Health of City X

APSherer

10/21/2022

Introduction

Healthcare is one of the most important fields in the world, and being able to determine the risk each individual has of developing a major health concern is something that is incredibly important. The goal of this project is to pick and few health concerns and try to determine the underlying health factors that are the most beneficial in predicting if a certain individual is at risk for a particular health concern.

Prerequisites

There are a few things that we need to do before we are able to do any kind of analysis. The first and most important of those things is to load the data set into R. In the code below I am loading the data set in and assigning it to a variable named "health."

```
health  <- read.csv("Health of CityX.csv")
```

The other important thing to do before going any further is determining which (if any) libraries are going to be used during the project. The only one I am going to use is a library called "tidyverse" which is actually a collection of a few different libraries. Again, this can be done by running the code as below.

```
library(tidyverse)
```

After doing both of those things, we have one final step before we are actually able to analyze the data.

Cleaning the Data

Before cleaning the data it may be beneficial to look at the structure and see what kind of data we are actually dealing with, and determine if there are any specific steps that should be taken to make sure our data is easier to work with. The code and output for this are shown below.

```
str(health)
```

```
## 'data.frame':    198 obs. of  28 variables:
##  $ Population2010        : int  3611 2552 1546 3009 3394 2687 4939 3045 3223 2697 ...
##  $ ACCESS2_CrudePrev     : num  10 8.2 15.5 12.3 13.4 11.1 13.6 14.4 13.8 16.5 ...
##  $ ARTHRITIS_CrudePrev   : num  19 23.4 8.3 13.5 12 18.5 10.9 9.6 10.6 11.7 ...
##  $ BINGE_CrudePrev       : num  24 20.8 27 27.1 28.9 24.4 28.7 29.6 29.7 26.6 ...
##  $ BPHIGH_CrudePrev      : num  23.8 28.7 13.4 18.3 17.2 23.9 15.7 14.8 15.7 18 ...
##  $ BPMED_CrudePrev       : num  73.9 77.9 50.7 63.6 60.7 72.3 58.7 55 56.3 59.9 ...
##  $ CANCER_CrudePrev      : num  7.5 8.8 2.2 4.1 3.8 6.4 3.3 2.9 3.2 3.3 ...
```

```
##  $ CASTHMA_CrudePrev     : num  7.3 7.3 8.8 7.9 7.5 7.6 7.7 7.6 7.9 7.7 ...
##  $ CHD_CrudePrev         : num  4.5 5.4 1.9 2.9 2.5 4.1 2.4 2.1 2.3 2.4 ...
##  $ CHECKUP_CrudePrev     : num  66.1 70.1 57.5 60.5 60 65.1 58.8 57.9 57.8 59.8 ...
##  $ CHOLSCREEN_CrudePrev  : num  83.7 88.8 65.9 76.3 77.8 83.3 72.9 72.2 72.5 76.6 ...
##  $ COLON_SCREEN_CrudePrev: num  72 74.6 64.1 66.8 66.8 70.3 67.2 65.3 64.4 65.6 ...
##  $ COPD_CrudePrev        : num  3.4 4.1 2.6 3.1 2.6 3.6 2.7 2.4 2.7 2.7 ...
##  $ COREM_CrudePrev       : num  42.4 43.8 35.5 38.3 39.1 40.5 37.2 37.4 36.5 35 ...
##  $ COREW_CrudePrev       : num  30.7 34.9 27.7 30.1 30.7 30.6 29.5 27.4 28.7 28.5 ...
##  $ CSMOKING_CrudePrev    : num  8.6 8.2 11.8 11.3 10.9 10.3 11.3 10.9 12.2 12.3 ...
##  $ DENTAL_CrudePrev      : num  79.2 81.9 69.3 74.7 74 76.8 72.9 72.5 71.1 70 ...
##  $ DIABETES_CrudePrev    : num  6.2 8 3.6 5.2 4.8 6.9 4.3 4.1 4.2 5.4 ...
##  $ HIGHCHOL_CrudePrev    : num  30.2 36.2 17.8 25.1 23.7 30.5 21.9 20.4 21.6 23.5 ...
##  $ KIDNEY_CrudePrev      : num  2.3 2.5 1.4 1.8 1.6 2.2 1.5 1.4 1.5 1.7 ...
##  $ LPA_CrudePrev         : num  15.8 15.9 17.5 16.8 15.9 17 16.3 16 16.6 18.5 ...
##  $ MAMMOUSE_CrudePrev    : num  80.5 80.4 80.8 80.4 80.7 80.2 80.4 81.2 80.2 81.7 ...
##  $ MHLTH_CrudePrev       : num  8.3 7.8 13.7 11 9.9 9.3 11 10.9 11.6 10.9 ...
##  $ OBESITY_CrudePrev     : num  19.9 21.4 20.2 21.8 21.3 22.4 20.2 20.6 21.5 24 ...
##  $ PHLTH_CrudePrev       : num  7.1 7.9 6.4 7.1 6.4 7.6 6.3 6 6.6 7.1 ...
##  $ SLEEP_CrudePrev       : num  24.4 23.6 26.7 26.4 27 26 26.7 26.3 27.3 28.8 ...
##  $ STROKE_CrudePrev      : num  2.1 2.4 1 1.5 1.3 2 1.2 1.1 1.2 1.4 ...
##  $ TEETHLOST_CrudePrev   : num  5.7 4.4 8.9 6.6 6.3 6 6.6 6.5 7.6 7.6 ...
```

This output may be overwhelming to look at at first, but all we are really looking for is to see what type of data each variable is. In general it is much easier to work with numeric data, and in this case most of the data is numeric, so we shouldn't have too many issues with that.

The first step in actually cleaning the data is to make sure we do not have any missing values, or NAs. The code below can help us answer this, as it should give the output of the number of NAs we have in our data set.

```
sum(is.na(health))
```

```
## [1] 2
```

This tells us that we have two NAs in our data set that we have to deal with in order to be able to analyze our data. There are a few different ways you can do this, but the simplest way, and the way I am going to use, is by just omitting the rows that have NAs in them. This isn't really the best practice, but two NAs is not very many, so it would seem excessive to have to come up with a better solution in this scenario. We can omit the NAs by following this block of code:

```
health <- na.omit(health)
```

Now if we check our data by using the same code as before, we should see that there we now have zero NAs in our data.

```
sum(is.na(health))
```

```
## [1] 0
```

When cleaning the data we are trying to make it as organized and usable as possible. One thing that I noticed when I looked at the structure of our health data is that nearly every variable ends with "_CrudePrev." While this suffix doesn't cause any issues with our analysis, it is not very convenient to have to type it out

every time we type out the name of any of our variables. In order to work around this, I am going to remove the suffix from the end of every variable. Because it is not exclusive to a few variables, we can safely assume that removing this will not affect our data, or our ability to interpret the data. I removed the suffix by using the code below.

```
names(health) <- sub('_CrudePrev', '', names(health))
```

And now if we look at the structure of our data again, we should be able to see that the suffix has been removed form every variable.

```
str(health)
```

```
## 'data.frame':    196 obs. of  28 variables:
##  $ Population2010: int  3611 2552 1546 3009 3394 2687 4939 3045 3223 2697 ...
##  $ ACCESS2       : num  10 8.2 15.5 12.3 13.4 11.1 13.6 14.4 13.8 16.5 ...
##  $ ARTHRITIS     : num  19 23.4 8.3 13.5 12 18.5 10.9 9.6 10.6 11.7 ...
##  $ BINGE         : num  24 20.8 27 27.1 28.9 24.4 28.7 29.6 29.7 26.6 ...
##  $ BPHIGH        : num  23.8 28.7 13.4 18.3 17.2 23.9 15.7 14.8 15.7 18 ...
##  $ BPMED         : num  73.9 77.9 50.7 63.6 60.7 72.3 58.7 55 56.3 59.9 ...
##  $ CANCER        : num  7.5 8.8 2.2 4.1 3.8 6.4 3.3 2.9 3.2 3.3 ...
##  $ CASTHMA       : num  7.3 7.3 8.8 7.9 7.5 7.6 7.7 7.6 7.9 7.7 ...
##  $ CHD           : num  4.5 5.4 1.9 2.9 2.5 4.1 2.4 2.1 2.3 2.4 ...
##  $ CHECKUP       : num  66.1 70.1 57.5 60.5 60 65.1 58.8 57.9 57.8 59.8 ...
##  $ CHOLSCREEN    : num  83.7 88.8 65.9 76.3 77.8 83.3 72.9 72.2 72.5 76.6 ...
##  $ COLON_SCREEN  : num  72 74.6 64.1 66.8 66.8 70.3 67.2 65.3 64.4 65.6 ...
##  $ COPD          : num  3.4 4.1 2.6 3.1 2.6 3.6 2.7 2.4 2.7 2.7 ...
##  $ COREM         : num  42.4 43.8 35.5 38.3 39.1 40.5 37.2 37.4 36.5 35 ...
##  $ COREW         : num  30.7 34.9 27.7 30.1 30.7 30.6 29.5 27.4 28.7 28.5 ...
##  $ CSMOKING      : num  8.6 8.2 11.8 11.3 10.9 10.3 11.3 10.9 12.2 12.3 ...
##  $ DENTAL        : num  79.2 81.9 69.3 74.7 74 76.8 72.9 72.5 71.1 70 ...
##  $ DIABETES      : num  6.2 8 3.6 5.2 4.8 6.9 4.3 4.1 4.2 5.4 ...
##  $ HIGHCHOL      : num  30.2 36.2 17.8 25.1 23.7 30.5 21.9 20.4 21.6 23.5 ...
##  $ KIDNEY        : num  2.3 2.5 1.4 1.8 1.6 2.2 1.5 1.4 1.5 1.7 ...
##  $ LPA           : num  15.8 15.9 17.5 16.8 15.9 17 16.3 16 16.6 18.5 ...
##  $ MAMMOUSE      : num  80.5 80.4 80.8 80.4 80.7 80.2 80.4 81.2 80.2 81.7 ...
##  $ MHLTH         : num  8.3 7.8 13.7 11 9.9 9.3 11 10.9 11.6 10.9 ...
##  $ OBESITY       : num  19.9 21.4 20.2 21.8 21.3 22.4 20.2 20.6 21.5 24 ...
##  $ PHLTH         : num  7.1 7.9 6.4 7.1 6.4 7.6 6.3 6 6.6 7.1 ...
##  $ SLEEP         : num  24.4 23.6 26.7 26.4 27 26 26.7 26.3 27.3 28.8 ...
##  $ STROKE        : num  2.1 2.4 1 1.5 1.3 2 1.2 1.1 1.2 1.4 ...
##  $ TEETHLOST     : num  5.7 4.4 8.9 6.6 6.3 6 6.6 6.5 7.6 7.6 ...
##  - attr(*, "na.action")= 'omit' Named int [1:2] 15 163
##   ..- attr(*, "names")= chr [1:2] "15" "163"
```

This should make our code significantly easier to read and use.

The final thing we need to do as a part of the cleaning process is create train and test sets for our data. This is a very important step because it allows us to create two different samples of our data so that we can use one sample to build models, and use the other sample to test their accuracy. For this project we are going to use 75% of the data for the train set, and 25% of the data for the test set. There is no right or wrong answer as to what kind of split you want to use here, but a split that is similar to this one is the most common. We will also set a seed, which allows anyone to be able to replicate the exact same train and test sets we are creating here. The following code is used to split the data.

```r
set.seed(2342)
trainHealth <- sample(nrow(health), 0.75*nrow(health), replace=FALSE)

healthTrain <- health[trainHealth,]
healthTest <- health[-trainHealth,]
```

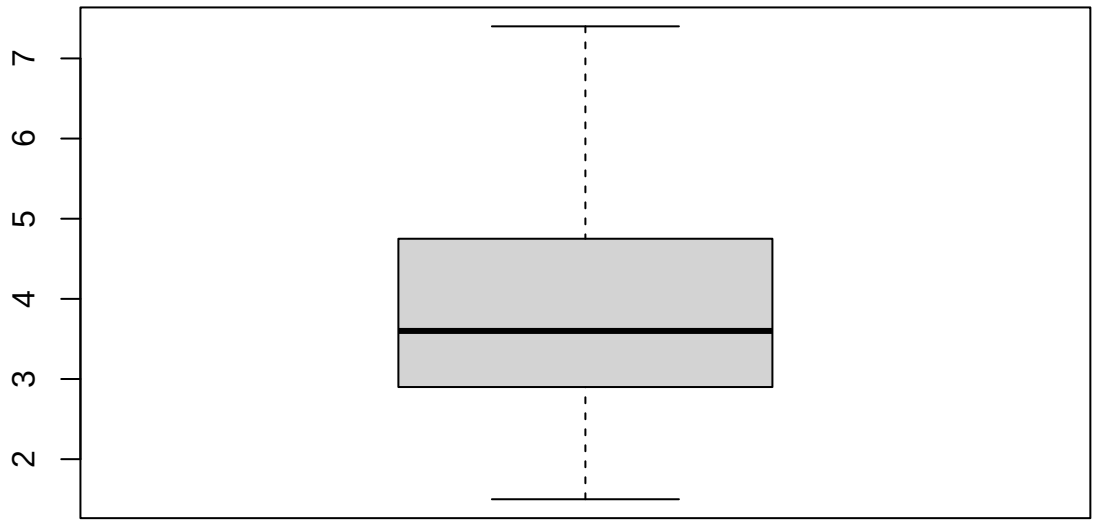
After running this code, we should have complete train and test sets for our data, and we can finally start with our analysis.

Analyzing the Data

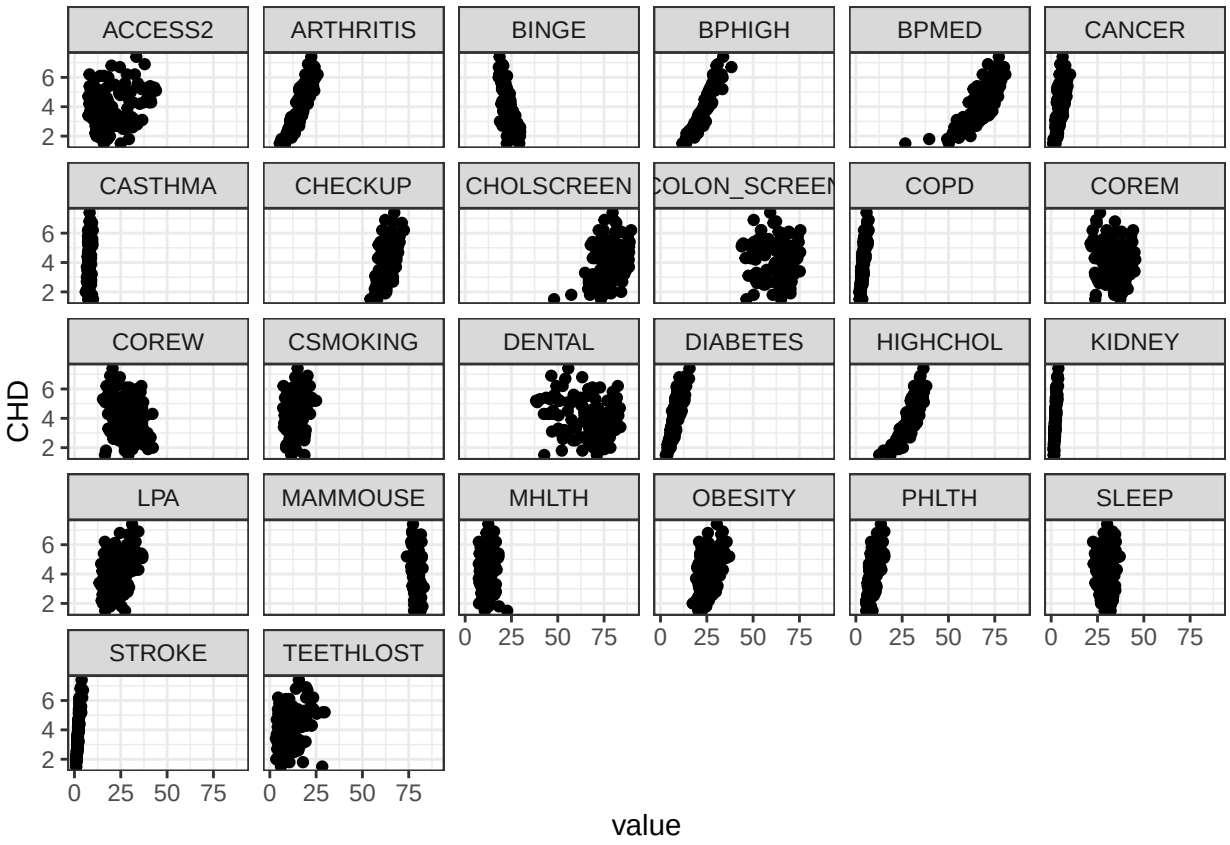
The first step in analyzing the data is to determine what questions we want to answer. For this project I am primarily looking at the question "How can we determine if a person is at risk for a major health concern?" This question is very vague, but we can refine it a little bit better by decided what we are considering "major health concerns." While there are many others that are important, I am primarily going to look at coronary heart disease or CHD. By doing this we can reword our original question to be more specific in what we are looking for. "What health traits provide the most information about CHD, and how can we use them to predict an individual's risk of CHD?"

CHD

The first thing I want to look do is look at a few different graphs to see if any of them tell us anything about the data. The first two graphs that will be shown below are a histogram which will show us the distribution of the data, and a box plot which will help us identify any outliers in our data set.



This does not give us too much information, other than that our data is at least close to normally distributed, and we do not appear to have any extreme outliers. There is one more set of graphs that I want to look at. I want to graph CHD against all other variables in the data set to see if we can determine a relationship that may not be linear, but that we can transform to make linear.



While these graphs are not the easiest to look at as there is a lot going on, they are good enough to see that there doesn't really appear to be a non-linear relationship between CHD and any of the variables.

The next thing I am going to do when trying to build a model to predict CHD is run a correlation test. This will tell us which variables in our data set have the highest correlation with CHD, and therefore may be useful when building a model. The code below will run the test, and the output will be listed below.

```
chd.cor <- cor(healthTrain[ , colnames(healthTrain) != "CHD"],
               healthTrain$CHD)
chd.cor
```

```
##                      [,1]
## Population2010 -0.1389959
## ACCESS2         0.2965629
## ARTHRITIS       0.8673891
## BINGE          -0.7763016
## BPHIGH          0.9457259
## BPMED           0.7585378
## CANCER          0.5968265
## CASTHMA         0.3366262
## CHECKUP         0.5834210
## CHOLSCREEN      0.2515294
## COLON_SCREEN   -0.1641321
## COPD            0.8869832
## COREM          -0.2285824
## COREW          -0.2874995
## CSMOKING        0.2973614
```

```
## DENTAL         -0.2830137
## DIABETES        0.8641683
## HIGHCHOL        0.8606513
## KIDNEY          0.9323009
## LPA             0.4741171
## MAMMOUSE       -0.3606510
## MHLTH           0.1598728
## OBESITY         0.5100001
## PHLTH           0.6940182
## SLEEP           0.1319581
## STROKE          0.9355337
## TEETHLOST       0.3917744
```

By looking at this output we can determine that there are four variables that have a correlation of above 90%. These variables are: KIDNEY: 94.8%, COPD: 90.0%, KIDNEY: 93.6%, STROKE: 94.4%. The first model I am going to look at is going to be a model that uses all four of these variables to predict CHD. This most likely will be accurate, but not very useful as it requires a lot of data to use, which is something we do not always have, but it will give us a good baseline to compare future models to. The code that generates the model will be listed below, as well as the summary of the model that will give us some useful information about the model and its accuracy.

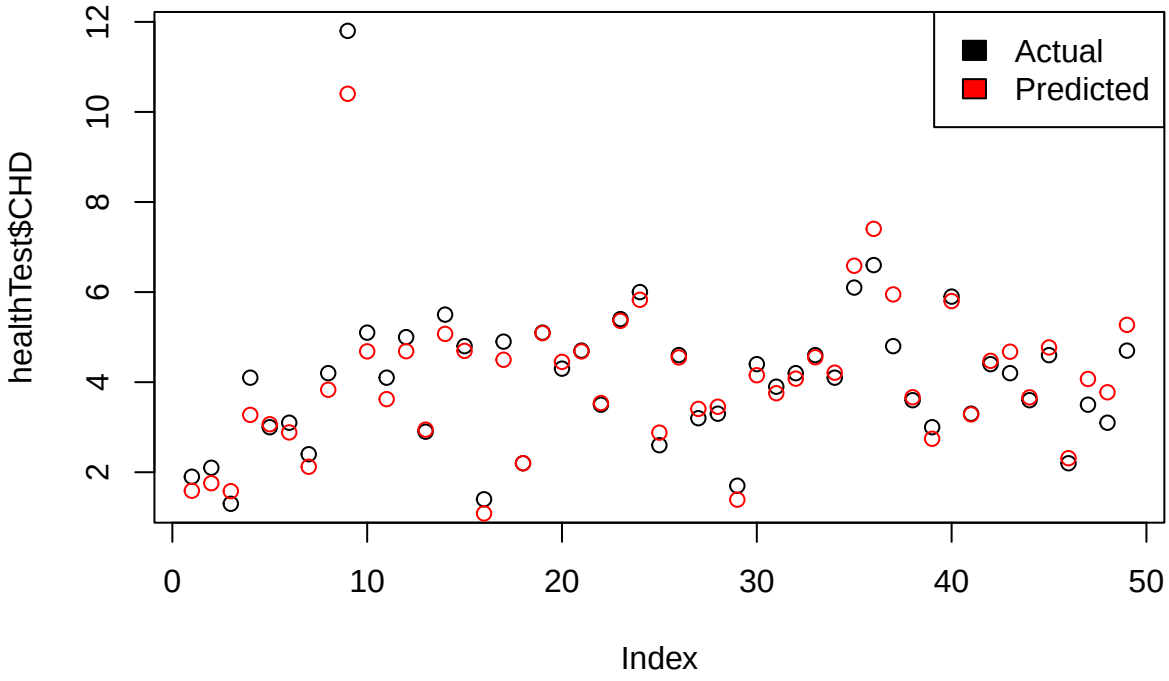
```
chd.model <- lm(CHD ~ BPHIGH + COPD + KIDNEY + STROKE, data=healthTrain)
summary(chd.model)
```

```
##
## Call:
## lm(formula = CHD ~ BPHIGH + COPD + KIDNEY + STROKE, data = healthTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9471 -0.2268  0.0328  0.1884  0.9402
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.92601    0.25457  -7.566 4.42e-12 ***
## BPHIGH       0.16000    0.01603   9.979  < 2e-16 ***
## COPD         0.17809    0.07357   2.421  0.01675 *
## KIDNEY       0.79535    0.26806   2.967  0.00353 **
## STROKE      -0.20532    0.22691  -0.905  0.36708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3368 on 142 degrees of freedom
## Multiple R-squared:  0.9307, Adjusted R-squared:  0.9288
## F-statistic: 477.1 on 4 and 142 DF,  p-value: < 2.2e-16
```

This model gives us an Adjusted R-squared of 0.9288, which is relatively high. In order to further test this model I am going to test it against our train set by predicting the data, determining the Mean Squared Error (MSE), which gives us average distance between the actual and predicted values squared, and looking at a few different plots to help aid in determining the accuracy of this model.

```
chd.predict <- predict(chd.model, healthTest)
mean((healthTest$CHD - chd.predict)^2)
```

```
## [1] 0.1677882
```



Our MSE for this model is 0.1677882, which is a very good MSE to get. The closer to 0 the better, but I am generally looking for anything under approximately 0.25. We can also see by looking at our plot that the predicted values seem to be relatively close to the actual values.

While this model does seem to be very accurate, as I said, it requires a lot of data. The next thing I want to do is see if I can reduce the number of variables, while still maintaining an accurate model.

In doing this, I am just going to create a one variable linear model for each of the four variables we have used, and evaluate each model using the same diagnostics as we did for the first model. You can see the code and explanations for the outputs for each model below.

Blood Pressure

```
chd.model.bphigh <- lm(CHD ~ BPHIGH, data=healthTrain)
summary(chd.model.bphigh)
```

```
##
## Call:
## lm(formula = CHD ~ BPHIGH, data = healthTrain)
```

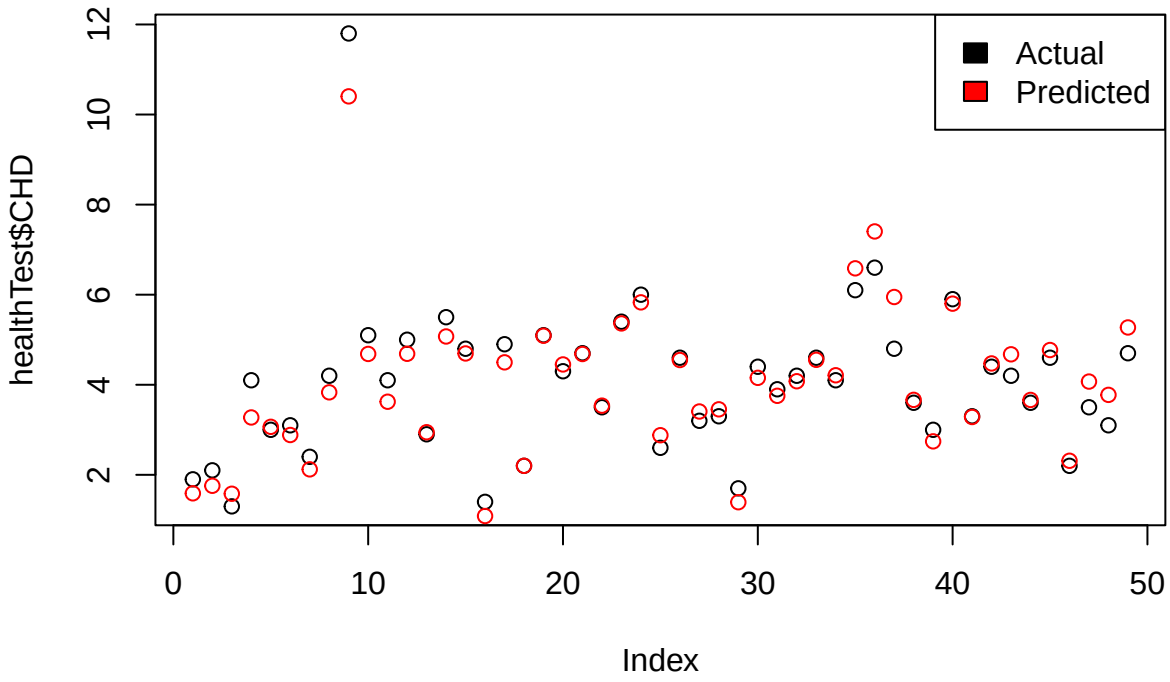
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.23865 -0.15846  0.01321  0.22973  1.16845
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.985489   0.170316  -11.66   <2e-16 ***
## BPHIGH       0.252220   0.007197   35.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4116 on 145 degrees of freedom
## Multiple R-squared:  0.8944, Adjusted R-squared:  0.8937
## F-statistic:  1228 on 1 and 145 DF,  p-value: < 2.2e-16
```

This model gives us an Adjusted R-squared value of 0.8937, which is less accurate than the original model, but still nearly 90%, which is relatively high.

We will again predict the points and use MSE and plots to evaluate them.

```
chd.predict.bphigh <- predict(chd.model.bphigh, healthTest)
mean((healthTest$CHD - chd.predict.bphigh)^2)
```

```
## [1] 0.273156
```



This model gives us a MSE of 0.2732, which is nearly double that of the last model, but still nearly under the target MSE I set of 0.25.

Kidney

```
chd.model.kidney <- lm(CHD ~ KIDNEY, data=healthTrain)
summary(chd.model.kidney)
```

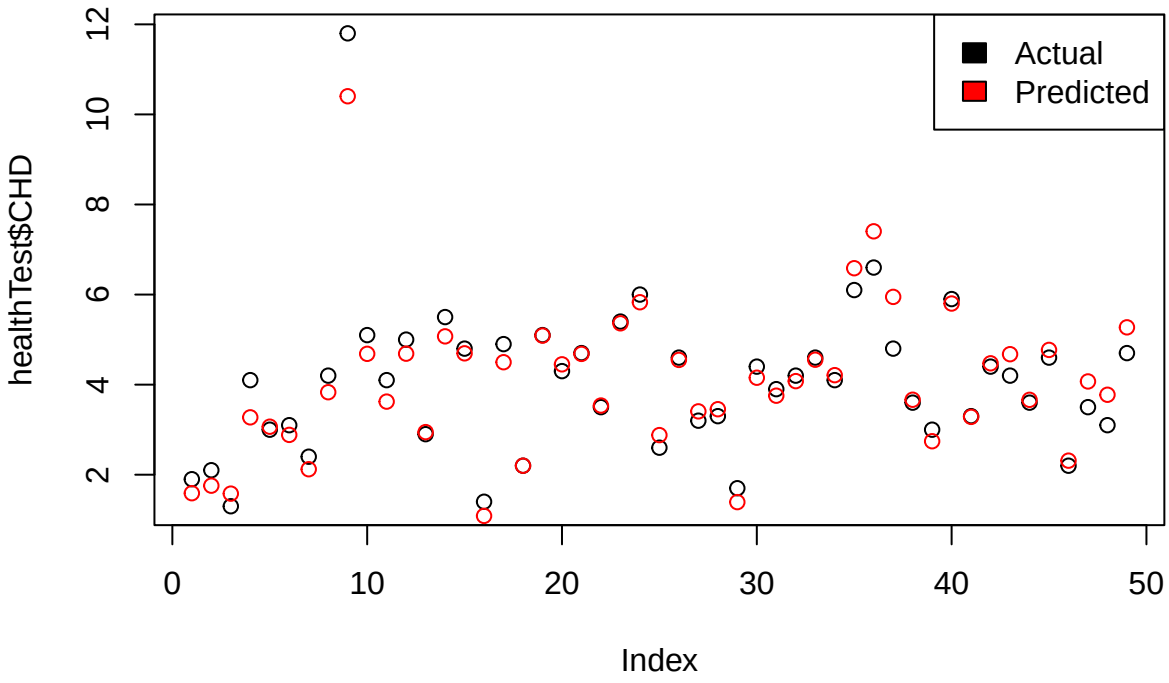
```
##
## Call:
## lm(formula = CHD ~ KIDNEY, data = healthTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1934 -0.2906 -0.0197  0.2871  1.5192
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.87165    0.15716  -5.546 1.34e-07 ***
## KIDNEY       2.09712    0.06756  31.039  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4581 on 145 degrees of freedom
## Multiple R-squared:  0.8692, Adjusted R-squared:  0.8683
## F-statistic: 963.4 on 1 and 145 DF,  p-value: < 2.2e-16
```

This model gives us an Adjusted R-squared value of 0.8692, which is not as good as our model using blood pressure.

Our plots and MSE are below.

```
chd.predict.kidney <- predict(chd.model.kidney, healthTest)
mean((healthTest$CHD - chd.predict.kidney)^2)
```

```
## [1] 0.3290362
```



This model gives us a MSE of 0.3290, which also is not as good as the blood pressure model, so we are going to assume this model is not as good. The plots also prove this as they are visibily not as accurate.

Stroke

```
chd.model.stroke <- lm(CHD ~ STROKE, data=healthTrain)
summary(chd.model.stroke)
```

```
##
## Call:
## lm(formula = CHD ~ STROKE, data = healthTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43122 -0.26156 -0.01734  0.25704  1.25010
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4291     0.1138    3.77 0.000237 ***
## STROKE        1.6744     0.0525   31.89  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4474 on 145 degrees of freedom
```

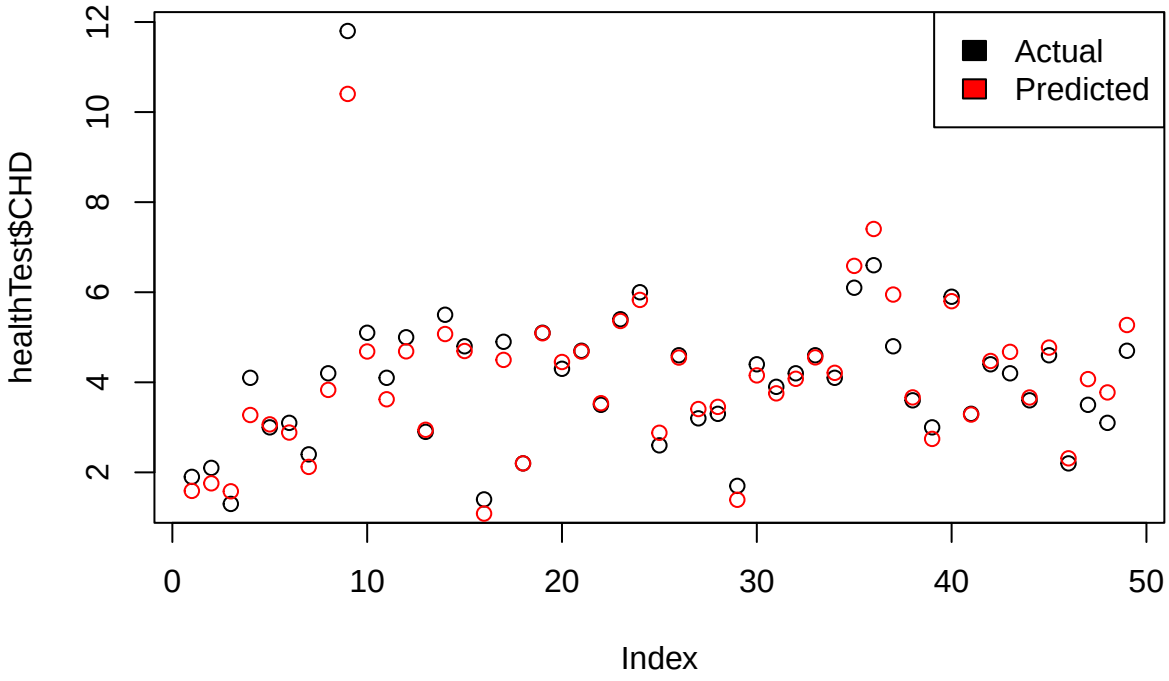
```
## Multiple R-squared:  0.8752, Adjusted R-squared:  0.8744
## F-statistic:  1017 on 1 and 145 DF,  p-value: < 2.2e-16
```

This model gives us an Adjusted R-squared value of 0.8752, which is better than the model using kidney, but not as good as the one using blood pressure.

Our plots and MSE are below.

```
chd.predict.stroke <- predict(chd.model.stroke, healthTest)
mean((healthTest$CHD - chd.predict.stroke)^2)
```

```
## [1] 0.2380226
```



This model gives us a MSE of 0.238, which is the best MSE we have gotten using our single variable linear models. This is somewhat surprising as our Adjusted R-squared did not seem to be great.

COPD

```
chd.model.copd <- lm(CHD ~ COPD, data=healthTrain)
summary(chd.model.copd)
```

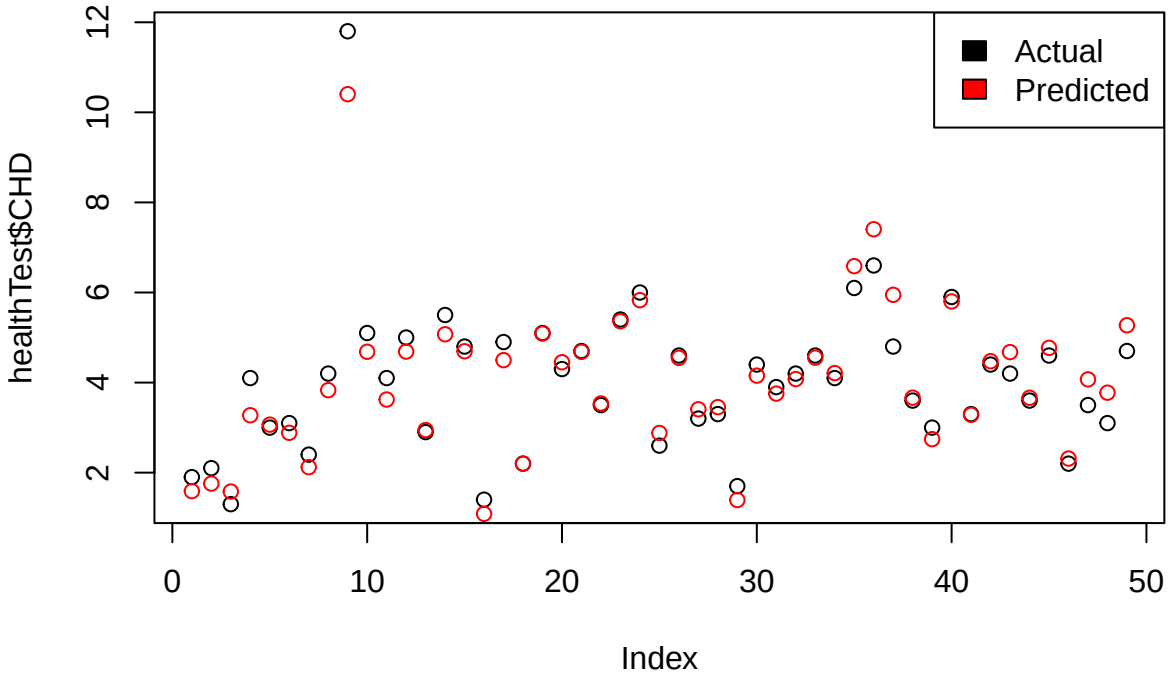
```
##
## Call:
```

```
## lm(formula = CHD ~ COPD, data = healthTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.91056 -0.39606 -0.01056  0.28742  1.89551
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06584    0.17660  -0.373     0.71
## COPD         0.99325    0.04295  23.128   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5849 on 145 degrees of freedom
## Multiple R-squared:  0.7867, Adjusted R-squared:  0.7853
## F-statistic: 534.9 on 1 and 145 DF,  p-value: < 2.2e-16
```

This model gives us an Adjusted R-squared value of 0.7862, which is the worst value we have gotten so far. Our plots and MSE are below.

```
chd.predict.copd <- predict(chd.model.copd, healthTest)
mean((healthTest$CHD - chd.predict.copd)^2)
```

```
## [1] 0.4852078
```



This model gives us a MSE of 0.48522, which again, is the worst value we have had so far.

Summary

So what do all of these R-squared and MSE values mean, and how are they useful to us? Well as I have said, they help us evaluate the models and how accurate they are. Our ultimate goal here is that we are trying to find the best possible prediction for coronary heart disease, or CHD. However the best model can mean a multitude of different things. If we only look at best as the most accurate, then we will run into some problems. For example, if we look at the models we built for CHD and evaluate them using this mindset, we can confidently say that the best model was the one that included all of our most correlated variables, blood pressure, kidney, stroke, and COPD. However, this is not really the best model, because some of these variables are very hard to obtain information on, and the chances of having all of this data are very slim.

So the real question we are trying to answer here is "Which of our models should be used to try to predict CHD?" The two models that appeared to be the most accurate based off of MSE and Adjusted R-squared is the model that used stroke as the predictor variable, and the model that used blood pressure as the predictor variable. Despite having a worse adjusted R-squared value, the model using stroke was more accurate when applied to the test set, so many people would pick this model to use. But one thing that many people may neglect to consider is the availability and convenience of obtaining data. If we are depending on this information to predict whether or not somebody is at risk of having coronary heart disease, what is going to be the easiest to collect: information about strokes, or the person's blood pressure. The answer would almost definitely be the blood pressure. Therefore, this is the model I would choose to provide a healthcare company with, because it is an accurate model, that also uses very obtainable data to predict a very serious health condition.