Heart Attack Indicators

Brendan Ball

07/12/2021

1. INTRODUCTION

For every 40 seconds, a person in the United States will experience a heart attack [1]. If this trend stays relatively constant throughout the year, almost one million Americans will face an episode of a heart attack. While having a heart attack will certainly seem unpredictable, there are specific factors that may affect the probability of experiencing one.

Several factors can influence the risk of having a heart attack. Some of these include age, sex, cholesterol levels, lifestyle choices, and countless other risk factors. As a result, the goal of this project is to detect any relationships between risk factors and the risk for heart attack.

From the Kaggle website, heart attack data was downloaded, with information being provided by 303 different subjects, each with different lived experiences [2]. Based on the data frame provided by the site, there are 15 different variables to be studied.

The following 15 variables introduced in this heart attack model are: **age:** age of the patient **sex:** sex og the patient (0: female; 1: male) **cp:** chest pain type (0: typical angina; 1: atypical angina; 2: non-anginal pain; 3: asymptomatic) **trtbps:** resting blood pressure (mmHg) **chol:** total cholesterol level (mg/dl) **fbs:** fasting blood sugar (>120mg/dl) where (1: true; 0: false) **restecg:** resting electrogardiogram results **thalachh:** maximum heart rate achieved **exng:** exercise induced angina (1: yes; 0: no) **oldpeak:** exercise related to rest **slp:** the slope of the peak exercise ST segment **caa:** number of major vessels (0-3 available) **thall:** thallium stress test result (0-3 available) **output:** chance of heart attack (1: higher chance; 0: lower chance)

Two different approaches will be done to analyze the heart attack data.

Specific Aim 1. Utilize principal component analysis (PCA) to reduce dimensionality while maintaining the variance. This will help find any relationships among the data that was overlooked from preliminary studies.

Specific Aim 2. Investigate the accuracy score with the Naive Bayes Model. A prediction that has a greater chance than 50:50 is desired to be considered as the first steps towards solving these prediction problems.

The following packages will be needed to successfully run the R Mark Down file:

```
# Install necessary packages for the code
if(!require(MASS)) install.packages("MASS", repos = "http://cran.us.r-project.org")
if(!require(rgl)) install.packages("rgl", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2",
                                       repos = "http://cran.us.r-project.org")
if(!require(lattice)) install.packages("lattice",
                                       repos = "http://cran.us.r-project.org")
if(!require(factoextra)) install.packages("factoextra",
                                          repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("dplyr",
```

```
                                      repos = "http://cran.us.r-project.org")
if(!require(stringr)) install.packages("stringr",
                                       repos = "http://cran.us.r-project.org")
if(!require(readxl)) install.packages("readr",
                                      repos = "http://cran.us.r-project.org")
if(!require(knitr)) install.packages("knitr",
                                     repos = "http://cran.us.r-project.org")
if(!require(corrplot)) install.packages("corrplot",
                                        repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret",
                                     repos = "http://cran.us.r-project.org")
if(!require(klaR)) install.packages("klaR",
                                    repos = "http://cran.us.r-project.org")
if(!require(e1071)) install.packages("e1071",
                                     repos = "http://cran.us.r-project.org")

# Downlaod the latest version of mixOmics if needed
# You may be prompted to "update all/some/none?", then type [a/s/n]
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install('mixOmics')
```

Once installed, the libraries can be opened if needed below:

```
# No warning messages, open the installed packages if needed
suppressWarnings(library (MASS, verbose = FALSE))
suppressWarnings(library (rgl, verbose = FALSE))
suppressWarnings(library (ggplot2, verbose = FALSE))
suppressWarnings(library (lattice, verbose = FALSE))
suppressWarnings(library (factoextra, verbose = FALSE))
suppressWarnings(library (dplyr, verbose = FALSE))
suppressWarnings(library (stringr, verbose = FALSE))
suppressWarnings(library (readr, verbose = FALSE))
suppressWarnings(library (corrplot, verbose = FALSE))
suppressWarnings(library (caret, verbose = FALSE))
suppressWarnings(library (klaR, verbose = FALSE))
suppressWarnings(library (e1071, verbose = FALSE))
suppressWarnings(library (mixOmics, verbose = FALSE))
```

The report covers heart attack data from Kaggle, which is then uploaded onto Github for easy download and import of the data [2]. The link to the Github repository is also listed below within the comments of the code.

```
# Download the URL file from the bbkazu5 github account
# https://github.com/bbkazu5/HeartPredictor

# Copy URL and read the CSV file
urlfile = "https://github.com/bbkazu5/HeartPredictor/raw/main/HeartData.csv"
heartdata <-read_csv(url(urlfile))

# Remove the additional column that was generated
suppressWarnings(heartdata <- subset(heartdata, select = -c(X16) ))
```

```
# Remove the url file from the environment
suppressWarnings(rm(urlfile))
```

2. METHODS & ANALYSIS

2.1. BASICS OF THE DATA

To better understand the data, and the current values available, some exploratory analysis was conducted. The first action taken was to determine the class of the downloaded data, which will be called as **heartdata**. The results of the class are reported below.

```
# Determine the class of the heart data
class(heartdata)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

A glimpse of the data set was also observed. There are a total of 303 rows, each representing a subject that was tested. A total of 15 columns are present, which each column represented below:

```
# Glimpse the data set of "heartdata"
glimpse(heartdata)
```

```
## Rows: 303
## Columns: 15
## $ SubjectID <chr> "Subject 1", "Subject 2", "Subject 3", "Subject 4", "Subject~
## $ age       <dbl> 63, 37, 41, 56, 57, 57, 56, 44, 52, 57, 54, 48, 49, 64, 58, ~
## $ sex       <dbl> 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, ~
## $ cp        <dbl> 3, 2, 1, 1, 0, 0, 1, 1, 2, 2, 0, 2, 1, 3, 3, 2, 2, 3, 0, 3, ~
## $ trtbps    <dbl> 145, 130, 130, 120, 120, 140, 140, 120, 172, 150, 140, 130, ~
## $ chol      <dbl> 233, 250, 204, 236, 354, 192, 294, 263, 199, 168, 239, 275, ~
## $ fbs       <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, ~
## $ restecg   <dbl> 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, ~
## $ thalachh  <dbl> 150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, ~
## $ exng      <dbl> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, ~
## $ oldpeak   <dbl> 2.3, 3.5, 1.4, 0.8, 0.6, 0.4, 1.3, 0.0, 0.5, 1.6, 1.2, 0.2, ~
## $ slp       <dbl> 0, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 1, 2, 0, 2, 2, ~
## $ caa       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, ~
## $ thall     <dbl> 1, 2, 2, 2, 2, 1, 2, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ output    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
```

The quartile range, mean, and median of each of the columns, excluding the subjectID column, is included below:

```
# Inspect the result summaries of the results
summary(heartdata)
```

```
##   SubjectID              age              sex               cp
##  Length:303         Min.   :29.00   Min.   :0.0000   Min.   :0.000
##  Class :character   1st Qu.:47.50   1st Qu.:0.0000   1st Qu.:0.000
```

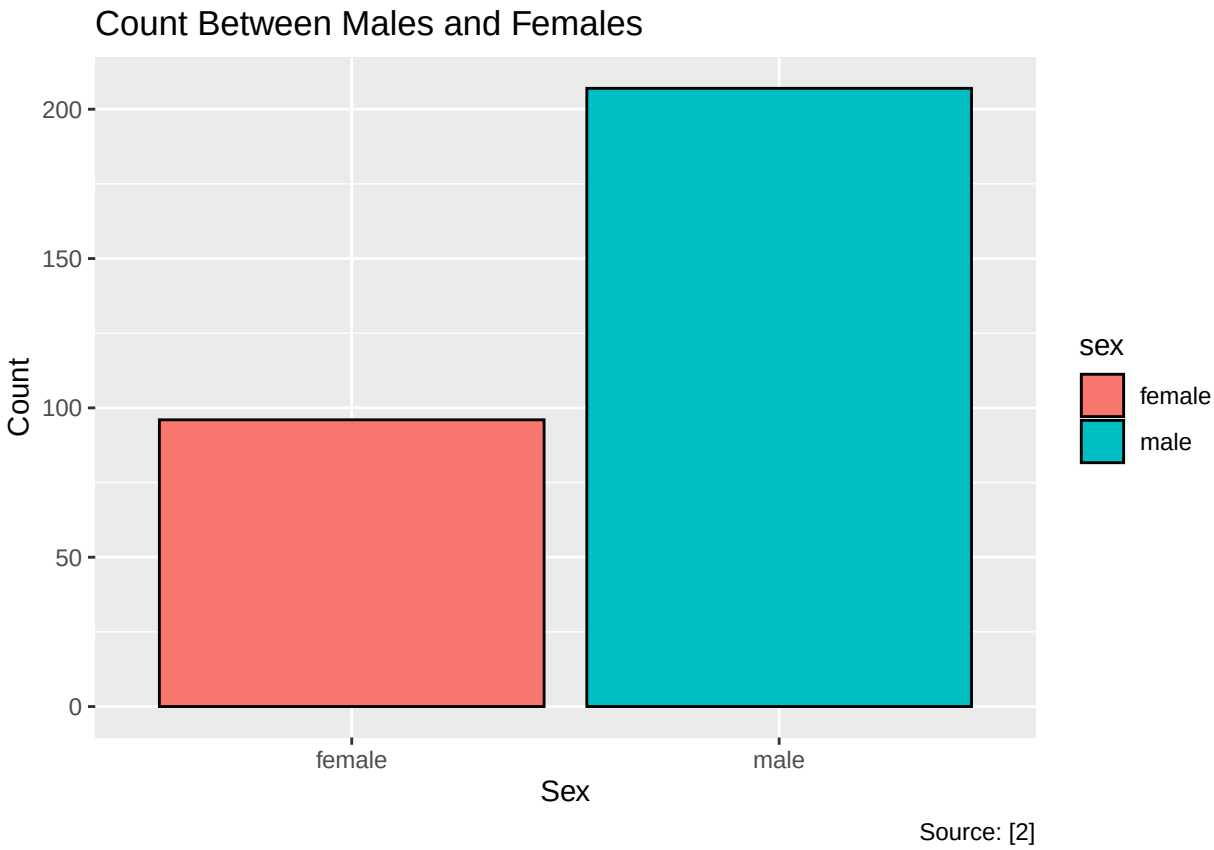
```
##  Mode  :character   Median :55.00   Median :1.0000   Median :1.000
##                     Mean   :54.37   Mean   :0.6832   Mean   :0.967
##                     3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.000
##                     Max.   :77.00   Max.   :1.0000   Max.   :3.000
##      trtbps           chol            fbs            restecg
##  Min.   : 94.0   Min.   :126.0   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:120.0   1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :130.0   Median :240.0   Median :0.0000   Median :1.0000
##  Mean   :131.6   Mean   :246.3   Mean   :0.1485   Mean   :0.5281
##  3rd Qu.:140.0   3rd Qu.:274.5   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :200.0   Max.   :564.0   Max.   :1.0000   Max.   :2.0000
##     thalachh          exng          oldpeak          slp
##  Min.   : 71.0   Min.   :0.0000   Min.   :0.00   Min.   :0.000
##  1st Qu.:133.5   1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000
##  Median :153.0   Median :0.0000   Median :0.80   Median :1.000
##  Mean   :149.6   Mean   :0.3267   Mean   :1.04   Mean   :1.399
##  3rd Qu.:166.0   3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000
##  Max.   :202.0   Max.   :1.0000   Max.   :6.20   Max.   :2.000
##       caa             thall           output
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000
##  Median :0.0000   Median :2.000   Median :1.0000
##  Mean   :0.7294   Mean   :2.314   Mean   :0.5446
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :4.0000   Max.   :3.000   Max.   :1.0000
```

2.2. EXPLORATION OF HEART ATTACK DATA

To qualitatively display each of the results, multiple visualizations are utilized. Bar charts, histograms, and density plots are used to illustrate the data that was downloaded. Before generating plots, some variables were renamed from binary numbers to factored labels, allowing user-friendly interpretation of the graphs.

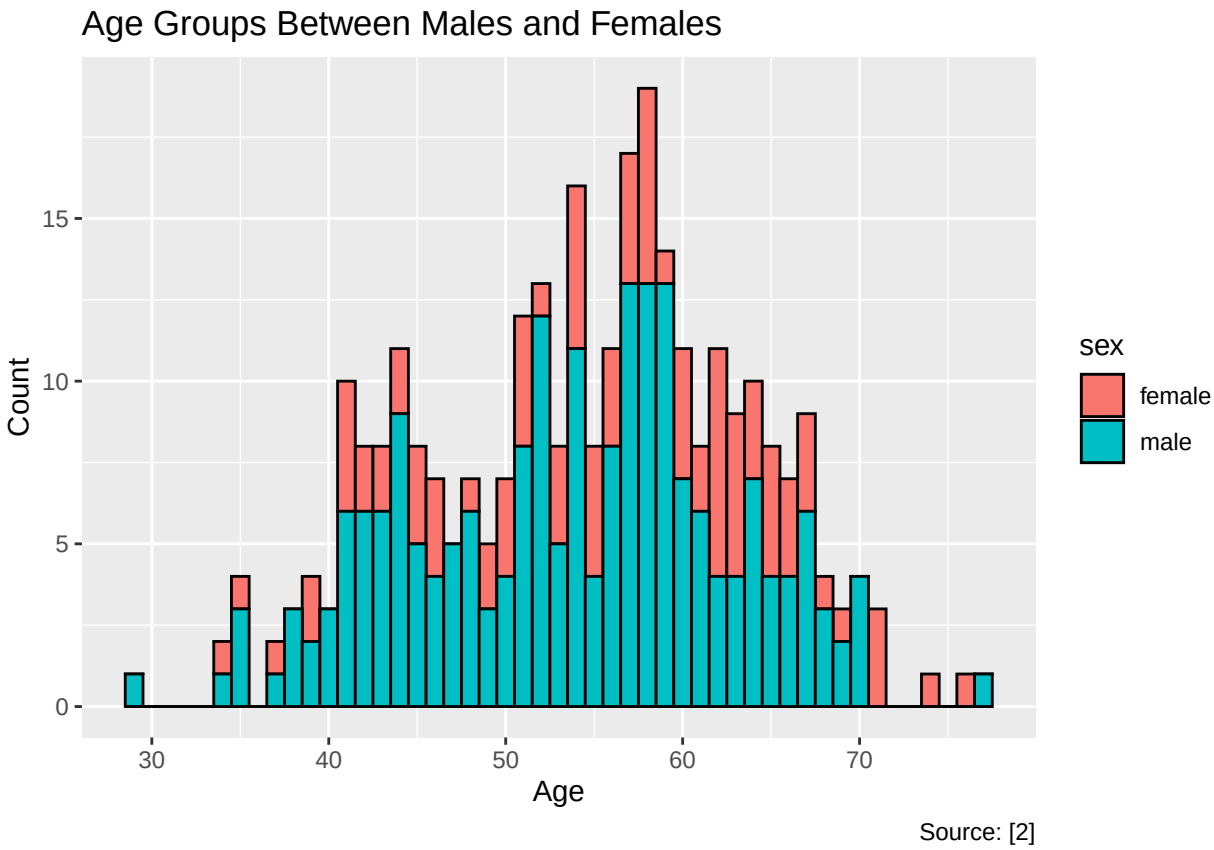
The first plot illustrates the comparison between male and female participants. There are almost more than double male participants than female subjects in the data pool. This vast difference in sex should be considered when looking into other sex-linked results, to prevent bias in the data.

```
# Create a visualization between males and female subjects
ggplot(data=heartdata_new,aes(x=sex,fill=sex))+geom_bar(col="black") +
  xlab("Sex") + ylab("Count") + ggtitle("Count Between Males and Females") +
  labs(caption = "Source: [2]")
```



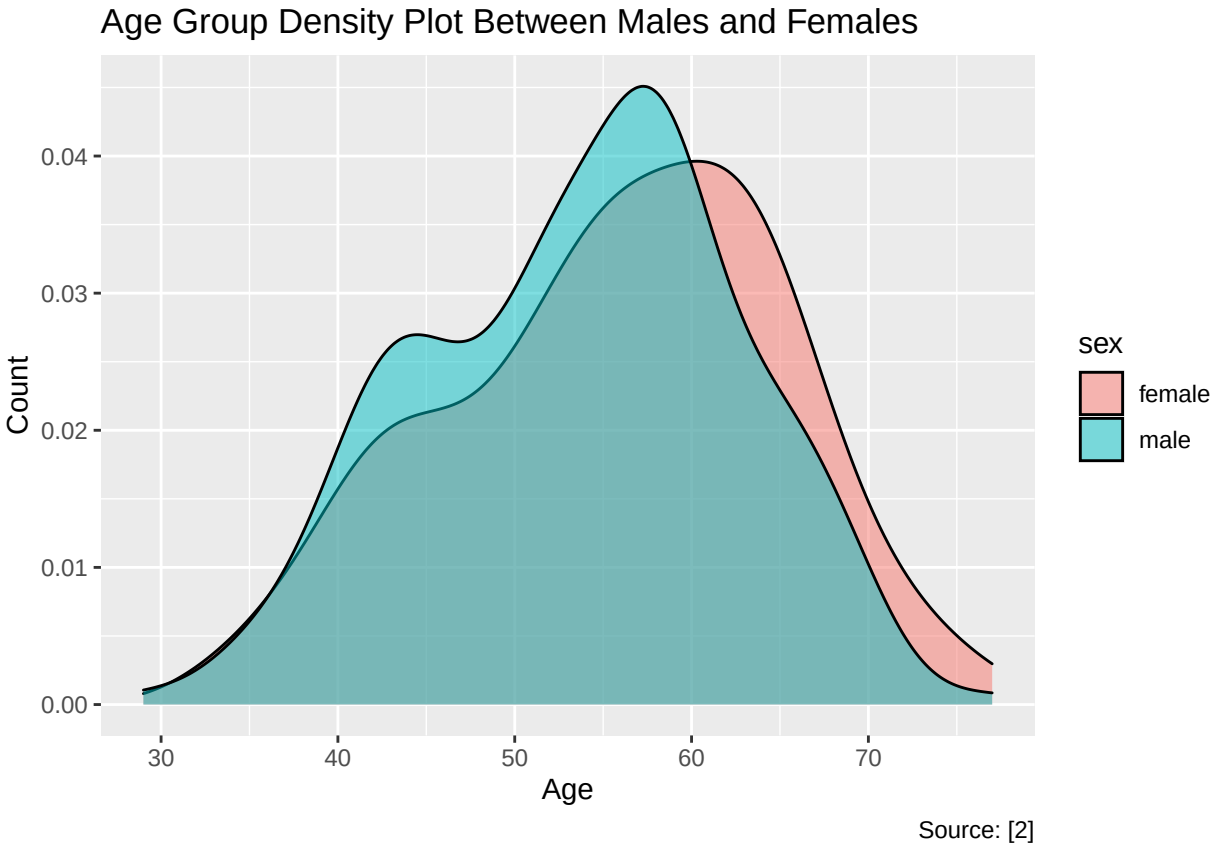
The distribution of age between different sex is also studied. While both sex seems to show similar trends in age distribution, a density plot will be used to confirm this suggestion.

```
# Create a visualization between males and female subjects and age
ggplot(data=heartdata_new,aes(x=age,fill=sex))+geom_histogram(binwidth=1,
                                                              col="black") +
xlab("Age") + ylab("Count") + ggtitle("Age Groups Between Males and Females") +
  labs(caption = "Source: [2]")
```



Using a density plot, it can be seen that in both sex groups, the distribution of age between the two are very similar.

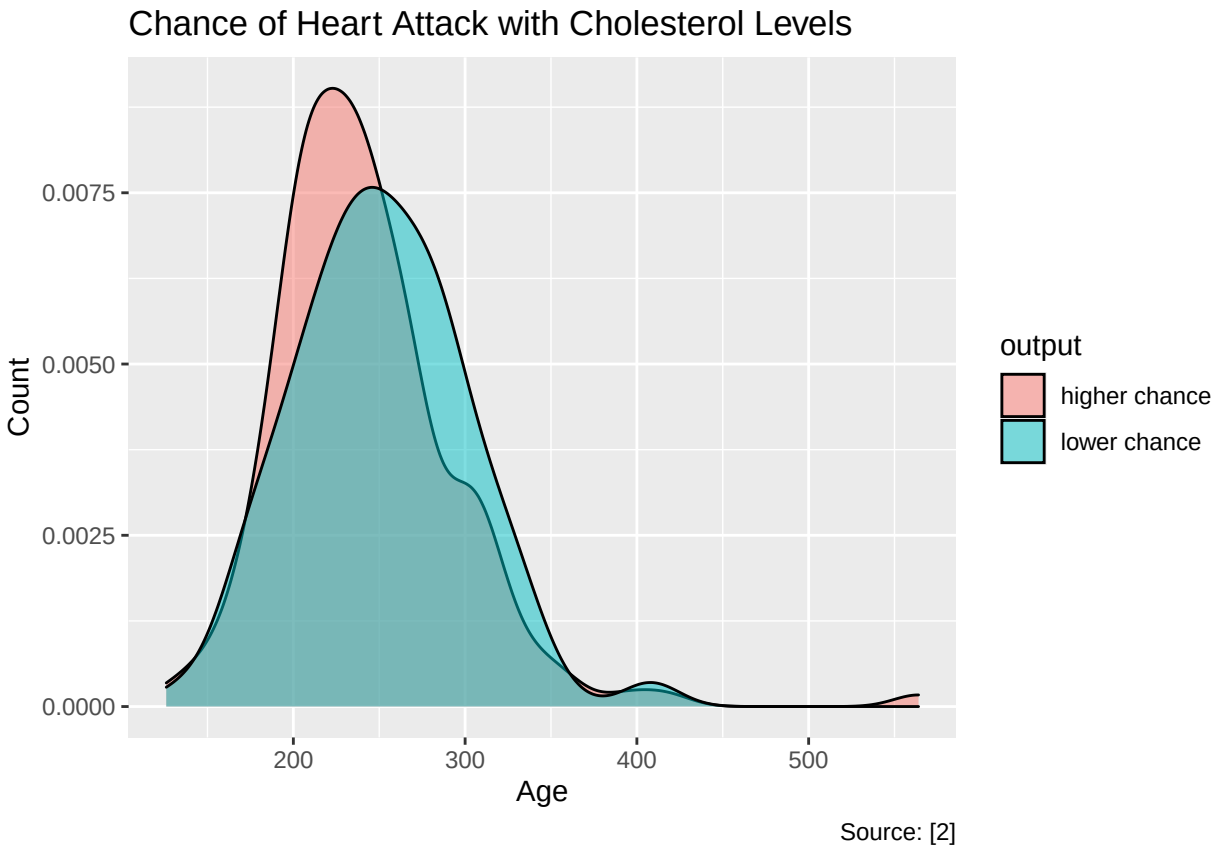
```
# Create a density plot between both sex groups to understand age distribution
ggplot(data=heartdata_new,aes(x=age,fill=sex))+geom_density(alpha = 0.5) +
  xlab("Age") + ylab("Count") +
  ggtitle("Age Group Density Plot Between Males and Females") +
  labs(caption = "Source: [2]")
```



Based from the density plot, it is now confirmed that the two sex groups have a relatively similar distribution in age groups. As a result, it is now safe to assume that any age-related results cannot be attributed to differences in age groups between males and females, as the age profile between subjects are similar to each other.

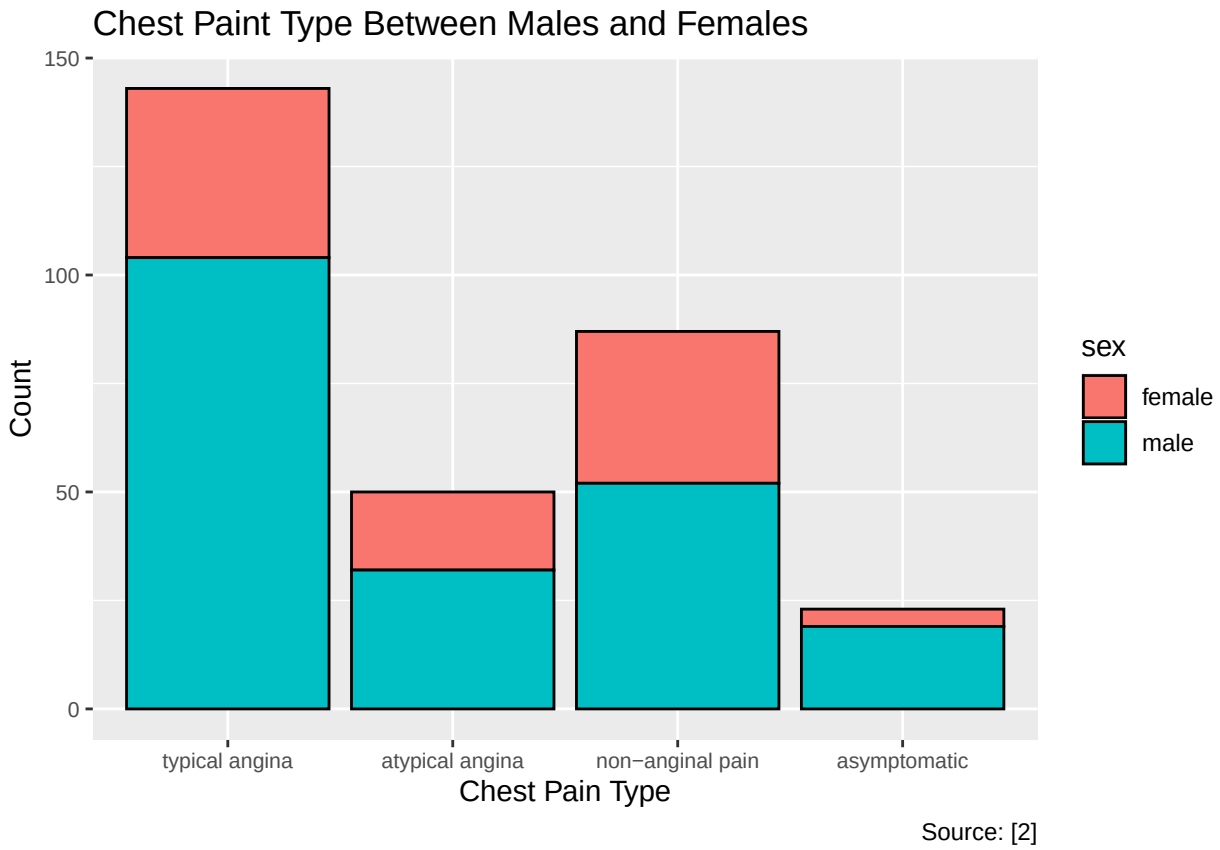
Another interest was the cholesterol data set. When color-coding the distribution of heart attack risk and the level of cholesterol, it is shown that people around 200-250mg/ml of total cholesterol has the highest chance of a heart attack compared to any other range.

```
# Create a density plot between both sex groups to understand age distribution
ggplot(data=heartdata_new,aes(x=chol,fill=output))+geom_density(alpha = 0.5) +
  xlab("Age") + ylab("Count") +
  ggtitle("Chance of Heart Attack with Cholesterol Levels") +
  labs(caption = "Source: [2]")
```



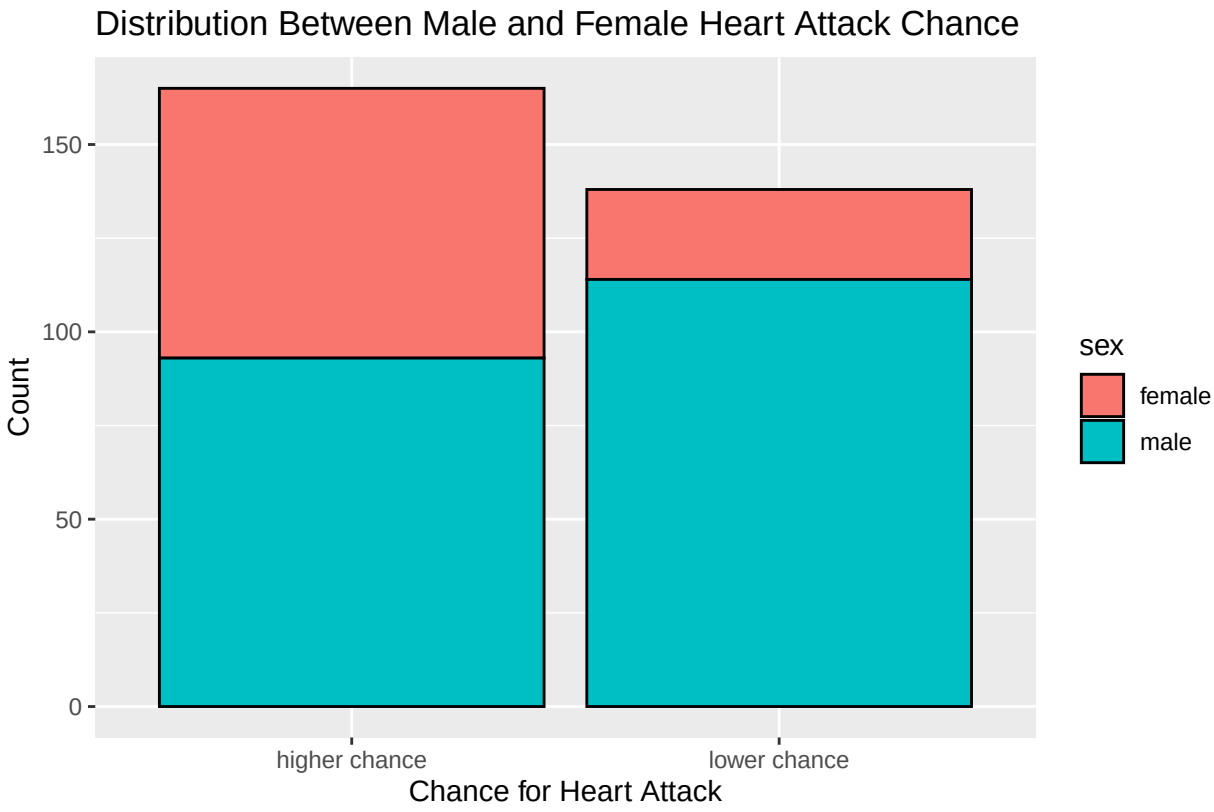
When observing the type of chest pain people experience the most, it is seen that males and females both have similar trends for the most common chest pain types. In both males and females, it seems that typical angina and non-anginal pain are the two leading pain types for both males and females alike.

```
# Create a plot showing the different groups of chest pain types among sex
ggplot(data=heartdata_new,aes(x=cp,fill=sex))+geom_bar(col="black")+
  theme(axis.text=element_text(size = 8)) + xlab("Chest Pain Type") +
  ylab("Count") + ggtitle("Chest Paint Type Between Males and Females") +
  labs(caption = "Source: [2]")
```



However, another interesting data visualization is the chance for heart attack between males and females.

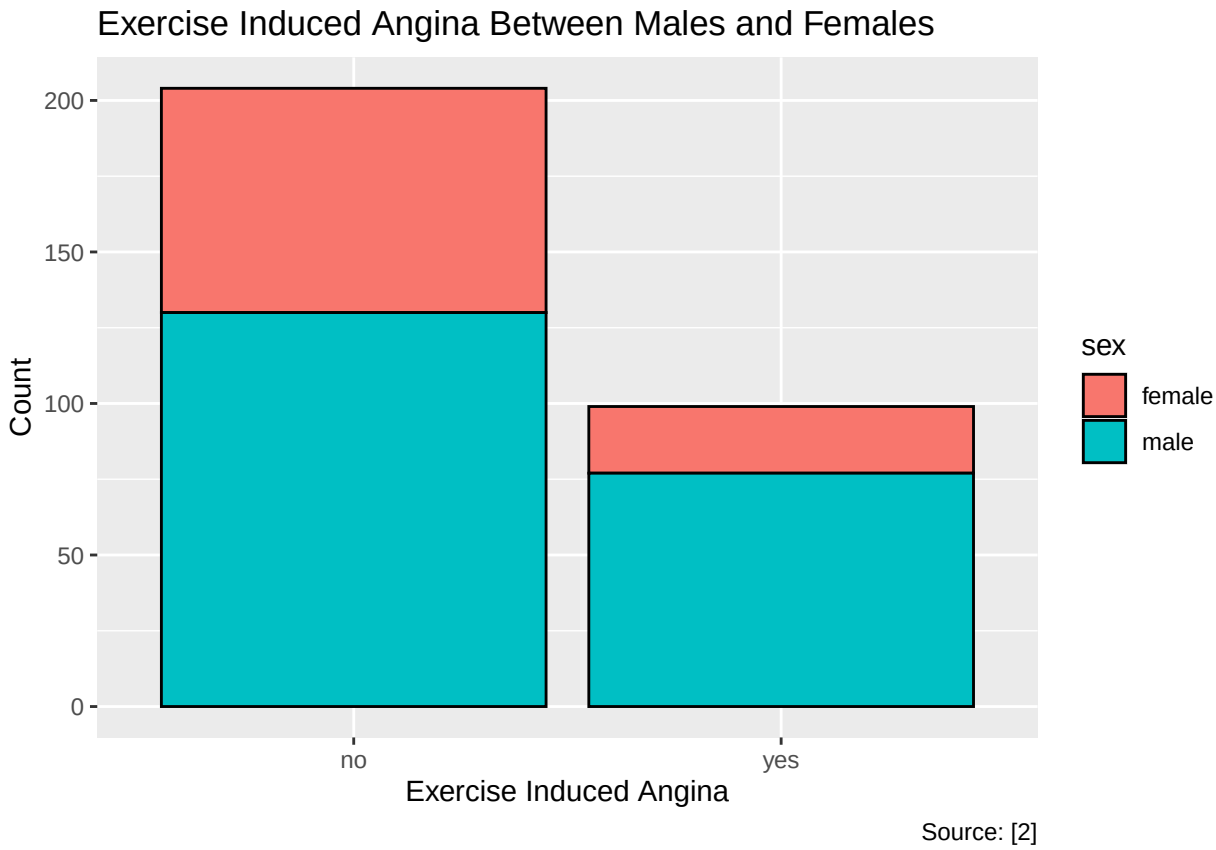
```
# Plot the chance for a heart attack between the two sex groups
ggplot(data=heartdata_new,aes(x=output,fill=sex))+geom_bar(col="black") +
  xlab("Chance for Heart Attack") + ylab("Count") +
  ggtitle("Distribution Between Male and Female Heart Attack Chance") +
  labs(caption = "Source: [2]")
```



Based off of the graph, it seems that females have a higher chance for a heart attack compared to males. More females in their group have a risk for heart attack, while more males have a lower chance. This trend also follows with current understanding of heart attack cases, where women have a higher chance due to their morphological and physical characteristics of their heart. For example, women typically have a higher blood pressure, cholesterol levels, and have heart attacks more frequently at older age than men [4].

In the exercise induced angina (EIA) data between the two sex groups, it shows that a larger percent of the females in their group experiences EIA compared to males.

```
# Plot the exercise induced angina (EIA) and sex relationship
ggplot(data=heartdata_new,aes(x=exng,fill=sex))+geom_bar(col="black") +
  xlab("Exercise Induced Angina") + ylab("Count") +
  ggtitle("Exercise Induced Angina Between Males and Females") +
  labs(caption = "Source: [2]")
```



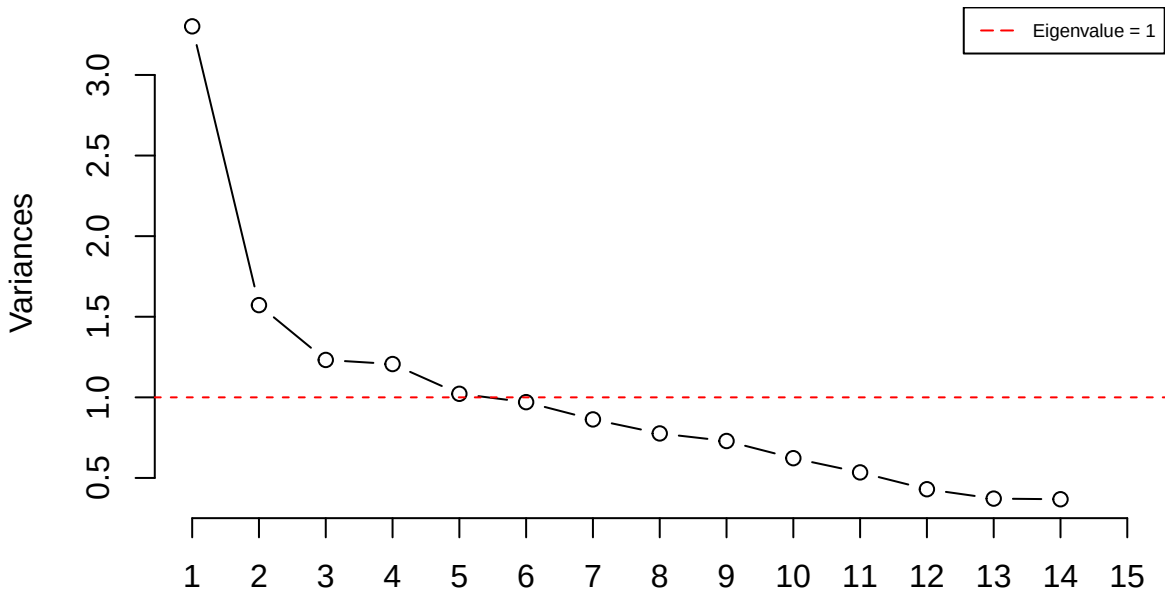
Source: [2]

2.3. PRINCIPAL COMPONENT ANALYSIS (PCA) MODELING

A common problem in complex data analysis emerges from vast numbers of variables. As a result, principal component analysis (PCA), plays an important role in addressing this challenge. In PCA, the dimensionality of a data set is reduced, while maintaining the variation present in the data set. This is done by transforming the variables into new variables, called principal components (PCs).

An initial analysis for PCA includes the creation of a scree plot, which provides a visual representation of the number of PCs needed in a data set. Typically, having an eigenvalue that is greater than 1 represents enough required components for analysis. The scree plot illustrates approximately 5 or 6 PCs to retain the majority of the variance in data.

```
# Standardize the data
heartdata_pr <- prcomp(heartdata[c(2:15)], center = TRUE, scale = TRUE)

# Prepare the Scree Plot for Heart Data
screeplot(heartdata_pr, type = "l", npcs = 15,
          main = "Scree Plot of the First 15 PCs")
abline(h = 1, col="red", lty=20)
legend("topright", legend=c("Eigenvalue = 1"),
       col=c("red"), lty=5, cex=0.6)
```



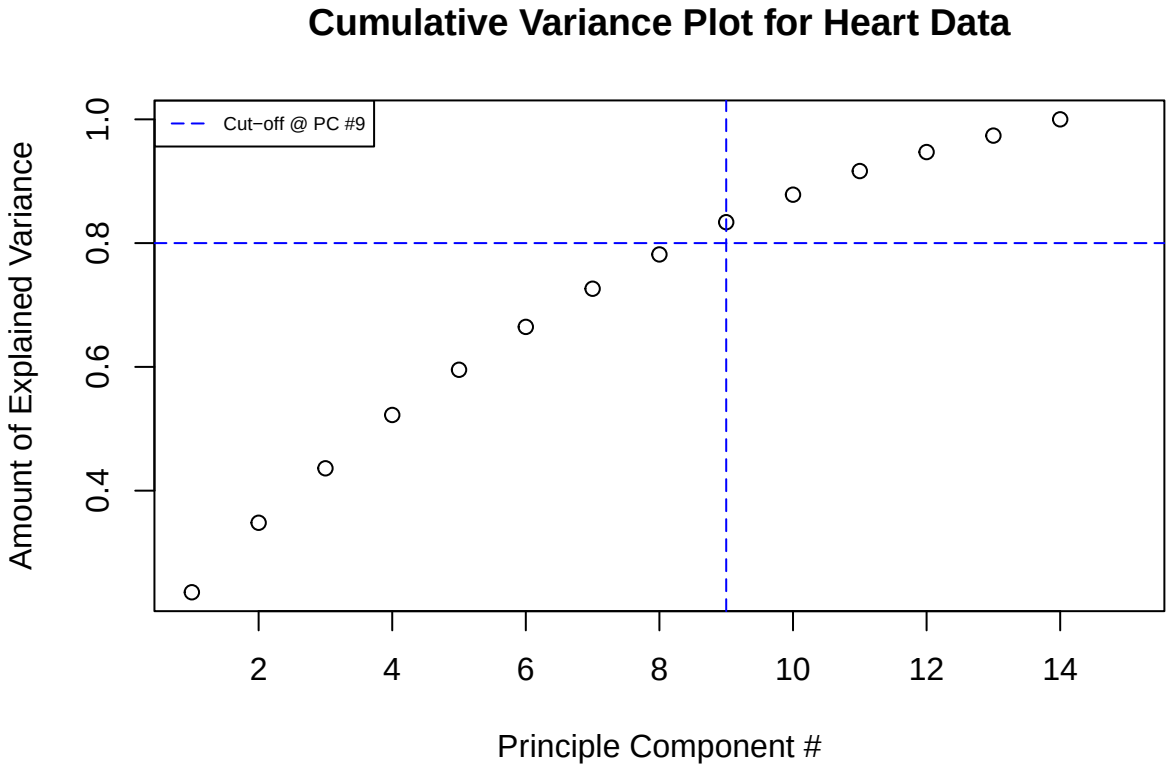
The PCA results can be summarized below, with standard deviation, proportion of variance, and cumulative proportion values.

```
# Create a summary of the results in a table format
summary(heartdata_pr)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4    PC5    PC6     PC7
## Standard deviation     1.8170 1.2539 1.1100 1.09847 1.0110 0.9850 0.92910
## Proportion of Variance 0.2358 0.1123 0.0880 0.08619 0.0730 0.0693 0.06166
## Cumulative Proportion  0.2358 0.3481 0.4361 0.52231 0.5953 0.6646 0.72627
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.88096 0.85393 0.78913 0.73103 0.65577 0.60982 0.60658
## Proportion of Variance 0.05544 0.05209 0.04448 0.03817 0.03072 0.02656 0.02628
## Cumulative Proportion  0.78170 0.83379 0.87827 0.91644 0.94716 0.97372 1.00000
```

To confirm the needed number of principal components, a cumulative variance plot for the heart data is created. The goal of this analysis will be to keep 80% of the explained variance. From the plot, the required PCs to achieve this value is 9 PCs, which is slightly more than the original scree plot suggestion of 5-6. As a result, 9 PCs will be used to ensure more variance is taken into account.

```
# Cumulative Variance Plot for 80% Variance
cumVar <- cumsum(heartdata_pr$sdev^2 / sum(heartdata_pr$sdev^2))
plot(cumVar[0:15], xlab = "Principle Component #",
     ylab = "Amount of Explained Variance",
     main = "Cumulative Variance Plot for Heart Data")
abline(v = 9, col="blue", lty=5)
abline(h = 0.80, col="blue", lty=5)
legend("topleft", legend=c("Cut-off @ PC #9"),
       col=c("blue"), lty=5, cex=0.6)
```

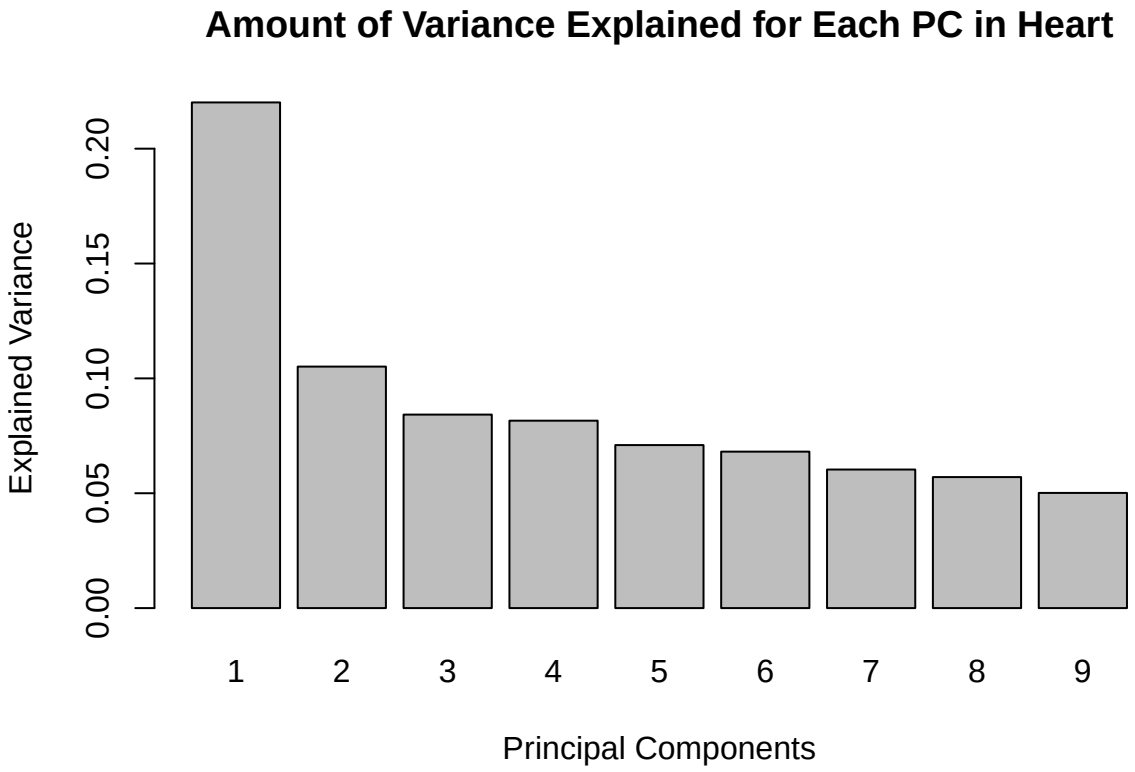



```
# Using the which() command to determine the PC# greater than or equal to 80%
print(c("Princial components at 80% variance is:", which(cumVar >= 0.80)[1]))
```

```
## [1] "Princial components at 80% variance is:"
## [2] "9"
```

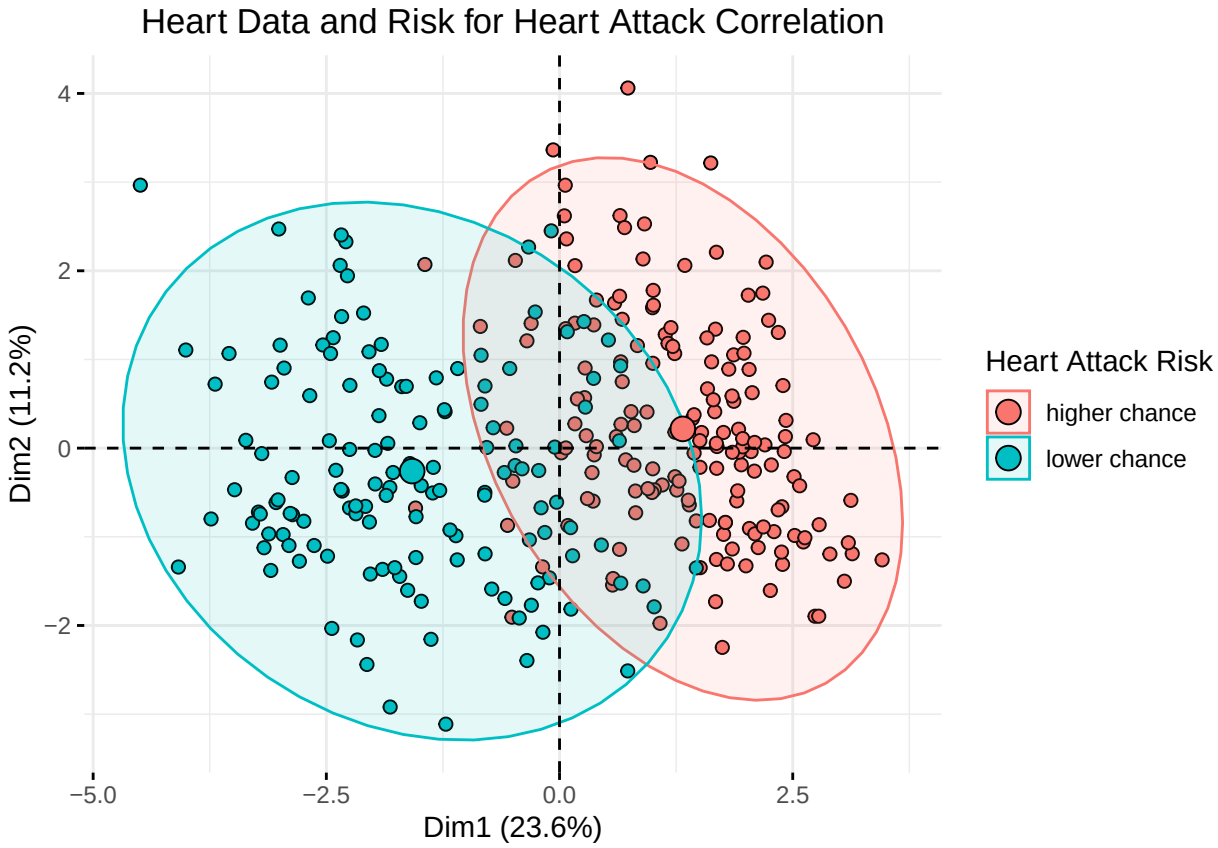
Seeing the contribution of each PC, the first PC provides a little more than 20% of the explained variance, while the remaining PCs offer about half, or less than half of that of the first PC per each component.

```
# Plotting the PCA of heart data to observe the explained variance, ncomp = 9
heartdata_PCA <- pca(heartdata, center = TRUE, scale = TRUE, ncomp = 9)
plot(heartdata_PCA, main = "Amount of Variance Explained for Each PC in Heart")
```



After learning that 9 PCs are needed to maintain 80% of the variance, PCA analysis among different variables is modeled. The data of these components can be split into two different classes.

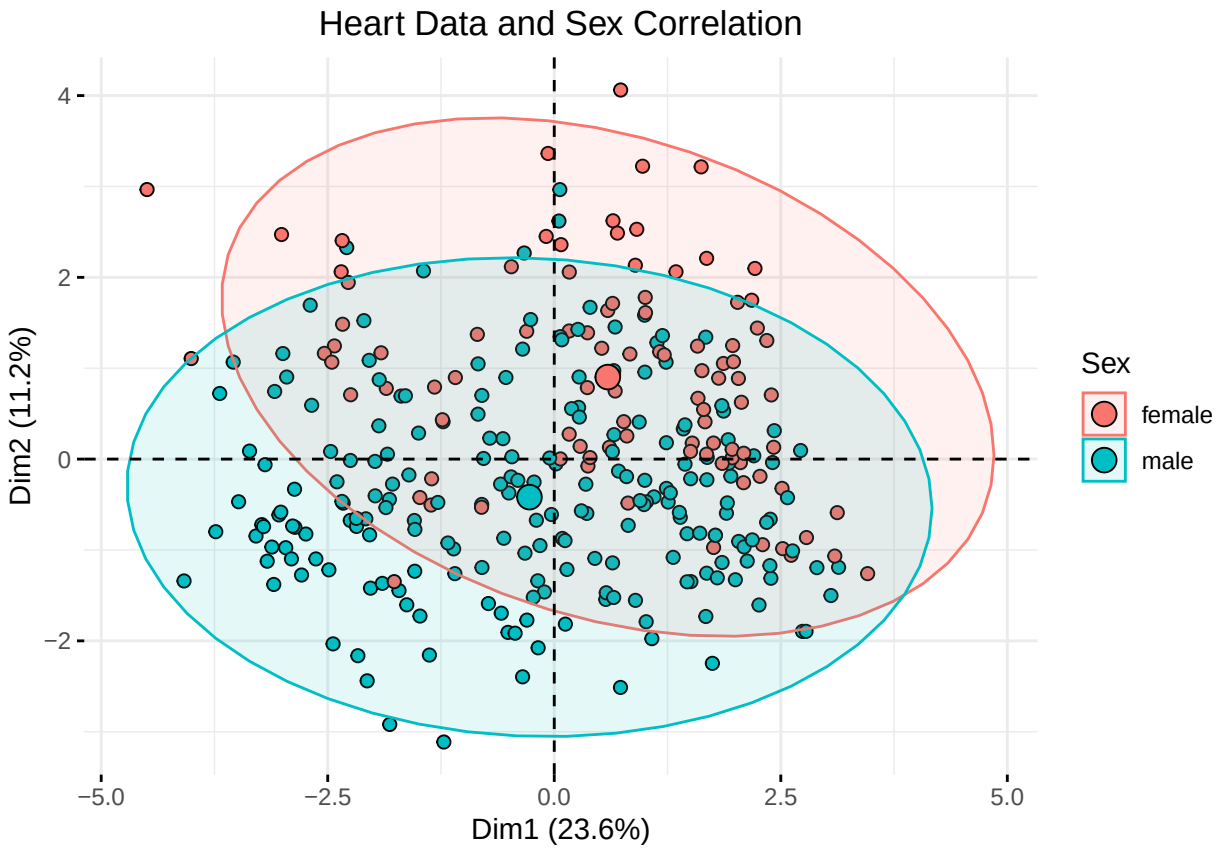
```
# Mutate the data to make chance be a factor, so it can color code the PCA figure
heartdata_output <- heartdata %>%
  mutate(output = factor(output,
                         levels = c(1, 0),
                         labels = c("higher chance", "lower chance")))

# Plot the PCA with color coding of chance of heart attack
fviz_pca_ind(heartdata_pr, geom.ind = "point", pointshape = 21,
  pointsize = 2,
  fill.ind = heartdata_output$output,
  col.ind = "black",
  pallette = "jco",
  addEllipses = TRUE,
  label = "var",
  col.var = "black",
  repel = TRUE,
  legend.title = "Heart Attack Risk") +
ggtitle("Heart Data and Risk for Heart Attack Correlation") +
  theme(plot.title = element_text(hjust = 0.5))
```



Based on the results above, we can infer that the values leaning to the left of the origin may indicate lower risk for heart attack, while data leaning more to the right may show signs of increased risk for heart attack.

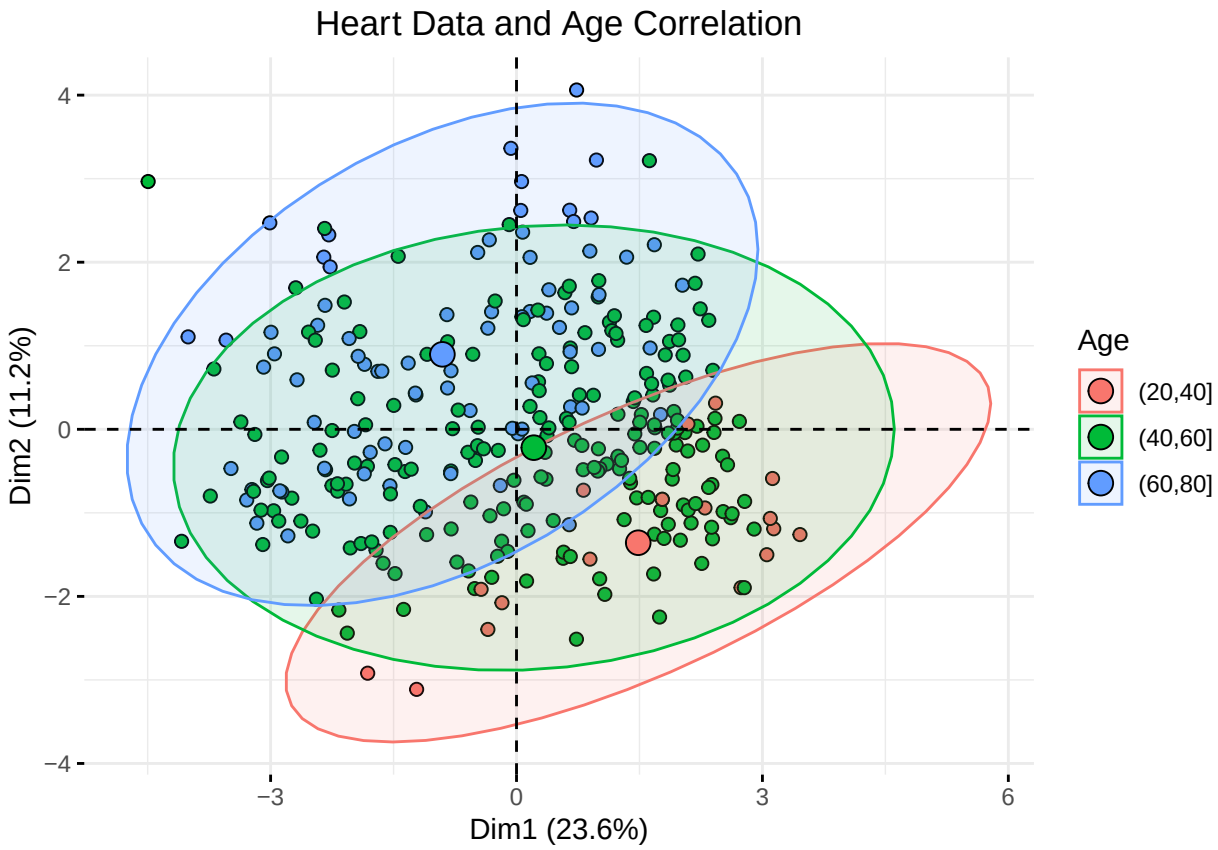
The next PCA analysis is in regard to sex and heart data.

```
# Mutate the data to make sex be a factor, so it can color code the PCA figure
heartdata_sex <- heartdata %>%
  mutate(sex = factor(sex,
                      levels = c(0, 1),
                      labels = c("female", "male")))

# Plot the PCA with color coding of Sex
fviz_pca_ind(heartdata_pr, geom.ind = "point", pointshape = 21,
  pointsize = 2,
  fill.ind = heartdata_sex$sex,
  col.ind = "black",
  pallette = "jco",
  addEllipses = TRUE,
  label = "var",
  col.var = "black",
  repel = TRUE,
  legend.title = "Sex") +
ggtitle("Heart Data and Sex Correlation") +
  theme(plot.title = element_text(hjust = 0.5))
```



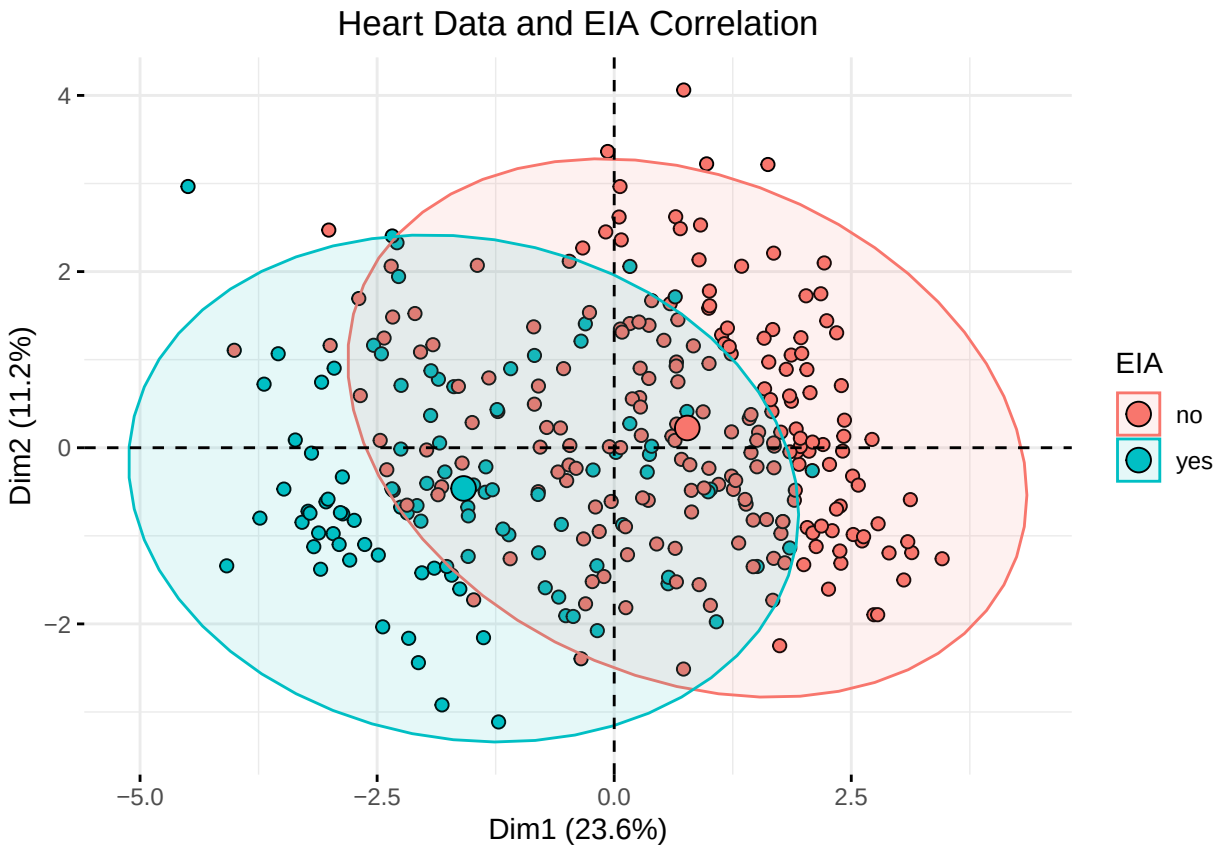
Different age groups (20-40, 40-6, and 60-80) are characterized in the data below:

```
# Plot the PCA with color coding of Age
fviz_pca_ind(heartdata_pr, geom.ind = "point", pointshape = 21,
  pointsize = 2,
  fill.ind = cut(heartdata$age, breaks = c(20, 40, 60, 80)),
  col.ind = "black",
  pallette = "jco",
  addEllipses = TRUE,
  label = "var",
  col.var = "black",
  repel = TRUE,
  legend.title = "Age") +
ggtitle("Heart Data and Age Correlation") +
  theme(plot.title = element_text(hjust = 0.5))
```



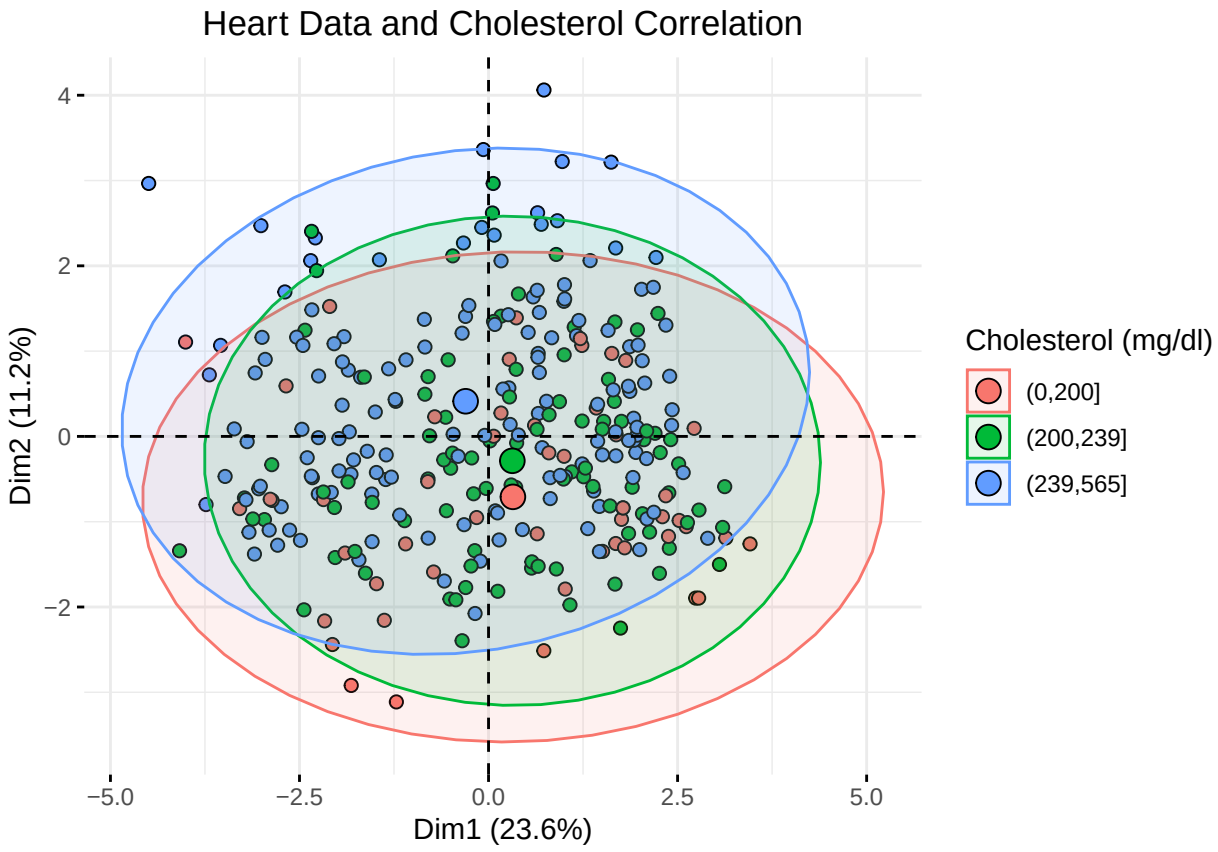
Principal component analysis results in respect to EIA correlation below:

```
# Mutate the data to make exng (exercise induced angina) be a factor
# This way, it can color code the PCA figure
heartdata_exng <- heartdata %>%
  mutate(exng = factor(exng,
                       levels = c(0, 1),
                       labels = c("no", "yes")))

# Plot the PCA with color coding of Exercised Induced Angina (EIA)
fviz_pca_ind(heartdata_pr, geom.ind = "point", pointshape = 21,
  pointsize = 2,
  fill.ind = heartdata_exng$exng,
  col.ind = "black",
  pallette = "jco",
  addEllipses = TRUE,
  label = "var",
  col.var = "black",
  repel = TRUE,
  legend.title = "EIA") +
ggtitle("Heart Data and EIA Correlation") +
  theme(plot.title = element_text(hjust = 0.5))
```



The final PCA result includes heart data with cholesterol levels being color coded.

```
# Total cholesterol ranges based on adults [3]
# Desirable Range: under 200mg/dl
# Borderline Range: 200-239mg/dl
# High Range: above 240mg/dl

# Plot the PCA with color coding of cholesterol
fviz_pca_ind(heartdata_pr, geom.ind = "point", pointshape = 21,
  pointsize = 2,
  fill.ind = cut(heartdata$chol, breaks = c(0, 200, 239, 565)),
  col.ind = "black",
  pallette = "jco",
  addEllipses = TRUE,
  label = "var",
  col.var = "black",
  repel = TRUE,
  legend.title = "Cholesterol (mg/dl)") +
ggtitle("Heart Data and Cholesterol Correlation") +
  theme(plot.title = element_text(hjust = 0.5))
```



2.4. RESULTS DISCUSSION FROM THE PCA

The PCA results showed interesting information. For example, the PCA for the risk of heart attacks had two distinct groups, which may indicate that the first two components can be separated into two distinct classes. This can be seen with the age and EIA results as well. However, it seems that the cholesterol shows less of a grouping among the different cholesterol levels. Based on these results, further studies could be conducted with discriminant analysis.

2.5. Linear Discriminant Analysis (LDA)

In linear discriminant analysis (LDA), the approach takes into consideration the various class groups, and because of this, better results may be achieved. The LDA approach does assume a normal distribution for the class, mean, and variance, which will all be assumed to be true in this model for simplicity. However, if further rigourous tests are to be done, this portion of the model should be reanalyzed.

```
# Mutate to make output (heart attack risk) be a factor of yes or no
heartdata_prep_lda <- heartdata %>%
  mutate(output = factor(output,
                         levels = c(1, 0),
                         labels = c("higher chance", "lower chance")))

# LDA analysis
heart_lda <- lda(output~., data = heartdata_prep_lda, center = TRUE,
                 scale = TRUE)
```

```r
# Print the results out
heart_lda$prior
```

```
## higher chance  lower chance
##     0.5445545     0.4554455
```

There is a 54.46% chance for heart attack risk, and a 45.54% chance for a reduced risk of a heart attack.

3. NAIVE BAYES THEOREM

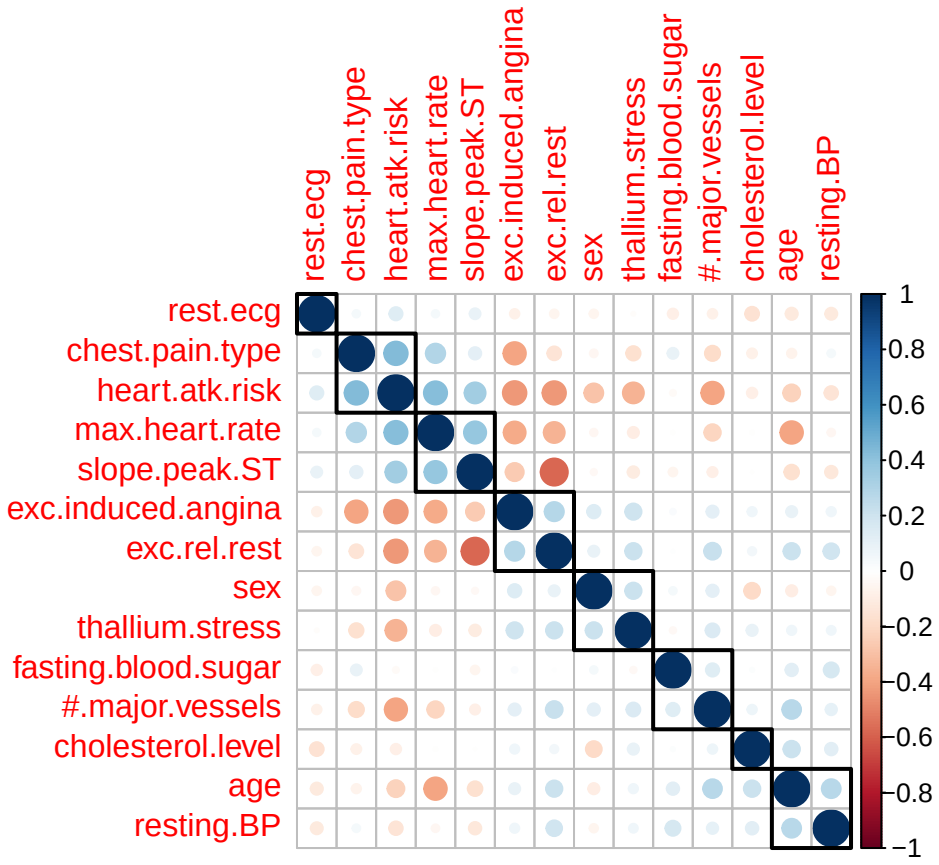
The Naive Bayes model is an easy predictor to generate for large data sets, and no complicated iterations are required to be successful. The Naive Bayes Theorem provides a way to determine the posterior probability, P(C|x), with the values P(c), P(x), and P(x|C). A conditional independence is assumed in this theorem, such that the predictor, x, on a given class, c, is not dependent on the values of the other predictors.

$$P(C|x) = \frac{p(C)|p(x|C)}{p(x)}$$

3.1. CREATING THE MODEL

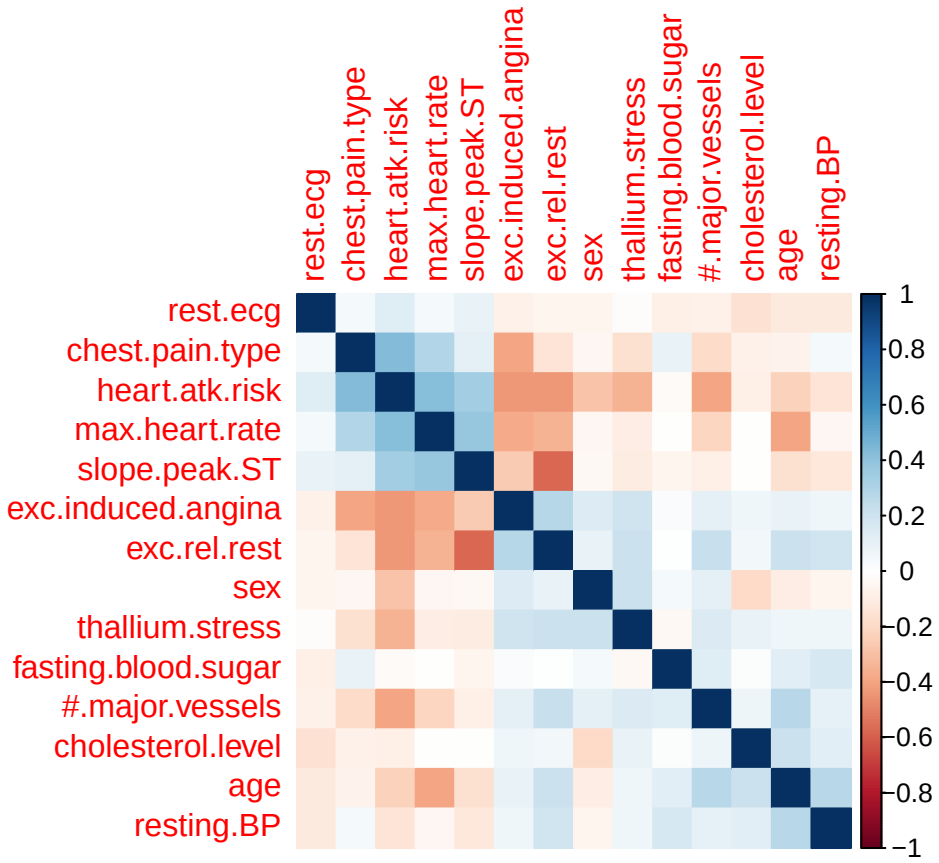
Before creating the Naive Bayes predictor, a correlation matrix will be created to learn more about the relationship between the data. Below, blue values represent higher correlation, while red values represent lower correlation. Any boxes around the values is based on a hierarchical clustering organization.

```r
# Rename columns to be better associated in following figures
heart1 <- heartdata
colnames(heart1)[4] <- "chest.pain.type"
colnames(heart1)[5] <- "resting.BP"
colnames(heart1)[6] <- "cholesterol.level"
colnames(heart1)[7] <- "fasting.blood.sugar"
colnames(heart1)[8] <- "rest.ecg"
colnames(heart1)[9] <- "max.heart.rate"
colnames(heart1)[10] <- "exc.induced.angina"
colnames(heart1)[11] <- "exc.rel.rest"
colnames(heart1)[12] <- "slope.peak.ST"
colnames(heart1)[13] <- "#.major.vessels"
colnames(heart1)[14] <- "thallium.stress"
colnames(heart1)[15] <- "heart.atk.risk"

# Create a correlation matrix
# Rectangles around the plot is based on hierarchical clustering.
corrMatrix <- cor(heart1[, 2:15])
corrplot(corrMatrix, order = "hclust", tl.cex = 1, addrect = 8)
```



Another similar correlation plot is provided below, with each box being fully colored instead to provide an easier color comparison among the data.

```
# Create a correlation matrix plot
corrplot(corrMatrix, method = 'color', order = 'hclust')
```



Since some machine learning algorithms could fail if correlation are too high, anything within a 0.9 value will be identified. As shown below, no values are within this high range, so no values will be removed for the machine learning algorithm.

```
# Find attributes that is correlated highly
correlated <- findCorrelation(corrMatrix, cutoff = 0.9)

# Print any correlated number of variables
print(correlated)
```

```
## integer(0)
```

3.2. CREATING THE NAIVE BAYES MODEL

The Naive Bayes Prediction model is made now made, with the training and predictor values being prepared. The data will be split in a 70:30 ratio (training:testing), and all dimensions of the split are proven to be nearly the same.

```
# Maintain reproducibility
set.seed(1)

# Shuffle the rows, but consistently every time it is ran
rows <- sample(nrow(heartdata))
heartdata_shuffle <- heartdata[rows, ]
```

```r
# Mutate to make output (heart attack risk) be a factor of yes or no
heart_model <- heartdata_shuffle %>%
  mutate(output = factor(output,
                         levels = c(1, 0),
                         labels = c("True", "False")))

heart_model <- heart_model[-1]

heart_model$output <- factor(heart_model$output, levels = c("True", "False"),
                             labels = c("High.Risk", "Low.Risk"))

# Split data into training and test data sets
indxTrain <- createDataPartition(y = heart_model$output,p = 0.7,list = FALSE)
training <- heart_model[indxTrain,]
testing <- heart_model[-indxTrain,]

# Check dimensions of the split
round(prop.table(table(heart_model$output)) * 100, digits = 1)
```

```
##
## High.Risk  Low.Risk
##      54.5      45.5
```

```r
round(prop.table(table(training$output)) * 100, digits = 1)
```

```
##
## High.Risk  Low.Risk
##      54.5      45.5
```

```r
round(prop.table(table(testing$output)) * 100, digits = 1)
```

```
##
## High.Risk  Low.Risk
##      54.4      45.6
```

```r
# Create objects x (predictor variables) and y (response variables)
x = training[,-14]
y = training$output
```

The Naive Bayes predictor below shows the an 82.6% accuracy.

```r
# Naive Bayes Model
model = train(x,y,'nb',trControl=trainControl(method='cv',number=10))
model
```

```
## Naive Bayes
##
## 213 samples
##  13 predictor
##   2 classes: 'High.Risk', 'Low.Risk'
```

```
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 191, 191, 192, 192, 192, 191, ...
## Resampling results across tuning parameters:
##
##   usekernel  Accuracy   Kappa
##   FALSE      0.8259091  0.6467368
##    TRUE      0.8256494  0.6453195
##
## Tuning parameter 'fL' was held constant at a value of 0
## Tuning
##  parameter 'adjust' was held constant at a value of 1
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were fL = 0, usekernel = FALSE and adjust
##  = 1.
```

Generating a confusion matrix, an accuracy of 76.67% was achieved, with a 95% confidence interval (CI) being between 66.57-84.94%. Based off of the p-value, the accuracy is shown to be a success.

```
# Confusion matrix generation
Predict <- predict(model,newdata = testing )
confusionMatrix(Predict, testing$output )
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  High.Risk Low.Risk
##   High.Risk        37        9
##   Low.Risk         12       32
##
##                Accuracy : 0.7667
##                  95% CI : (0.6657, 0.8494)
##     No Information Rate : 0.5444
##     P-Value [Acc > NIR] : 1.061e-05
##
##                   Kappa : 0.5324
##
##  Mcnemar's Test P-Value : 0.6625
##
##             Sensitivity : 0.7551
##             Specificity : 0.7805
##          Pos Pred Value : 0.8043
##          Neg Pred Value : 0.7273
##              Prevalence : 0.5444
##          Detection Rate : 0.4111
##    Detection Prevalence : 0.5111
##       Balanced Accuracy : 0.7678
##
##        'Positive' Class : High.Risk
##
```

The Naive Bayes model shows to be a promising machine learning technique from the data approach. However, other machine learning techniques should be studied before confirming this predictor to be the best.

4. CONCLUSION & FUTURE WORK

This report analyzed the heart attack data provided by Kaggle [2]. Based from the data, it was first observed that women are more likely to face higher risk for heart attack compared to males. Additionally, typical angina is shown to be one of the most populous forms of chest pain among both genders. Using PCA, different types of variables were modeled to display any differences in the data. This was accomplished by using only 9 principal components.

Utilizing a Naive Bayes technique, an accuracy of 77% was achieved. This shows that the prediction made is better than if a person were to guess 50:50 if a person were to be higher risk for a heart attack. With an algorithm that can predict a person's likeliness for a heart attack, more lives can be saved in the United States, or even globally.

While the Naive Bayes method works much better than guessing, there is still more work to be done to maximize the accuracy of a heart attack prediction. Because of this, future work can involve other machine model techniques such as k-Nearest Neighbors (kNN), Neural PCA, or even Random Forest approaches to test and rank against the Naive Bayes technique. With a ranking of different machine learning techniques, the accuracy can be optimized, and more lives can be saved.

Additionally, this report was limited to only 303 subjects. With a larger sample size, there can be less variability within the greater population. Thus, more sample results should be collected in future studies to provide better training for the machine learning tools available. While these limitations will provide the groundwork for future investigations, this report illustrates the importance of machine learning algorithms, and how even a technique that is not fully trained to its potential can influence the probability of a person being notified of his or her risks for a heart attack.

5. REFERENCES

1. American Heart Association, "What is a Heart Attack?," www.heart.org, 2021. [Online]. Available: https://www.heart.org/en/health-topics/heart-attack/about-heart-attacks. [Accessed: 13-Jul-2021].

2. R. Rahman, "Heart Attack Analysis & Prediction Dataset," Kaggle, 22-Mar-2021. [Online]. Available: https://www.kaggle.com/rashikrahmanpritom/heart-attack-analysis-prediction-dataset. [Accessed: 13-Jul-2021].

3. D. Weatherspoon, "Cholesterol levels by age: Differences and recommendations," Medical News Today, 05-Jan-2020. [Online]. Available: https://www.medicalnewstoday.com/articles/315900#recommended-levels. [Accessed: 13-Jul-2021].

4. H. and V. Team, "Women or Men - Who Has a Higher Risk of Heart Attack?," Health Essentials from Cleveland Clinic, 30-Sep-2020. [Online]. Available: https://health.clevelandclinic.org/women-men-higher-risk-heart-attack/. [Accessed: 13-Jul-2021].

6. ACKNOWLEDGEMENTS

I would like to thank the HarvardX community, as well as the staff members and Dr. Rafael Irizarry for being supportive, and providing an atmosphere where I can effectively learn R coding for the first time. I have learned a lot from this 9-course series, and I aim to learn more in near future!