# Predicting Heart Disease and Chest Pain Type

Cam Lunn, Atticus Patrick, & Owen Patrick

5/11/2022

## Abstract

We will be analyzing a public heart disease data set from kaggle where each row is an individual patient. As of now, our aim is to look into the key factors that determine heart disease and predict the occurrence of heart disease in individuals based on a number of heart-health related predictor variables. A secondary goal is to look at chest pain type and to try and predict this in patients as well. The data used in this study consists of 5 independent sub-datasets of heart health related data. The main response variables looked at in the study are heart disease status and chest pain type. We found that the data are well suited to make predictions for heart disease status when using a decision tree. Additionally, we discovered that predicting chest pain type was very difficult and could not fit an accurate model using KNN, QDA, or a decision tree.

## Introduction

Every year, 25% of all deaths in the US are attributed to heart disease. There are many different types, which respectively can have different root causes. Malfunctions of the valves, arteries, and other physiological components can lead to a patient developing heart disease. On the other hand, lack of exercise, diet, and other environmental and even genetic factors can play a role in this outcome as well. To be succinct: heart disease is one of the biggest health-related killer the United States faces. If we can better understand the variables that comprise the complex system of developing heart disease, we have a better shot at preventing it from happening. The main goal for this study is to determine what factors are associated with heart disease, and if they can be used to predict a patient's outcome for it, as well as what factors are associated with chest pain, and which of these factors can be used to predict types of chest pain.

Our goals / hypotheses:

1) Exploratory analysis: look at descriptive statistics, and group means. See if there are any relationships between variables, and look at a correlation matrix of the numeric variables.

2) Use PCA to see which variables are most important and related to each other.

3) See if heart disease and chest pain type can be classified:

a) LDA/QDA
b) KNN
c) Decision Tree

4) See if factor analysis is applicable.

## Data Description

| Name | Description | Levels |
|---|---|---|
| Age | Age of the patient | 28 yrs - 77 yrs |
| Sex | Sex of the patient | Male, Female |
| exang | exercise induced angina | (1 = yes; 0 = no) |
| caa | number of major vessels | (0-3) |
| cp | Chest Pain type chest pain type | Value 1: typical angina [TA] |
| | | Value 2: atypical angina [ATA] |
| | | Value 3: non-anginal pain [NAP] |
| | | Value 4: asymptomatic [ASY] |
| trtbps | resting blood pressure (in mm Hg) | 0 - 200 mm Hg |
| chol | cholestoral in mg/dl fetched via BMI sensor | 0-603 mg/dl |
| fbs | (fasting blood sugar > 120 mg/dl) | (1 = true; 0 = false) |
| rest_ecg | resting electrocardiographic results | Value 0: normal |
| | | Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or |
| | | Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria |
| thalach | maximum heart rate achieved | 60 - 202 bpm |
| target | chance of a heart attack | 0= less chance of heart attack; 1= more chance of heart attack |

---

HEART2: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

## Data Cleaning, Setup, & Exploration

We did some feature engineering and created levels within the 'Age' variable: {[28-37], [38-47], [48-57], [58-67], [68-77]}

```
knitr::opts_chunk$set(echo = TRUE)
pacman::p_load(tidyverse, rstatix, class,
               rpart, rpart.plot, dplyr, corrplot, MASS, caret, MVN,
               factoextra, psych)
source("Partial F Test function.R")
```

```r
heart2 <- read.csv("heart2.csv")

# ---------------------- Data for Classifying Heart Disease ----------------
-------

heart <- heart2  %>%
    mutate(Age = if_else(Age >= 28 & Age <= 37,
                         "28-37",
                         if_else(Age >= 38 & Age <= 47,
                                 "38-47",
                         if_else(Age >= 48 & Age <= 57,
                                 "48-57",
                         if_else(Age >= 58 & Age <= 67,
                                 "58-67",
                         if_else(Age >= 68 & Age <= 77,
                                 "68-77", "Not seen"))))),
           HeartDisease = if_else(HeartDisease == "0",
                       "Unaffected",
                       "Affected") %>% factor())

# ---------------------- Data for Classifying Chest Pain -------------------
---

heart_CP <- heart2  %>%
    mutate(Age = if_else(Age >= 28 & Age <= 37,
                         "28-37",
                         if_else(Age >= 38 & Age <= 47,
                                 "38-47",
                         if_else(Age >= 48 & Age <= 57,
                                 "48-57",
                         if_else(Age >= 58 & Age <= 67,
                                 "58-67",
                         if_else(Age >= 68 & Age <= 77,
                                 "68-77", "Not seen"))))),
           ChestPainType = factor(ChestPainType,
                      levels = c("TA", "ATA", "NAP", "ASY")))

# Need to edit this?

N <- nrow(heart); p <- ncol(heart %>%dplyr::select(where(is.numeric)));

# number of groups in ChestPainType, k_pain.
k_pain <- n_distinct(heart$ChestPainType)
# number of groups in age, k_age
k_age <- n_distinct(heart$Age)
# number of groups in Heart Disease, k_HD
k_HD <- n_distinct(heart$HeartDisease)

# Combined 5 datasets (Cleveland, Long Beach, Switzerland, Hungarian, &
```

```
Stalog)
skimr::skim(heart)
```

*Data summary*

| | |
|---|---|
| Name | heart |
| Number of rows | 918 |
| Number of columns | 12 |

_____

| | |
|---|---|
| Column type frequency: | |
| character | 6 |
| factor | 1 |
| numeric | 5 |

_____

| | |
|---|---|
| Group variables | None |

**Variable type: character**

| skim_variable | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|---|---|---|---|---|---|---|---|
| Age | 0 | 1 | 5 | 5 | 0 | 5 | 0 |
| Sex | 0 | 1 | 1 | 1 | 0 | 2 | 0 |
| ChestPainType | 0 | 1 | 2 | 3 | 0 | 4 | 0 |
| RestingECG | 0 | 1 | 2 | 6 | 0 | 3 | 0 |
| ExerciseAngina | 0 | 1 | 1 | 1 | 0 | 2 | 0 |
| ST_Slope | 0 | 1 | 2 | 4 | 0 | 3 | 0 |

**Variable type: factor**

| skim_variable | n_missing | complete_rate | ordered | n_unique | top_counts |
|---|---|---|---|---|---|
| HeartDisease | 0 | 1 | FALSE | 2 | Aff: 508, Una: 410 |

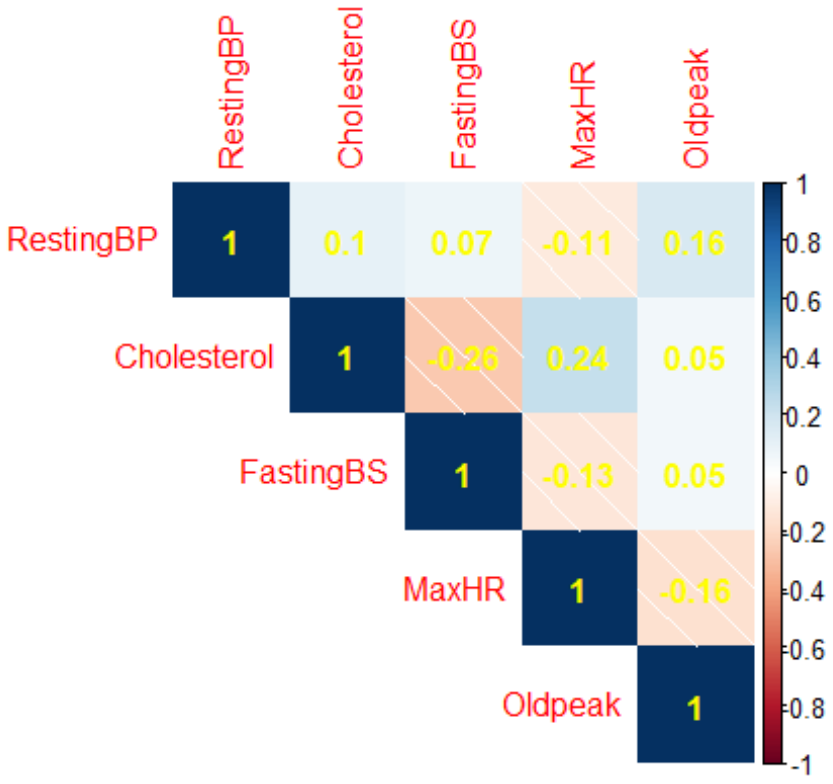
**Variable type: numeric**

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| RestingBP | 0 | 1 | 132.40 | 18.51 | 0.0 | 120.00 | 130.0 | 140.0 | 200.0 | ▁▁▂▇▁ |
| Cholesterol | 0 | 1 | 198.80 | 109.38 | 0.0 | 173.25 | 223.0 | 267.0 | 603.0 | ▂▇▁ |
| FastingBS | 0 | 1 | 0.23 | 0.42 | 0.0 | 0.00 | 0.0 | 0.0 | 1.0 | ▇▁▁▁▂ |
| MaxHR | 0 | 1 | 136. | 25.4 | 60. | 120. | 138. | 156. | 202. | ▁▂▅▇ |

| skim_variable | n_missing | complete_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 81 | 6 | 0 | 00 | 0 | 0 | 0 | ▂ |
| Oldpeak | 0 | 1 | 0.89 | 1.07 | -2.6 | 0.00 | 0.6 | 1.5 | 6.2 | ▁▆▂▁ |

### Correlation Plot of Numeric Variables

```
R <- cor(heart %>%dplyr::select(where(is.numeric)))

corrplot(R,
        method="shade",
        type="upper",
        addCoef.col = "yellow")
```

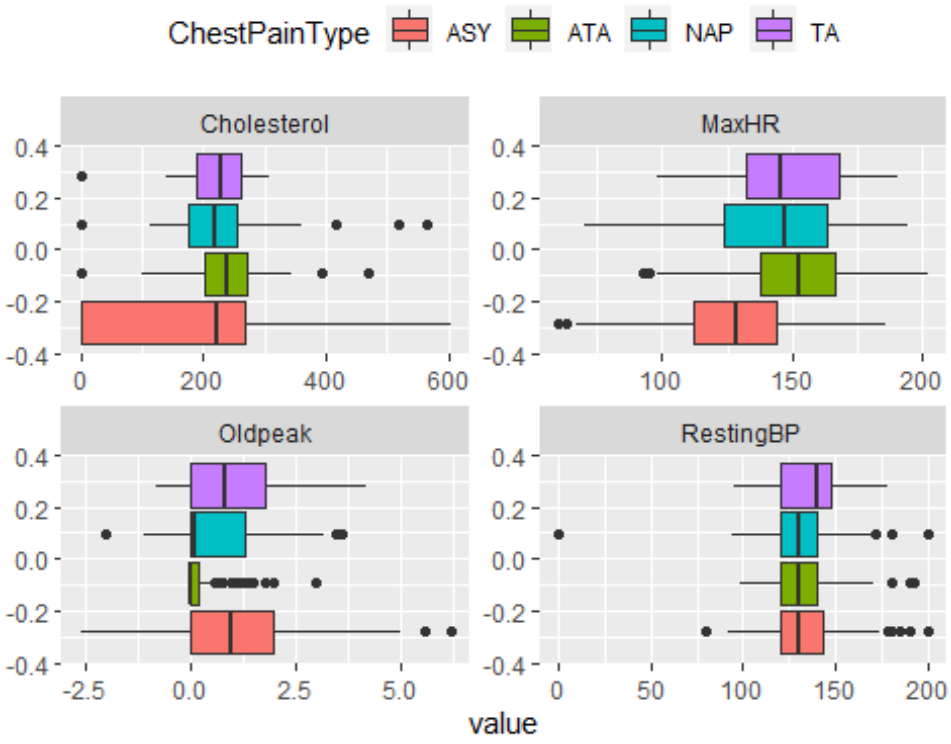



```
table(heart$Sex)

##
##   F   M
## 193 725
```

Variances of the numeric variables: RestingBP - 342.7739 Cholesterol - 11964.89 MaxHR - 648.2286 Oldpeak - 1.137572 As shown by the correlation plot above of the numeric variables in our data set, there does not appear to be any high correlations between variables.

### A Look at Chest Pain Type Boxplots

```
heart %>%
  pivot_longer(cols = c(RestingBP, Cholesterol, Oldpeak, MaxHR),
               names_to = "attribute",
               values_to = "value") %>%

  ggplot(mapping = aes(x = value,
                       fill = ChestPainType)) +
  geom_boxplot() +
  facet_wrap(facets = ~ attribute,
             scales = "free") +
  labs(fill = "ChestPainType") +
  theme(legend.position = "top")
```

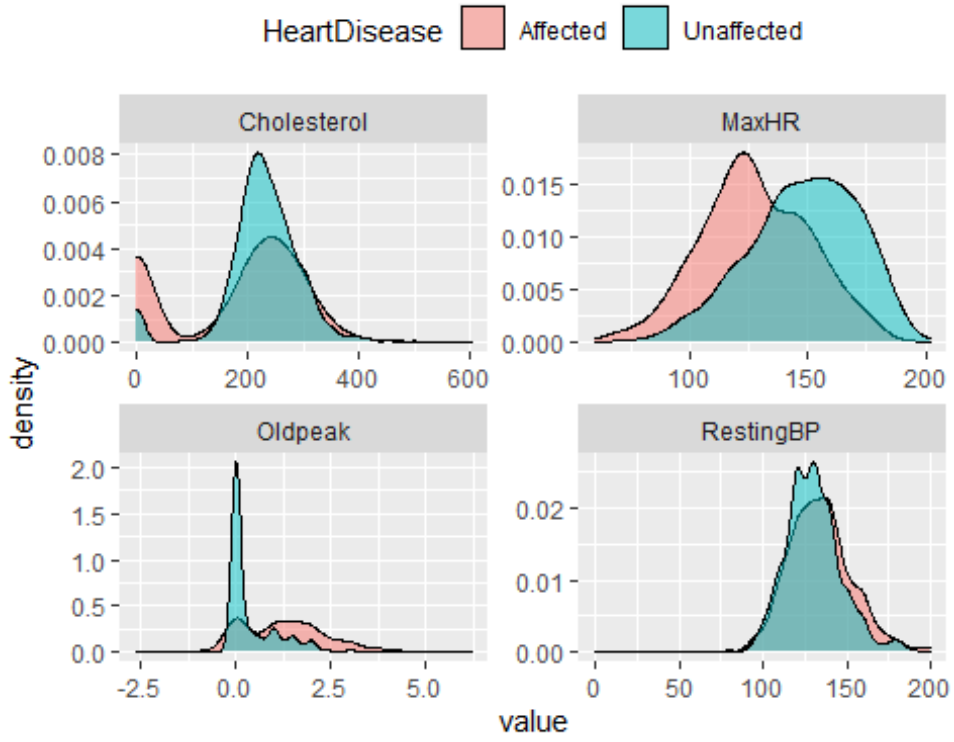



Above is a set of box plots showing the distribution of the four chest pain types in each of the 5 numeric variables in our data set. The chest pain types appear relatively equal across Cholesterol and RestingBP, while MaxHR is noticeably lower for those with ASY, and both ASY and TA are noticeably higher in Oldpeak.

### Density Plots of Heart Disease

```
heart %>%
  pivot_longer(cols = c(RestingBP, Cholesterol, Oldpeak, MaxHR),
               names_to = "attribute",
               values_to = "value") %>%

  ggplot(mapping = aes(x = value,
```

```
                      fill = HeartDisease)) +
  geom_density(alpha = .5) +
  facet_wrap(facets = ~ attribute,
             scales = "free") +
  labs(fill = "HeartDisease") +
  theme(legend.position = "top")
```



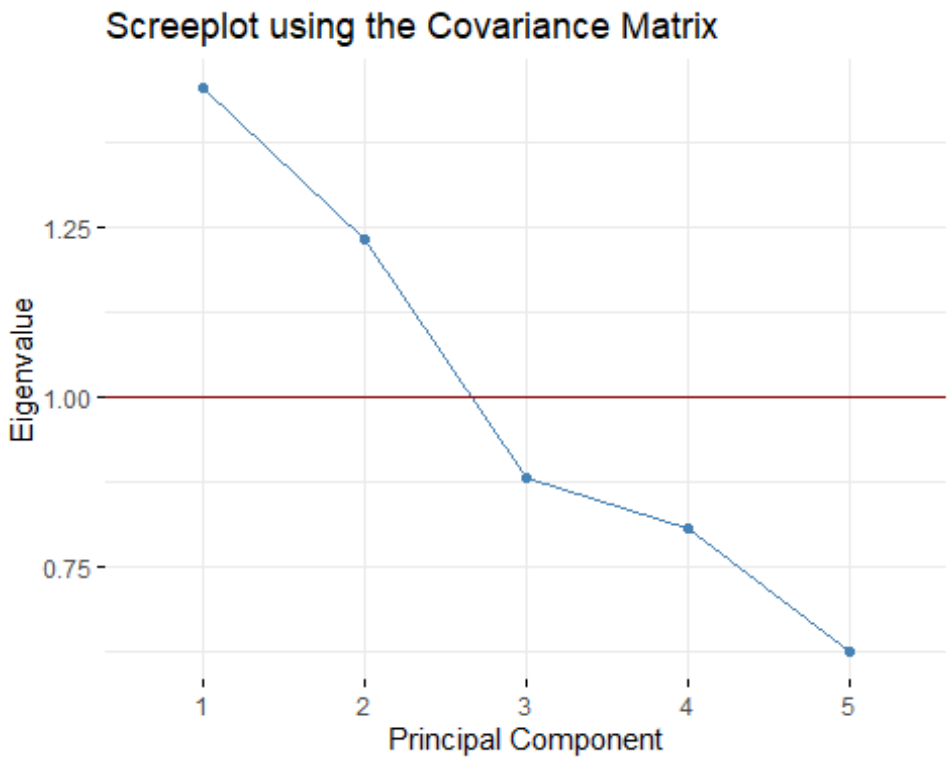

Also shown above is a set of density plots showing the distribution of those affected or unaffected by heart disease in each of the 5 numeric variables in our data set. It appears that MaxHR has a higher median for those without heart disease when compared to those with heart disease. It appears that Oldpeak and RestingBP have slightly higher medians for those with heart disease when compared to those without heart disease. Cholesterol level appears to be relatively equal between the two.

### Check some group means

```
HD_means <-
  heart %>%
  group_by(HeartDisease, Age) %>%
  summarize(across(.cols = c(Cholesterol, Oldpeak, MaxHR),
                   .fns = mean))

## `summarise()` has grouped output by 'HeartDisease'. You can override using
the
## `.groups` argument.

view(HD_means)
```

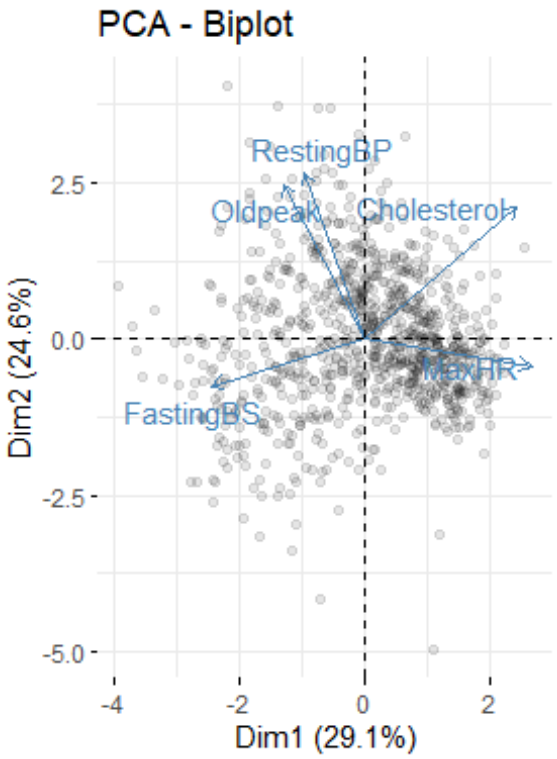
## PCA to Check Significance of Variables

```
(heart_R_PCA <- prcomp(heart %>% dplyr::select(where(is.numeric)),
                       scale. = T))

## Standard deviations (1, .., p=5):
## [1] 1.2065604 1.1097962 0.9388502 0.8981496 0.7902227
##
## Rotation (n x k) = (5 x 5):
##                     PC1        PC2         PC3         PC4        PC5
## RestingBP   -0.2081438  0.6200490 -0.50283439  0.47498599 -0.3062048
## Cholesterol  0.5213448  0.4907686 -0.09691372 -0.07149774  0.6876348
## FastingBS   -0.5280012 -0.1762990 -0.61339632 -0.39352023  0.3987733
## MaxHR        0.5721234 -0.1010971 -0.48932536 -0.44257379 -0.4765955
## Oldpeak     -0.2806515  0.5773937  0.34938655 -0.64695182 -0.2173330

summary(heart_R_PCA)

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5
## Standard deviation     1.2066 1.1098 0.9389 0.8981 0.7902
## Proportion of Variance 0.2912 0.2463 0.1763 0.1613 0.1249
## Cumulative Proportion  0.2912 0.5375 0.7138 0.8751 1.0000

fviz_screeplot(X = heart_R_PCA,
               choice = "eigenvalue",
               geom = "line",
               linecolor = "steelblue",
               ncp = p) +

  labs(title = "Screeplot using the Covariance Matrix",
       x = "Principal Component") +

  geom_hline(yintercept = 1,
             color = "darkred")
```




### Correlation Matrix PCA Biplot

```
fviz_pca(X = heart_R_PCA,
         axes = c(1, 2),                # Which PCs to plot
         geom = c("point"),
         alpha.ind = .1,
         repel = T)  +    # text adds name of country.

  coord_equal()
```




We used PCA to check variable dependencies, as well as significance of the variables. To no surprise, PCA wasn't super useful because there wasn't much collinearity between the numeric variables (as shown in the correlation matrix). This is shown in the screeplot, because the first two PC's only account for around 55%, and ⅘ of the PC's would get us to ~88% of the proportion covered. The biplot also shows this because the direction of the vector's do not overlap - they point in mostly different directions.

### Check Differences Using MANOVA

We want to create a MANOVA model to check if there is a difference in mean chest pain type between predictor variables. Our null hypothesis is that there is no difference in mean chest pain type between any of the predictor variables while our alternative hypothesis is that there is a difference.

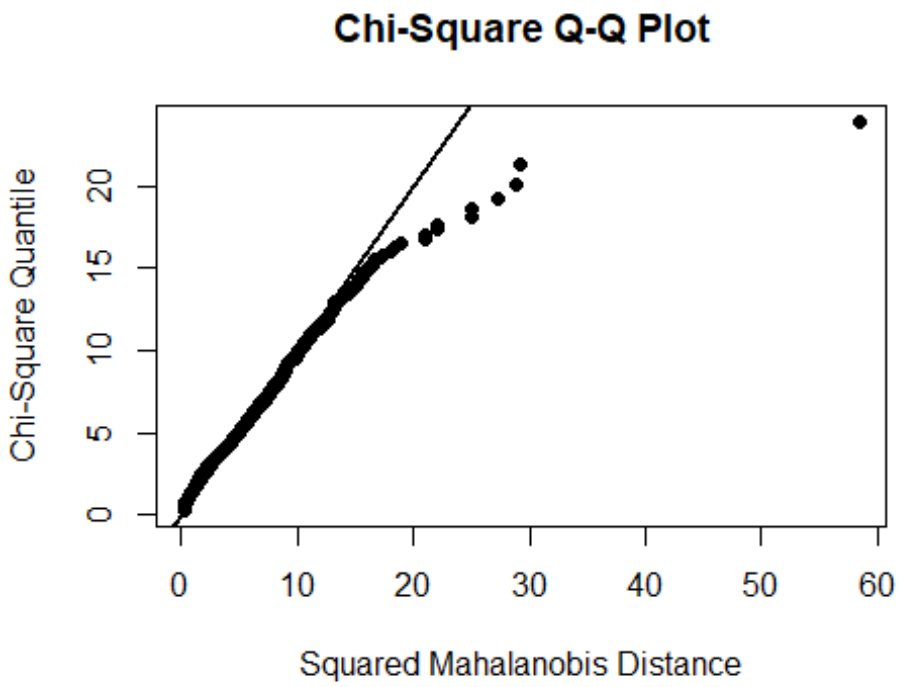
```
heart_man <- manova(cbind(RestingBP, Cholesterol, FastingBS, MaxHR, Oldpeak,
HeartDisease) ~ ChestPainType,
    data = heart)

summary(heart_man)
```

```
##                Df Pillai approx F num Df den Df    Pr(>F)
## ChestPainType   3 0.3506   20.092     18   2733 < 2.2e-16 ***
## Residuals     914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on our test statistic which is very close to zero, we can conclude that there is a difference in mean chest pain type between at least one pair of predictor variables.

### Check Assumptions

```
# Not normal
mvn(data = heart_man$residuals,
    desc = F,
    multivariatePlot = "qq",
    univariateTest = "SW",
    mvnTest = "mardia")
```




```
## $multivariateNormality
##               Test        Statistic               p value Result
## 1 Mardia Skewness 589.108111990132 5.60870103553041e-90     NO
## 2 Mardia Kurtosis 10.9939764879178                     0     NO
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##           Test     Variable Statistic   p value Normality
## 1 Shapiro-Wilk  RestingBP      0.9640  <0.001      NO
## 2 Shapiro-Wilk Cholesterol     0.9037  <0.001      NO
## 3 Shapiro-Wilk  FastingBS      0.6462  <0.001      NO
## 4 Shapiro-Wilk    MaxHR        0.9950  0.0041      NO
## 5 Shapiro-Wilk   Oldpeak       0.9354  <0.001      NO
## 6 Shapiro-Wilk HeartDisease    0.9078  <0.001      NO
```

```
box_m(data =  heart[, c(4, 5, 6, 8, 10)],
      group = heart$ChestPainType)
```

```
## # A tibble: 1 × 4
##   statistic  p.value parameter method
##       <dbl>    <dbl>     <dbl> <chr>
## 1      214. 8.59e-24        45 Box's M-test for Homogeneity of Covariance
Matri…
```

Checking assumptions: The first assumption checked was to see if the data is multivariate normal. After performing a test for mardia skewness and mardia kurtosis, it appears that the data is not multivariate normal as the test for normality gave a p-value of 3.997e-82 for mardia skewness and a p-value of 0 for mardia kurtosis. As shown by the QQ plot below, there is evidence of skewness as well.

**Let's see what's actually useful:**

```
Partial_F(Y = heart_CP %>%
              dplyr::select(FastingBS, RestingBP, Cholesterol, Oldpeak,
MaxHR, HeartDisease),
          x = heart_CP$ChestPainType)
```

```
##              Partial_Test     F_stat P_value
## FastingBS       0.6584917  0.7256171  0.5368
## RestingBP       0.6594567  1.1707212  0.3198
## Cholesterol     0.6606126  1.7038807  0.1646
## Oldpeak         0.6719081  6.9138243  0.0001
## MaxHR           0.6846037 12.7696034  0.0000
## HeartDisease    0.7735874 53.8128436  0.0000
```

```
# ---------------------------------------- #
Partial_F(Y = heart_CP %>%
              dplyr::select(RestingBP, Cholesterol, Oldpeak, MaxHR,
HeartDisease),
          x = heart_CP$ChestPainType)
```

```
##              Partial_Test    F_stat P_value
## RestingBP       0.6612433  1.267504  0.2843
## Cholesterol     0.6630763  2.111879  0.0972
## Oldpeak         0.6732557  6.801000  0.0002
## MaxHR           0.6862712 12.796601  0.0000
## HeartDisease    0.7814023 56.618608  0.0000
```

```
# ---------------------------------------- #

Partial_F(Y = heart_CP %>%
              dplyr::select(Cholesterol, Oldpeak, MaxHR, HeartDisease),
          x = heart_CP$ChestPainType)
```

```
##              Partial_Test    F_stat P_value
## Cholesterol     0.6658167  2.100288  0.0986
## Oldpeak         0.6762186  6.877206  0.0001
```

```
## MaxHR           0.6880401 12.306053  0.0000
## HeartDisease    0.7840135 56.380493  0.0000

# ---------------------------------------- #



Partial_F(Y = heart_CP %>%
              dplyr::select(Oldpeak, MaxHR, HeartDisease),
          x = heart_CP$ChestPainType)

##              Partial_Test    F_stat P_value
## Oldpeak         0.6799320  6.444781   3e-04
## MaxHR           0.6935460 12.660708   0e+00
## HeartDisease    0.7959784 59.429485   0e+00

# ---------------------------------------- #


Partial_F(Y = heart_CP %>%
              dplyr::select(MaxHR, HeartDisease),
          x = heart_CP$ChestPainType)

##              Partial_Test   F_stat P_value
## HeartDisease     47.05307 20756.33       0
## MaxHR           125.66093 55940.72       0

heart_man <- manova(cbind(MaxHR, HeartDisease) ~ ChestPainType,
    data = heart_CP)

summary(heart_man)

##                Df  Pillai approx F num Df den Df    Pr(>F)
## ChestPainType   3 0.32149   58.354      6   1828 < 2.2e-16 ***
## Residuals     914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# ---------------------------------------- #

# --------------CHECKING HEART DISEASE AS OUTCOME -----------------#

# ---------------------------------------- #
Partial_F(Y = heart %>%
              dplyr::select(RestingBP, Cholesterol, Oldpeak, MaxHR),
          x = heart$HeartDisease)

##             Partial_Test     F_stat P_value
## RestingBP      0.6907685   1.434188  0.2314
## Cholesterol    0.7211359  41.634334  0.0000
```

```
## MaxHR          0.7683385 104.120706  0.0000
## Oldpeak        0.8125844 162.693164  0.0000

# ---------------------------------------- #
Partial_F(Y = heart %>%
              dplyr::select(Cholesterol, Oldpeak, MaxHR),
          x = heart$HeartDisease)

##             Partial_Test    F_stat P_value
## Cholesterol    0.7212734  40.36289       0
## MaxHR          0.7726544 108.34848       0
## Oldpeak        0.8194031 170.20465       0

# Stratify by ChestPain Type:
heart_man <- manova(cbind(Cholesterol, Oldpeak, MaxHR) ~ ChestPainType +
HeartDisease,
    data = heart)

summary(heart_man)

##                Df  Pillai approx F num Df den Df    Pr(>F)
## ChestPainType   3 0.25093   27.779      9   2739 < 2.2e-16 ***
## HeartDisease    1 0.17711   65.356      3    911 < 2.2e-16 ***
## Residuals     913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#Age, Sex, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR,
ExerciseAngina, Oldpeak, ST_Slope, HeartDisease
```

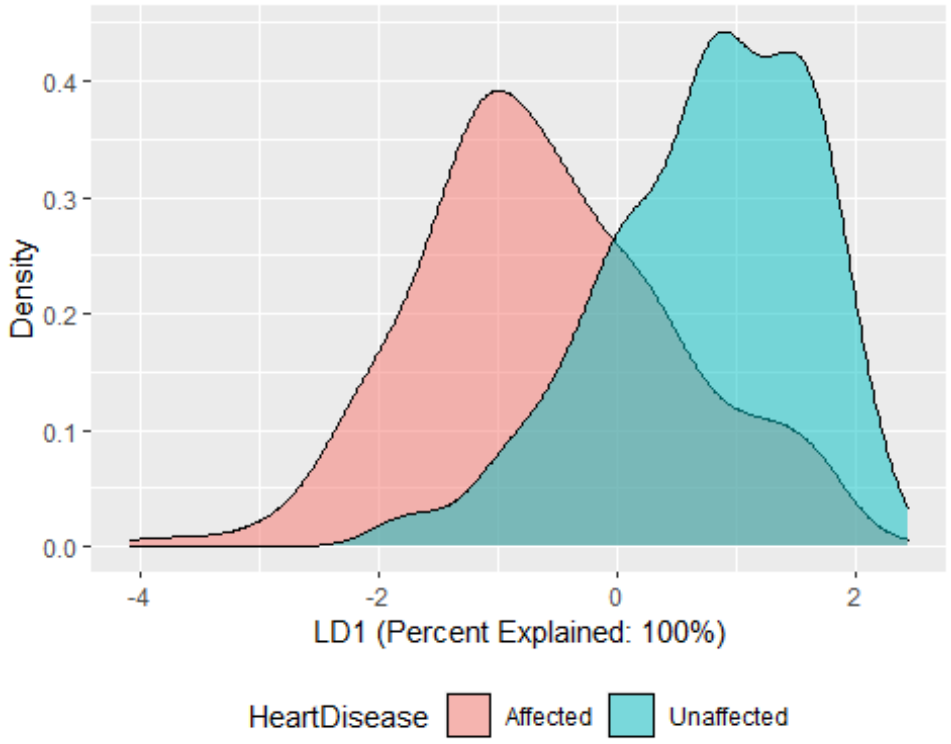
Through running partial f tests and removing our insignificant variables we found that Oldpeak, MaxHR, HeartDisease are the only variables we want to keep when predicting chest pain type. These are the only variables that contribute unique information and are important predictors. As for predicting heart disease, we found that MaxHR and Oldpeak were useful predictors with the addition of Cholesterol.

### Linear Discriminant Analysis

```
# ------------ Plot the discriminant for HEART DISEASE ------------------

heart_HD_lda <- MASS::lda(HeartDisease ~ cbind(Cholesterol, Oldpeak, MaxHR),
                     data = heart)

ld_sep_pct <- round(heart_HD_lda$svd^2/sum(heart_HD_lda$svd^2)*100,
                    digits = 1)

heart_HD <-
  data.frame(heart,
             predict(heart_HD_lda)$x)

heart_HD_lda$scaling
```

```
##                                                          LD1
## cbind(Cholesterol, Oldpeak, MaxHR)Cholesterol  0.003568154
## cbind(Cholesterol, Oldpeak, MaxHR)Oldpeak     -0.739881610
## cbind(Cholesterol, Oldpeak, MaxHR)MaxHR        0.025414879
```

```r
gg_lda_density <-
  heart_HD %>%
  ggplot(mapping = aes(x = LD1,
                       fill = HeartDisease)) +

  theme(legend.position = "bottom") +

  labs(x = paste0("LD1 (Percent Explained: ", ld_sep_pct[1], "%)"),
       y = paste0("Density"))

gg_lda_density +
  geom_density(alpha = .5)
```

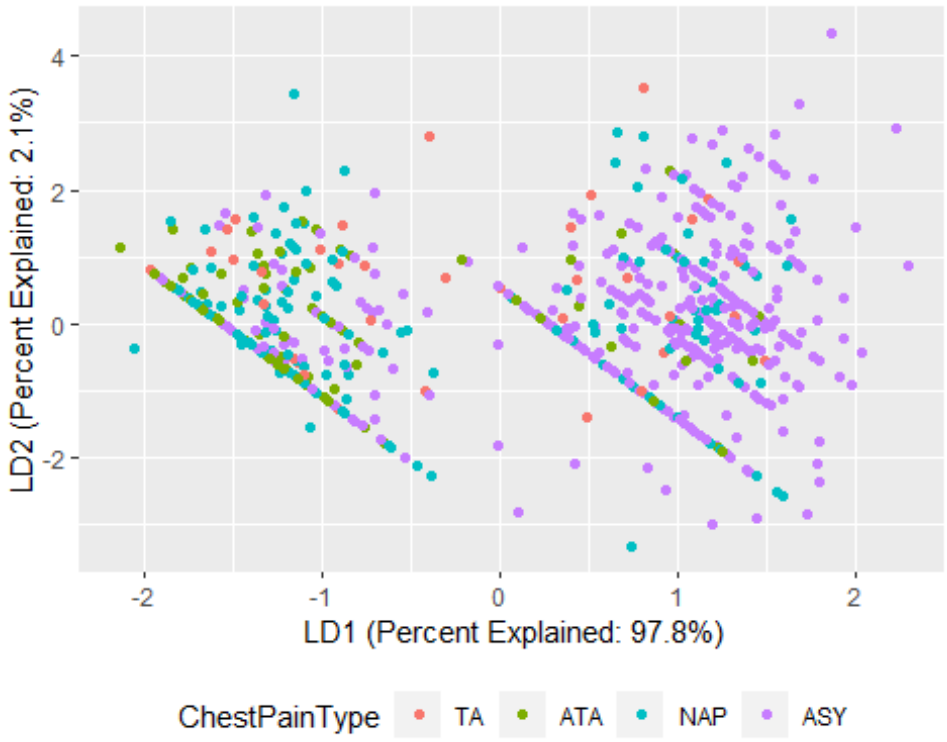



```r
# ------------------ Plot Discrimant for CHEST PAIN --------------------
heart_CP_lda <- MASS::lda(ChestPainType ~ cbind(Oldpeak, MaxHR,
HeartDisease),
                       data = heart_CP)

ld_sep_pct <- round(heart_CP_lda$svd^2/sum(heart_CP_lda$svd^2)*100,
                    digits = 1)

heart_CPLDA <-
```

```
  data.frame(heart_CP,
             predict(heart_CP_lda)$x)


heart_CP_lda$scaling
```

```
##                                                          LD1         LD2
## cbind(Oldpeak, MaxHR, HeartDisease)Oldpeak       0.21976240 0.776235286
## cbind(Oldpeak, MaxHR, HeartDisease)MaxHR        -0.01439442 0.028101296
## cbind(Oldpeak, MaxHR, HeartDisease)HeartDisease  1.82646534 0.004871112
##                                                          LD3
## cbind(Oldpeak, MaxHR, HeartDisease)Oldpeak      -0.64684148
## cbind(Oldpeak, MaxHR, HeartDisease)MaxHR         0.03017646
## cbind(Oldpeak, MaxHR, HeartDisease)HeartDisease  1.82786822
```

```
gg_lda_scatter_CP <-
  heart_CPLDA %>%
  ggplot(mapping = aes(x = LD1,
                       y = LD2,
                       color = ChestPainType)) +

  theme(legend.position = "bottom") +

  labs(x = paste0("LD1 (Percent Explained: ", ld_sep_pct[1], "%)"),
       y = paste0("LD2 (Percent Explained: ", ld_sep_pct[2], "%)"))

gg_lda_scatter_CP +
  geom_point()
```



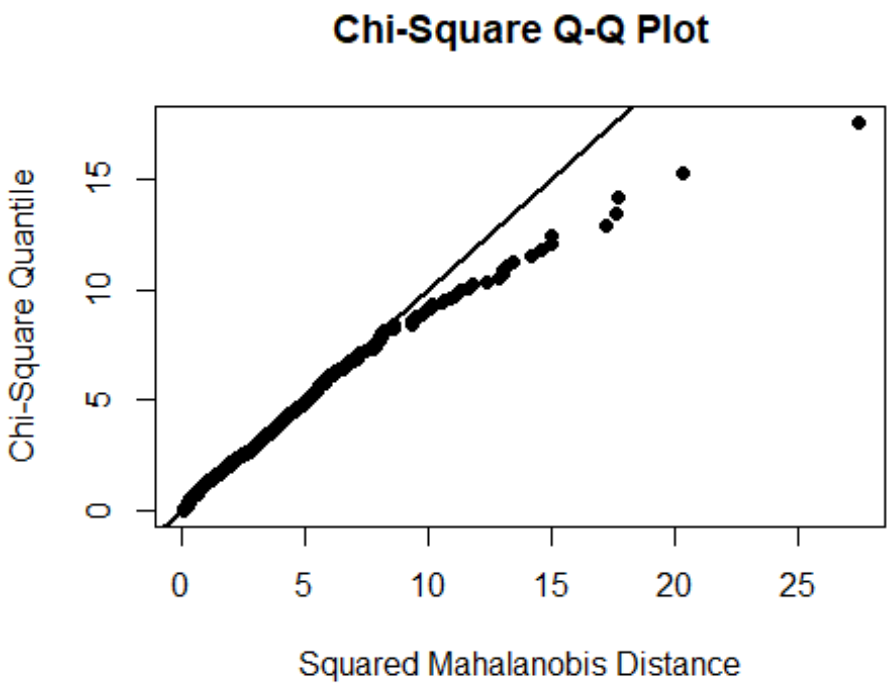

First, LDA was performed based on heart disease status and chest pain type. As shown in the first graph below, the data is fairly well separated by the first linear discriminant based on those affected or unaffected by heart disease. As shown in the second graph below, the four types of chest pain are not very well separated by LD1 and LD2.

## Predicting Heart Disease

With our initial set up and data exploration complete, we are ready to move on to our methods. First, we decided to try to predict heart disease (affected or unaffected) using our set of predictor variables. We used QDA, KNN, and a classification tree to carry out these predictions.

### QDA: Predicting Heart Disease

```
# Using best model:
heart_man_HD <- manova(cbind(Cholesterol, Oldpeak, MaxHR) ~ HeartDisease,
    data = heart)


# Not normal
mvn(data = heart_man$residuals,
    desc = F,
    multivariatePlot = "qq",
    univariateTest = "SW",
    mvnTest = "mardia")
```




```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 148.731699425912 6.79634995118434e-27     NO
## 2 Mardia Kurtosis 5.96290087467747 2.47798759289708e-09     NO
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##           Test    Variable Statistic   p value Normality
## 1 Shapiro-Wilk Cholesterol    0.9348  <0.001      NO
## 2 Shapiro-Wilk   Oldpeak      0.9515  <0.001      NO
## 3 Shapiro-Wilk    MaxHR       0.9951  0.0046      NO

box_m(data =  heart[, c("Cholesterol", "Oldpeak", "MaxHR")],
      group = heart$HeartDisease)

## # A tibble: 1 × 4
##   statistic  p.value parameter method
##       <dbl>    <dbl>     <dbl> <chr>
## 1      219. 1.66e-44         6 Box's M-test for Homogeneity of Covariance
Matri…
```

To continue with discriminant analysis, a box's m test was performed to test for equal covariance matrices (as explained in the descriptive statistics section above). After rejecting the null hypothesis, Quadratic Discriminant Analysis for both heart disease and chest pain type was carried out.

```
# Not normal and reject box_m test:
qda_heart_HD_cv <- MASS::qda(formula = HeartDisease~ cbind(Cholesterol,
Oldpeak, MaxHR),
                      data = heart,
                      CV = T)

# Confusion Matrix
table(predicted = qda_heart_HD_cv$class,
      actual = heart$HeartDisease) %>%
  confusionMatrix()
```

```
## Confusion Matrix and Statistics
##
##             actual
## predicted    Affected Unaffected
##   Affected        379         84
##   Unaffected      129        326
##
##                Accuracy : 0.768
##                  95% CI : (0.7393, 0.7949)
##     No Information Rate : 0.5534
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5355
##
##  Mcnemar's Test P-Value : 0.002571
##
##             Sensitivity : 0.7461
##             Specificity : 0.7951
##          Pos Pred Value : 0.8186
##          Neg Pred Value : 0.7165
##              Prevalence : 0.5534
##          Detection Rate : 0.4129
##    Detection Prevalence : 0.5044
##       Balanced Accuracy : 0.7706
##
##        'Positive' Class : Affected
##
```

When predicting heart disease, QDA performed fairly well, achieving an accuracy score of about 76.8%.

```
# Find the pooled standard deviations:
sd_heart_HD <-
  summary(heart_man)$SS$Residuals %>%
  diag() %>%
  sqrt()/sqrt(N-k_HD)
```

```r
# Standardize the data using the pooled standard deviations:


# Now we need to divide each variable by the pooled sd:
heart_sc_HD <-
  scale(heart[, c("Cholesterol", "Oldpeak", "MaxHR")],
        center = T,
        scale = sd_heart_HD) %>%
  data.frame()


heart_sc_HD$HeartDisease <- heart$HeartDisease
```

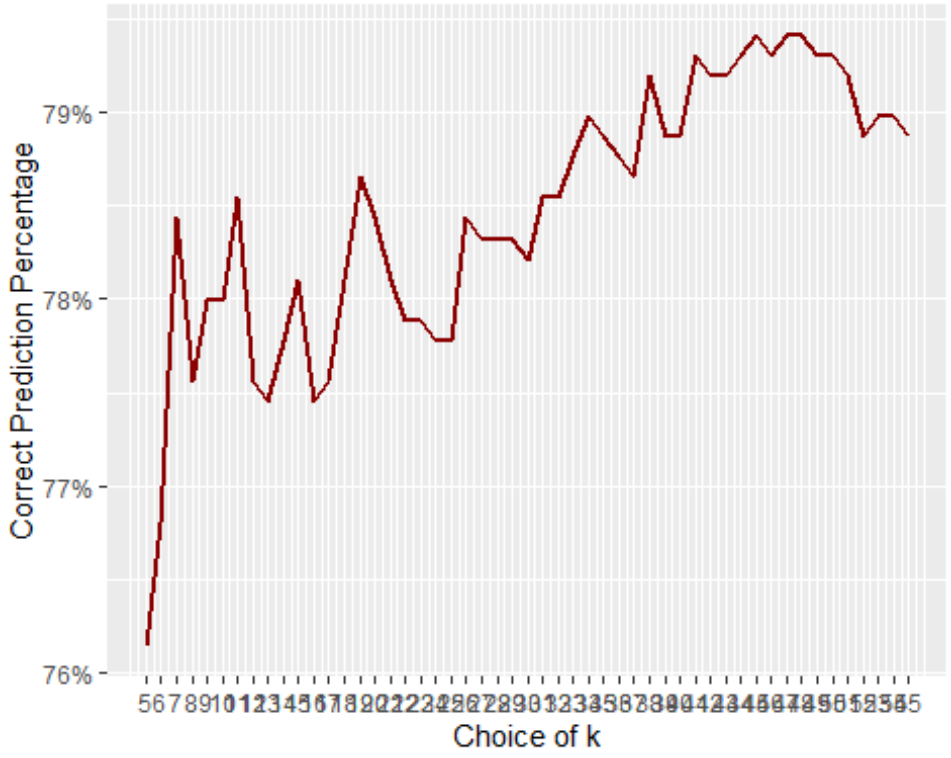
### KNN Classification: Predicting Heart Disease

```r
## Creating a loop to find the best choice for k
RNGversion("4.0.0")
set.seed(123)

# ---------------------- HEART DISEASE ---------------------- #
sqrt(N/k_HD)
```

```
## [1] 21.42429
```

```r
k_choice <-5:55

# data.frame to store the predictions for different choices of k
knn_predictions <- data.frame(Actual = heart$HeartDisease)


# Function knn.cv() performs KNN using cross-validation
# and returns the predicted class based on the nearest neighbors.

# Looping through the different choices of k for knn
for (i in k_choice){

  knn_temp <- class::knn.cv(train = heart_sc_HD %>% dplyr::select(-
HeartDisease),
                            cl = heart_sc_HD$HeartDisease,
                            k = i)

  # adding the predicted column to the data set
  knn_predictions <-
    knn_predictions %>%
    add_column(knn_temp)
}

# Changing the column names to better describe the results
colnames(knn_predictions) <- c('Actual', paste0("k", k_choice))
```

```r
# Calculating the error rate for each choice of k:
knn_predictions %>%
  pivot_longer(cols = starts_with("k"),
               names_to = "k_choice",
               values_to = "prediction") %>%
  group_by(k_choice) %>%
  summarize(incorrect = sum(Actual != prediction),
            positive_rate = mean(Actual == prediction)) %>%
  mutate(k = parse_number(k_choice)) %>%

  ggplot(mapping = aes(x = k,
                       y = positive_rate)) +
  geom_line(color = "darkred",
            size = 1) +

  labs(x = "Choice of k",
       y = "Correct Prediction Percentage") +

  scale_x_continuous(breaks = k_choice) +
  scale_y_continuous(labels = scales::percent)
```




```r
# ---------------------- HEART DISEASE ---------------------- #

# Best choice of kNN model
heart_knn <- knn.cv(train = heart_sc_HD %>% dplyr::select(-HeartDisease),
                            cl = heart_sc_HD$HeartDisease,
```

```
                          k = 47)

# Confusion matrix
data.frame(actual = heart$HeartDisease,
           predicted = heart_knn) %>%
  table() %>%
  confusionMatrix()
```

```
## Confusion Matrix and Statistics
##
##             predicted
## actual       Affected Unaffected
##   Affected        396        112
##   Unaffected       77        333
##
##                Accuracy : 0.7941
##                  95% CI : (0.7665, 0.8198)
##     No Information Rate : 0.5153
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.5869
##
##  Mcnemar's Test P-Value : 0.01339
##
##             Sensitivity : 0.8372
##             Specificity : 0.7483
##          Pos Pred Value : 0.7795
##          Neg Pred Value : 0.8122
##              Prevalence : 0.5153
##          Detection Rate : 0.4314
##    Detection Prevalence : 0.5534
##       Balanced Accuracy : 0.7928
##
##        'Positive' Class : Affected
##
```

The next algorithm used was KNN, where the choices for k were looped through to find the ideal choice when carrying out the algorithm. K = 47 was determined to be the best choice for predicting heart disease as it yielded the highest accuracy rate. The KNN algorithm performed relatively well when predicting heart disease status with an accuracy score of 79.41%.

### Classification Tree: Predicting Heart Disease

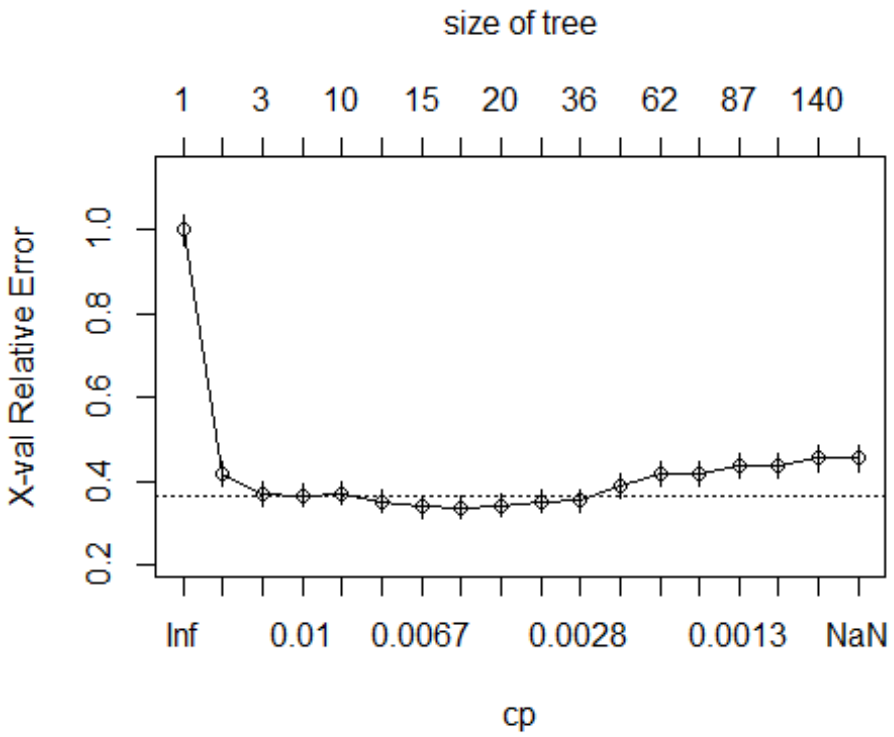
```
# Include the two lines below at the top of the R code to ensure your answer
matches the solutions
RNGversion("4.0.0")
set.seed(123)


# Create the full classification tree
```

```r
heart_tree2 <- rpart(HeartDisease ~ .- ChestPainType,
                     data = heart,
                     minsplit = 2,
                     minbucket = 1,
                     cp = -1,
                     method = "class")


# Looking at the cp table to find the optimal pruning value:
# simplest tree where xerror < min(xerror) + min(xstd)
printcp(heart_tree2)
```

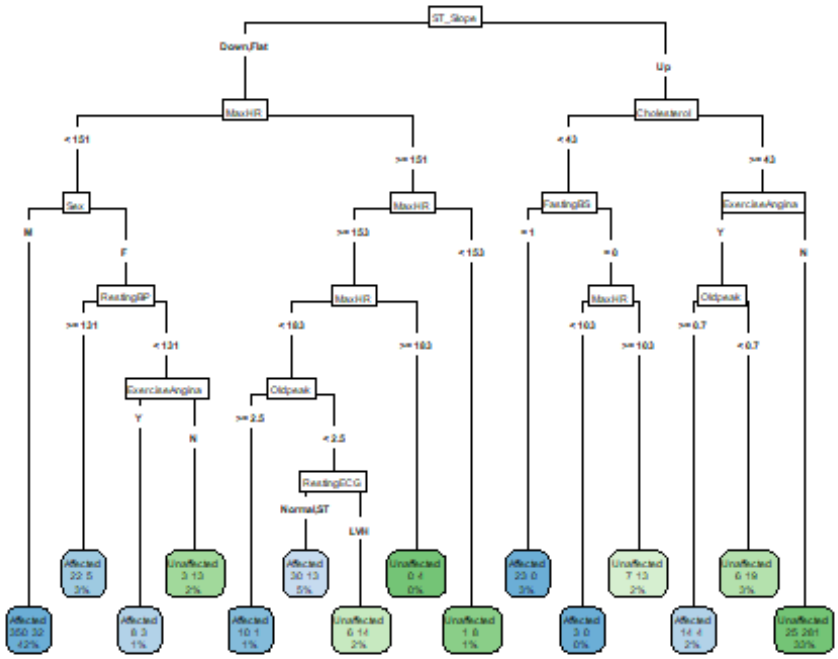
```
##
## Classification tree:
## rpart(formula = HeartDisease ~ . - ChestPainType, data = heart,
##     method = "class", minsplit = 2, minbucket = 1, cp = -1)
##
## Variables actually used in tree construction:
##  [1] Age            Cholesterol    ExerciseAngina FastingBS      MaxHR
##  [6] Oldpeak        RestingBP      RestingECG     Sex            ST_Slope
##
## Root node error: 410/918 = 0.44662
##
## n= 918
##
##             CP nsplit rel error  xerror     xstd
## 1   0.58292683      0  1.000000 1.00000 0.036738
## 2   0.04878049      1  0.417073 0.41707 0.028771
## 3   0.01219512      2  0.368293 0.36829 0.027396
## 4   0.00853659      4  0.343902 0.36585 0.027323
## 5   0.00813008      9  0.297561 0.37073 0.027468
## 6   0.00731707     12  0.273171 0.35122 0.026875
## 7   0.00609756     14  0.258537 0.33902 0.026489
## 8   0.00487805     16  0.246341 0.33659 0.026411
## 9   0.00365854     19  0.231707 0.34146 0.026567
## 10  0.00325203     26  0.204878 0.35122 0.026875
## 11  0.00243902     35  0.173171 0.35366 0.026951
## 12  0.00182927     55  0.124390 0.38780 0.027965
## 13  0.00162602     61  0.112195 0.41707 0.028771
## 14  0.00139373     78  0.080488 0.41707 0.028771
## 15  0.00121951     86  0.068293 0.43659 0.029278
## 16  0.00097561    124  0.021951 0.43659 0.029278
## 17  0.00081301    139  0.002439 0.45366 0.029703
## 18 -1.00000000    142  0.000000 0.45366 0.029703
```

```r
plotcp(heart_tree2)
```

```
## Warning in sqrt(cp0 * c(Inf, cp0[-length(cp0)])): NaNs produced
```




```
# Prune the tree

p_heart_tree2<- prune(heart_tree2, cp= 0.00731707)


# Plot the pruned tree

rpart.plot(p_heart_tree2,
           type=5,
           extra = 101)
```




```
# Display the confusion matrix
pheart_tree_pred2 <- predict(object = p_heart_tree2,
                             newdata = heart,
                             type = 'class')

data.frame(actual = heart$HeartDisease,
           predicted = pheart_tree_pred2) %>%
  table() %>%
  confusionMatrix()
```

```
## Confusion Matrix and Statistics
##
##             predicted
## actual       Affected Unaffected
##   Affected        460         48
##   Unaffected       58        352
##
##                Accuracy : 0.8845
##                  95% CI : (0.8621, 0.9045)
##     No Information Rate : 0.5643
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7659
##
##  Mcnemar's Test P-Value : 0.382
##
##             Sensitivity : 0.8880
```

```
##             Specificity : 0.8800
##          Pos Pred Value : 0.9055
##          Neg Pred Value : 0.8585
##              Prevalence : 0.5643
##          Detection Rate : 0.5011
##    Detection Prevalence : 0.5534
##       Balanced Accuracy : 0.8840
##
##        'Positive' Class : Affected
##
```

The first step in creating a decision tree for predicting heart disease status was creating the tree and determining the best complexity parameter (cp) for the final pruned tree. The ideal value for cp was determined to be 0.00731707:

xerror < min(xerror) + min(xstd)

0.35122 < 0.33659 + 0.026411

0.35122 gives a CP value of 0.00731707

The output of the pruned tree is shown above This tree returned an accuracy score of 88.45%. ST_Slope was the first predictor variable considered. After this the tree considers all other predictor variables in its decisions besides age which is an interesting take away. This means that age on its own is not a very useful variable for predicting if someone has heart disease, according to this model. With about 88% accuracy this is by far our best model for predicting heart disease.

## Predicting Chest Pain Type

After predicting Heart disease with relatively high success, we decided to move on and attempt to predict chest pain type. We used the same three methods: QDA, KNN, and a classification tree.

### QDA: Predicting Chest Pain

```
# Not normal
qda_heart_CP_cv <- MASS::qda(formula = ChestPainType ~ cbind(Oldpeak, MaxHR,
HeartDisease),
                      data = heart_CP,
                      CV = T)

# Confusion Matrix
table(predicted = qda_heart_CP_cv$class,
      actual = heart_CP$ChestPainType) %>%
  confusionMatrix()

## Confusion Matrix and Statistics
##
##          actual
```

```
## predicted  TA ATA NAP ASY
##       TA    0   0   2   1
##       ATA  15 136 103  76
##       NAP   5  12  18   9
##       ASY  26  25  80 410
##
## Overall Statistics
##
##                Accuracy : 0.6144
##                  95% CI : (0.582, 0.646)
##     No Information Rate : 0.5403
##     P-Value [Acc > NIR] : 3.475e-06
##
##                   Kappa : 0.3606
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: TA Class: ATA Class: NAP Class: ASY
## Sensitivity           0.000000     0.7861    0.08867     0.8266
## Specificity           0.996560     0.7396    0.96364     0.6896
## Pos Pred Value        0.000000     0.4121    0.40909     0.7579
## Neg Pred Value        0.949727     0.9371    0.78833     0.7719
## Prevalence            0.050109     0.1885    0.22113     0.5403
## Detection Rate        0.000000     0.1481    0.01961     0.4466
## Detection Prevalence  0.003268     0.3595    0.04793     0.5893
## Balanced Accuracy     0.498280     0.7629    0.52615     0.7581
```

When predicting chest pain type, QDA was not as effective as predicting heart disease, as it achieved an accuracy score of 61.44%. As shown in the confusion matrix, the most commonly misclassified chest pain types were NAP (52.615% balanced accuracy) and TA (49.828% balanced accuracy).

```
# Make Manova model:

heart_man_CP <- manova(cbind(Oldpeak, MaxHR, HeartDisease) ~ ChestPainType,
    data = heart_CP)

# Find the pooled standard deviations:
sd_heart_CP <-
  summary(heart_man_CP)$SS$Residuals %>%
  diag() %>%
  sqrt()/sqrt(N-k_pain)



# Standardize the data using the pooled standard deviations:
```

```
typeof(heart_CP$HeartDisease)
```

```
## [1] "integer"
```

```
# Now we need to divide each variable by the pooled sd:
heart_sc_CP <-
  scale(heart_CP[, c("Oldpeak", "MaxHR", "HeartDisease")],
        center = T,
        scale = sd_heart_CP) %>%
  data.frame()

heart_sc_CP$ChestPainType <- heart_CP$ChestPainType
```

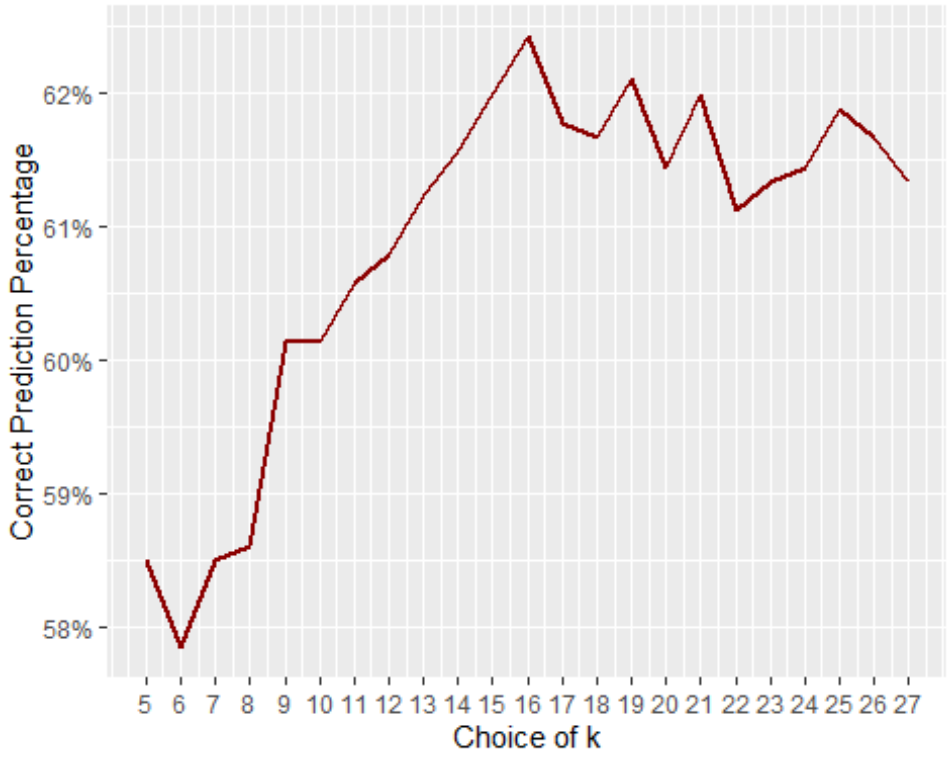
### KNN Classification: Predicting Chest Pain

```
## Creating a loop to find the best choice for k
RNGversion("4.0.0")
set.seed(123)
# ---------------------- CHEST PAIN ---------------------- #
sqrt(N/k_pain)
```

```
## [1] 15.14926
```

```
k_choice <-5:27

# data.frame to store the predictions for different choices of k
knn_predictions <- data.frame(Actual = heart_CP$ChestPainType)


# Function knn.cv() performs KNN using cross-validation
# and returns the predicted class based on the nearest neighbors.

# Looping through the different choices of k for knn
for (i in k_choice){

  knn_temp <- class::knn.cv(train = heart_sc_CP %>% dplyr::select(-
ChestPainType),
                            cl = heart_sc_CP$ChestPainType,
                            k = i)

  # adding the predicted column to the data set
  knn_predictions <-
    knn_predictions %>%
    add_column(knn_temp)
}

# Changing the column names to better describe the results
colnames(knn_predictions) <- c('Actual', paste0("k", k_choice))

# Calculating the error rate for each choice of k:
```

```r
knn_predictions %>%
  pivot_longer(cols = starts_with("k"),
               names_to = "k_choice",
               values_to = "prediction") %>%
  group_by(k_choice) %>%
  summarize(incorrect = sum(Actual != prediction),
            positive_rate = mean(Actual == prediction)) %>%
  mutate(k = parse_number(k_choice)) %>%

  ggplot(mapping = aes(x = k,
                       y = positive_rate)) +
  geom_line(color = "darkred",
            size = 1) +

  labs(x = "Choice of k",
       y = "Correct Prediction Percentage") +

  scale_x_continuous(breaks = k_choice) +
  scale_y_continuous(labels = scales::percent)
```




```r
# ---------------------- CHEST PAIN ---------------------- #
# # Best choice of kNN model
heart_knn <- knn.cv(train = heart_sc_CP%>% dplyr::select(-ChestPainType),
                            cl = heart_sc_CP$ChestPainType,
                            k = 16)
```

```r
# Confusion matrix
data.frame(actual = heart_CP$ChestPainType,
           predicted = heart_knn) %>%
  table() %>%
  confusionMatrix()
```

```
## Confusion Matrix and Statistics
##
##       predicted
## actual  TA ATA NAP ASY
##    TA    0   9   9  28
##    ATA   0 102  29  42
##    NAP   0  73  38  92
##    ASY   0  50  14 432
##
## Overall Statistics
##
##                Accuracy : 0.6231
##                  95% CI : (0.5908, 0.6545)
##     No Information Rate : 0.6471
##     P-Value [Acc > NIR] : 0.9394
##
##                   Kappa : 0.3509
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: TA Class: ATA Class: NAP Class: ASY
## Sensitivity                 NA     0.4359    0.42222     0.7273
## Specificity            0.94989     0.8962    0.80072     0.8025
## Pos Pred Value              NA     0.5896    0.18719     0.8710
## Neg Pred Value              NA     0.8228    0.92727     0.6161
## Prevalence             0.00000     0.2549    0.09804     0.6471
## Detection Rate         0.00000     0.1111    0.04139     0.4706
## Detection Prevalence   0.05011     0.1885    0.22113     0.5403
## Balanced Accuracy           NA     0.6660    0.61147     0.7649
```

When predicting chest pain type, the choices for k were looped through to find the ideal choice when carrying out the algorithm. K = 16 was determined to be the best choice as it yielded the highest accuracy rate. The KNN algorithm performed fairly poorly when predicting chest pain status with an accuracy score of 62.31%.

### Classification Tree: Predicting Chest Pain

```r
# Include the two lines below at the top of the R code to ensure your answer matches the solutions
RNGversion("4.0.0")
set.seed(123)
typeof(heart_CP$HeartDisease)
```

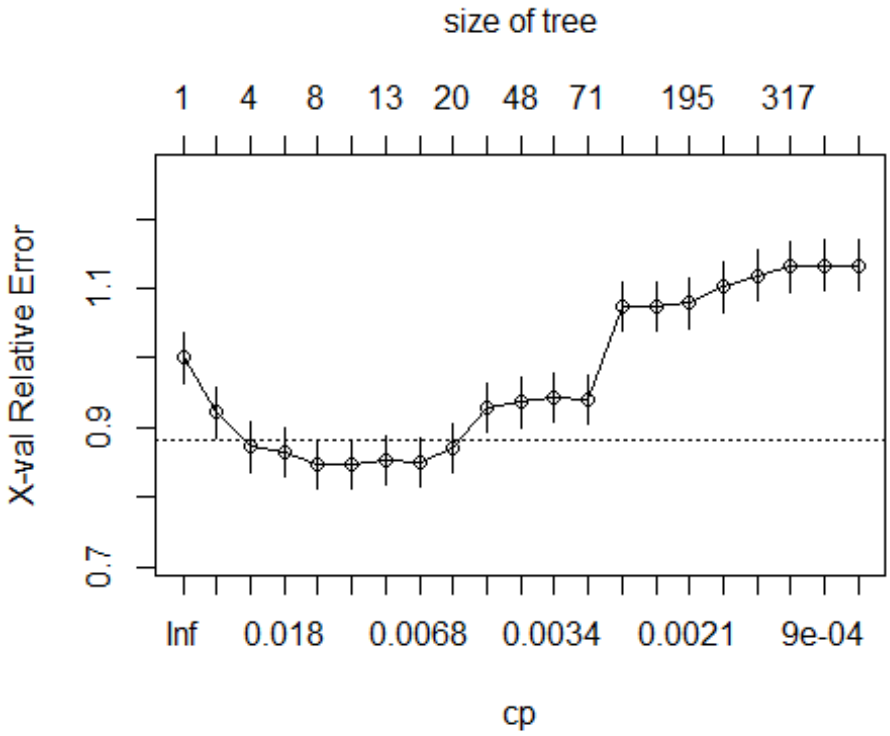
```
## [1] "integer"
```

```
# Create the full classification tree
heart_tree_CP <- rpart(ChestPainType ~ .-HeartDisease,
                    data = heart_CP,
                    minsplit = 2,
                    minbucket = 1,
                    cp = -1,
                    method = "class")


# Looking at the cp table to find the optimal pruning value:
# simplest tree where xerror < min(xerror) + min(xstd)
printcp(heart_tree_CP)
```

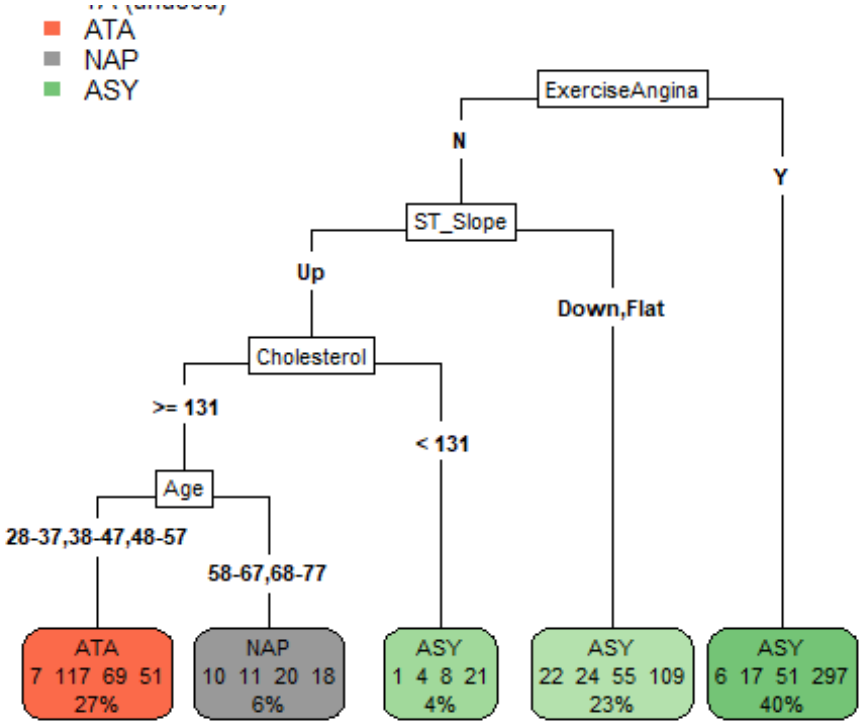
```
##
## Classification tree:
## rpart(formula = ChestPainType ~ . - HeartDisease, data = heart_CP,
##     method = "class", minsplit = 2, minbucket = 1, cp = -1)
##
## Variables actually used in tree construction:
##  [1] Age            Cholesterol    ExerciseAngina FastingBS      MaxHR
##  [6] Oldpeak        RestingBP      RestingECG     Sex            ST_Slope
##
## Root node error: 422/918 = 0.45969
##
## n= 918
##
##             CP nsplit rel error  xerror     xstd
## 1   0.04976303      0  1.000000 1.00000 0.035782
## 2   0.04028436      2  0.900474 0.92180 0.035479
## 3   0.02132701      3  0.860190 0.87204 0.035186
## 4   0.01500790      4  0.838863 0.86493 0.035138
## 5   0.00947867      7  0.793839 0.84597 0.035001
## 6   0.00829384      9  0.774882 0.84597 0.035001
## 7   0.00710900     12  0.748815 0.85308 0.035054
## 8   0.00651659     15  0.727488 0.85071 0.035036
## 9   0.00473934     19  0.701422 0.86967 0.035170
## 10  0.00394945     37  0.616114 0.92891 0.035514
## 11  0.00355450     47  0.575829 0.93602 0.035548
## 12  0.00315956     61  0.523697 0.94313 0.035580
## 13  0.00236967     70  0.495261 0.94076 0.035570
## 14  0.00222156    154  0.296209 1.07346 0.035896
## 15  0.00207346    186  0.210900 1.07346 0.035896
## 16  0.00203114    194  0.194313 1.07820 0.035897
## 17  0.00157978    203  0.175355 1.10190 0.035896
## 18  0.00118483    248  0.099526 1.11848 0.035884
## 19  0.00101557    316  0.018957 1.13033 0.035871
## 20  0.00078989    323  0.011848 1.13270 0.035868
## 21 -1.00000000    335  0.000000 1.13270 0.035868
```

```
plotcp(heart_tree_CP)
```

```
## Warning in sqrt(cp0 * c(Inf, cp0[-length(cp0)])): NaNs produced
```




```
# Prune the tree

p_heart_tree_CP<- prune(heart_tree_CP, cp= 0.02132701)


# Plot the pruned tree

rpart.plot(p_heart_tree_CP,
           type=5,
           extra = 101)
```




```
# Display the confusion matrix
pheart_tree_pred <- predict(object = p_heart_tree_CP,
                            newdata = heart_CP,
                            type = 'class')

data.frame(actual = heart_CP$ChestPainType,
           predicted = pheart_tree_pred) %>%
  table() %>%
  confusionMatrix()

## Confusion Matrix and Statistics
##
##       predicted
## actual  TA ATA NAP ASY
##    TA    0   7  10  29
##    ATA   0 117  11  45
##    NAP   0  69  20 114
##    ASY   0  51  18 427
##
## Overall Statistics
##
##                Accuracy : 0.6144
##                  95% CI : (0.582, 0.646)
##     No Information Rate : 0.6699
##     P-Value [Acc > NIR] : 0.9998
##
##                   Kappa : 0.3279
```

```
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: TA Class: ATA Class: NAP Class: ASY
## Sensitivity                 NA     0.4795    0.33898     0.6943
## Specificity            0.94989     0.9169    0.78696     0.7723
## Pos Pred Value              NA     0.6763    0.09852     0.8609
## Neg Pred Value              NA     0.8295    0.94545     0.5545
## Prevalence             0.00000     0.2658    0.06427     0.6699
## Detection Rate         0.00000     0.1275    0.02179     0.4651
## Detection Prevalence   0.05011     0.1885    0.22113     0.5403
## Balanced Accuracy           NA     0.6982    0.56297     0.7333
```

The ideal value for cp for a decision tree for chest pain was determined to be 0.02132701:

xerror < min(xerror) + min(xstd)

0.87204 < 0.84597 + 0.035001

0.87204 gives a CP value of 0.02132701

The output of the pruned tree is shown above. This tree returned an accuracy score of 61.44%. Excercise angina was the first factor considered in the tree with those affected classified as having ASY chest pain. For those with no exercise angina, people with an ST slope of Down or flat were classified as having ASY chest pain as well. Next, those with cholesterol under 131 were also classified as having ASY chest pain. People with cholesterol over or equal to 131 were than either classified as having ATA chest pain (in ages 28 to 57) or NAP chest pain (in age groups 58 to 77).

## Factor Analysis

```
# Using correlation matrix to check if factor analysis would be worth it:
KMO(R)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = R)
## Overall MSA =  0.52
## MSA for each item =
##   RestingBP Cholesterol   FastingBS       MaxHR     Oldpeak
##        0.49        0.49        0.55        0.54        0.53
```

Since none of the values are greater than .55 Kaiser-Meyer-Olkin (KMO) index, Kasier suggests that our data is “miserable” for Factor Analysis.

## Conclusion

Our goal was to use patient health data to predict whether someone has heart disease as well as what kind of chest pain they are likely to have. With cardiovascular illness related deaths so prevalent in the United States, it is vital that work is done to catch heart disease and chest pain in patients before it is too late. This project provides useful insight into what the most significant indicators of heart disease and chest pain are and simultaneously allows us to see what preventative measures can be taken to reduce risk of heart disease. We had success in meeting our research objectives, most notably in predicting heart disease status. Our best model was the decision tree which had an accuracy of about 88.45% in predicting heart disease status. This means that given data of a new patient in the same format as used in the model, we have around an 88.45% chance of correctly predicting whether or not they have heart disease. We did not have much success with predicting heart pain type and only achieved an accuracy of around 62% with our best model. KNN was very marginally better than our decision tree with an accuracy of 62.31% versus 61.44%. Because this difference is so small and decision trees are more easily interpretable, we concluded that the decision tree is the best method for predicting chest pain type. Overall, the decision tree method proved to be the most accurate and interpretable out of all three methods we attempted using.

## Limitations and Recommendations

One limitation we encountered in our data set was that the data were not multivariate normal (MVN). When conducting mardia's test for MVN we found very strong evidence in favor of rejecting the null hypothesis that the data are MVN. Mardia's tests for skewness and kurtosis yielded p-values close to zero giving us this evidence. Additionally, our chi-square QQ-plot indicates that the data are not MVN. In this plot there is a significant portion of observations whose squared Mahalanobis distances are much greater than their chi-square quantile values. This leads to a deviation from a straight line in the plot indicating non-normality. Luckily, multivariate normality is not required for QDA, although we could have had an even more accurate model if it was present. KNN and the classification tree are non-parametric and therefore by definition do not require multivariate normality.

As for the data itself, we can predict whether a patient has heart disease and what type of chest pain they have fairly well, but we do not have the full picture in terms of the patients profiles. In a perfect world the data would include more descriptive statistics including diet, exercise, smoking habits, drinking habits, etc. With these other variables we would be able to see what habits contribute to chest pain and heart disease in addition to cholesterol, resting blood pressure, resting ecg etc. In terms of drawing conclusions, we do not have any data about race, ethnicity, or comorbidities. We are unable to see how heart disease and chest pain differs between these groups of people and therefore miss out on being able to make predictions specific to particular groups. Also, there is a major class imbalance within the sex variable. There are 725 males and only 193 females in the data set making our findings heavily influenced by data about men.