# Section 1: Getting Started

## Introduction

World Health Organization has estimated 12 million deaths occur worldwide, every year due to Heart diseases. Half the deaths in the United States and other developed countries are due to cardio vascular diseases. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high risk patients and in turn reduce the complications. This analysis intends to pinpoint the most relevant/risk factors of heart disease as well as predict the overall risk using logistic regression and random forest algorithms.

## Research question

We would like to do Exploratory Data Analysis on the data to get an understanding of the different features and how they are related to one another. We would also look at the variables and see if they are related to one another.As part of this exercise we are trying to build a classification problem to answer our main question about the chances of having a heart disease based on the various medical prognosis and tests. The variables are self-explanatory and refer to the attributes of medical test of any normal person going for checkup and the response variable is ‘presence of heart disease’.

1. What are the chances of having a heart disease based on the Vital data collected from a sample of patients of different age group and occupation?
2. What is the most significant predictor for a Heart disease?
3. Is their a significant correlation between the feature variables that could predict chances of having a heart disease for a particular individual?
4. Is there a possibility that based on the analysis of the vital data if we could prevent a possible heart failure from happening?
5. How far can it be predicted that a heart disease is possible based on the cholesterol, blood pressure and age?

## Approach

The approach I took to analyze the various data sets consists of looking at each of the different datasets obtained from the different geographic locations. The dataset was curtailed into a smaller set using 14 out of the total 76 attributes for this analysis. Then after cleaning all the datasets and looking for missing values it was finally decided to go with the most accurate dataset which was from Cleveland.  
I am planning to use Logistic Regression algorithm using glm and random forest in order to do a comparison to the output. The random forest algorithm works by aggregating the predictions made by multiple decision trees. I would be using bootstrapped dataset created from the original dataset.

## How my approach addresses (fully or partially) the problem

When the random forest is used for classification and is presented with a new sample, the final prediction is made by taking the majority of the predictions made by each individual decision tree in the forest. In the event, it is used for regression and it is presented with a new sample, the final prediction is made by taking the average of the predictions made by each individual decision tree in the forest. The final approach to organize the dataset is to get answers to the following questions.

1. What is the distribution of the data for the important features?
2. How does the features have there values normalized or do they have biases?
3. Is there any collinearity between the variables/features?
4. What are most important features from the list of all the features available in the dataset?

## Describing the Data and the methodology used for the research.

I have looked into various sources for this dataset that could be helpful in predicting the possible heart failure for an individual based on patient data that are collected for each of the patients on a regular checkup. Finally the below dataset was found to be most suitable. It was available in kaggle also.

Link to the dataset: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>

This dataset integrates all the databases present in Heart Disease Dataset available at UCI Machine Learning Repository. Original one contains 4 databases: Cleveland, Hungarian, Long Beach, and Switzerland. Most of the work has been done using Cleveland dataset only. The authors of the databases have requested:

...that any publications resulting from the use of the data include the   
 names of the principal investigator responsible for the data collection  
 at each institution. They would be:  
  
 1. Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.  
 2. University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.  
 3. University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.  
 4. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation:  
 Robert Detrano, M.D., Ph.D.

Loading the initial required libraries.

library(tidyr)  
library(dplyr)  
library(ggplot2)  
library(broom)  
library(randomForest)

# Q1: Data importing and cleaning steps are explained in the text and in the DataCamp exercises.(Tell me why you are doing the data cleaning activities that you perform). Follow a logical process.

## Explaining the steps on how to import and clean my data

### A brief overview about the source data

The actual database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. But we would be using all the other datasets also. The “goal” field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

The names and social security numbers of the patients were removed from the database, replaced with dummy values. We are using all four processed files which also exist in the dataset directory.

### Summary of the various datasets are given below.

Database: # of instances:  
 Cleveland: 303  
 Hungarian: 294  
 Switzerland: 123  
 Long Beach VA: 200

### Attribute Information: The attributes that are defined in the below datasets are defined here.

It also shows the position of the attributes in the actual files: Only 14 used: V1. #3 (age) : Age in years  
V2. #4 (sex) : sex (1 = male; 0 = female) V3. #9 (cp) : chest pain type (1:typical angina, 2:atypical angina, 3:non-anginal pain, 4: asymptomatic) V4. #10 (trestbps) : resting blood pressure (in mm Hg on admission to the hospital) V5. #12 (chol) : serum cholestoral in mg/dl V6. #16 (fbs) : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) V7. #19 (restecg) : resting electrocardiographic results 0: normal, 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) 2: showing probable or definite left ventricular hypertrophy by Estes’ criteria V8. #32 (thalach) : maximum heart rate achieved V9. #38 (exang) : exercise induced angina (1 = yes; 0 = no) V10. #40 (oldpeak) : ST depression induced by exercise relative to rest  
V11. #41 (slope) : the slope of the peak exercise ST segment 1: upsloping 2: flat 3: downsloping V12. #44 (ca) : number of major vessels (0-3) colored by flourosopy V13. #51 (thal) : 3 = normal; 6 = fixed defect; 7 = reversable defect  
V14. #58 (num) : the predicted attribute diagnosis of heart disease (angiographic disease status) 0: < 50% diameter narrowing (No heart disease) 1: > 50% diameter narrowing ( Yes Heart disease)

### Explanation for why clean up is necessary

Before we apply any algorithm on our data, it is obvious that the data should be tidy or structured. But in the real world, the data mostly we initially see is unstructured. So in order to make it tidy and to further apply any algorithm to derive the insights, data has to be cleaned. The major reason why the data is not tidy is because of the presence of missing values and outliers. So as a first step I would start by importing each of the 4 files which have already being processed, into 4 different data frames and looking at those separately in order to get a better understanding of the values of the attributes and how they are distributed.

### Given below is the detailed step-by-step process of cleaning the data.

#populating the cleveland file into dataframe  
wd <- getwd()  
cleveland\_f <- "processed.cleveland.data"  
path\_to\_file <- paste(wd,'/dataset/',cleveland\_f, sep = "")  
## path\_to\_file  
  
cleveland\_df <- read.csv(path\_to\_file, header = FALSE)  
## dim(cleveland\_df)  
## summary(cleveland\_df)  
## head(cleveland\_df)  
  
#populating the Hungarian file into dataframe  
hungarian\_f <- "processed.hungarian.data"  
path\_to\_file <- paste(wd,'/dataset/',hungarian\_f, sep = "")  
## path\_to\_file  
  
hungarian\_df <- read.csv(path\_to\_file, header = FALSE)  
## dim(hungarian\_df)  
## summary(hungarian\_df)  
## head(hungarian\_df)  
  
#populating the Switzerland file into dataframe  
switzerland\_f <- "processed.switzerland.data"  
path\_to\_file <- paste(wd,'/dataset/',switzerland\_f, sep = "")  
## path\_to\_file  
  
switzerland\_df <- read.csv(path\_to\_file, header = FALSE)  
## dim(switzerland\_df)  
## summary(switzerland\_df)  
## head(switzerland\_df)  
  
#populating the Long Beach, CA data file into dataframe  
long\_beach\_f <- "processed.va.data"  
path\_to\_file <- paste(wd,'/dataset/',long\_beach\_f, sep = "")  
## path\_to\_file  
  
long\_beach\_df <- read.csv(path\_to\_file, header = FALSE)  
## dim(long\_beach\_df)  
## summary(long\_beach\_df)  
## head(long\_beach\_df)  
  
dim(cleveland\_df)

[1] 303 14

dim(hungarian\_df)

[1] 294 14

dim(switzerland\_df)

[1] 123 14

dim(long\_beach\_df)

[1] 200 14 After having a initial look at the datasets, now adding the 4 individual dataframes into separate data datasets and also adding the column names for each dataset in order to do some more analysis on each of those, as we know that the data do not have the column names.

# Creating new data frames with the existing dataframe towards our final dataset.  
data.cle <- cleveland\_df  
data.hun <- hungarian\_df  
data.sw <- switzerland\_df  
data.va <- long\_beach\_df

### Adding the new columns for each of the data source name.

Also adding an extra variable to keep the Source name along with the data set. Now looking at the data briefly. This would help in the model building phase after I combine all the datasets.

names(data.cle) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "out")  
names(data.hun) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "out")  
names(data.sw) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "out")  
names(data.va) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restecg", "thalach", "exang", "oldpeak", "slope", "ca", "thal", "out")  
  
# Adding the extra column to identify the source for the file  
data1 <- data.cle %>% mutate(datasrc = "Cleveland")   
data2 <- data.hun %>% mutate(datasrc = "Hungarian")   
data3 <- data.sw %>% mutate(datasrc = "Switzerland")   
data4 <- data.va %>% mutate(datasrc = "Long Beach VA")  
  
head(data1)  
head(data2)  
head(data3)  
head(data4)

We are trying to clean up the dataset individually.

### Working on cleaning and tidying the Cleveland Dataset

Here for our problem, we are only going to attempt to distinguish the presence of heart disease (values 1,2,3,4) from absence of heart disease (value 0). Therefore, we replace all labels greater than 1 by 1. Then taking summary of each of the data sets. Showing below our findings on each of the various datasets.

data1$out[data1$out > 0] <- 1  
summary(data1)  
glimpse(data1)

We see that the data set is showing mean for categorical variables also. Hence we need to re-specify the column types. We know a categorical variable (a variable that takes on a finite amount of values) is a factor. As we can see, sex is incorrectly treated as a number when in reality it can only be 1 if male and 0 if female. We can use the transform method to change the in built type of each feature.

data1 <- transform(  
 data1,  
 age=as.integer(age),  
 sex=as.factor(sex),  
 cp=as.factor(cp),  
 trestbps=as.integer(trestbps),  
 chol=as.integer(chol),  
 fbs=as.factor(fbs),  
 restecg=as.factor(restecg),  
 thalach=as.integer(thalach),  
 exang=as.factor(exang),  
 oldpeak=as.numeric(oldpeak),  
 slope=as.factor(slope),  
 ca=as.factor(ca),  
 thal=as.factor(thal),  
 out=as.factor(out),  
 datasrc=as.character(datasrc)  
)  
summary(data1)

## age sex cp trestbps chol fbs   
## Min. :29.00 0: 97 1: 23 Min. : 94.0 Min. :126.0 0:258   
## 1st Qu.:48.00 1:206 2: 50 1st Qu.:120.0 1st Qu.:211.0 1: 45   
## Median :56.00 3: 86 Median :130.0 Median :241.0   
## Mean :54.44 4:144 Mean :131.7 Mean :246.7   
## 3rd Qu.:61.00 3rd Qu.:140.0 3rd Qu.:275.0   
## Max. :77.00 Max. :200.0 Max. :564.0   
## restecg thalach exang oldpeak slope ca thal   
## 0:151 Min. : 71.0 0:204 Min. :0.00 1:142 ? : 4 ? : 2   
## 1: 4 1st Qu.:133.5 1: 99 1st Qu.:0.00 2:140 0.0:176 3.0:166   
## 2:148 Median :153.0 Median :0.80 3: 21 1.0: 65 6.0: 18   
## Mean :149.6 Mean :1.04 2.0: 38 7.0:117   
## 3rd Qu.:166.0 3rd Qu.:1.60 3.0: 20   
## Max. :202.0 Max. :6.20   
## out datasrc   
## 0:164 Length:303   
## 1:139 Class :character   
## Mode :character   
##   
##   
##

glimpse(data1)

## Observations: 303  
## Variables: 15  
## $ age <int> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 56, 44, 52, …  
## $ sex <fct> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, …  
## $ cp <fct> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, 3, 2, 4, 3, 2, …  
## $ trestbps <int> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140, 140, 140, …  
## $ chol <int> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203, 192, 294, …  
## $ fbs <fct> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, …  
## $ restecg <fct> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, …  
## $ thalach <int> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155, 148, 153, …  
## $ exang <fct> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, …  
## $ oldpeak <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1, 0.4, 1.3, …  
## $ slope <fct> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 1, …  
## $ ca <fct> 0.0, 3.0, 2.0, 0.0, 0.0, 0.0, 2.0, 0.0, 1.0, 0.0, 0.0, 0.0, …  
## $ thal <fct> 6.0, 3.0, 7.0, 3.0, 3.0, 3.0, 3.0, 3.0, 7.0, 7.0, 6.0, 3.0, …  
## $ out <fct> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, …  
## $ datasrc <chr> "Cleveland", "Cleveland", "Cleveland", "Cleveland", "Clevela…

Now also we find that there are missing values for some of the variables in each dataset. Here we see missing/unknown values for the below variables. ca : number of major vessels (0-3) colored by flourosopy thal : 3 = normal; 6 = fixed defect; 7 = reversable defect  
Now after replacing the missing values (“?”) with NA we can use the colSums function to count the number of missing values. It shows that we have 4 and 2 rows with missing values for thal and ca respectively.

data1[ data1 == "?"] <- NA  
colSums(is.na(data1))

age sex cp trestbps chol fbs restecg thalach   
 0 0 0 0 0 0 0 0

exang oldpeak slope ca thal out datasrc 0 0 0 4 2 0 0 Let us have a look at those 6 rows and see if we could find anything interesting in those rows.

data1 %>% filter(is.na(ca) | is.na(thal))

age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal 1 53 0 3 128 216 0 2 115 0 0.0 1 0.0 2 52 1 3 138 223 0 0 169 0 0.0 1 3.0 3 43 1 4 132 247 1 2 143 1 0.1 2 7.0 4 52 1 4 128 204 1 0 156 1 1.0 2 0.0 5 58 1 2 125 220 0 0 144 0 0.4 2 7.0 6 38 1 3 138 175 0 0 173 0 0.0 1 3.0 out datasrc 1 0 Cleveland 2 0 Cleveland 3 1 Cleveland 4 1 Cleveland 5 0 Cleveland 6 0 Cleveland Now as we are not experts in this type of data let us take these 6 rows out from the cleveland dataset.Now looking at the new cleaner dataset we see there are no missing records now and the count of the dataset is 303 - 6 = 296. This is only being done on the Cleveland dataset as that seems to be missing the least number of variables. We would use this dataset set separately as our final data set for prediction purposes.

data11 <- data1 %>% filter(!is.na(ca)) %>% filter(!is.na(thal))  
colSums(is.na(data11))

## age sex cp trestbps chol fbs restecg thalach   
## 0 0 0 0 0 0 0 0   
## exang oldpeak slope ca thal out datasrc   
## 0 0 0 0 0 0 0

# Recasting the above 2 columns as factor shows the summary info correctly  
data11$ca <- factor(data11$ca)  
data11$thal <- factor(data11$thal)  
summary(data11)

## age sex cp trestbps chol fbs   
## Min. :29.00 0: 96 1: 23 Min. : 94.0 Min. :126.0 0:254   
## 1st Qu.:48.00 1:201 2: 49 1st Qu.:120.0 1st Qu.:211.0 1: 43   
## Median :56.00 3: 83 Median :130.0 Median :243.0   
## Mean :54.54 4:142 Mean :131.7 Mean :247.4   
## 3rd Qu.:61.00 3rd Qu.:140.0 3rd Qu.:276.0   
## Max. :77.00 Max. :200.0 Max. :564.0   
## restecg thalach exang oldpeak slope ca thal   
## 0:147 Min. : 71.0 0:200 Min. :0.000 1:139 0.0:174 3.0:164   
## 1: 4 1st Qu.:133.0 1: 97 1st Qu.:0.000 2:137 1.0: 65 6.0: 18   
## 2:146 Median :153.0 Median :0.800 3: 21 2.0: 38 7.0:115   
## Mean :149.6 Mean :1.056 3.0: 20   
## 3rd Qu.:166.0 3rd Qu.:1.600   
## Max. :202.0 Max. :6.200   
## out datasrc   
## 0:160 Length:297   
## 1:137 Class :character   
## Mode :character   
##   
##   
##

### Repeating the same above processes on the remaining dataset and coming up with a clean/tidy dataset

## Hungarian Dataset  
data2$out[data2$out > 1] <- 1  
## summary(data2)  
## glimpse(data2)  
  
data2 <- transform(  
 data2,  
 age=as.integer(age),  
 sex=as.factor(sex),  
 cp=as.factor(cp),  
 trestbps=as.integer(trestbps),  
 chol=as.integer(chol),  
 fbs=as.factor(fbs),  
 restecg=as.factor(restecg),  
 thalach=as.integer(thalach),  
 exang=as.factor(exang),  
 oldpeak=as.numeric(oldpeak),  
 slope=as.factor(slope),  
 ca=as.factor(ca),  
 thal=as.factor(thal),  
 out=as.factor(out),  
 datasrc=as.character(datasrc)  
)  
## summary(data2)  
## glimpse(data2)  
  
data2[ data2 == "?"] <- NA  
colSums(is.na(data2))

## Swiss Dataset  
data3$out[data3$out > 1] <- 1  
## summary(data2)  
## glimpse(data2)  
  
data3 <- transform(  
 data3,  
 age=as.integer(age),  
 sex=as.factor(sex),  
 cp=as.factor(cp),  
 trestbps=as.integer(trestbps),  
 chol=as.integer(chol),  
 fbs=as.factor(fbs),  
 restecg=as.factor(restecg),  
 thalach=as.integer(thalach),  
 exang=as.factor(exang),  
 oldpeak=as.numeric(oldpeak),  
 slope=as.factor(slope),  
 ca=as.factor(ca),  
 thal=as.factor(thal),  
 out=as.factor(out),  
 datasrc=as.character(datasrc)  
)  
## summary(data2)  
## glimpse(data2)  
  
data3[ data3 == "?"] <- NA  
colSums(is.na(data3))

## Long Beach, VA Dataset  
data4$out[data4$out > 1] <- 1  
## summary(data2)  
## glimpse(data2)  
  
data4 <- transform(  
 data4,  
 age=as.integer(age),  
 sex=as.factor(sex),  
 cp=as.factor(cp),  
 trestbps=as.integer(trestbps),  
 chol=as.integer(chol),  
 fbs=as.factor(fbs),  
 restecg=as.factor(restecg),  
 thalach=as.integer(thalach),  
 exang=as.factor(exang),  
 oldpeak=as.numeric(oldpeak),  
 slope=as.factor(slope),  
 ca=as.factor(ca),  
 thal=as.factor(thal),  
 out=as.factor(out),  
 datasrc=as.character(datasrc)  
)  
## summary(data2)  
## glimpse(data2)  
  
data4[ data4 == "?"] <- NA  
colSums(is.na(data4))

### Combining the dataframes together now.

This is a combined data set with all the 4 datasets combined into one big dataset.But after looking at the various data sets it became clear that except for the Cleveland dataset all others do have a lot of missing data for some of the variables. Showing the summary on the combined dataset it looks like that the dataset from Long Beach VA and Swiss has a significantly higher number of outcomes that shows more patients having heart disease(out = 1) than without (out = 0).

combined.data <- rbind(data1,data2,data3,data4)  
  
combined.data %>% group\_by(datasrc,out) %>% summarise(n())  
## glimpse(combined.data)  
## head(combined.data)  
colSums(is.na(combined.data))

## What does the final data set look like

With a clean dataset, show what the final data set looks like. However, do not print off a data frame with 200+ rows; show me the data in the most condensed form possible. So after doing the initial round of Exploratory Data Analysis, I am planning on using the Clean and Tidy Cleveland data set for my predictions. I will name it the final\_data. The summary detail is shown below. This dataset do not have any missing values or outliers.

final\_data <- data11  
  
glimpse(final\_data)

## Observations: 297  
## Variables: 15  
## $ age <int> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 56, 44, 52, …  
## $ sex <fct> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, …  
## $ cp <fct> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, 3, 2, 4, 3, 2, …  
## $ trestbps <int> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140, 140, 140, …  
## $ chol <int> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203, 192, 294, …  
## $ fbs <fct> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, …  
## $ restecg <fct> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, …  
## $ thalach <int> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155, 148, 153, …  
## $ exang <fct> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, …  
## $ oldpeak <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1, 0.4, 1.3, …  
## $ slope <fct> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 1, …  
## $ ca <fct> 0.0, 3.0, 2.0, 0.0, 0.0, 0.0, 2.0, 0.0, 1.0, 0.0, 0.0, 0.0, …  
## $ thal <fct> 6.0, 3.0, 7.0, 3.0, 3.0, 3.0, 3.0, 3.0, 7.0, 7.0, 6.0, 3.0, …  
## $ out <fct> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, …  
## $ datasrc <chr> "Cleveland", "Cleveland", "Cleveland", "Cleveland", "Clevela…

colSums(is.na(final\_data))

## age sex cp trestbps chol fbs restecg thalach   
## 0 0 0 0 0 0 0 0   
## exang oldpeak slope ca thal out datasrc   
## 0 0 0 0 0 0 0

summary(final\_data)

## age sex cp trestbps chol fbs   
## Min. :29.00 0: 96 1: 23 Min. : 94.0 Min. :126.0 0:254   
## 1st Qu.:48.00 1:201 2: 49 1st Qu.:120.0 1st Qu.:211.0 1: 43   
## Median :56.00 3: 83 Median :130.0 Median :243.0   
## Mean :54.54 4:142 Mean :131.7 Mean :247.4   
## 3rd Qu.:61.00 3rd Qu.:140.0 3rd Qu.:276.0   
## Max. :77.00 Max. :200.0 Max. :564.0   
## restecg thalach exang oldpeak slope ca thal   
## 0:147 Min. : 71.0 0:200 Min. :0.000 1:139 0.0:174 3.0:164   
## 1: 4 1st Qu.:133.0 1: 97 1st Qu.:0.000 2:137 1.0: 65 6.0: 18   
## 2:146 Median :153.0 Median :0.800 3: 21 2.0: 38 7.0:115   
## Mean :149.6 Mean :1.056 3.0: 20   
## 3rd Qu.:166.0 3rd Qu.:1.600   
## Max. :202.0 Max. :6.200   
## out datasrc   
## 0:160 Length:297   
## 1:137 Class :character   
## Mode :character   
##   
##   
##

## Questions for future step

What do you not know how to do right now that you need to learn to answer your questions?  
I do not know confidently how to manage datasets having missing values for some of the variables. I am planning on learning that for my next steps. So as of now I am planning on using the Cleveland data set as that is mostly complete and so would help in the model building.  
  
Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.  
I am planning on applying random forest model for this classification problem along with the Generalized Linear Model and see the effectiveness of each models prediction.   
Also before I apply any machine learning techniques I am planning on visualizing the dataset using scatter plots for some of the variables.(e.g, age,sex and some others)   
Also I would like to find the correlation and test between the various predictor variables.

# Step: 3

## Discuss how you plan to uncover new information in the data that is not self-evident.

Taking a summary of the final dataset I see that age(int), trestbps - resting blood pressure (on admission), chol - serum cholestorol are all measured against different scales and should be compared with each other.

glimpse(final\_data)

## Observations: 297  
## Variables: 15  
## $ age <int> 63, 67, 67, 37, 41, 56, 62, 57, 63, 53, 57, 56, 56, 44, 52, …  
## $ sex <fct> 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, …  
## $ cp <fct> 1, 4, 4, 3, 2, 2, 4, 4, 4, 4, 4, 2, 3, 2, 3, 3, 2, 4, 3, 2, …  
## $ trestbps <int> 145, 160, 120, 130, 130, 120, 140, 120, 130, 140, 140, 140, …  
## $ chol <int> 233, 286, 229, 250, 204, 236, 268, 354, 254, 203, 192, 294, …  
## $ fbs <fct> 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, …  
## $ restecg <fct> 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, …  
## $ thalach <int> 150, 108, 129, 187, 172, 178, 160, 163, 147, 155, 148, 153, …  
## $ exang <fct> 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, …  
## $ oldpeak <dbl> 2.3, 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 0.6, 1.4, 3.1, 0.4, 1.3, …  
## $ slope <fct> 3, 2, 2, 3, 1, 1, 3, 1, 2, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 1, …  
## $ ca <fct> 0.0, 3.0, 2.0, 0.0, 0.0, 0.0, 2.0, 0.0, 1.0, 0.0, 0.0, 0.0, …  
## $ thal <fct> 6.0, 3.0, 7.0, 3.0, 3.0, 3.0, 3.0, 3.0, 7.0, 7.0, 6.0, 3.0, …  
## $ out <fct> 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, …  
## $ datasrc <chr> "Cleveland", "Cleveland", "Cleveland", "Cleveland", "Clevela…

summary(final\_data)

## age sex cp trestbps chol fbs   
## Min. :29.00 0: 96 1: 23 Min. : 94.0 Min. :126.0 0:254   
## 1st Qu.:48.00 1:201 2: 49 1st Qu.:120.0 1st Qu.:211.0 1: 43   
## Median :56.00 3: 83 Median :130.0 Median :243.0   
## Mean :54.54 4:142 Mean :131.7 Mean :247.4   
## 3rd Qu.:61.00 3rd Qu.:140.0 3rd Qu.:276.0   
## Max. :77.00 Max. :200.0 Max. :564.0   
## restecg thalach exang oldpeak slope ca thal   
## 0:147 Min. : 71.0 0:200 Min. :0.000 1:139 0.0:174 3.0:164   
## 1: 4 1st Qu.:133.0 1: 97 1st Qu.:0.000 2:137 1.0: 65 6.0: 18   
## 2:146 Median :153.0 Median :0.800 3: 21 2.0: 38 7.0:115   
## Mean :149.6 Mean :1.056 3.0: 20   
## 3rd Qu.:166.0 3rd Qu.:1.600   
## Max. :202.0 Max. :6.200   
## out datasrc   
## 0:160 Length:297   
## 1:137 Class :character   
## Mode :character   
##   
##   
##

Trying to see if there is a high correlation among the independant variables that could lead to multicollinearity problem For this we would use the library called psych. Also we are trying to find the correlation among the numeric variables that are independant to each other. In the below diagram, we see the scatter plot showing the correlation coefficient among the independant variables. It shows that there are no significant correlation between the variables and hence they are truly independant. The highest correlation is 0.29 between age and resting blood pressure and the next is between age and serum cholestorol is 0.20. But none of these are high. (>.80). Hence this relationship is not self evident from the data which I found out using the tool. So no multicollinerity problem exists here.

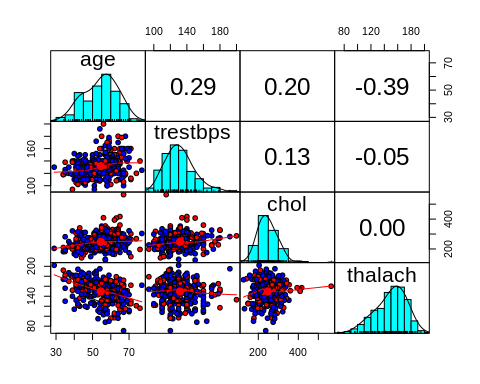
library(psych)  
  
mydata.cor <- final\_data %>% select(age,trestbps,chol,thalach,sex)  
  
head(mydata.cor)

## age trestbps chol thalach sex  
## 1 63 145 233 150 1  
## 2 67 160 286 108 1  
## 3 67 120 229 129 1  
## 4 37 130 250 187 1  
## 5 41 130 204 172 0  
## 6 56 120 236 178 1

cor(mydata.cor[,-5])

## age trestbps chol thalach  
## age 1.0000000 0.29047626 2.026435e-01 -3.945629e-01  
## trestbps 0.2904763 1.00000000 1.315357e-01 -4.910766e-02  
## chol 0.2026435 0.13153571 1.000000e+00 -7.456799e-05  
## thalach -0.3945629 -0.04910766 -7.456799e-05 1.000000e+00

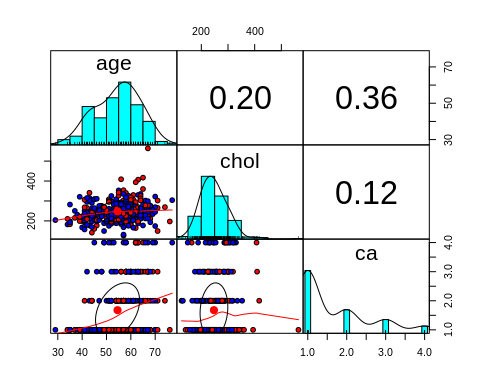
# Not including the binary column Sex for correlation test  
pairs.panels(mydata.cor[,-5],  
 gap = 0,  
 bg = c("red","blue")[mydata.cor$sex],  
 pch = 21)

 ### Looking for multicollinerity among the following variables

library(psych)  
  
## glimpse(final\_data)  
  
mydata.cor1 <- final\_data %>% select(age,chol,ca,sex)  
  
head(mydata.cor1)

## age chol ca sex  
## 1 63 233 0.0 1  
## 2 67 286 3.0 1  
## 3 67 229 2.0 1  
## 4 37 250 0.0 1  
## 5 41 204 0.0 0  
## 6 56 236 0.0 1

## cor(mydata.cor1[,-5])  
  
# Not including the binary column Sex for correlation test  
pairs.panels(mydata.cor1[,-4],  
 gap = 0,  
 bg = c("red","blue")[mydata.cor1$sex],  
 pch = 21)

 ### Plotting the data to see any story

## glimpse(final\_data)  
  
mydata.plot1 <- final\_data %>% select(age,chol,ca,sex)  
  
head(mydata.plot1)

## age chol ca sex  
## 1 63 233 0.0 1  
## 2 67 286 3.0 1  
## 3 67 229 2.0 1  
## 4 37 250 0.0 1  
## 5 41 204 0.0 0  
## 6 56 236 0.0 1

ggplot(data = mydata.plot1, aes(x = age, y = chol, color = ca)) +   
 geom\_point() +   
 facet\_wrap(~sex, nrow = 2)



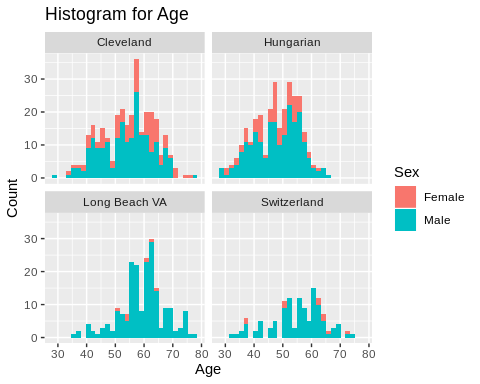
## What are different ways you could look at this data to answer the questions you want to answer?

Using the final dataset that I considered in my last analysis is the Cleveland dataset as it has almost all the variables with complete values. The assumption of the existence of heart disease here is since we’re trying to predict significant heart disease those with higher level stenoses(>50%) would be treated as heart disease. I am not able to find any other values as the dataset has been made with the 50% diameter narrowing as the cut off point. First I am trying to plot histograms to show how the age and sex are related to the presence of the heart disease for the combined dataset. As per my earlier data analysis age and sex variables did not have any missing values in the combined dataset. It shows that Long Beach, VA and Swiss datasets females have less representation like 6 and 10 only. Also we can see that

combined.data %>% group\_by(datasrc,sex) %>% summarise(n())

## # A tibble: 8 x 3  
## # Groups: datasrc [4]  
## datasrc sex `n()`  
## <chr> <fct> <int>  
## 1 Cleveland 0 97  
## 2 Cleveland 1 206  
## 3 Hungarian 0 81  
## 4 Hungarian 1 213  
## 5 Long Beach VA 0 6  
## 6 Long Beach VA 1 194  
## 7 Switzerland 0 10  
## 8 Switzerland 1 113

p <- ggplot(data = combined.data, aes(age))   
p <- p + geom\_histogram(aes(fill = sex))  
p <- p+ scale\_fill\_discrete(name="Sex",  
 labels=c("Female", "Male"))  
p <- p + labs(title="Histogram for Age") + labs(x="Age", y="Count")  
p + facet\_wrap(~datasrc)

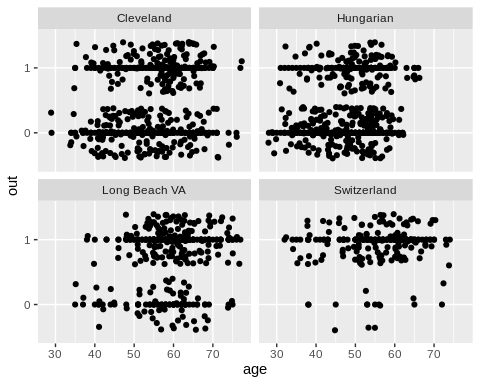
 ## Do you plan to slice and dice the data in different ways, create new variables, or join separate data frames to create new summary information? Explain. ## How could you summarize your data to answer key questions?

Now looking at the summary of the data by Output, it shows that surprisingly both Swiss and Long beach datasets have more people with Heart disease than not. It could be that only the patients who had severe problem did only come to do the checkup. So my decision of picking up the Cleveland datset for my model training was a good choice as that would be less biased.

combined.data %>% group\_by(datasrc,out) %>% summarise(n())

## # A tibble: 8 x 3  
## # Groups: datasrc [4]  
## datasrc out `n()`  
## <chr> <fct> <int>  
## 1 Cleveland 0 164  
## 2 Cleveland 1 139  
## 3 Hungarian 0 188  
## 4 Hungarian 1 106  
## 5 Long Beach VA 0 51  
## 6 Long Beach VA 1 149  
## 7 Switzerland 0 8  
## 8 Switzerland 1 115

p <- ggplot(combined.data, aes(x = age, y = out)) + geom\_point() + geom\_jitter()  
p + facet\_wrap(~datasrc)



## What types of plots and tables will help you to illustrate the findings to your questions? Ensure that all graph plots have axis titles, legend if necessary, scales are appropriate, appropriate geoms used, etc.).

I have used the above correlation plots to see if there are any significant collinearity between the various numeric predictors.

## What do you not know how to do right now that you need to learn to answer your questions?

I am trying to understand if all the predictors are significant for the output variable. I am planning to do some more tests to get that clarified.

## Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.

My approach here is to apply logistic regression and random forest model, and wish to analyse the likelihood of a patient having heart disease based on changes to the above independent variables. Note that in this case, we are not looking to obtain a specific coefficient value but rather an odds ratio; a ratio that tells us the likelihood of an occurance of heart disease based on any independent variable.

Let us start building the model. We would split the data first into Training and Test Set with a 80, 20 split.

# Loading package caret for data splitting  
library(caret)  
library(e1071)  
set.seed(1234)  
  
# Creating the data partition with 80 20 split between training and test sets  
in\_train <- createDataPartition(final\_data$out, p = 0.8, list = FALSE)  
  
training <- final\_data[in\_train,]  
test <- final\_data[-in\_train,]  
  
# Looking at the training set  
glimpse(training)

## Observations: 238  
## Variables: 15  
## $ age <int> 67, 67, 37, 41, 56, 62, 63, 57, 56, 44, 52, 57, 48, 54, 49, …  
## $ sex <fct> 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, …  
## $ cp <fct> 4, 4, 3, 2, 2, 4, 4, 4, 3, 2, 3, 3, 2, 4, 2, 1, 2, 4, 3, 3, …  
## $ trestbps <int> 160, 120, 130, 130, 120, 140, 130, 140, 130, 120, 172, 150, …  
## $ chol <int> 286, 229, 250, 204, 236, 268, 254, 192, 256, 263, 199, 168, …  
## $ fbs <fct> 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ restecg <fct> 2, 2, 0, 2, 0, 2, 2, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, …  
## $ thalach <int> 108, 129, 187, 172, 178, 160, 147, 148, 142, 173, 162, 174, …  
## $ exang <fct> 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, …  
## $ oldpeak <dbl> 1.5, 2.6, 3.5, 1.4, 0.8, 3.6, 1.4, 0.4, 0.6, 0.0, 0.5, 1.6, …  
## $ slope <fct> 2, 2, 3, 1, 1, 3, 2, 2, 2, 1, 1, 1, 3, 1, 1, 2, 2, 2, 2, 1, …  
## $ ca <fct> 3.0, 2.0, 0.0, 0.0, 0.0, 2.0, 1.0, 0.0, 1.0, 0.0, 0.0, 0.0, …  
## $ thal <fct> 3.0, 7.0, 3.0, 3.0, 3.0, 3.0, 7.0, 6.0, 6.0, 7.0, 7.0, 3.0, …  
## $ out <fct> 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, …  
## $ datasrc <chr> "Cleveland", "Cleveland", "Cleveland", "Cleveland", "Clevela…

#Training a random Forest Model  
  
fit\_rf <- train(out~., method = "rf", data = training[-15], trcontrol = trainControl(method = "none"))  
  
# Building a confusion Matrix on the random forest model  
  
cm\_rf <- confusionMatrix(predict(fit\_rf, test[-15]), test$out)  
  
# Training a Generalised Logistic model with binomial distribution  
  
fit\_glm <- train(out~., method = "glm",family = "binomial", data = training[-15])  
  
# Building the confusion matrix on the logistic regression model  
cm\_glm <- confusionMatrix(predict(fit\_glm, test[-15]), test$out)  
  
# Summary and the confusion matrix on the random forest model  
summary(fit\_rf)

## Length Class Mode   
## call 5 -none- call   
## type 1 -none- character  
## predicted 238 factor numeric   
## err.rate 1500 -none- numeric   
## confusion 6 -none- numeric   
## votes 476 matrix numeric   
## oob.times 238 -none- numeric   
## classes 2 -none- character  
## importance 20 -none- numeric   
## importanceSD 0 -none- NULL   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 14 -none- list   
## y 238 factor numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## xNames 20 -none- character  
## problemType 1 -none- character  
## tuneValue 1 data.frame list   
## obsLevels 2 -none- character  
## param 1 -none- list

cm\_rf

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 28 4  
## 1 4 23  
##   
## Accuracy : 0.8644   
## 95% CI : (0.7502, 0.9396)  
## No Information Rate : 0.5424   
## P-Value [Acc > NIR] : 1.458e-07   
##   
## Kappa : 0.7269   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.8750   
## Specificity : 0.8519   
## Pos Pred Value : 0.8750   
## Neg Pred Value : 0.8519   
## Prevalence : 0.5424   
## Detection Rate : 0.4746   
## Detection Prevalence : 0.5424   
## Balanced Accuracy : 0.8634   
##   
## 'Positive' Class : 0   
##

# Summary and the confusion matrix on the logistic regression model  
summary(fit\_glm)

##   
## Call:  
## NULL  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.1136 -0.4633 -0.1296 0.3128 2.8486   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -8.598912 3.288852 -2.615 0.008934 \*\*   
## age -0.008504 0.028981 -0.293 0.769189   
## sex1 1.438169 0.601467 2.391 0.016798 \*   
## cp2 1.711488 0.890556 1.922 0.054629 .   
## cp3 0.916825 0.785270 1.168 0.242997   
## cp4 2.746182 0.801504 3.426 0.000612 \*\*\*  
## trestbps 0.020829 0.012885 1.616 0.105991   
## chol 0.008443 0.005565 1.517 0.129260   
## fbs1 -0.952437 0.710604 -1.340 0.180141   
## restecg1 12.526380 944.119757 0.013 0.989414   
## restecg2 0.259319 0.436477 0.594 0.552433   
## thalach -0.011167 0.012617 -0.885 0.376098   
## exang1 0.467080 0.501182 0.932 0.351359   
## oldpeak 0.378638 0.279652 1.354 0.175749   
## slope2 1.645072 0.530657 3.100 0.001935 \*\*   
## slope3 1.270777 1.066384 1.192 0.233391   
## ca1.0 2.450232 0.591359 4.143 3.42e-05 \*\*\*  
## ca2.0 3.801541 0.932499 4.077 4.57e-05 \*\*\*  
## ca3.0 3.030888 1.241329 2.442 0.014620 \*   
## thal6.0 -0.106036 0.875823 -0.121 0.903635   
## thal7.0 1.445934 0.507816 2.847 0.004408 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 328.58 on 237 degrees of freedom  
## Residual deviance: 149.92 on 217 degrees of freedom  
## AIC: 191.92  
##   
## Number of Fisher Scoring iterations: 14

cm\_glm

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 25 2  
## 1 7 25  
##   
## Accuracy : 0.8475   
## 95% CI : (0.7301, 0.9278)  
## No Information Rate : 0.5424   
## P-Value [Acc > NIR] : 7.195e-07   
##   
## Kappa : 0.6971   
##   
## Mcnemar's Test P-Value : 0.1824   
##   
## Sensitivity : 0.7812   
## Specificity : 0.9259   
## Pos Pred Value : 0.9259   
## Neg Pred Value : 0.7812   
## Prevalence : 0.5424   
## Detection Rate : 0.4237   
## Detection Prevalence : 0.4576   
## Balanced Accuracy : 0.8536   
##   
## 'Positive' Class : 0   
##

Since this is a classification problem, we use a confusion matrix to evaluate the performance of our model. Recall that values on the diagonal correspond to true positives and true negatives (correct predictions) whereas the others correspond to false positives and false negatives. Looking at the confusion matrix it looks like the predictions that I am getting from both the models are on the high accuracy (> 80%).

Analysis of the 2 models based on the confusion matrix data. Looking at the information from the confusion matrix we can interpret the quality of the classifier. In Total the Test Set had 59 observations while the training set had 238 observations. The Accuracy is the proportion of the cases that were classified correctly. So we can see that for the random forest model (28+23)=51 times out of 59 it was able to predict correctly. Which is 86.44% For the Logistic Regression that number is (25+25) = 50 out of 59 which is 84.75% Also for the both the models the accuracy is more than the No Information Rate and which is about 54% and that is good for any model.

Recall is calculated by out of all the positive cases, how much was predicted correctly. In our case the random forest has a Recall of 87.5% i.e., 28 out of 32 For the Logistic regression that value is 78.12% 25 out of 32.

Precision is out of all the positive cases we have predicted correctly, how many are actually positive. In our case the random forest has a precision is 85.19% which is 28 out of 32. For the Logistic regression that value is 92.59% which is 25 out of 27

So I can conclude that overall the random forest was a better fit than the logictic regression model.