Submission by: Sahibzada Saadoon Hammad

Assignment #3

1. Extract as much as possible information coming out from the data set using graphical tools as described in Chapter 2 of Everett.

Data Statistics:

No of rows	No of columns
5	65

Column	Description	Units
HR	Heart beat rate of the baby	Beats/min
BW	It is the weight of a baby at the time of birth	Grams
Factor 68	Variable from spectral analysis of 24-hour recordings of electrocardiograms ad respiratory movements	-
Gesage	Gestation is the period between conception and birth	Weeks

Column Wise Mean			
HR	**BW**	**Factor68**	**Gesage**
130.1538462	3321.384615	0.3332154	39.8153846

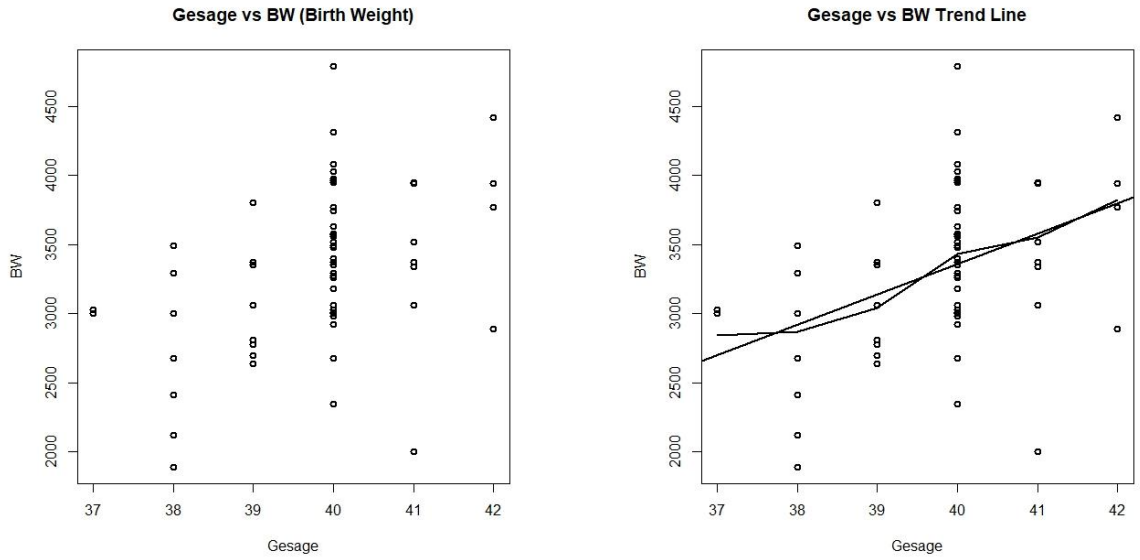
Correlation Matrix:

	HR	BW	Factor68	Gesage
HR	1	-0.02192954	0.2098967	0.04031584
BW	-0.02192954	1	-0.0785167	0.42490365
Factor68	0.20989675	-0.0785167	1	-0.2457091
Gesage	0.04031584	0.42490365	-0.2457091	1

From the table above it can be seen that the correlation between Birth Weight and Gestational Age is high which means that these pair are highly correlated as compared to the correlation between other variables. For example, the correlation between HR and BW is negative which means that if one variable is increasing

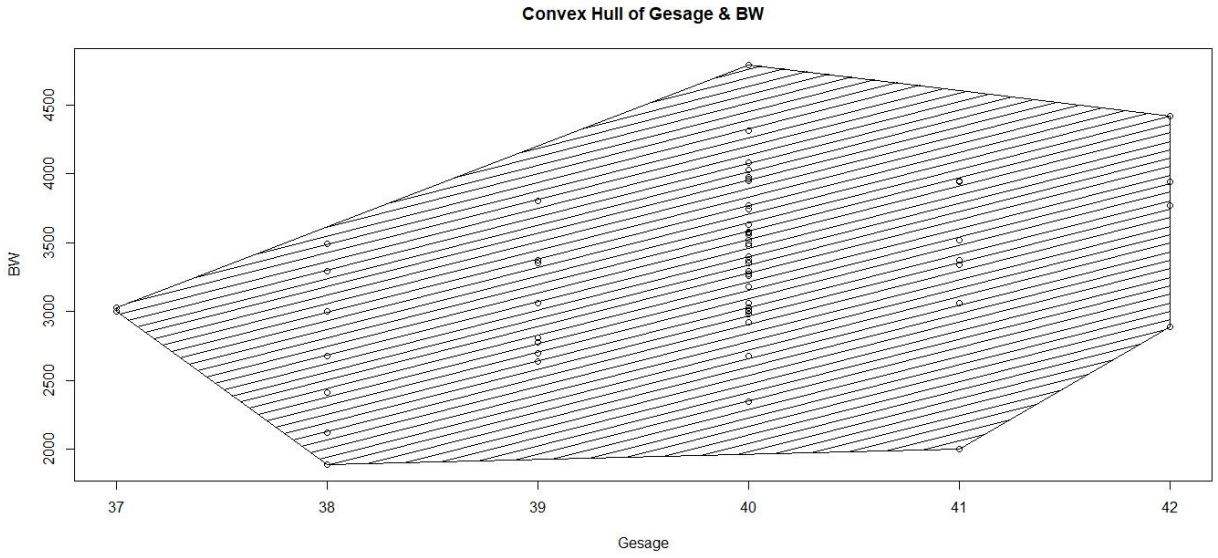
Scatterplot:

By having a look at the correlation matrix above, scatterplot for Gestation Age and BW is given below:



The above graph with regression line shows that there is dependency of Gestational age with the Birth Weight (BW). With the increase of Gesage, the BW is also increasing, so we can say that there is some sort of dependency between these two variables.

Convex Hull:

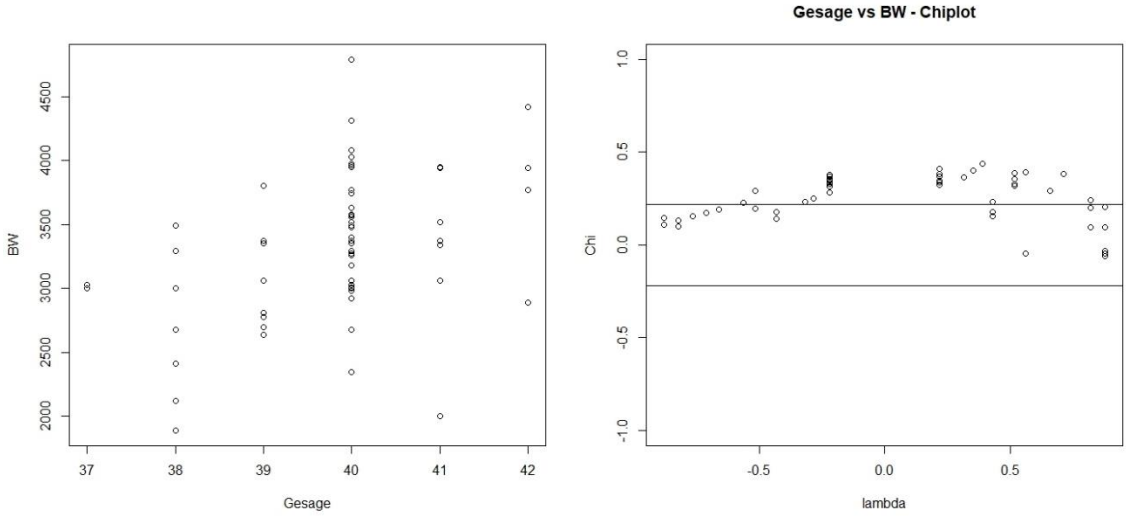


The hull function in R gives us the insight to possible outliers in the data by computing a convex hull around the distribution of data. From the above graphs points at the boundary can be indicated as possible outliers but we cannot declare these point as final outliers. In this data BW

with value of 4500 be a possible outlier. But before removing this value we have to perform other statistical method on our data. The effect on correlation between these two variables with and without convex hull can be seen from the table below:

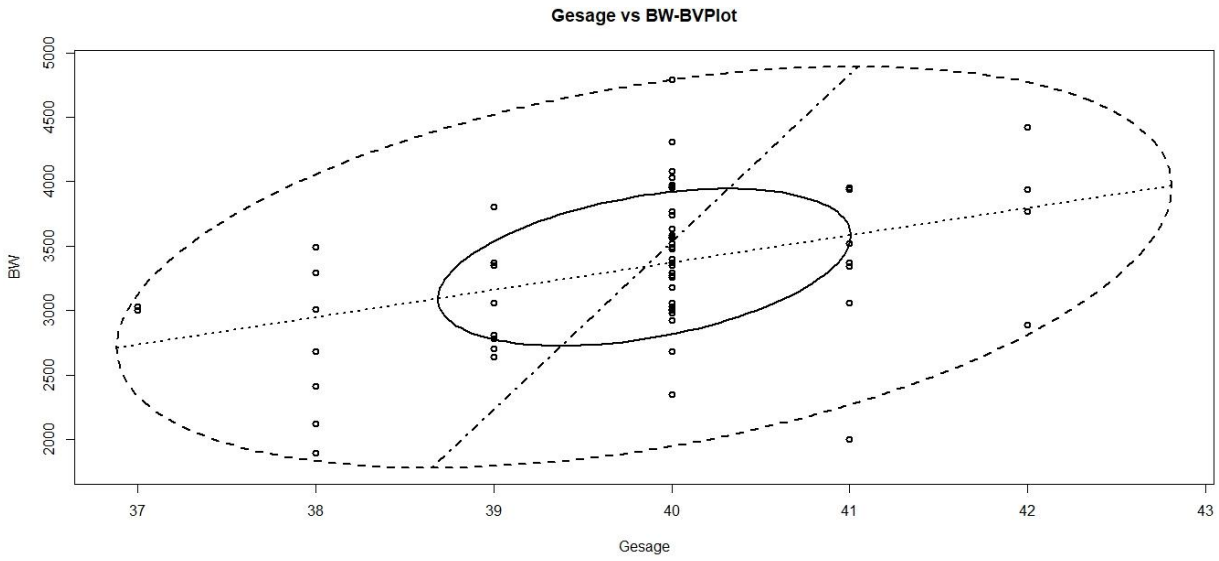
cor(Gesage,BW)	cor(Gesage[-hull],BW[-hull])
0.4249037	0.5038747

Chi-plot:



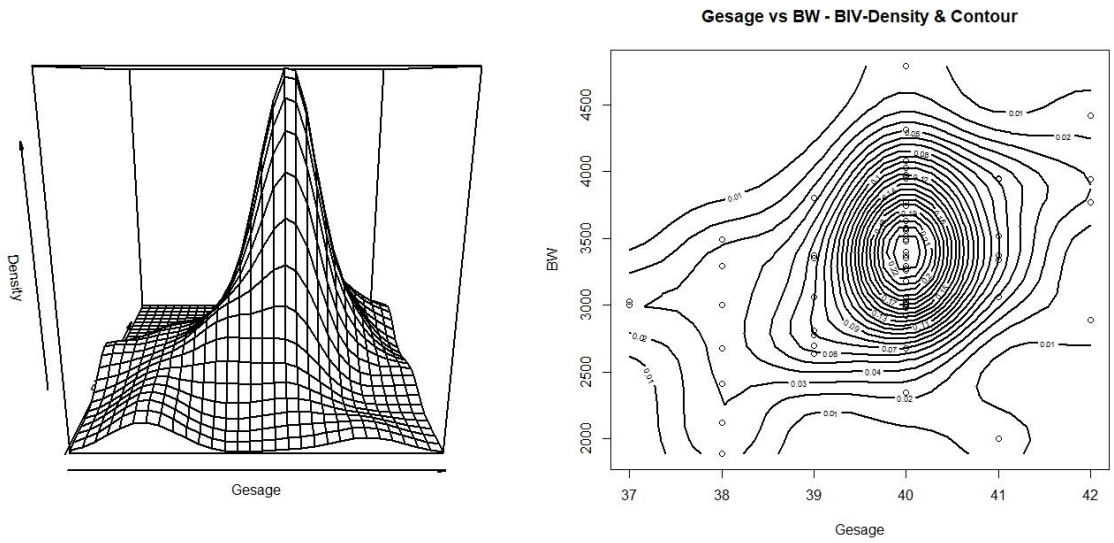
Chi plot is another method to find dependency between variables. It can be seen from the plot above that there are less number points in the horizontal band. There is clear dependency between variables as most of the points lie outside the horizontal band.

Bivariate Boxplot:



Bivariate box plot is used to identify outliers in data. In consists of two concentric ellipses. The ellipse on the inside indicates that it contains 50% of the data. The ellipse on the outside helps to identify outlier values. In this case, the plot shows that there is **one value** outside the ellipses which means that it is an outlier. It does not show the same outliers as show in the convex hull plot calculated above.

Perspective and Contour Plot:



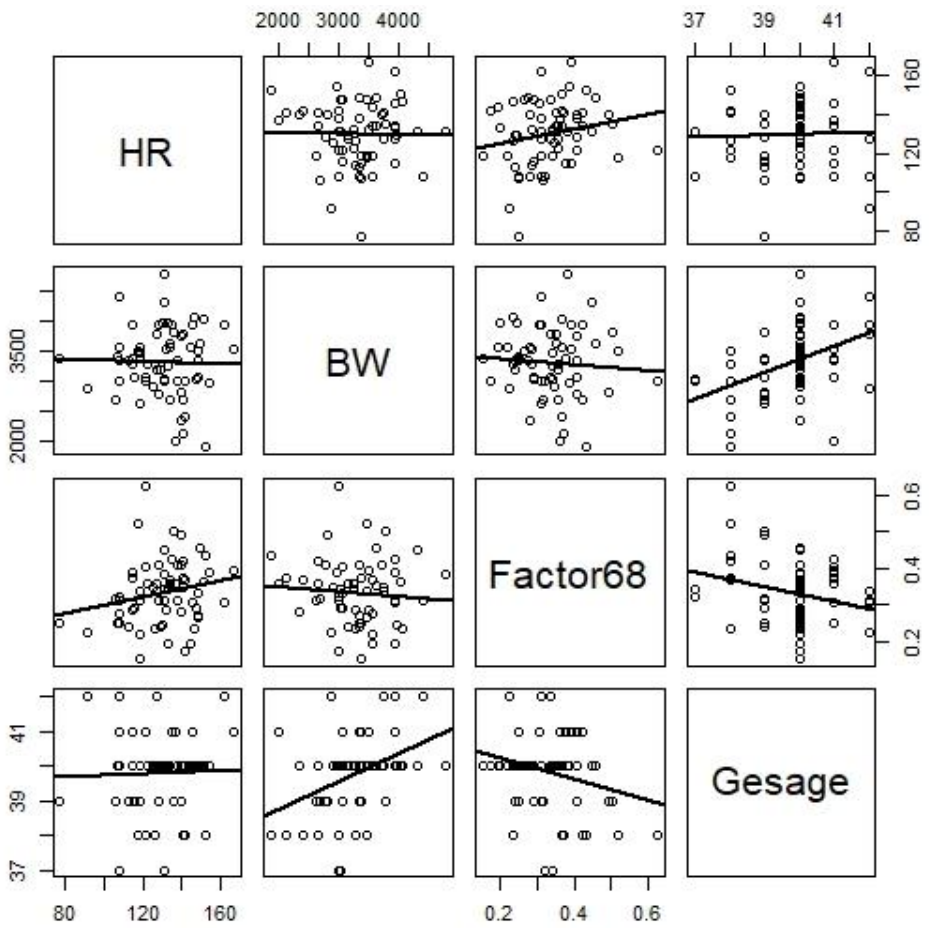
From the Bivariate density graph, the data has little skewness towards the right side which means that the data is negatively skewed and it cannot be termed as a normal distribution. Skewness is the asymmetry in statistical distribution. Similarly, it is can be easily seen from the contour plot that there is skewness in the bivariate density of Gesage and BW

Scatterplot Matrix:

Scatterplot matrix is a better way to look at data graphically as it gives plots all variables. By looking at the scatterplot matrix above it can be seen that:

- There is a positive correlation between Gesage and BW and it obvious from the trend line that with the increase of Gesage, second variable BW is increasing.
- Gesage and HR are not dependant.
- Gesage and Facotr68 are inversely proportional which means that if one factor is increasing and other is decreasing.

Similarly, other relations between variables can be derived from the scatterplot matrix.



R Code:

// loading the data

sids<-source("F:\\UJI\\Courses\\Applied Mathematics\\Assignment 3\\sids.dat")$value

attach(sids)

sidsdata = sids[, -1]

// calculating the column means

center<-colMeans(sidsdata)

// calculating variance

var(sidsdata)

// calculating correlation

```
cor(sidsdata)

// plotting the graph between Gesage and BW

par(mfrow=c(1,2))

par(pty="s")

plot(Gesage,BW,pch=1,lwd=2)

title("Gesage vs BW (Birth Weight)",lwd=2)

plot(Gesage,BW,pch=1,lwd=2)

abline(lm(BW~Gesage),lwd=2)

lines(lowess(Gesage,BW),lwd=2)

title("Gesage vs BW Trend Line",lwd=2)

par(mfrow=c(1,2))

hist(BW,lwd=2)

boxplot(BW,lwd=2)

hull<-chull(Gesage,BW)

plot(Gesage,BW,pch=1)

title("Convex Hull of Gesage & BW",lwd=2)

polygon(Gesage[hull],BW[hull],density=15,angle=30)

cor(Gesage,BW)

cor(Gesage[-hull],BW[-hull])


par(mfrow=c(1,1))

chiplot(Gesage,BW,vlabs=c("Gesage","BW"))
```

```
title("Gesage vs BW - Chiplot",lwd=2)

dev.off()

par(mfrow=c(1,1))

bvbox(cbind(Gesage,BW),xlab="Gesage",ylab="BW")

title("Gesage vs BW-BVPlot",lwd=2)

par(mfrow=c(1,1))

h2d<-hist2d(Gesage,BW)

persp(h2d,xlab="Gesage",ylab="BW",zlab="Frequency")

// load bivdensity function from file

par(mfrow=c(1,1))

den1<-bivden(Gesage,BW)

persp(den1$seqx,den1$seqy,den1$den,xlab="Gesage",ylab="BW",zlab="Density",lwd=2)

title("Gesage vs BW-BIV-Density",lwd=2)

par(mfrow=c(1,2))

den1<-bivden(Gesage,BW)

persp(den1$seqx,den1$seqy,den1$den,xlab="Gesage",ylab="BW",zlab="Density",lwd=2)

plot(Gesage,BW)

contour(den1$seqx,den1$seqy,den1$den,lwd=2,nlevels=20,add=TRUE)

title("Gesage vs BW - BIV-Density & Contour",lwd=2)

pairs(sidsdata,panel=function(x,y) {abline(lsfit(x,y)$coef,lwd=2)

                                points(x,y)})
```

2. Perform a PCA and interpret the results.

PCA:

A summary of the PCA computed for our data is given below:

	Comp 1	**Comp 2**	**Comp 3**	**Comp 4**
Standard Deviation	1.2402115	1.0707983	0.8948664	0.717273
Proportion of Variance	0.3845312	0.2866522	0.2001965	0.1286201
Cumulative Variance	0.3845312	0.6711834	0.8713799	1

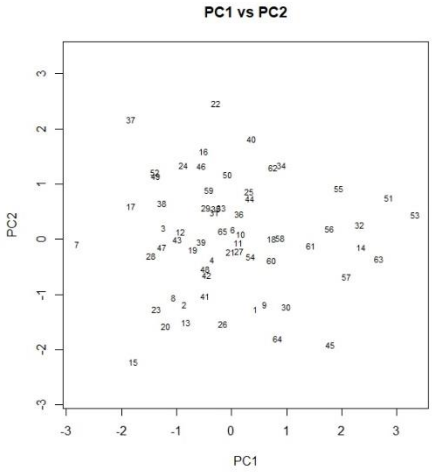
The normal practice is to take into account those components which cover approximately 80% of the cumulative variance of given variables. In this case 67% is too low to consider with only 2 components, so there will be 3 principal components which cover 87% of the cumulative variance.

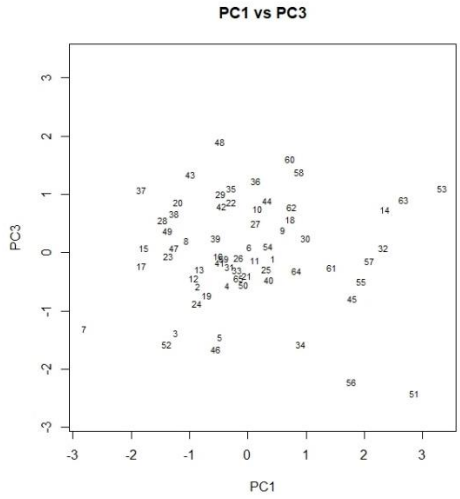
LOADINGS	**Comp 1**	**Comp 2**	**Comp 3**
HR	0.148	0.76	0.584
BW	-0.59	0.31	-0.488
Factor68	0.444	0.517	-0.633
Gesage	-0.658	0.242	0.142

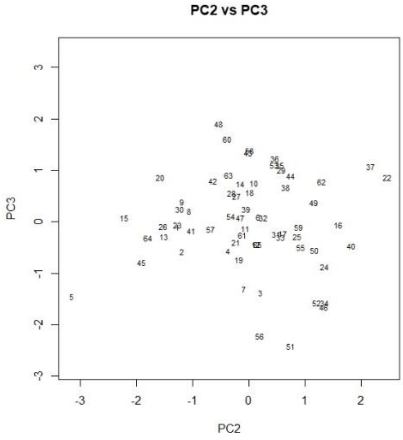
From loadings we can see that **BW** and **Gesage** contribute to **component 1**. **HR** (heart rate) contributes to **component 2** and **Factor68** contributes to **component 3**.

Now that component have been calculated, PC scores are calculated for our data. Value for all components of data are given in the table below:

PC Scores	**Comp 1**	**Comp 2**	**Comp 3**
1	0.4242506	-1.260824	-0.094567
2	-0.86315	-1.183272	-0.573037
3	-1.240918	0.2060685	-1.381845
4	-0.35403	-0.367613	-0.565086
5	-0.476356	-3.156639	-1.444465







Discussion: It can be seen from the graphs above that components are not correlated to each other. The distribution of points in the graphs is random.

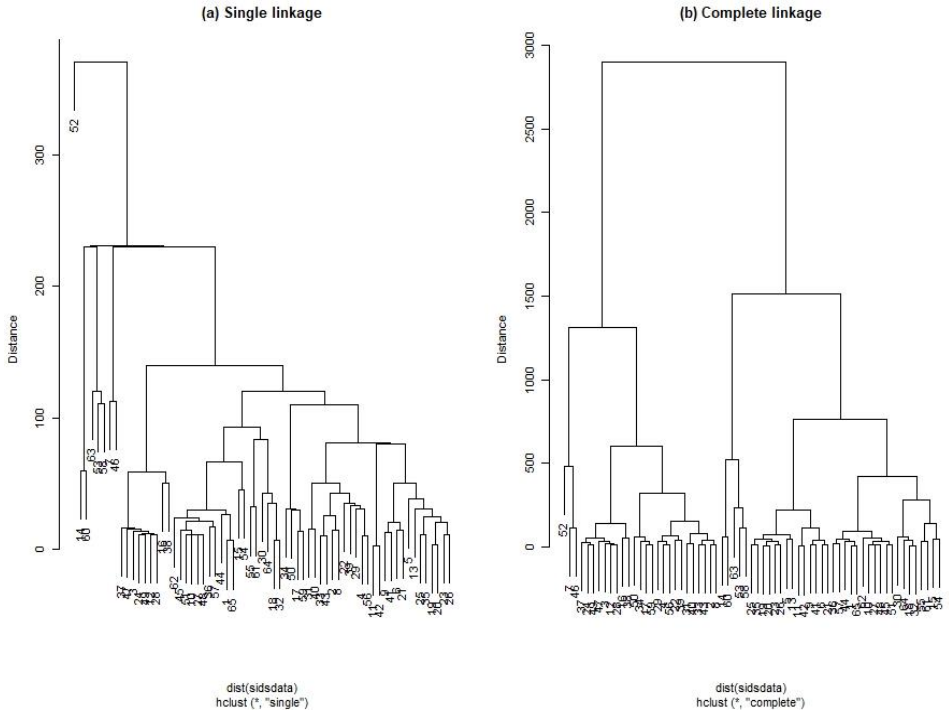
R Code:

```
sidsdata<-source("F:\\UJI\\Courses\\Applied Mathematics\\Assignment
3\\sids.dat")$value

attach(sids);
sidsdata = sids[, -1]
 pairs(sidsdata)
pairs(sidsdata,panel=function(x,y) {abline(lsfit(x,y)$coef,lwd=2)

                        points(x,y)})

cor(sidsdata.dat[,])
```

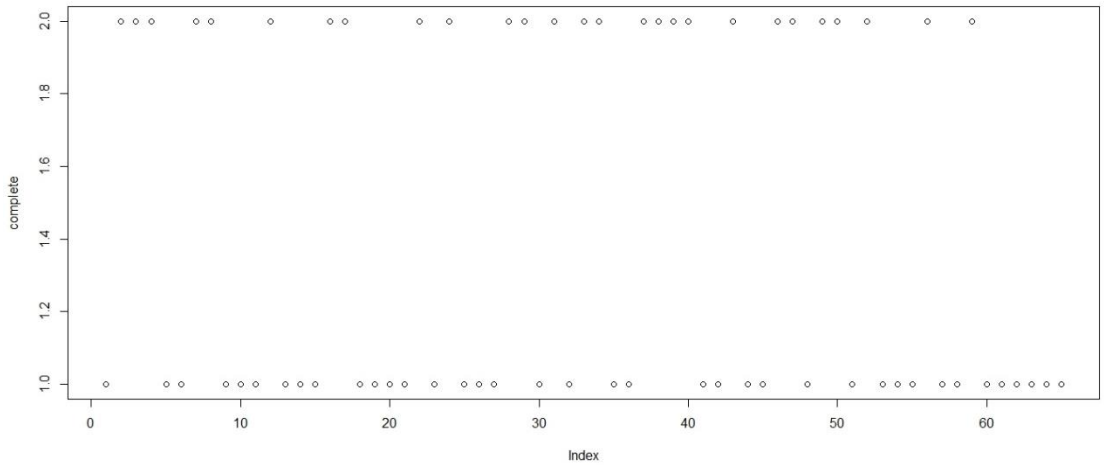
```
//calculates the Principal components

sidsdata.pc<-princomp(sidsdata,cor=TRUE)

summary(sidsdata.pc,loadings=TRUE)

// outputs the component score
sidsdata.pc$scores[,1:3]

// graph of PC1 and PC2

par(pty="s")

plot(sidsdata.pc$scores[,1],sidsdata.pc$scores[,2],

ylim=range(sidsdata.pc$scores[,1]),

xlab="PC1",ylab="PC2",type="n",lwd=2)

text(sidsdata.pc$scores[,1],sidsdata.pc$scores[,2],

labels=abbreviate(row.names(sidsdata)),cex=0.7,lwd=2)

title("PC1 vs PC2",lwd=2)


// graph of PC1 and PC3

par(pty="s")

plot(sidsdata.pc$scores[,1],sidsdata.pc$scores[,3],

ylim=range(sidsdata.pc$scores[,1]),

xlab="PC1",ylab="PC3",type="n",lwd=2)

text(sidsdata.pc$scores[,1],sidsdata.pc$scores[,3],

labels=abbreviate(row.names(sidsdata)),cex=0.7,lwd=2)

title("PC1 vs PC3",lwd=2)

// graph of PC2 and PC3s
```

```
par(pty="s")

plot(sidsdata.pc$scores[,2],sidsdata.pc$scores[,3],

ylim=range(sidsdata.pc$scores[,1]),

xlab="PC2",ylab="PC3",type="n",lwd=2)

text(sidsdata.pc$scores[,2],sidsdata.pc$scores[,3],

labels=abbreviate(row.names(sidsdata)),cex=0.7,lwd=2)

title("PC2 vs PC3",lwd=2)
```

3. Find homogeneous clusters amongst the individuals without considering the variable Group (use hierarchical and k-means methods under two distinct choices of distance methods). Compare the results with the existing groups given by variable "group". Then perform LDA and classify into these groups the following two new observations: Obs1: (110,3320,0.240,39); Obs2: (120,3310,0.298,37).



Complete Linkage:

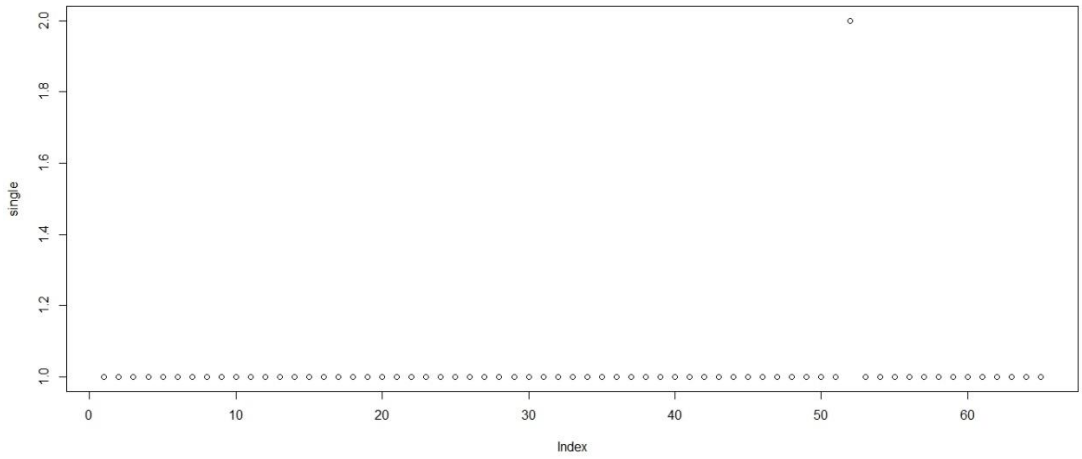


complete<-cutree(hclust(dist(sidsdata),method="complete"),k=2)

Means of Clusters:

HR	BW	Factor68	Gesage
127.1605263	2959.868421	0.3284211	39.5
134.366667	3830.185185	0.339963	40.259259

Single Linkage:



single<-cutree(hclust(dist(sidsdata),method=" single "),k=2)

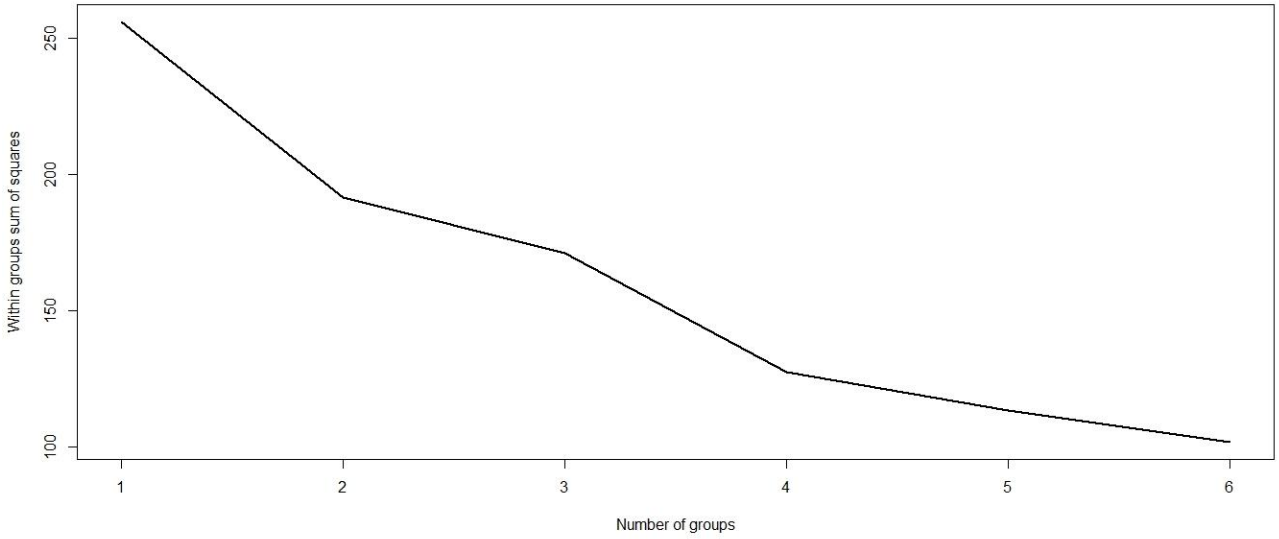
Means of Clusters:

HR	BW	Factor68	Gesage

130.1344	3298.438	0.332438	39.8125
131.4	4790	0.383	40

K Means:

Given data is in different units, to get our data on the same standard units **Original data** is divided by the standard deviation of each column. Doing so will normalize the data. After normalizing data, K means will be used to find the number of clusters. Using K-means function will help in finding the appropriate number of cutoff value i.e. the number of clusters.



From the graph above it can be observed that major break in line is at 2, so the number of clusters considered for this data is **2.**

Cluster	Cluster Size
1	51
2	14

Means of Clusters:

HR	BW	Factor68	Gesage
7.969128	6.143158	3.586216	36.99289
7.897064	5.03391	4.627234	35.11763

Comparison:

Sample From Data for clusters comparison for different groups

Observation Number	HR	BW	Factor68	Gesage	Original Group	Complete Linkage	Single Linkage	K-Means
1	115.6	3060	0.291	39	1	1	1	1
2	108.2	3570	0.277	40	1	2	1	1
3	114.2	3950	0.39	41	1	2	1	1
4	118.8	3480	0.339	40	1	2	1	1
5	76.9	3370	0.248	39	1	1	1	1
6	132.6	3260	0.342	40	1	1	1	1
7	107.7	4420	0.31	42	1	2	1	1
8	118.2	3560	0.22	40	1	2	1	1
9	126.6	3290	0.233	38	1	1	1	2
59	140.9	3770	0.349	40	2	2	1	1

From the table and graphs above, it is obvious that cluster group in complete linkage method is not the same as original group for observation 2-4,7-8. The cluster group is similar for **first observation** in all methods as the original data. For **observation 59** the cluster group in single linkage is different from the original group.

New Observations:

Obs1: (110,3320,0.240,39);

Obs2: (120,3310,0.298,37).

The new observation are add to data and then groups for new observations are predicted.

```
> newdata
      HR   BW Factor68 Gesage
[1,] 110 3320    0.240     39
[2,] 120 3310    0.298     37
```

Prediction:

The predict function gives us the table below which shows a value of 0.85 probability for **first observation** and 0.64 probability for **second observation** which means that both observation will be in **Group 1**.

Observation	Group 1	Group 2
1	0.8536492	0.1463508
2	0.6490989	0.3509011

R Code:

```
// dendograms

par(mfrow=c(1,3))
```

```
plclust(hclust(dist(sidsdata),method="single"),labels=row.names(sidsdata),ylab="Distanc
e")

title("(a) Single linkage")

plclust(hclust(dist(sidsdata),method="complete"),labels=row.names(sidsdata),ylab="Dist
ance")

title("(b) Complete linkage")

pairs(sidsdata,panel=function(x,y) {abline(lsfit(x,y)$coef,lwd=2)

                                points(x,y)})

complete<-cutree(hclust(dist(sidsdata),method="complete"),k=2)

sidsdata.clus<-lapply(1:2,function(nc) row.names(sidsdata)[complete==nc])

sidsdata.mean<-lapply(1:2,function(nc) apply(sidsdata[complete==nc,],2,mean))

sidsdata.mean

sidsdata.clus


single<-cutree(hclust(dist(sidsdata),method="single"),k=2)

sidsdata.clus<-lapply(1:2,function(nc) row.names(sidsdata)[single==nc])

sidsdata.mean<-lapply(1:2,function(nc) apply(sidsdata[single==nc,],2,mean))
sidsdata.mean
sidsdata.clus

sidswc<-cbind(sidsdata[,1],sidsdata[,2],sidsdata[,3],sidsdata[,4])
//This will calculate standard deviation of each column
colsd<-sapply(sidsdata,sd)

//This will normalize data by dividing col value by its SD
ndata<-sweep(sidsdata,2,colsd,FUN='/')

n<-length(ndata[,1])

wss1<-(n-1)*sum(apply(ndata,2,var))
```

```
wss<-numeric(0)

for(i in 2:6) {

        W<-sum(kmeans(ndata,i)$withinss)

        wss<-c(wss,W)

}

wss<-c(wss1,wss)

plot(1:6,wss,type="l",xlab="Number of groups",ylab="Within groups sum of
squares",lwd=2)

kmean<-kmeans(ndata,2)
kmean
lapply(1:2,function(nc) apply(ndata[kmean$cluster==nc,],2,mean))

library(MASS)
dis<-lda(Group~HR+BW+Factor68+Gesage,data=sids,prior=c(0.5,0.5))
newdata<-rbind(c(110, 3320, 0.240, 39),c(120, 3310, 0.298, 37))
colnames(newdata)<-colnames(sidsdata)
newdata<-data.frame(newdata)
predict(dis,newdata=newdata)
```