Week 1 Data Prepration

Kevin McBeth, Keunjoo Park, Ryan Knudson-Fitzpatrick

January 16, 2018

Introduction

For this walkthrough, we will be cleaning a heart disease data set which has assigned -9's for it's missing values. These values will be imputed using k nearest neighbors in the DMwR package.

```
library(data.table)
library(DMwR)
library(corrplot)
library(raster)
library(caret)
library(FNN)
library(e1071)
```

Loading the Data

Loading the data, and converting to a data table.

```
fn <- 'C:\\Users\\Kevin\\Desktop\\Workspaces\\R\\MSDS680\\data\\heart.disease.data'

# as.is leaves characters as characters instead of converting to factors, so we can substitute NA for ?
df <- read.csv(fn, as.is = T)
heart.dt <- as.data.table(df)
```

Basic Exploratory Data Analysis

Basic exploratory data analysis, including looking at the structure, summary statistics, and the partiuclar 'missing' values of -9 are located.

```
str(heart.dt)
```

```
## Classes 'data.table' and 'data.frame':   282 obs. of  15 variables:
##  $ age     : int  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : int  1 1 1 1 0 1 0 0 1 1 ...
##  $ cp      : int  1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: int  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : int  233 286 229 250 204 236 268 354 254 203 ...
##  $ cigs    : int  50 40 20 0 0 20 0 0 0 20 ...
##  $ years   : int  20 40 35 0 0 20 0 0 0 25 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 0 1 ...
##  $ dm      : int  -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 ...
##  $ famhist : int  1 1 1 1 1 1 1 1 0 1 ...
##  $ restecg : int  2 2 2 0 2 0 2 0 2 2 ...
##  $ thalach : int  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : int  0 1 1 0 0 0 0 1 0 1 ...
##  $ thal    : int  6 3 7 3 3 3 3 3 7 7 ...
##  $ num     : int  0 2 1 0 0 0 3 0 2 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(heart.dt)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.41   Mean   :0.6773   Mean   :3.163   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##       chol            cigs           years            fbs
##  Min.   :126.0   Min.   :-9.00   Min.   :-9.00   Min.   :0.0000
##  1st Qu.:213.0   1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.:0.0000
##  Median :244.0   Median :10.00   Median :15.00   Median :0.0000
##  Mean   :249.1   Mean   :16.46   Mean   :14.83   Mean   :0.1489
##  3rd Qu.:277.0   3rd Qu.:30.00   3rd Qu.:30.00   3rd Qu.:0.0000
##  Max.   :564.0   Max.   :99.00   Max.   :54.00   Max.   :1.0000
##        dm            famhist          restecg         thalach
##  Min.   :-9.000   Min.   :0.0000   Min.   :0.000   Min.   : 71.0
##  1st Qu.:-9.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:133.2
##  Median :-9.000   Median :1.0000   Median :2.000   Median :153.5
##  Mean   :-8.184   Mean   :0.6206   Mean   :1.014   Mean   :149.8
##  3rd Qu.:-9.000   3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:165.8
##  Max.   : 1.000   Max.   :1.0000   Max.   :2.000   Max.   :202.0
##      exang             thal             num
##  Min.   :0.0000   Min.   :-9.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 3.000   1st Qu.:0.0000
##  Median :0.0000   Median : 3.000   Median :0.0000
##  Mean   :0.3262   Mean   : 4.582   Mean   :0.9078
##  3rd Qu.:1.0000   3rd Qu.: 7.000   3rd Qu.:2.0000
##  Max.   :1.0000   Max.   : 7.000   Max.   :4.0000
```

Data Cleaning

Cleaning the data by removing the dm column due to too many missing data portions, setting all -9 values to NA, looking at the summary statistics, imputing the data with missing values, and then changing the target variable, num to either 0 or 1. 1 if the value is greater than 1. Finally, converting the

```
heart.dt[, dm:=NULL]
heart.dt[heart.dt == -9] <- NA
summary(heart.dt)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.41   Mean   :0.6773   Mean   :3.163   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##
##       chol            cigs           years            fbs
##  Min.   :126.0   Min.   : 0.00   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:213.0   1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.:0.0000
##  Median :244.0   Median :10.00   Median :15.00   Median :0.0000
##  Mean   :249.1   Mean   :16.92   Mean   :15.26   Mean   :0.1489
```

```
##  3rd Qu.:277.0   3rd Qu.:30.00   3rd Qu.:30.00   3rd Qu.:0.0000
##  Max.   :564.0   Max.   :99.00   Max.   :54.00   Max.   :1.0000
##                  NA's   :5       NA's   :5
##     famhist          restecg         thalach          exang
##  Min.   :0.0000   Min.   :0.000   Min.   : 71.0   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:133.2   1st Qu.:0.0000
##  Median :1.0000   Median :2.000   Median :153.5   Median :0.0000
##  Mean   :0.6206   Mean   :1.014   Mean   :149.8   Mean   :0.3262
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:165.8   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :2.000   Max.   :202.0   Max.   :1.0000
##
##       thal            num
##  Min.   :3.000   Min.   :0.0000
##  1st Qu.:3.000   1st Qu.:0.0000
##  Median :3.000   Median :0.0000
##  Mean   :4.679   Mean   :0.9078
##  3rd Qu.:7.000   3rd Qu.:2.0000
##  Max.   :7.000   Max.   :4.0000
##  NA's   :2
```

```
heart.dt.nona <- knnImputation(heart.dt)
heart.dt.nona[num >= 1, ]$num <- 1
summary(heart.dt.nona)
```

```
##       age             sex              cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.41   Mean   :0.6773   Mean   :3.163   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##       chol            cigs           years            fbs
##  Min.   :126.0   Min.   : 0.00   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:213.0   1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.:0.0000
##  Median :244.0   Median :11.98   Median :15.00   Median :0.0000
##  Mean   :249.1   Mean   :16.96   Mean   :15.35   Mean   :0.1489
##  3rd Qu.:277.0   3rd Qu.:30.00   3rd Qu.:30.00   3rd Qu.:0.0000
##  Max.   :564.0   Max.   :99.00   Max.   :54.00   Max.   :1.0000
##     famhist          restecg         thalach          exang
##  Min.   :0.0000   Min.   :0.000   Min.   : 71.0   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:133.2   1st Qu.:0.0000
##  Median :1.0000   Median :2.000   Median :153.5   Median :0.0000
##  Mean   :0.6206   Mean   :1.014   Mean   :149.8   Mean   :0.3262
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:165.8   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :2.000   Max.   :202.0   Max.   :1.0000
##       thal            num
##  Min.   :3.000   Min.   :0.0000
##  1st Qu.:3.000   1st Qu.:0.0000
##  Median :3.000   Median :0.0000
##  Mean   :4.677   Mean   :0.4433
##  3rd Qu.:7.000   3rd Qu.:1.0000
##  Max.   :7.000   Max.   :1.0000
```

```
heart.dt.nona$num <- as.factor(heart.dt.nona$num)
```

Creating Training and Testing Sets

Using the caret package, we can split the data, and make it reproducible by setting the seed.

```
set.seed(42)
targets = heart.dt.nona$num
features = heart.dt.nona[,-'num', with=F]
summary(features)
```

```
##       age             sex               cp           trestbps
##  Min.   :29.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:48.00   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:120.0
##  Median :55.00   Median :1.0000   Median :3.000   Median :130.0
##  Mean   :54.41   Mean   :0.6773   Mean   :3.163   Mean   :131.6
##  3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##       chol            cigs           years            fbs
##  Min.   :126.0   Min.   : 0.00   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:213.0   1st Qu.: 0.00   1st Qu.: 0.00   1st Qu.:0.0000
##  Median :244.0   Median :11.98   Median :15.00   Median :0.0000
##  Mean   :249.1   Mean   :16.96   Mean   :15.35   Mean   :0.1489
##  3rd Qu.:277.0   3rd Qu.:30.00   3rd Qu.:30.00   3rd Qu.:0.0000
##  Max.   :564.0   Max.   :99.00   Max.   :54.00   Max.   :1.0000
##     famhist          restecg         thalach          exang
##  Min.   :0.0000   Min.   :0.000   Min.   : 71.0   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:133.2   1st Qu.:0.0000
##  Median :1.0000   Median :2.000   Median :153.5   Median :0.0000
##  Mean   :0.6206   Mean   :1.014   Mean   :149.8   Mean   :0.3262
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:165.8   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :2.000   Max.   :202.0   Max.   :1.0000
##       thal
##  Min.   :3.000
##  1st Qu.:3.000
##  Median :3.000
##  Mean   :4.677
##  3rd Qu.:7.000
##  Max.   :7.000
```

```
trainIdx = createDataPartition(targets, p = 0.7)$Resample1
training.features <- features[trainIdx]
training.targets <- targets[trainIdx]
testing.features <- features[-trainIdx, ]
testing.targets <- targets[-trainIdx]
```

Running a confusion matrix after predicting with k =3, we find that our accuracy is quite low. To try to improve on this, we will try principal component analysis as an exploratory tool to reduce the number of variables, followed by weighting if necessary or finding the correct number of nearest neighbors to minimize our total error.

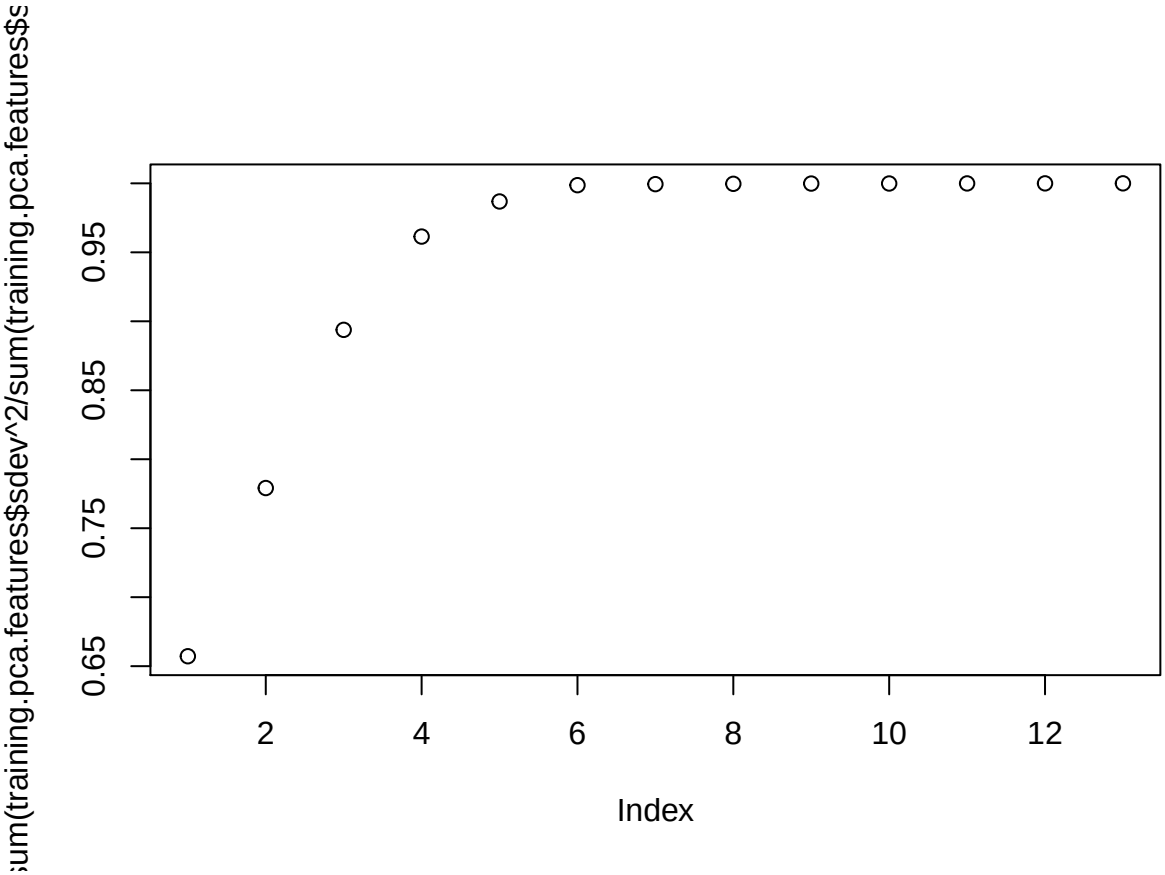
```
testing.predictions <- knn(train = training.features, test = testing.features, cl = training.targets, k

confusionMatrix(testing.predictions, testing.targets)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 29 15
##          1 18 22
##
##                Accuracy : 0.6071
##                  95% CI : (0.4945, 0.712)
##     No Information Rate : 0.5595
##     P-Value [Acc > NIR] : 0.2215
##
##                   Kappa : 0.2098
##  Mcnemar's Test P-Value : 0.7277
##
##             Sensitivity : 0.6170
##             Specificity : 0.5946
##          Pos Pred Value : 0.6591
##          Neg Pred Value : 0.5500
##              Prevalence : 0.5595
##          Detection Rate : 0.3452
##    Detection Prevalence : 0.5238
##       Balanced Accuracy : 0.6058
##
##        'Positive' Class : 0
##
```

Feature Selection

Selecting features is quite important, and running principal component analysis allows us to identify which features we could focus on. This can be dummy checked by looking at a correlation plot of the data to see which variables are associated with our target variable. Looking at the cumulative plot, we identify that 95% of our variance is caused by the first four principal components. Those particular variables are associated with chol, thalach, cigs, years, and trestbps.

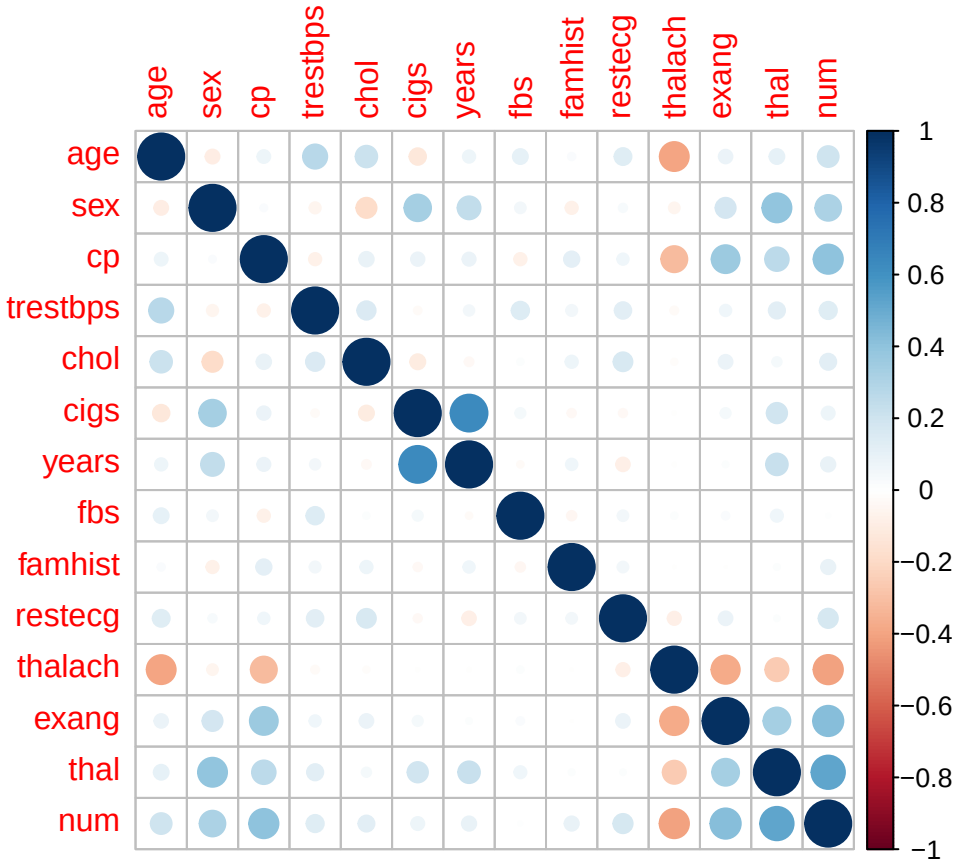
```
training.pca.features <- prcomp(training.features)
plot(cumsum(training.pca.features$sdev^2 / sum(training.pca.features$sdev^2)))
```



Correlation Plot

With the principal component analysis in mind, we look at the numeric correlation plot to identify target variables of interest. In this, one can find things susch as sex, cp, thalach, exang, and thal are the most important. One possible reasoning for the difference in our principal component analysis and these variables is the mere fact that these variables have very little variance. Still, from the above two analysis, we most likely want to get rid of restecg, famhist, fbs.

```
heart.dt.numeric <- heart.dt.nona[,lapply(.SD, as.numeric)]
corrplot(cor(heart.dt.numeric))
```




```
heart.dt.nona[, restecg:=NULL]
heart.dt.nona[, famhist:=NULL]
heart.dt.nona[, fbs:=NULL]
```

Second KNN Results

By removing these three variables, we had no change in accuracy for our test set, indicating that these three variables have no impact on the strength of the model.

```
set.seed(42)
targets = heart.dt.nona$num
features = heart.dt.nona[,-'num', with=F]
trainIdx = createDataPartition(targets, p = 0.7)$Resample1
training.features <- features[trainIdx]
training.targets <- targets[trainIdx]
testing.features <- features[-trainIdx, ]
testing.targets <- targets[-trainIdx]

testing.predictions <- knn(train = training.features, test = testing.features, cl = training.targets, k

confusionMatrix(testing.predictions, testing.targets)
```

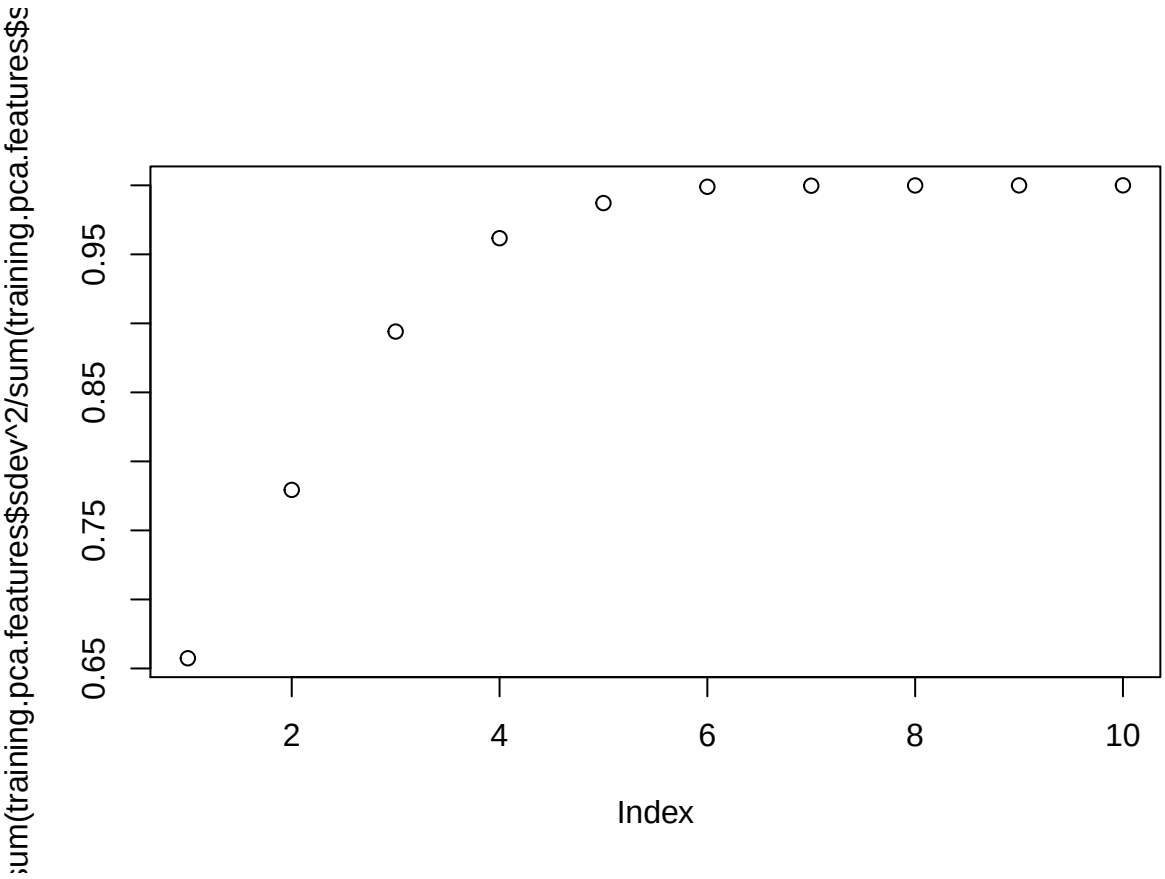
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
```

```
##          0 29 15
##          1 18 22
##
##                Accuracy : 0.6071
##                  95% CI : (0.4945, 0.712)
##     No Information Rate : 0.5595
##     P-Value [Acc > NIR] : 0.2215
##
##                   Kappa : 0.2098
##  Mcnemar's Test P-Value : 0.7277
##
##             Sensitivity : 0.6170
##             Specificity : 0.5946
##          Pos Pred Value : 0.6591
##          Neg Pred Value : 0.5500
##              Prevalence : 0.5595
##          Detection Rate : 0.3452
##    Detection Prevalence : 0.5238
##       Balanced Accuracy : 0.6058
##
##        'Positive' Class : 0
##
```

Second correlation plot

Looking at the PCA plot, we note that not much has changed since removing the three variables which contributed nothing to our model.

```
training.pca.features <- prcomp(training.features)
plot(cumsum(training.pca.features$sdev^2 / sum(training.pca.features$sdev^2)))
```




```
training.pca.features
```

```
## Standard deviations (1, .., p=10):
##  [1] 53.0451061 22.8480542 22.1576390 17.0150343 10.4371377  7.1110093
##  [7]  1.8046266  0.8963304  0.4282247  0.3923430
##
## Rotation (n x k) = (10 x 10):
##                     PC1          PC2          PC3           PC4
## age      -0.0436939220  0.181853179  0.061296020 -1.704587e-01
## sex       0.0015317541  0.002446140 -0.006371661  1.601380e-03
## cp       -0.0018357175  0.012627108 -0.001783676  9.577733e-03
## trestbps -0.0385152835  0.111797338 -0.073722309 -9.653268e-01
## chol     -0.9978264636  0.003909849 -0.019696893  5.012819e-02
## cigs      0.0288614883  0.222401992 -0.771504636  1.450527e-01
## years     0.0007963841  0.162496937 -0.563357114 -3.341985e-02
## thalach  -0.0103242428 -0.936894346 -0.278322799 -1.194761e-01
## exang    -0.0005398660  0.006460760  0.001010809 -6.441999e-08
## thal     -0.0018829368  0.023862204 -0.017911675 -7.009468e-03
##                     PC5           PC6          PC7           PC8
## age      -0.3270816595 -0.9082946457  0.010784458 -0.0103328147
## sex       0.0010031941 -0.0001966138  0.086027144  0.0814821911
## cp       -0.0009411144  0.0130352227  0.083441200 -0.9842245919
## trestbps  0.1711099668  0.1387351542 -0.012151081 -0.0066884811
## chol      0.0229714534  0.0297137019 -0.002557946  0.0027163013
## cigs      0.5305883918 -0.2273861067 -0.013738426  0.0005411134
## years    -0.7610940736  0.2745201777 -0.021158625  0.0048815962
```

```
## thalach  -0.0490783030 -0.1653625258  0.020728846 -0.0135971397
## exang     0.0026434562  0.0043141771  0.067961110 -0.1304669653
## thal     -0.0023429397  0.0164738339  0.989788650  0.0852824138
##                   PC9         PC10
## age       0.001309683 -0.000726751
## sex       0.672239176  0.730754361
## cp       -0.050377393  0.146195438
## trestbps  0.000250818  0.002927073
## chol      0.000725725  0.001103987
## cigs     -0.003553987 -0.003812991
## years     0.000653062 -0.002919961
## thalach   0.003013724 -0.002680794
## exang     0.731172707 -0.666091193
## thal     -0.104492282 -0.030115020
```

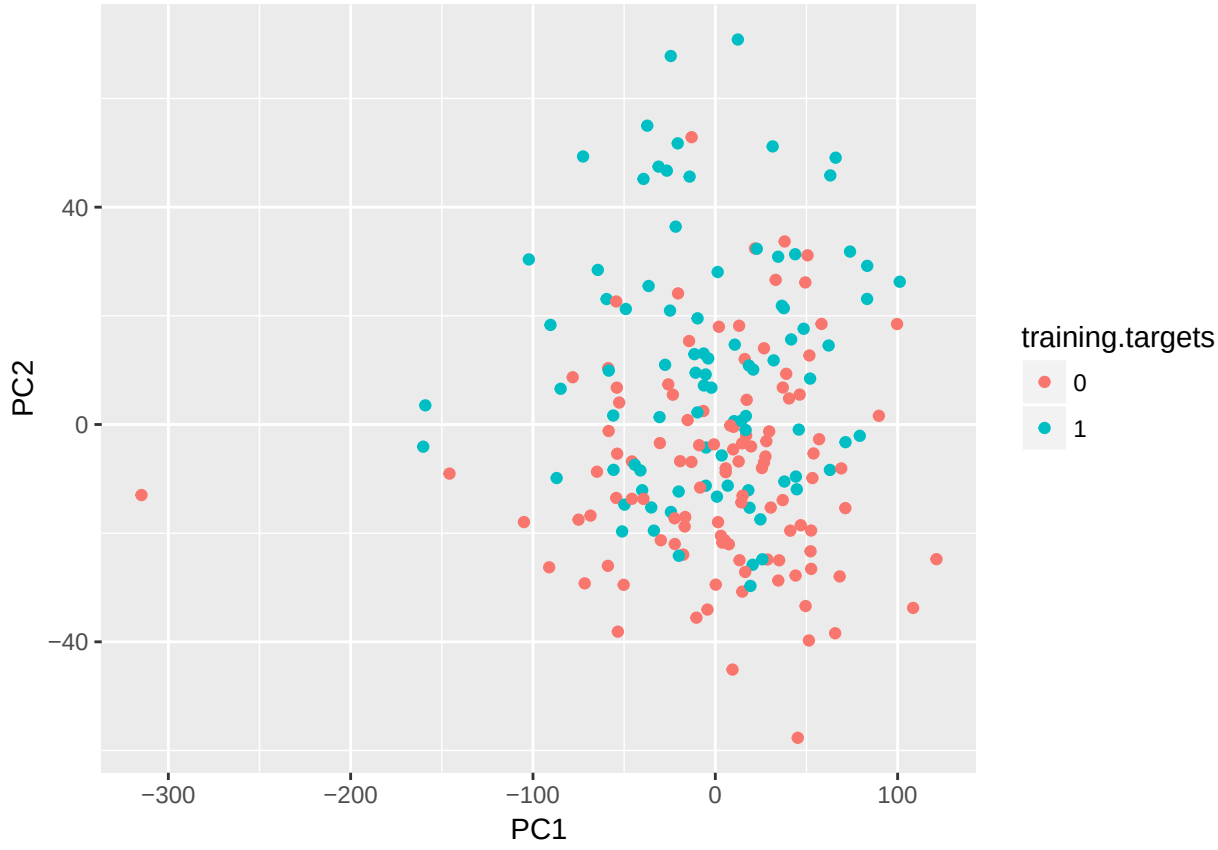
PCA component plotting

Plotting the PCA component analysis with this reduced variable set, we see a hard to differentiate 2 dimensional plot, especially for individuals not suffering from heart disease. Clearly, heart disease is present in the normal population, and gives an indication why these variables are a poor result in guessing whether or not an individual suffers heart disease in our confusion matrix.

```
training.pca.dt <- as.data.table(training.pca.features$x)
summary(training.pca.dt)
```

```
##       PC1                PC2               PC3              PC4
##  Min.   :-314.719   Min.   :-57.663   Min.   :-49.58   Min.   :-52.988
##  1st Qu.: -27.292   1st Qu.:-15.925   1st Qu.:-16.73   1st Qu.:-11.002
##  Median :   5.716   Median : -3.564   Median : -2.04   Median :  2.147
##  Mean   :   0.000   Mean   :  0.000   Mean   :  0.00   Mean   :  0.000
##  3rd Qu.:  36.136   3rd Qu.: 13.037   3rd Qu.: 18.70   3rd Qu.: 11.182
##  Max.   : 121.286   Max.   : 70.854   Max.   : 41.25   Max.   : 31.488
##       PC5                PC6               PC7               PC8
##  Min.   :-33.286   Min.   :-20.1714   Min.   :-3.4841   Min.   :-1.5602
##  1st Qu.: -4.946   1st Qu.: -5.1716   1st Qu.:-1.4626   1st Qu.:-0.6314
##  Median :  1.541   Median :  0.2396   Median :-0.8153   Median :-0.1698
##  Mean   :  0.000   Mean   :  0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.:  5.320   3rd Qu.:  4.8957   3rd Qu.: 1.8353   3rd Qu.: 0.5837
##  Max.   : 42.224   Max.   : 17.6472   Max.   : 3.5391   Max.   : 2.2485
##       PC9                PC10
##  Min.   :-1.03408   Min.   :-1.079678
##  1st Qu.:-0.37497   1st Qu.:-0.252433
##  Median : 0.06014   Median :-0.003466
##  Mean   : 0.00000   Mean   : 0.000000
##  3rd Qu.: 0.29700   3rd Qu.: 0.313741
##  Max.   : 0.98051   Max.   : 0.884525
```

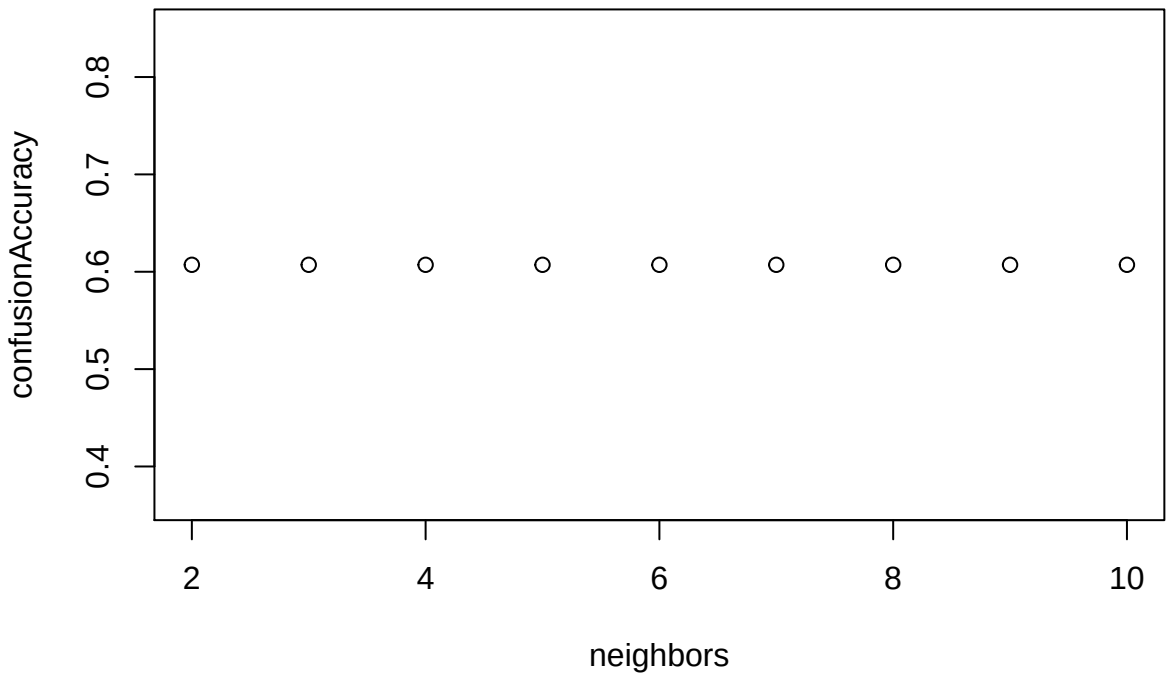
```
sp <- ggplot(training.pca.dt, aes(x=PC1, y=PC2, color=training.targets)) + geom_point()
print(sp)
```



Model Hyperparameter Tuning

in this section we will tune our plot to choose the proper level of k, the number of nearest neighbors. To attempt to do this, we will try different numbers of k and then plot the resulting confusion matrix overall accuracy vs k. As one can see, the result is a model which predicts to a 61% prediction, regardless of the number of k, indicating our model is sufficiently poor for all k's. When looking at the PCA, one can see that the data is highly mixed based on the features we look at.

```
# loop through 2-10 nearest neighbors and check cross-validation score
neighbors <- seq(2, 10)
confusionAccuracy <- c()  # accuracy of confusion matrix

for (k in neighbors) {
  predictions <- knn(train = training.features, test = testing.features, cl = training.targets, k = k)
  confusionAccuracy <- c(confusionAccuracy, confusionMatrix(testing.predictions, testing.targets)$overal
}

plot(neighbors, confusionAccuracy)
```


confusionAccuracy
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
neighbors