### Installing and loading libraries

```
install.packages("GGally")

install.packages("DataExplorer")

library(readr)

library(GGally)

library(DataExplorer)

library(DMwR)

library(corrplot)

library(NbClust)
```

### Setting up the working directory

```
setwd("D:/great learning/Capstone/6. Coronory Heart Risk Study")

mydata = read.csv("Coronary_heart_risk_study.csv", header = TRUE)

attach(mydata)

str(mydata)

summary(mydata)

head(mydata)
```

### Univariate Analysis

```
hist(male)

hist(age, col = "Red")

hist(education, col = "Blue")

hist(cigsPerDay, col = "Purple")

hist(totChol, col = "Orange")

hist(sysBP, col = "Pink")

hist(diaBP, col = "Green")
hist(BMI, col = "Brown")

hist(heartRate, col = "Yellow")

hist(glucose, col = "Violet")
```

### Bivariate Analysis

```
ggpairs(mydata, mapping = NULL, columns = 1:ncol(mydata))
```

### Boxplot to check outliers

```
boxplot(male, col = "Blue", main = "Boxplot for sex")

boxplot(age, col = "Yellow", main = "Boxplot for age")

boxplot(education, col = "Purple", main = "Boxplot for education")

boxplot(currentSmoker, col = "Orange", main = "Boxplot for Current Smoker")

boxplot(cigsPerDay, col = "Pink", main = "Boxplot for Cigaretes Per Day")

boxplot(BPMeds, col = "Brown", main = "Boxplot for Blood Pressure Medication")

boxplot(prevalentStroke, col = "Violet", main = "Boxplot for prevalent stroke")

boxplot(prevalentHyp, col = "Green", main = "Boxplot for prevalent hypertensive")

boxplot(diabetes, col = "Grey", main = "Boxplot for diabetes")

boxplot(totChol, col = "Red", main = "Boxplot for total cholesterol")

boxplot(sysBP, col = "Yellow", main = "Boxplot for systolic blood pressure")

boxplot(diaBP, col = "Brown", main = "Boxplot for diastolic blood pressure")

boxplot(BMI, col = "Violet", main = "Boxplot for Body Mass Index")

boxplot(heartRate, col = "Orange", main = "Boxplot for Heart Rate")

boxplot(glucose, col = "Green", main = "Boxplot for glucose")
```

### Missing Value treatement/Imputation and plot of missing value

```
colSums(is.na(mydata))
```

```
sum(is.na(mydata))
plot_missing(mydata)
mydata$education[is.na(mydata$education)] = mean(mydata$education, na.rm = T)
mydata$cigsPerDay[is.na(mydata$cigsPerDay)] = median(mydata$cigsPerDay, na.rm = T)
mydata$BPMeds[is.na(mydata$BPMeds)] = median(mydata$BPMeds, na.rm = T)
mydata$totChol[is.na(mydata$totChol)] = median(mydata$totChol, na.rm = T)
mydata$BMI[is.na(mydata$BMI)] = median(mydata$BMI, na.rm = T)
mydata$heartRate[is.na(mydata$heartRate)] = median(mydata$heartRate, na.rm = T)
mydata$glucose[is.na(mydata$glucose)] = median(mydata$glucose, na.rm = T)
plot_missing(mydata)
```

### Outlier Treatement

```
a = mydata$BMI
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
BMI = a
boxplot(BMI, horizontal = TRUE, main = "Boxplot for BMI after outlier treatement")
```

```
a = mydata$totChol
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
```

```
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
totChol = a
boxplot(totChol, horizontal = TRUE, main = "Boxplot for totChol after outlier treatement")

a = mydata$sysBP
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
sysBP = a
boxplot(sysBP, horizontal = TRUE, main = "Boxplot for sysBP after outlier treatement")

a = mydata$diaBP
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
diaBP = a
boxplot(diaBP, horizontal = TRUE, main = "Boxplot for diaBP after outlier treatement")

a = mydata$heartRate
```

```
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
heartRate = a
boxplot(heartRate, horizontal = TRUE, main = "Boxplot for heartRate after outlier treatement")

a = mydata$glucose
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
glucose = a
boxplot(glucose, horizontal = TRUE, main = "Boxplot for glucose after outlier treatement")

a = mydata$cigsPerDay
qnt = quantile(a, probs=c(0.25,0.75), na.rm=T)
caps = quantile(a, probs=c(0.05,0.95), na.rm=T)
h = 1.5*IQR(a, na.rm=T)
a[a<(qnt[1]-h)] = caps[1]
a[a>(qnt[2]+h)] = caps[2]
print(a)
boxplot(a)
```

```
cigsPerDay = a

boxplot(cigsPerDay, horizontal = TRUE, main = "Boxplot for cigsPerDay after outlier treatement")
```

### Correlation plot after missing value treatement

```
mydata.corr = mydata[,c(2,3,5,10,11,12,13,14,15)]

mydata.corr2 = cor(mydata.corr)

corrplot(mydata.corr2, method = "number")
```

### Scale the data

```
mydata.scaled = scale(mydata[ ,c(2,3,5,10,11,12,13,14,15)])

colSums(is.na(mydata))

print(mydata.scaled, digits = 3)

apply(mydata.scaled,2,mean)

apply(mydata.scaled,2,sd)

View(mydata.scaled)
```

### Imbalance Analysis (Smote Analysis)

```
library(caTools)

table(TenYearCHD)

split = sample.split(mydata$TenYearCHD, SplitRatio = 0.7)

smote.train = subset(mydata, split == TRUE)

smote.test = subset(mydata, split == FALSE)

mydata$TenYearCHD = as.factor(mydata$TenYearCHD)

mydata.smote = SMOTE(TenYearCHD~.,mydata, perc.over = 3000, k = 4, perc.under = 250)

table(mydata.smote$TenYearCHD)
```

### Clustering

```
mydata2 = mydata[,c(2,3,5,10,11,12,13,14,15)]
seed = 1000
set.seed(seed)
nc = NbClust(mydata2, min.nc = 2, max.nc = 7, method = "kmeans")

library(cluster)
seed = 1000
set.seed(seed)
clust = kmeans(x=mydata.scaled, centers = 2, nstart = 5)
print(clust)
clusplot(mydata.scaled, clust$cluster, color = TRUE, shade = TRUE, label = 2, lines = 1)

mydata.scaled$cluster = clust$cluster
View(mydata)
TenYearCHD.profile = aggregate(mydata.scaled,list(mydata.scaled$cluster), FUN = "mean")
print(TenYearCHD.profile)
```