Pramod Aravind Byakod, 113436877, IDA HW 2

##1

```
##asbio package
library("asbio")
x = c(3,4,2,1,7,6,5)
y = c(4,3,7,6,5,2,1)
condis = ConDis.matrix(x,y)
concord = sum(condis == 1, na.rm = T) # no. of concordant pairs
discord = sum(condis == -1, na.rm = T) # no. of discordance pairs
answer = c("concord" = concord, "discord" = discord)
```

#concord discord

 # 6 15

##2

#Final Animal selected in the very last step of outliers example is

#body brain

#Human 62 1320

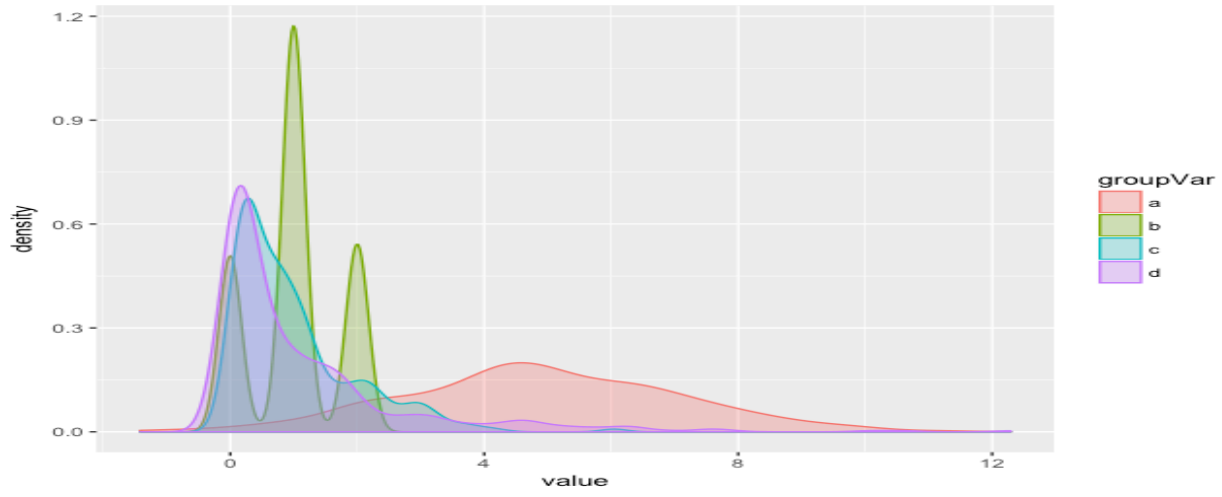
##3

##3.a

```
#Creating different distributions
a = rnorm(500, 5, 2), b = rbinom(500, 2, 0.5) ,c = rexp(500, 1) ,d = rchisq(500,df =1)
df = data.frame(a,b,c,d)
library(reshape2) #convert the data to long format
df2 = melt(df, measure.vars = c("a","b","c","d"))
colnames(df2) = c("groupVar","value")
library(plyr)
```

##3.b

```
library(ggplot2) #Plot the density plot across its distribution
ggplot(df2,aes(x=value,color=groupVar))+geom_density(aes(group=groupVar,fill=groupVar),alpha=0.3)
```



##4

##4.a

#The data that collected in back 1800 times is far diffrent from the data that collected in recents times as evolution of the data gathering techniques, hence shark data is collected in untidy manner, it impacts the timliness of the data.

##4.b

```
sharkattacks_data = read.csv("ISE 5103 GSAF.csv", header = T) #Loading shark data
GSAFdata = sharkattacks_data[ which(sharkattacks_data$Year >= 2000), ] #GSAFdata contains incidents occurring on or after the year 2000
```

##4.c

```
library(lubridate)
new_date = dmy(GSAFdata$Date) #Formatting date field
GSAFdata = data.frame(GSAFdata,new_date)
```

##4.d

```
missing_date_percent=(sum(is.na(GSAFdata$new_date))/length(GSAFdata$new_date))*100
```

#[1] 2.558001

##4.e

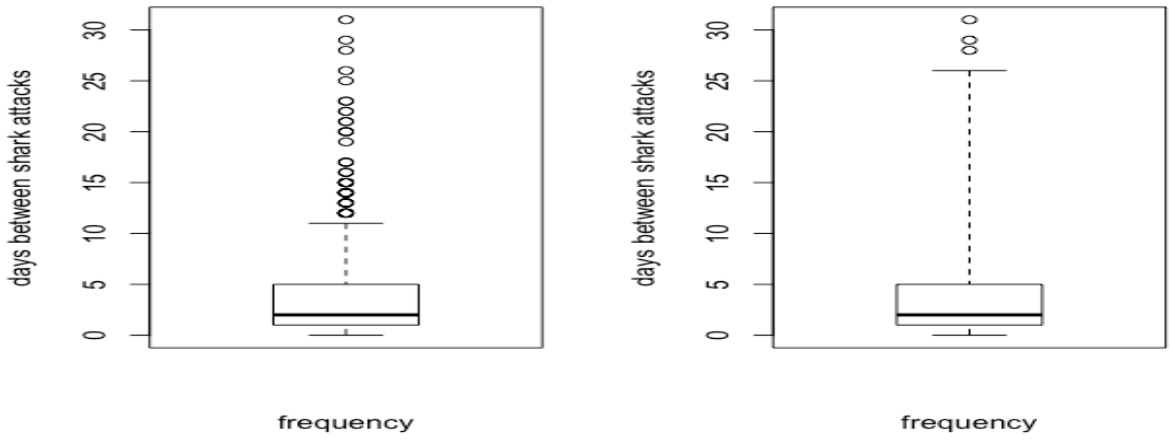
```
GSAFdata = GSAFdata[!is.na(GSAFdata$new_date),] #Delete all the rows which have "NA" new_date column entry
```

##4.f.i

```
GSAFdata = GSAFdata[order(GSAFdata$new_date,decreasing = FALSE), ] #Sort the data frame
daysBetween = diff(GSAFdata$new_date)
daysBetween = append(daysBetween, 0, 0) # placing 0 at first index
GSAFdata = data.frame(GSAFdata,daysBetween) #Add the daysBetween column
```

##4.f.ii

```
par(mfrow  = c(1,2))
# we can see lot of outliers within this plot
boxplot(GSAFdata$daysBetween,ylab = "days between shark attacks", xlab = "frequency")
adjbox(GSAFdata$daysBetween,ylab = "days between shark attacks", xlab = "frequency")
invisible(dev.off())
```



#Many outliers are there when we do plot using boxplot, but with adjplot there are few outliers

##4.f.iii

grubbs.test(GSAFdata$daysBetween,type=10)

 # Grubbs test for one outlier

#data: GSAFdata$daysBetween

#G = 7.12870, U = 0.96894, p-value = 5.547e-10

#alternative hypothesis: highest value 31 is an outlier

##4.g

par(mfrow = c(1,2))

qqnorm(GSAFdata$daysBetween,main="Days between shark attacks")

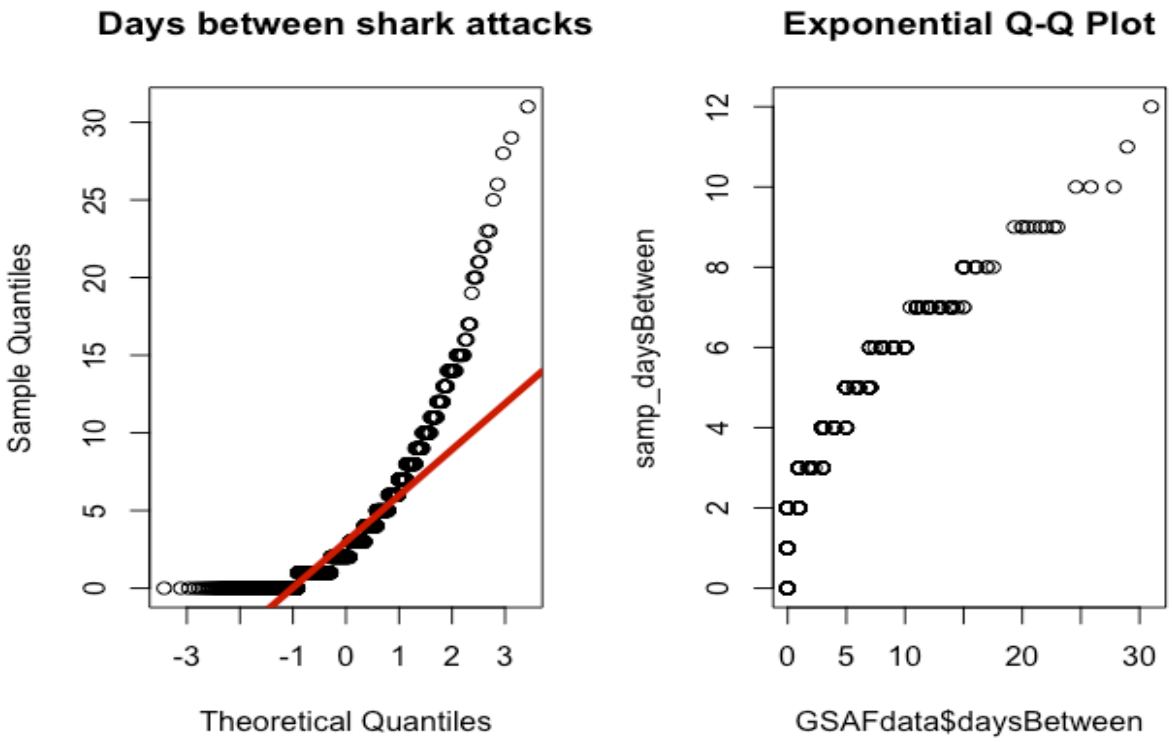
qqline(GSAFdata$daysBetween, col = 'red3', lwd = 4)

samp_daysBetween = rpois(1556, lambda=mean(GSAFdata$daysBetween)) #Creating distribution sample

#Plotting against sample data

qqplot(GSAFdata$daysBetween, samp_daysBetween, main="Exponential Q-Q Plot")

invisible(dev.off())



#Above plot clearly indicates that days are exponentially distributed

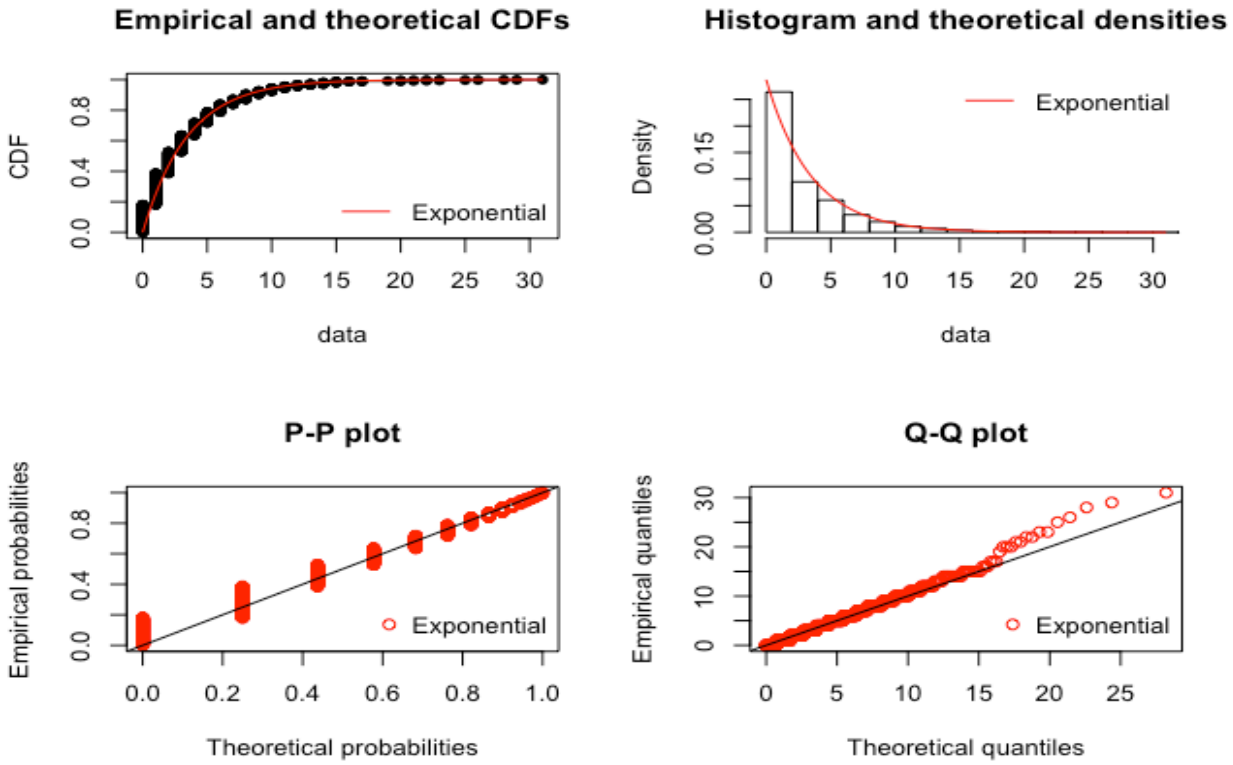
##4.h

library(fitdistrplus)

DaysBetween = GSAFdata$daysBetween

fite = fitdist(DaysBetween[2:1638],"exp")

```
par(mfrow = c(2,2))
cdfcomp(fite, legendtext = "Exponential")
denscomp(fite, legendtext = "Exponential")
ppcomp(fite, legendtext = "Exponential")
qqcomp(fite, legendtext = "Exponential")
```



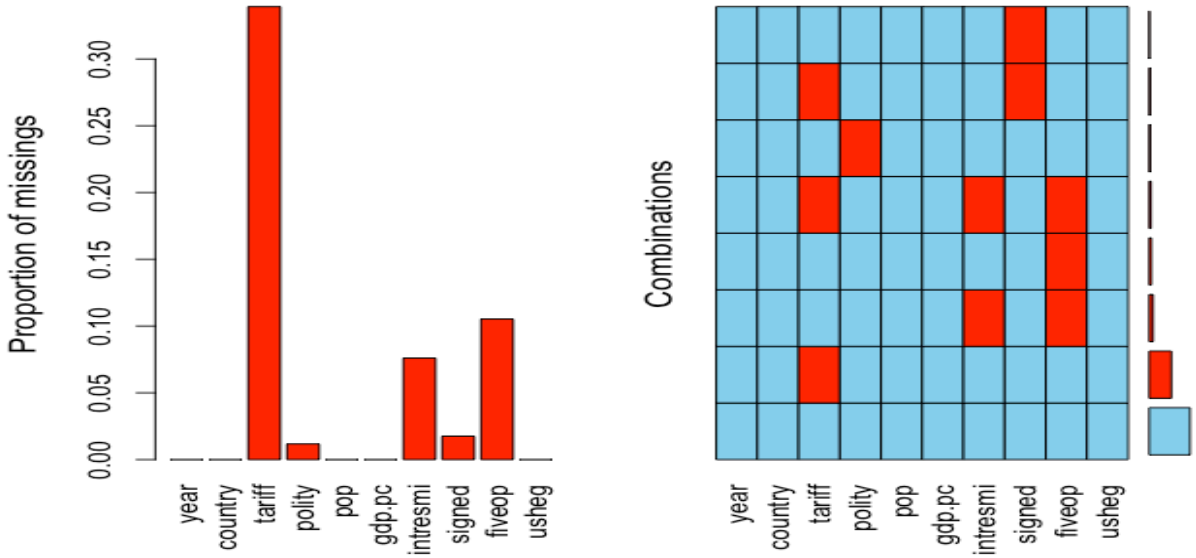

```
gofstat(fite)
```

#Goodness-of-fit statistics
 # 1-mle-exp
#Kolmogorov-Smirnov statistic 0.1808186
#Cramer-von Mises statistic 5.6463528
#Anderson-Darling statistic Inf
#Goodness-of-fit criteria
 # 1-mle-exp
#Akaike's Information Criterion 7363.516
#Bayesian Information Criterion 7368.917
##4.i
#Yes, shark attacks occur as a Poission process. There is no obvious answer but more shark attacksare happening in US, Australia, SouthAfricsThat.

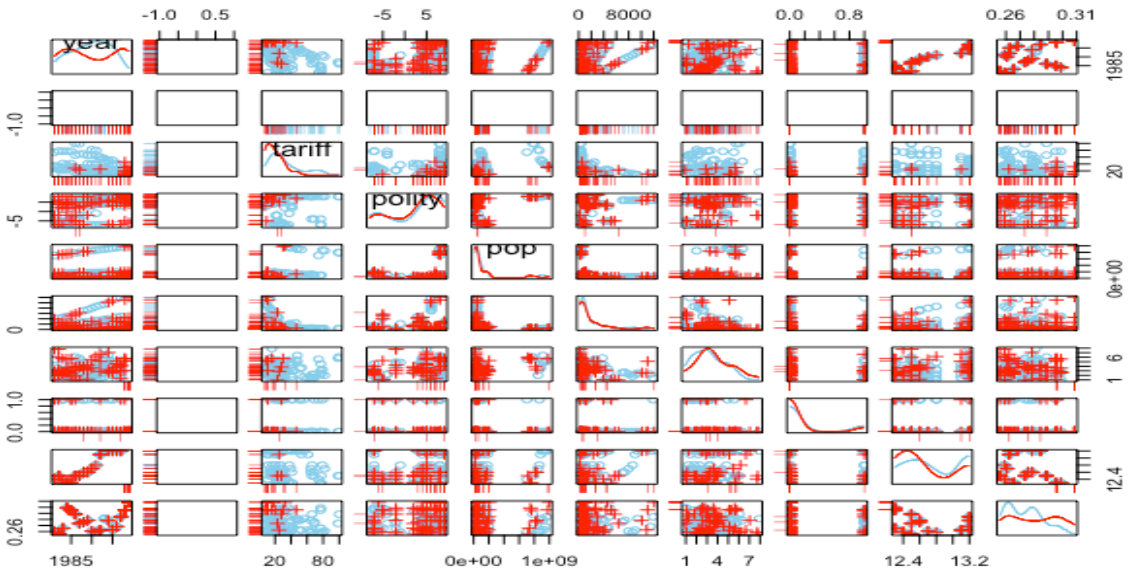
##5
##5.a

```
library(Amelia)
library(VIM)
```

data("freetrade")
aggr(freetrade,delimiter = NULL, plot = TRUE, prop = TRUE) #Missingness in freetrade using aggr



# similar to scatterplot notes the missing values
scattmatrixMiss(freetrade, selection = "any")



##5.b
#replacing all NA to 0
freetrade[is.na(freetrade)] = 0
#replacing all values greater than 0 to 1
freetrade[freetrade$tariff>0,]$tariff <- 1
#chisq.test to determine the missingness

chisq.test(freetrade$country, freetrade$tariff)

**#	Pearson's Chi-squared test**

#data: freetrade$country and freetrade$tariff

#X-squared = 23.064, df = 8, p-value = 0.003283

#from the results the p value is less than 0.05, says that missingness of tariff significantly dependent on the country values by rejecting null hypothesis

#chisq test conducted excluding Nepal

freetradeWOnepal <- freetrade[(freetrade$country!="Nepal"),]

chisq.test(freetradeWOnepal$tariff, freetradeWOnepal$country)

**#	Pearson's Chi-squared test**

#data: freetradeWOnepal$tariff and freetradeWOnepal$country

#X-squared = 15.836, df = 7, p-value = 0.02666

#p value is less than 0.05 so we reject the null hypothesis, tariff and country are dependent f we remove nepal

#chisq test conducted excluding Philippines

freetradeWOphilippines <- freetrade[(freetrade$country!="Philippines"),]

chisq.test(freetradeWOphilippines$tariff, freetradeWOphilippines$country)

**#	Pearson's Chi-squared test**

#data: freetradeWOphilippines$tariff and freetradeWOphilippines$country

#X-squared = 11.486, df = 7, p-value = 0.1188

#p value is greater than 0.05 so we failed to reject the null hypothesis, Means tariff and country are independent if we remove philippines

 # Nepal has mroe NA values unlike Philipines doesnt have any, Hence removal of philipine might effect the overall sampel size but not the no of NA values# where removal of Nepal affects both the NA count and the total sample size. This can be depicted by performing chi square test seperately

##6

##6.a.i

data(mtcars)

#Correlation matrix to know the dependencies between attributes

corMat = cor(mtcars, use = "everything")

##6.a.ii

eig_mtcars = eigen(corMat,symmetric = T)

##6.a.iii

pca_mtcars = prcomp(mtcars, scale. = T)

##6.a.iv

eig_mtcars

$vectors

**[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]**

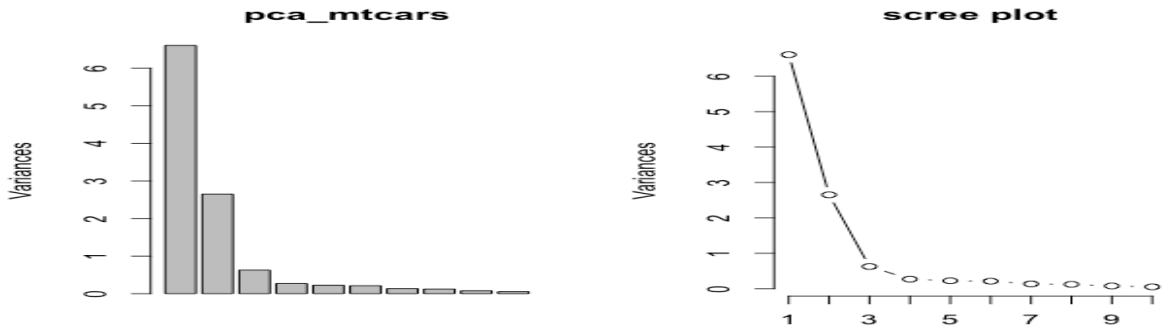
[1,] 0.3625305 -0.01612440 -0.22574419 -0.022540255 -0.10284468 -0.10879743 0.367723810

[2,] -0.3739160 -0.04374371 -0.17531118 -0.002591838 -0.05848381 0.16855369 0.057277736

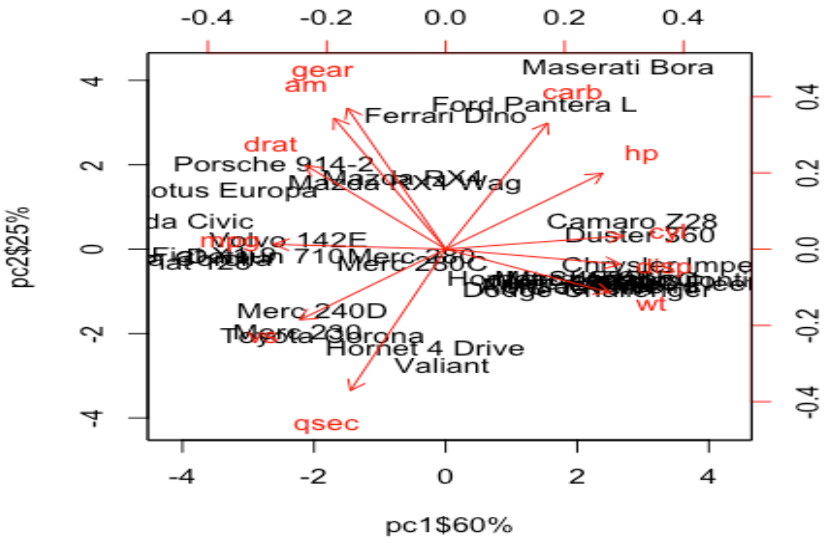
pca_mtcars

PC1 PC2 PC3 PC4 PC5 PC6 PC7
mpg -0.3625305 0.01612440 -0.22574419 -0.022540255 0.10284468 -0.10879743 0.367723810
cyl 0.3739160 0.04374371 -0.17531118 -0.002591838 0.05848381 0.16855369 0.057277736
#pca values and eigen vectors are same
##6.a.v

```
invisible(dev.off())
plot(pca_mtcars)
screeplot(pca_mtcars, type = "line", npcs = 10, main = "scree plot")
```

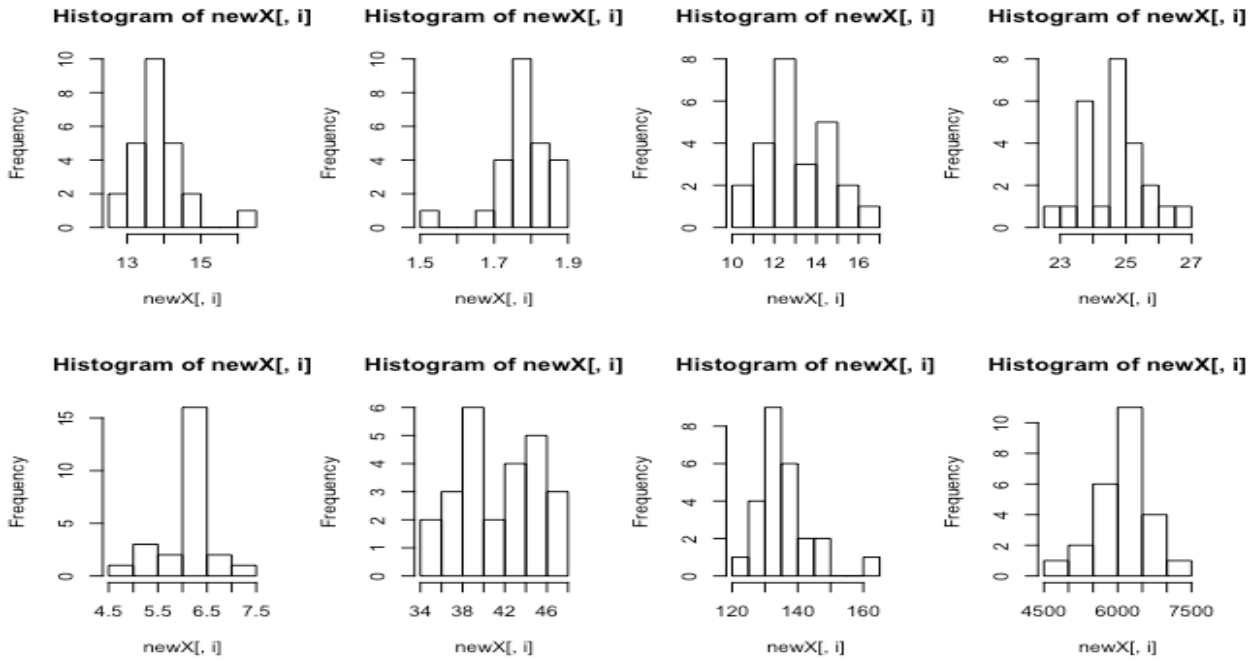



```
biplot(pca_mtcars,scale = 0, xlab = "pc1$60%", ylab = "pc2$25%")
```



#From the biplot pc1 component explains around 60% of variance of data and pc2 explains 25% of data, and if we take the maseri bora car in the plot this vehicle can be categorised as high end car with max weight and with highest horspower in terms of capacity
##6.b.i

```
library(HSAUR2)
data("heptathlon")
par(mfrow = c(2,4))
invisible(apply(heptathlon[,1:8],2,hist))
```



#Seems distribution is normal in three cases with little right skewness to the right and left
##6.b.ii

```
#apply(heptathlon,2, grubbs.test)
invisible(dev.off())
library("outliers")
grubbs.test(heptathlon$hurdles) #Grubbs test on hurdles
heptathlon[heptathlon$hurdles == "16.42", ] #To find the person as an outlier
grubbs.test(heptathlon$highjump) #Grubbs test on highjump
heptathlon[heptathlon$highjump == "1.5", ] #To find the person as an outlier
grubbs.test(heptathlon$shot)  #Grubbs test on shot
heptathlon[heptathlon$shot == "10", ] #To find the person as an outlier
grubbs.test(heptathlon$run200m)  #Grubbs test on run200
heptathlon[heptathlon$run200m == "22.56", ] #To find the person as an outlier
grubbs.test(heptathlon$run800m) #Grubbs test on run800
heptathlon[heptathlon$run800m == "163.43", ] #To find the person as an outlier
grubbs.test(heptathlon$longjump) #Grubbs test on longjump
heptathlon[heptathlon$longjump == "4.88", ]  #To find the person as an outlier
```

#We can see from the above tests that Launa is the competitor who is an outlier

```
heptathlon = heptathlon[(heptathlon$hurdles != 16.42), ] #Remove Launa
```

##6.b.iii

```
hurdles_max = max(heptathlon$hurdles)
r200_max = max(heptathlon$run200m)
r800_max = max(heptathlon$run800m)
#Transforming data
heptathlon$hurdles = hurdles_max-heptathlon$hurdles
heptathlon$run200m = r200_max-heptathlon$run200m
```

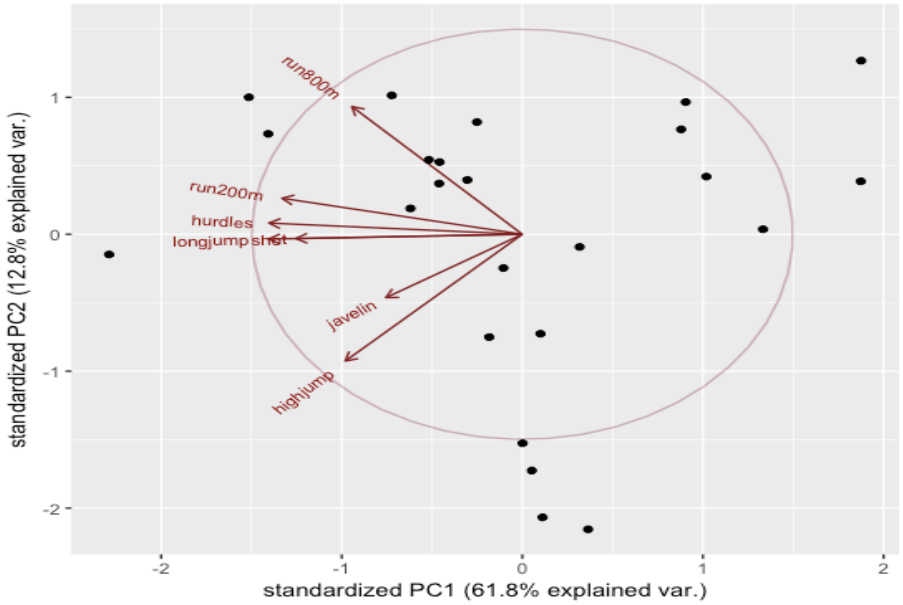
heptathlon$run800m = r800_max-heptathlon$run800m

#6.b.iv

Hpca = prcomp(heptathlon[,-8], scale. = T)

#6.b.v

ggbiplot(Hpca,scale =1, var.scale = 1,varname.size = 3, labels.size=10, circle = TRUE)



summary(Hpca)

#Importance of components%s:

PC1 PC2 PC3 PC4 PC5 PC6 PC7

#Standard deviation 2.0793 0.9482 0.9109 0.68320 0.54619 0.33745 0.26204

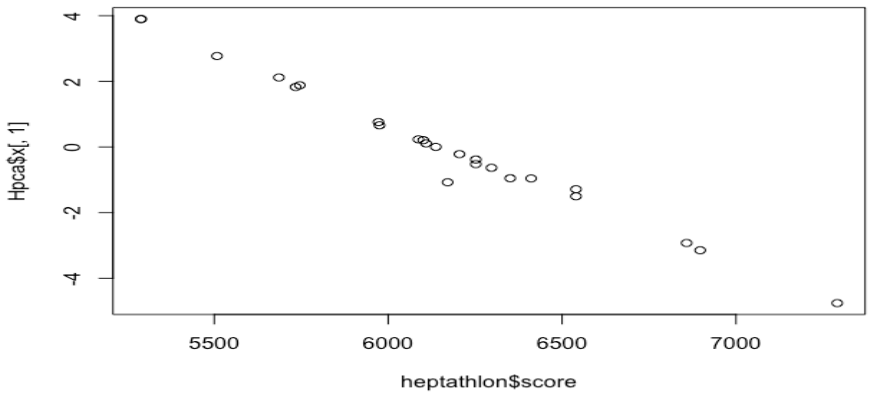
#Proportion of Variance 0.6177 0.1284 0.1185 0.06668 0.04262 0.01627 0.00981

#Cumulative Proportion 0.6177 0.7461 0.8646 0.93131 0.97392 0.99019 1.00000

#From the plot pc1 is mainly describes hurdles, longjump and run200m, where as pc2 mainly describes run800m, highjump data

##6.b.vi

plot(heptathlon$score, Hpca$x[, 1])



hpca_cor = cor(heptathlon$score, Hpca$x[, 1])

#Strong correlation between score and the projection values on the PC1 axis implies that the PC1 is a good indicator of the overall scores assigned to the athletes

##6.c.i

```
classDigits_data = read.csv("classDigits.csv", header = T)
classDigits_data = classDigits_data[,-1]
#Eigen vectors have been extracted from "rotation" attribute of the prcomp function.
eig_digitdata = prcomp(classDigits_data)
```

#"eig_digitdata$rotation" prints all the eigen vectors

##6.c.ii

```
eig_matrix= matrix(eig_digitdata$center,28,28,byrow=TRUE)
writeJPEG(eig_matrix,target="meanDigit.jpg") #Creat JPEG image for MeanData
```

##6.c.iii

#Reconstruction matrix for #15

```
reconstuct_15_5 = eig_digitdata$center + (eig_digitdata$x[15,1:5] %*%
t(eig_digitdata$rotation[,1:5]))
reconstuct_mat_15_5 = matrix(reconstuct_15_5,28,28,byrow=TRUE)
writeJPEG(reconstuct_mat_15_5,target="image15-5.jpg")
reconstuct_15_20 = eig_digitdata$center + (eig_digitdata$x[15,1:20] %*%
t(eig_digitdata$rotation[,1:20]))
reconstuct_mat_15_20 = matrix(reconstuct_15_20,28,28,byrow=TRUE)
writeJPEG(reconstuct_mat_15_20,target="image15-20.jpg")
```

#Reconstruction matrix for #100

```
reconstuct_15_100 = eig_digitdata$center + (eig_digitdata$x[15,1:100] %*%
t(eig_digitdata$rotation[,1:100]))
reconstuct_mat_15_100 = matrix(reconstuct_15_100,28,28,byrow=TRUE)
writeJPEG(reconstuct_mat_15_100,target="image15-100.jpg")
reconstuct_100_5 = eig_digitdata$center + (eig_digitdata$x[100,1:5] %*%
t(eig_digitdata$rotation[,1:5]))
reconstuct_mat_100_5 = matrix(reconstuct_100_5,28,28,byrow=TRUE)
writeJPEG(reconstuct_mat_100_5,target="image100-5.jpg")
reconstuct_100_20 = eig_digitdata$center + (eig_digitdata$x[100,1:20] %*%
t(eig_digitdata$rotation[,1:20]))
reconstuct_mat_100_20 = matrix(reconstuct_100_20,28,28,byrow=TRUE)
writeJPEG(reconstuct_mat_100_20,target="image100-20.jpg")
reconstuct_100_100 = eig_digitdata$center + (eig_digitdata$x[100,1:100] %*%
t(eig_digitdata$rotation[,1:100]))
reconstuct_mat_100_100 = matrix(reconstuct_100_100,28,28,byrow=TRUE)
writeJPEG(reconstuct_mat_100_100,target="image100-100.jpg")
```