets ………………………………………………………………...…16**
6. **Logistic Regression ………………………………………………………………………17**
   1. Backward Stepwise Regression …………………………………………..26
   2. Table of Model Information ……………………………………………….38
      1. *AIC (Akaike Information Criteria)* ………………………………...40
      2. *Null Deviance & Residual Deviance* …………………………….40
      3. *AUC (Area Under Curve)…………………………………………...*41
   3. ROC (Receiver Operating Characteristics)……………………………..41
   4. Accuracy Tables …………………………………………………………….42
   5. Test Models ……………………………………………………………………43
   6. ROC for Test Data …………………………………………………………...44
   7. Test Data Accuracy …………………………………………………………44
   8. Best Logistic Regression Model ……………………………………………45
7. **Support Vector Machines** ……………………………………………………………..**46**
   1. SVM Train Accuracy ………………………………………………………...*69*
   2. SVM Train Tables ……………………………………………………………..71
   3. Test SVM ……………………………………………………………………….72
   4. Test SVM Accuracy & Table ……………………………………………….75
   5. Best SVM Model ……………………………………………………………...76
8. **Best Model between SVM & Logistic Regression ………………………………….76**
9. **Conclusion ………………………………………………………………………………..77**

**INTRODUCTION**

In this project, our group will study and attempt to predict the possibilities of a given patient to have heart disease. In the project, logistic regression and support vector machines are applied on the ‘heart.csv’ data set, which contains data on 303 patients consisting of 13 risk factors and a zero-one valued variable that indicates if the patient has heart disease.

First, the group ran logistic regression on 250 random patients of the data set to train a model. Our aim is for the best complete logistic regression equation by training the models with adjusted parameters. We used that model for the remaining portion of the data set to test its accuracy against the actual values in ‘heart.csv’.

The group also ran support vector machines on 250 random patients. As we did in logistic regression analysis, we performed accuracy tests to get the best model for the test data set. All appropriate results, graphs, plots, logs, model’s efficiency and performance, and comments will be included in this report.

It will be a great exercise to analyze this healthcare data set and make predictions. It plays a significant role in the healthcare industry.

**READ, VIEW, STRUCTURE, SUMMARY**

The first step started with the initial functions for the data set- read, view, structure, and summary. The ‘heart.csv’ data set has 303 observations of 15 variables (12 numeric, 3 factors). We can also use ‘dim()’ function to find the observations’ and variables’ numbers, but we use ‘str()’ for the data type and to see the first 10 values. We used ‘summary()’ to get the statistics and for a better look at the data.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Read, View, Structure, Summary   
setwd("~/Desktop")  
heart <- read.csv("heart.csv")  
str(heart)

## 'data.frame': 303 obs. of 15 variables:  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Age : int 63 67 67 37 41 56 62 57 63 53 ...  
## $ Sex : int 1 1 1 1 0 1 0 0 1 1 ...  
## $ ChestPain: Factor w/ 4 levels "asymptomatic",..: 4 1 1 2 3 3 1 1 1 1 ...  
## $ RestBP : int 145 160 120 130 130 120 140 120 130 140 ...  
## $ Chol : int 233 286 229 250 204 236 268 354 254 203 ...  
## $ Fbs : int 1 0 0 0 0 0 0 0 0 1 ...  
## $ RestECG : int 2 2 2 0 2 0 2 0 2 2 ...  
## $ MaxHR : int 150 108 129 187 172 178 160 163 147 155 ...  
## $ ExAng : int 0 1 1 0 0 0 0 1 0 1 ...  
## $ Oldpeak : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ Slope : int 3 2 2 3 1 1 3 1 2 3 ...  
## $ Ca : int 0 3 2 0 0 0 2 0 1 0 ...  
## $ Thal : Factor w/ 3 levels "fixed","normal",..: 1 2 3 2 2 2 2 2 3 3 ...  
## $ AHD : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 2 1 2 2 ...

summary(heart)

## X Age Sex ChestPain   
## Min. : 1.0 Min. :29.00 Min. :0.0000 asymptomatic:144   
## 1st Qu.: 76.5 1st Qu.:48.00 1st Qu.:0.0000 nonanginal : 86   
## Median :152.0 Median :56.00 Median :1.0000 nontypical : 50   
## Mean :152.0 Mean :54.44 Mean :0.6799 typical : 23   
## 3rd Qu.:227.5 3rd Qu.:61.00 3rd Qu.:1.0000   
## Max. :303.0 Max. :77.00 Max. :1.0000   
##   
## RestBP Chol Fbs RestECG   
## Min. : 94.0 Min. :126.0 Min. :0.0000 Min. :0.0000   
## 1st Qu.:120.0 1st Qu.:211.0 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :130.0 Median :241.0 Median :0.0000 Median :1.0000   
## Mean :131.7 Mean :246.7 Mean :0.1485 Mean :0.9901   
## 3rd Qu.:140.0 3rd Qu.:275.0 3rd Qu.:0.0000 3rd Qu.:2.0000   
## Max. :200.0 Max. :564.0 Max. :1.0000 Max. :2.0000   
##   
## MaxHR ExAng Oldpeak Slope   
## Min. : 71.0 Min. :0.0000 Min. :0.00 Min. :1.000   
## 1st Qu.:133.5 1st Qu.:0.0000 1st Qu.:0.00 1st Qu.:1.000   
## Median :153.0 Median :0.0000 Median :0.80 Median :2.000   
## Mean :149.6 Mean :0.3267 Mean :1.04 Mean :1.601   
## 3rd Qu.:166.0 3rd Qu.:1.0000 3rd Qu.:1.60 3rd Qu.:2.000   
## Max. :202.0 Max. :1.0000 Max. :6.20 Max. :3.000   
##   
## Ca Thal AHD   
## Min. :0.0000 fixed : 18 No :164   
## 1st Qu.:0.0000 normal :166 Yes:139   
## Median :0.0000 reversable:117   
## Mean :0.6722 NA's : 2   
## 3rd Qu.:1.0000   
## Max. :3.0000   
## NA's :4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CLEAN DATA, CORRELATION PLOT**

The data set contains columns with numeric, integer and factor data types. We started out with only three factors ChestPain, Thal, and AHD. We converted Sex, Fbs, RestECG, ExAng, Slope, and Ca to factors because they either have 2,3, or 4 factor levels. The rest of the variables will stay as integers, while ‘Oldpeak’ will be a numeric type because it contains decimals. Our group renamed the factor levels to its appropriate description. We also removed the rows with NA values – 6 NA’s in Ca and Thal columns – to avoid errors we used na.omit(). We can check if the data set contains NA values using anyNA() to check True or False. We also removed ‘X’ column because it’s identical to row numbers. We will identify them by their characteristics. The following is a brief description table of the attributes with their range from the data set:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Explanation** | **Type of Variable** | **Ranges** |
| Age | In Years | Integer | Continuous |
| Sex | Female or Male | Categorical | 0,1 |
| ChestPain | asymptomatic , nonanginal nontypical , typical | Categorical | 1,2,3,4 |
| RestBP | Rest Blood Pressure | Integer | Continuous |
| Chol | Cholesterol | Integer | Continuous |
| Fbs | Fasting blood suger > 120 mg/dl (1 = true) | Categorical | 0,1 |
| RestEcg | Normal , STT, Hypertrophy | Categorical | 0,1,2 |
| MaxHR | Maximum Heart Rate | Integer | Continuous |
| ExAng | Exercise induced angina,  (1 =yes) | Categorical | 0,1 |
| Oldpeak | ST depression induced by exercise relative to rest | Numeric | Continuous |
| Slope | slope of the peak exercise ST segment | Categorical | 1,2,3 |
| Ca | number of major vessels (0-3) colored by flourosopy | Categorical | 1,2,3,4 |
| Thal | normal, fixed, reversible | Categorical | 1,2,3 |
| AHD | Yes, No | Categorical | 1,2 |

Information found on <https://rpubs.com/mbbrigitte/heartdisease>

# 303 obs. of 15 variables (12 numeric, 3 factors)

anyNA(heart)

## [1] TRUE  
  
### Clean

heart1 <- heart[,-c(1)]  
heart1 <- na.omit(heart1)  
anyNA(heart1)

## [1] FALSE

# Convert to Factors   
heart1$Sex <- as.factor(heart1$Sex)  
levels(heart1$Sex) <- c("female","male")   
heart1$ChestPain <- as.factor(heart1$ChestPain)  
heart1$Fbs <- as.factor(heart1$Fbs)  
levels(heart1$Fbs) <- c("false","true")  
heart1$RestECG <- as.factor(heart1$RestECG)  
levels(heart1$RestECG) <- c("normal","stt","hypertrophy")  
heart1$ExAng <- as.factor(heart1$ExAng)  
levels(heart1$ExAng) <- c("no","yes")  
heart1$Slope <- as.factor(heart1$Slope)  
levels(heart1$Slope) <- c("upsloping","flat","downsloping")  
heart1$Ca <- as.factor(heart1$Ca)  
str(heart1)

## 'data.frame': 297 obs. of 14 variables:  
## $ Age : int 63 67 67 37 41 56 62 57 63 53 ...  
## $ Sex : Factor w/ 2 levels "female","male": 2 2 2 2 1 2 1 1 2 2 ...  
## $ ChestPain: Factor w/ 4 levels "asymptomatic",..: 4 1 1 2 3 3 1 1 1 1 ...  
## $ RestBP : int 145 160 120 130 130 120 140 120 130 140 ...  
## $ Chol : int 233 286 229 250 204 236 268 354 254 203 ...  
## $ Fbs : Factor w/ 2 levels "false","true": 2 1 1 1 1 1 1 1 1 2 ...  
## $ RestECG : Factor w/ 3 levels "normal","stt",..: 3 3 3 1 3 1 3 1 3 3 ...  
## $ MaxHR : int 150 108 129 187 172 178 160 163 147 155 ...  
## $ ExAng : Factor w/ 2 levels "no","yes": 1 2 2 1 1 1 1 2 1 2 ...  
## $ Oldpeak : num 2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...  
## $ Slope : Factor w/ 3 levels "upsloping","flat",..: 3 2 2 3 1 1 3 1 2 3 ...  
## $ Ca : Factor w/ 4 levels "0","1","2","3": 1 4 3 1 1 1 3 1 2 1 ...  
## $ Thal : Factor w/ 3 levels "fixed","normal",..: 1 2 3 2 2 2 2 2 3 3 ...  
## $ AHD : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 2 1 2 2 ...  
## - attr(\*, "na.action")= 'omit' Named int 88 167 193 267 288 303  
## ..- attr(\*, "names")= chr "88" "167" "193" "267" ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

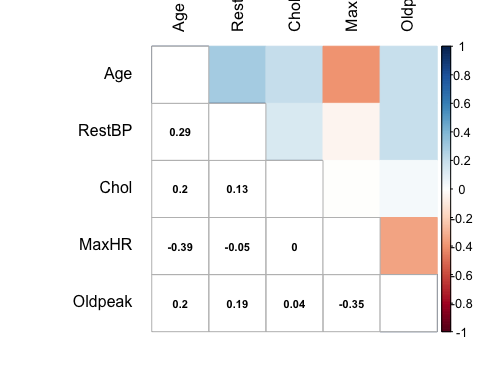
**EXPLORATORY DATA ANALYSIS**

We first looked at the numeric variables to check their correlation with one another. As seen in the chart below, there isn’t any strong correlation between the numeric variables.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Correlation of continuous variables  
library(corrplot)

heart2 <- heart1[, c("Age","RestBP","Chol","MaxHR","Oldpeak")]  
heart2 <- as.data.frame(heart2)  
g <- cor(heart2)  
corrplot.mixed(g, upper = "color", lower = "number", lower.col = "black", number.cex = .7,tl.pos = "lt", tl.col = "black", tl.offset=1)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Next, we looked at each categorical variable in a frequency table with AHD. When looking at Sex we see that the probability of males getting AHD is higher than females. For ChestPain, the asymptomatic factor sees more yes outcomes than the other three categories. In Fbs, or fasting blood sugar, which is true if the person is under or equal to 120 Fbs, false is the majority of the outcomes. We observed from the table that if a patient doesn’t get exercise induced angina (ExAng), AHD shows up less often. When a patient has 0 colored vessels, it appears that a no outcome happens more often. The more vessels that are colored, the more yes outcomes. An upsloping Slope has a higher chance of no outcomes. Most patients that have Thal of normal show more no outcomes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

###Tables of variables  
not\_num <- heart1[, -which(names(heart1) %in% c("Age","RestBP", "Chol", "MaxHR", "Oldpeak"))]  
label\_val <- function(x){  
 w <- table( x , AHD = heart1$AHD)  
 return(w)  
}  
apply(not\_num, 2, label\_val)

## $Sex  
## AHD  
## x No Yes  
## female 71 25  
## male 89 112  
##   
## $ChestPain  
## AHD  
## x No Yes  
## asymptomatic 39 103  
## nonanginal 65 18  
## nontypical 40 9  
## typical 16 7  
##   
## $Fbs  
## AHD  
## x No Yes  
## false 137 117  
## true 23 20  
##   
## $RestECG  
## AHD  
## x No Yes  
## hypertrophy 67 79  
## normal 92 55  
## stt 1 3  
##   
## $ExAng  
## AHD  
## x No Yes  
## no 137 63  
## yes 23 74  
##

##

$Slope  
## AHD  
## x No Yes  
## downsloping 9 12  
## flat 48 89  
## upsloping 103 36  
##   
## $Ca  
## AHD  
## x No Yes  
## 0 129 45  
## 1 21 44  
## 2 7 31  
## 3 3 17  
##   
## $Thal  
## AHD  
## x No Yes  
## fixed 6 12  
## normal 127 37  
## reversable 27 88  
##   
## $AHD  
## AHD  
## x No Yes  
## No 160 0  
## Yes 0 137

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To better understand the data, the mean, median, and standard deviations of all the numeric variables were taken. The mean shows that yes outcomes on average are older. The standard deviation of age for the no outcomes shows that the data is more spread out. The mean and median show cholesterol is higher in patients with AHD. The MaxHR is lower in patients with AHD.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#Mean, median, and SD for all columns that are not factors ###############   
mean <- sapply(heart1[,c("Age", "RestBP", "Chol", "MaxHR", "Oldpeak")], tapply, INDEX=heart1$AHD, mean)  
median <- sapply(heart1[,c("Age", "RestBP", "Chol", "MaxHR", "Oldpeak")], tapply, INDEX=heart1$AHD, median)  
sd <- sapply(heart1[,c("Age", "RestBP", "Chol", "MaxHR", "Oldpeak")], tapply, INDEX=heart1$AHD, sd)  
stat\_list <- list(MEAN=mean, MEDIAN=median, SD=sd)  
stat\_list

## $MEAN  
## Age RestBP Chol MaxHR Oldpeak  
## No 52.64375 129.175 243.4938 158.5813 0.598750  
## Yes 56.75912 134.635 251.8540 139.1095 1.589051  
##   
## $MEDIAN  
## Age RestBP Chol MaxHR Oldpeak  
## No 52 130 235.5 161 0.2  
## Yes 58 130 253.0 142 1.4  
##   
## $SD  
## Age RestBP Chol MaxHR Oldpeak  
## No 9.551151 16.37399 53.75755 19.04330 0.7871601  
## Yes 7.899670 18.89673 49.67994 22.71067 1.3050061

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

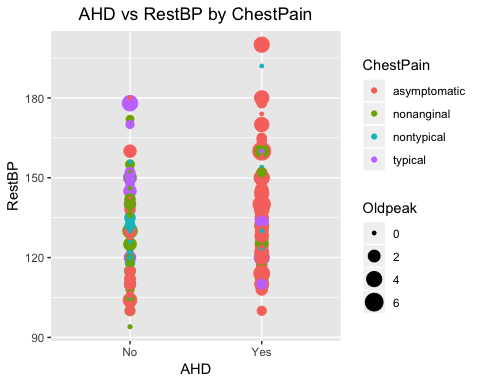
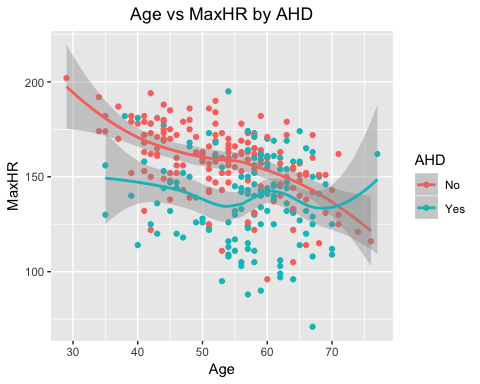
The graphs below give us insight into some variables in the data set and their interactions. The first graph shows that MaxHR decreases with the patients’ age, and the chances of AHD increases. The graph also showed as a patient gets older the ability of MaxHR to discriminate between “yes” and “no” decreases. The second graph shows that ChestPain varies in no outcomes, but asymptotic is seen more frequently in yes outcomes. It also shows that Oldpeak (depression induced by exercise) is generally higher for patients with AHD.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Plots: variables  
library(ggplot2)

ggplot(heart1, aes(x=Age, y=MaxHR, color=AHD)) + geom\_point() + geom\_smooth(method = "loess") +  
 ggtitle("Age vs MaxHR by AHD") + theme(plot.title = element\_text(hjust = 0.5))

ggplot(heart1, aes(x=AHD, y=RestBP, color= ChestPain, size=Oldpeak)) + geom\_point() +   
 ggtitle("AHD vs RestBP by ChestPain") + theme(plot.title = element\_text(hjust = 0.5))

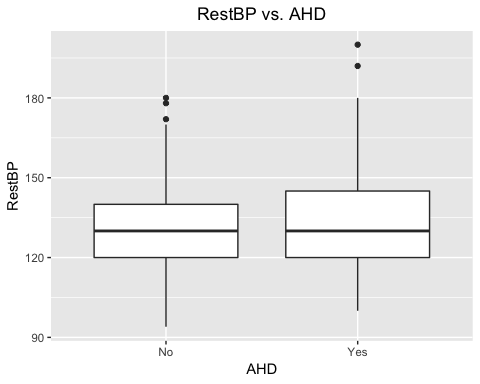
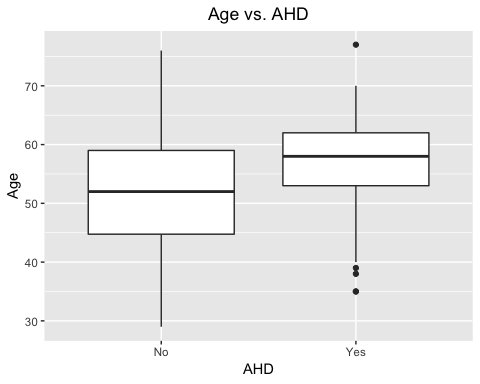


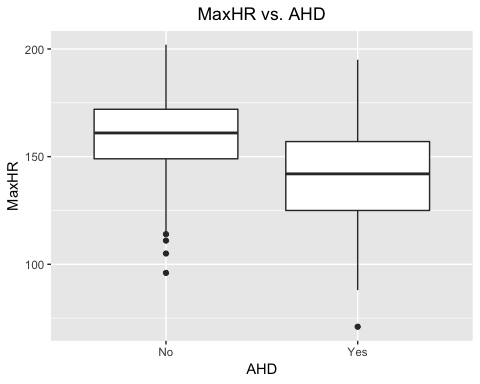
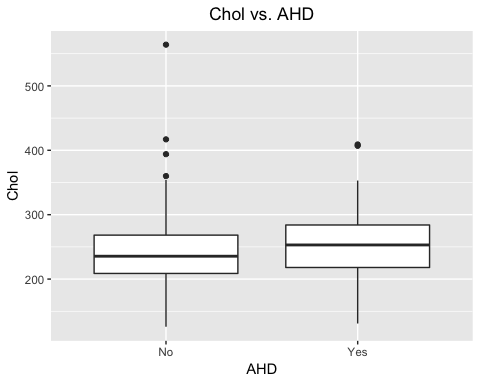
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

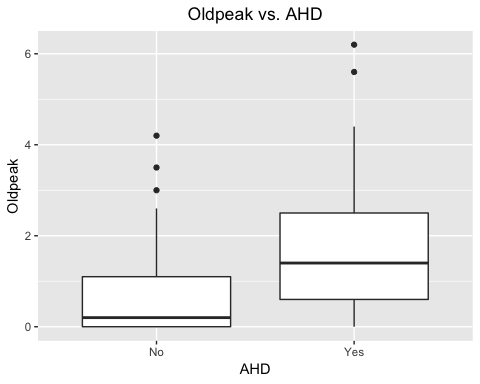
Next our group constructed Boxplots with AHD on the x-axis and the numeric variables on the y-axis. Observations showed similar statistics to the mean, median and standard deviations. Age for yes outcome is a bit higher, and the no outcomes have a wider spread. Cholesterol is a bit higher in yes outcomes. The Boxplot of Oldpeak showed that yes outcomes tend to be higher. We seemed to miss that in the statistics.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

####### Box plots   
####################################################  
num <- heart1[,c("Age", "RestBP", "Chol", "MaxHR", "Oldpeak", "AHD")]  
for(i in names(num)[-6]){  
 plot <- ggplot(num, aes\_string("AHD", y = i)) + geom\_boxplot() + ggtitle(paste0(i, " vs. AHD")) +   
 theme(plot.title = element\_text(hjust = 0.5))  
 print(plot)  
}





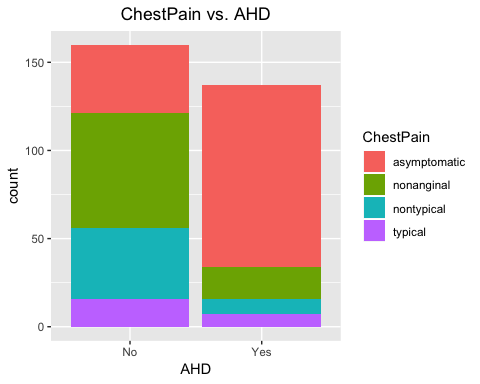
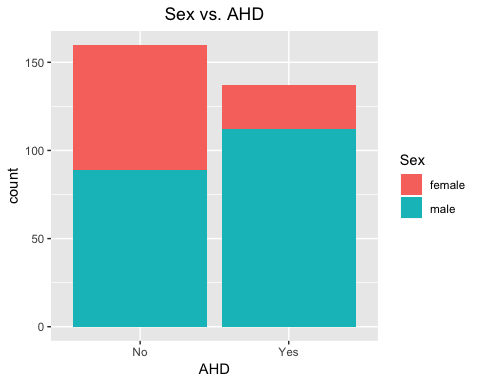


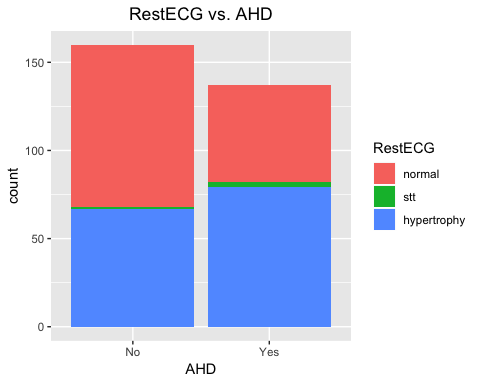
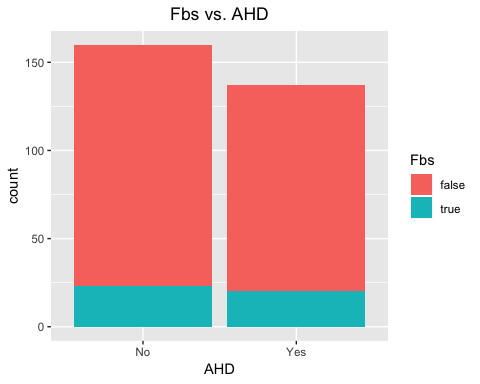
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

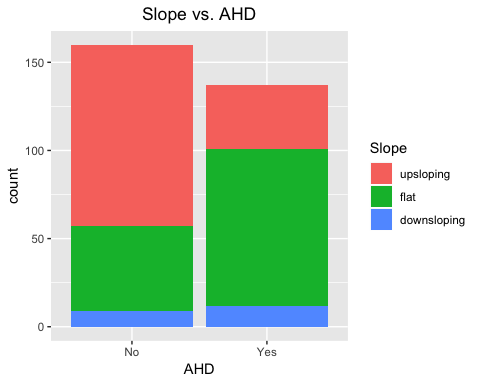
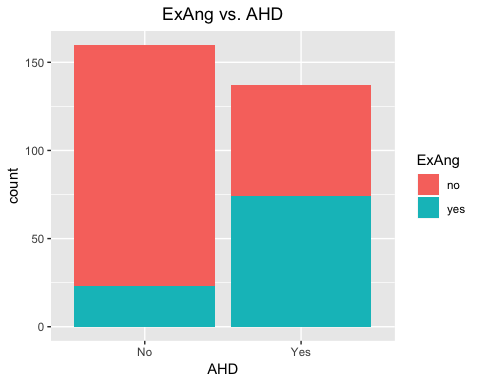
Next, we crafted bar charts of the same information from the tables earlier, which gave our group more of an understanding of the data. The AHD variable is on the x-axis and the categorical variable on the y-axis. As seen in the ChestPain graph, the asymptomatic patients make up the majority of the yes outcomes. The bar charts for Fbs and RestECG show that they may not be good predictors for AHD because of the similarity in yes and no outcomes. The boxplot of Slope shows that a flat slope makes up a majority of the yes patients, which we missed from the tables.

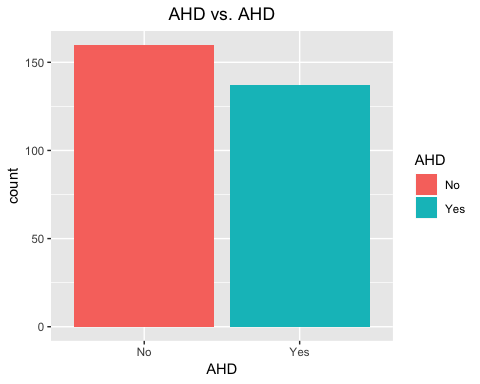
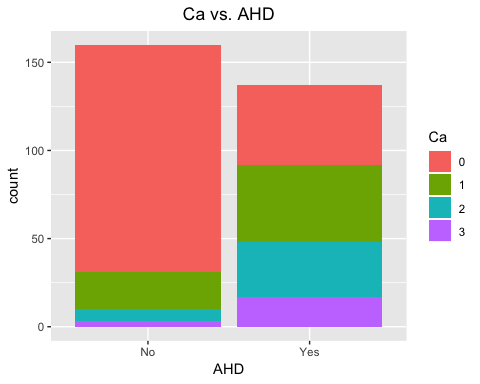
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

######## Bar Chart   
not\_num <- heart1[ , -which(names(heart1) %in% c("Age", "RestBP", "Chol", "MaxHR", "Oldpeak"))]  
for(i in names(not\_num)[-8]){  
 plot <- ggplot(not\_num, aes\_string("AHD", fill = i)) + geom\_bar() + ggtitle(paste0(i, " vs. AHD")) +   
 theme(plot.title = element\_text(hjust = 0.5))   
 print(plot)  
}









\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TRAIN & TEST DATA SETS**

The heart data set will be split into train and test data sets. We decided to use sample() to randomly choose 80% of the rows for the train data set and the rest for test. We also used set.seed(123) to ensure producing the same results every time, no matter the computer we worked on.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Train & Test Data Sets  
set.seed(123)  
s <- sample(1:nrow(heart1), nrow(heart1)\*0.8)  
train <- heart1[s,]  
test <- heart1[-s,]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**LOGISTIC REGRESSION**

First, we created an acc() function to save time and code lines for getting the accuracy of the models. Then, we performed individual logistic regression using the libraries for functions that provide statistical details.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Accuracy Function for models #########################################  
acc <- function(x){  
 w <- sum(diag(x))/ sum(x)\*100  
 return(w)  
}  
### Individual logistic regression ###################################  
library(tidyverse)

library(broom)  
library(pROC)

**TrainAge**

trainAge <- glm(AHD ~ Age, data=train, family='binomial')   
summary(trainAge)

##   
## Call:  
## glm(formula = AHD ~ Age, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6209 -1.1111 -0.8038 1.1378 1.6499   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.82791 0.82708 -3.419 0.000628 \*\*\*  
## Age 0.05038 0.01490 3.380 0.000724 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 316.04 on 235 degrees of freedom  
## AIC: 320.04  
##   
## Number of Fisher Scoring iterations: 4

pred\_age <- predict(trainAge, train, type = "response")  
roccurve\_age <- roc(train$AHD ~ pred\_age)  
pred2\_age <- ifelse(pred\_age > 0.5, 1,0)  
tab\_age <- table(Pred = pred2\_age, Actual = train$AHD)  
tab\_age

## Actual  
## Pred No Yes  
## 0 78 46  
## 1 45 68

**TrainSex**

trainSex <- glm(AHD ~ Sex, data=train, family='binomial')   
summary(trainSex)

##   
## Call:  
## glm(formula = AHD ~ Sex, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3335 -1.3335 -0.7466 1.0290 1.6815   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.1350 0.2709 -4.189 2.80e-05 \*\*\*  
## Sexmale 1.4946 0.3143 4.756 1.97e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 302.89 on 235 degrees of freedom  
## AIC: 306.89  
##   
## Number of Fisher Scoring iterations: 4

pred\_sex <- predict(trainSex, train, type = "response")  
roccurve\_sex <- roc(train$AHD ~ pred\_sex)  
pred2\_sex <- ifelse(pred\_sex > 0.5, 1,0)  
tab\_sex <- table(Pred = pred2\_sex, Actual = train$AHD)  
tab\_sex

## Actual  
## Pred No Yes  
## 0 56 18  
## 1 67 96

**TrainChestPain**

trainChestPain <- glm(AHD ~ ChestPain, data=train, family='binomial')   
summary(trainChestPain)

##   
## Call:  
## glm(formula = AHD ~ ChestPain, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6176 -0.7585 -0.7521 0.7938 1.6739   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.9933 0.2137 4.647 3.36e-06 \*\*\*  
## ChestPainnonanginal -2.1113 0.3518 -6.002 1.95e-09 \*\*\*  
## ChestPainnontypical -2.0919 0.4403 -4.751 2.02e-06 \*\*\*  
## ChestPaintypical -1.6864 0.5099 -3.308 0.000941 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 273.81 on 233 degrees of freedom  
## AIC: 281.81  
##   
## Number of Fisher Scoring iterations: 4

pred\_cp <- predict(trainChestPain, train, type = "response")  
roccurve\_cp <- roc(train$AHD ~ pred\_cp)  
pred2\_cp <- ifelse(pred\_cp > 0.5, 1,0)  
tab\_cp <- table(Pred = pred2\_cp, Actual = train$AHD)  
tab\_cp

## Actual  
## Pred No Yes  
## 0 93 33  
## 1 30 81

**TrainRestBP**

trainRestBP <- glm(AHD ~ RestBP, data=train, family='binomial')   
summary(trainRestBP)

##   
## Call:  
## glm(formula = AHD ~ RestBP, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.4689 -1.1335 -0.9605 1.2088 1.4247   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.100216 0.966603 -2.173 0.0298 \*  
## RestBP 0.015354 0.007267 2.113 0.0346 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 323.60 on 235 degrees of freedom  
## AIC: 327.6  
##   
## Number of Fisher Scoring iterations: 4

pred\_rbp <- predict(trainRestBP, train, type = "response")  
roccurve\_rbp <- roc(train$AHD ~ pred\_rbp)  
pred2\_rbp <- ifelse(pred\_rbp > 0.5, 1,0)  
tab\_rbp <- table(Pred = pred2\_rbp, Actual = train$AHD)  
tab\_rbp

## Actual  
## Pred No Yes  
## 0 82 66  
## 1 41 48

**TrainChol**

trainChol <- glm(AHD ~ Chol, data=train, family='binomial')   
summary(trainChol)

##   
## Call:  
## glm(formula = AHD ~ Chol, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.393 -1.140 -1.086 1.205 1.301   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.520826 0.624475 -0.834 0.404  
## Chol 0.001800 0.002471 0.728 0.466  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 327.68 on 235 degrees of freedom  
## AIC: 331.68  
##   
## Number of Fisher Scoring iterations: 3

pred\_chol <- predict(trainChol, train, type = "response")  
roccurve\_chol <- roc(train$AHD ~ pred\_chol)  
pred2\_chol <- ifelse(pred\_chol > 0.5, 1,0)  
tab\_chol <- table(Pred = pred2\_chol, Actual = train$AHD)  
tab\_chol

## Actual  
## Pred No Yes  
## 0 102 93  
## 1 21 21

**TrainFbs**

trainFbs <- glm(AHD ~ Fbs, data=train, family='binomial')   
summary(trainFbs)

##   
## Call:  
## glm(formula = AHD ~ Fbs, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.177 -1.141 -1.141 1.214 1.214   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.08618 0.13847 -0.622 0.534  
## Fbstrue 0.08618 0.40253 0.214 0.830  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 328.16 on 235 degrees of freedom  
## AIC: 332.16  
##   
## Number of Fisher Scoring iterations: 3

pred\_fbs <- predict(trainFbs, train, type = "response")  
roccurve\_fbs <- roc(train$AHD ~ pred\_fbs)  
pred2\_fbs <- ifelse(pred\_fbs > 0.5, 1,0)  
tab\_fbs <- table(Pred = pred2\_fbs, Actual = train$AHD)  
tab\_fbs

## Actual  
## Pred No Yes  
## 0 123 114

**TrainRestECG**

trainRestECG <- glm(AHD ~ RestECG, data=train, family='binomial')   
summary(trainRestECG)

##   
## Call:  
## glm(formula = AHD ~ RestECG, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.665 -1.031 -1.031 1.114 1.331   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.3545 0.1903 -1.863 0.0624 .  
## RestECGstt 1.4532 1.1703 1.242 0.2143   
## RestECGhypertrophy 0.5061 0.2646 1.913 0.0558 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 323.30 on 234 degrees of freedom  
## AIC: 329.3  
##   
## Number of Fisher Scoring iterations: 4

pred\_ecg <- predict(trainRestECG, train, type = "response")  
roccurve\_ecg <- roc(train$AHD ~ pred\_ecg)  
pred2\_ecg <- ifelse(pred\_ecg > 0.5, 1,0)  
tab\_ecg <- table(Pred = pred2\_ecg, Actual = train$AHD)  
tab\_ecg

## Actual  
## Pred No Yes  
## 0 67 47  
## 1 56 67

**TrainMaxHR**

trainMaxHR <- glm(AHD ~ MaxHR, data=train, family='binomial')   
summary(trainMaxHR)

##   
## Call:  
## glm(formula = AHD ~ MaxHR, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.999 -0.979 -0.597 1.061 2.052   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 6.317145 1.129113 5.595 2.21e-08 \*\*\*  
## MaxHR -0.042522 0.007424 -5.728 1.02e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 286.97 on 235 degrees of freedom  
## AIC: 290.97  
##   
## Number of Fisher Scoring iterations: 4

pred\_hr <- predict(trainMaxHR, train, type = "response")  
roccurve\_hr <- roc(train$AHD ~ pred\_hr)  
pred2\_hr <- ifelse(pred\_hr > 0.5, 1,0)  
tab\_hr <- table(Pred = pred2\_hr, Actual = train$AHD)  
tab\_hr

## Actual  
## Pred No Yes  
## 0 94 43  
## 1 29 71

**TrainExAng**

trainExAng <- glm(AHD ~ ExAng, data=train, family='binomial')   
summary(trainExAng)

##   
## Call:  
## glm(formula = AHD ~ ExAng, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7653 -0.9040 -0.9040 0.6876 1.4781   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.6838 0.1669 -4.097 4.19e-05 \*\*\*  
## ExAngyes 2.0056 0.3272 6.130 8.77e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 283.64 on 235 degrees of freedom  
## AIC: 287.64  
##   
## Number of Fisher Scoring iterations: 4

pred\_ang <- predict(trainExAng, train, type = "response")  
roccurve\_ang <- roc(train$AHD ~ pred\_ang)  
pred2\_ang <- ifelse(pred\_ang > 0.5, 1,0)  
tab\_ang <- table(Pred = pred2\_ang, Actual = train$AHD)  
tab\_ang

## Actual  
## Pred No Yes  
## 0 107 54  
## 1 16 60

**TrainOldpeak**

trainOldpeak <- glm(AHD ~ Oldpeak, data=train, family='binomial')   
summary(trainOldpeak)

##   
## Call:  
## glm(formula = AHD ~ Oldpeak, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.3441 -0.8642 -0.8031 1.0383 1.6054   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.9662 0.1986 -4.864 1.15e-06 \*\*\*  
## Oldpeak 0.8685 0.1486 5.845 5.08e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 281.78 on 235 degrees of freedom  
## AIC: 285.78  
##   
## Number of Fisher Scoring iterations: 4

pred\_op <- predict(trainOldpeak, train, type = "response")  
roccurve\_op <- roc(train$AHD ~ pred\_op)  
pred2\_op <- ifelse(pred\_op > 0.5, 1,0)  
tab\_op <- table(Pred = pred2\_op, Actual = train$AHD)  
tab\_op

## Actual  
## Pred No Yes  
## 0 94 47  
## 1 29 67

**TrainSlope**

trainSlope <- glm(AHD ~ Slope, data=train, family='binomial')   
summary(trainSlope)

##   
## Call:  
## glm(formula = AHD ~ Slope, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.5007 -0.7832 -0.7832 0.8855 1.6317   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.0245 0.2203 -4.650 3.31e-06 \*\*\*  
## Slopeflat 1.7585 0.2994 5.873 4.28e-09 \*\*\*  
## Slopedownsloping 1.2252 0.5006 2.448 0.0144 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 289.80 on 234 degrees of freedom  
## AIC: 295.8  
##   
## Number of Fisher Scoring iterations: 4

pred\_sp <- predict(trainSlope, train, type = "response")  
roccurve\_sp <- roc(train$AHD ~ pred\_sp)  
pred2\_sp <- ifelse(pred\_sp > 0.5, 1,0)  
tab\_sp <- table(Pred = pred2\_sp, Actual = train$AHD)  
tab\_sp

## Actual  
## Pred No Yes  
## 0 78 28  
## 1 45 86

**TrainCa**

trainCa <- glm(AHD ~ Ca, data=train, family='binomial')   
summary(trainCa)

##   
## Call:  
## glm(formula = AHD ~ Ca, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.9103 -0.8075 -0.8075 0.9104 1.5996   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.9534 0.1935 -4.927 8.34e-07 \*\*\*  
## Ca1 1.6199 0.3422 4.734 2.20e-06 \*\*\*  
## Ca2 2.6021 0.5253 4.954 7.28e-07 \*\*\*  
## Ca3 2.4939 0.6650 3.750 0.000177 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 272.25 on 233 degrees of freedom  
## AIC: 280.25  
##   
## Number of Fisher Scoring iterations: 4

pred\_ca <- predict(trainCa, train, type = "response")  
roccurve\_ca <- roc(train$AHD ~ pred\_ca)  
pred2\_ca <- ifelse(pred\_ca > 0.5, 1,0)  
tab\_ca <- table(Pred = pred2\_ca, Actual = train$AHD)  
tab\_ca

## Actual  
## Pred No Yes  
## 0 96 37  
## 1 27 77

**TrainThal**

trainThal <- glm(AHD ~ Thal, data=train, family='binomial')   
summary(trainThal)

##   
## Call:  
## glm(formula = AHD ~ Thal, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7189 -0.7653 -0.7653 0.7199 1.6559   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.6931 0.5477 1.266 0.2057   
## Thalnormal -1.7714 0.5836 -3.035 0.0024 \*\*  
## Thalreversable 0.5250 0.6014 0.873 0.3827   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 265.23 on 234 degrees of freedom  
## AIC: 271.23  
##   
## Number of Fisher Scoring iterations: 4

pred\_t <- predict(trainThal, train, type = "response")  
roccurve\_t <- roc(train$AHD ~ pred\_t)  
pred2\_t <- ifelse(pred\_t > 0.5, 1,0)  
tab\_t <- table(Pred = pred2\_t, Actual = train$AHD)  
tab\_t

## Actual  
## Pred No Yes  
## 0 97 33  
## 1 26 81

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Backward Stepwise Regression**

First, we performed logistic regression on all the variables individually. Before getting into more complex models, we used backward stepwise regression to help pick 5 out of the 7 models. Backward stepwise regression is a method that fits regression models to find the best subset of variables in terms of significance and accuracy. It starts with all the variables in the model and works its way down until it goes through every possible combination for a model. It uses AIC (the lower the better) to get the best models. We created model 7 with the variables thought to be most significant.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Model 1**

### Models   
model <- glm(AHD ~., data = train, family = "binomial")  
summary(model)

##   
## Call:  
## glm(formula = AHD ~ ., family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.99925 -0.48213 -0.07463 0.32329 2.78268   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.030533 3.076284 -1.635 0.101994   
## Age -0.017435 0.027798 -0.627 0.530529   
## Sexmale 2.204728 0.642642 3.431 0.000602 \*\*\*  
## ChestPainnonanginal -1.749436 0.567086 -3.085 0.002036 \*\*   
## ChestPainnontypical -0.477396 0.618606 -0.772 0.440276   
## ChestPaintypical -2.285238 0.754624 -3.028 0.002459 \*\*   
## RestBP 0.026808 0.012699 2.111 0.034777 \*   
## Chol 0.002069 0.004606 0.449 0.653226   
## Fbstrue -0.385481 0.686523 -0.561 0.574458   
## RestECGstt 1.238269 2.296952 0.539 0.589823   
## RestECGhypertrophy 0.273001 0.436223 0.626 0.531427   
## MaxHR -0.014717 0.012657 -1.163 0.244941   
## ExAngyes 0.837018 0.497097 1.684 0.092218 .   
## Oldpeak 0.337001 0.248782 1.355 0.175543   
## Slopeflat 1.582824 0.549587 2.880 0.003976 \*\*   
## Slopedownsloping 0.600006 0.927612 0.647 0.517743   
## Ca1 1.946410 0.550077 3.538 0.000403 \*\*\*  
## Ca2 3.540357 0.896848 3.948 7.9e-05 \*\*\*  
## Ca3 2.015544 0.953616 2.114 0.034551 \*   
## Thalnormal 0.382949 0.877964 0.436 0.662707   
## Thalreversable 1.564187 0.873863 1.790 0.073459 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 152.22 on 216 degrees of freedom  
## AIC: 194.22  
##   
## Number of Fisher Scoring iterations: 6

pred <- predict(model, train, type = "response")  
roccurve <- roc(train$AHD ~ pred)  
pred1 <- ifelse(pred > 0.5, 1,0)  
tab1 <- table(Pred = pred1, Actual = train$AHD)

## Used stepwise regression to get our best models   
# Model with the lowest AIC   
step(model, scope = list(upper = model), direction = "backward")

## Start: AIC=194.22  
## AHD ~ Age + Sex + ChestPain + RestBP + Chol + Fbs + RestECG +   
## MaxHR + ExAng + Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - RestECG 2 152.85 190.85  
## - Chol 1 152.41 192.41  
## - Fbs 1 152.53 192.53  
## - Age 1 152.61 192.61  
## - MaxHR 1 153.61 193.61  
## - Oldpeak 1 154.13 194.13  
## <none> 152.22 194.22  
## - ExAng 1 155.06 195.06  
## - RestBP 1 156.92 196.92  
## - Thal 2 159.89 197.89  
## - Slope 2 161.55 199.55  
## - ChestPain 3 167.87 203.87  
## - Sex 1 165.82 205.82  
## - Ca 3 180.99 216.99  
##   
## Step: AIC=190.85  
## AHD ~ Age + Sex + ChestPain + RestBP + Chol + Fbs + MaxHR + ExAng +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Age 1 153.17 189.17  
## - Chol 1 153.21 189.21  
## - Fbs 1 153.23 189.23  
## - MaxHR 1 154.35 190.35  
## <none> 152.85 190.85  
## - Oldpeak 1 154.90 190.90  
## - ExAng 1 155.62 191.62  
## - RestBP 1 158.19 194.19  
## - Thal 2 160.35 194.35  
## - Slope 2 162.38 196.38  
## - ChestPain 3 168.35 200.35  
## - Sex 1 166.89 202.89  
## - Ca 3 181.49 213.49  
##   
## Step: AIC=189.17  
## AHD ~ Sex + ChestPain + RestBP + Chol + Fbs + MaxHR + ExAng +   
## Oldpeak + Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Chol 1 153.44 187.44  
## - Fbs 1 153.56 187.56  
## - MaxHR 1 154.37 188.37  
## <none> 153.17 189.17  
## - Oldpeak 1 155.45 189.45  
## - ExAng 1 155.88 189.88  
## - RestBP 1 158.20 192.20  
## - Thal 2 160.63 192.63  
## - Slope 2 162.40 194.40  
## - ChestPain 3 168.99 198.99  
## - Sex 1 167.86 201.86  
## - Ca 3 183.41 213.41  
##   
## Step: AIC=187.44  
## AHD ~ Sex + ChestPain + RestBP + Fbs + MaxHR + ExAng + Oldpeak +   
## Slope + Ca + Thal  
##   
## Df Deviance AIC  
## - Fbs 1 153.82 185.82  
## - MaxHR 1 154.56 186.56  
## <none> 153.44 187.44  
## - Oldpeak 1 155.76 187.76  
## - ExAng 1 156.16 188.16  
## - RestBP 1 158.65 190.65  
## - Thal 2 161.08 191.08  
## - Slope 2 162.97 192.97  
## - ChestPain 3 169.71 197.71  
## - Sex 1 167.90 199.90  
## - Ca 3 183.98 211.98  
##   
## Step: AIC=185.82  
## AHD ~ Sex + ChestPain + RestBP + MaxHR + ExAng + Oldpeak + Slope +   
## Ca + Thal  
##   
## Df Deviance AIC  
## - MaxHR 1 154.87 184.87  
## <none> 153.82 185.82  
## - ExAng 1 156.49 186.49  
## - Oldpeak 1 156.61 186.61  
## - RestBP 1 158.74 188.74  
## - Thal 2 161.95 189.95  
## - Slope 2 163.07 191.07  
## - ChestPain 3 171.46 197.46  
## - Sex 1 168.02 198.02  
## - Ca 3 184.13 210.13  
##   
## Step: AIC=184.87  
## AHD ~ Sex + ChestPain + RestBP + ExAng + Oldpeak + Slope + Ca +   
## Thal  
##   
## Df Deviance AIC  
## <none> 154.87 184.87  
## - Oldpeak 1 157.92 185.92  
## - ExAng 1 158.24 186.24  
## - RestBP 1 159.41 187.41  
## - Thal 2 162.63 188.63  
## - Slope 2 167.26 193.26  
## - Sex 1 168.67 196.67  
## - ChestPain 3 174.62 198.62  
## - Ca 3 186.41 210.41

##   
## Call: glm(formula = AHD ~ Sex + ChestPain + RestBP + ExAng + Oldpeak +   
## Slope + Ca + Thal, family = "binomial", data = train)  
##   
## Coefficients:  
## (Intercept) Sexmale ChestPainnonanginal   
## -7.19608 2.12545 -1.95330   
## ChestPainnontypical ChestPaintypical RestBP   
## -0.60757 -2.40501 0.02424   
## ExAngyes Oldpeak Slopeflat   
## 0.88802 0.40363 1.68302   
## Slopedownsloping Ca1 Ca2   
## 0.59804 1.92238 3.27691   
## Ca3 Thalnormal Thalreversable   
## 1.99755 0.36758 1.52795   
## Degrees of Freedom: 236 Total (i.e. Null); 222 Residual  
## Null Deviance: 328.2   
## Residual Deviance: 154.9 AIC: 184.9

**Model 2**

## Model 2  
model2 <- glm(AHD ~ Age + Sex + ChestPain + RestBP + Chol + Fbs + MaxHR + ExAng +   
 Oldpeak + Slope + Ca + Thal , data = train, family = "binomial")  
summary(model2)

##   
## Call:  
## glm(formula = AHD ~ Age + Sex + ChestPain + RestBP + Chol + Fbs +   
## MaxHR + ExAng + Oldpeak + Slope + Ca + Thal, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.0620 -0.4628 -0.0774 0.3623 2.7446   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.177467 3.053929 -1.695 0.090010 .   
## Age -0.015431 0.027451 -0.562 0.574021   
## Sexmale 2.203347 0.633256 3.479 0.000503 \*\*\*  
## ChestPainnonanginal -1.749035 0.568662 -3.076 0.002100 \*\*   
## ChestPainnontypical -0.491915 0.619904 -0.794 0.427467   
## ChestPaintypical -2.253444 0.747402 -3.015 0.002569 \*\*   
## RestBP 0.027791 0.012472 2.228 0.025865 \*   
## Chol 0.002699 0.004462 0.605 0.545268   
## Fbstrue -0.423022 0.689540 -0.613 0.539556   
## MaxHR -0.015270 0.012687 -1.204 0.228741   
## ExAngyes 0.822551 0.495336 1.661 0.096795 .   
## Oldpeak 0.341505 0.244088 1.399 0.161781   
## Slopeflat 1.596329 0.547652 2.915 0.003559 \*\*   
## Slopedownsloping 0.633805 0.916540 0.692 0.489239   
## Ca1 1.962197 0.549035 3.574 0.000352 \*\*\*  
## Ca2 3.506434 0.893039 3.926 8.62e-05 \*\*\*  
## Ca3 2.057150 0.964537 2.133 0.032942 \*   
## Thalnormal 0.369222 0.867829 0.425 0.670505   
## Thalreversable 1.527422 0.862905 1.770 0.076712 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 152.85 on 218 degrees of freedom  
## AIC: 190.85  
##   
## Number of Fisher Scoring iterations: 6

pred2 <- predict(model2, train, type = "response")  
roccurve2 <- roc(train$AHD ~ pred2)  
pred2 <- ifelse(pred2 > 0.5, 1,0)  
tab2 <- table(Pred = pred2, Actual = train$AHD)

**Model 3**

# Model 3  
model3 <- glm(AHD ~ Sex + ChestPain + RestBP + Chol + Fbs + MaxHR + ExAng +   
 Oldpeak + Slope + Ca + Thal,data = train, family = "binomial")  
summary(model3)

##   
## Call:  
## glm(formula = AHD ~ Sex + ChestPain + RestBP + Chol + Fbs + MaxHR +   
## ExAng + Oldpeak + Slope + Ca + Thal, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.05092 -0.46088 -0.08028 0.37864 2.75534   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.982118 2.706035 -2.211 0.027060 \*   
## Sexmale 2.236421 0.629968 3.550 0.000385 \*\*\*  
## ChestPainnonanginal -1.774963 0.568253 -3.124 0.001787 \*\*   
## ChestPainnontypical -0.507813 0.618284 -0.821 0.411460   
## ChestPaintypical -2.254690 0.744553 -3.028 0.002460 \*\*   
## RestBP 0.025630 0.011775 2.177 0.029516 \*   
## Chol 0.002285 0.004354 0.525 0.599663   
## Fbstrue -0.429823 0.686845 -0.626 0.531450   
## MaxHR -0.012778 0.011860 -1.077 0.281325   
## ExAngyes 0.813366 0.494050 1.646 0.099697 .   
## Oldpeak 0.357760 0.242056 1.478 0.139407   
## Slopeflat 1.551372 0.539090 2.878 0.004005 \*\*   
## Slopedownsloping 0.602630 0.914809 0.659 0.510056   
## Ca1 1.893296 0.533332 3.550 0.000385 \*\*\*  
## Ca2 3.360735 0.848908 3.959 7.53e-05 \*\*\*  
## Ca3 1.966445 0.957894 2.053 0.040084 \*   
## Thalnormal 0.366327 0.865811 0.423 0.672221   
## Thalreversable 1.520653 0.860956 1.766 0.077356 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 153.17 on 219 degrees of freedom  
## AIC: 189.17  
##   
## Number of Fisher Scoring iterations: 6

pred3 <- predict(model3, train, type = "response")  
roccurve3 <- roc(train$AHD ~ pred3)  
pred3 <- ifelse(pred3 > 0.5, 1,0)  
tab3 <- table(Pred = pred3, Actual = train$AHD)

**Model 4**

# Model 4  
model4 <- glm(AHD ~ Sex + ChestPain + RestBP + Fbs + MaxHR + ExAng + Oldpeak +   
 Slope + Ca + Thal, data = train, family = "binomial")  
summary(model4)

##   
## Call:  
## glm(formula = AHD ~ Sex + ChestPain + RestBP + Fbs + MaxHR +   
## ExAng + Oldpeak + Slope + Ca + Thal, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.03389 -0.45009 -0.07373 0.36038 2.76890   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.55747 2.55536 -2.175 0.029643 \*   
## Sexmale 2.18623 0.62309 3.509 0.000450 \*\*\*  
## ChestPainnonanginal -1.79565 0.56803 -3.161 0.001571 \*\*   
## ChestPainnontypical -0.51457 0.61774 -0.833 0.404848   
## ChestPaintypical -2.28303 0.74209 -3.077 0.002094 \*\*   
## RestBP 0.02601 0.01175 2.214 0.026852 \*   
## Fbstrue -0.42185 0.68364 -0.617 0.537188   
## MaxHR -0.01227 0.01175 -1.045 0.296204   
## ExAngyes 0.81013 0.49180 1.647 0.099500 .   
## Oldpeak 0.36098 0.24216 1.491 0.136055   
## Slopeflat 1.57216 0.53845 2.920 0.003503 \*\*   
## Slopedownsloping 0.60881 0.91237 0.667 0.504593   
## Ca1 1.89894 0.53404 3.556 0.000377 \*\*\*  
## Ca2 3.35085 0.84339 3.973 7.1e-05 \*\*\*  
## Ca3 1.97962 0.94493 2.095 0.036171 \*   
## Thalnormal 0.41108 0.85824 0.479 0.631955   
## Thalreversable 1.56873 0.85237 1.840 0.065705 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 153.44 on 220 degrees of freedom  
## AIC: 187.44  
##   
## Number of Fisher Scoring iterations: 6

pred4 <- predict(model4, train, type = "response")  
roccurve4 <- roc(train$AHD ~ pred4)  
pred4 <- ifelse(pred4 > 0.5, 1,0)  
tab4 <- table(Pred = pred4, Actual = train$AHD)

**Model 5**

#Model 5  
model5 <- glm(AHD ~ Sex + ChestPain + RestBP + MaxHR + ExAng + Oldpeak + Slope +   
 Ca + Thal , data = train, family = "binomial")  
summary(model5)

##   
## Call:  
## glm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + ExAng +   
## Oldpeak + Slope + Ca + Thal, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.00039 -0.45552 -0.08663 0.36274 2.78450   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.53732 2.55169 -2.170 0.030002 \*   
## Sexmale 2.16521 0.62368 3.472 0.000517 \*\*\*  
## ChestPainnonanginal -1.85842 0.56185 -3.308 0.000941 \*\*\*  
## ChestPainnontypical -0.53324 0.61244 -0.871 0.383922   
## ChestPaintypical -2.33102 0.74029 -3.149 0.001639 \*\*   
## RestBP 0.02498 0.01162 2.150 0.031526 \*   
## MaxHR -0.01185 0.01173 -1.010 0.312658   
## ExAngyes 0.80391 0.49269 1.632 0.102748   
## Oldpeak 0.38753 0.23845 1.625 0.104117   
## Slopeflat 1.53312 0.53423 2.870 0.004108 \*\*   
## Slopedownsloping 0.53928 0.90484 0.596 0.551181   
## Ca1 1.84827 0.52534 3.518 0.000434 \*\*\*  
## Ca2 3.29991 0.83495 3.952 7.74e-05 \*\*\*  
## Ca3 1.84701 0.90328 2.045 0.040876 \*   
## Thalnormal 0.47744 0.85045 0.561 0.574525   
## Thalreversable 1.64553 0.84220 1.954 0.050720 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 153.82 on 221 degrees of freedom  
## AIC: 185.82  
##   
## Number of Fisher Scoring iterations: 6

pred5 <- predict(model5, train, type = "response")  
roccurve5 <- roc(train$AHD ~ pred5)  
pred5 <- ifelse(pred5 > 0.5, 1,0)  
tab5 <- table(Pred = pred5, Actual = train$AHD)

**Model 6**

# Model 6  
model6 <- glm(AHD ~ Sex + ChestPain + RestBP + ExAng + Oldpeak + Slope + Ca +   
 Thal , data = train, family = "binomial")  
summary(model6)

##   
## Call:  
## glm(formula = AHD ~ Sex + ChestPain + RestBP + ExAng + Oldpeak +   
## Slope + Ca + Thal, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.88089 -0.45947 -0.08808 0.39685 2.84821   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.19608 2.01158 -3.577 0.000347 \*\*\*  
## Sexmale 2.12545 0.61975 3.430 0.000605 \*\*\*  
## ChestPainnonanginal -1.95330 0.55255 -3.535 0.000408 \*\*\*  
## ChestPainnontypical -0.60757 0.60695 -1.001 0.316814   
## ChestPaintypical -2.40501 0.74225 -3.240 0.001195 \*\*   
## RestBP 0.02424 0.01174 2.064 0.038998 \*   
## ExAngyes 0.88802 0.48523 1.830 0.067236 .   
## Oldpeak 0.40363 0.23807 1.695 0.089987 .   
## Slopeflat 1.68302 0.51461 3.270 0.001074 \*\*   
## Slopedownsloping 0.59804 0.88275 0.677 0.498109   
## Ca1 1.92238 0.51898 3.704 0.000212 \*\*\*  
## Ca2 3.27691 0.84305 3.887 0.000101 \*\*\*  
## Ca3 1.99755 0.89996 2.220 0.026445 \*   
## Thalnormal 0.36758 0.83721 0.439 0.660620   
## Thalreversable 1.52795 0.82913 1.843 0.065355 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 154.87 on 222 degrees of freedom  
## AIC: 184.87  
##   
## Number of Fisher Scoring iterations: 6

pred6 <- predict(model6, train, type = "response")  
roccurve6 <- roc(train$AHD ~ pred6)  
pred6 <- ifelse(pred6 > 0.5, 1,0)  
tab6 <- table(Pred = pred6, Actual = train$AHD)

**Model 7**

# Model 7  
model7 <- glm(AHD ~ ChestPain + Slope + Ca + Thal + RestBP , data = train, family = "binomial")  
summary(model7)

##   
## Call:  
## glm(formula = AHD ~ ChestPain + Slope + Ca + Thal + RestBP, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.7520 -0.5836 -0.1807 0.4843 2.9565   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.71438 1.56301 -2.376 0.017481 \*   
## ChestPainnonanginal -2.04648 0.48604 -4.211 2.55e-05 \*\*\*  
## ChestPainnontypical -1.09749 0.56078 -1.957 0.050338 .   
## ChestPaintypical -1.89135 0.63677 -2.970 0.002976 \*\*   
## Slopeflat 1.62558 0.42105 3.861 0.000113 \*\*\*  
## Slopedownsloping 1.28548 0.71444 1.799 0.071973 .   
## Ca1 1.96520 0.46876 4.192 2.76e-05 \*\*\*  
## Ca2 2.76950 0.70386 3.935 8.33e-05 \*\*\*  
## Ca3 2.26878 0.92742 2.446 0.014431 \*   
## Thalnormal -0.85482 0.72989 -1.171 0.241531   
## Thalreversable 1.21128 0.75318 1.608 0.107785   
## RestBP 0.02091 0.01035 2.020 0.043424 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.21 on 236 degrees of freedom  
## Residual deviance: 177.79 on 225 degrees of freedom  
## AIC: 201.79  
##   
## Number of Fisher Scoring iterations: 5

pred7 <- predict(model7, train, type = "response")  
roccurve7 <- roc(train$AHD ~ pred7)  
pred7 <- ifelse(pred7 > 0.5, 1,0)  
tab7 <- table(Pred = pred7, Actual = train$AHD)

**Table of Model Information**

info\_list <- list(TrainAge = list(Accuracy = acc(tab\_age), Null\_Deviance = trainAge$null.deviance, Residual\_Deviance = trainAge$deviance, AIC = trainAge$aic, AUC = auc(roccurve\_age)),  
 TrainSex = list(Accuracy = acc(tab\_sex), Null\_Deviance = trainSex$null.deviance, Residual\_Deviance = trainSex$deviance, AIC = trainSex$aic, AUC = auc(roccurve\_sex)),  
 TrainChestPain = list(Accuracy = acc(tab\_cp), Null\_Deviance = trainChestPain$null.deviance, Residual\_Deviance = trainChestPain$deviance, AIC = trainChestPain$aic, AUC = auc(roccurve\_cp)),  
 TrainRestBP = list(Accuracy = acc(tab\_rbp), Null\_Deviance = trainRestBP$null.deviance, Residual\_Deviance = trainRestBP$deviance, AIC = trainRestBP$aic, AUC = auc(roccurve\_rbp)),  
 TrainChol = list(Accuracy = acc(tab\_chol), Null\_Deviance = trainChol$null.deviance, Residual\_Deviance = trainChol$deviance, AIC = trainChol$aic, AUC = auc(roccurve\_chol)),  
 TrainFbs = list(Accuracy = acc(tab\_fbs), Null\_Deviance = trainFbs$null.deviance, Residual\_Deviance = trainFbs$deviance, AIC = trainFbs$aic, AUC = auc(roccurve\_fbs)),  
 TrainRestECG = list(Accuracy = acc(tab\_ecg), Null\_Deviance = trainRestECG$null.deviance, Residual\_Deviance = trainRestECG$deviance, AIC = trainRestECG$aic, AUC = auc(roccurve\_ecg)),  
 TrainMaxHR = list(Accuracy = acc(tab\_hr), Null\_Deviance = trainMaxHR$null.deviance, Residual\_Deviance = trainMaxHR$deviance, AIC = trainMaxHR$aic, AUC = auc(roccurve\_hr)),  
 TrainExAng = list(Accuracy = acc(tab\_ang), Null\_Deviance = trainExAng$null.deviance, Residual\_Deviance = trainExAng$deviance, AIC = trainExAng$aic, AUC = auc(roccurve\_ang)),  
 TrainOldpeak = list(Accuracy = acc(tab\_op), Null\_Deviance = trainOldpeak$null.deviance, Residual\_Deviance = trainOldpeak$deviance, AIC = trainOldpeak$aic, AUC = auc(roccurve\_op)),  
 TrainSlope = list(Accuracy = acc(tab\_sp), Null\_Deviance = trainSlope$null.deviance, Residual\_Deviance = trainSlope$deviance, AIC = trainSlope$aic, AUC = auc(roccurve\_sp)),  
 TrainCa = list(Accuracy = acc(tab\_ca), Null\_Deviance = trainCa$null.deviance, Residual\_Deviance = trainCa$deviance, AIC = trainCa$aic, AUC = auc(roccurve\_ca)),  
 TrainThal = list(Accuracy = acc(tab\_t), Null\_Deviance = trainThal$null.deviance, Residual\_Deviance = trainThal$deviance, AIC = trainThal$aic, AUC = auc(roccurve\_t)),  
 Model1 = list(Accuracy = acc(tab1), Null\_Deviance = model$null.deviance, Residual\_Deviance = model$deviance, AIC = model$aic, AUC = auc(roccurve)),  
 Model2 = list(Accuracy = acc(tab2), Null\_Deviance = model2$null.deviance, Residual\_Deviance = model2$deviance, AIC = model2$aic, AUC = auc(roccurve2)),  
 Model3 = list(Accuracy = acc(tab3), Null\_Deviance = model3$null.deviance, Residual\_Deviance = model3$deviance, AIC = model3$aic, AUC = auc(roccurve3)),  
 Model4 = list(Accuracy = acc(tab4), Null\_Deviance = model4$null.deviance, Residual\_Deviance = model4$deviance, AIC = model4$aic, AUC = auc(roccurve4)),  
 Model5 = list(Accuracy = acc(tab5), Null\_Deviance = model5$null.deviance, Residual\_Deviance = model5$deviance, AIC = model5$aic, AUC = auc(roccurve5)),  
 Model6 = list(Accuracy = acc(tab6), Null\_Deviance = model6$null.deviance, Residual\_Deviance = model6$deviance, AIC = model6$aic, AUC = auc(roccurve6)),  
 Model7 = list(Accuracy = acc(tab7), Null\_Deviance = model7$null.deviance, Residual\_Deviance = model7$deviance, AIC = model7$aic, AUC = auc(roccurve7)))  
  
info1 <- do.call(rbind, info\_list)  
info2 <- as.data.frame(info1)  
info <- rownames\_to\_column(info2, "Models")  
info$Accuracy <- round(as.numeric(info$Accuracy),2)  
info$Null\_Deviance <- round(as.numeric(info$Null\_Deviance),2)  
info$Residual\_Deviance <- round(as.numeric(info$Residual\_Deviance),2)  
info$AUC <- round(as.numeric(info$AUC),3)  
info$AIC <- round(as.numeric(info$AIC),2)  
info

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Although the table that R gives us in the console looks comprehensive enough, we created a table that offers a better look.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Models | Accuracy | Null Deviance | Residual Deviance | AIC | AUC |
| TrainAge | 61.6 | 328.21 | 316.04 | 320.04 | 0.631 |
| TrainSex | 64.14 | 328.21 | 302.89 | 306.89 | 0.649 |
| TrainChestPain | 73.42 | 328.21 | 273.81 | 281.81 | 0.74 |
| TrainRestBP | 54.85 | 328.21 | 323.6 | 327.6 | 0.563 |
| TrainChol | 51.9 | 328.21 | 327.68 | 331.68 | 0.551 |
| TrainFbs | 51.9 | 328.21 | 328.16 | 332.16 | 0.504 |
| TrainRestECG | 56.54 | 328.21 | 323.3 | 329.3 | 0.57 |
| TrainMaxHR | 69.62 | 328.21 | 286.97 | 290.97 | 0.736 |
| TrainExAng | 70.46 | 328.21 | 283.64 | 287.64 | 0.698 |
| TrainOldpeak | 67.93 | 328.21 | 281.78 | 285.78 | 0.733 |
| TrainSlope | 69.2 | 328.21 | 289.8 | 295.8 | 0.704 |
| TrainCa | 73 | 328.21 | 272.25 | 280.25 | 0.745 |
| TrainThal | 75.11 | 328.21 | 265.23 | 271.23 | 0.755 |
| Model 1 | 86.08 | 328.21 | 152.22 | 194.22 | 0.937 |
| Model 2 | 86.5 | 328.21 | 152.85 | 190.85 | 0.936 |
| Model 3 | 85.65 | 328.21 | 153.17 | 189.17 | 0.936 |
| Model 4 | 85.65 | 328.21 | 153.44 | 187.44 | 0.936 |
| Model 5 | 84.81 | 328.21 | 153.82 | 185.82 | 0.935 |
| Model 6 | 85.23 | 328.21 | 154.87 | 184.87 | 0.936 |
| Model 7 | 84.81 | 328.21 | 177.79 | 201.79 | 0.914 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

To evaluate the performance of a logistic regression model, we must consider a few metrics:

**1. AIC (Akaike Information Criteria)** – The metric of adjusted R-squared in logistic regression is AIC. AIC is the measure of fit which penalizes a model for the number of model coefficients. Therefore, we always prefer a model with minimum AIC value.

**2. Null Deviance and Residual Deviance** – Null deviance indicates the response predicted by a model with nothing but an intercept. Lower the value, better the model. Residual deviance indicates the response predicted by a model of adding independent variables. Again, lower the value, better the model.

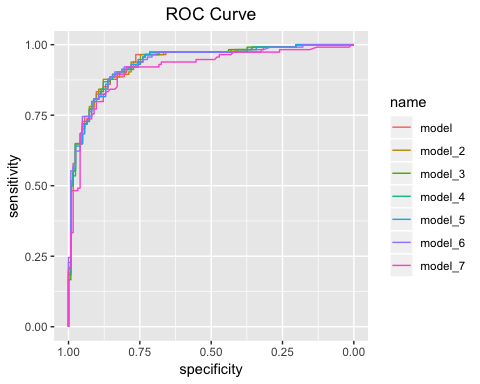
**3. AUC (Area Under Curve)** – The AUC is the area under ROC curve which will be plotted next. AUC is the measure the area under of the ROC curve. The closer to one the better the model is at differentiating between positive and negative classes; in our case, yes or no for AHD.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ROC (Receiver Operating Characteristics)**

ROC plots the false positive rate on the x-axis and the true positive rate on the y-axis. We want models that predict a higher true positive rate. So, we aim for a model with a curve that is closer to one on the true positive rate axis or y-axis. In this ROC, we see that most models were close together, but model 7 was a little below the rest. For that reason, we didn’t choose model 7 as our best model.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Accuracy Tables**

We looked for models with less false negative outcomes, which were the outcomes that were predicted to be no but are actually yes. In reality, false negative results are the worst for the hospital and the patients. Models 1,2,6 had the fewest false negative outcomes at 19.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

tables <- list(model1 = tab1,  
 model2 = tab2,  
 model3 = tab3,  
 model4 = tab4,  
 model5 = tab5,  
 model6 = tab6,   
 model7 = tab7)  
tables

##

$model1  
## Actual  
## Pred No Yes  
## 0 109 19  
## 1 14 95

## $model2  
## Actual  
## Pred No Yes  
## 0 110 19  
## 1 13 95  
##   
## $model3  
## Actual  
## Pred No Yes  
## 0 109 20  
## 1 14 94  
##   
##

$model4  
## Actual  
## Pred No Yes  
## 0 109 20  
## 1 14 94  
##   
## $model5  
## Actual  
## Pred No Yes  
## 0 108 21  
## 1 15 93  
##   
## $model6  
## Actual  
## Pred No Yes  
## 0 107 19  
## 1 16 95  
##   
## $model7  
## Actual  
## Pred No Yes  
## 0 108 21  
## 1 15 93

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

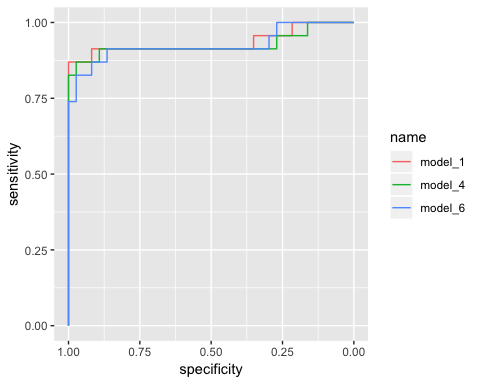
**Test Models**

For the test data, we chose the best three models - model 1, model 4, and model 6. We chose model 1 because it offers the second-best accuracy at 80.06, the highest AUC, and low false negative outcomes. Model 4 was chosen because it showed the most consistent performance across the measureable variables. Model 6 was chosen because it offered the lowest AIC, higher accuracy than other models, an AUC that was one of the highest, and all of the variables of the model were statistically significant (if our alpha was .1). Also, model 6 provided low false negative outcomes.

pred\_test1 <- predict(model, test, type = 'response')  
prediction\_1 <- ifelse(pred\_test1 > 0.5, 1, 0)  
tab\_1 <- table(Predicted = prediction\_1, Actual = test$AHD)  
roccurve\_1 <- roc(test$AHD ~ pred\_test1)  
### Model 4  
pred\_test4 <- predict(model4, test, type = 'response')  
prediction\_4 <- ifelse(pred\_test4 > 0.5, 1, 0)  
tab\_4 <- table(Predicted = prediction\_4, Actual = test$AHD)  
roccurve\_4 <- roc(test$AHD ~ pred\_test4)  
#### Model 6  
pred\_test6 <- predict(model6, test, type = 'response')  
prediction\_6 <- ifelse(pred\_test6 > 0.5, 1, 0)  
tab\_6 <- table(Predicted = prediction\_6, Actual = test$AHD)  
roccurve\_6 <- roc(test$AHD ~ pred\_test6)

**ROC for Test Data**

ggroc(list(model\_1 = roccurve\_1, model\_4 = roccurve\_4, model\_6 = roccurve\_6))



**Test Data Accuracy**

test\_info\_list <- list(Model1 = list(Accuracy = acc(tab\_1), AUC = auc(roccurve\_1)),  
 Model4 = list(Accuracy = acc(tab\_4), AUC = auc(roccurve\_4)),  
 Model6 = list(Accuracy = acc(tab\_6), AUC = auc(roccurve\_6)))  
test\_info1 <- do.call(rbind, test\_info\_list)  
test\_info2 <- as.data.frame(test\_info1)  
test\_info <- rownames\_to\_column(test\_info2, "Models")  
test\_info

##

Models Accuracy AUC  
## 1 Model1 93.33333 0.9341951  
## 2 Model4 90 0.9259694  
## 3 Model6 90 0.9259694

**Test Data Tables**

tables\_test <- list(model1 = tab\_1,  
 model4 = tab\_4,  
 model6 = tab\_6)  
   
tables\_test

## $model1  
## Actual  
## Predicted No Yes  
## 0 36 3  
## 1 1 20  
##   
## $model4  
## Actual  
## Predicted No Yes  
## 0 34 3  
## 1 3 20  
##   
## $model6  
## Actual  
## Predicted No Yes  
## 0 35 4  
## 1 2 19

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Best Logistic Regression Model**

Our group chose Model 6 as the best model. Accuracy and AUC ranked in the top two or three for most of the training data set. Of any model trained with the training data set, Model 6 had the lowest AIC. In the training data set, its false negative outcomes ranked one of the lowest, but in the test data set, it had an unexpected high number of false negative outcomes compared to the other models we tested. That could mean that our model is over-fitted for the training data set or that we need a larger test data set. Model 6 is the only model that had all variables being statistically significant if alpha was 0.1.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SUPPORT VECTOR MACHINES**

The same train and test data for logistic regression were used for SVM. In the first model, we used 3 different kernels – linear, sigmoid, and polynomial. After seeing the number of support vectors from the three kernels for SVM1, we decided to just use sigmoid and linear kernels for the rest of the models. The polynomial kernel provides a high number of support vectors and a low accuracy.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SVM1**

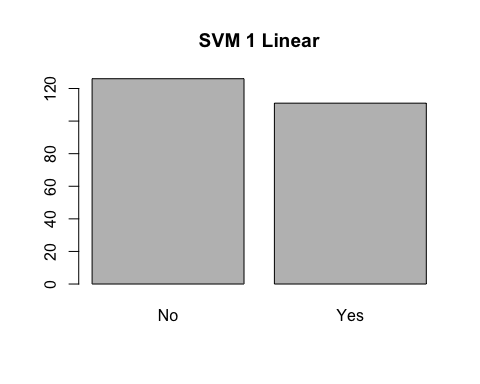
library(e1071)  
svm1\_lin <- svm(AHD ~ ., data = train, kernel = "linear")  
summary(svm1\_lin)

##   
## Call:  
## svm(formula = AHD ~ ., data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.04761905   
##   
## Number of Support Vectors: 100  
##   
## ( 52 48 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred1\_lin <- predict(svm1\_lin, train, type = "class")  
tab\_lin <- table(Predicited = pred1\_lin, Actual = train$AHD)  
tab\_lin

## Actual  
## Predicited No Yes  
## No 107 19  
## Yes 16 95

plot(pred1\_lin, main = "SVM 1 Linear")



**SVM 1 Sigmoid**

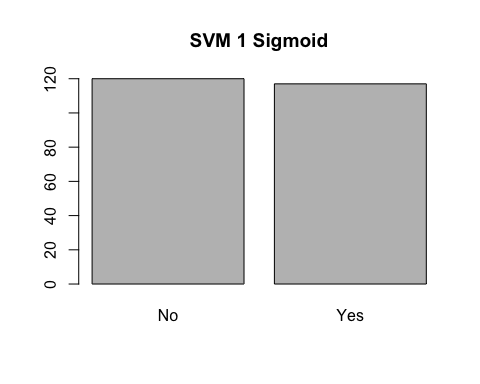
######### Sigmoid ##########  
svm1\_sig <- svm(AHD ~ ., data = train, kernel = "sigmoid")  
summary(svm1\_sig)

##   
## Call:  
## svm(formula = AHD ~ ., data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.04761905   
## coef.0: 0   
##   
## Number of Support Vectors: 133  
##   
## ( 66 67 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred1\_sig <- predict(svm1\_sig, train, type = "class")  
tab\_sig <- table(Predicited = pred1\_sig, Actual = train$AHD)  
tab\_sig

## Actual  
## Predicited No Yes  
## No 98 22  
## Yes 25 92

plot(pred1\_sig, main = "SVM 1 Sigmoid")



**SVM 1 Polynomial**

svm1\_poly <- svm(AHD ~ ., data = train, kernel = "polynomial")  
summary(svm1\_poly)

##   
## Call:  
## svm(formula = AHD ~ ., data = train, kernel = "polynomial")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: polynomial   
## cost: 1   
## degree: 3   
## gamma: 0.04761905   
## coef.0: 0   
##   
## Number of Support Vectors: 212  
##   
## ( 105 107 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred1\_poly <- predict(svm1\_poly, train, type = "class")  
tab\_poly <- table(Predicited = pred1\_poly, Actual = train$AHD)  
tab\_poly

## Actual  
## Predicited No Yes  
## No 120 58  
## Yes 3 56

acc(tab\_poly)

## [1] 74.2616

plot(pred1\_poly, main = "SVM 1 Polynomial")

**SVM 2**

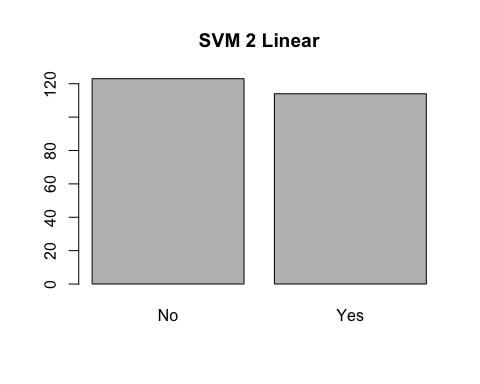
svm2 <- svm(AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca, data = train, kernel = "linear")  
summary(svm2)

##   
## Call:  
## svm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak +   
## Slope + Ca, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.07692308   
##   
## Number of Support Vectors: 103  
##   
## ( 52 51 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred2 <- predict(svm2, train, type = "class")  
plot(pred2, main = "SVM 2 Linear")

tab2 <- table(Predicited = pred2, Actual = train$AHD)  
tab2

## Actual  
## Predicited No Yes  
## No 104 19  
## Yes 19 95



**SVM 2 Sigmoid**

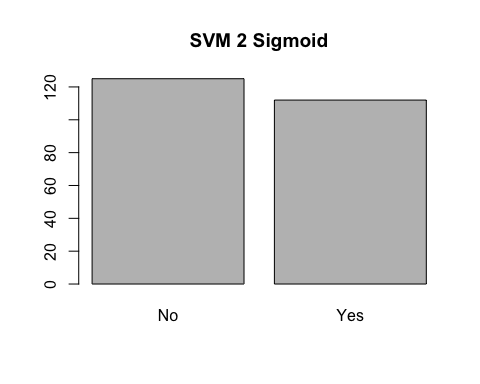
svm2\_sig <- svm(AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca, data = train, kernel = "sigmoid")  
summary(svm2\_sig)

##   
## Call:  
## svm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak +   
## Slope + Ca, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.07692308   
## coef.0: 0   
##   
## Number of Support Vectors: 140  
##   
## ( 70 70 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred2\_sig <- predict(svm2\_sig, train, type = "class")  
plot(pred2\_sig, main = "SVM 2 Sigmoid")

tab2\_sig <- table(Predicited = pred2\_sig, Actual = train$AHD)  
tab2\_sig

## Actual  
## Predicited No Yes  
## No 101 24  
## Yes 22 90

  
**SVM 3**

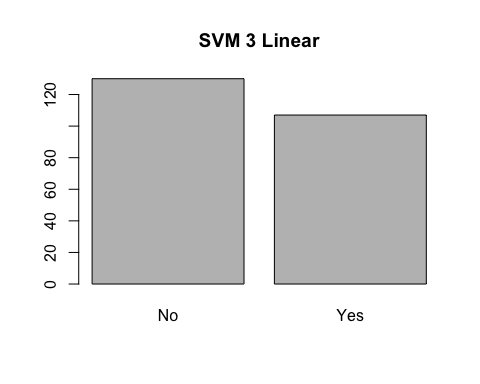
svm3 <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal, data = train, kernel = "linear")  
summary(svm3)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.0625   
##   
## Number of Support Vectors: 99  
##   
## ( 49 50 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred3 <- predict(svm3, train, type = "class")  
plot(pred3, main = "SVM 3 Linear")

tab3 <- table(Predicited = pred3, Actual = train$AHD)  
tab3

## Actual  
## Predicited No Yes  
## No 108 22  
## Yes 15 92



**SVM 3 Sigmoid**

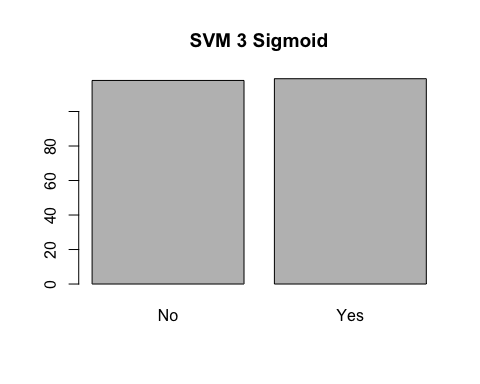
svm3\_sig <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal, data = train, kernel = "sigmoid")  
summary(svm3\_sig)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.0625   
## coef.0: 0   
##   
## Number of Support Vectors: 132  
##   
## ( 67 65 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred3\_sig <- predict(svm3\_sig, train, type = "class")  
plot(pred3\_sig, main = "SVM 3 Sigmoid")

tab3\_sig <- table(Predicited = pred3\_sig, Actual = train$AHD)  
tab3\_sig

## Actual  
## Predicited No Yes  
## No 99 19  
## Yes 24 95



**SVM 4**

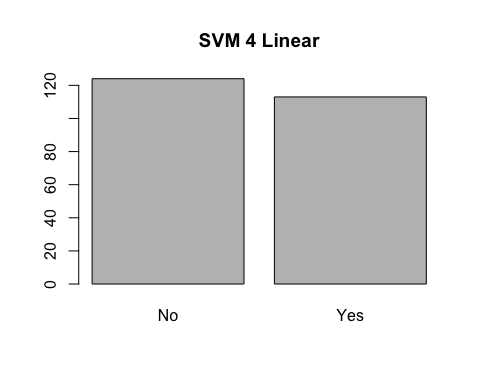
svm4 <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Chol, data = train, kernel = "linear")  
summary(svm4)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + Chol, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.05882353   
##   
## Number of Support Vectors: 98  
##   
## ( 50 48 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred4 <- predict(svm4, train, type = "class")  
plot(pred4, main = "SVM 4 Linear")

tab4 <- table(Predicited = pred4, Actual = train$AHD)  
tab4

## Actual  
## Predicited No Yes  
## No 106 18  
## Yes 17 96



**SVM 4 Sigmoid**

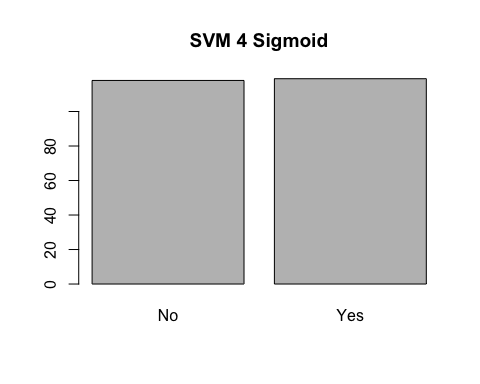
svm4\_sig <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Chol, data = train, kernel = "sigmoid")  
summary(svm4\_sig)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + Chol, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.05882353   
## coef.0: 0   
##   
## Number of Support Vectors: 133  
##   
## ( 66 67 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred4\_sig <- predict(svm4\_sig, train, type = "class")  
plot(pred4\_sig, main = "SVM 4 Sigmoid")

tab4\_sig <- table(Predicited = pred4\_sig, Actual = train$AHD)  
tab4\_sig

## Actual  
## Predicited No Yes  
## No 99 19  
## Yes 24 95



**SVM 5**

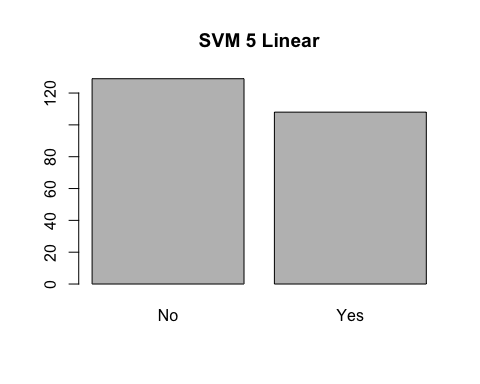
svm5 <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + ExAng, data = train, kernel = "linear")  
summary(svm5)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + ExAng, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.05882353   
##   
## Number of Support Vectors: 97  
##   
## ( 49 48 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred5 <- predict(svm5, train, type = "class")  
plot(pred5, main = "SVM 5 Linear")

tab5 <- table(Predicited = pred5, Actual = train$AHD)  
tab5

## Actual  
## Predicited No Yes  
## No 108 21  
## Yes 15 93



**SVM 5 Sigmoid**

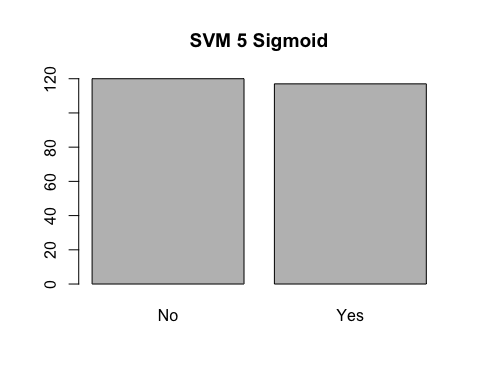
svm5\_sig <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + ExAng, data = train, kernel = "sigmoid")  
summary(svm5\_sig)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + ExAng, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.05882353   
## coef.0: 0   
##   
## Number of Support Vectors: 130  
##   
## ( 65 65 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred5\_sig <- predict(svm5\_sig, train, type = "class")  
plot(pred5\_sig, main = "SVM 5 Sigmoid")

tab5\_sig <- table(Predicited = pred5\_sig, Actual = train$AHD)  
tab5\_sig

## Actual  
## Predicited No Yes  
## No 99 21  
## Yes 24 93



**SVM 6**

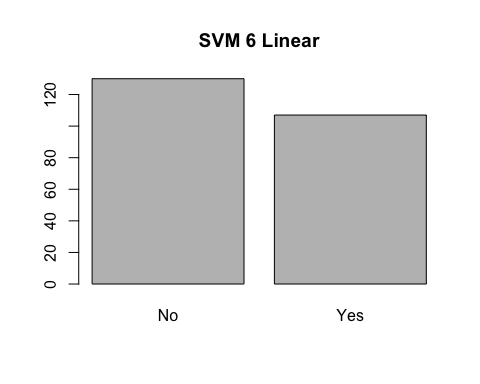
svm6 <- svm(AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + ExAng + Chol, data = train, kernel = "linear")  
summary(svm6)

##   
## Call:  
## svm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak +   
## Slope + Ca + Thal + ExAng + Chol, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.05882353   
##   
## Number of Support Vectors: 99  
##   
## ( 48 51 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred6 <- predict(svm6, train, type = "class")  
plot(pred6, main = "SVM 6 Linear")

tab6 <- table(Predicited = pred6, Actual = train$AHD)  
tab6

## Actual  
## Predicited No Yes  
## No 109 21  
## Yes 14 93



**SVM 6 Sigmoid**

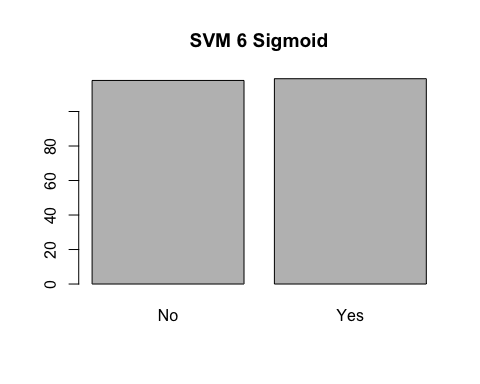
svm6\_sig <- svm(AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + ExAng + Chol, data = train, kernel = "sigmoid")  
summary(svm6\_sig)

##   
## Call:  
## svm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak +   
## Slope + Ca + Thal + ExAng + Chol, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.05882353   
## coef.0: 0   
##   
## Number of Support Vectors: 130  
##   
## ( 65 65 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred6\_sig <- predict(svm6\_sig, train, type = "class")  
plot(pred6\_sig, main = "SVM 6 Sigmoid")

tab6\_sig <- table(Predicited = pred6\_sig, Actual = train$AHD)  
tab6\_sig

## Actual  
## Predicited No Yes  
## No 97 21  
## Yes 26 93



**SVM 7**

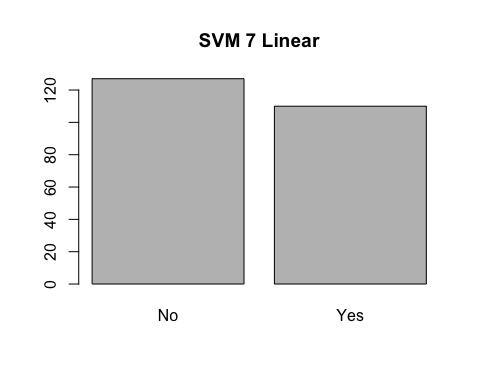
svm7 <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + RestECG, data = train, kernel = "linear")  
summary(svm7)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + RestECG, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.05555556   
##   
## Number of Support Vectors: 99  
##   
## ( 49 50 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred7 <- predict(svm7, train, type = "class")  
plot(pred7, main = "SVM 7 Linear")

tab7 <- table(Predicited = pred7, Actual = train$AHD)  
tab7

## Actual  
## Predicited No Yes  
## No 108 19  
## Yes 15 95



**SVM 7 Sigmoid**

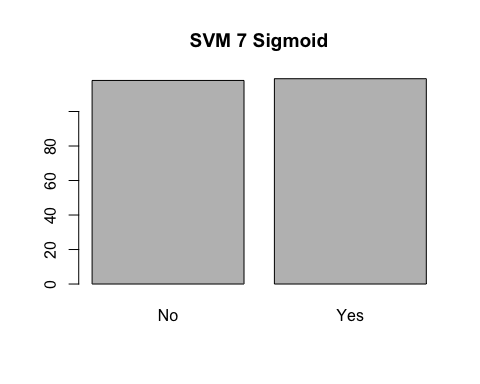
svm7\_sig <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + RestECG, data = train, kernel = "sigmoid")  
summary(svm7\_sig)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + RestECG, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.05555556   
## coef.0: 0   
##   
## Number of Support Vectors: 137  
##   
## ( 69 68 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred7\_sig <- predict(svm7\_sig, train, type = "class")  
plot(pred7\_sig, main = "SVM 7 Sigmoid")

tab7\_sig <- table(Predicited = pred7\_sig, Actual = train$AHD)  
tab7\_sig

## Actual  
## Predicited No Yes  
## No 99 19  
## Yes 24 95



**SVM 8**

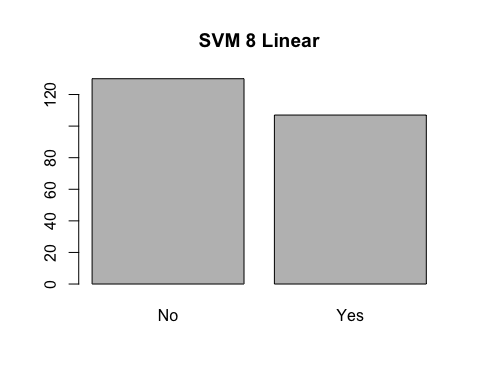
svm8 <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Fbs, data = train, kernel = "linear")  
summary(svm8)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + Fbs, data = train, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.05882353   
##   
## Number of Support Vectors: 98  
##   
## ( 49 49 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred8 <- predict(svm8, train, type = "class")  
plot(pred8,main = "SVM 8 Linear")

tab8 <- table(Predicited = pred8, Actual = train$AHD)  
tab8

## Actual  
## Predicited No Yes  
## No 108 22  
## Yes 15 92



**SVM 8 Sigmoid**

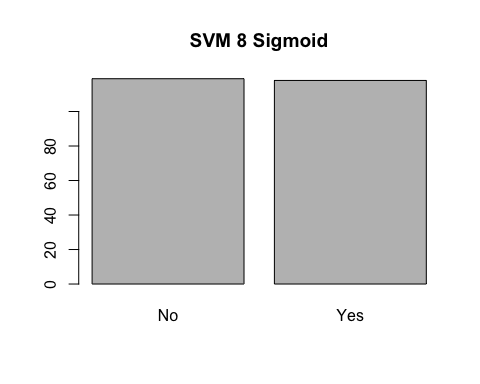
svm8\_sig <- svm(AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Fbs, data = train, kernel = "sigmoid")  
summary(svm8\_sig)

##   
## Call:  
## svm(formula = AHD ~ Age + Sex + ChestPain + RestBP + MaxHR +   
## Oldpeak + Slope + Ca + Thal + Fbs, data = train, kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.05882353   
## coef.0: 0   
##   
## Number of Support Vectors: 134  
##   
## ( 66 68 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred8\_sig <- predict(svm8\_sig, train, type = "class")  
plot(pred8\_sig,main = "SVM 8 Sigmoid")

tab8\_sig <- table(Predicited = pred8\_sig, Actual = train$AHD)  
tab8\_sig

## Actual  
## Predicited No Yes  
## No 100 19  
## Yes 23 95



**SVM 9**

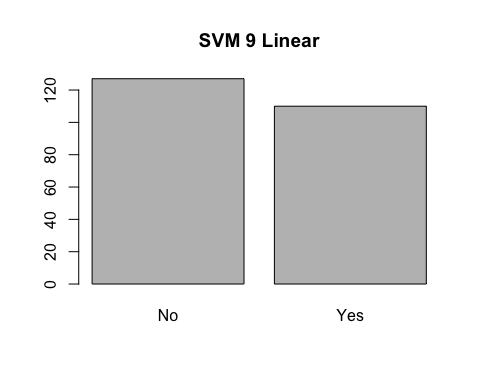
svm9 <- svm(AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Fbs + RestECG + Chol + ExAng, data = train, kernel = "linear")  
summary(svm9)

##   
## Call:  
## svm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak +   
## Slope + Ca + Thal + Fbs + RestECG + Chol + ExAng, data = train,   
## kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.05   
##   
## Number of Support Vectors: 98  
##   
## ( 49 49 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred9 <- predict(svm9, train, type = "class")  
plot(pred9, main = "SVM 9 Linear")

tab9 <- table(Predicited = pred9, Actual = train$AHD)  
tab9

## Actual  
## Predicited No Yes  
## No 108 19  
## Yes 15 95



**SVM 9 Sigmoid**

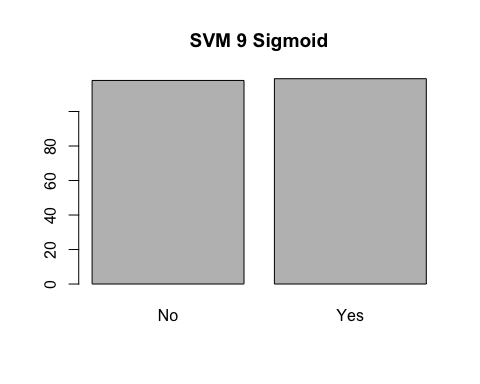
svm9\_sig <- svm(AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Fbs + RestECG + Chol + ExAng, data = train, kernel = "sigmoid")  
summary(svm9\_sig)

##   
## Call:  
## svm(formula = AHD ~ Sex + ChestPain + RestBP + MaxHR + Oldpeak +   
## Slope + Ca + Thal + Fbs + RestECG + Chol + ExAng, data = train,   
## kernel = "sigmoid")  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: sigmoid   
## cost: 1   
## gamma: 0.05   
## coef.0: 0   
##   
## Number of Support Vectors: 133  
##   
## ( 67 66 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## No Yes

pred9\_sig <- predict(svm9\_sig, train, type = "class")  
plot(pred9\_sig, main = "SVM 9 Sigmoid")

tab9\_sig <- table(Predicited = pred9\_sig, Actual = train$AHD)  
tab9\_sig

## Actual  
## Predicited No Yes  
## No 98 20  
## Yes 25 94



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SVM Train Accuracy**

For SVM, we looked for models with a high accuracy and a low number of support vectors. We noticed that all of the sigmoid models don’t go below 100 support vectors. Also, the accuracy of the sigmoid models is a little less than the linear models. From the graphs of the model’s predictions, the sigmoid models picked almost a 50/50 split of yes to no. That’s not a good result because from the overall yes to no graph, there are more no outcomes. The linear models predict more no outcomes, which goes along with the data set.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

##For train   
svm\_list <- list(svm1 = list(svm1\_lin$tot.nSV,acc(tab\_lin)),  
 svm1\_sig = list(svm1\_sig$tot.nSV,acc(tab\_sig)),  
 svm2 = list(svm2$tot.nSV,acc(tab2)),  
 svm2\_sig = list(svm2\_sig$tot.nSV,acc(tab2\_sig)),  
 svm3 = list(svm3$tot.nSV,acc(tab3)),  
 svm3\_sig = list(svm3\_sig$tot.nSV,acc(tab3\_sig)),  
 svm4 = list(svm4$tot.nSV ,acc(tab4)),  
 svm4\_sig = list(svm4\_sig$tot.nSV ,acc(tab4\_sig)),  
 svm5 = list(svm5$tot.nSV ,acc(tab5)),  
 svm5\_sig = list(svm5\_sig$tot.nSV ,acc(tab5\_sig)),  
 svm6 = list(svm6$tot.nSV ,acc(tab6)),  
 svm6\_sig = list(svm6\_sig$tot.nSV ,acc(tab6\_sig)),  
 svm7 = list(svm7$tot.nSV ,acc(tab7)),  
 svm7\_sig = list(svm7\_sig$tot.nSV ,acc(tab7\_sig)),  
 svm8 = list(svm8$tot.nSV ,acc(tab8)),  
 svm8\_sig = list(svm8\_sig$tot.nSV ,acc(tab8\_sig)),  
 svm9 = list(svm9$tot.nSV ,acc(tab9)),  
 svm9\_sig = list(svm9\_sig$tot.nSV ,acc(tab9\_sig)))  
  
svm\_info1 <- do.call(rbind, svm\_list)  
svm\_info2 <- as.data.frame(svm\_info1 )  
svm\_info <- rownames\_to\_column(svm\_info2, "Models")  
names(svm\_info)[2] <- "Total Support Vectors"  
names(svm\_info)[3] <- "Accuracy"  
svm\_info$Accuracy <- round(as.numeric(svm\_info$Accuracy), 2)  
svm\_info

Again, we created another table for a better view.

|  |  |  |
| --- | --- | --- |
| **Models** | **Support Vectors** | **Accuracy** |
| svm1 | 100 | 85.23 |
| svm1\_sig | 133 | 80.17 |
| svm2 | 103 | 83.97 |
| svm2\_sig | 140 | 80.59 |
| svm3 | 99 | 84.39 |
| svm3\_sig | 132 | 81.86 |
| svm4 | 98 | 85.23 |
| svm4\_sig | 133 | 81.86 |
| svm5 | 97 | 84.81 |
| svm5\_sig | 130 | 81.01 |
| svm6 | 99 | 85.23 |
| svm6\_sig | 130 | 80.17 |
| svm7 | 99 | 85.65 |
| svm7\_sig | 137 | 81.86 |
| svm8 | 98 | 84.39 |
| svm8\_sig | 134 | 82.28 |
| svm9 | 98 | 85.65 |
| svm9\_sig | 133 | 81.01 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SVM Train Tables**

In the tables of predicted values vs. the actual values, we were looking for models that had low false negative outcomes. SVMs 1, 2, 7, and 9 all had 19 false negatives, while SVM 4 had 18 false negatives.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

svm\_tables <- list(svm1 = tab\_lin,  
 svm2 = tab2,  
 svm3 = tab3,  
 svm4 = tab4,  
 svm5 = tab5,  
 svm6 = tab6,  
 svm7 = tab7,  
 svm8 = tab8,  
 svm9 = tab9)  
  
svm\_tables

##

$svm1  
## Actual  
## Predicited No Yes  
## No 107 19  
## Yes 16 95  
##   
## $svm2  
## Actual  
## Predicited No Yes  
## No 104 19  
## Yes 19 95  
##   
## $svm3  
## Actual  
## Predicited No Yes  
## No 108 22  
## Yes 15 92  
##   
## $svm4  
## Actual  
## Predicited No Yes  
## No 106 18  
## Yes 17 96  
##   
## $svm5  
## Actual  
## Predicited No Yes  
## No 108 21  
## Yes 15 93  
##   
##

$svm6  
## Actual  
## Predicited No Yes  
## No 109 21  
## Yes 14 93  
##   
## $svm7  
## Actual  
## Predicited No Yes  
## No 108 19  
## Yes 15 95  
##   
## $svm8  
## Actual  
## Predicited No Yes  
## No 108 22  
## Yes 15 92  
##   
## $svm9  
## Actual  
## Predicited No Yes  
## No 108 19  
## Yes 15 95

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Test SVM**

As we did in logistic regression, we chose the best three SVM models. We picked SVM 4 because of the low number of support vectors and it ranking secondhighest in accuracy. It also had the fewest number of false negative outcomes, 18. We decided that SVM 7 offered a good choice to use with the test data set. SVM 7 had a high accuracy and the lowest number of support vectors. SVM 9 was another model with a high accuracy of 85.86 - tied with SVM 7 - but SVM 9 had one less support vector than SVM 7. Both models also have 19 false negative outcomes, which is second best after SVM4.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pred\_4 <- predict(svm4, test, type = "class")  
plot(pred\_4,main = "SVM 4 (Test Data)")

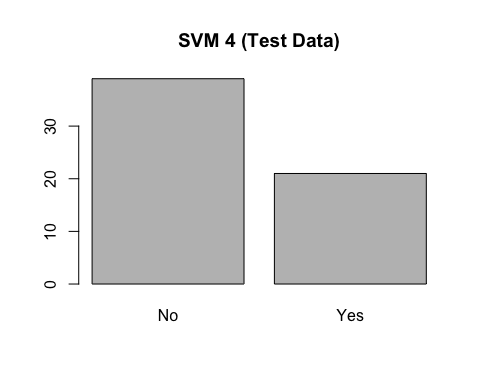
tab\_4 <- table(Predicited = pred\_4, Actual = test$AHD)  
tab\_4

##

Actual  
## Predicited No Yes  
## No 36 3  
## Yes 1 20

acc(tab\_4)

## [1] 93.33333



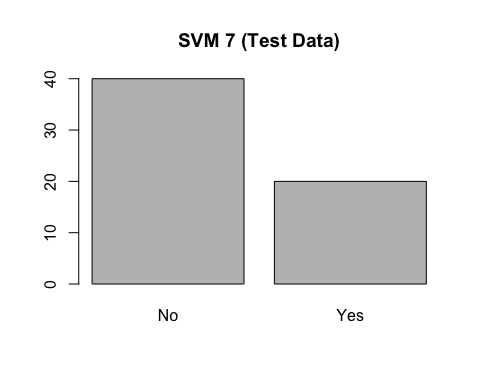
#### svm 7  
pred\_7 <- predict(svm7, test, type = "class")  
plot(pred\_7, main = "SVM 7 (Test Data)")

tab\_7 <- table(Predicited = pred\_7, Actual = test$AHD)  
tab\_7

## Actual  
## Predicited No Yes  
## No 36 4  
## Yes 1 19

acc(tab\_7)

## [1] 91.66667



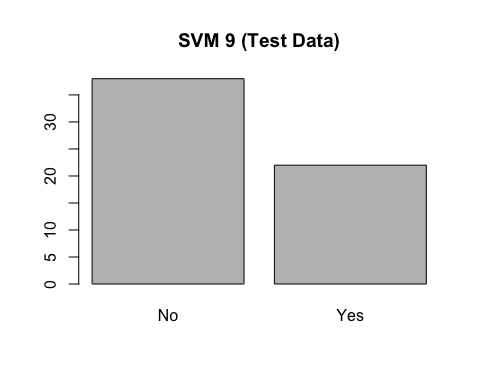
pred\_9 <- predict(svm9, test, type = "class")  
plot(pred\_9, main = "SVM 9 (Test Data)")

tab\_9 <- table(Predicited = pred\_9, Actual = test$AHD)  
tab\_9

## Actual  
## Predicited No Yes  
## No 35 3  
## Yes 2 20

acc(tab\_9)

## [1] 91.66667



**Test SVM Accuracy and Table**

list(svm4 = list(Total\_Support\_Vectors = svm4$tot.nSV, Table = tab\_4 ,Accuracy = acc(tab\_4)),  
 svm7 = list(Total\_Support\_Vectors = svm7$tot.nSV,Table = tab\_7 ,Accuracy = acc(tab\_7)),  
 svm9 = list(Total\_Support\_Vectors = svm9$tot.nSV,Table = tab\_9 ,Accuracy = acc(tab\_9)))

## $svm4  
## $svm4$Total\_Support\_Vectors  
## [1] 98  
##   
## $svm4$Table  
## Actual  
## Predicited No Yes  
## No 36 3  
## Yes 1 20  
##   
## $svm4$Accuracy  
## [1] 93.33333  
##   
##

## $svm7  
## $svm7$Total\_Support\_Vectors  
## [1] 99  
##   
## $svm7$Table  
## Actual  
## Predicited No Yes  
## No 36 4  
## Yes 1 19  
##   
## $svm7$Accuracy  
## [1] 91.66667  
##   
## $svm9  
## $svm9$Total\_Support\_Vectors  
## [1] 98  
##   
## $svm9$Table  
## Actual  
## Predicited No Yes  
## No 35 3  
## Yes 2 20  
##   
## $svm9$Accuracy  
## [1] 91.66667

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Best SVM Model**

We chose our best model to be SVM 4 after considering how the models function with the train and test data sets. In the test data set, it had the highest accuracy and only three false negative outcomes. SVM 4 provided the lowest number of support vectors of 98 and the second highest accuracy of 85.23 (first highest was 85.65). In the test data set, SVM 4 had the highest accuracy of 93.33 (second highest was 91.66).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Best Model between SVM & Logistic Model**

For Logistic Regression, the best model we found was Model 6 (AHD ~ Sex + ChestPain + RestBP + ExAng + Oldpeak + Slope + Ca + Thal). The train data set gave an accuracy of 85.23 and an AIC of 184.87, and the test data set gave an accuracy of 90 and an AUC of 0.926). For SVM, the best model was SVM 4 (AHD ~ Age + Sex + ChestPain + RestBP + MaxHR + Oldpeak + Slope + Ca + Thal + Chol, data = train, kernel = "linear"). The train data set for SVM4 offered 98 support vectors and an 85.23 accuracy. When it was time to pick between Logistic Regression and SVM, we chose SVM 4. We only had three metrics to compare the Logistic Regression and SVM models – train data set accuracy, test data set accuracy, and false negative outcomes for both the data sets. Both the models had the same accuracy with the training data set of 85.23. For the test data set, SVM 4 provided the better accuracy. When comparing, we observed that the SVM 4 model experienced fewer false negative outcomes than model 6 in the train data set. SVM 4 also offered fewer false negative outcomes than model 6 with the test data set. We would need a larger data set to really test their differences. In this project, we concluded that SVM 4 was the best model.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CONCLUSION**

The goal of this project was to analyze the heart disease data to make predictions. After using logistic regression and support vector machines methodologies, we found our best model. The statistical details mentioned earlier made us conclude that SVM 4 was the best model to use for this data set. We would use this model to study and make predictions for patients or hospitals if we were to get more observations. The SVM 4 model gave us an accuracy of 93.33 for the test data set. We described the reasons for choosing the models we used to get the best accuracy. We also described the methodologies we used for efficiency; since it’s best to be efficient when dealing with data, we chose methodologies accordingly.