Heart Data Exploratory Data Analysis

Harper Schwab

9/17/2020

Overview

Angina pectoris, commonly known as chronic chest pain,is a heart disease that occurs when the heart does not get enough blood. It is usually caused by strained artieries due to closure or blockage, called ischemia. Angina is not related to heart burn or lung infection. Commonly, angina pectoris is experienced durring exercise, or intense emotions. That is because the heart works harder durring those times, the added strain of the blocked arteries causes discomfort and pain. Treatments include rest and nitroglycerin. The nitroglycerin works to relax the arteries and blood vessels to increase the blood supply.

Walkthrough of Data Cleaning

Before we begin we need to import the tools we will use for the data analysis. To start, data is almost always messy. After the library imports, take a look at the first few lines of the heart data.

```
#Setup and clear
rm(list=ls())
setwd("C:/Users/Harper Schwab/Desktop/RFolder")
#Load myfunctions.R
source("C:/Users/Harper Schwab/Desktop/RFolder/myfunctions.R")
#load a library
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------
----------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------------------------------
----------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
#
#importing data
heart <- read.csv("dirtyheart.csv", header = TRUE)
#
summary(heart)
```

```
##       age             sex               cp           trestbps
##  Min.   : 0.00   Min.   :0.0000   Min.   :1.000   Min.   : 94.0
##  1st Qu.:46.00   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   :53.08   Mean   :0.6768   Mean   :3.158   Mean   :131.7
##  3rd Qu.:60.50   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:140.0
##  Max.   :77.00   Max.   :1.0000   Max.   :4.000   Max.   :200.0
##  NA's   :12      NA's   :6                        NA's   :13
##       chol            fbs            restecg          thalach
##  Min.   :126.0   Min.   :0.0000   Min.   :0.0000   Min.   : 71.0
##  1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:132.5
##  Median :240.0   Median :0.0000   Median :1.0000   Median :153.0
##  Mean   :245.2   Mean   :0.1515   Mean   :0.5217   Mean   :149.4
##  3rd Qu.:274.0   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
##  Max.   :564.0   Max.   :1.0000   Max.   :2.0000   Max.   :202.0
##  NA's   :13      NA's   :6        NA's   :4        NA's   :8
##      exang           oldpeak         slope             ca
##  Min.   :0.0000   Min.   :0.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00   1st Qu.:1.000   1st Qu.:0.0000
##  Median :0.0000   Median :0.80   Median :1.000   Median :0.0000
##  Mean   :0.3267   Mean   :1.05   Mean   :1.403   Mean   :0.7322
##  3rd Qu.:1.0000   3rd Qu.:1.60   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :6.20   Max.   :2.000   Max.   :4.0000
##  NA's   :3        NA's   :5      NA's   :5       NA's   :8
##       thal           target
##  Min.   :3.000   Min.   :0.000
##  1st Qu.:3.000   1st Qu.:0.000
##  Median :3.000   Median :1.000
##  Mean   :4.734   Mean   :0.539
##  3rd Qu.:7.000   3rd Qu.:1.000
##  Max.   :7.000   Max.   :1.000
##  NA's   :2       NA's   :8
```

You can tell a few things. The names are confusing and some data is unknown, like what the binary values are telling us. Here are a few other demographic commands.

```
names(heart)
```

```
##  [1] "age"      "sex"      "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "target"
```

```
str(heart)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 NA 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: int  145 130 130 120 120 140 NA 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : int  6 3 7 3 3 3 3 3 7 7 ...
##  $ target  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
tail(heart)
```

```
##     age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 298  59   1  4      164  176   1       0      90     0     1.0     1  2    7
## 299  57   0  1      140  241   0      NA     123     1     0.2     1  0    7
## 300  45   1  4      110  264   0       1     132     0     1.2     1  0    7
## 301  68   1  4      144  193   1       1     141     0     3.4     1  2    7
## 302   0   1  2      130  131   0       1     115     1     1.2     1  1    3
## 303  57   0  3      130  236   0       0     174     0     0.0     1  1    3
##     target
## 298      0
## 299      0
## 300      0
## 301      0
## 302      0
## 303      0
```

```
class(heart)
```

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

```
dim(heart)
```

```
## [1] 303  14
```

Renaming Columns

The first thing I like to do to make my future easier is to rename the columns.

```
names(heart)
```

```
##  [1] "age"      "sex"      "cp"       "trestbps" "chol"     "fbs"
##  [7] "restecg"  "thalach"  "exang"    "oldpeak"  "slope"    "ca"
## [13] "thal"     "target"
```

```
names(heart) <- c('Age', 'Sex', 'ChestPain', 'RestBP', 'Cholestoral', 'FastingBS',
                  'RestECG', 'MaxHR', 'ExAngina', 'OldPeak', 'StSlope', 'NumMajorVessels', 'Thal',
'HeartAttack')
names(heart)
```

```
##  [1] "Age"            "Sex"            "ChestPain"      "RestBP"
##  [5] "Cholestoral"    "FastingBS"      "RestECG"        "MaxHR"
##  [9] "ExAngina"       "OldPeak"        "StSlope"        "NumMajorVessels"
## [13] "Thal"           "HeartAttack"
```

Filling in missing values

We cant do propper analysis with missing values. There are a few ways to replace them that I had to do with this dataset. The first was finding an average of a column and replacing the missing values with the mean.

```
#For age, RestBP, Cholestoral, OldPeak and MaxHR, will input averages:
heart$Age <- ifelse(is.na(heart$Age), mean(heart$Age, na.rm=TRUE), heart$Age)
heart$RestBP <- ifelse(is.na(heart$RestBP), mean(heart$RestBP, na.rm=TRUE), heart$RestBP)
heart$Cholestoral <- ifelse(is.na(heart$Cholestoral), mean(heart$Cholestoral, na.rm=TRUE), heart$C
holestoral)
heart$MaxHR <- ifelse(is.na(heart$MaxHR), mean(heart$MaxHR, na.rm=TRUE), heart$MaxHR)
heart$OldPeak <- ifelse(is.na(heart$OldPeak), mean(heart$OldPeak, na.rm=TRUE), heart$OldPeak)
```

Next is for binary values. I made it replace with a 1 or 0 randomly.

```
#Sex, FastingBS, RestECG, ExAngina, and HeartAttack are binary data so need to input either one or
zero.
heart$Sex <- ifelse(is.na(heart$Sex), sample(0:1,1), heart$Sex)
heart$FastingBS <- ifelse(is.na(heart$FastingBS), sample(0:1,1), heart$FastingBS)
heart$RestECG <- ifelse(is.na(heart$RestECG), sample(0:1,1), heart$RestECG)
heart$ExAngina <- ifelse(is.na(heart$ExAngina), sample(0:1,1), heart$ExAngina)
heart$HeartAttack <- ifelse(is.na(heart$HeartAttack), sample(0:1,1), heart$HeartAttack)
```

The next were special cases. Because they had specific values in either a range or a list replaced with a random value in the range, or a randome number for a list. First is the range, second is the list.

```
#StSlope is on a scale of 0:2 and NumMajorVessels is from 0:4 I will do similar to Binary but a la
rger random sample group
heart$StSlope <- ifelse(is.na(heart$StSlope), sample(0:2,1), heart$StSlope)
heart$NumMajorVessels <- ifelse(is.na(heart$NumMajorVessels), sample(0:4,1), heart$NumMajorVessel
s)
#
#Thal is a little different it uses 3, 6, or 7 s o I need it to choose one of those at random
#creating a list to choose from
list = c(3,6,7)
#now choosing randomly and replacing
heart$Thal <- ifelse(is.na(heart$Thal), sample(list,1), heart$Thal)
```

Renaming values in columns

Many of the data that we couldn't put an average in for are really factors. We need to replace the numerical data with words that corrospond with each value. If you follow my comments you can see what I am assigning to each value.

```
#Now renaming values to make them make sense.
#for Sex, 1=male and 0 =female
heart$Sex <- factor(heart$Sex, levels=c(0,1), labels=c("Female","Male"))
#heartattack is yes or no.
heart$HeartAttack <- factor(heart$HeartAttack, levels=c(0,1), labels=c("No","Yes"))
#ChestPain
# 1: typical angina
# 2: atypical angina
# 3: non-anginal pain
# 4: asymptomatic
heart$ChestPain <- factor(heart$ChestPain, levels=c(1,2,3,4), labels=c("Typical Angina","Atypical
 Angina", "Non-anginal Pain", "Asymptomatic"))
#
#For slope, 1: upsloping  2: flat 3: downsloping
heart$StSlope <- factor(heart$StSlope, levels=c(0,1,2), labels=c("Upsloping ","Flat", "Downslopin
g"))
#
#Fasting blood sugar > 120mg/d 1 is true and 0 is false
heart$FastingBS <- factor(heart$FastingBS, levels=c(0,1), labels=c("False","True"))
#
#RestECG 0 = normal, 1 is ST-T wave abnormality(Twave inversion/St change>0.05 mV). 2 is probable
 L ventricle hypertrophy
heart$RestECG <- factor(heart$RestECG, levels=c(0,1,2), labels=c("Normal"," St-T abnormality", "LV
Hypertrophy"))
#
#ExAgnina: Exercized induced Agina Yes/No 1= yes 0 =no
heart$ExAngina <- factor(heart$ExAngina, levels=c(0,1), labels=c("No","Yes"))
#
#Thal 3 = normal; 6 = fixed defect; 7 = reversable defect
heart$Thal <- factor(heart$Thal, levels=c(3,6,7), labels=c("Normal","Fixed Defect", "Reversable De
fect"))
#
head(heart)
```

```
##        Age    Sex        ChestPain RestBP Cholestoral FastingBS
## 1 63.00000   Male   Typical Angina    145         233      True
## 2 37.00000   Male     Asymptomatic    130         250     False
## 3 41.00000 Female     Asymptomatic    130         204     False
## 4 53.08247   Male Non-anginal Pain    120         236     False
## 5 57.00000 Female  Atypical Angina    120         354     False
## 6 57.00000   Male  Atypical Angina    140         192     False
##             RestECG MaxHR ExAngina OldPeak     StSlope NumMajorVessels
## 1            Normal   150       No     2.3  Upsloping               0
## 2  St-T abnormality   187       No     3.5  Upsloping               0
## 3            Normal   172       No     1.4 Downsloping               0
## 4  St-T abnormality   178       No     0.8 Downsloping               0
## 5  St-T abnormality   163      Yes     0.6 Downsloping               0
## 6  St-T abnormality   148       No     0.4        Flat               0
##                Thal HeartAttack
## 1      Fixed Defect         Yes
## 2            Normal         Yes
## 3 Reversable Defect         Yes
## 4            Normal         Yes
## 5            Normal         Yes
## 6            Normal         Yes
```

dplyr specific commands

dplyr is a great tool for data analysis and cleanup. Here are a few helpful commands to use.

Rename is useful to completely rename specific variables in order to make life easier or, just understand it better.

```
#using the rename command
#giving NumBloodVessels a beter name
rename(heart, "NumBCColored" = "NumMajorVessels")
#Renaming FBS because it doesn't have a real value just a t/F statement
rename(heart, "FastignBS>120mg/d" = "FastingBS")
```

Filter is great to create new smaller datasets with specific criteria. I wanted to see all people who had a heart attack. And I wanted to see people who had heart attacks and experienced Angina.

```
# filtering and creating new datasets
HeartAttack <- filter(heart, HeartAttack == "Yes") #group of all heart attacks
head(HeartAttack)
```

```
##        Age    Sex        ChestPain RestBP Cholestoral FastingBS
## 1 63.00000   Male   Typical Angina    145         233      True
## 2 37.00000   Male     Asymptomatic    130         250     False
## 3 41.00000 Female     Asymptomatic    130         204     False
## 4 53.08247   Male Non-anginal Pain    120         236     False
## 5 57.00000 Female  Atypical Angina    120         354     False
## 6 57.00000   Male  Atypical Angina    140         192     False
##             RestECG MaxHR ExAngina OldPeak     StSlope NumMajorVessels
## 1            Normal   150       No     2.3  Upsloping               0
## 2  St-T abnormality   187       No     3.5  Upsloping               0
## 3            Normal   172       No     1.4 Downsloping               0
## 4  St-T abnormality   178       No     0.8 Downsloping               0
## 5  St-T abnormality   163      Yes     0.6 Downsloping               0
## 6  St-T abnormality   148       No     0.4        Flat               0
##                Thal HeartAttack
## 1      Fixed Defect         Yes
## 2            Normal         Yes
## 3 Reversable Defect         Yes
## 4            Normal         Yes
## 5            Normal         Yes
## 6            Normal         Yes
```

```
HA_ExAg <- filter(
  heart,                    #Group for heart attack and exercise induced Angina
  HeartAttack == "Yes",
  ExAngina == "Yes"
)
head(HeartAttack)
```

```
##        Age    Sex        ChestPain RestBP Cholestoral FastingBS
## 1 63.00000   Male   Typical Angina    145         233      True
## 2 37.00000   Male     Asymptomatic    130         250     False
## 3 41.00000 Female     Asymptomatic    130         204     False
## 4 53.08247   Male Non-anginal Pain    120         236     False
## 5 57.00000 Female  Atypical Angina    120         354     False
## 6 57.00000   Male  Atypical Angina    140         192     False
##             RestECG MaxHR ExAngina OldPeak     StSlope NumMajorVessels
## 1            Normal   150       No     2.3  Upsloping               0
## 2  St-T abnormality   187       No     3.5  Upsloping               0
## 3            Normal   172       No     1.4 Downsloping               0
## 4  St-T abnormality   178       No     0.8 Downsloping               0
## 5  St-T abnormality   163      Yes     0.6 Downsloping               0
## 6  St-T abnormality   148       No     0.4        Flat               0
##                Thal HeartAttack
## 1      Fixed Defect         Yes
## 2            Normal         Yes
## 3 Reversable Defect         Yes
## 4            Normal         Yes
## 5            Normal         Yes
## 6            Normal         Yes
```

Select is great for creating new data sets by choosing specific columns you want on the new table.

```
#Using the Select Command
Agesmall <- select(heart, Age, Sex, RestBP ,HeartAttack)
head(Agesmall)
AgeChol <- select(heart, Age, Cholestoral)
head(AgeChol)
```

Arrange is used to order by size of one variable, the entire data frame. in the first line, I arranged the entire dataset by descending age.

```
# using arrange
heart <- arrange(heart, -Age, Sex)
AgeChol <- arrange(AgeChol, -Age, Cholestoral)
Agesmall <- arrange(Agesmall, -RestBP, Age)
head(AgeChol)
```

```
##   Age Cholestoral
## 1  77         304
## 2  74         269
## 3  71         149
## 4  71         265
## 5  71         302
## 6  70         174
```

```
head(Agesmall)
```

```
##        Age    Sex RestBP HeartAttack
## 1 56.00000 Female    200          No
## 2 54.00000   Male    192          No
## 3 55.00000 Female    180          No
## 4 64.00000 Female    180         Yes
## 5 68.00000   Male    180          No
## 6 53.08247   Male    178         Yes
```

Mutate is great to add new columns. I added a new column that showed how far away each subject was from the average age.

```
#Using Mutate to add a column that shows the distance form the mean age
heart <- mutate(
  heart,
  DifferenceFromMeanAge = abs(Age-mean(Age)))
head(heart)
```

```
##   Age    Sex        ChestPain RestBP Cholestoral FastingBS           RestECG
## 1  77   Male  Atypical Angina    125         304     False            Normal
## 2  74 Female     Asymptomatic    120         269     False            Normal
## 3  71 Female Non-anginal Pain    160         302     False  St-T abnormality
## 4  71 Female     Asymptomatic    110         265      True            Normal
## 5  71 Female     Asymptomatic    112         149     False  St-T abnormality
## 6  70   Male Non-anginal Pain    156         245     False            Normal
##   MaxHR ExAngina OldPeak     StSlope NumMajorVessels              Thal
## 1   162      Yes     0.0 Downsloping               3            Normal
## 2   121      Yes     0.2 Downsloping               1            Normal
## 3   162       No     0.4 Downsloping               2            Normal
## 4   130       No     0.0 Downsloping               1 Reversable Defect
## 5   125       No     1.6        Flat               0            Normal
## 6   143       No     0.0 Downsloping               0            Normal
##   HeartAttack DifferenceFromMeanAge
## 1          No              23.91753
## 2         Yes              20.91753
## 3         Yes              17.91753
## 4         Yes              17.91753
## 5         Yes              17.91753
## 6         Yes              16.91753
```

Summarize works to create a small summary od the criteria you asked. I wanted a summary of the averge person's blood presure and cholestoral, so I asked for the mean of resting blood pressure and mean of cholestoral.

```
#using Summarise
# summarize
average_person <- summarize(
  heart,
  meanrestBP = mean(RestBP),
  meanChol = mean(Cholestoral),
  na.rm = TRUE
)
average_person
```

```
##   meanrestBP meanChol na.rm
## 1   131.6862 245.2138  TRUE
```

The group_by command helps create a table where you can see values based on what category you select, in this example I want to see the mean resting blood pressure and mean cholestoral based on what type of chest pain the subject has.

```
#now using the group_by command
# group_by
grouped <- group_by(heart, ChestPain)
head(grouped)
```

```
## # A tibble: 6 x 15
## # Groups:   ChestPain [3]
##     Age Sex   ChestPain RestBP Cholestoral FastingBS RestECG MaxHR ExAngina
##   <dbl> <fct> <fct>      <dbl>       <dbl> <fct>     <fct>   <dbl> <fct>
## 1    77 Male  Atypical~    125         304 False     "Norma~   162 Yes
## 2    74 Fema~ Asymptom~    120         269 False     "Norma~   121 Yes
## 3    71 Fema~ Non-angi~    160         302 False     " St-T~   162 No
## 4    71 Fema~ Asymptom~    110         265 True      "Norma~   130 No
## 5    71 Fema~ Asymptom~    112         149 False     " St-T~   125 No
## 6    70 Male  Non-angi~    156         245 False     "Norma~   143 No
## # ... with 6 more variables: OldPeak <dbl>, StSlope <fct>,
## #   NumMajorVessels <int>, Thal <fct>, HeartAttack <fct>,
## #   DifferenceFromMeanAge <dbl>
```

```
Average_person_summary <- heart %>%
  group_by(ChestPain) %>%
  summarize(
    meanrestBP = mean(RestBP),
    meanChol = mean(Cholestoral)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

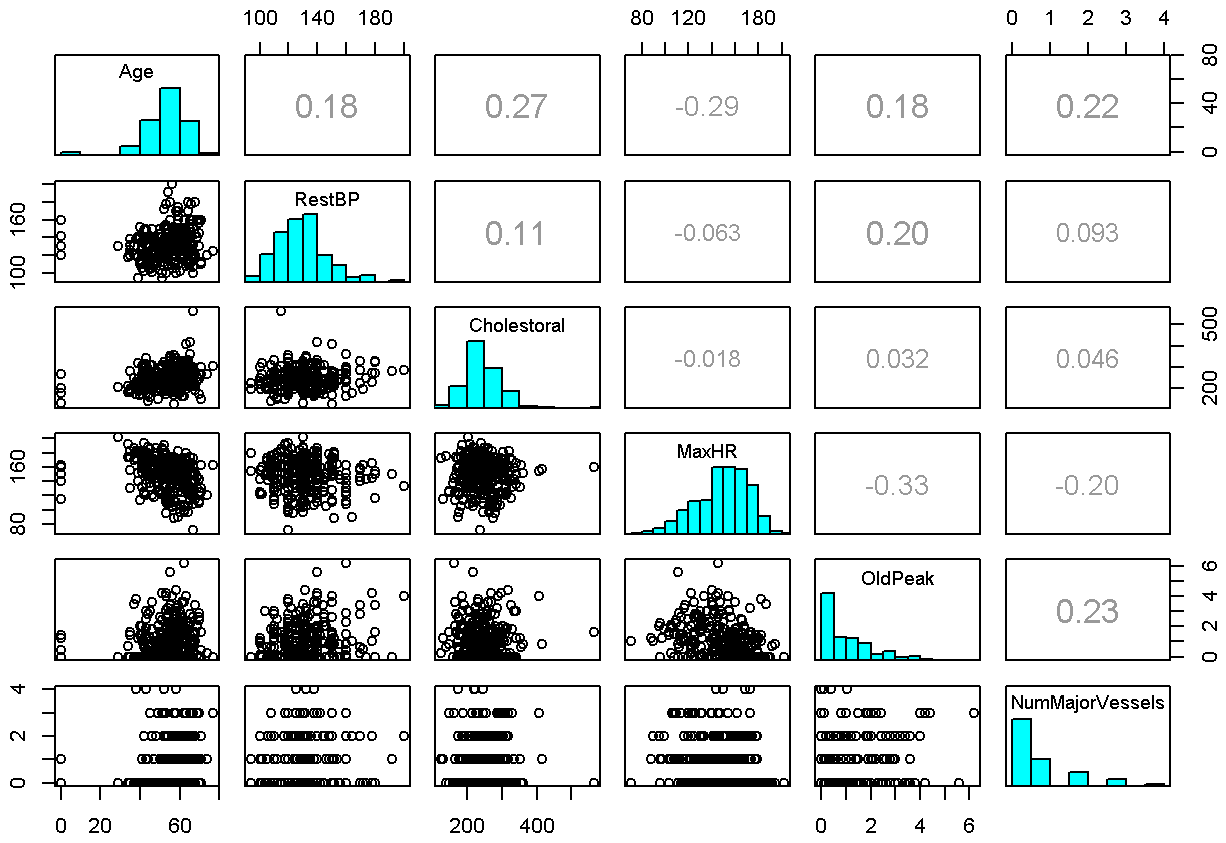
```
head(Average_person_summary)
```

```
## # A tibble: 4 x 3
##   ChestPain        meanrestBP meanChol
##   <fct>                 <dbl>    <dbl>
## 1 Typical Angina         130.     255.
## 2 Atypical Angina        129.     244.
## 3 Non-anginal Pain       132.     246.
## 4 Asymptomatic           133.     244.
```

Pairwise plot

A pairwise plot is a great way to get a lot of information from. However when working with non numerical data, you need to create a dataframe of only your numerical data like I do below. Another thing that helps is making sure that the data is normal. I did that by doing a rank-z transform to resting bloop pressure, cholestoral and maximum heart rate.

```
#Now I can do a little analysis of numerical data
numerics<- heart[, c(1, 4,5,8, 10, 12)]
#
#Some data transformations to make the data normal
heart$Cholestoral <-rz.transform(heart$Cholestoral)
heart$RestBP <- rz.transform(heart$RestBP)
heart$MaxHR <- rz.transform(heart$MaxHR)
#
#doing a pairwise plot of all the numerics to see general correlations.
pairs(numerics, upper.panel = panel.cor, diag.panel = panel.hist)
```



Resources

“Angina Pectoris (Stable Angina).” Www.heart.org, 31 July 2015, www.heart.org/en/health-topics/heart-attack/angina-chest-pain/angina-pectoris-stable-angina.