HomeWork_2 Sai Saketh Boyanapalli

1 Concordance and Discordance Given vectors

```
##    1  2  3  4  5  6  7
## 1 NA NA NA NA NA NA NA
## 2 -1 NA NA NA NA NA NA
## 3 -1 -1 NA NA NA NA NA
## 4 -1 -1  1 NA NA NA NA
## 5  1  1 -1 -1 NA NA NA
## 6 -1 -1 -1 -1  1 NA NA
## 7 -1 -1 -1 -1  1  1 NA
```

So, if we look at the above matrix there are 6 Concardant Pairs and 15 Discordant Pairs.

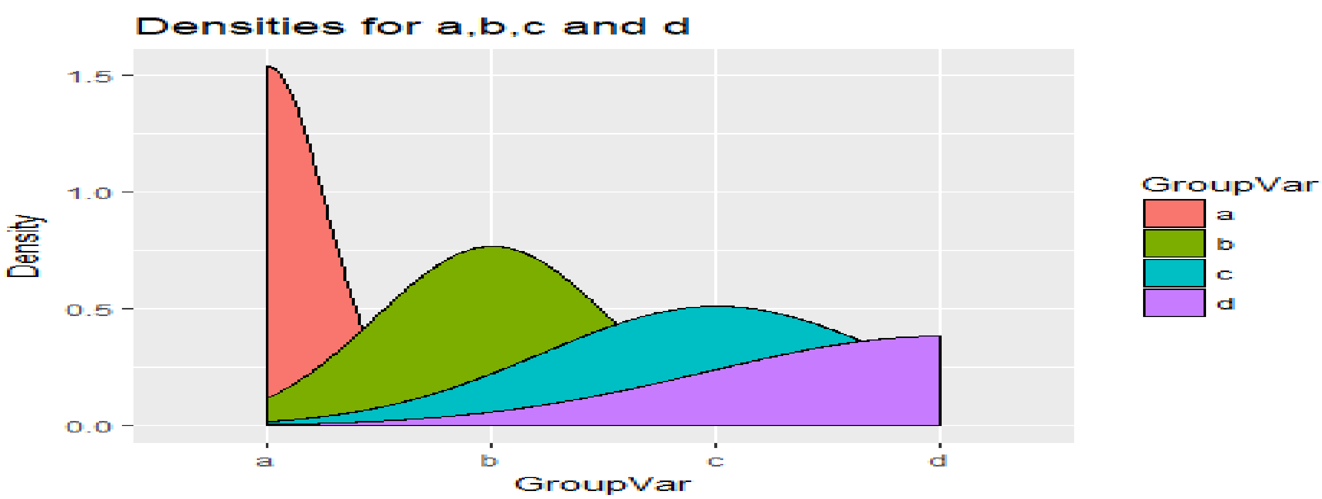
Question 2 Outlier example ANS. Its us **Humans,** The animal Selected at the end is Human.

Question 3 Generating data and advanced density plots.

3 a)

Code is provided

3 b)



Question 4 Shark Attacks

4 a)

```
GSAF <- read.csv("ISE 5103 GSAF5.csv")
```

If we closely look at the data, during the earlier stages the technology and communications are not as par as what we have today. So, the data might be missing lot of useful information and analysis of this can be misleading. Recency and Obsolescence. 4 b)

```
GSAFData <- GSAF[c(GSAF$Year >= 2000),] # selecting attacks from year 2000 onwards
```

4 c)

```
DateTimeObject <- as.Date(GSAFData$Date, "%d-%b-%y")
GSAFData <- data.frame(GSAFData, DateTimeObject)
```

4 d)

```
## Percentage of missing values: 7.4360499702558
```

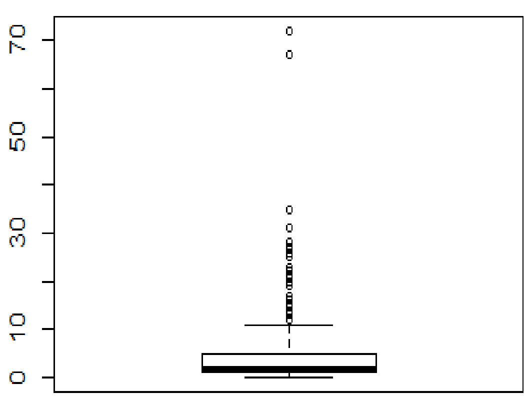
4 e)

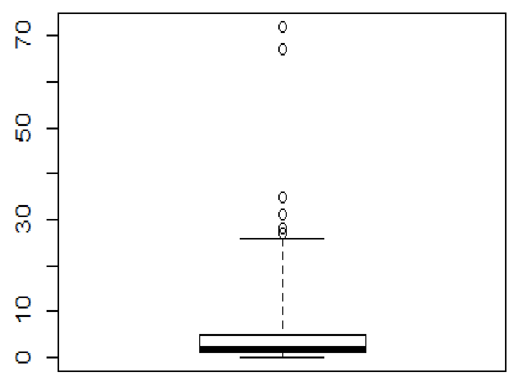
```
GSAFData <- GSAFData[!is.na(GSAFData$DateTimeObject),] # deleting rows with
NA values in column DateTimeObject
```

4 f) i)

```
DaysBetween <- as.numeric(diff(GSAFData$DateTimeObject)) # difference in da
ys on DateTimeObject
# adding DaysBetween to DataFrame with first row element as NA
GSAFData <- data.frame(GSAFData, DaysBetween = c(NA, DaysBetween))
GSAFData$DaysBetween[GSAFData$DaysBetween < 0 | GSAFData$DaysBetween > 100]
<- 0

4 f) ii)
```

DAYS BETWEEN SHARK ATTACKS





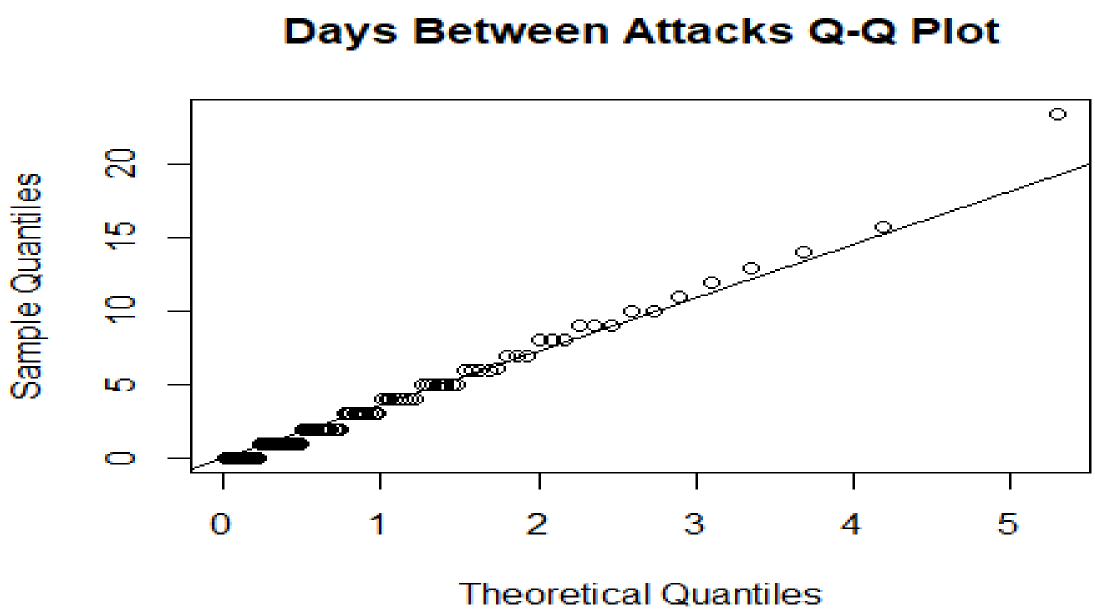
BOX PLOT ADJUSTED BOX PLOT

We can see there are many outliers in the boxplot and adj box plot tries to adjust this but we still see lot of outliers and most data is between 0 - 10 days.

4 f) iii) WE can see from the boxplot that there are lot of ouliers so, neither of them are applicable in this case. Since Grubbs's test just points one outlier in the data at a time so, its very hard to

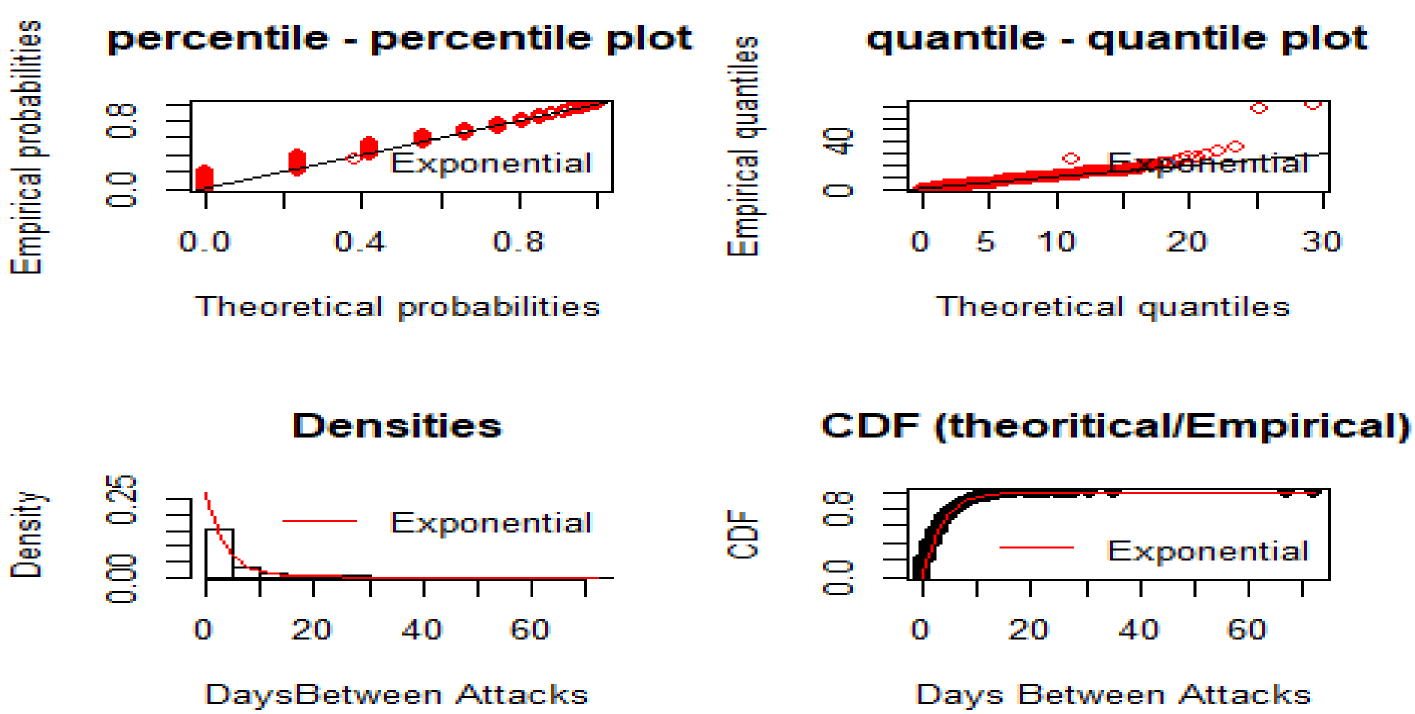
remove outliers one by one and in case of Generalized ESD it will allow to detect multiple outliers but is not robust.

4 g)



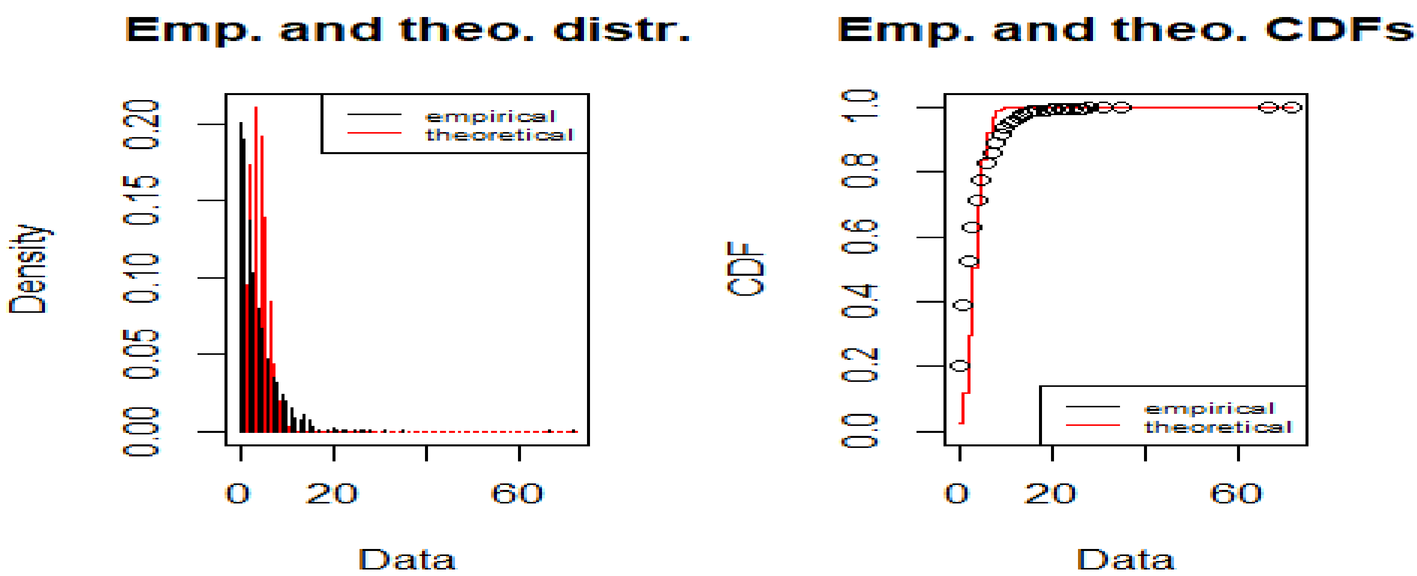
Here we see the difference between days and the theoretical values match So, we can say it follows exponential distribution.

4 h)



From the above distribution we can see that the values of days between is exponentially distributed.

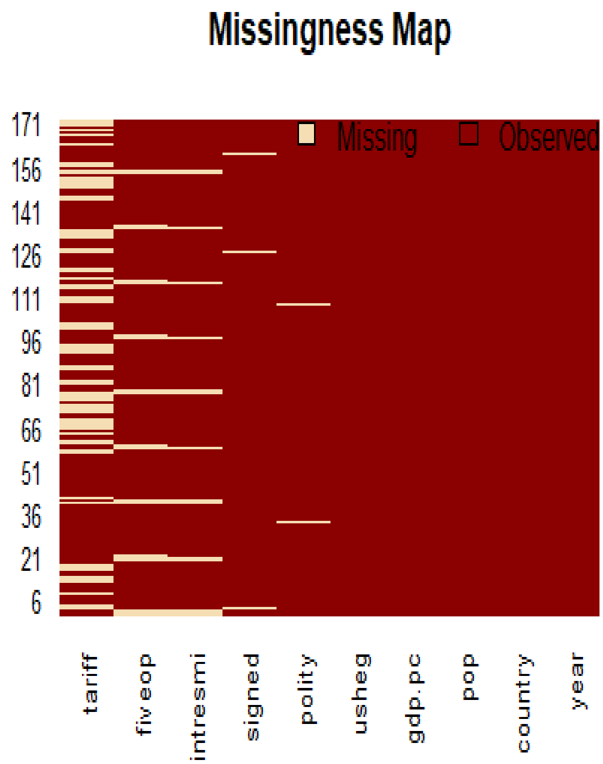
4 i)

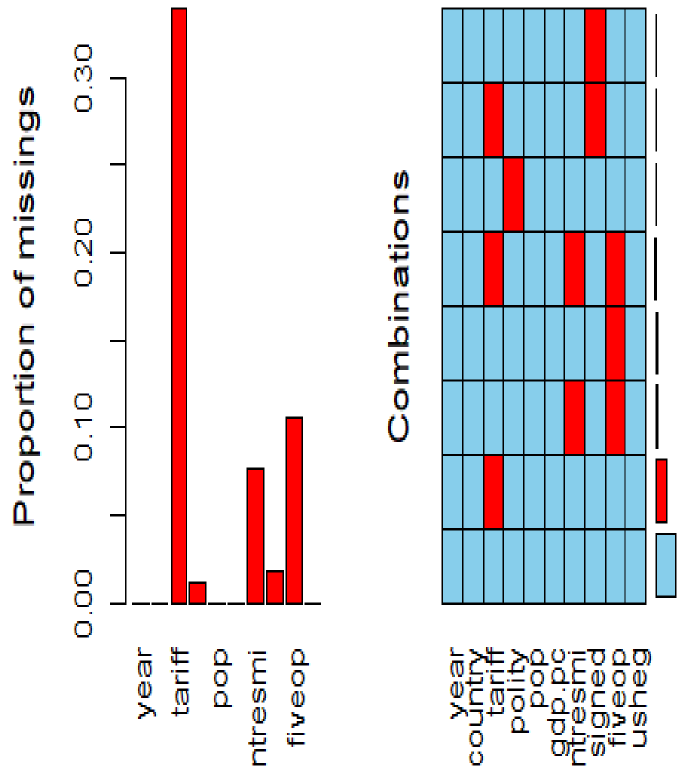


I respond to the claim not so positively, from the both graphs above the empirical and theoretical values don't match perfectly but kind of line up. Earlier we have questioned on timeliness of the data so, and While converting the date to date object in R we have skipped through lot of data

Question 5 Missing Data

5 a)





We can see that tariff is missing most of the data and in other variables some have missing values and some don't

5 b)

```
##  Pearson's Chi-squared test
## data:  freetrade$tariff and freetrade$country
## X-squared = 831.96, df = 736, p-value = 0.007819
```

Here we can see that the p - value is less than 0.05 So, we reject the null hypothesis and conclude that 2 - variables are dependent.

```
## X-squared = 684.79, df = 602, p-value = 0.01063 For Drop Nepal
```

Again, we get p - value less than 0.05 so, we reject null hypotheis and conclude that two variables are dependent.

```
## X-squared = 639.33, df = 574, p-value = 0.03012 For Phillipines
```

Again, we get p - value less than 0.05 so, we reject null hypothesis. but the p value is increasing for phillipines. Missingness in tarrif is dependent on Country and There is no effect removing Nepal and becomes independent if we remove Philippines.

Question 6 Principal Component Analysis

6 a) i)

```r
data("mtcars") # importing data mtcars
corMat <- cor(mtcars, mtcars) # creating corelation matrix using method Ken
dall
```

6 a) ii)

```
eigen() decomposition
$values
 [1] 6.60840025 2.65046789 0.62719727 0.26959744 0.22345110 0.21159612 0.13526199
 [8] 0.12290143 0.07704665 0.05203544 0.02204441

$vectors
           [,1]        [,2]        [,3]         [,4]        [,5]        [,6]
 [1,]  0.3625305 -0.01612440 -0.22574419 -0.022540255 -0.10284468  0.10879743
 [2,] -0.3739160 -0.04374371 -0.17531118 -0.002591838 -0.05848381 -0.16855369
 [3,] -0.3681852  0.04932413 -0.06148414  0.256607885 -0.39399530  0.33616451
```

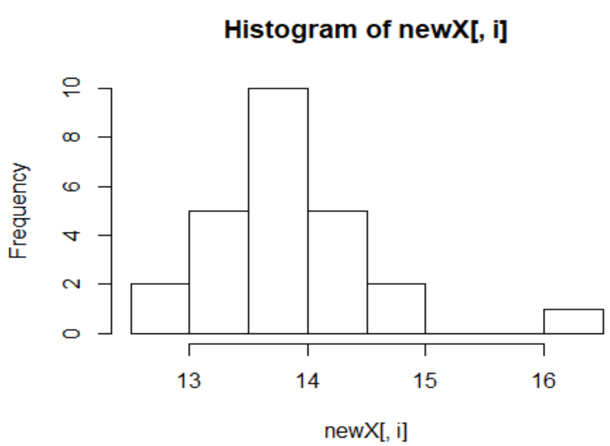
6 a) iii)

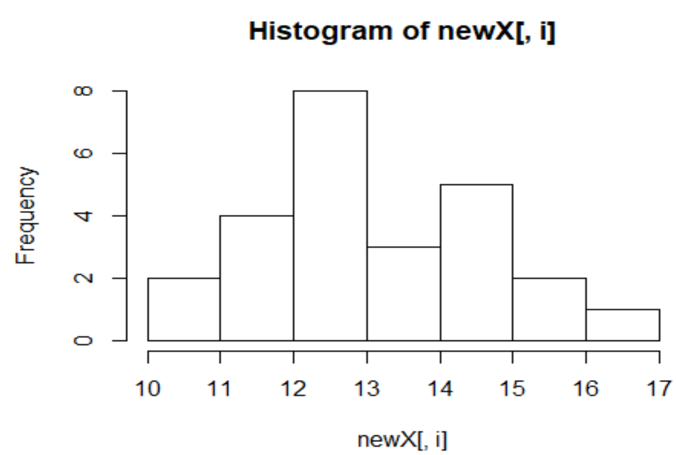
```
$sdev
 [1] 2.5706809 1.6280258 0.7919579 0.5192277 0.4727061 0.4599958 0.3677798
 [8] 0.3505730 0.2775728 0.2281128 0.1484736

$rotation
            PC1         PC2         PC3          PC4         PC5         PC6
mpg  -0.3625305  0.01612440 -0.22574419 -0.022540255  0.10284468 -0.10879743
cyl   0.3739160  0.04374371 -0.17531118 -0.002591838  0.05848381  0.16855369
disp  0.3681852 -0.04932413 -0.06148414  0.256607885  0.39399530 -0.33616451
```

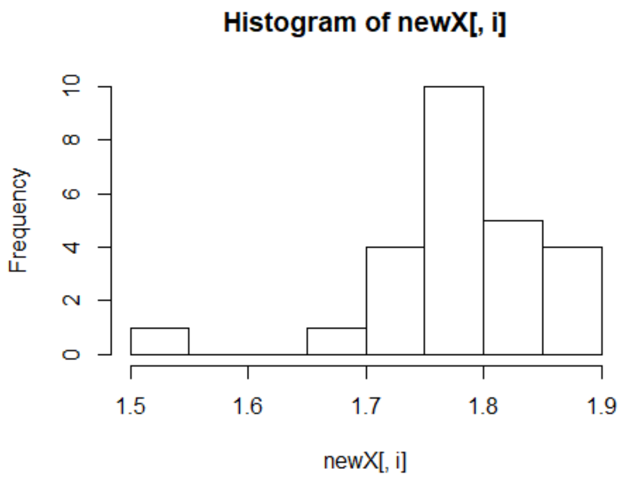
6 a) iv) Principal components match with eigen vectors. Since principal components are eigen vectors

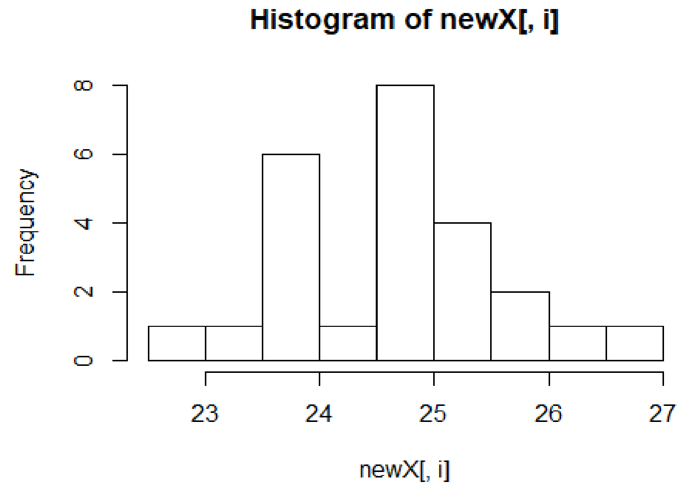
6 a) v) we can see that the dot product of two PCA components is **0** So, we can say that they are **orthogonal** to each other.

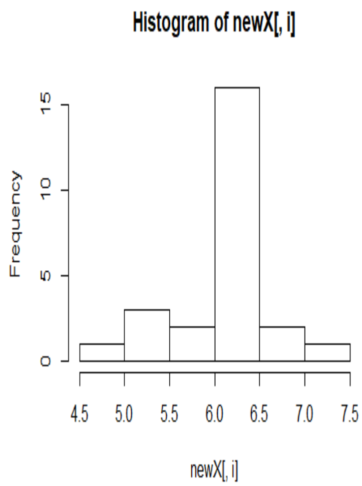
6 b i)

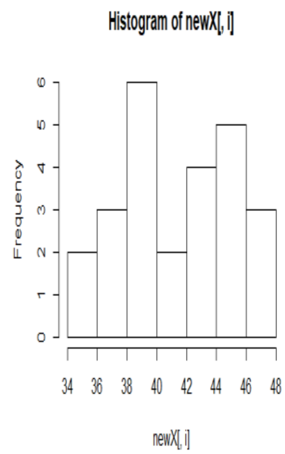


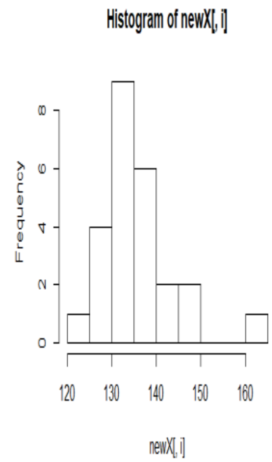


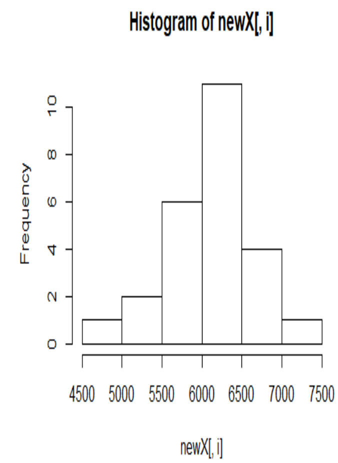












We can see that just one or 2 histograms are skewed and most of them are are reasonably normal.

6 b) ii)

According to this test **G.launa** is an outlier. For events run800m, Longjump, Highjump, Hurdles.

```
# removing Outlier
heptathlon1 <- heptathlon[-25,]
```

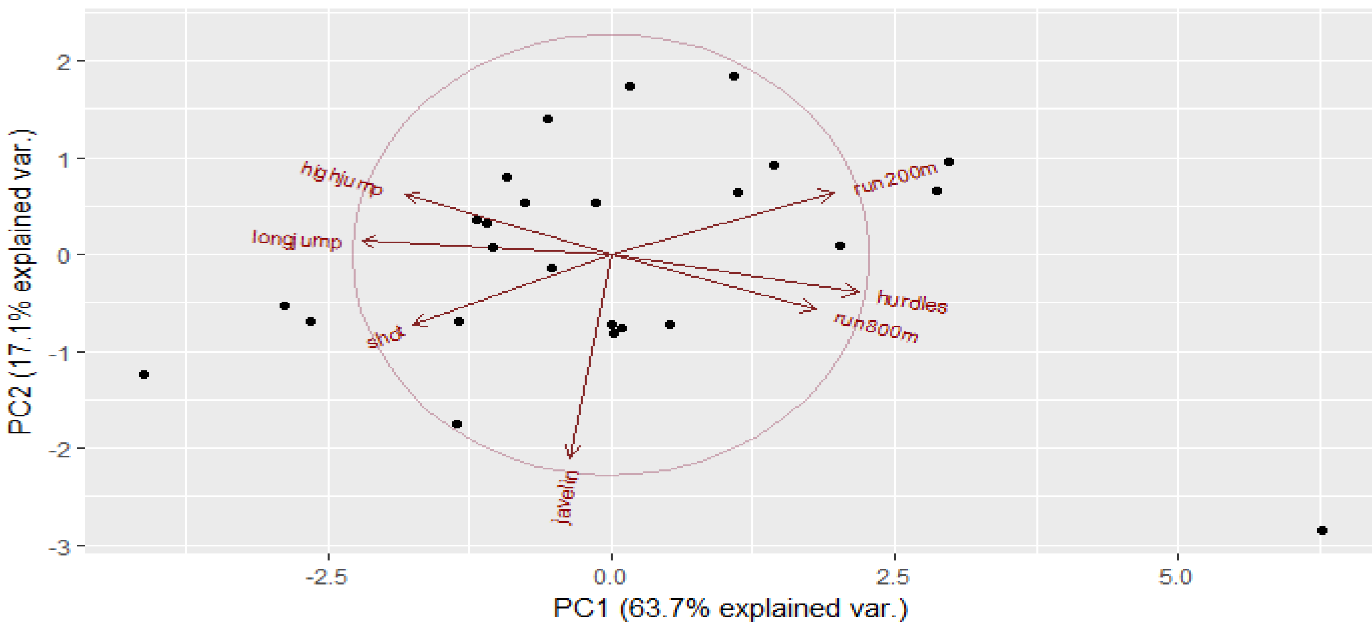
6 b) iii)

```
goodlargevalues <- function(max,columnName){
 for (v in 1:nrow(heptathlon)) {
  heptathlon[v,columnName] <- max - heptathlon[v,columnName]

 }
  return(heptathlon)
}
# making large values good for 200m, 800m run and hurdles
heptathlon2 = goodlargevalues(max(heptathlon$run200m), "run200m")
```

```r
heptathlon2 = goodlargevalues(max(heptathlon$run800m), "run800m")
heptathlon2 = goodlargevalues(max(heptathlon$hurdles), "hurdles")
```

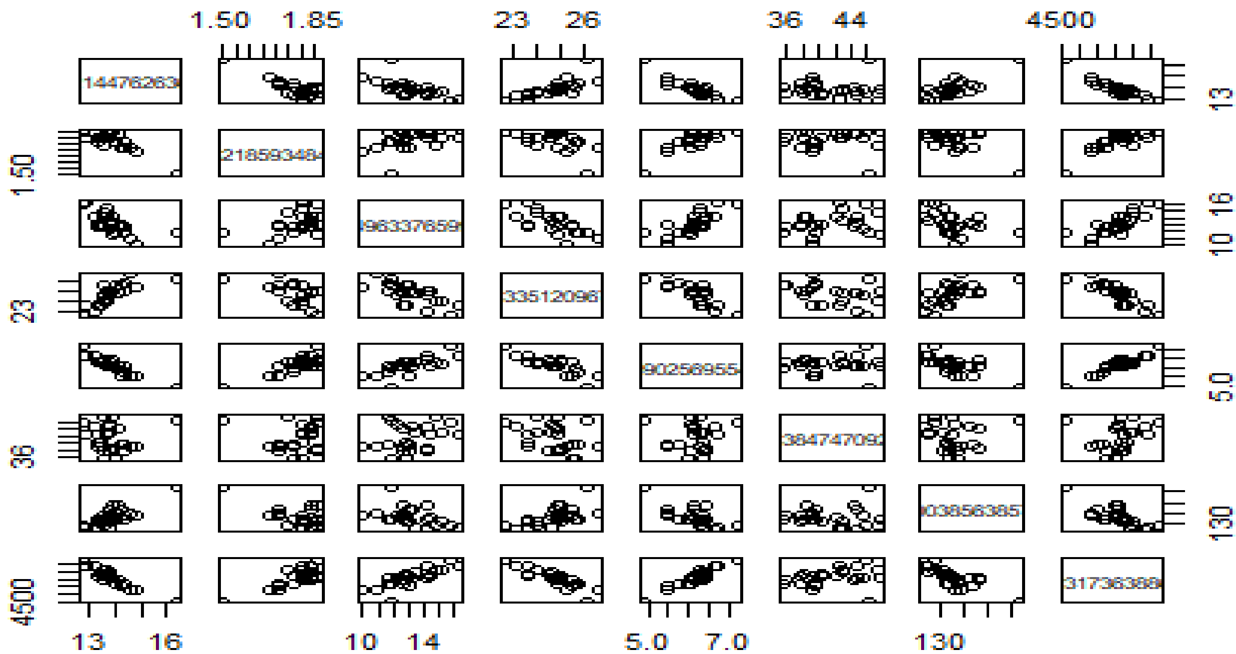
6 b) iv) hpca <- prcomp(heptathlon[,1:7], scale=T) # principal component analysis on heptathlon

6 b) v)



Here vectors represent events and points represent Atlethes. With PC1 retaining 63.7% of the data and PC2 retaining 17.1% of the data.

6 b) vi)



6 c) i)

```
##                      PC1           PC2          PC3           PC4
## pixel0  2.219274e-20 -5.732181e-19 6.287447e-20 -1.759315e-19
## pixel1  2.081668e-17  1.110223e-16 2.081668e-17  8.326673e-17
## pixel2 -1.942890e-16  0.000000e+00 4.857226e-17 -4.163336e-17
##                      PC5
## pixel0  2.794486e-19
## pixel1 -8.326673e-17
## pixel2  5.551115e-17
```

6 c) ii)

```
digitMatrix <- matrix(pcaclass$center,28,28,byrow=T)
# Provide a 28*28 matrix for all mean values byrow and call it"digitmatrix"
.
library(jpeg)
writeJPEG(digitMatrix,target="meanDigit.jpg")
```

6 c) iii)

```
imageReconstruction <- function(k,imageno,imagefilename){
  #This takes argument k = no of principal components, image no to be selec
ted, output file name.
reconstruct <- pcaclass$x[,1:k]%*%t(pcaclass$rotation[,1:k])
completeReconstruct = scale(reconstruct, center = -1 *pcaclass$center)
writeJPEG(matrix(completeReconstruct[imageno,],28,28,byrow=TRUE), target =
imagefilename)
}
imageReconstruction(5,15,"image15,k5")
```

```
imageReconstruction(20,15,"image15,k5")
imageReconstruction(100,15,"image15,k5")
imageReconstruction(5,100,"image15,k5")
imageReconstruction(20,100,"image15,k5")
imageReconstruction(100,100,"image15,k5")
```

6 c) iV)

```
classTest <- read.csv("class7test.csv") # reading data
classTest1 <- classTest[,-c(1,2,787)] # removing colums 1, 2, 787
classTestReconstruct = scale(classTest1, center = pcaclass$center, pcaclass
$scale)%*%pcaclass$rotation

Mahalonabis average distance images 1 – 7 with the original data

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

[137.5314, 155.129 , 115.7664, 157.2432, 179.0976 , 495.8419]

[2:3, 2:4, 2:5, 2:6, 2:7]

[190.005, 121.9345, 147.4192, 217.8039, 567.2395]

[3:4, 3:5, 3:6, 3:7]

[138.2786, 142.8912, 239.3133, 466.1846

[4:5, 4:6, 4:7]

[160.2702, 206.7595, 515.8127]

[5:6, 5:7]

[ 233.4709, 482.7759]

[6:7]

[573.4578]
```

6 c) v)

For test images 5, 6 and 7 the lowest value of k

Are

108

78

96