Heart Disease Indicators:Analyzing CDC Survey Data To Identify High Risk Individuals

Arjun Ravi and Ben Sikora

12/06/2021

Introduction

In this project, our question of interest is: how well can we predict heart disease given other health factors and survey responses? According to the Center for Disease Control and Prevention (CDC), heart disease is the leading cause of death for men, women, and people of most racial and ethnic groups in the United States. One person dies every 36 seconds in the United States from cardiovascular disease. About 659,000 people in the United States die from heart disease each year – 1 in every 4 deaths. In turn, it is critical to predict the factors underlying heart disease so we can help people prevent it, especially those who have familial risk. That said, more information is critical to fighting this immense problem.

We obtained a heart disease dataset from Kaggle, an open-source data website. The dataset originally comes from the CDC and was collected from 2011 to 2015. The data is survey data on individual health characteristics, meaning the individual is responding to whether they have the condition in question or answering questions about themselves. There is a single binary target variable, whether or not the individual had heart disease or a heart attack. In addition, there are twenty-one covariates that are either categorical or numeric. Notably, even the numeric variables have been split into further categories for sake of the anonymity of the individuals.

The survey asks the following question of interest to respondents coded as HeartDiseaseorAttack: "Have you ever had a heart disease or heart attack?" The other variables are included in Appendix: Variables.

Data Cleaning and Selection

The contributor of the dataset to Kaggle did extensive cleaning of the data already, so there were no missing observations. After reading in the data, we simply put the variables in the correct type to proceed with our analysis. Moreover, the data originally includes over 250,000 observations. To reasonably conduct our analysis with supervised learning approaches, we simply took a random sample of 15,000 observations.

Upon an initial look, our data seems to match what one would expect about the make-up of the health of Americans. About 9.5% of individuals in our sample have had heart disease or a heart attack. The vast majority, like with heart disease, do not have the conditions in question (stroke, high cholesterol, high BP, etc) but many do. One striking observation, though, was that almost half of the individuals in our sample are smokers.

Principal Component Analysis

We proceed with principal component analysis and create a scree plot and biplot for the first two principal components (Appendix: PCA, Scree plot and Biplot).

We identify that the first principal component accounts for about 16.91% of the total variability whereas the first two principal components account for about 25% of the total variability. Given the range of our data

collection, we were fairly surprised to see that the two principal components captured as much variability as they did.

We also identify that the most important variables that contribute to dimension one are GenHlth, DiffWalk, PhysHlth, Income, and HighBP. The most important variables for the second dimension are Age, AnyHealthcare, NoDocbcCost, HighBP, and HighChol. Notably, the only common variable between these two groups is HighBP (Appendix: PCA, Contribution Summary).

Given that we have proceeded with a dimension reduction approach, we want to see if we captured enough variability to identify whether individuals had heart disease/attack or not (For the code please see Appendix: PCA, Class Separation).

Clearly, this graph indicates that principal component analysis does a poor job of actually predicting heart disease or attack. It is not able to separate the two groups with precision. This is not surprising given that PCA is trying to capture the variability of the whole data, when we are only looking at heart disease for this project.

Logistic Regression

We then move to supervised methods and work on a logistic regression model for predicting heart disease or attack. Because we had over twenty covariates, we decided to use a Lasso Penalized Regression to choose the relevant variables for our analysis (Appendix: Logistic Regression, Lasso Penalized Regression Variables). The variables selected were: high blood pressure, high cholesterol, stroke, diabetes2, general health, difficulty walking, sex, and age. While diabetes1 is not selected by the lasso model, we still decide to include the diabetes variable in order to count diabetes2. Not many of the other variables are shocking to us, but we are surprised that other variables like smoker were not in the specification.

When we ran the final model we found this result:

log-odds of heart disease/attack= -7.45 + 0.46*HighBP + 0.64*HighChol + 1.66*Stroke - 0.08*Diabetes1 + 0.29*Diabetes2 + 0.51*GenHlth + 0.82*Sex + 0.26*Age + 0.38*DiffWalk

Stroke has the highest log-odds. Controlling for the other variables in the regression, if you have had a stroke, the log-odds of having heart disease or a heart attack increases by 1.66. That said, we were surprised to see that sex had as high an impact as it did. In fact, it was the second most significant predictor. We expected to find that high blood pressure or high cholesterol would have had a stronger effect. We also still feel comfortable moving forward with diabetes1 despite its high p-value because the log-odds were by the far the smallest so its likely that its inclusion did not have much of an effect (Appendix: Logistic Regression, Model).

Since our model contains continuous variables, we ran a Homser-Lemeshow goodness-of-fit test (Appendix: Logisitic Regression, Model Diagnostics). Initially a p-value of 0.0004735 was concerning to us, because this meant that we reject H_0 and conclude that the model does not fit the data well. Yet, we understand that the high sample size could have impacted the significance of the p-value so we fit the same model again with only 5000 observations and found a higher p-value of 0.07025 Although the p-value is not as high as we would have liked, with a smaller sample size and a p cutoff of 0.05, we are not able to reject H_0 and can conclude that the model fits the data well (Appendix: Logistic Regression, Model Diagnostics).

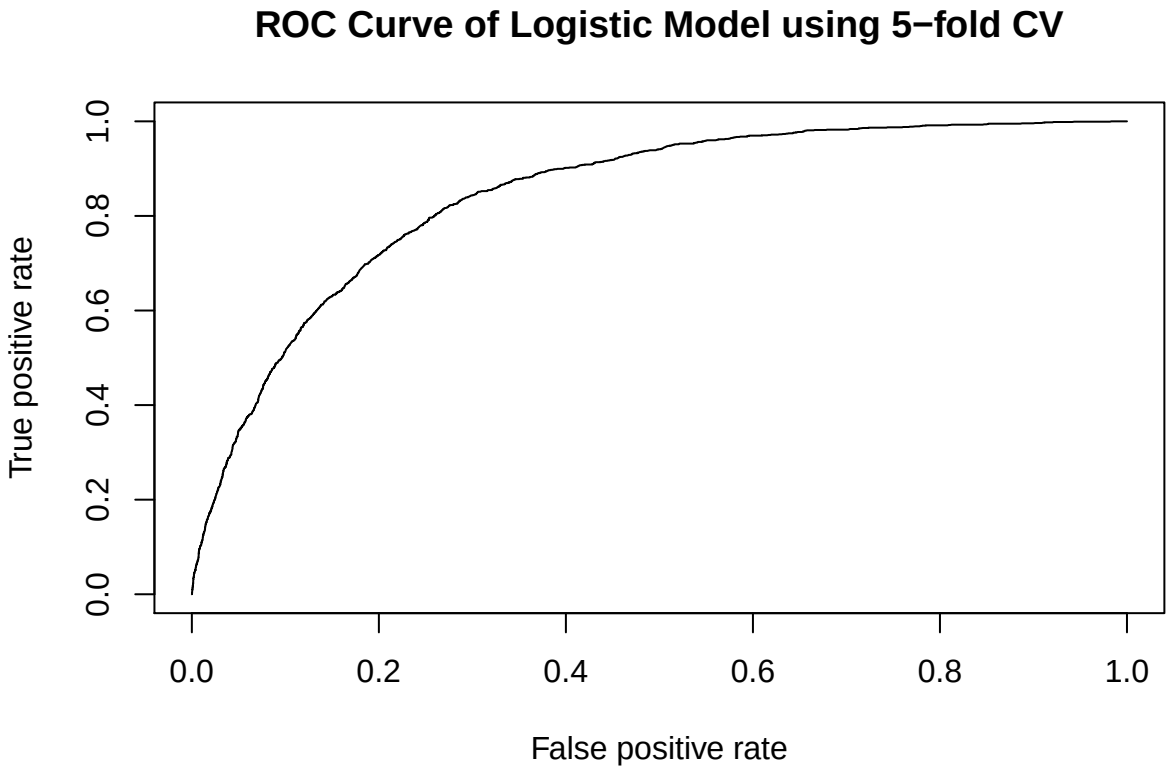
We also run a multicollinearity test and find no issues (Appendix: Logistic Regression, Model Diagnostics).

We use Pearson standardized residuals, with a cutoff of 3, to find outliers in our model. Overall, we find 299 outliers with the vast majority of those, 297, being individuals that are predicted to not have heart disease when in fact they do. This means that, in general, our model struggles to correctly predict those with heart disease more so than those who do not have heart disease (Appendix: Logistic Regression, Model Diagnostics). Logically it also makes sense that our model would struggle to predict positives because heart disease is a non homogeneous disease.

We then look for high leverage observations and are surprised to see 1023 in our dataset. At the same time, however, we use dffits and Cook's Distance to try to find influential observations and are able to find none with a cutoff of 1. The high leverage observation imply that there are many individuals with a unique combination of covariates, which does makes sense as this is a subsample of the entire US population. Yet, we believe that none prove to be influential because these observations become diluted in our much larger sample pool. If we used a smaller sample size, we would most likely expect for there to be less high leverage observations but perhaps more influential ones (Appendix: Logistic Regression, Model Diagnostics).

Overall, our model diagnostics had some results that were cause for concern, but when considering the vast sample size and the overall fit of the model we felt comfortable making predictions with these diagnostics in mind.

After running model diagnostics, we then move to our overall model performance. We run a ROC curve using 5 fold cross validation, and find an AUC of 0.8438352 (See code in Appendix: Logistic Regression, ROC Curve and AUC Value). While not terrible, we were hoping for an overall better AUC and prediction.



We then use the ROC curve, to determine a cutoff point for our final model. We want to make sure that we have a high sensitivity when we select a cutoff point because, for heart disease, it is much more important to tell someone they have heart disease when they do not rather than vice versa. Keeping this in mind, we look for cutoffs that have a high true positive rate, above 0,7, and a low but still high false positive rate, below 0.2. The cutoff we came to is 0.14 (Appendix: Logistic Regression, Choosing Cutoff For Final Model).

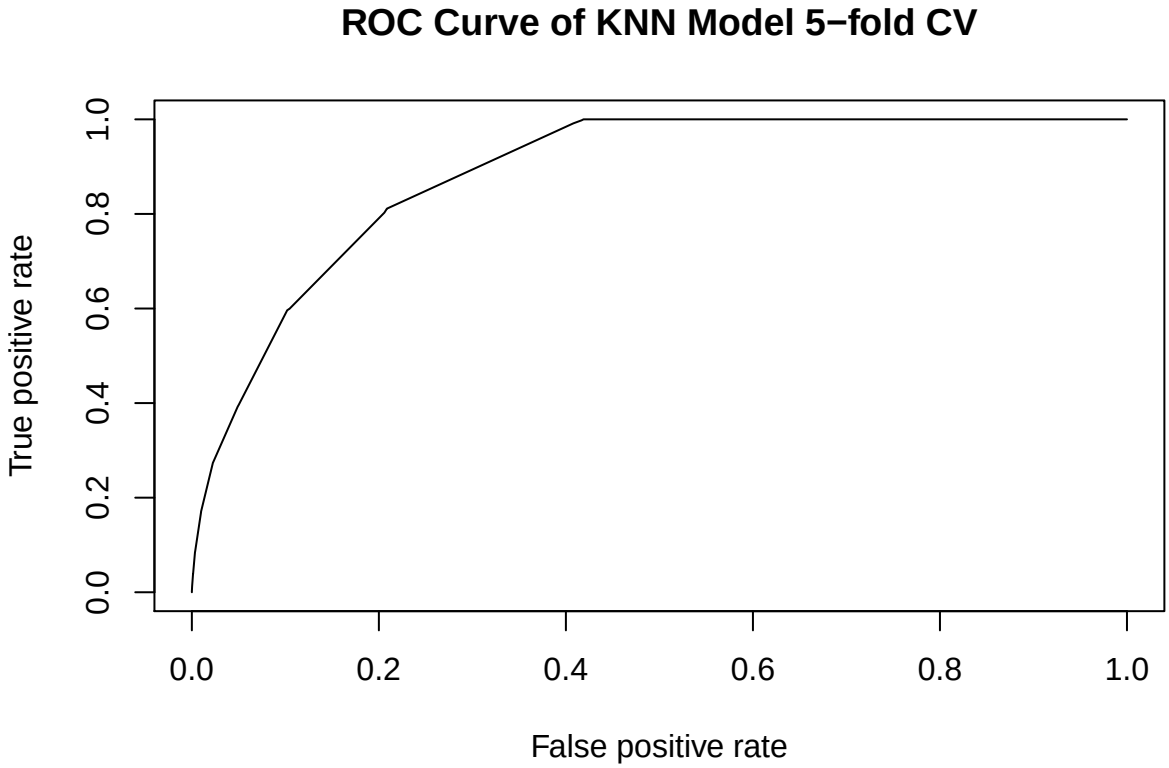
Using a 0.14 cutoff rate, we create our final prediction table (Appendix: Logistic Regression, Prediction Tables of Final Model).

Overall, the model's total accuracy is 81.55%, sensitivity is 65.9%, and specificity is 83.2%. We were hoping to have higher sensitivity. That said, considering how our model performs with the outliers and how the vast majority of our sample did not have heart disease, it makes sense that we struggle to predict true positives. For comparison, we also include a prediction table with a 0.5 cutoff rate. While this table has a higher accuracy, it has significantly worse sensitivity and proves why it was important to pick an optimal cutoff (Appendix: Logistic Regression, Prediction Tables of Final Model).

k-Nearest Neighbor

The next technique we use is k-Nearest Neighbor. We perform an initial analysis, using k 1 through 12 and 5 fold cross validation, to decide on which k to use and the k with the lowest error rate is 12 (Appendix:k-Nearest Neighbor, Choosing K). We decide not to go above 12, given the computational intensity of the technique. In turn, we create an ROC curve for the k-Nearest Neighbor with 5-fold cross validation and find an AUC of .886 (Appendix: k-Nearest Neighbor, ROC Curve and AUC Value).

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



We run an analysis of the cutoff rate very similar to the one that we conducted in logistic regression. There are actually no cutoffs that had a false positive rate less than 0.2 and a true positive rate higher than 0.7 so we increase the false positive rate to 0.25. We ultimately decide on a cutoff rate of 0.15 in order to balance overall accuracy with sensitivity (Appendix: k-Nearest Neighbor, Choosing Cutoff for Final Model).

Our final model has a total accuracy of 79.35%, sensitivity of 80.96%, and specificity of 79.18%. While we did ultimately have a lower total accuracy compared to the logistic regression model, we were pleased to see a much higher sensitivity rate (Appendix: k-Nearest Neighbor, Prediction Tables of Final Model).

We also try running a k-Nearest Neighbor with just the lasso selected variables because knn is sensitive to the noise of many variables. However, we find that this model has a lower AUC and lower sensitivity so we prefer the original knn with all the variables (Appendix: k-Nearest Neighbor, KNN with Lasso Variables).

Performance Comparison

In summary, the k-Nearest Neighbor performs slightly better than the logistic regression. A clear sign of this lies in the fact that the AUC is higher for the k-Nearest Neighbor, .886 versus .84. Despite the total accuracy being higher for logistic regression, we looked mostly at the sensitivity of the models as we value that for predicting heart disease. For sensitivity, k-Nearest Neighbor performed far better than logistic regression, 80.96% versus 65.9%. For all of these reasons, we believe that the k-Nearest Neighbor is the preferred supervised learning technique for this project.

Conclusion and Discussion

This project evaluates how well we can predict heart disease or attack given other health factors using survey responses from the CDC. We are able to address this question with various analyses. We try dimension reduction through Principal Component Analysis, and find that two principal components accounts for about 25% of the variability of the data. But overall, we find that principal component analysis on its own could not do much to separate those who had heart disease or attack from those who did not. Since PCA is trying

to capture the variability of the dataset, we do not expect it to separate the individuals with heart disease.

We then move on to supervised learning techniques and prefer k-Nearest Neighbor to the logistic regression. In order to do these, we perform a lasso penalized regression that informs us of the key variables for the regression. These are high blood pressure, high cholesterol, stroke, diabetes, general health, difficulty walking, sex, and age. This gives us critical insight into how health can affect heart disease. Moreover, the logistics regression model itself informs us that while stroke may be the strongest predictor of heart disease, there are several factors out of our control. Notably, that sex is the second strongest predictor in our model. The k-Nearest Neighbor provides stronger prediction with a higher AUC and higher sensitivity.

In conclusion, our project provides insight into the health conditions underlying heart disease, but there are limitations that research should continue to address in the path forward. The first limitation lies in the data itself. This data is survey data and therefore we are relying on the accuracy of the respondents. Many of these are health related variables and some people may be taking their own blood pressure incorrectly or lie about the number of times they engage in physical activity. We also only use less than 10% of the whole sample, so we might have eliminated some variability in doing so and should continue to test our results on the remainder of the sample. Finally our models are able to give fairly good results, but we do not recommend using our prediction model alone and further testing and analysis should be conducted to see if someone does in fact have heart disease.

We also find some model diagnostic issues with logistic regression surrounding the goodness of fit test and high-leverage observations that are not necessarily concerning but should be looked into further. Moreover, k-Nearest Neighbor, while valuable, is not transparent in what variables are contributing to its classification. And also, k-Nearest Neighbor is a weaker supervised model because the training algorithm requires that you store the entire training data, all 15,000 observations, while you only need the equation for the logistic regression model. This means that k-Nearest Neighbor would be both computationally and financially expensive to implement.

Our project provides some answers to the question, but techniques better suited to categorical variables might work better. We also had difficulty using decision trees and random forest but these methods would provide important supplementary evidence to the important variables that we selected in the process. We hope more research addresses this key question that affects more than 1 out of every 4 Americans.

Appendix

Variables

HighBP - Have you been told that you have high blood pressure by a doctor, nurse, or other health professional

HighChol - Have you ever been told by a doctor, nurse or other health professional that your blood cholesterol is high?

CholCheck - Have you had a cholesterol check within past five years?

BMI - Body Mass Index (BMI)

Smoker - Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes]

Stroke - Have you ever been told you have a stroke?

Diabetes - 0 is no diabetes, 1 is pre-diabetes, and 2 is diabetes

PhysActivity - Adults who reported doing physical activity or exercise during the past 30 days other than their regular job

Fruits - Consume Fruit 1 or more times per day

Veggies - Consume Vegetables 1 or more times per day

HvyAlcoholConsump - Heavy drinkers (adult men having more than 14 drinks per week and adult women having more than 7 drinks per week)

AnyHealthcare - Have you have any kind of health care coverage, including health insurance, prepaid plans such as HMOs?

NoDocbcCost - Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?

GenHlth - What would you say that in general your health is on a scale of 1-5?

MentHlth - Now thinking about your mental health, which includes stress, depression, and problems with emotions, for how many days during the past 30 days have you been worried about your mental health?

PhysHlth - Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days have you exercised?

DiffWalk - Do you have serious difficulty walking or climbing stairs?

Sex - Male or Female

Age - 12 Age Levels

Education - 6 Education Levels

Income - 8 Income Levels

Summary Statistics and Charts

Histograms for Numeric Variables

```
summary(heart)
```

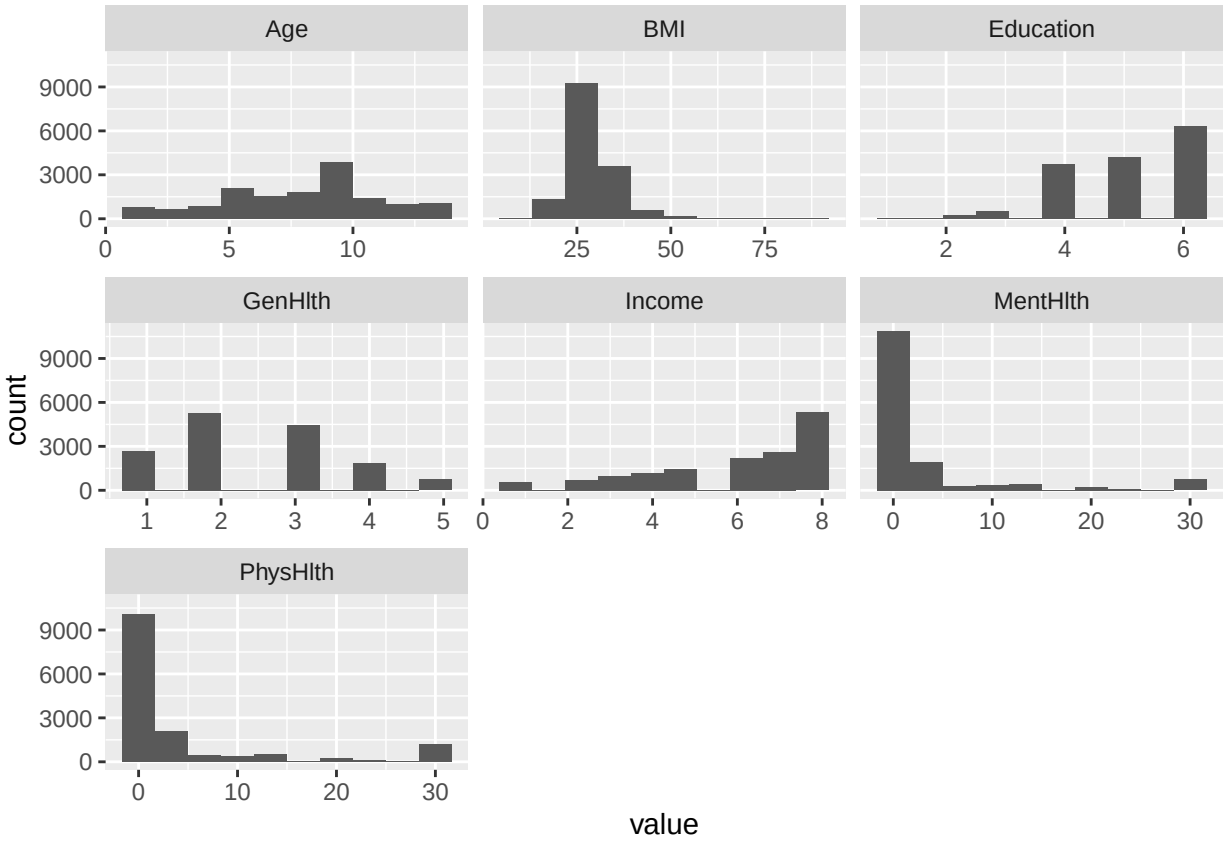
```
##  HeartDiseaseorAttack HighBP   HighChol CholCheck      BMI        Smoker
##  0:13576              0:8613   0:8718   0:  545   Min.   :13.00   0:8417
##  1: 1424              1:6387   1:6282   1:14455   1st Qu.:24.00   1:6583
##                                                  Median :27.00
##                                                  Mean   :28.46
##                                                  3rd Qu.:31.00
##                                                  Max.   :92.00
##  Stroke    Diabetes  PhysActivity Fruits   Veggies   HvyAlcoholConsump
##  0:14380   0:12628   0: 3708      0:5481   0: 2784   0:14171
##  1:  620   1:  292   1:11292      1:9519   1:12216   1:  829
##            2: 2080
##
##
##
##  AnyHealthcare NoDocbcCost    GenHlth         MentHlth         PhysHlth
##  0:  735       0:13691     Min.   :1.000   Min.   : 0.000   Min.   : 0.000
##  1:14265       1: 1309     1st Qu.:2.000   1st Qu.: 0.000   1st Qu.: 0.000
##                            Median :2.000   Median : 0.000   Median : 0.000
##                            Mean   :2.518   Mean   : 3.213   Mean   : 4.367
##                            3rd Qu.:3.000   3rd Qu.: 2.000   3rd Qu.: 3.000
##                            Max.   :5.000   Max.   :30.000   Max.   :30.000
##  DiffWalk  Sex           Age           Education        Income
##  0:12396   0:8429   Min.   : 1.000   Min.   :1.00   Min.   :1.000
##  1: 2604   1:6571   1st Qu.: 6.000   1st Qu.:4.00   1st Qu.:5.000
##                     Median : 8.000   Median :5.00   Median :7.000
##                     Mean   : 8.051   Mean   :5.05   Mean   :6.055
##                     3rd Qu.:10.000   3rd Qu.:6.00   3rd Qu.:8.000
##                     Max.   :13.000   Max.   :6.00   Max.   :8.000
```

```
library(ggplot2)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
```

```
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
num_cols <- unlist(lapply(heart, is.numeric))         # Identify numeric columns

heart_num <- heart[ , num_cols]                        # Subset numeric columns of data

ggplot(gather(heart_num), aes(value)) +
    geom_histogram(bins = 10) +
    facet_wrap(~key, scales = 'free_x')
```

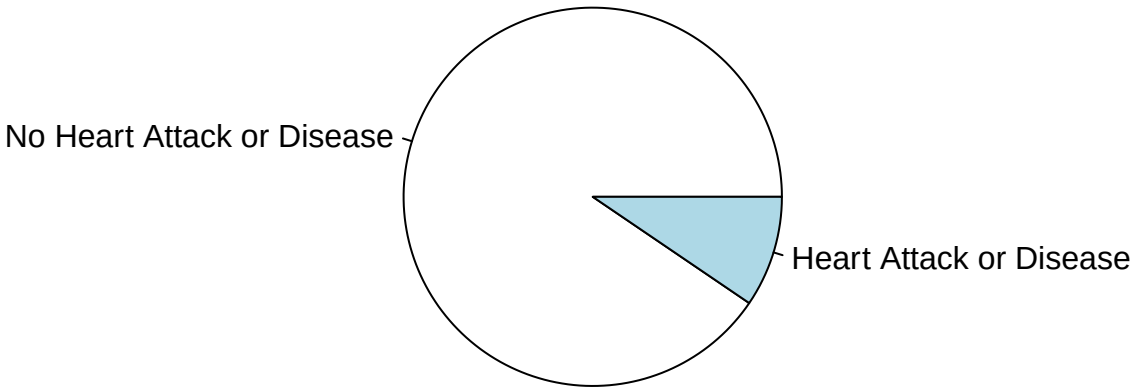


Pie Charts for Selected Factor Variables

```
summary(heart$HeartDiseaseorAttack)
```

```
##     0     1
## 13576  1424
```

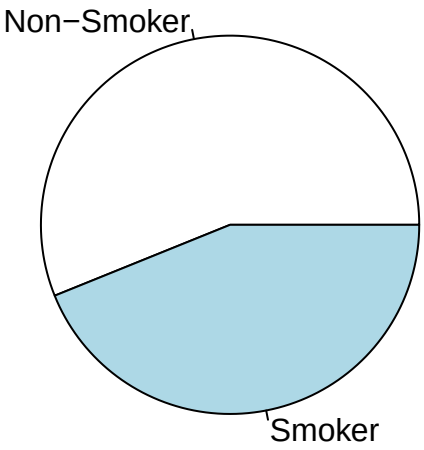
```
HeartDiseaseLabels <- c("No Heart Attack or Disease", "Heart Attack or Disease")
pie(table(heart$HeartDiseaseorAttack), main="Heart Attack Distribution", labels=HeartDiseaseLabels)
```




```
summary(heart$Smoker)
```

```
##    0    1
## 8417 6583
```

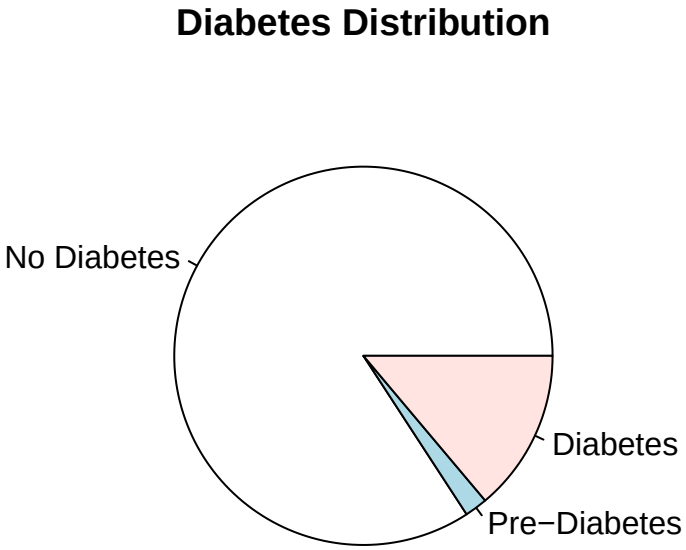
```
SmokerLabels <- c("Non-Smoker", "Smoker")
pie(table(heart$Smoker), main="Smoker Distribution", labels=SmokerLabels)
```




```
summary(heart$Diabetes)
```

```
##     0     1     2
## 12628   292  2080
```

```
DiabetesLabels <- c("No Diabetes", "Pre-Diabetes", "Diabetes")
pie(table(heart$Diabetes), main="Diabetes Distribution", labels=DiabetesLabels)
```

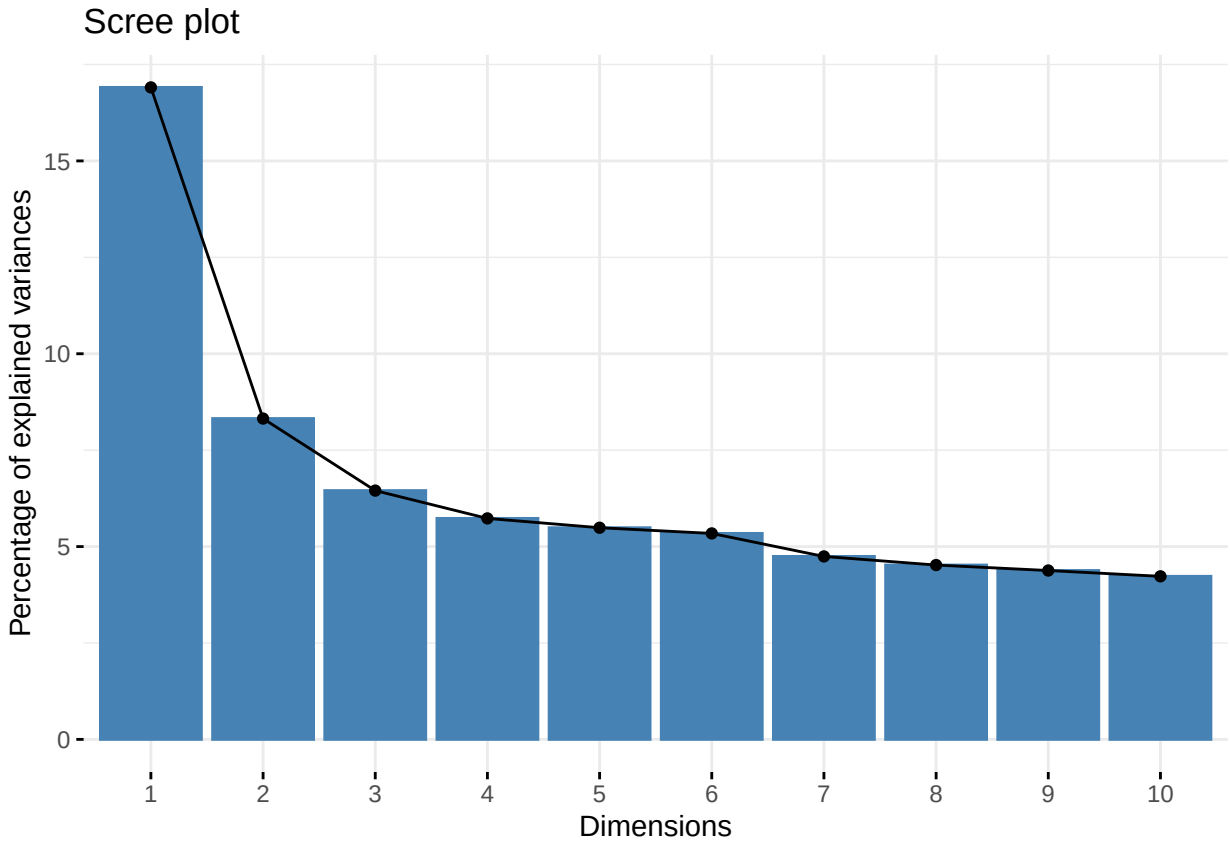


PCA

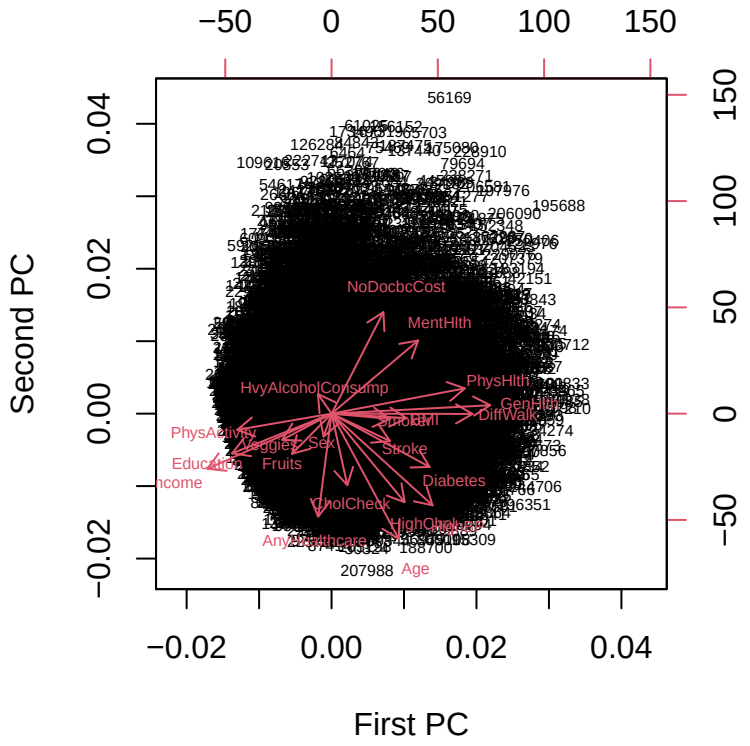
Scree plot and Biplot

A scree plot and biplot of the PCA.

```
fviz_eig(heart.pca)
```



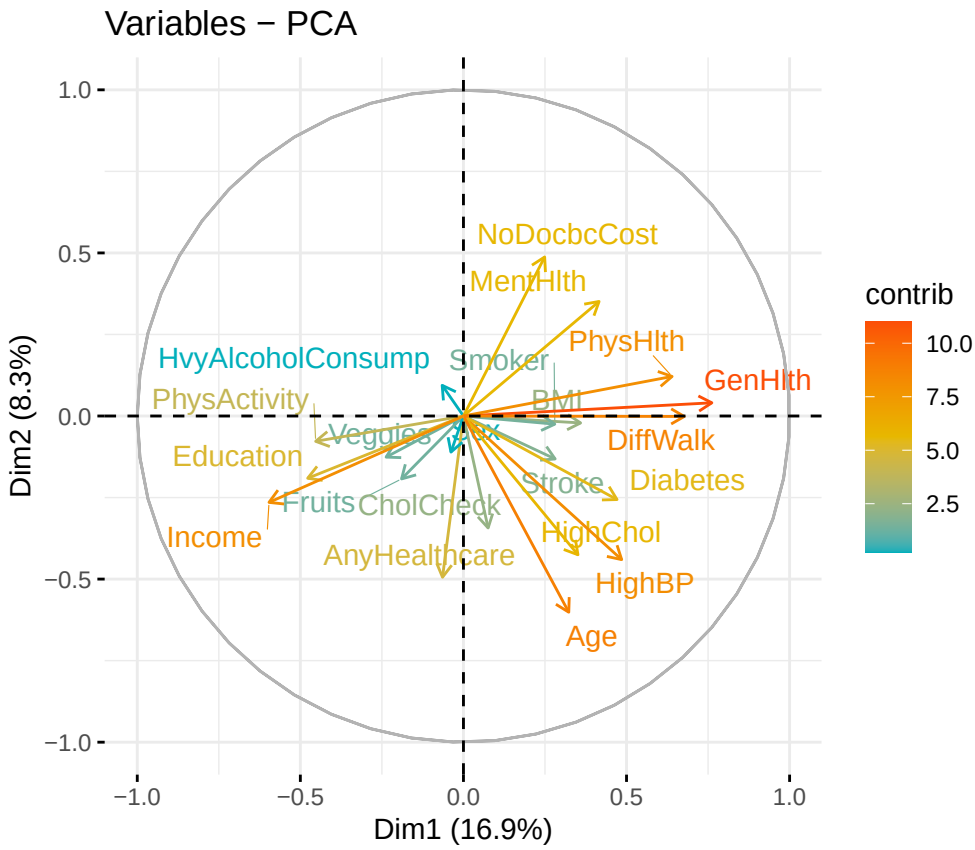

```
biplot(heart.pca , scale = T, cex=0.5, xlab="First PC", ylab="Second PC")
```



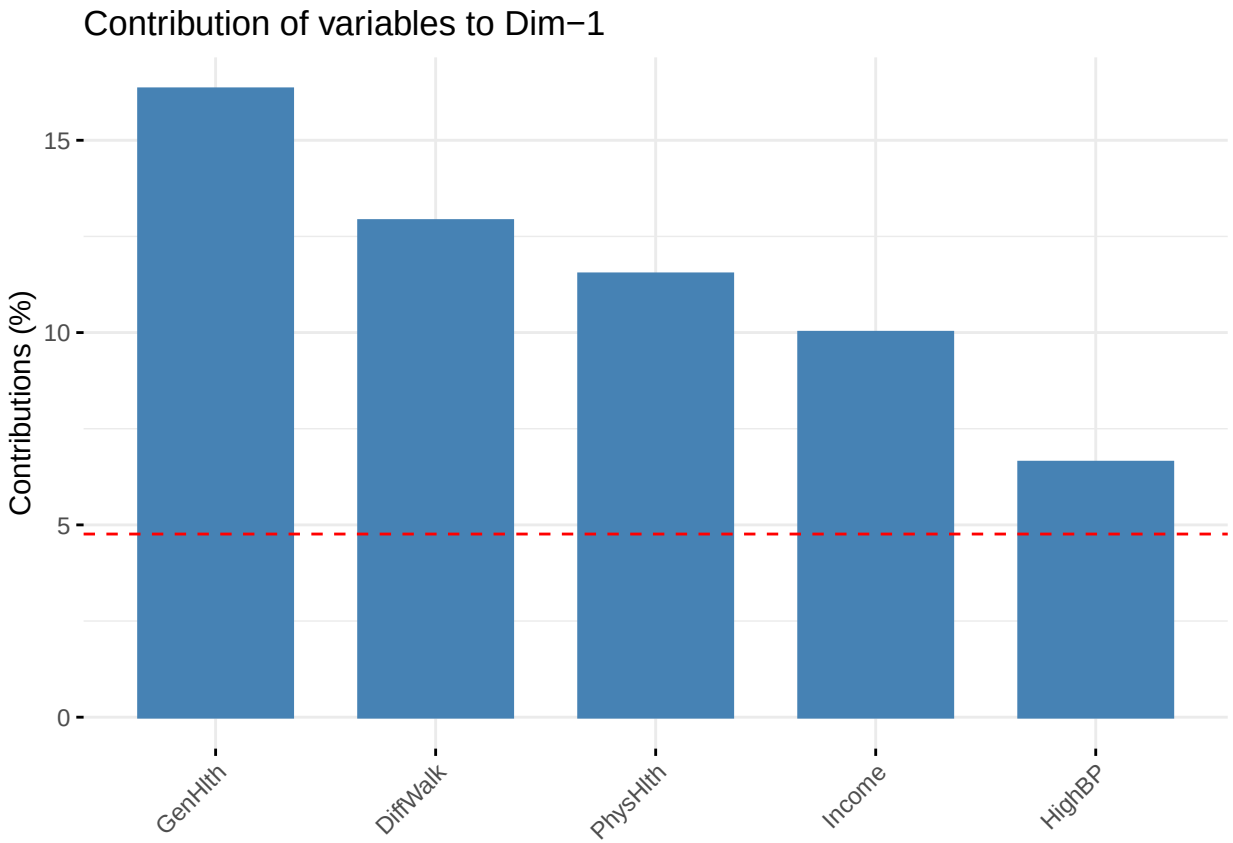
Contribution Summary

This section gives a summary of the contributions of the variables to the PC

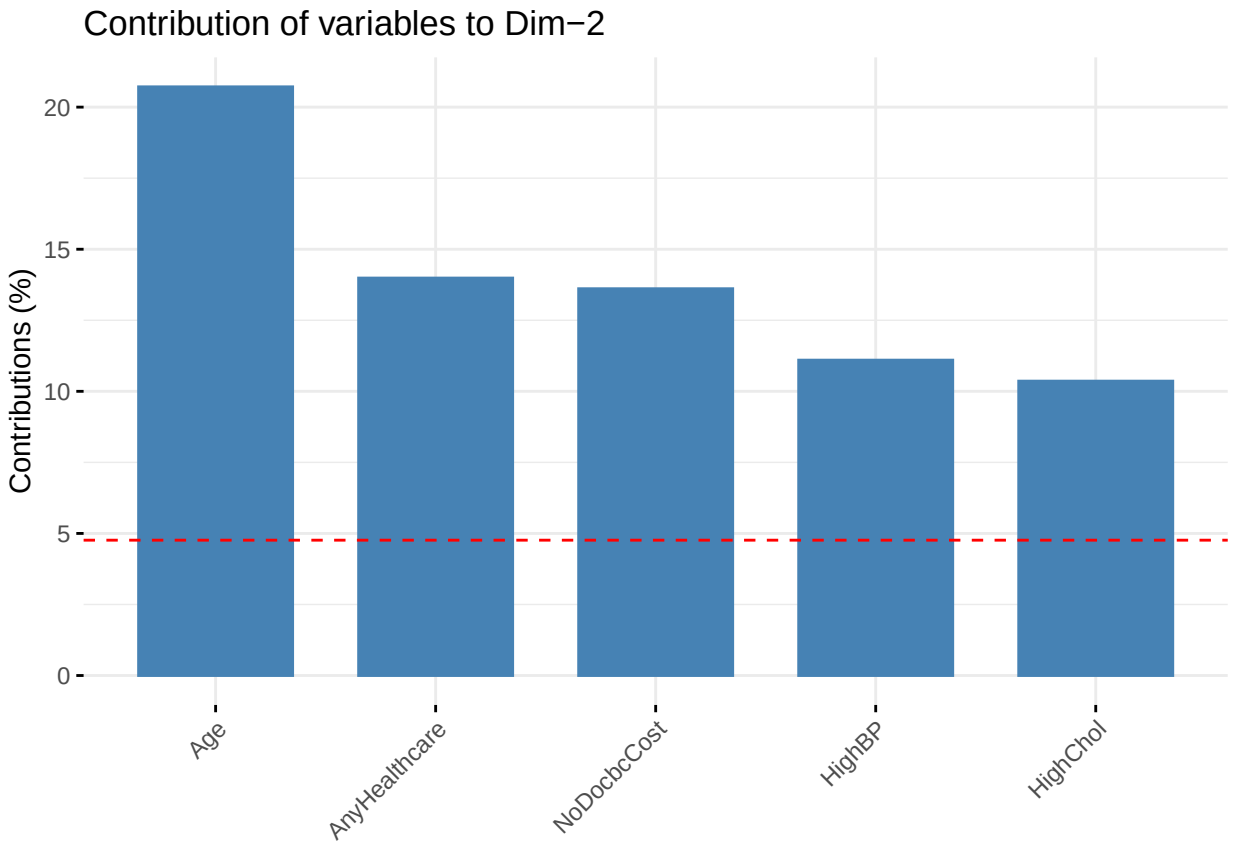
```
varar <- get_pca_var(heart.pca)

fviz_pca_var(heart.pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```




```
fviz_contrib(heart.pca, choice = "var", axes = 1, top = 5)
```




```
fviz_contrib(heart.pca, choice = "var", axes = 2, top = 5)
```



Class Separation

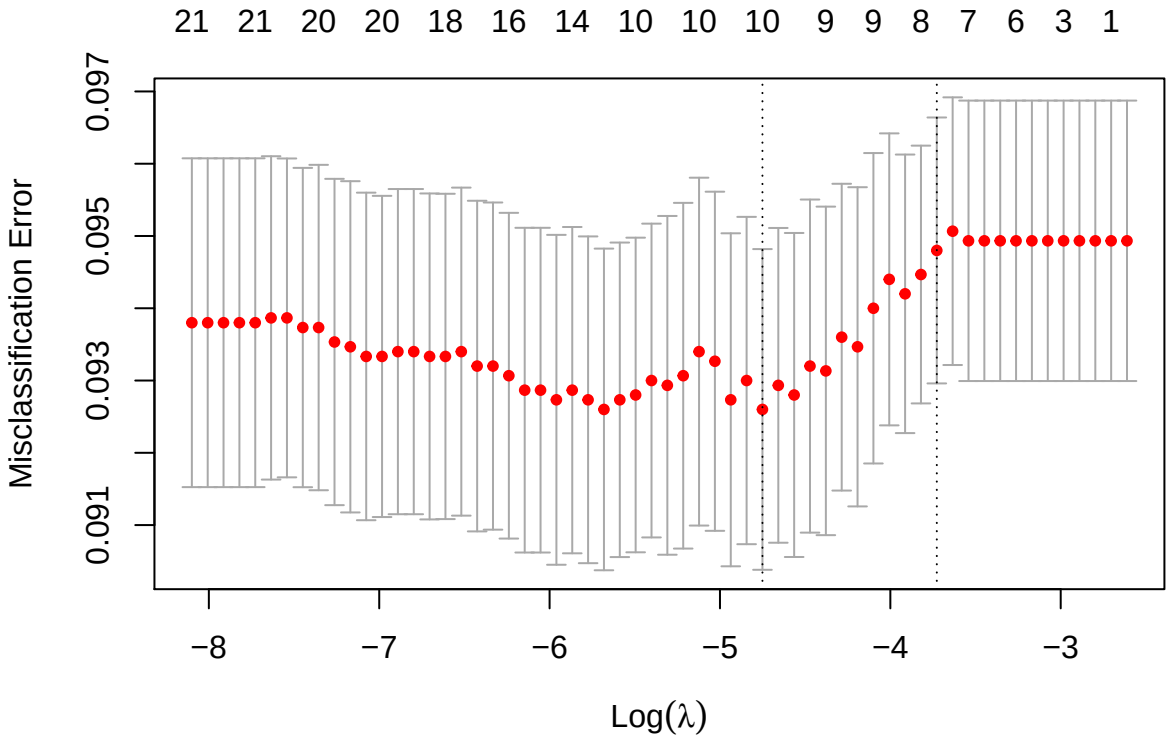
```
fviz_pca_ind(heart.pca,
label="none", # hide individual labels
habillage = heart.numeric$HeartDiseaseorAttack, # color by groups
addEllipses = TRUE, # Concentration ellipses
legend.title = "Heart Disease" )
```

Logistic Regression

Lasso Penalized Regression Variables

We use these results to predict our covariates for logistic regression. While diabetes1 was ultimately not a variable selected by the regression, diabetes2 was. And since the diabetes category comes together, we decided to include it in our model.

```
library(glmnet)

X = model.matrix(HeartDiseaseorAttack ~ ., data=heart)
Y = as.numeric(heart$HeartDiseaseorAttack)

set.seed(1234)
cvfit = cv.glmnet(x=X[,c(-1)], y=Y, family="binomial", type.measure="class")
plot(cvfit)
```




```
coef(cvfit, s=cvfit$lambda.1se)
```

```
## 23 x 1 sparse Matrix of class "dgCMatrix"
##                                s1
## (Intercept)        -4.72457024
## HighBP1             0.20195646
## HighChol1           0.21474173
## CholCheck1          .
## BMI                 .
## Smoker1             .
## Stroke1             0.79354852
## Diabetes1           .
## Diabetes2           0.09771233
## PhysActivity1       .
## Fruits1             .
## Veggies1            .
## HvyAlcoholConsump1  .
## AnyHealthcare1      .
## NoDocbcCost1        .
## GenHlth             0.35502232
## MentHlth            .
## PhysHlth            .
## DiffWalk1           0.13482635
## Sex1                0.10098976
## Age                 0.13056058
## Education           .
## Income              .
```

Model

Our lasso logistic model and coefficients. We still felt comfortable moving forward with diabetes1 despite its high p-value because the log-odds were by the far the smallest for any variable so it likely did not have much

of an effect.

```
fit.lasso <- glm(HeartDiseaseorAttack ~ HighBP+HighChol+Stroke+ Diabetes+ GenHlth+
                   Sex+ Age +DiffWalk,
                 family="binomial", data = heart)
summary(fit.lasso)
```

```
##
## Call:
## glm(formula = HeartDiseaseorAttack ~ HighBP + HighChol + Stroke +
##     Diabetes + GenHlth + Sex + Age + DiffWalk, family = "binomial",
##     data = heart)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1888  -0.4257  -0.2468  -0.1325   3.4355
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.45207    0.18176 -40.999  < 2e-16 ***
## HighBP1      0.45910    0.07058   6.505 7.79e-11 ***
## HighChol1    0.64006    0.06659   9.612  < 2e-16 ***
## Stroke1      1.16586    0.09870  11.812  < 2e-16 ***
## Diabetes1   -0.07513    0.19739  -0.381    0.703
## Diabetes2    0.29314    0.07269   4.033 5.51e-05 ***
## GenHlth      0.51279    0.03342  15.344  < 2e-16 ***
## Sex1         0.82400    0.06345  12.987  < 2e-16 ***
## Age          0.26391    0.01374  19.209  < 2e-16 ***
## DiffWalk1    0.37516    0.07485   5.012 5.39e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9414.2  on 14999  degrees of freedom
## Residual deviance: 7243.0  on 14990  degrees of freedom
## AIC: 7263
##
## Number of Fisher Scoring iterations: 6
```

Model Diagnostics

Testing for multicollinearity. Had to use GVIF^(1/(2*Df)) since diabetes has 2 degrees of freedom. No GVIF^(1/(2Df)) Values are above five so we can conclude that there is no multicollinearity.

```
library(car)
```

```
## Loading required package: carData
```

```
vif(fit.lasso)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## HighBP   1.118124  1        1.057414
## HighChol 1.056269  1        1.027749
## Stroke   1.028390  1        1.014096
## Diabetes 1.112331  2        1.026972
## GenHlth  1.367800  1        1.169530
```

```
## Sex      1.041855  1        1.020713
## Age      1.043690  1        1.021612
## DiffWalk 1.347774  1        1.160937
```

Hosmer-Lemeshow goodness-of-fit tests for our lasso model. The first test was the model with our full sample size but the second was for our model but with only 5000 observations.

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
##Full Sample Size
res = hoslem.test(fit.lasso$y, fit.lasso$fitted.values)
res
```

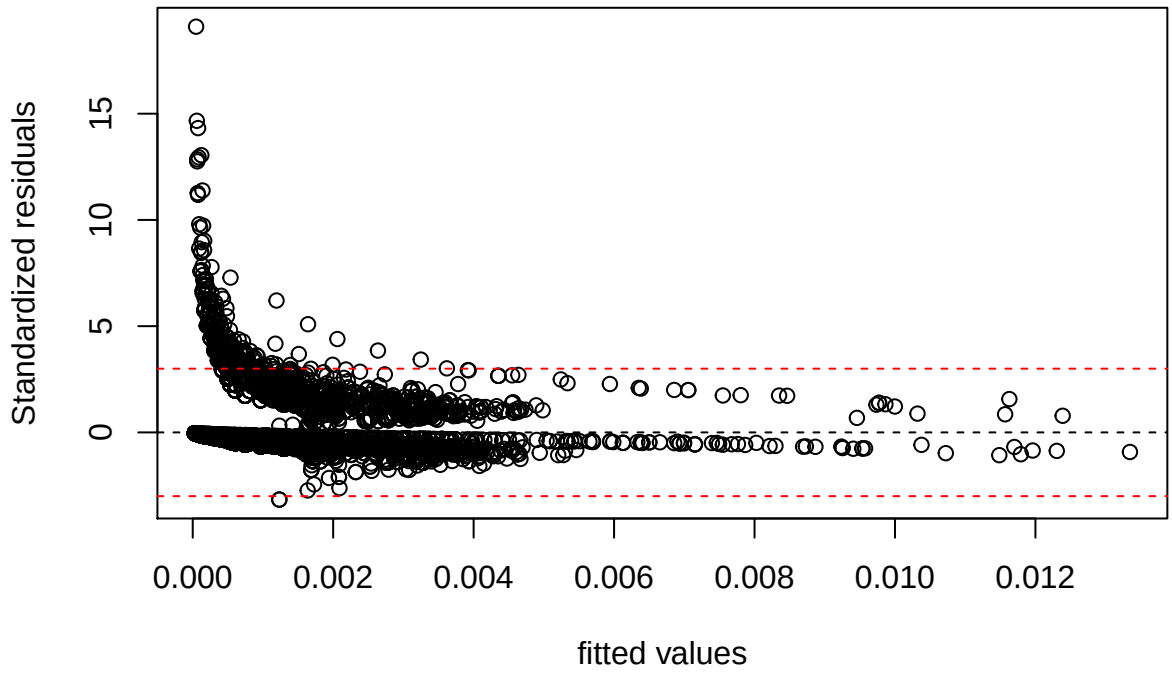
```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  fit.lasso$y, fit.lasso$fitted.values
## X-squared = 28.004, df = 8, p-value = 0.0004735
```

```
##Adjusted Sample Size Holsem Test
set.seed(1)
fit.lasso.small = glm(HeartDiseaseorAttack ~ HighBP+HighChol+Stroke+ Diabetes+ GenHlth+
                    Sex+ Age +DiffWalk,
                  family="binomial", data = heart[sample(nrow(heart), 5000), ])
res.small= hoslem.test(fit.lasso.small$y, fit.lasso.small$fitted.values)
res.small
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  fit.lasso.small$y, fit.lasso.small$fitted.values
## X-squared = 14.473, df = 8, p-value = 0.07025
```

Identifying outliers through Pearson standard residuals

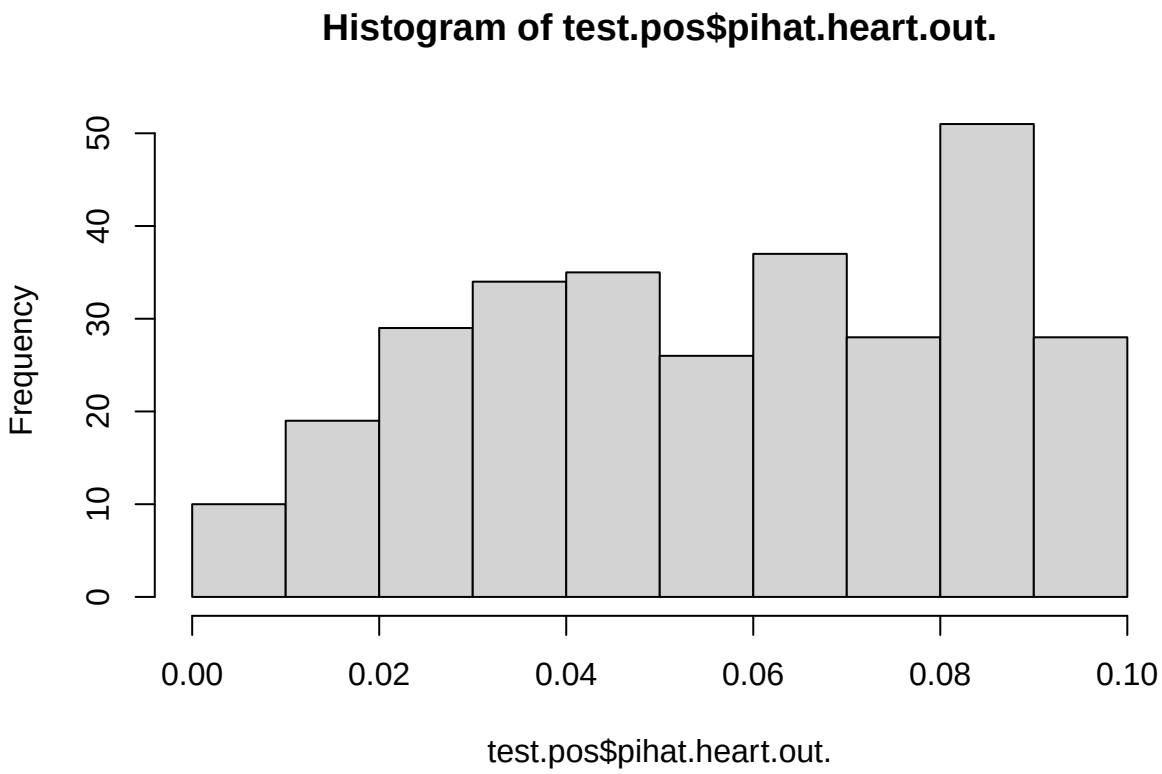
```
e = rstandard(fit.lasso, type = "pearson")
hi = hatvalues(fit.lasso)
plot(e ~ hi,
xlab="fitted values", ylab="Standardized residuals")
abline(0,0, lty=2)
abline(-3, 0, lty=2, col="red")
abline(3, 0, lty=2, col="red")
```




```
##Selecting Outliers and Table of Positive and Negative Values
heart.out= which(abs(e)>3)
pihat= predict(fit.lasso, type="response")
test= data.frame(heart$HeartDiseaseorAttack[heart.out], pihat[heart.out],
           e[heart.out])
table(test$heart.HeartDiseaseorAttack.heart.out.)
```

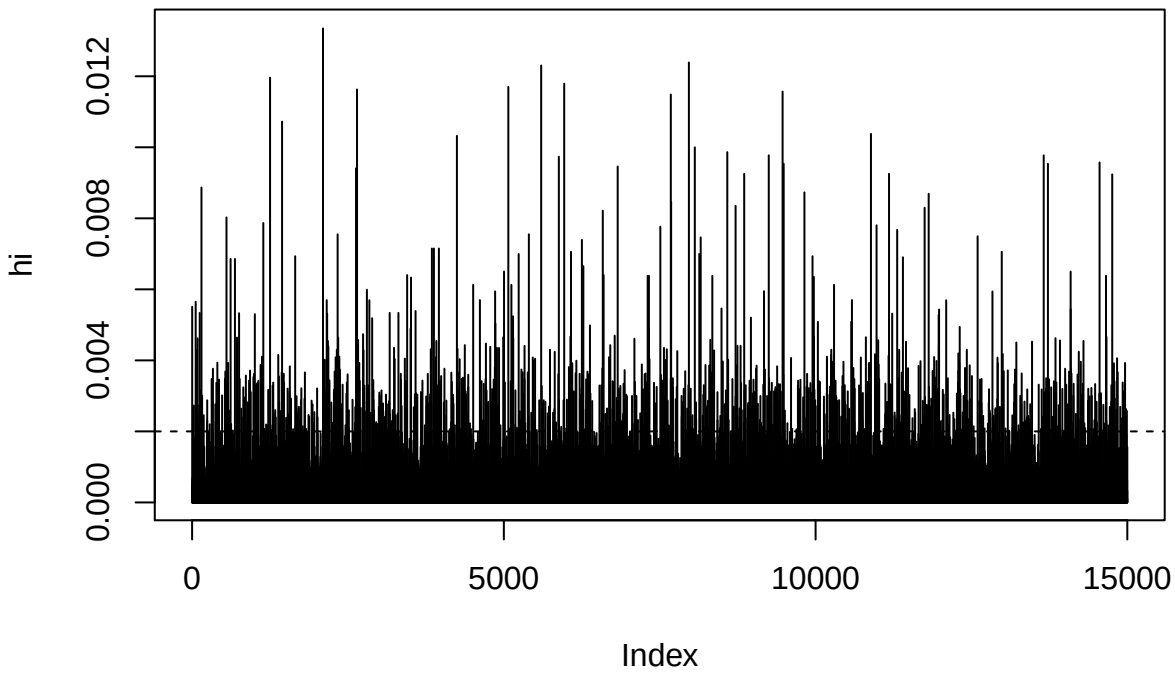
```
##
##   0   1
##   2 297
```

```
##Histogram of Probabilities
test.pos= test[which(test$heart.HeartDiseaseorAttack.heart.out.==1),]

hist(test.pos$pihat.heart.out.)
```



Identifying High Leverage Values

```
##Hat Values
hi = hatvalues(fit.lasso)
plot(hi, type="h")
abline(h= 3*mean(hi), lty=2)
```



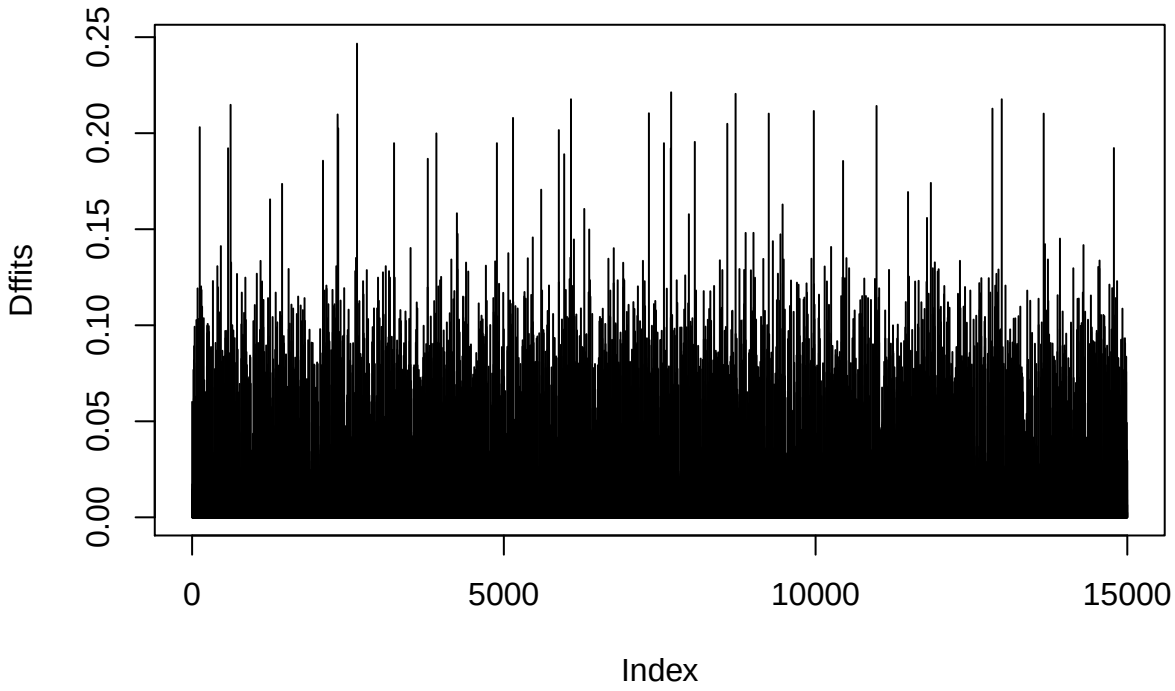

```
highleverage= cbind(heart[, ], hi)[hi>3*mean(hi),]
```

```
##Number of High Leverage Observations
nrow(highleverage)
```

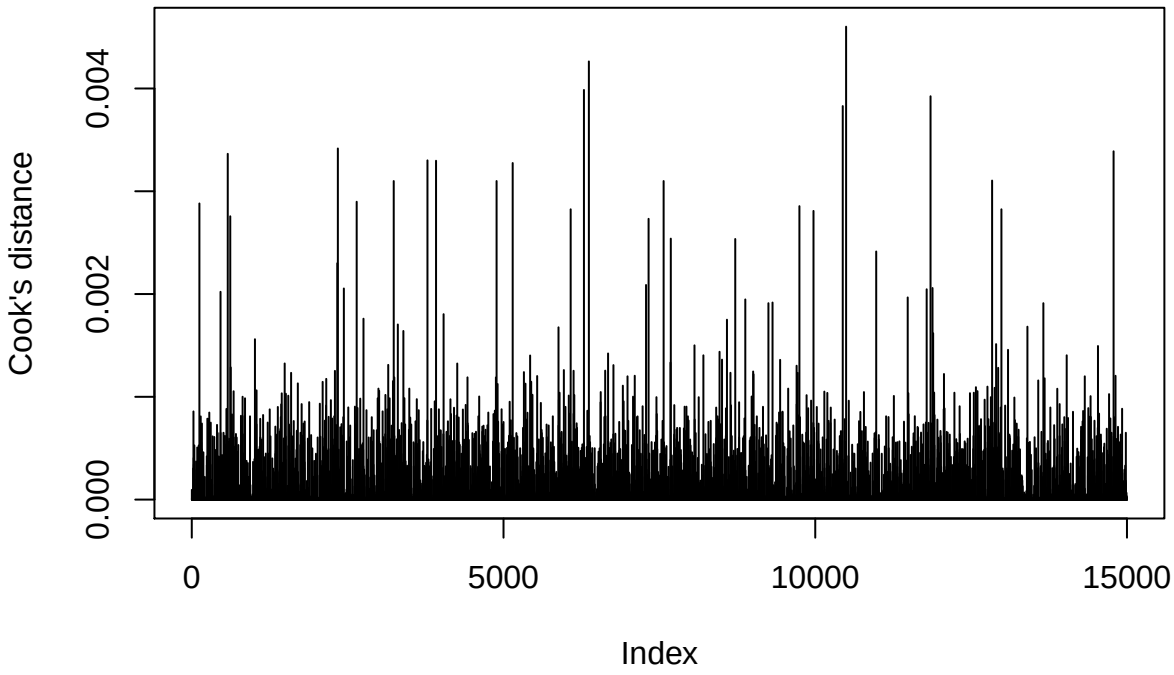
```
## [1] 1023
```

No Influential Observations Using dffits and Cooks Distance. Both using cutoff of 1.

```
##Dffits
dfyhat <- dffits(fit.lasso)
plot(abs(dfyhat), type= "h",  ylab="Dffits")
```

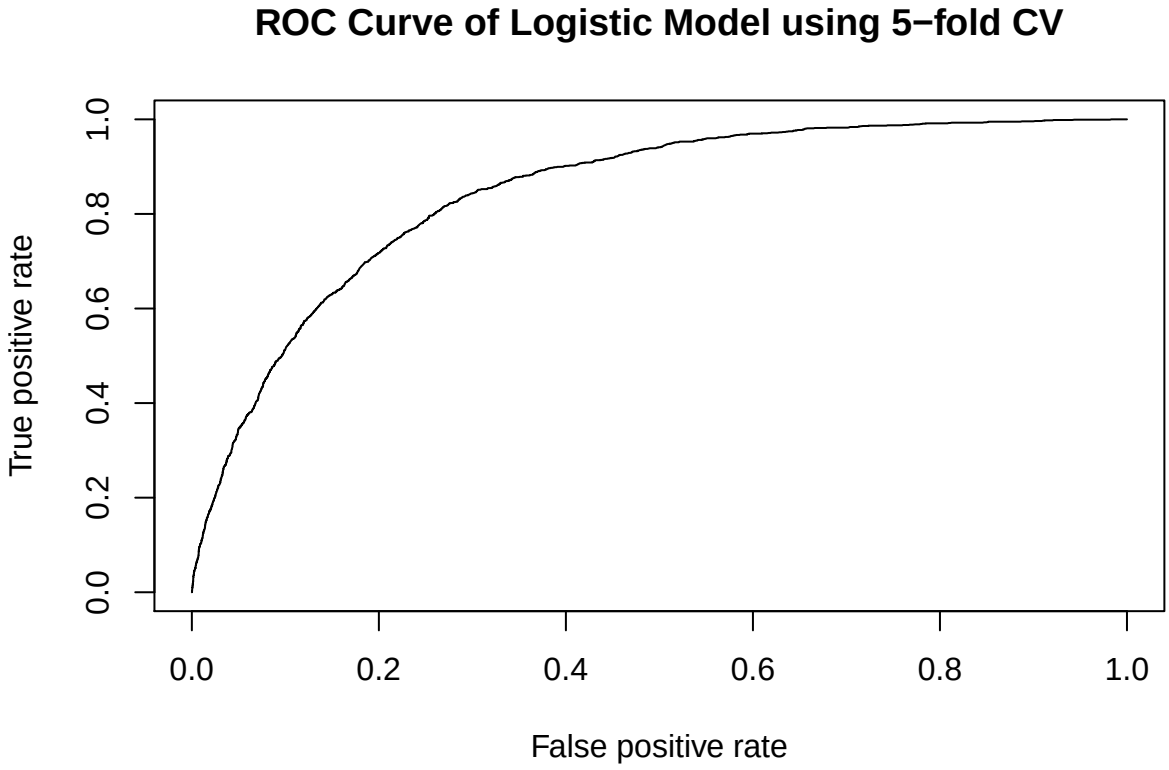



```
##Cooks Distance
cdist <- cooks.distance(fit.lasso)
plot(cdist, type="h", ylab="Cook's distance")
```



ROC Curve and AUC Value

```
library(glmnet)
library(ROCR)
set.seed(1)
n <- nrow(heart)
nrep <- 1 # repetitions of k-fold CV
kfolds <- 5 # 5-fold CV
cv.pred <- matrix(NA, nrow=n, ncol=nrep)
for(j in 1:nrep) {
  folds.i <- sample(rep(1:kfolds, length= n))
  for (k in 1:kfolds) {
    test.i = which(folds.i == k)
    train.dat = heart[-test.i, ]
    test.dat = heart[test.i, ]
    fit.train = glm(HeartDiseaseorAttack ~HighBP+HighChol+Stroke+ Diabetes+ GenHlth+
                    Sex+ Age +DiffWalk,
                    family="binomial", data=train.dat)
    cv.pred[test.i,j] = predict(fit.train, newdata=test.dat) }}

pred = prediction(cv.pred, matrix(rep(heart$HeartDiseaseorAttack,nrep),
                                  ncol=nrep))
perf = performance(pred, "tpr", "fpr")
plot(perf, main= "ROC Curve of Logistic Model using 5-fold CV")
```



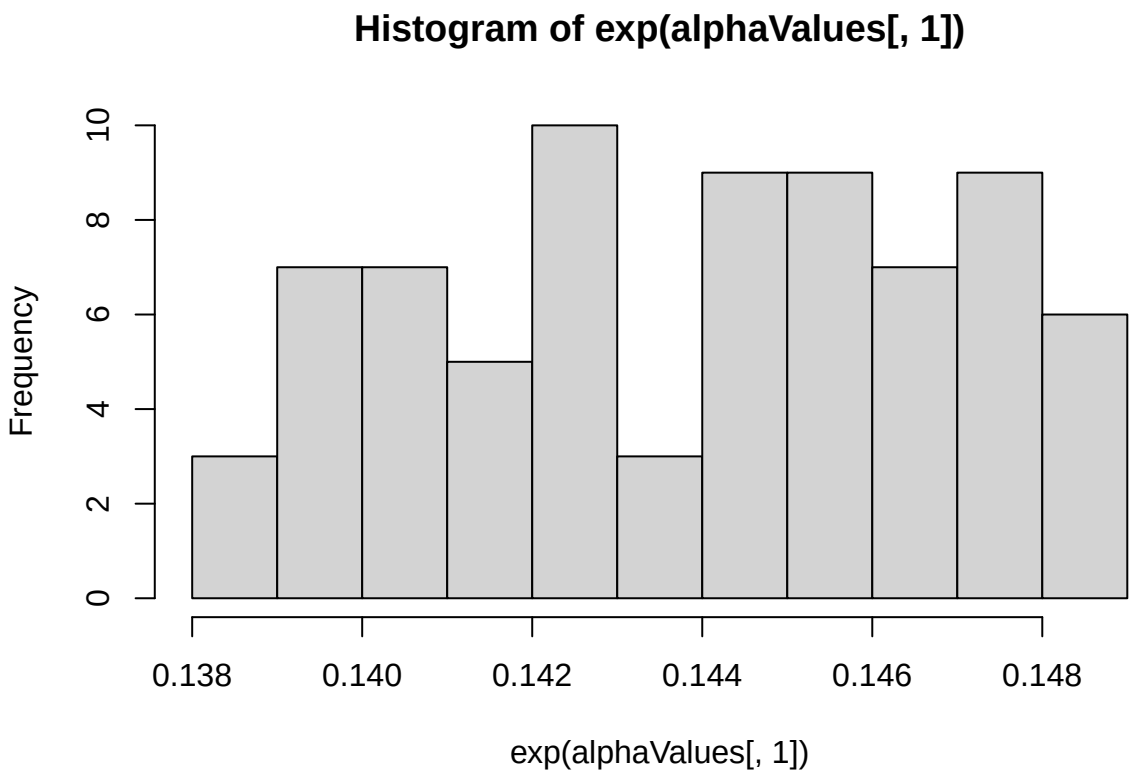

```
auc.perf = performance(pred, "auc")@y.values
unlist(auc.perf)
```

```
## [1] 0.8438352
```

Choosing Cutoff For Final Model

Since we had a min of 0.1389 and a max of 0.148, with our specific attributes, we decided to use a clean cutoff of 0.14.

```
alphaWhich= which(perf@x.values[[1]] < 0.2 & perf@y.values[[1]] >0.7)
alphaValues= cbind(perf@alpha.values[[1]], perf@x.values[[1]], perf@y.values[[1]])[alphaWhich,]
alphaValues= as.data.frame(alphaValues)
hist(exp(alphaValues[,1]))
```




```
##Min Cuttoff
exp(min(alphaValues[,1]))
```

```
## [1] 0.1389491
```

```
##Max Cuttoff
exp(max(alphaValues[,1]))
```

```
## [1] 0.1486268
```

Prediction Tables of Final Model

Table of final model with cutoff of 0.14

```
pihat.test= predict(fit.lasso, type="response")
yhat.test <- pihat.test>0.14
table(heart$HeartDiseaseorAttack, yhat.test)
```

```
##    yhat.test
##     FALSE  TRUE
##   0 11299  2277
##   1   485   939
```

Table of final model with cutoff of 0.5

```
pihat.test= predict(fit.lasso, type="response")
yhat.test <- pihat.test>0.5
table(heart$HeartDiseaseorAttack, yhat.test)
```

```
##    yhat.test
##     FALSE  TRUE
##   0 13411   165
##   1  1246   178
```

K Nearest Neighbor

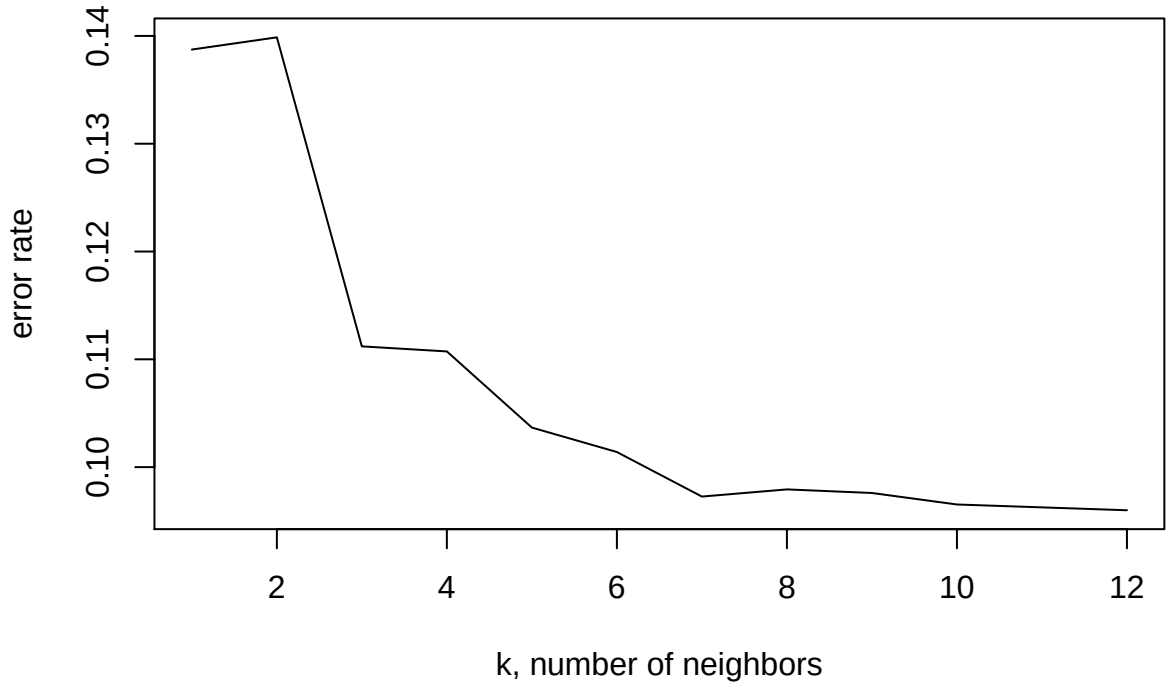
Choosing K

We found that 12 neighbors had the lowest error rate for our model.

```
library(caret)
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x))) }
heart.norm= apply(heart.numeric[,-1], 2, normalize)
heart.norm = data.frame(heart.numeric$HeartDiseaseorAttack, heart.norm)
names(heart.norm) <- names(heart.numeric)


# 5-fold CV to choose k
set.seed(1)
fit <- train(as.factor(HeartDiseaseorAttack) ~ .,
             method = "knn",
             tuneGrid = expand.grid(k = 1:12),
             trControl = trainControl(method="cv", number=5),
             metric = "Accuracy",
             data = heart.norm)
fit
```

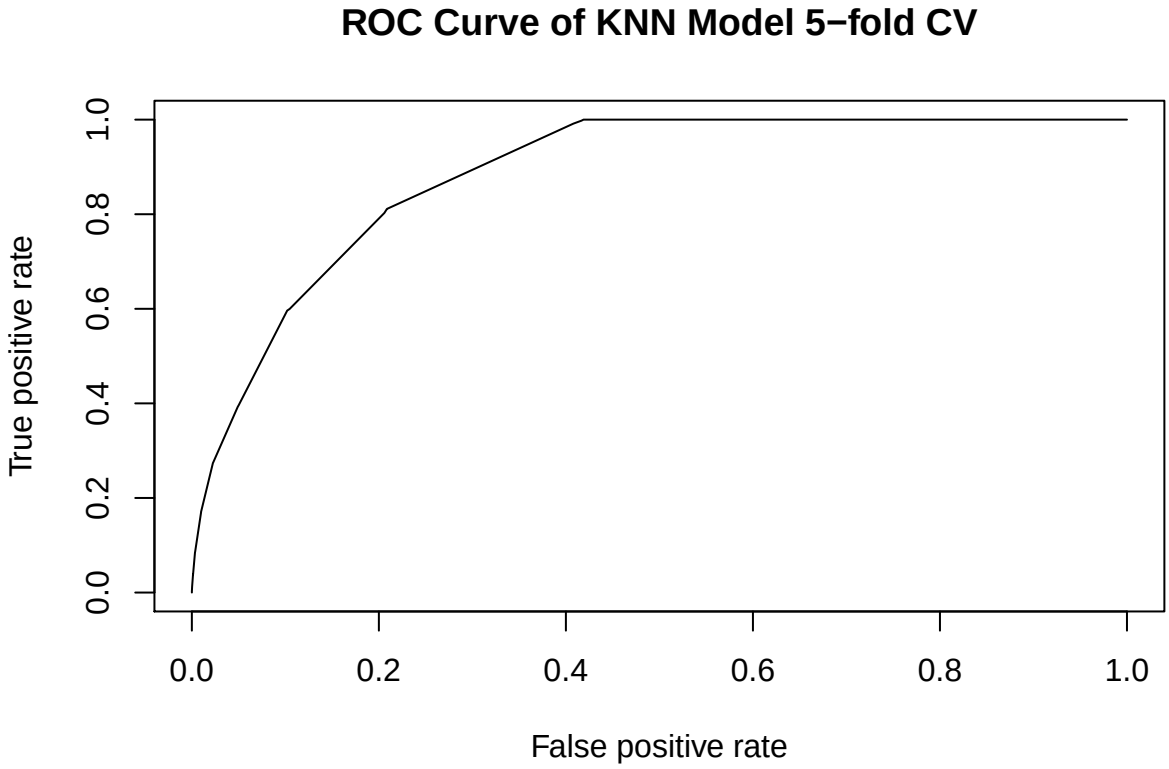
```
## k-Nearest Neighbors
##
## 15000 samples
##    21 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 12000, 12000, 12001, 11999, 12000
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    1  0.8612659  0.1742409
##    2  0.8601330  0.1630382
##    3  0.8887998  0.1581342
##    4  0.8892665  0.1624904
##    5  0.8963331  0.1341546
##    6  0.8985998  0.1554122
##    7  0.9027331  0.1524016
##    8  0.9020663  0.1322776
##    9  0.9023997  0.1197473
##   10  0.9034663  0.1293950
##   11  0.9037333  0.1074285
##   12  0.9040002  0.1079979
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 12.
```

```
plot(fit$results[,1], 1-fit$results[,2], type="l",
xlab="k, number of neighbors", ylab="error rate")
```



ROC Curve and AUC

```
pihat= predict(fit, type= "prob")
knn.pred= prediction(pihat[,2], heart.norm$HeartDiseaseorAttack)
knn.perf= performance(knn.pred, "tpr", "fpr")
plot(knn.perf, main= "ROC Curve of KNN Model 5-fold CV")
```



```
auc.perf = performance(knn.pred, "auc")@y.values
unlist(auc.perf)
```

```
## [1] 0.8852507
```

Choosing Cutoff For Final Model

Since we had a min of 0.133 and a max of 0.167, using our specific parameters, we decided to use a clean cutoff of 0.15.

```
alphaWhich= which(knn.perf@x.values[[1]] < 0.25 & knn.perf@y.values[[1]] >0.7)
alphaValues= cbind(knn.perf@alpha.values[[1]], knn.perf@x.values[[1]], knn.perf@y.values[[1]])[alphaWhi
alphaValues= as.data.frame(alphaValues)

##Min Cuttoff
min(alphaValues[,1])
```

```
## [1] 0.1333333
```

```
##Max Cuttoff
max(alphaValues[,1])
```

```
## [1] 0.1666667
```

Prediction Tables of Final Model

Final Table With Cutoff of 0.15

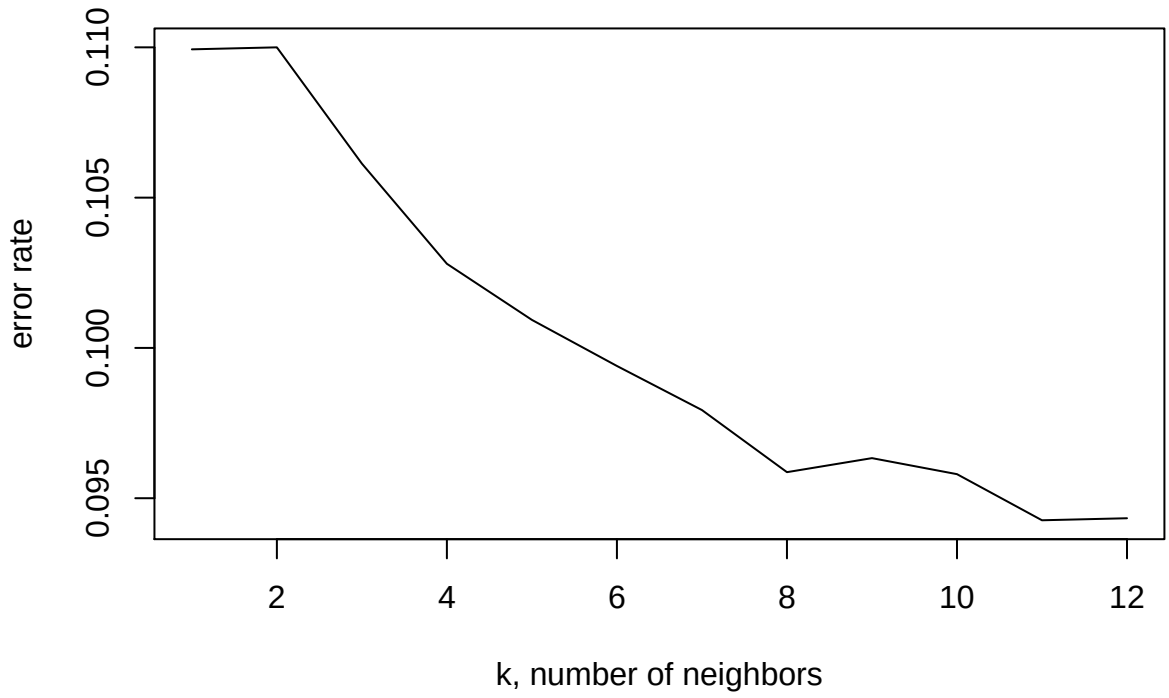
```
yhat.test <- pihat[,2] > 0.15
table(heart.norm$HeartDiseaseorAttack, yhat.test)
```

```
##    yhat.test
##     FALSE  TRUE
##   0 10750  2826
##   1   271  1153
```

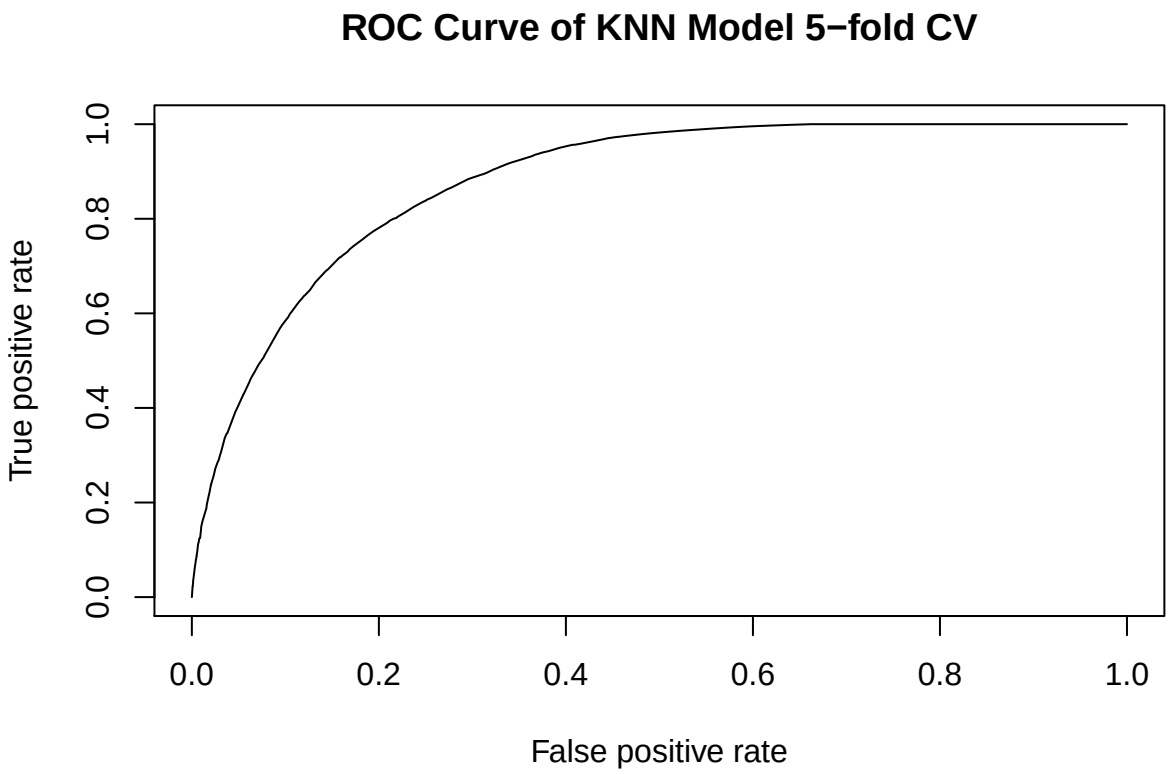
KNN with Lasso Variables

K-Nearest Neighbor with only Lasso Predicted Values

```
library(caret)
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x))) }
heart.norm= apply(heart.numeric[,-1], 2, normalize)
heart.norm = data.frame(heart.numeric$HeartDiseaseorAttack, heart.norm)
names(heart.norm) <- names(heart.numeric)
heart.norm= heart.norm[, c("HeartDiseaseorAttack", "HighBP", "HighChol", "Stroke", "Diabetes", "GenHlth
             ,"Age", "DiffWalk")]

# 5-fold CV to choose k for lasso model
set.seed(1)
fit <- train(as.factor(HeartDiseaseorAttack) ~ .,
             method = "knn",
             tuneGrid = expand.grid(k = 1:12),
             trControl = trainControl(method="cv", number=5),
             metric = "Accuracy",
             data = heart.norm)
plot(fit$results[,1], 1-fit$results[,2], type="l",
xlab="k, number of neighbors", ylab="error rate")
```




```
#ROC Curve of Lasso Model
pihat= predict(fit, type= "prob")
knn.pred= prediction(pihat[,2], heart.norm$HeartDiseaseorAttack)
knn.perf= performance(knn.pred, "tpr", "fpr")
plot(knn.perf, main= "ROC Curve of KNN Model 5-fold CV")
```




```
auc.perf = performance(knn.pred, "auc")@y.values
```

```
unlist(auc.perf)
```

```
## [1] 0.8789011
```

```
#
```

Choosing a Cutoff for Final Lasso Model

```
alphaWhich= which(knn.perf@x.values[[1]] < 0.25 & knn.perf@y.values[[1]] >0.7)
alphaValues= cbind(knn.perf@alpha.values[[1]], knn.perf@x.values[[1]], knn.perf@y.values[[1]])[alphaWhi
alphaValues= as.data.frame(alphaValues)

##Min Cuttoff
min(alphaValues[,1])
```

```
## [1] 0.1016949
```

```
##Max Cuttoff
max(alphaValues[,1])
```

```
## [1] 0.1666667
```

Final Lasso Table with Cutoff of 0.13

```
yhat.test <- pihat[,2] > 0.13
table(heart.norm$HeartDiseaseorAttack, yhat.test)
```

```
##    yhat.test
##     FALSE  TRUE
##   0 10972  2604
##   1   327  1097
```