Università degli Studi di Padova

Dipartimento Matematica

Master's Degree in Data Science

Academic Year 2023/2024

Statistical Learning

Heart Disease Prediction

Lorenzo Baietti	Francesco Carlesso	Matteo Mazzini
ID: 2130676	ID: 2125806	ID: 2107797

Contents

1. Introduction

Project and Dataset Description

Heart Diseases are the leading cause of death in the world, they can be caused by different kind of factors such as genetics, lifestyle, medical conditions, environmental factors etc. Early diagnosis is crucial for carrying out a successful treatment, therefore we decided to perform an analysis to understand which are the most influential biometrics that define the condition, focusing on a binary classification task which uses parameters that can be obtained simply by performing clinical tests. The dataset we are going to analyze and use for our task is the *Heart Failure Prediction Dataset* which was created by combining different datasets, already available in the UCI Machine Learning Repository, over common features. This dataset contains 918 patient records described by 11 variables plus a binary target for the diagnosis. Out of all the patients, 508 have a positive diagnosis.

Variables Overview

Variable	Description
Age	Age of the patient [Years]
Sex	Sex of the patient [M: Male; F: Female]
ChestPainType	Chest Pain Type [TA: Typical Angina; ATA: Atypical Angina; NAP: Non-Anginal Pain; ASY: Asymptomatic]
RestingBP	Resting Blood Pressure [mmHg]
Cholesterol	Serum Cholesterol [mm/dL]
FastingBS	Fasting Blood Sugar [1: if FastingBS > 120 mg/dL; 0: otherwise]
RestingECG	Resting Electrocardiogram Results [Normal: normal; ST: having ST-T wave abnormality; LVH: showing probable or definite left ventricular hypertrophy]
MaxHR	Maximum Heart Rate Achieved [Range(60-120)]
ExerciseAngina	Exercise-induced Angina [Y: Yes; N: No]
Oldpeak	ST segment depression compared to resting [Numerical value]
ST_Slope	Slope of the peak exercise ST segment [Up: upsloping; Flat: flat; Down: downsloping]
HeartDisease	Response [1: if the patient is diagnosed with Heart Disease; 0: otherwise]

Terminology

Angina is a type of chest pain caused by reduced blood flow to the heart. Typically, it is described as squeezing, pressure, heaviness, tightness in the chest. Angina is usually a symptom of an underlying heart problem.

ST segment is an electrically neutral part of the electrocardiogram (ECG) that represents the interval between ventricular depolarization (QRS complex) and repolarization (T wave). It is defined by the segment connecting the end of the S wave and the beginning of the T wave.

Oldpeak is a measure of the ST segment depression induced by exercise, compared to the ST segment observed with the resting ECG results.

2. Data Preprocessing

First, we import all the necessary packages to manipulate the data, then we load the dataset and check its structure.

```r
library(car)
library(class)
library(corrplot)
library(glmnet)
library(MASS)
library(pROC)
```

```r
data <- read.csv('heart_data.csv', header = TRUE, sep = ',')
str(data)
```

```
## 'data.frame':    918 obs. of  12 variables:
##  $ Age           : int  40 49 37 48 54 39 45 54 37 48 ...
##  $ Sex           : chr  "M" "F" "M" "F" ...
##  $ ChestPainType : chr  "ATA" "NAP" "ATA" "ASY" ...
##  $ RestingBP     : int  140 160 130 138 150 120 130 110 140 120 ...
##  $ Cholesterol   : int  289 180 283 214 195 339 237 208 207 284 ...
##  $ FastingBS     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ RestingECG    : chr  "Normal" "Normal" "ST" "Normal" ...
##  $ MaxHR         : int  172 156 98 108 122 170 170 142 130 120 ...
##  $ ExerciseAngina: chr  "N" "N" "N" "Y" ...
##  $ Oldpeak       : num  0 1 0 1.5 0 0 0 0 1.5 0 ...
##  $ ST_Slope      : chr  "Up" "Flat" "Up" "Flat" ...
##  $ HeartDisease  : int  0 1 0 1 0 0 0 0 1 0 ...
```

We notice that our categorical variables are stored as either `chr` or `int` type, and we need to convert them to `Factor` type to perform our analysis.

```r
data$Sex <- as.factor(data$Sex)
data$ChestPainType <- as.factor(data$ChestPainType)
data$RestingECG <- as.factor(data$RestingECG)
data$ExerciseAngina <- as.factor(data$ExerciseAngina)
data$ST_Slope <- as.factor(data$ST_Slope)
data$FastingBS <- as.factor(data$FastingBS)
data$HeartDisease <- as.factor(data$HeartDisease)
```

```r
any(is.na(data))
```

```
## [1] FALSE
```

```r
summary(data)
```

```
##       Age         Sex     ChestPainType   RestingBP      Cholesterol
##  Min.   :28.00   F:193   ASY:496       Min.   :  0.0   Min.   :  0.0
##  1st Qu.:47.00   M:725   ATA:173       1st Qu.:120.0   1st Qu.:173.2
##  Median :54.00           NAP:203       Median :130.0   Median :223.0
##  Mean   :53.51           TA : 46       Mean   :132.4   Mean   :198.8
##  3rd Qu.:60.00                         3rd Qu.:140.0   3rd Qu.:267.0
##  Max.   :77.00                         Max.   :200.0   Max.   :603.0
##  FastingBS  RestingECG       MaxHR       ExerciseAngina    Oldpeak
##  0:704      LVH   :188   Min.   : 60.0   N:547          Min.   :-2.6000
##  1:214      Normal:552   1st Qu.:120.0   Y:371          1st Qu.: 0.0000
##             ST    :178   Median :138.0                  Median : 0.6000
##                          Mean   :136.8                  Mean   : 0.8874
##                          3rd Qu.:156.0                  3rd Qu.: 1.5000
##                         Max.   :202.0                  Max.   : 6.2000
##  ST_Slope   HeartDisease
##  Down: 63   0:410
##  Flat:460   1:508
##  Up  :395
```

It seems like there are no missing values in our dataset, however, from the summary we can see that the variables `RestingBP` and `Cholesterol` have `0.0` as minimum value. Given that, unless the patient is dead, blood pressure has to be greater than zero, we investigate further using `MaxHR` to establish if the measurement was taken from an alive patient.

```r
sum(data$RestingBP == 0 & data$MaxHR > 0)
```

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

```r
sum(data$Cholesterol == 0)
```

```
## [1] 172
```

As we can see, the `0.0` value for blood pressure has been assigned to an alive patient. Furthermore, 0 cholesterol is biologically impossible to observe, even in a deceased individual, thus we conjecture that it was simply how missing values were represented. Considering this, we decide to replace these NA values with the median of the respective column, given that this central tendency measure is less sensitive to extreme observations.

```r
data$Cholesterol[data$Cholesterol==0] <- NA
data$RestingBP[data$RestingBP==0] <- NA
data$Cholesterol[is.na(data$Cholesterol)] <- median(data$Cholesterol, na.rm = TRUE)
data$RestingBP[is.na(data$RestingBP)] <- median(data$RestingBP, na.rm = TRUE)
summary(data)
```

```
##       Age        Sex     ChestPainType   RestingBP      Cholesterol
##  Min.   :28.00   F:193   ASY:496       Min.   : 80.0   Min.   : 85.0
##  1st Qu.:47.00   M:725   ATA:173       1st Qu.:120.0   1st Qu.:214.0
##  Median :54.00           NAP:203       Median :130.0   Median :237.0
##  Mean   :53.51           TA : 46       Mean   :132.5   Mean   :243.2
##  3rd Qu.:60.00                         3rd Qu.:140.0   3rd Qu.:267.0
##  Max.   :77.00                         Max.   :200.0   Max.   :603.0
##  FastingBS  RestingECG      MaxHR       ExerciseAngina    Oldpeak
##  0:704     LVH   :188   Min.   : 60.0   N:547          Min.   :-2.6000
##  1:214     Normal:552   1st Qu.:120.0   Y:371          1st Qu.: 0.0000
##            ST    :178   Median :138.0                  Median : 0.6000
##                         Mean   :136.8                  Mean   : 0.8874
##                         3rd Qu.:156.0                  3rd Qu.: 1.5000
##                         Max.   :202.0                  Max.   : 6.2000
##  ST_Slope   HeartDisease
##  Down: 63   0:410
##  Flat:460   1:508
##  Up  :395
```

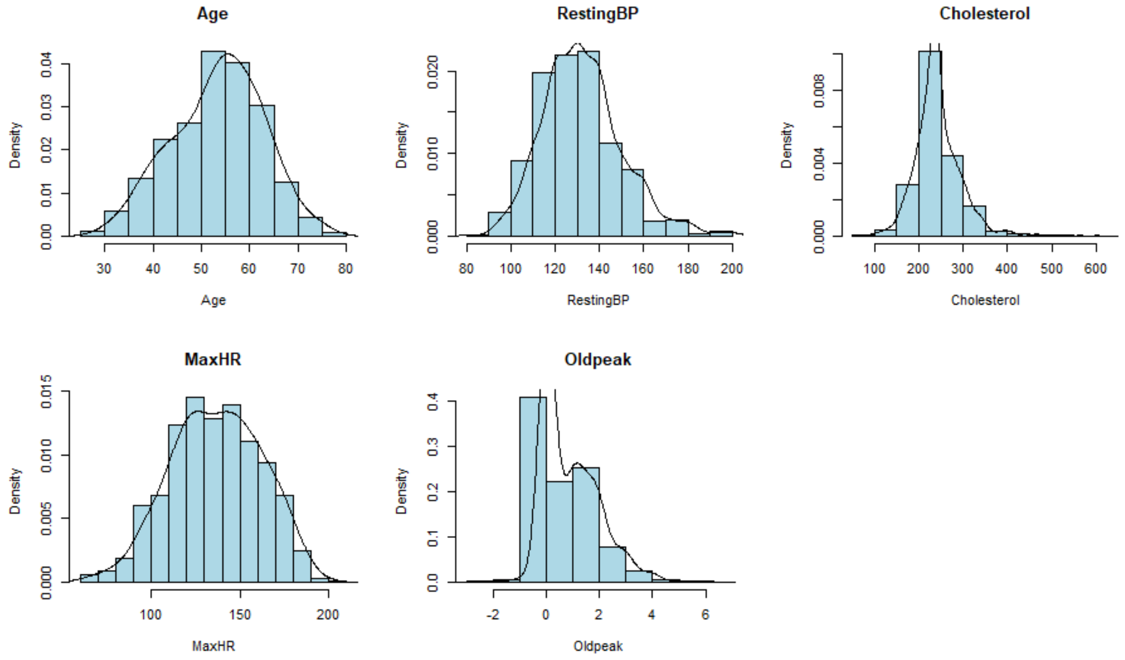
Now the data is ready to be analyzed and processed.

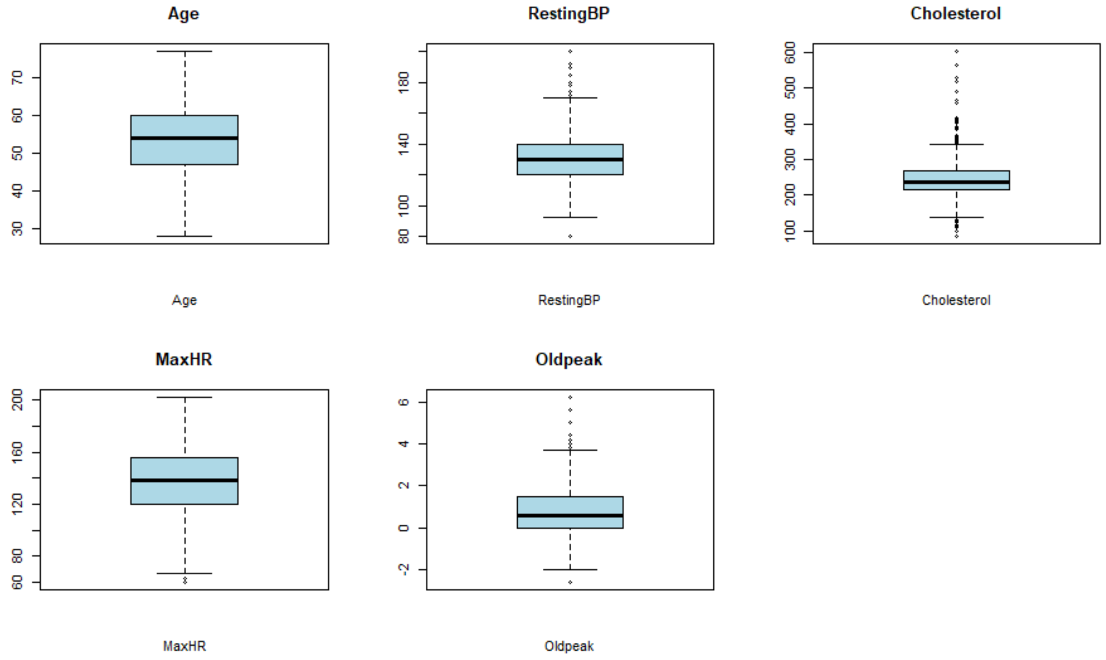
3. Data Exploration

3.1 Univariate Analysis

We perform a univariate analysis on all our variables to get a sense of their distributions and characteristics.

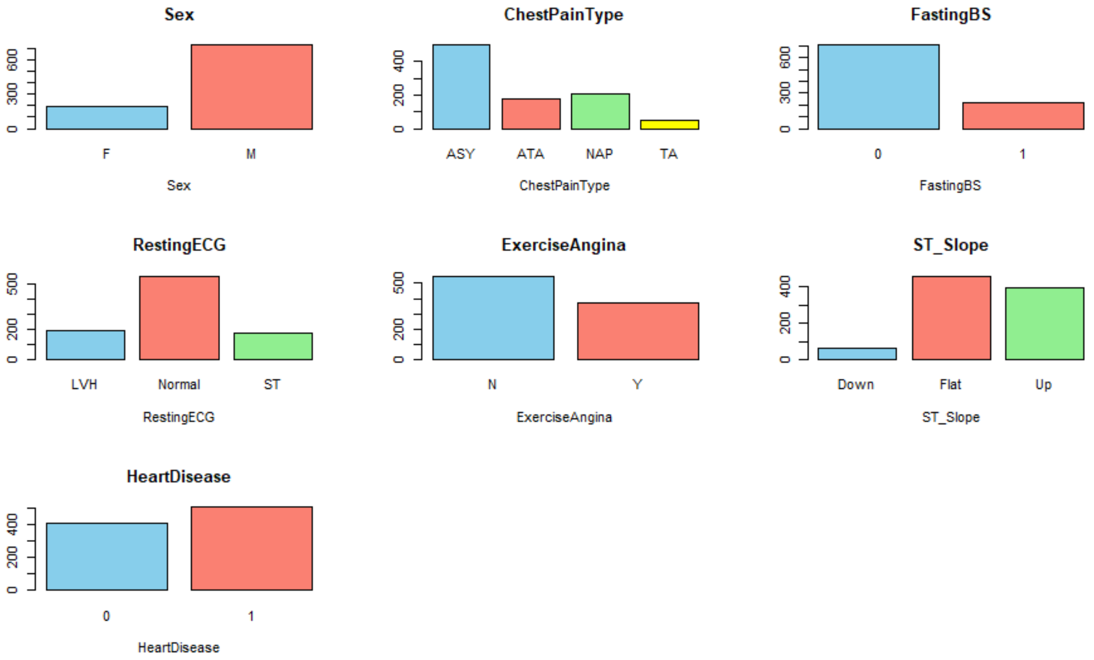
Numerical Variables





`Age` and `MaxHR` seem to be normally distributed, `RestingBP` presents a mild right skewness, while `Cholesterol` and `Oldpeak` are also right skewed with many outliers, where the latter seems also to be bimodal.

Categorical Variables

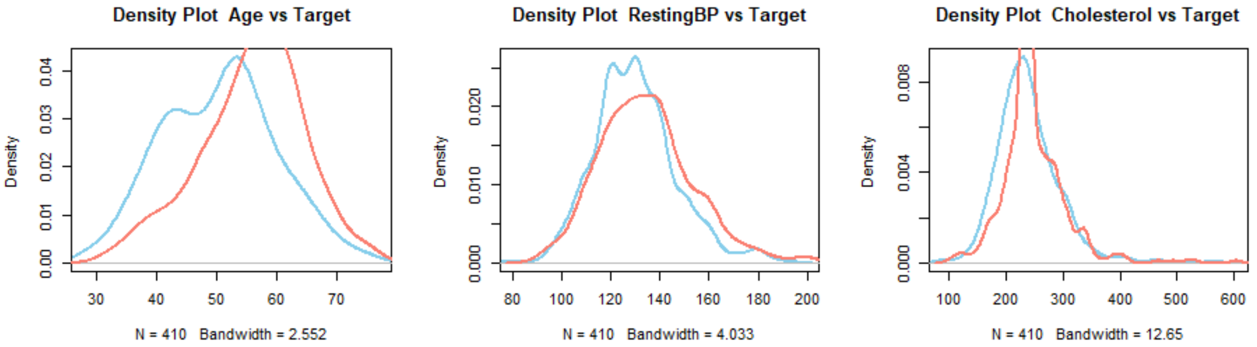


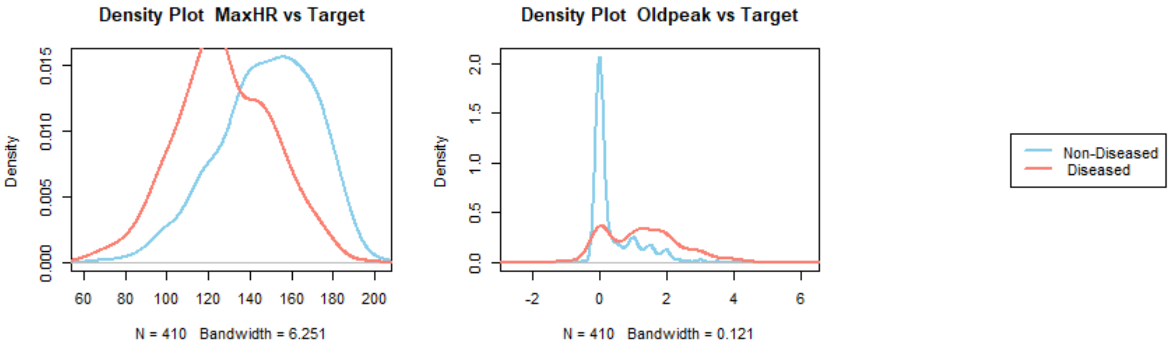
Much less females observations than males in `Sex`, `FastingBS` also unbalanced, `ChestPainType` is mostly composed by asymptomatic patients, `RestingECG` is predominantly categorized as normal, `ExerciseAngina` is not too unbalanced but fewer patients experience it than do not, `ST_Slope` has a majority of flat and up-sloping observations, while down-sloping ST are much less common. Our response variable `HeartDisease` is quite balanced, with more patients manifesting the condition.

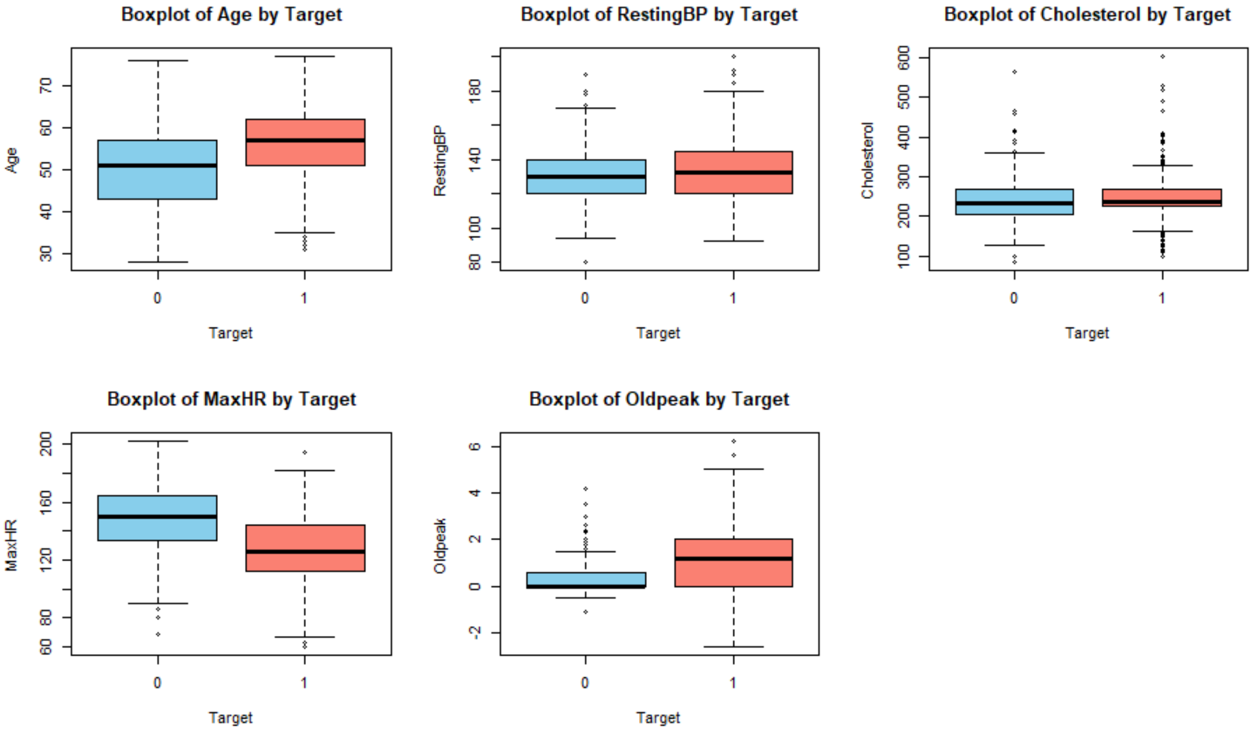
3.2 Bivariate Analysis

We now perform a bivariate analysis to understand the relationships between the candidate predictors and the response variable. We start from the numerical variables and then analyze the categorical variables, which require a different treatment from the numerical ones, we inspect them using a contingency table and performing chi-squared tests.

Numerical Variables





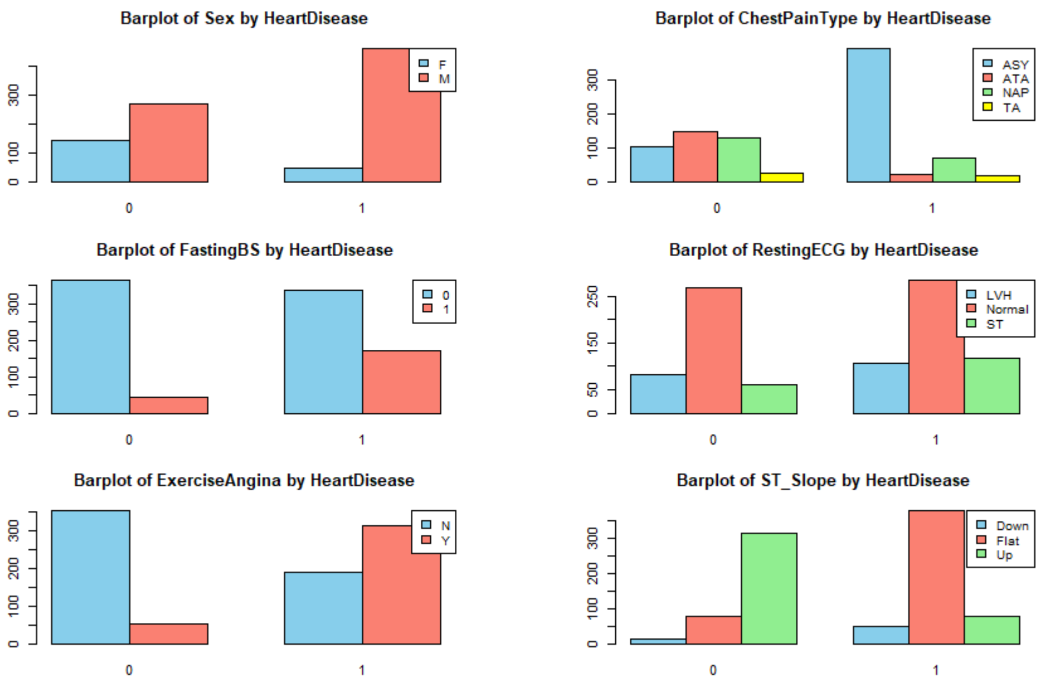


- `Age`: individuals with heart disease tend to be older.
- `RestingBP`: slightly higher resting blood pressure is observed in individuals with heart disease.
- `Cholesterol`: higher cholesterol levels are more common among those with heart disease.
- `MaxHR`: lower maximum heart rate is seen in individuals with heart disease.
- `Oldpeak`: higher oldpeak values are associated with heart disease.

When analyzing `Cholesterol` we have to consider that a bunch of values have been replaced with the median, so the considerations made might be an underestimation. Furthermore, we can spot again the bimodal distribution in the `Oldpeak` variable, which is probably given by a subgroup that has a positive diagnosis.

The consideration about `MaxHR` also needs some explaining. By intuition, one could associate a lower heart rate to a more efficient heart functioning and thus an healthy condition. This is in general true when we are talking about a resting measurement, however, we have to consider that in our data this parameter was taken during physical exercise, and in this state, medical literature suggests that it is important for the heart to be able to reach high frequencies. Hence, low frequencies during exercise can indicate heart problems, which is consistent with our analysis.

Categorical Variables



Much more males have a heart disease diagnosis with respect to females, these could be in part explained from the unbalanced observations, but for the positive diagnosis the difference is very significant. Most individuals with heart disease are asymptomatic in terms of chest pain, while ATA and NAP chest pains are more common in individuals without heart disease. This could seem counter intuitive, but it will be discussed later on in our analysis. Individuals with heart disease have a higher proportion of fasting blood sugar $\geq$ 120 mg/dL compared to those without heart disease. Individuals with heart disease have a higher proportion of LVH and ST observations in resting ECG results than individuals without the condition, although it does not seem to be significant. Exercise-induced angina is more common in individuals with heart disease compared to those without it. Individuals with heart disease have a higher proportion of flat and down-sloping ST segments, while healthy individuals mostly show up-sloping ST segments, the difference in proportion for down-sloping ST segments is less noticeable but we need to consider the fact that the overall observations for this category are much less than the others.

```r
calculate_chi_square <- function(data, target_var, categorical_var) {
  contingency_table <- table(data[[categorical_var]], data[[target_var]])
  chi_square_test <- chisq.test(contingency_table)
  return(list(contingency_table = contingency_table, chi_square_test = chi_square_test))
}

cat_vars <- data[, sapply(data, is.factor)]
for (col in colnames(cat_vars[,-7])) {
  result <- calculate_chi_square(data, "HeartDisease",col)
  cat(paste("Variable:", col, "\n"))
  print(result$chi_square_test)
}
```

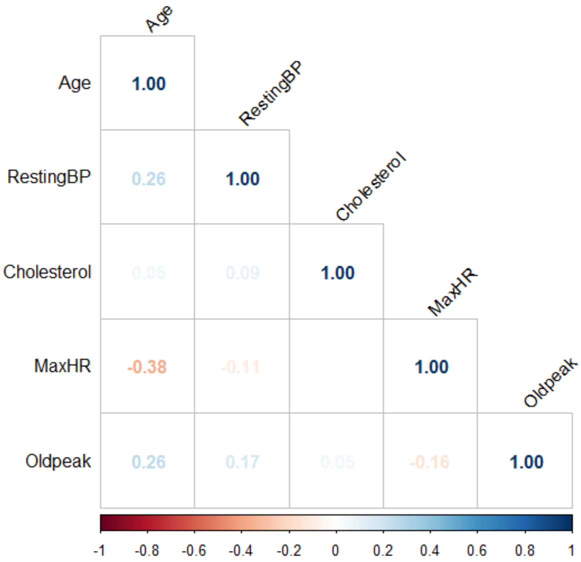
```
## Variable: Sex
##  Pearson's Chi-squared test with Yates' continuity correction
## data:  contingency_table
## X-squared = 84.145, df = 1, p-value < 2.2e-16
##
```

```
## Variable: ChestPainType
##  Pearson's Chi-squared test
## data:  contingency_table
## X-squared = 268.07, df = 3, p-value < 2.2e-16
##
## Variable: FastingBS
##  Pearson's Chi-squared test with Yates' continuity correction
## data:  contingency_table
## X-squared = 64.321, df = 1, p-value = 1.057e-15
##
## Variable: RestingECG
##  Pearson's Chi-squared test
## data:  contingency_table
## X-squared = 10.931, df = 2, p-value = 0.004229
##
## Variable: ExerciseAngina
##  Pearson's Chi-squared test with Yates' continuity correction
## data:  contingency_table
## X-squared = 222.26, df = 1, p-value < 2.2e-16
##
## Variable: ST_Slope
##  Pearson's Chi-squared test
## data:  contingency_table
## X-squared = 355.92, df = 2, p-value < 2.2e-16
```

Using a significance level of 0.05 we can confidently reject the null hypothesis for all of our categorical variables, this means that, based on our data, there is enough evidence to conclude that a significant association exists between these variables and the diagnosis.

3.3 Correlation Analysis

As a last step to our EDA, we perform a correlation analysis to understand the relationships between the numerical variables and to grasp which could be the variables characterized by a multicollinearity problem, thus we provide a correlation matrix.



Age seems to be somewhat ‘central’ for all the other numeric variables, with the only exception of Cholesterol. The strongest correlation observed is between `Age` and `MaxHR` (-0.38) telling us that as age increases, the maximum heart rate that one can achieve during exercise decreases. This is consistent with the previous explanation on the heart rate, and can also suggest that age is an important factor in for predicting heart disease. Anyways, there are no particularly strong correlations between the variables, so we can proceed with the modeling phase without worrying about multicollinearity issues.

4. Data Modeling

4.1 Splitting and scaling

Before starting the modeling process, we split our dataset into training and test sets, with the training set comprising 80% of the data and the test set comprising the remaining 20%. This division allows us to train our models on the majority of the data while reserving a separate subset for evaluating the model's performance on unseen data, which helps to prevent overfitting.

```
set.seed(123)
train_indices <- sample(1:nrow(data), 0.8 * nrow(data))
test_indices <- setdiff(1:nrow(data), train_indices)
train_set <- data[train_indices, ]
test_set <- data[test_indices, ]
```

Given that our numerical variables are measured in different units, we proceed with standardization to ensure they are on a comparable scale. Standardization transforms the data such that it has a mean of zero and a standard deviation of one. By doing so, we ensure that each feature contributes equally to the model's learning process, preventing features with larger scales from dominating those with smaller scales.

```
numeric_vars_train <- train_set[, sapply(train_set, is.numeric)]
cat_vars_train <- train_set[, sapply(train_set, is.factor)]

numeric_vars_test <- test_set[, sapply(test_set, is.numeric)]
cat_vars_test <- test_set[, sapply(test_set, is.factor)]

data_num_scaled_train <- scale(numeric_vars_train)
data_num_scaled_df_train <- as.data.frame(data_num_scaled_train)
train_set <- cbind(data_num_scaled_df_train, cat_vars_train)

data_num_scaled_test <- scale(numeric_vars_test)
data_num_scaled_df_test <- as.data.frame(data_num_scaled_test)
test_set <- cbind(data_num_scaled_df_test, cat_vars_test)
```

4.2 Simple Logistic Regression

Logistic regression is a statistical method used to model the probability of a binary outcome based on one or more predictor variables. In this context, we use logistic regression to estimate the probability of a patient having heart disease based on some observed parameters. Logistic regression works by fitting a logistic function (also known as the sigmoid function) to the data. This function maps any input value to a value between 0 and 1, which can be interpreted as a probability. The logistic function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where z is a linear combination of the input features. In other words, the logistic regression model calculates z as:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Here, β_0 is the intercept, and $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients of the predictor variables $x_1, x_2, \ldots, x_n$.

By using logistic regression, we can understand the relationship between the patient's parameters and the likelihood of heart disease, making it a powerful tool for predictive modeling and decision-making.

```
lr_model <- glm(HeartDisease ~ . , data=train_set, family=binomial)
lr_model_null <- glm(HeartDisease ~ +1 , data=train_set, family=binomial)
anova(lr_model_null, lr_model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ +1
## Model 2: HeartDisease ~ Age + RestingBP + Cholesterol + MaxHR + Oldpeak +
##     Sex + ChestPainType + FastingBS + RestingECG + ExerciseAngina +
##     ST_Slope
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       733    1007.44
## 2       718     494.81 15   512.63 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
sort(vif(lr_model))
```

```
##  [1] 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000 1.000000
##  [9] 1.028989 1.032733 1.035209 1.045696 1.051951 1.054720 1.058819 1.071657
## [17] 1.093353 1.106600 1.106864 1.112434 1.135282 1.137876 1.159651 1.195699
## [25] 1.213187 1.225147 1.288866 1.294761 1.344791 1.429030 2.000000 2.000000
## [33] 3.000000
```

```r
summary(lr_model)
```

```
## Call:
## glm(formula = HeartDisease ~ ., family = binomial, data = train_set)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.064792   0.568842  -1.872 0.061226 .
## Age               0.204656   0.136348   1.501 0.133360
## RestingBP         0.023059   0.121995   0.189 0.850079
## Cholesterol       0.155349   0.116530   1.333 0.182490
## MaxHR            -0.155330   0.137893  -1.126 0.259972
## Oldpeak           0.405825   0.136723   2.968 0.002995 **
## SexM              1.667146   0.298144   5.592 2.25e-08 ***
## ChestPainTypeATA -1.944001   0.364373  -5.335 9.54e-08 ***
## ChestPainTypeNAP -1.788925   0.282340  -6.336 2.36e-10 ***
## ChestPainTypeTA  -1.232921   0.476543  -2.587 0.009675 **
## FastingBS1        1.123566   0.289740   3.878 0.000105 ***
## RestingECGNormal  0.002042   0.293660   0.007 0.994451
## RestingECGST      0.107087   0.390844   0.274 0.784093
## ExerciseAnginaY   0.702816   0.268107   2.621 0.008757 **
## ST_SlopeFlat      1.458303   0.470061   3.102 0.001920 **
## ST_SlopeUp       -0.789930   0.485254  -1.628 0.103553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1007.44  on 733  degrees of freedom
## Residual deviance:  494.81  on 718  degrees of freedom
## AIC: 526.81
##
## Number of Fisher Scoring iterations: 5
```

To assess the performance and significance of our model, we compared it to the null model, which only includes an intercept. Using a chi-squared test, we found that our logistic regression model is significantly different from the null model, indicating that our predictor variables collectively have a meaningful impact on the probability of heart disease. Additionally, the Variance Inflation Factor (VIF) values confirmed that there are no multicollinearity issues among the predictor variables. However, examining the summary of our logistic regression model revealed that some variables are not statistically significant. Therefore, we proceeded with feature selection to refine our model.

4.3 Stepwise Logistic Regression

```
step_model_lr <- stepAIC(lr_model, direction = 'both')
```

```
summary(step_model_lr)
```

```
## Call:
## glm(formula = HeartDisease ~ Age + Oldpeak + Sex + ChestPainType +
##     FastingBS + ExerciseAngina + ST_Slope, family = binomial,
##     data = train_set)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.0686     0.5248  -2.036  0.04174 *
## Age                0.2647     0.1232   2.149  0.03167 *
## Oldpeak            0.3860     0.1332   2.898  0.00376 **
## SexM               1.6536     0.2933   5.638 1.72e-08 ***
## ChestPainTypeATA  -1.9730     0.3578  -5.514 3.51e-08 ***
## ChestPainTypeNAP  -1.8518     0.2783  -6.653 2.87e-11 ***
## ChestPainTypeTA   -1.2985     0.4722  -2.750  0.00596 **
## FastingBS1         1.1498     0.2872   4.004 6.23e-05 ***
## ExerciseAnginaY    0.7999     0.2581   3.099  0.00194 **
## ST_SlopeFlat       1.5095     0.4615   3.271  0.00107 **
## ST_SlopeUp        -0.8356     0.4754  -1.758  0.07882 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1007.44  on 733  degrees of freedom
## Residual deviance:  498.25  on 723  degrees of freedom
## AIC: 520.25
##
## Number of Fisher Scoring iterations: 5
```

```
sort(vif(step_model_lr))
```

```
##  [1] 1.000000 1.000000 1.000000 1.000000 1.000000 1.024368 1.029006 1.032858
##  [9] 1.040649 1.058854 1.066795 1.069122 1.079230 1.082949 1.112215 1.143023
## [17] 1.155407 1.237023 1.356612 2.000000 3.000000
```

```
anova(lr_model, step_model_lr, test="Chisq")
```

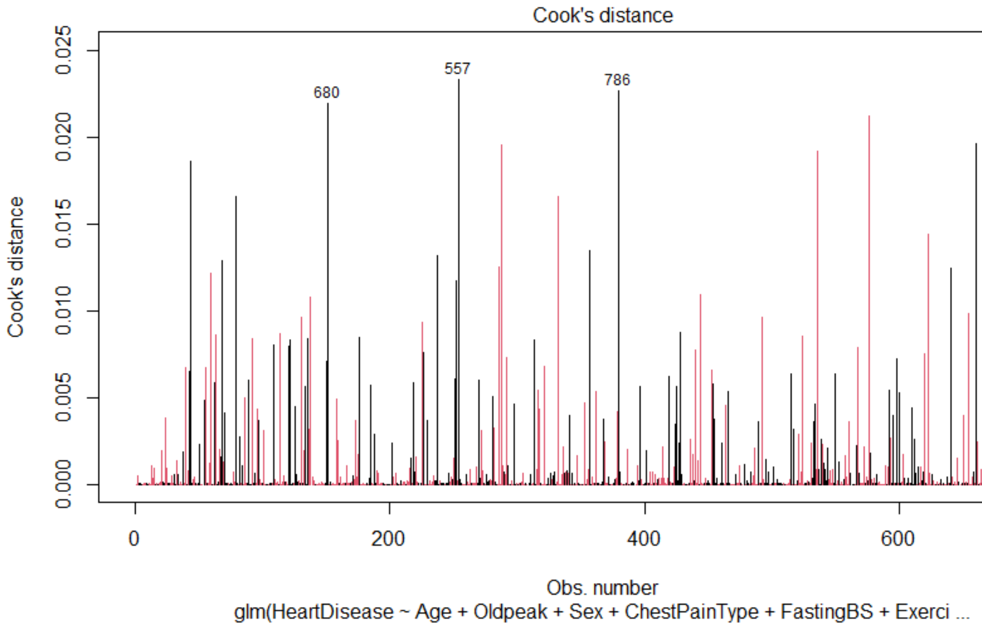
```
## Analysis of Deviance Table
##
## Model 1: HeartDisease ~ Age + RestingBP + Cholesterol + MaxHR + Oldpeak +
##     Sex + ChestPainType + FastingBS + RestingECG + ExerciseAngina +
##     ST_Slope
## Model 2: HeartDisease ~ Age + Oldpeak + Sex + ChestPainType + FastingBS +
##     ExerciseAngina + ST_Slope
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       718     494.81
## 2       723     498.25 -5  -3.4432    0.632
```

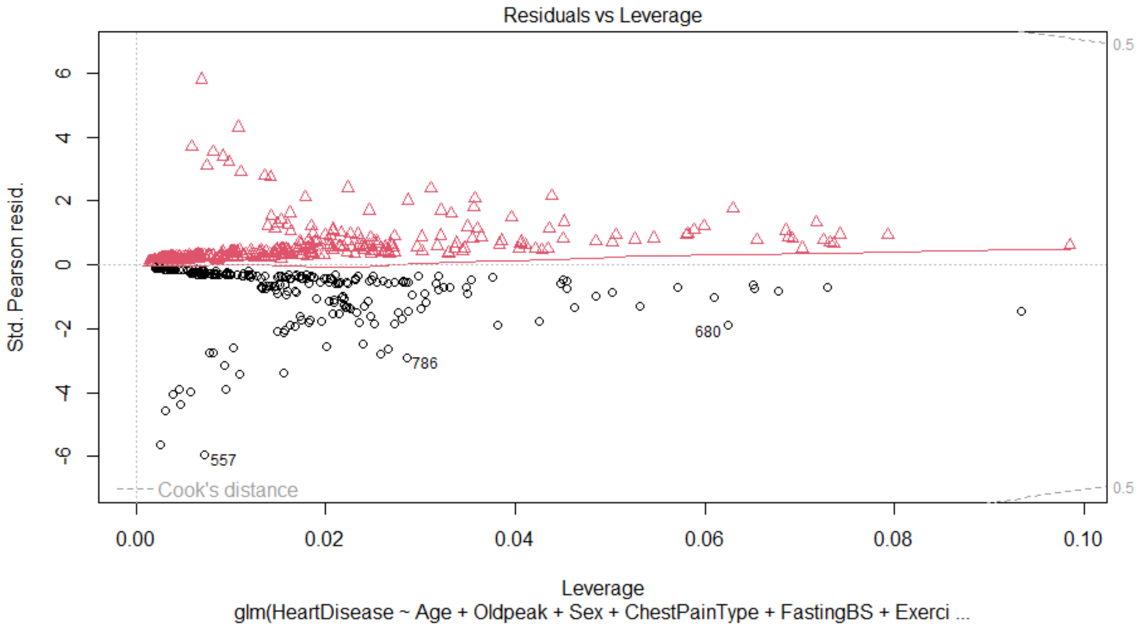
After performing stepwise feature selection, we found that the new, more parsimonious model is not statistically different from the full model. This result suggests that the predictors removed during the selection process do not significantly contribute to the model. Therefore, we prefer using this more parsimonious model as it simplifies interpretation and reduces the risk of overfitting, while still maintaining the model's predictive power. We then rely on the confusion matrix of our model to derive metrics such as accuracy, recall, precision, type I error, F1 score, and the AUC. Although all of the metrics are informative in their own way, our main focus will be to maximize the recall, also known as true positive rate, since it is crucial for our model to minimize type 2 errors (false negatives), being in the context of medical diagnosis. To streamline this process, we build two reusable functions for creating the confusion matrix and computing the associated metrics for any given model.

```r
compute_confusion_matrix <- function(Predicted, Actual) {
  conf_matrix <- table(Predicted, Actual)
  conf_df <- as.data.frame(matrix(0, nrow = 2, ncol = 2))
  colnames(conf_df) <- c("True Non-Disiased", "True Disiased")
  rownames(conf_df) <- c("Pred. Non-Disiased", "Pred. Disiased")
  conf_df[, 1:2] <- t(conf_matrix)
  conf_df["Total",] <- colSums(conf_df)
  conf_df <- cbind(conf_df, Total = rowSums(conf_df))
  return(conf_df)
}

pred_lr_prob <- predict(step_model_lr, test_set, type = "response")
pred_lr <- ifelse(pred_lr_prob > 0.5, 1, 0)
conf_matrix_lr <- compute_confusion_matrix(test_set$HeartDisease, pred_lr)

compute_metrics <- function(conf_matrix, Actual, Predicted_prob) {
  acc <- round((conf_matrix[1,1]+conf_matrix[2,2])/conf_matrix[3,3],3)
  prec <- round(conf_matrix[2,2]/conf_matrix[2,3],3)
  rec <- round(conf_matrix[2,2]/conf_matrix[3,2],3)
  spec <- round(conf_matrix[1,1]/conf_matrix[3,1],3)
  type_1 <- round(conf_matrix[2,1]/conf_matrix[3,1],3)
  f1_score <- round(2 * (prec * rec) / (prec + rec),3)
  roc_out <- roc(Actual, as.numeric(Predicted_prob))
  cat("Accuracy:", acc, "\n")
  cat("Precision:", prec, "\n")
  cat("Recall:", rec, "\n")
  cat("Specificity:", spec, "\n")
  cat("Type 1 error:", type_1, "\n")
  cat("F1 Score: ", f1_score, "\n")
  cat("AUC: ", auc(roc_out), "\n")
}
compute_metrics(conf_matrix_lr, test_set$HeartDisease, pred_lr_prob)
```

```
## Accuracy: 0.891
## Precision: 0.875
## Recall: 0.929
## Specificity: 0.849
## Type 1 error: 0.151
## F1 Score:  0.901
## AUC: 0.936
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	73	7	80
Pred. Positive	13	91	104
Total	86	98	184





```
lev_points <- c(680,557,786)
train_set_clean <- train_set[!rownames(train_set) %in% lev_points, ]
lr_model <- glm(HeartDisease ~ . , data=train_set_clean, family=binomial)
step_model_lr <- stepAIC(lr_model, direction = 'both')
```

```
pred_lr_prob <- predict(step_model_lr, test_set, type = "response")
pred_lr <- ifelse(pred_lr_prob > 0.5, 1, 0)
conf_matrix_lr <- compute_confusion_matrix(test_set$HeartDisease, pred_lr)
compute_metrics(conf_matrix_lr, test_set$HeartDisease, pred_lr_prob)
```

```
## Accuracy: 0.897
## Precision: 0.883
## Recall: 0.929
## Specificity: 0.86
## Type 1 error: 0.14
## F1 Score:  0.905
## AUC: 0.936
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	74	7	81
Pred. Positive	12	91	103
Total	86	98	184

We computed a residual diagnostic to see if there are some points that influence our model. We identified three influential points using Cook's distance and leverage plot. We plan to exclude these points and rebuild the model to compare results and assess their impact on model performance. From the result we can see that model achieve better quality metrics, especially precision and specificity gain significant points with same recall's level making this more powerful model. Furthermore, this can be appreciated also by looking at the AIC, which went down from 526.81 to 503.91.

```
summary(step_model_lr)
```

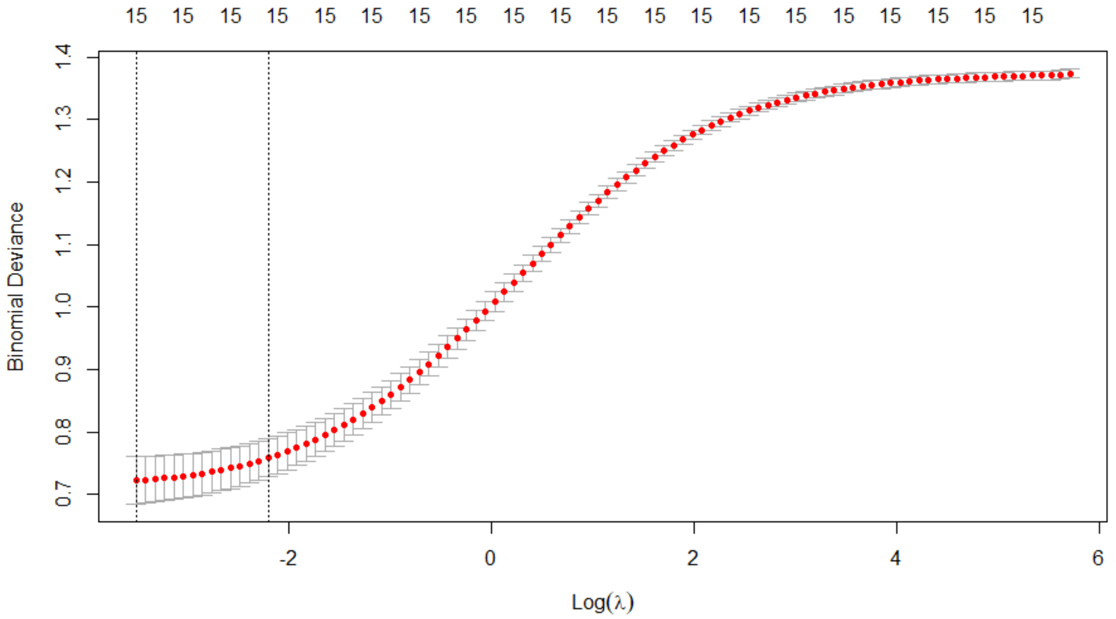
```
## Call:
## glm(formula = HeartDisease ~ Age + Cholesterol + Oldpeak + Sex + 
##     ChestPainType + FastingBS + ExerciseAngina + ST_Slope, family = binomial, 
##     data = train_set_clean)
## 
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       -0.8818     0.5541  -1.591  0.11152    
## Age                0.3228     0.1259   2.565  0.01033 *  
## Cholesterol        0.1702     0.1165   1.461  0.14396    
## Oldpeak            0.3707     0.1362   2.722  0.00649 ** 
## SexM               1.7621     0.3018   5.839 5.25e-09 ***
## ChestPainTypeATA  -1.9908     0.3630  -5.484 4.17e-08 ***
## ChestPainTypeNAP  -1.8842     0.2841  -6.631 3.33e-11 ***
## ChestPainTypeTA   -0.9766     0.4960  -1.969  0.04895 *  
## FastingBS1         1.2847     0.2983   4.308 1.65e-05 ***
## ExerciseAnginaY    0.7856     0.2621   2.997  0.00273 ** 
## ST_SlopeFlat       1.2205     0.4929   2.476  0.01328 *  
## ST_SlopeUp        -1.1373     0.5057  -2.249  0.02453 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 1002.52  on 730  degrees of freedom
## Residual deviance:  479.91  on 719  degrees of freedom
## AIC: 503.91
## Number of Fisher Scoring iterations: 5
```

4.4 Ridge Logistic Regression

Ridge logistic regression adds a penalty term to the loss function that discourages large coefficients. This regularization term helps prevent overfitting by shrinking the coefficients, thereby improving the model's generalization to new data. It effectively balances the trade-off between fitting the training data and maintaining simplicity in the model.

```
X_train <- model.matrix(HeartDisease~., train_set)[,-1]
y_train <- as.numeric(as.character(train_set$HeartDisease))
X_test <- model.matrix(HeartDisease~., test_set)[,-1]
y_test <- as.numeric(as.character(test_set$HeartDisease))
```

```r
ridge_cv <- cv.glmnet(X_train, y_train, alpha = 0, family = "binomial",
type.measure = "deviance", nfolds = 10)
plot(ridge_cv)
```



```r
lambda = ridge_cv$lambda.min
cat("The value for the minimum lambda is ", lambda)
```

```
## The value for the minimum lambda is  0.03025753
```

```r
pred_ridge_prob <- predict(ridge_cv, X_test, type = "response", s = lambda)
pred_ridge <- ifelse(pred_ridge_prob > 0.5, 1, 0)

conf_matrix_ridge <- compute_confusion_matrix(y_test, pred_ridge)
compute_metrics(conf_matrix_ridge, y_test, pred_ridge_prob)
```

```
## Accuracy: 0.891
## Precision: 0.89
## Recall: 0.908
## Specificity: 0.872
## Type 1 error: 0.128
## F1 Score:  0.899
## AUC: 0.939
```

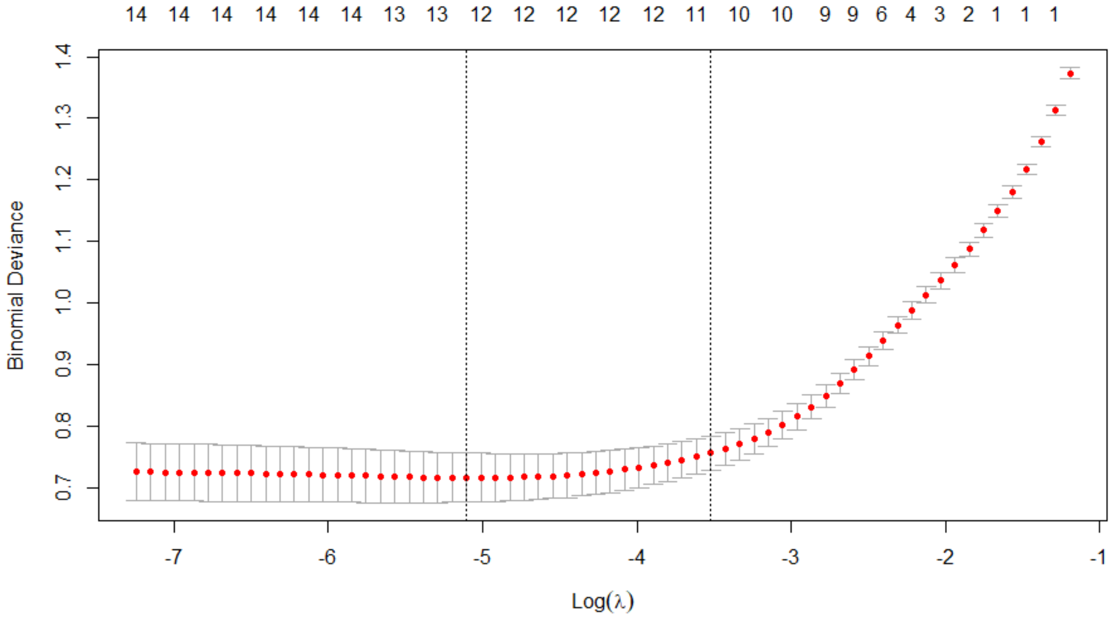
Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	75	9	84
Pred. Positive	11	89	100
Total	86	98	184

We used a cross validation to find the best lambda. The relatively tight confidence intervals in the graph indicate that the results are quite stable across the different folds of the cross-validation. The plot shows that the deviance increases with increasing lambda, which is expected as high penalization reduces the model's complexity. The use of deviance respect to classification error is more appropriate since it is more sensitive to the probability estimates and the value of lambda min generated, using deviance brings better results in term of metrics than the one generated by classification error. Predictors shrunk near zero, and therefore less effective, include `RestingBP` and `RestingECG`. Other predictors with smaller absolute values are `Age`, `Cholesterol`, and `MaxHR`. As we can see from the metrics, the model achieves great results in terms of recall. AIC is -376.33.

4.5 Lasso Logistic Regression

Lasso logistic regression, adds a penalty term to the loss function that encourages sparsity by shrinking some coefficients to exactly zero. This allows for feature selection within the model, as less important predictors are eliminated. Ridge logistic regression was also adding a penalty term to shrink coefficients but it does not force them to zero, resulting in no feature selection.

```
lasso_cv <- cv.glmnet(X_train, y_train, alpha = 1, family = "binomial",
type.measure = "deviance", nfolds = 10)
plot(lasso_cv)
```



```
lambda = lasso_cv$lambda.min
cat("The value for the minimum lambda is ", lambda)
```

```
## The value for the minimum lambda is  0.006079443
```

```
pred_lasso_prob <- predict(lasso_cv, X_test, type = "response", s = lambda)
pred_lasso <- ifelse(pred_lasso_prob > 0.5, 1, 0)

conf_matrix_lasso <- compute_confusion_matrix(y_test, pred_lasso)
compute_metrics(conf_matrix_lasso, y_test, pred_lasso_prob)
```

```
## Accuracy: 0.897
## Precision: 0.891
## Recall: 0.918
## Specificity: 0.872
## Type 1 error: 0.128
## F1 Score:  0.904
## AUC: 0.938
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	75	8	83
Pred. Positive	11	90	101
Total	86	98	184

```
lasso_coef <- coef(lasso_cv, s = "lambda.min")
lasso_coef
```

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                           s1
## (Intercept)      -0.8245285
## Age               0.1726629
## RestingBP         .
## Cholesterol       0.1029986
## MaxHR            -0.1535263
## Oldpeak           0.3503436
## SexM              1.4432060
## ChestPainTypeATA -1.7148267
## ChestPainTypeNAP -1.5552032
## ChestPainTypeTA  -0.8949806
## FastingBS1        0.9570158
## RestingECGNormal  .
## RestingECGST      .
## ExerciseAnginaY   0.6980938
## ST_SlopeFlat      1.2456712
## ST_SlopeUp       -0.8541404
```

As before we used cross validation technique to get the best value of lambda and we use again deviance type measure instead of classification error. Respect to ridge, lasso has a more aggressive penalization, so the model is more sparse and the number of employed predictors is lower. This is the best logistic regression seen so far looking at the metrics score. the model presents significant parameter shrinkage, particularly notable with `RestingBP` and `RestingECG`, which have been effectively reduced to zero. Additionally, `Cholesterol` and `MaxHR` exhibit greater shrinkage compared to the ridge regression model, suggesting reduced importance attributed to these variables in the model's predictive performance. AIC is -391.77.

4.6 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a statistical method used primarily for classification tasks. It identifies linear combinations of features that best separate different classes or groups within a dataset. LDA achieves this by maximizing the distance between the means of different classes while minimizing the variation within each class. When dealing with Linear Discriminant Analysis (LDA), we assume that the variables are normally distributed and exhibit equal variance-covariance (homoscedasticity) across classes. However, our exploratory data analysis (EDA) indicated that these assumptions are not fully met in our data. Despite this, we proceed with the analysis while keeping these deviations in mind.

```r
lda_model <- lda(HeartDisease ~ ., data = train_set)
lda_model$scaling
```

```
##                           LD1
## Age               0.106580957
## RestingBP        -0.001823631
## Cholesterol       0.082577174
## MaxHR            -0.094971224
## Oldpeak           0.185613959
## SexM              0.778479102
## ChestPainTypeATA -1.162774068
## ChestPainTypeNAP -1.076879599
## ChestPainTypeTA  -0.657118694
## FastingBS1        0.536573591
## RestingECGNormal -0.016032759
## RestingECGST      0.059964785
## ExerciseAnginaY   0.407646663
## ST_SlopeFlat      0.673548325
## ST_SlopeUp       -0.774050565
```

```
pred_lda_prob <- predict(lda_model, test_set,type ='response')$posterior[,2]
pred_lda <- as.factor(ifelse(pred_lda_prob > 0.5, 1, 0))
conf_matrix_lda <- compute_confusion_matrix(test_set$HeartDisease, pred_lda)
compute_metrics(conf_matrix_lda, test_set$HeartDisease, pred_lda_prob)
```

```
## Accuracy: 0.897
## Precision: 0.891
## Recall: 0.918
## Specificity: 0.872
## Type 1 error: 0.128
## F1 Score:  0.904
## AUC: 0.937
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	75	8	83
Pred. Positive	11	90	101
Total	86	98	184

Each coefficient in Linear Discriminant Analysis (LDA) indicates the importance and direction of the influence of a predictor variable on class discrimination. A positive coefficient suggests that an increase in the variable is associated with a higher probability of belonging to a particular class, whereas a negative coefficient indicates the opposite.

Strongly Influencing Variables:

- `SexM`, `ExerciseAnginaY`, and `ST_SlopeFlat` have significant positive coefficients, indicating a strong positive influence on class discrimination.

Weakly Influencing Variables:

- `RestingBP`, `RestingECGNormal`, and `RestingECGST` have coefficients very close to zero, indicating that their influence on class discrimination is minimal.

By computing the metrics of this model, we observe its validity since it gives us the same result of the lasso logistic regression model in terms of metrics scores. AIC is 677.22.

4.7 Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a classification technique similar to LDA, but it allows for each class to have its own covariance matrix. This means QDA can model more complex decision boundaries that are quadratic, unlike LDA which assumes linear boundaries and equal covariance matrices across classes. This makes QDA more flexible but also more prone to overfitting, especially with smaller datasets.

```
qda_model <- qda(HeartDisease ~ ., data = train_set)

pred_qda_prob <- predict(qda_model, test_set,type ='response')$posterior[,2]
pred_qda <- as.factor(ifelse(pred_qda_prob>0.5,1,0))
conf_matrix_qda <- compute_confusion_matrix(test_set$HeartDisease, pred_qda)
compute_metrics(conf_matrix_qda, test_set$HeartDisease, pred_qda_prob)
```

```
## Accuracy: 0.864
## Precision: 0.861
## Recall: 0.888
## Specificity: 0.837
## Type 1 error: 0.163
## F1 Score:  0.874
## AUC: 0.915
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	72	11	83
Pred. Positive	14	87	101
Total	86	98	184

The QDA model metrics score appears to be worse than the LDA model, which could be because the relationships between the variables and the classes are more linear. AIC is 1069.21, and this makes LDA, with its linear decision boundaries, more effective. Therefore, we consider LDA as the best discriminant model.

5. Data Interpretation

From the previous analysis we saw that the two best predictive models are the Lasso logistic regression and the LDA, so we keep these two models and we focus now on their interpretation.

5.1 Lasso Model

After some experimentation with different thresholds, we concluded the best one was the standard `0.5`, which means that our model classifies patients as diseased when the probability is greater or equal than 50%. From the lasso coefficients, we can identify the most important predictors and major risk factors for heart disease. The absolute values of the coefficients represent the magnitude of the effect that a one-unit increase in a variable has on the logit, while keeping other variables fixed. The sign of the coefficient indicates the direction of the effect: positive coefficients suggest an increased probability of heart disease, making them risk factors, while negative coefficients suggest a decreased probability.

The most influential variables are:

- `Oldpeak`, `Sex`, `ChestPainType`, `FastingBS`, `ExerciseAngina` and `ST_Slope`.

Other important variables, but with slightly less significant values, include:

- `Age`, `Cholesterol`, and `MaxHR`.

The odds ratios are calculated as the exponential of the coefficients. It is a measure used in statistics to quantify the strength of the association between two events. Specifically, in the context of logistic regression, it represents the change in odds of the outcome occurring for a one-unit increase in the predictor variable, while holding other variables constant.

```r
lasso_coef <- coef(lasso_cv, s = "lambda.min")
odds_ratios <- exp(lasso_coef)
odds_ratios
```

```
## 16 x 1 Matrix of class "dgeMatrix"
##                        s1
## (Intercept)      0.4384417
## Age              1.1884654
## RestingBP        1.0000000
## Cholesterol      1.1084899
## MaxHR            0.8576782
## Oldpeak          1.4195552
## SexM             4.2342491
## ChestPainTypeATA 0.1799949
## ChestPainTypeNAP 0.2111465
## ChestPainTypeTA  0.4086155
## FastingBS1       2.6039144
## RestingECGNormal 1.0000000
## RestingECGST     1.0000000
## ExerciseAnginaY  2.0099177
## ST_SlopeFlat     3.4752666
## ST_SlopeUp       0.4256489
```

We can see that each unit increase in `Age` increases the probability of the event by approximately 20%. A unit increase in `Cholesterol` and `Oldpeak` increases the odds by 11% and 42%, respectively. A low `MaxHR` is considered a risk factor since a one-unit increase decreases the probability of being diseased by 14%.

The major risk factors identified are Sex, Fasting Blood Sugar, Exercise-Induced Angina, and ST Slope. In particular:

Being male makes having the disease approximately 4.23 times more probable compared to being female. Having a fasting blood sugar level greater than 120 mg/dl increases the probability of the disease by approximately 160%. Exercise-induced angina doubles the probability of the disease. A flat ST slope increases the probability of the disease by approximately 248% compared to patients with a downward slope. An upward ST slope decreases the probability of the disease by approximately 57% compared to patients with a downward slope. Another important risk factor in the model is being asymptomatic with respect to chest pain type. Specifically, the presence of certain types of chest pain (like ATA, NAP, and TA) significantly reduces the probability of the disease compared to the reference category.

5.2 LDA Model

Based on the LDA coefficients, the major risk factors for the outcome of interest appear to be same as in lasso model and they are:

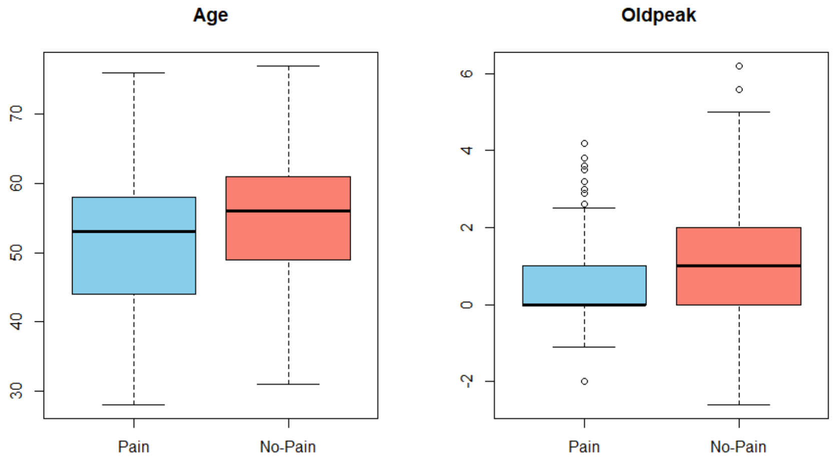
Sex (`Sex = M`): Being male significantly increases the likelihood of belonging to a specific class. Chest Pain Type (`ChestPainType = ASY)`: Specifically, having certain types of chest pain (ATA, NAP, TA) decreases the likelihood of belonging to a specific class. `Oldpeak`: Higher values of oldpeak increases the likelihood of belonging to a specific class, indicating greater risk. Fasting Blood Sugar (`FastingBS = 1`): Having a fasting blood sugar level greater than 120 mg/dl increases the likelihood of belonging to a specific class. Exercise-Induced Angina (`ExerciseAngina = Y`): The presence of exercise-induced angina increases the likelihood of belonging to a specific class. ST Slope (`ST_Slope = Flat`): A flat slope of the peak exercise ST segment increases the likelihood of belonging to a specific class, suggesting increased risk compared to other slopes. compared to the reference category.

We observe that both the LDA and Lasso logistic regression models achieve identical metric results. However, we tend to prefer the Lasso logistic regression model because it offers several advantages over LDA. Firstly, Lasso logistic regression does not assume a specific distribution of predictors, making it more robust when dealing with real-world data that may not conform to normality assumptions. Secondly, Lasso regression's ability to shrink coefficients to zero allows for automatic variable selection, which simplifies the model and enhances interpretability without sacrificing the predictive performance. Therefore, based on these considerations, we choose the Lasso logistic regression model as the best model for our analysis, as it provides a balance between performance, robustness, and interpretability suitable for our specific objectives.

5.3 Considerations on the ChestPainType Variable

We observed that both models identify asymptomatic chest pain type as a risk factor, while having any other form of chest pain decreases the probability of being diagnosed with the disease. This seems counterintuitive and warrants further investigation to better understand this phenomenon. An explanation on why being asymptomatic to chest pain seemed to be strongly correlated with heart disease is that most of the heart diseases do not bring chest pain as a symptom, and a fallacy in reasoning could be associating these kind of diseases with only heart attacks, which instead are just one subcategory of heart diseases. Anyways, to explore the matter further, we divided the ChestPainType variable into two groups: the first group consists of asymptomatic patients, while the second group includes patients with any other type of chest pain. Our objective is to examine how these two groups interact with other risk factor variables in our dataset.

```r
patient_with_chest_pain <- data[data$ChestPainType != 'ASY', ]
patient_without_chest_pain <- data[data$ChestPainType == 'ASY', ]
```



```
#ST_Slope
ST_table_with_chest_pain <- table(patient_with_chest_pain$ST_Slope)
ST_table_without_chest_pain <- table(patient_without_chest_pain$ST_Slope)
ST_contingency_table <-rbind(ST_table_with_chest_pain, ST_table_without_chest_pain)

#ExerciseAngina
EA_table_with_chest_pain <- table(patient_with_chest_pain$ExerciseAngina)
EA_table_without_chest_pain <- table(patient_without_chest_pain$ExerciseAngina)
EA_contingency_table <-rbind(EA_table_with_chest_pain, EA_table_without_chest_pain)

chisq.test(ST_contingency_table)
```

```
##  Pearson's Chi-squared test
## data:  ST_contingency_table
## X-squared = 118.94, df = 2, p-value < 2.2e-16
```

```
chisq.test(EA_contingency_table)
```

```
##  Pearson's Chi-squared test with Yates' continuity correction
## data:  EA_contingency_table
## X-squared = 168.01, df = 1, p-value < 2.2e-16
```

Based on the boxplot and contingency table analysis, it appears that asymptomatic patients often demonstrate higher values in risk factors like `Oldpeak`, `Age`, `ST_Slope`, and `ExerciseAngina`. This observation implies that in the context of assessing these specific risk factors, the classification based on chest pain type may not significantly enhance the predictive power beyond what is already captured by these other variables. Therefore, while chest pain type is traditionally considered a crucial symptom for diagnosing heart disease, its utility in predicting these specific risk factors may be less pronounced compared to other physiological markers.

```
X_train_reduced <- model.matrix(HeartDisease~., train_set[,-c(7)])[,-1]
y_train <- as.numeric(as.character(train_set$HeartDisease))

X_test_reduced <- model.matrix(HeartDisease~., test_set[,-c(7)])[,-1]
y_test <- as.numeric(as.character(test_set$HeartDisease))
lasso_red <- cv.glmnet(X_train_reduced, y_train, alpha = 1, family = "binomial",
type.measure = "deviance", nfolds = 10)
pred_lasso_red_prob <- predict(lasso_red, X_test_reduced, type = "response", s = lambda)
pred_lasso_red <- ifelse(pred_lasso_prob > 0.5, 1, 0)
```

```r
# Confusion Matrix
conf_matrix_lasso_reduced <- compute_confusion_matrix(y_test, pred_lasso_red)

# Metrics
compute_metrics(conf_matrix_lasso_reduced, y_test, pred_lasso_red_prob)
```

```
## Accuracy: 0.897
## Precision: 0.891
## Recall: 0.918
## Specificity: 0.872
## Type 1 error: 0.128
## F1 Score:  0.904
## AUC: 0.932
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	75	8	83
Pred. Positive	11	90	101
Total	86	98	184

```r
lasso_red_coef <- coef(lasso_red, s = "lambda.min")
lasso_red_coef
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)      -1.6266004
## Age               0.1614345
## RestingBP         .
## Cholesterol       0.1150675
## MaxHR            -0.2902414
## Oldpeak           0.4078527
## SexM              1.4634192
## FastingBS1        1.0023697
## RestingECGNormal  .
## RestingECGST      .
## ExerciseAnginaY   1.0347072
## ST_SlopeFlat      1.2347232
## ST_SlopeUp       -0.9446777
```

```r
odds_ratios_red <- exp(lasso_red_coef)
odds_ratios_red
```

```
## 13 x 1 Matrix of class "dgeMatrix"
##                         s1
## (Intercept)      0.1965968
## Age              1.1751955
## RestingBP        1.0000000
## Cholesterol      1.1219492
## MaxHR            0.7480830
## Oldpeak          1.5035856
## SexM             4.3207077
## FastingBS1       2.7247309
## RestingECGNormal 1.0000000
## RestingECGST     1.0000000
## ExerciseAnginaY  2.8142821
## ST_SlopeFlat     3.4374270
## ST_SlopeUp       0.3888049
```

We therefore removed the ChestPainType variable to evaluate its impact on our models. Surprisingly, even without this variable, the models produced practically identical results, indicating that chest pain type might introduce a confounding effect. Despite being initially considered one of the most significant predictors, its exclusion did not notably affect the models' predictive accuracy. Looking at the odds ratios we can see that the values of the impact of the predictors have slightly changed and their weight has been redistributed to account for the removed variable. In particular, predictors like `MaxHR`, `Oldpeak`, `Sex = M`, `FastingBS = 1`, and `Exercise Angina = Y`, are now compensating the removal, with the latter being where most of the weight was redistributed. Furthermore AIC is -397.77, the lowest observed so far. All of these findings suggest that chest pain type does not contribute additional information beyond what is already captured by other variables in our models. Consequently, we have opted to retain this streamlined model as our final one.

6. Conclusions and Potential Applications

In conclusion, the optimal predictive model chosen is the Lasso logistic regression excluding the `ChestPainType` variable to mitigate potential confounding effects. The primary risk factors for heart disease identified are: male sex, high oldpeak values, fasting blood sugar higher than 120 mg/dL, exercise angina, and flat ST slope. Secondary risk factors are: age, cholesterol and low maximum heart rate achieved during exercise. An interesting thing to notice is that most of these risk factors can be evaluated by performing cardiac stress tests, which suggests the singular utility that the model can have in predicting heart disease.

Based on this insight, we plan to build a model using only stress test related variables and easily obtainable demographic variables such as Age and Sex. We will then compare its performance with our full model. If the difference in performance is negligible, this simplified model could serve as a dependable baseline, potentially reducing the need for additional tests like blood analyses in routine medical assessments, helping practitioners to reach a faster diagnosis and to potentially start the treatment process in a timely manner.

```r
# Lasso regression with demographic and stress test parameters
X_train_stress <- model.matrix(HeartDisease~., train_set[,-c(3,7,8)])[,-1]
y_train <- as.numeric(as.character(train_set$HeartDisease))

X_test_stress <- model.matrix(HeartDisease~., test_set[,-c(3,7,8)])[,-1]
y_test <- as.numeric(as.character(test_set$HeartDisease))
lasso_stress <- cv.glmnet(X_train_stress, y_train, alpha = 1, family = "binomial",
type.measure = "deviance", nfolds = 10)
lambda = lasso_cv$lambda.min
cat("The value for the minimum lambda is ", lambda)
```

```
## The value for the minimum lambda is  0.006079443
```

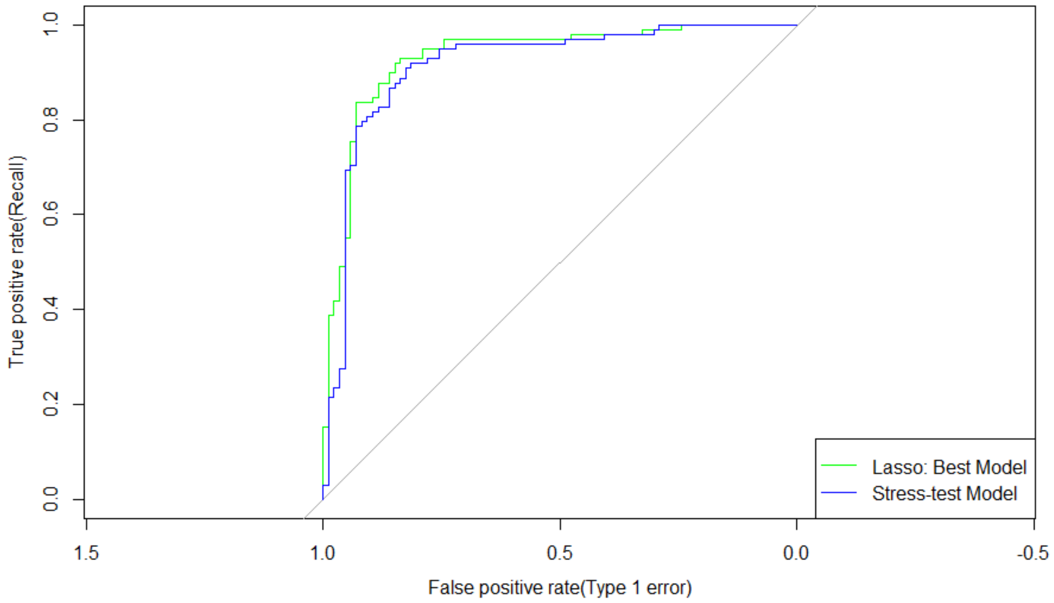
```r
pred_lasso_stress_prob <- predict(lasso_stress, X_test_stress, type = "response",
s = lambda)
pred_lasso_stress <- ifelse(pred_lasso_stress_prob > 0.5, 1, 0)

conf_matrix_lasso_stress<- compute_confusion_matrix(y_test, pred_lasso_stress)
compute_metrics(conf_matrix_lasso_stress, y_test, pred_lasso_stress_prob)
```

```
## Accuracy: 0.864
## Precision: 0.869
## Recall: 0.878
## Specificity: 0.849
## Type 1 error: 0.151
## F1 Score:  0.873
## AUC: 0.916
```

Confusion Matrix	True Negative	True Positive	Total
Pred. Negative	73	12	85
Pred. Positive	13	86	99
Total	86	98	184

```r
lasso_stress_coef <- coef(lasso_stress, s = "lambda.min")
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)      -1.19398965
## Age               0.21687717
## RestingBP         .
## MaxHR            -0.28785933
## Oldpeak           0.37272782
## SexM              1.45398128
## RestingECGNormal -0.01137237
## RestingECGST      .
## ExerciseAnginaY   1.00844060
## ST_SlopeFlat      1.03445915
## ST_SlopeUp       -1.14091231
```



Model	Accuracy	Precision	Recall	Specificity	Type 1 error	F1 Score	AUC	AIC
Lasso Best	0.897	0.891	0.918	0.872	0.128	0.904	0.932	-397.77
Stress-test	0.864	0.869	0.878	0.849	0.151	0.873	0.916	-349.28

The final model using only stress test and demographic parameters remains highly effective, and this can be seen from the ROC plots which are very close-up. While there is a slight reduction in metrics compared to the full model, the results are still robust and reliable. Therefore we can confirm that this simplified model can be utilized in everyday clinical practice to predict the presence of heart disease in patients, based solely on cardiac stress tests.