Exploratory Data Analysis Project

AIM: 6 Different methods for Outlier Detection in MUSK Datasets having 554 rows & 170 columns using R.

CODE:

#OUTLIER USING PCA

```
library(caret)
library(utils)
data=clean1[,3:169]
data
classi<-clean1[,169]
classi
par(mfrow=c(1,2))
View(data)
c1=cor(data)
c1
meanx8=mean(data$X8)
meanx23=mean(data$X23)
sdx8=sd(data$X8)
sdx23=sd(data$X23)
x <- rnorm(475,meanx8,sdx8)
y <- rnorm(475,meanx23,sdx23)
plot(x,y)
for (i in 1:nrow(c1)){
  correlations <-  which((c1[i,] > 0.98) & (c1[i,] != 1))

  if(length(correlations)> 0){
    print(colnames(c1)[i])
    print(correlations)
  }
}
data_bind <- cbind(data$X8,data$X23)
data_bind1 <- cbind(data$X40,data$X76,data$X6)

bp <- function(X,fac){
  med <-sapply(X,median)
  q25 <-sapply(X,function(x)quantile(x,prob=0.25))
  q75 <-sapply(X,function(x)quantile(x,prob=0.75))
  erg <- t(apply(X, 1, function(x) abs(med-x)-fac*(q75-q25)))
  return(as.vector(which(rowSums(erg>0)>0)))}

bp_app <-function(X,a){
  outliers <- rep(1,length(X[,1]))
  outliers[bp(X,a)] <- 2
  outliers<- as.factor(outliers)
  levels(outliers) <- c("No Outlier","Outlier")
  print(table(outliers))
  if(table(outliers)[2] > 0)plot(X,col=outliers,pch=18)
```

```
  return(outliers)}


dat_std <- apply(dat,2,function(x)x/(max(x)-min(x)))

par(mfcol=c(2,2))
print("Outlier detection using PCA with Factor 1.5")
PCA1.5_1 <-bp_app(as.data.frame(princomp(data_bind)$scores)[,1:2],1.5)
PCA1.5_1
plot(data_bind,col=PCA1.5_1,pch=18)
print("Outlier detection using PCA with Factor 3")
PCA3_1 <-bp_app(as.data.frame(princomp(data_bind)$scores)[,1:2],3)
plot(data_bind,col=PCA3_1,pch=18)


print("Outlier detection using PCA with Factor 1.5")
PCA1.5_2 <-bp_app(as.data.frame(princomp(data_bind1)$scores)[,1:2],1.5)
PCA1.5_2
plot(data_bind,col=PCA1.5_2,pch=18)
print("Outlier detection using PCA with Factor 3")
PCA3_2 <-bp_app(as.data.frame(princomp(data_bind1)$scores)[,1:2],3)
plot(data_bind,col=PCA3_2,pch=18)


#OUTLIER USING COOKS DISTANCE

library(caret)
library(utils)
data=clean1[,3:169]
data
c1=cor(data)
c1
for (i in 1:nrow(c1)){
  correlations <-  which((c1[i,] > 0.98) & (c1[i,] != 1))

  if(length(correlations)> 0){
    print(colnames(c1)[i])
    print(correlations)
  }
}

par(mfrow=c(4,2))
data_bind23<-cbind(data$X8,data$X11.1,data$X6)
mod23 <- lm(data$X23 ~ data_bind23, data=data)
cooksd23 <- cooks.distance(mod23)
plot(cooksd23, pch="*", cex=1.2, main="")
abline(h = 4*mean(cooksd23, na.rm=T), col="red")
text(x=1:length(cooksd23)+1, y=cooksd23, labels=ifelse(cooksd23>4*mean(cooksd23,
na.rm=T),names(cooksd23),""), col="red")
influential <- as.numeric(names(cooksd23)[(cooksd23 > 4*mean(cooksd23, na.rm=T))])  #
influential row numbers
```

```
data_bind.28<-cbind(data$X.32)
mod.28 <- lm(data$X.28 ~ data_bind.28, data=data)
cooksd.28 <- cooks.distance(mod.28)
plot(cooksd.28, pch="*", cex=1.2, main="Influential Obs by Cooks distance")
abline(h = 4*mean(cooksd.28, na.rm=T), col="red")
text(x=1:length(cooksd.28)+1, y=cooksd.28, labels=ifelse(cooksd.28>4*mean(cooksd.28,
na.rm=T),names(cooksd.28),""), col="red")
influential <- as.numeric(names(cooksd.28)[(cooksd.28 > 4*mean(cooksd.28, na.rm=T))])  #
influential row numbers

data_bind63<-cbind(data$X80,data$X51)
mod63 <- lm(data$X63 ~ data_bind63, data=data)
cooksd63 <- cooks.distance(mod63)
plot(cooksd63, pch="*", cex=1.2, main="Influential Obs by Cooks distance")
abline(h = 4*mean(cooksd63, na.rm=T), col="red")
text(x=1:length(cooksd63)+1, y=cooksd63, labels=ifelse(cooksd63>4*mean(cooksd63,
na.rm=T),names(cooksd63),""), col="red")
influential <- as.numeric(names(cooksd63)[(cooksd63 > 4*mean(cooksd63, na.rm=T))])  #
influential row numbers

data_bind.177<-cbind(data$X.146,data$X.206)
mod.177 <- lm(data$X.177 ~ data_bind.177, data=data)
cooksd.177 <- cooks.distance(mod.177)
plot(cooksd.177, pch="*", cex=1.2, main="")
abline(h = 4*mean(cooksd63, na.rm=T), col="red")
text(x=1:length(cooksd.177)+1, y=cooksd.177, labels=ifelse(cooksd.177>4*mean(cooksd.177,
na.rm=T),names(cooksd63),""), col="red")
influential <- as.numeric(names(cooksd.177)[(cooksd.177 > 4*mean(cooksd.177, na.rm=T))])

data_bind32<-cbind(data$X.28)
mod32 <- lm(data$X32 ~ data_bind32, data=data)
cooksd32 <- cooks.distance(mod32)
plot(cooksd32, pch="*", cex=1.2, main="Influential Obs by Cooks distance")
abline(h = 4*mean(cooksd32, na.rm=T), col="red")
text(x=1:length(cooksd32)+1, y=cooksd32, labels=ifelse(cooksd32>4*mean(cooksd32,
na.rm=T),names(cooksd63),""), col="red")
influential <- as.numeric(names(cooksd32)[(cooksd32 > 4*mean(cooksd32, na.rm=T))])

data_bind40<-cbind(data$X76,data$X6)
mod40 <- lm(data$X40 ~ data_bind40, data=data)
cooksd40 <- cooks.distance(mod40)
plot(cooksd40, pch="*", cex=1.2, main="Influential Obs by Cooks distance")
abline(h = 4*mean(cooksd40, na.rm=T), col="red")
text(x=1:length(cooksd40)+1, y=cooksd40, labels=ifelse(cooksd40>4*mean(cooksd40,
na.rm=T),names(cooksd40),""), col="red")
influential <- as.numeric(names(cooksd40)[(cooksd40 > 4*mean(cooksd40, na.rm=T))])

data_bind6<-cbind(data$X23,data$X40)
mod6 <- lm(data$X6 ~ data_bind6, data=data)
cooksd6 <- cooks.distance(mod6)
plot(cooksd6, pch="*", cex=1.2, main="Influential Obs by Cooks distance")
abline(h = 4*mean(cooksd6, na.rm=T), col="red")
```

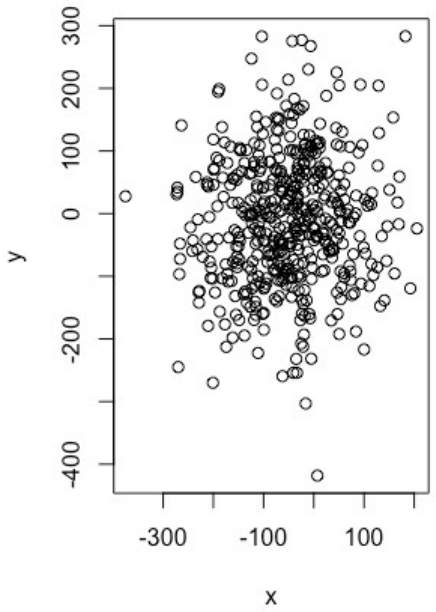
```
text(x=1:length(cooksd6)+1, y=cooksd6, labels=ifelse(cooksd6>4*mean(cooksd6,
na.rm=T),names(cooksd6),""), col="red")
influential <- as.numeric(names(cooksd6)[(cooksd6 > 4*mean(cooksd6, na.rm=T))])

#OUTLIER USING SCATTER PLOT

library(caret)
library(utils)
library(car)
data2=clean1[,3:169]
classi<-clean1[,169]
data8=data$X8
data23=data$X23
data40=data$X40
data76=cbind(data$X76,data$X6)

scatterplot(data8 ~ data23 | classi , data=data2,main="",xlab="",ylab="",col = "red")
scatterplotMatrix(~ data8 + data23 + data40 | classi,data=data2, main="",legend.pos="bottomright")

scatterplot(data40 ~ data$X76 | classi , data=data2,main="",xlab="",ylab="",col = "red")
scatterplotMatrix(~ data40 + data$X76 + data$X6 | classi,data=data2,
main="",legend.pos="bottomright")

#OUTLIER USING DBSCAN METHOD

library("dbscan")
library(caret)
library(utils)
data=clean1[,3:169]
meanx8=mean(data$X8)
meanx23=mean(data$X23)
sdx8=sd(data$X8)
sdx23=sd(data$X23)
dat <- data.frame(data$X8,data$X23)

par(mfcol=c(2,2))
data_std <- apply(dat,2,function(x)x/(max(x)-min(x)))

kNNdistplot(data_std, k =  10)
abline(h=0.1,lwd=2)
dbcl1_1<-dbscan(data_std,eps=0.1,minPts=10)
plot(as.data.frame(dat),pch=18,col=ifelse(dbcl1_1$cluster==0,2,1))

kNNdistplot(data_std, k =  20)
abline(h=0.05,lwd=2)
dbcl1_2 <- dbscan(data_std,eps=0.05,minPts=10)
plot(as.data.frame(dat),pch=18,col=ifelse(dbcl1_2$cluster==0,2,1))


#OUTLIER USING KMEANS METHOD
```

```
library(caret)
library(utils)
data=clean1[,3:169]
#data
classi<-clean1[,169]
#classi
par(mfrow=c(1,2))
#View(data)
c1=cor(data)
c1
meanx8=mean(data$X8)
meanx23=mean(data$X23)
sdx8=sd(data$X8)
sdx23=sd(data$X23)
x <- rnorm(475,meanx8,sdx8)
y <- rnorm(475,meanx23,sdx23)
plot(x,y)
findCorrelation(c1,cutoff = 0.99,names = TRUE)
dat <- data.frame(data$X8,data$X23)
#typeof(data_bind)
#typeof(dat)
par(mfcol= c(3,1))
dat_std <- apply(dat,2,function(x)x/(max(x)-min(x)))
cl1_2 <- kmeans(dat_std,200)
ind <- as.vector(which(table(cl1_2$cluster)< 5))
out <- ifelse(cl1_2$cluster %in% ind,1,2)
plot(as.data.frame(dat),pch=18,col=out, main="B1) MUSK:=200 , minpoints=5")

cl3_2 <- kmeans(dat_std,300)
ind <- as.vector(which(table(cl3_2$cluster)< 5))
out <- ifelse(cl3_2$cluster %in% ind,1,2)
plot(as.data.frame(dat),pch=18,col=out, main="B2) MUSK:=300 , minpoints=5")

cl2_2 <- kmeans(dat_std,200)
ind <- as.vector(which(table(cl2_2$cluster)< 10))
out <- ifelse(cl2_2$cluster %in% ind,1,2)
plot(as.data.frame(dat),pch=18,col=out, main="B3) MUSK:=200 , minpoints=10")

#OUTLIER USING EUCLIDIAN DISTANCE

library(caret)
library(utils)
data=clean1[,3:169]
par(mfrow=c(1,2))
View(data)
c1=cor(data)
c1
findCorrelation(c1,cutoff = 0.99,names = TRUE)
c2=cor(data1)
data_bind <- cbind(data$X8,data$X23)
dat <- data.frame(data$X8,data$X23)
```

```
euclid <- function(X,fac){
  med <-sapply(X,median)
  erg <- t(apply(X, 1, function(x) (med-x)^2))
  dist <- sqrt(rowSums(erg))
  #   print(plot(dist))
  return(dist > fac*median(dist))}

euclid_app <-function(X,a){
  outliers <- rep(1,length(X[,1]))
  outliers[euclid(X,a)] <- 2
  outliers<- as.factor(outliers)
  levels(outliers) <- c("No Outlier","Outlier")
  print(table(outliers))
  if(table(outliers)[2] > 0)plot(X,col=outliers,pch=18)
  #return(outliers)
}
### musk dataset ###
par(mfrow=c(1,2))
dat_std <- apply(dat,2,function(x)x/(max(x)-min(x)))
print("Euclid methode on musk data")
euclid_app(as.data.frame(dat_std),5)
euclid_app(as.data.frame(dat_std),3)
```

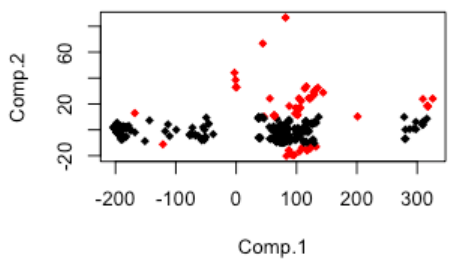
OUTPUT:

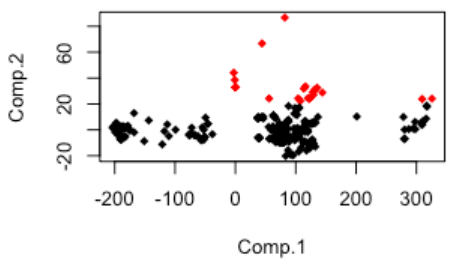
```
#OUTLIER USING PCA
x <- rnorm(475,meanx8,sdx8)
y <- rnorm(475,meanx23,sdx23)
```

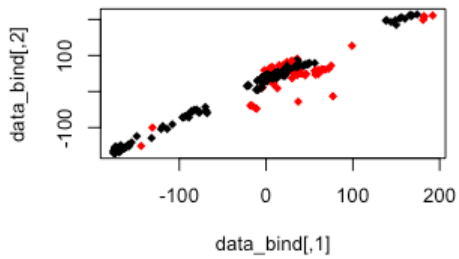


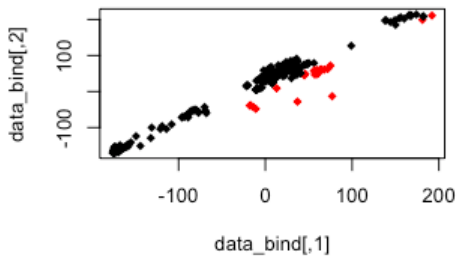
outliers
No Outlier Outlier
424 51

outliers
No Outlier Outlier
454 21



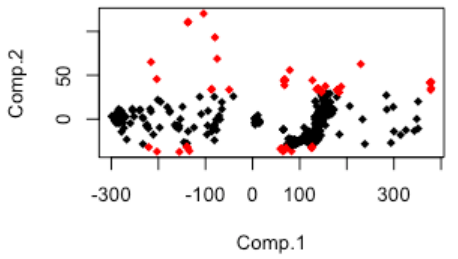


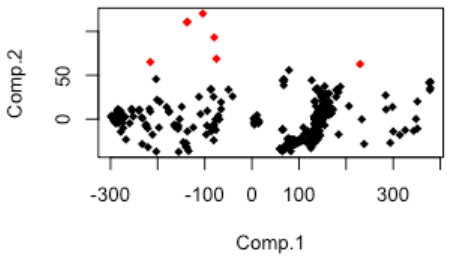


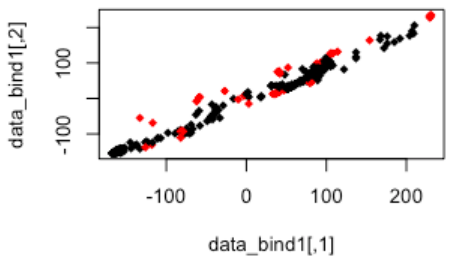


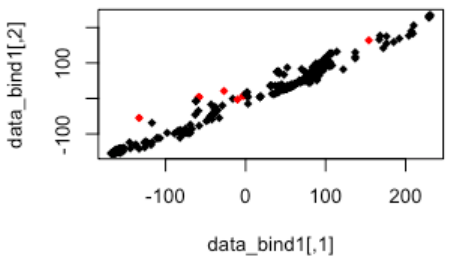
outliers
No Outlier Outlier
425 50

outliers
No Outlier Outlier
468 7

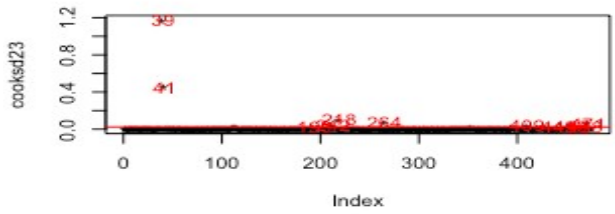


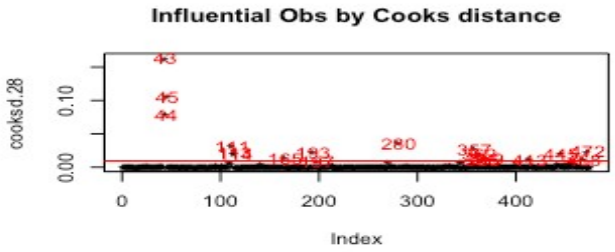


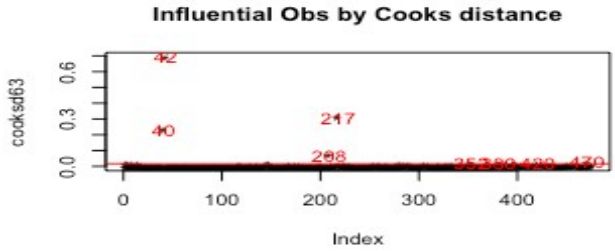


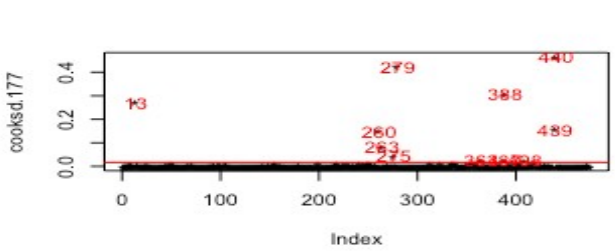


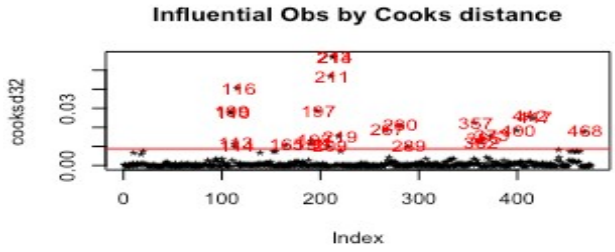
#OUTLIER USING COOKS DISTANCE

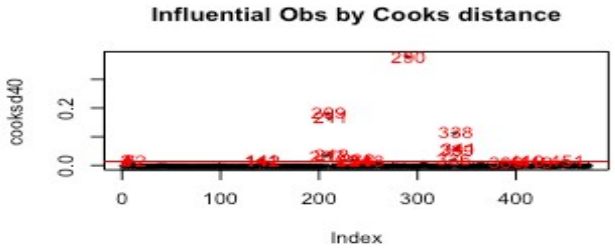


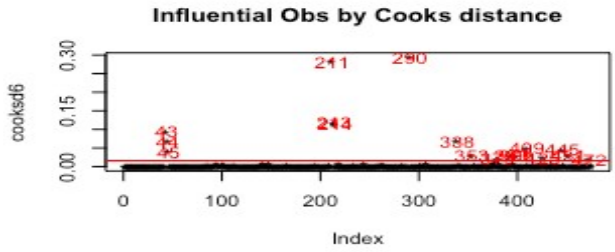




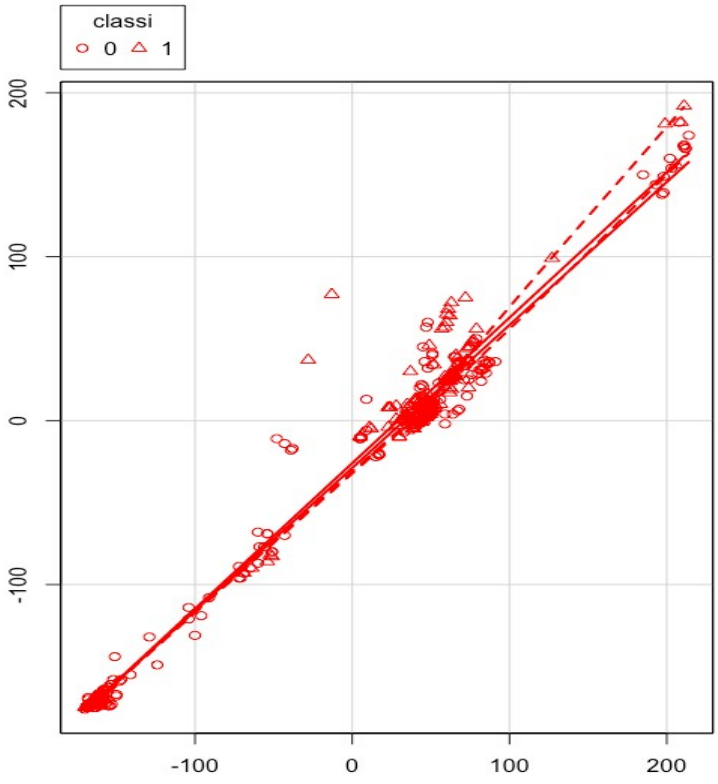


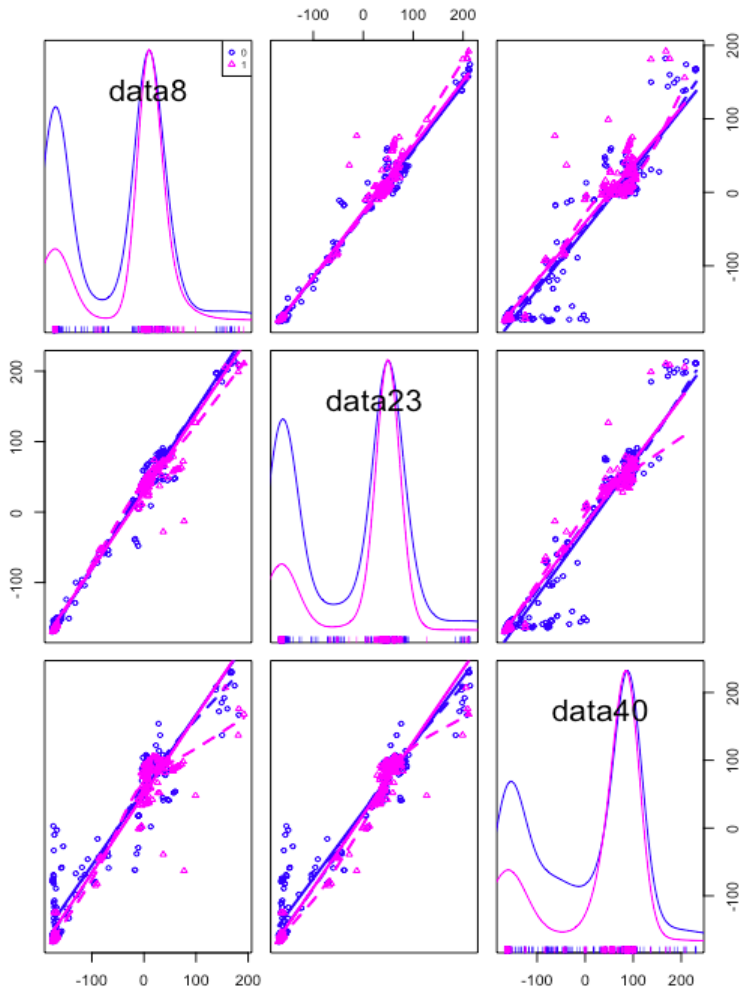


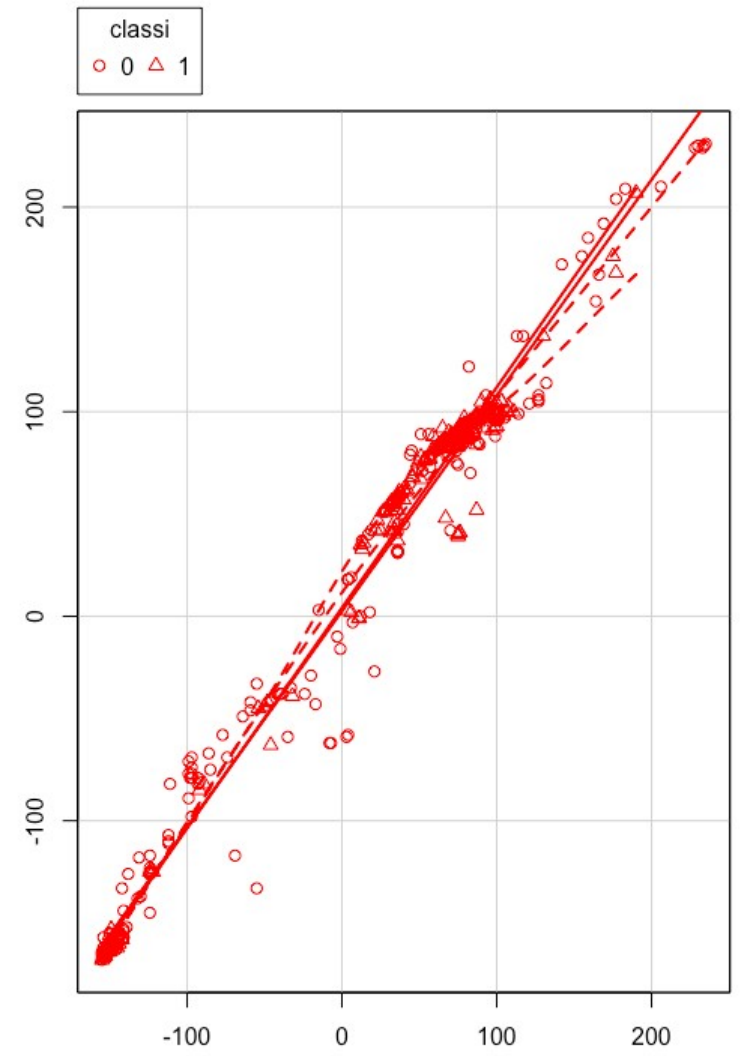


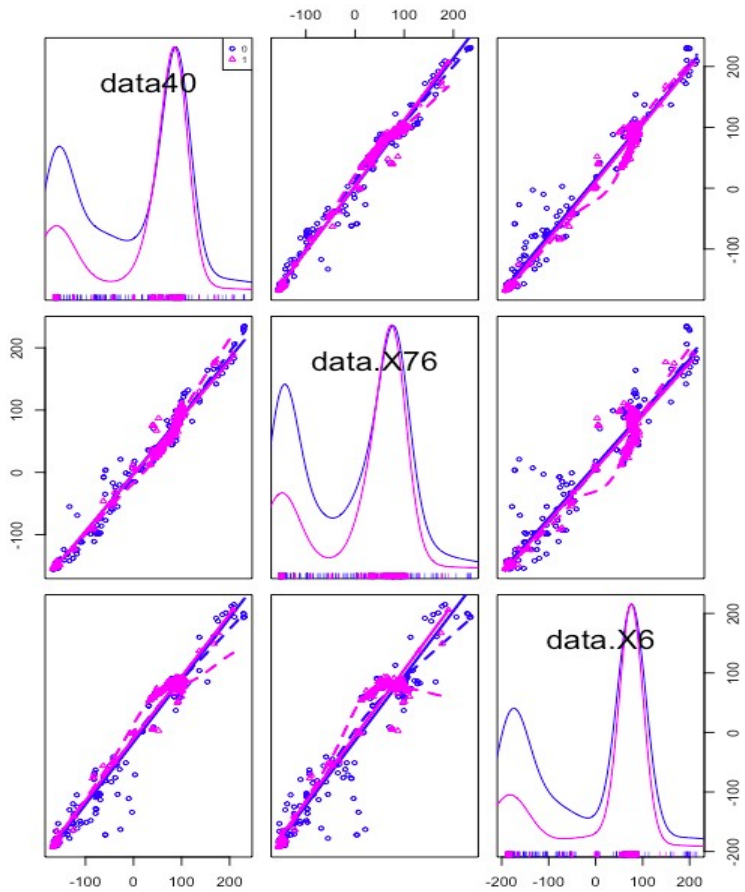


#OUTLIER USING SCATTER PLOT

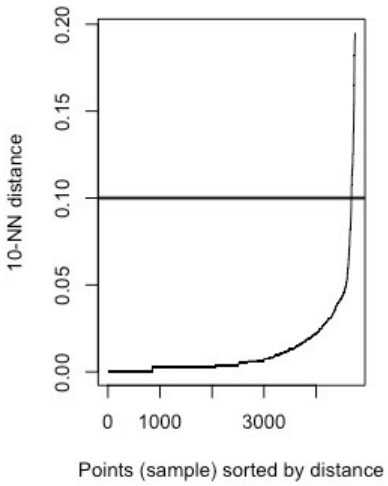


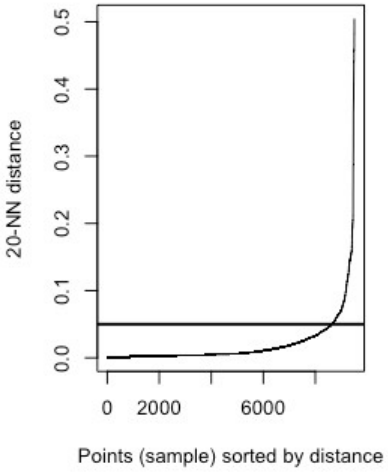


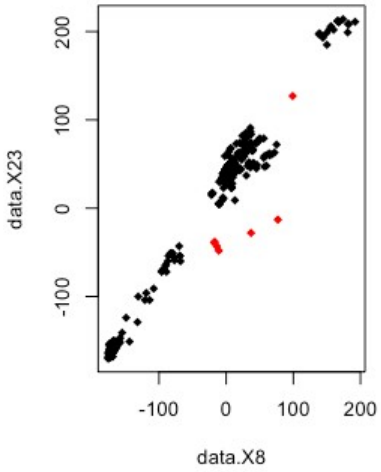

classi
0
1
200
100
0
-100
-100
0
100
200

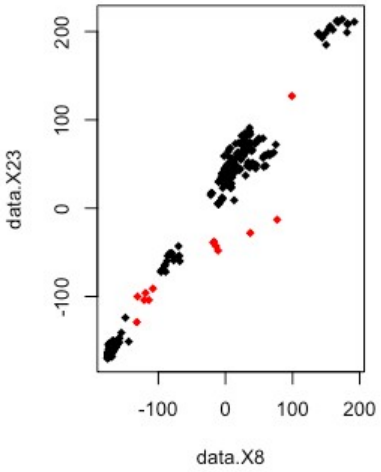

data40
data.X76
data.X6
-100
0
100
200
-200

#OUTLIER USING DBSCAN METHOD

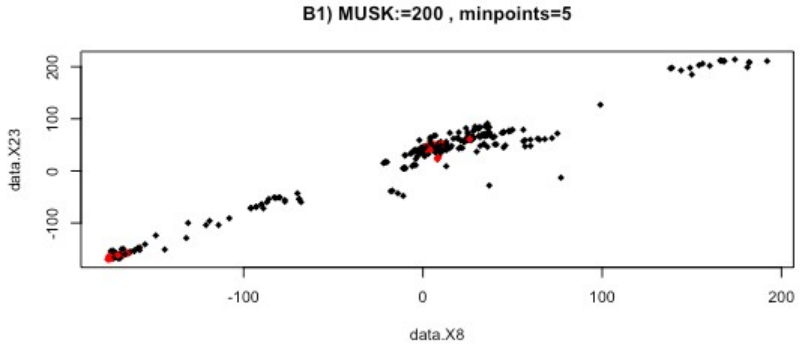


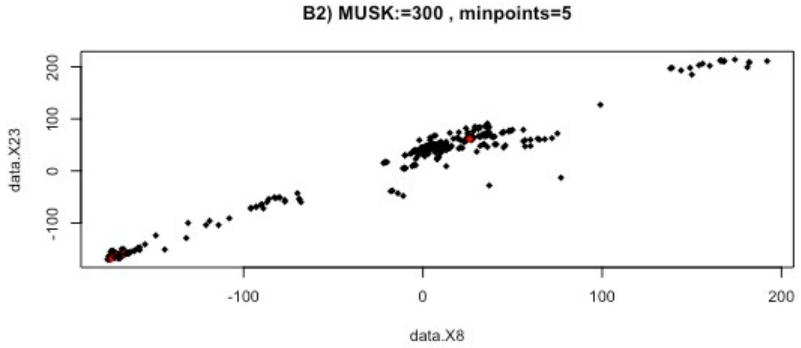


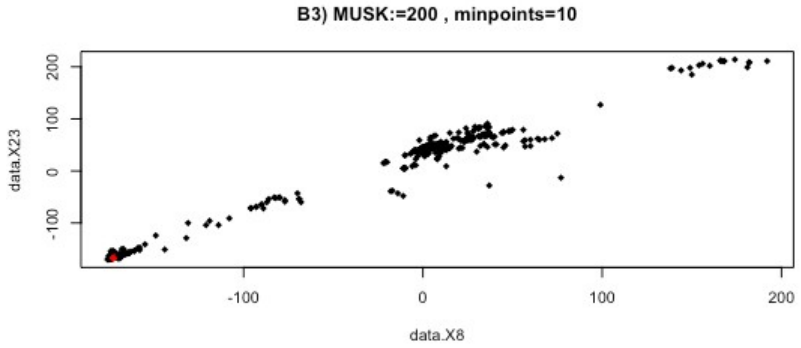




#OUTLIER USING KMEANS



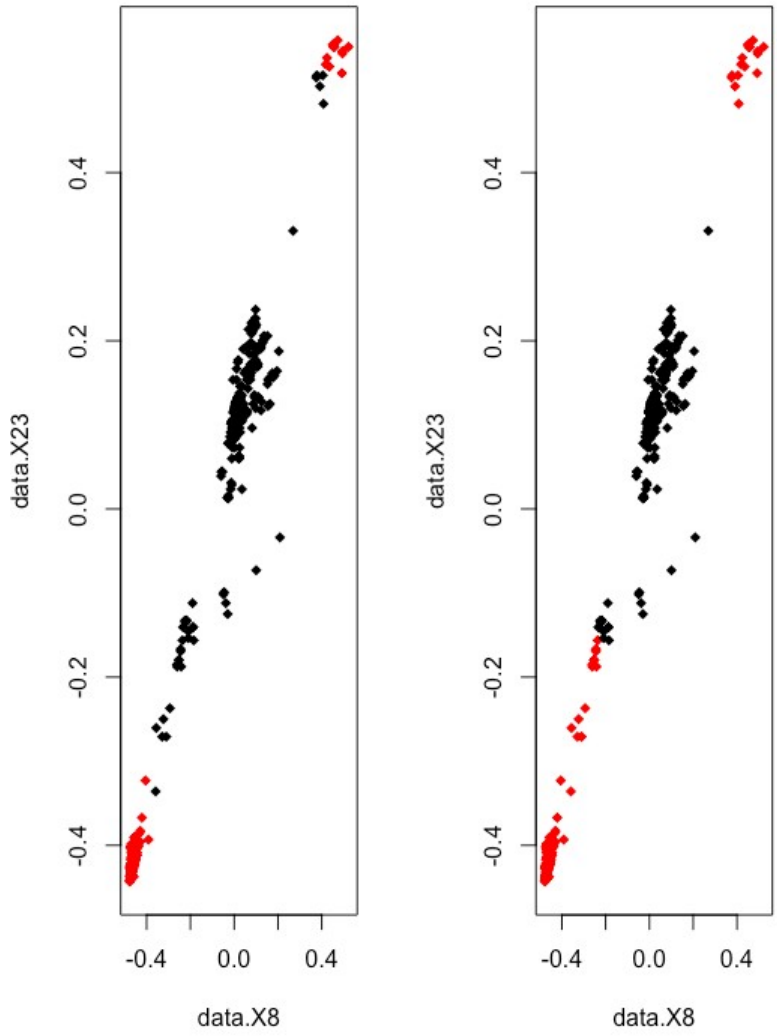




#OUTLIER USING EUCLIDEAN DISTANCE

outliers
No Outlier Outlier
316 159

outliers
No Outlier Outlier
297 178



INFERENCE: Outlier is calculated for MUSK Dataset using different method and red dot area outlier as mentioned below :
1.PCA: 88 percentage of variance is covered and outlier is detected in same direction between highly correlated. No more outlier were detected using factor 3.
2.Cook's Distance: The points having distance greater than 3 times of mean are considered as outlier's , denoted in red color.
3.Scatter Plot: 0 represent non-musk and 1 represent musk respectively in the scatterplot and we can identify outlier's.
4.DBScan: eps value is et to point 0.05 and minimum point is set to 10 and 20.

5.K-Means Method: since the number of point in cluster is not fixed for specific dataset. So, we tried with 200 and 300 point.
6.Euclidean distance :I estimate the euclid distances of every observation to the vector of medians. Every observations whose distance is larger than 3 times the median of the distances is an outlier