Multivariate Analysis - Assignment 3

Harshad Kumar Elangovan - 19200349

10/05/2020

UCD School of Mathematics and Statistics Exam Honour Code.

I confirm that I have not given aid, or sought and/or received aid for this assignment.

Name: Harshad Kumar Elangovan
Student Number: 19200349

Question1

Loading the dataset

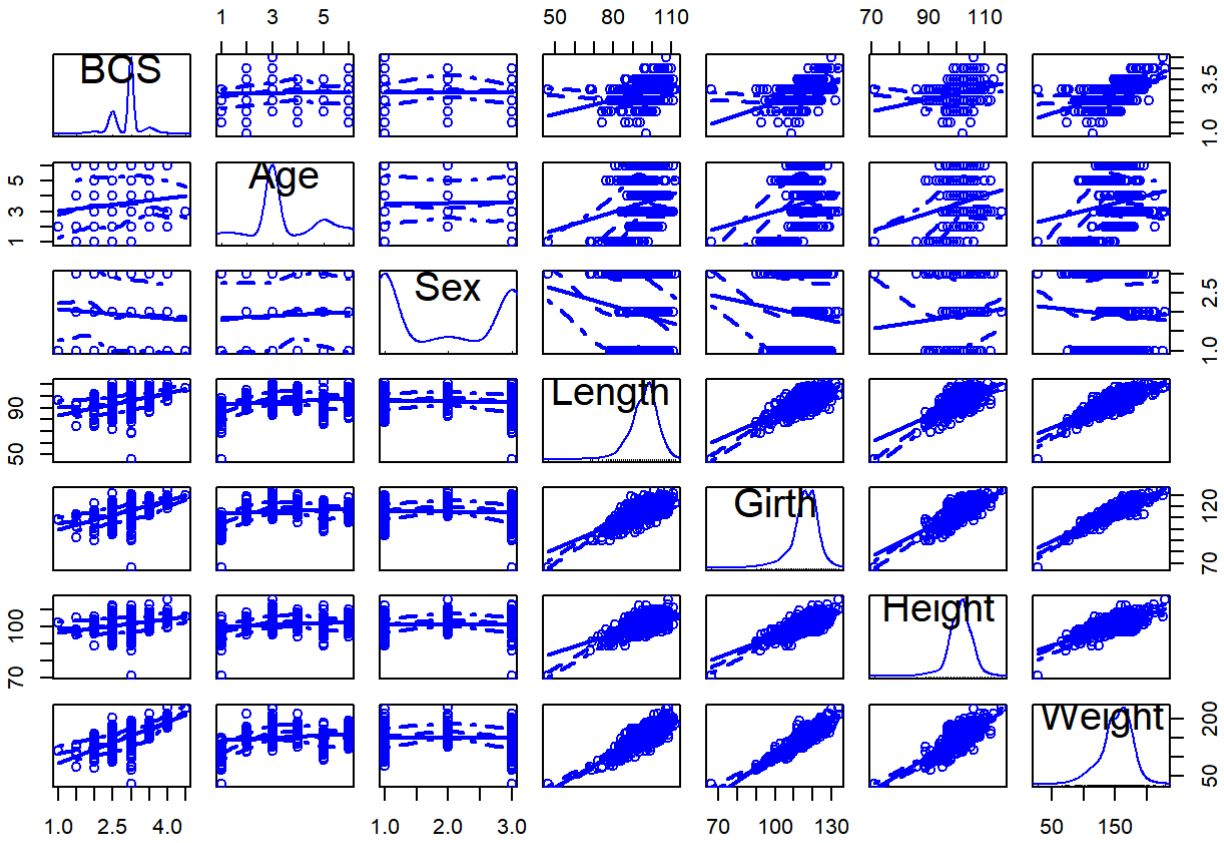
```
donkeys<-read.csv("Donkeys.csv")
```

1a - Remove the outlier

Plotting the variables

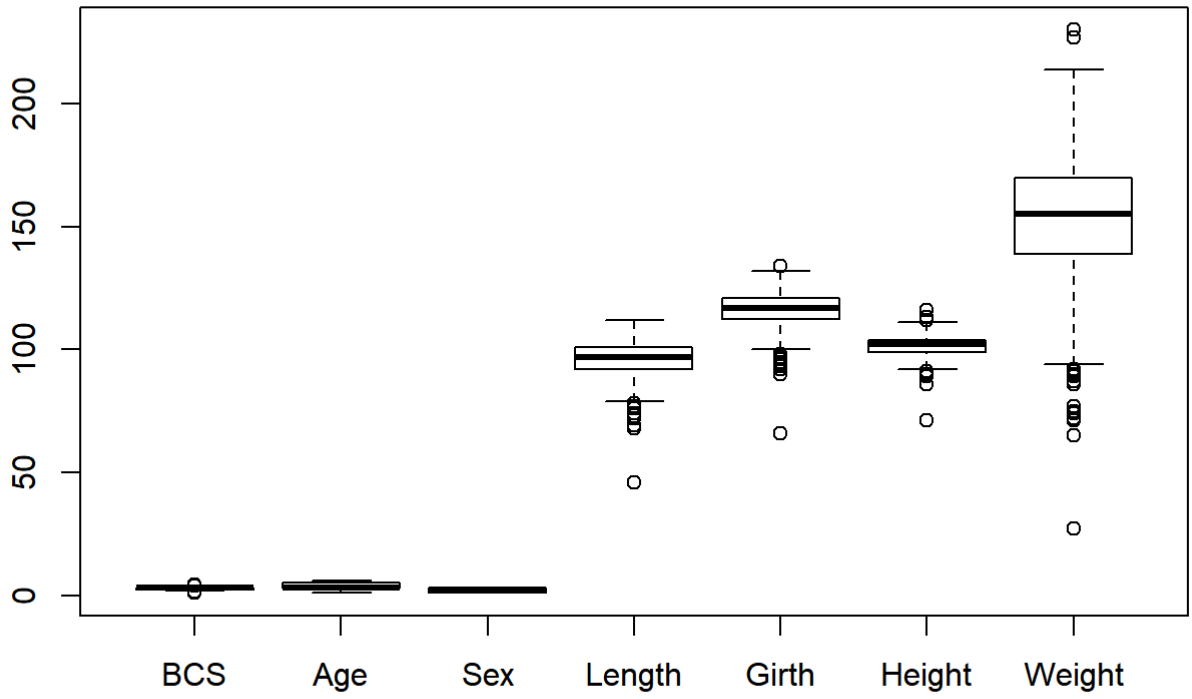
We can check the distribution of the variables using the Scatter plot. This will provide the variation in data with all the parameter.

```
library(car)
scatterplotMatrix(donkeys)
```

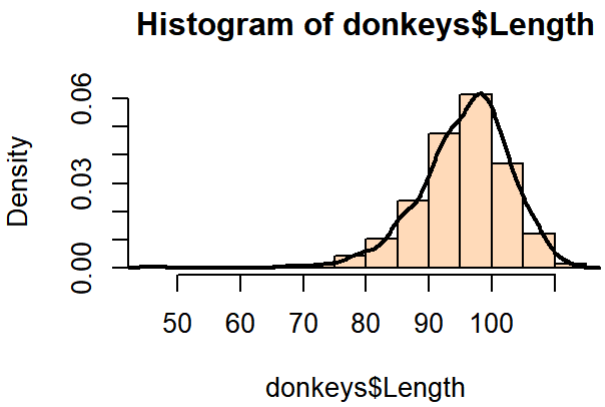


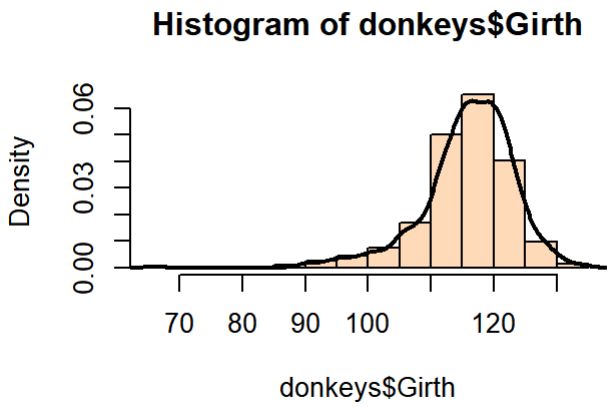
From the plot,we see that there is an outlying donkey which affects the dataset. We can cross verify it with several other plots.

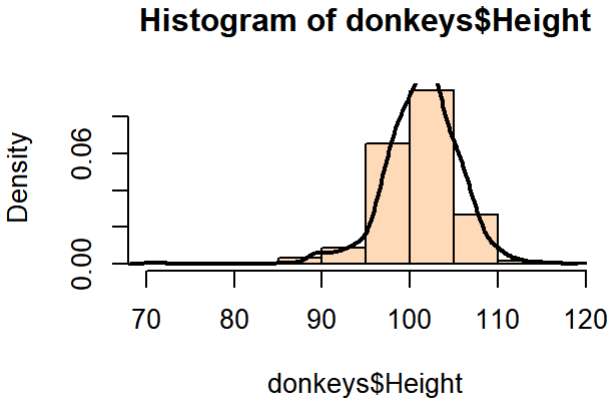
```
#Boxplot of Donkeys dataset
boxplot(donkeys)
```

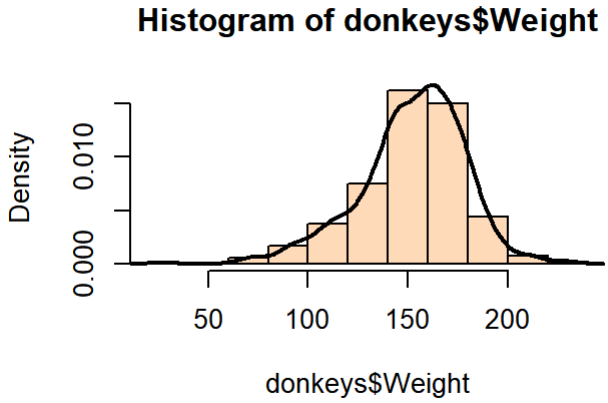



```
par(mfrow=c(2,2))
hist(donkeys$Length,prob = TRUE,col="peachpuff")
lines(density(donkeys$Length),lwd = 2)
hist(donkeys$Girth,prob = TRUE,col="peachpuff")
lines(density(donkeys$Girth),lwd = 2)
hist(donkeys$Height,prob = TRUE,col="peachpuff")
lines(density(donkeys$Height),lwd = 2)
hist(donkeys$Weight,prob = TRUE,col="peachpuff")
lines(density(donkeys$Weight),lwd = 2)
```

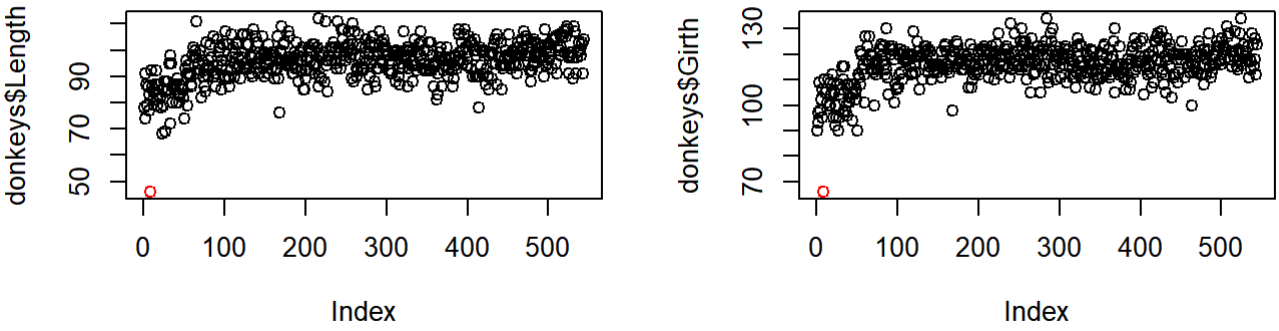


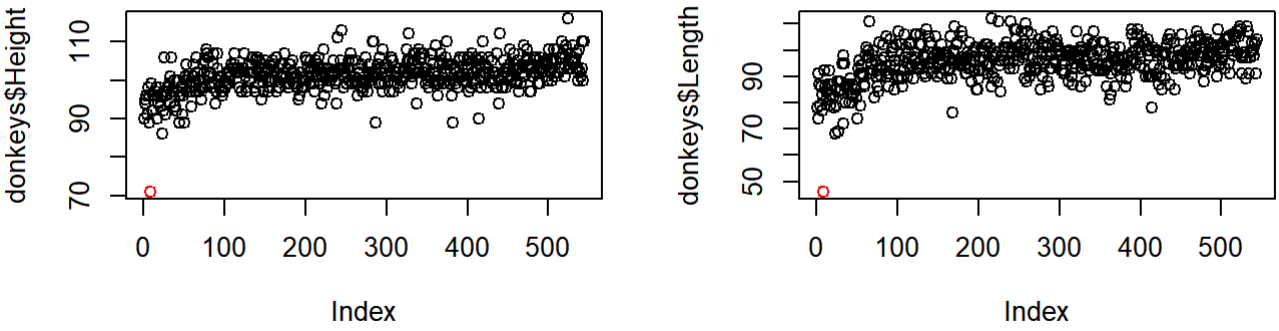







```
plot(donkeys$Length,col=ifelse(donkeys$Length==donkeys$Length[which(donkeys$Length==min(boxplot
(donkeys$Length,plot = FALSE)$out))], "red", "black"))
plot(donkeys$Girth,col=ifelse(donkeys$Girth==donkeys$Girth[which(donkeys$Girth==min(boxplot(don
keys$Girth,plot = FALSE)$out))], "red", "black"))
plot(donkeys$Height,col=ifelse(donkeys$Height==donkeys$Height[which(donkeys$Height==min(boxplot
(donkeys$Height,plot = FALSE)$out))], "red", "black"))
plot(donkeys$Length,col=ifelse(donkeys$Length==donkeys$Length[which(donkeys$Weight==min(boxplot
(donkeys$Weight,plot = FALSE)$out))], "red", "black"))
```



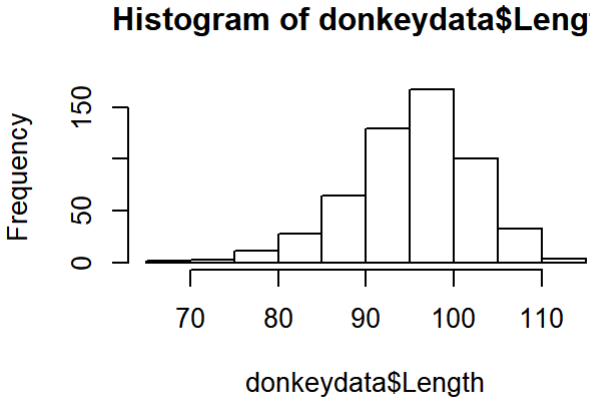


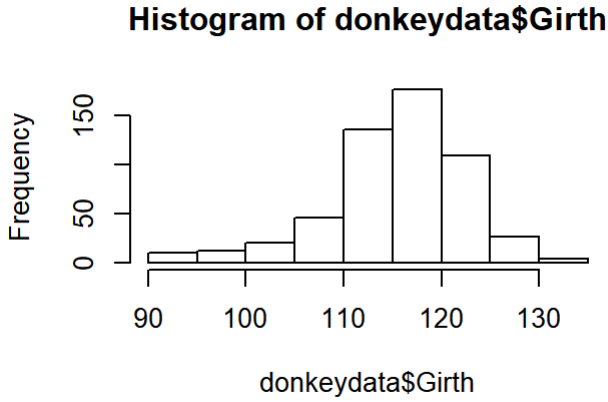
From these plot, we see that there is a donkey with a low parameters which is unusual compared with other data.So, we will remove the outlier from the dataset for better visualization of the data.

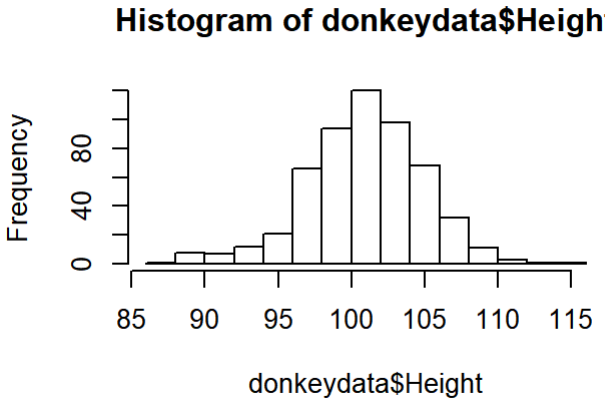
```
# From the above plot, we outlier is the lowest value in the data set. So,we will find the row
 number and remove the corresponding row.
which(donkeys$Weight==min(boxplot(donkeys$Weight,plot = FALSE)$out))
```

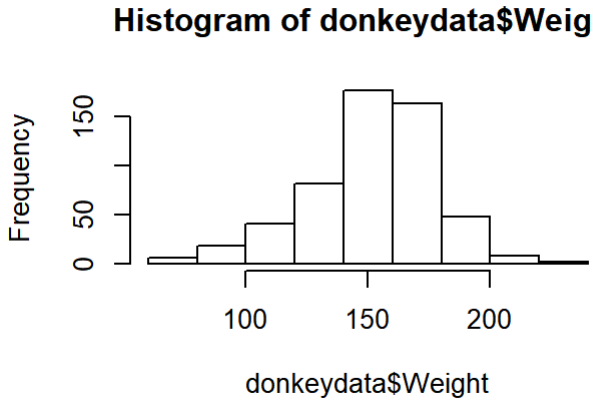
```
## [1] 9
```

```
donkeydata<-donkeys[-which(donkeys$Weight==min(boxplot(donkeys$Weight,plot = FALSE)$out)),]

par(mfrow=c(2,2))
hist(donkeydata$Length)
hist(donkeydata$Girth)
hist(donkeydata$Height)
hist(donkeydata$Weight)
```









1b - A principal components analysis is employed to reduce the dimension of the donkey data. Which variables should be used in such an analysis?

We will be using the numeric data for the Principal Component Analysis for reducing the dimension of the dataset,neglecting the categorical variables. So, from this dataset, we will use the Length,Girth,Height and Weight of donkeys ignoring the values of BCS,Age and Sex as these are categorical variables.

1c- Would you advise performing principal components analysis on the correlation matrix or covariance matrix of the appropriate donkey variables? Explain you reasoning.

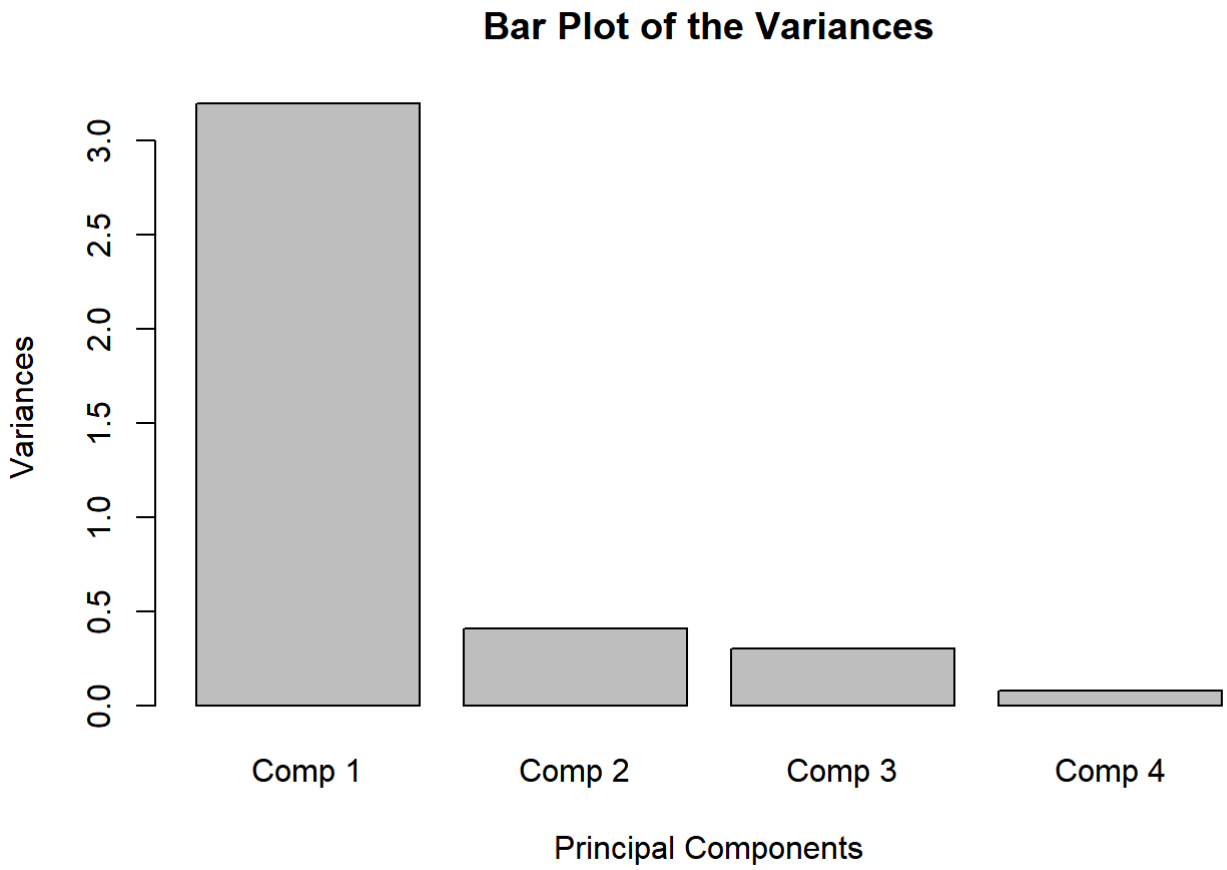
From the data, we see that the variables Length,Girth and Height are in same scale (cm) where as Weight is in different scale (kg). So, we have to standardize the dataset and then perform Principal Component Analysis on the correlation matrix of the donkey dataset. This is because standardizing each of the variables centers the dataset to mean 0 and standard deviation 1 and then correlation matrix is carried out for calulating the PCA with required eigen values and eigen vectors. So, we should use the covariance matrix when the variable scales are similar and the correlation matrix when variables are on different scales.

1d - Write your own function that could be used to apply principal components analysis to a multivariate data set. You should not use any inbuilt PCA functions that are available in R, but should derive the method from first principles and write your own code to implement the method accordingly. Your function should output objects that would be of interest to someone using your function.

The Principal Component Analysis can be used as a dimension reduction technique or as a method for identifying associations among variables. This method is mostly used in a situation when there are lots of data with redendency in it. Huge amount of variables can make the dataset un interpretable. Also, it might have duplicate values which might be derived from other variables. So, these dataset are required to be reduced before modeling or before being used for any analysis. An example of this dimension reduction method is Image processing. The pixel dimensions of an image can be reduced for compressing it to a less resolution format,although its structure remains unchanged.

-The PCA function can be written using the correlation or covariance matrix.
-The matrix is then used for calculating the eigen values and eigen vectors.
-The matrix can also be used in Singular Value Decomposition.
-The values are then further used for producing the objects of the PCA and fix_sign is enabled for fixing the sign of the loading and scores which is then used with Jackknife for PCA validation.
This reduction technique is carried out on the numerical variables of the dataset which produces standard deviations,loadings,scores,variances and proportion of variances.

```
#Function for calculating  principal components analysis to a multivariate data set
princompFunc<-function(X){
dataset<-X

#Calculating the Correlation matrix
m<-apply(dataset,2,mean)
dataset<-sweep(dataset,2,m,"-")

s<-apply(dataset,2,sd)
dataset<-sweep(dataset,2,s,"/")

R<-(t(as.matrix(dataset))%*%as.matrix(dataset))/dim(dataset)[1]

#Calculating the Eigen Values and Eigen vectors
eigendata<-eigen(R)
eigenvectors<-t(eigendata$vectors)

#newdataset
newData<-eigenvectors %*% t(dataset)

#variables
sdev<-sqrt(eigendata$values)
center<-m
loadings<-t(round(eigenvectors,3))
scores<-t(newData)
pcanames<-c()
# for Fix_sign
for(i in 1:dim(newData)[1]){
  if(i%%2!=0){
    loadings[,i]<- (-1)*loadings[,i]
    scores[,i]<-(-1)*scores[,i]
  }

  pcanames<-c(pcanames,paste('Comp',i))

}
names(sdev)<- c(pcanames)
colnames(loadings)<- c(pcanames)
rownames(loadings)<-colnames(dataset)
colnames(scores)<- c(pcanames)
summary.variance<-rbind(sdev,round(sdev^2/sum(sdev^2),7),cumsum(sdev^2/sum(sdev^2)))
rownames(summary.variance)<-c('Standard deviation','Proportion of Variance','Cumulative Proport
ion')
functiondata<-list(sdev=sdev,center=center,loadings=loadings,scores=scores,summary.variance=sum
mary.variance)
return(functiondata)
}

#Saving the values returned from the function to a variable
PCAvalues<-princompFunc(donkeydata[4:7])

screeplot(PCAvalues,xlab="Principal Components", main = "Bar Plot of the Variances")
```




```
PCAvalues$summary.variance
```

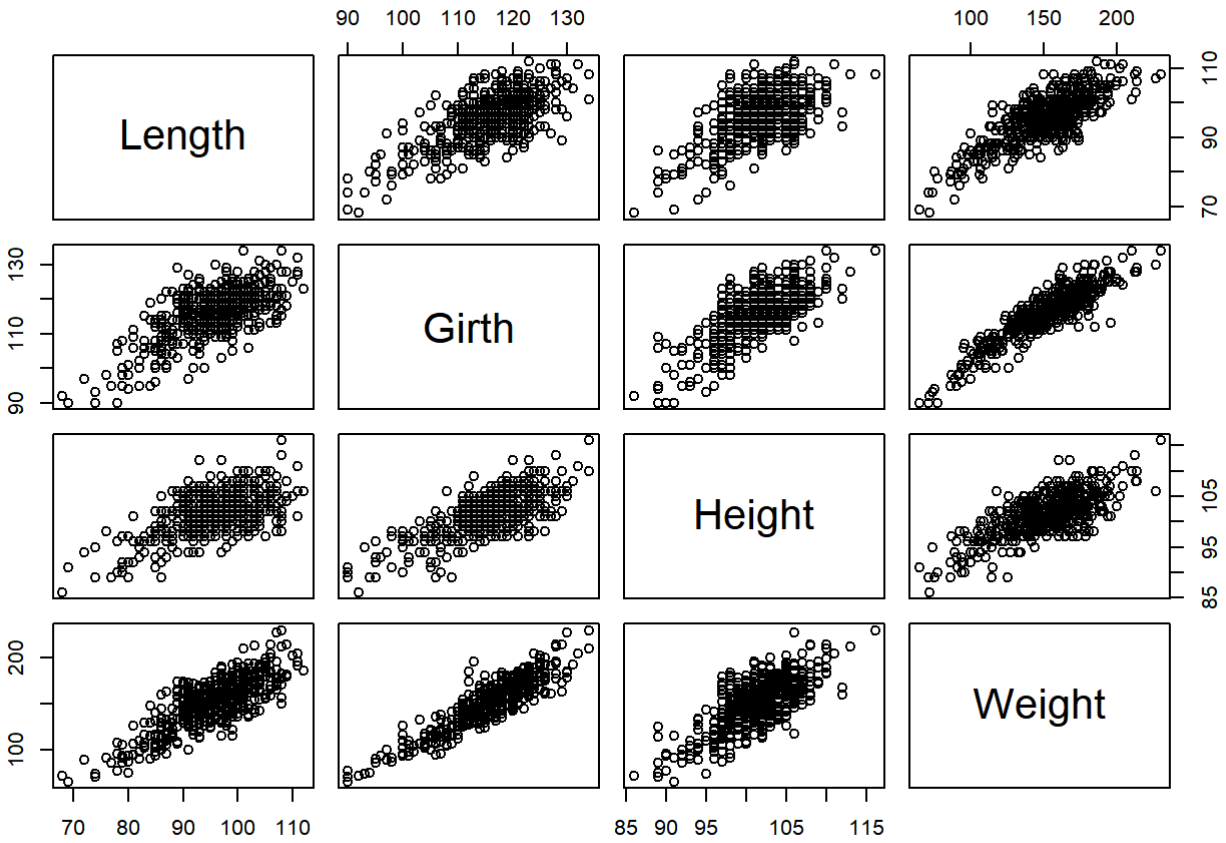
```
##                            Comp 1    Comp 2    Comp 3    Comp 4
## Standard deviation     1.7885844 0.6408011 0.5506744 0.2823671
## Proportion of Variance 0.8012341 0.1028459 0.0759504 0.0199696
## Cumulative Proportion  0.8012341 0.9040800 0.9800304 1.0000000
```

From the above plots of variation of the Principal components,we can see that the first principal component explains about 80% of variation and 10% of variation by PC2.This shows that the components one and two are enough to summarise the donkey dataset.

1e - Set the seed in R to your student number. Randomly sample 5 values between 1 and 500, and remove the corresponding donkeys from the data set. All subsequent analyses in question 1 should be conducted on this version of the dataset. From the output of the application of your own PCA function to the appropriate variables, how many principal components are required to summarise the donkey data? Use suitable plot(s) to motivate your decision.

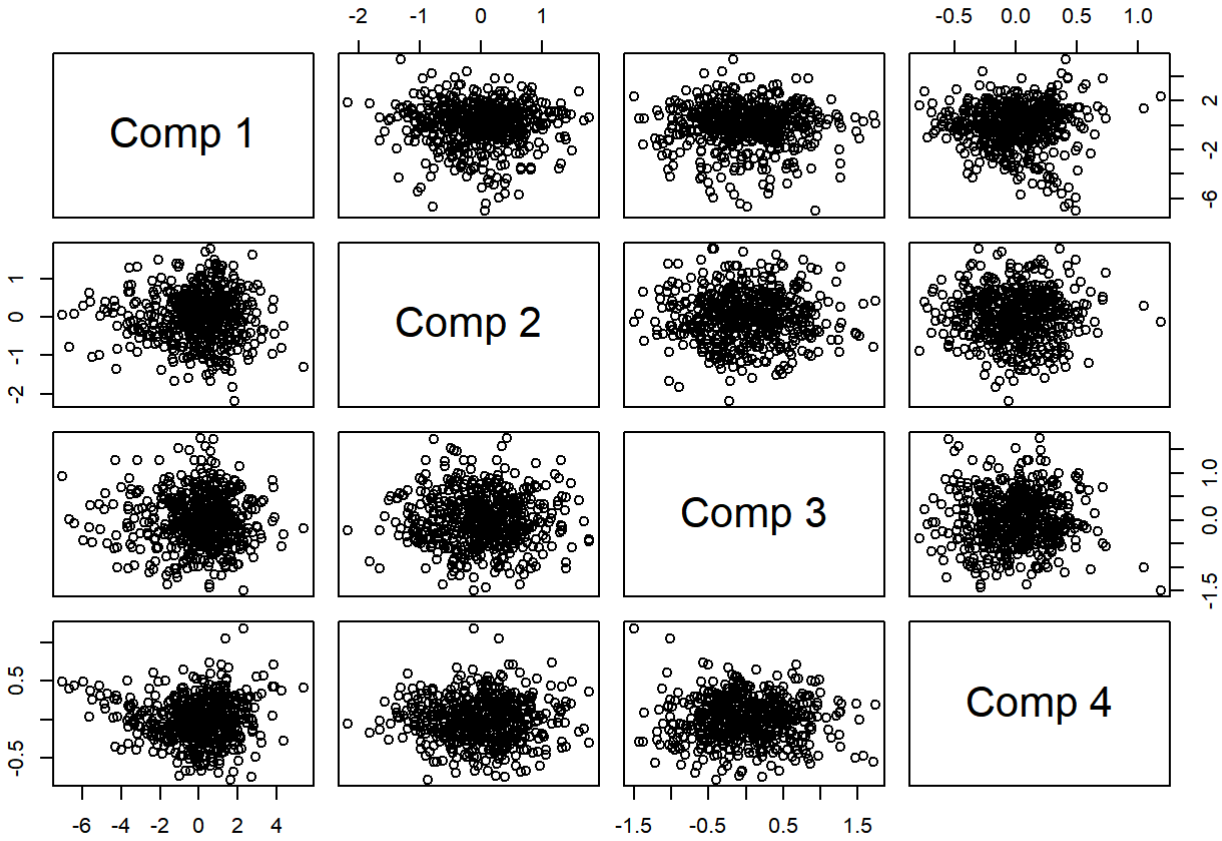
After adding the seed, five random sample values are saved to teh sampleval and the corresponding rows are removed from the dataset. This dataset is then again used in the PCA function created in part D and required principla componen analysis is carried out.

```
set.seed(19200349)
sampleval<- sample(1:500,5)
newDataset<-donkeydata[-sampleval,]
plot(newDataset[4:7])
```



From the plot, we see that the values of these four numerical variables - Length, Girth,Height and Weight are highly correlated. Correlation indicates that there is redundancy in the data. Due to this redundancy, PCA can be used to reduce the original variables into a smaller number of new variables(principal components), explaining most of the variance in the original variables.

```
PCAvalues1<-princompFunc(newDataset[4:7])
pairs(PCAvalues1$scores)
```



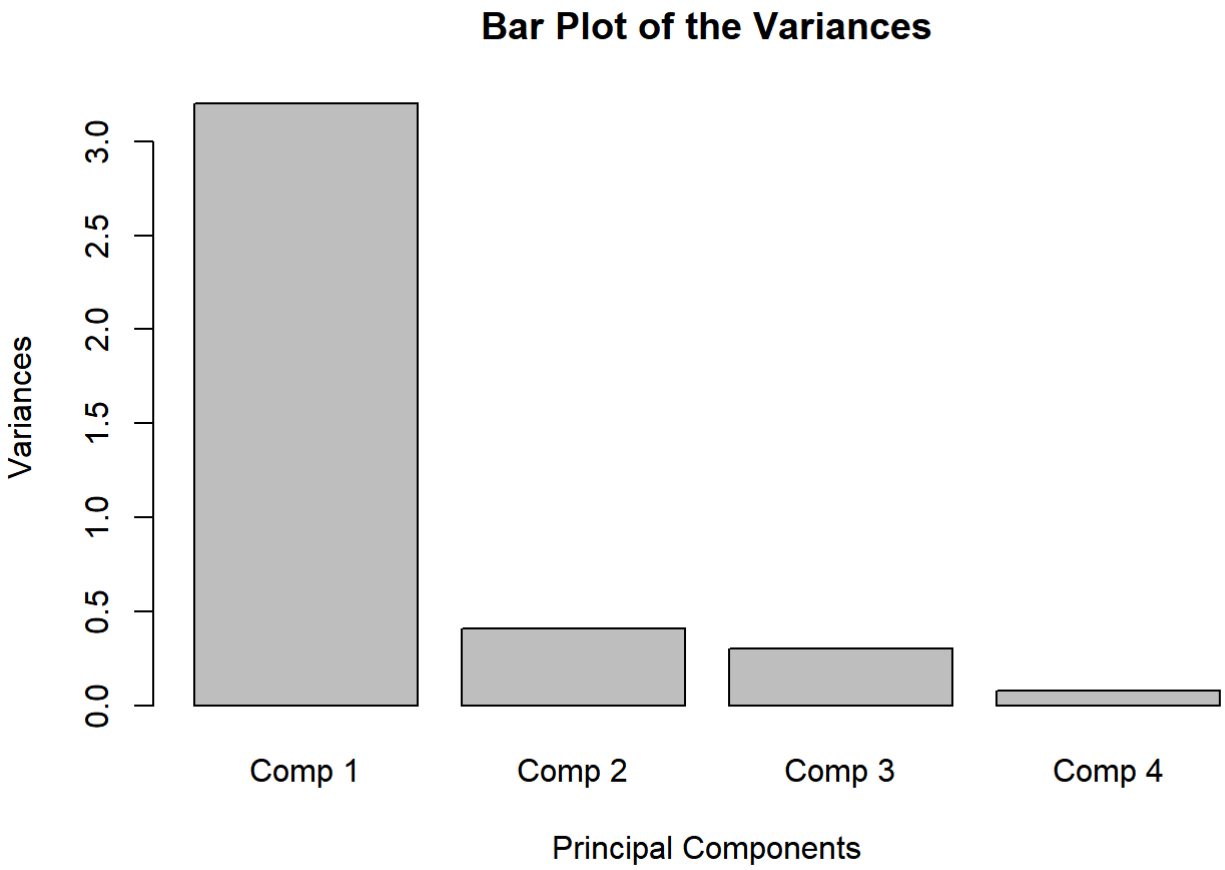
After the Principal Component Analysis, we see that the dimensionnality of the data is reduce by removing the noise and redundancy in the data.The datapoints are uncorrelated with each other. The correlation table shows the values in exp(-15) which confirms that the variables are highly uncorrelated.

```
cor(PCAvalues1$scores)
```

```
##                Comp 1        Comp 2        Comp 3        Comp 4
## Comp 1  1.000000e+00  1.175854e-15 -1.735859e-15 -1.692501e-15
## Comp 2  1.175854e-15  1.000000e+00  5.154496e-17 -6.177671e-15
## Comp 3 -1.735859e-15  5.154496e-17  1.000000e+00 -6.024041e-15
## Comp 4 -1.692501e-15 -6.177671e-15 -6.024041e-15  1.000000e+00
```

The eigenvalues measure the amount of variation retained by each principal component. Eigenvalues are large for the first PCs and small for the subsequent PCs. That is, the first PCs corresponds to the directions with the maximum amount of variation in the data set.

```
screeplot(PCAvalues1,xlab="Principal Components", main = "Bar Plot of the Variances")
```




```
#Printing Standard Deviation
print(PCAvalues1$sdev^2)
```

```
##     Comp 1     Comp 2     Comp 3     Comp 4
## 3.20157313 0.41077353 0.30063608 0.07958232
```

The variance or Eigenvalue of the first Principal Component is the largest and accounts to direction of the maximum variation in the dataset. We can confirm this by checking the variation proportion of the components.

```
print("Importance of components:")
```

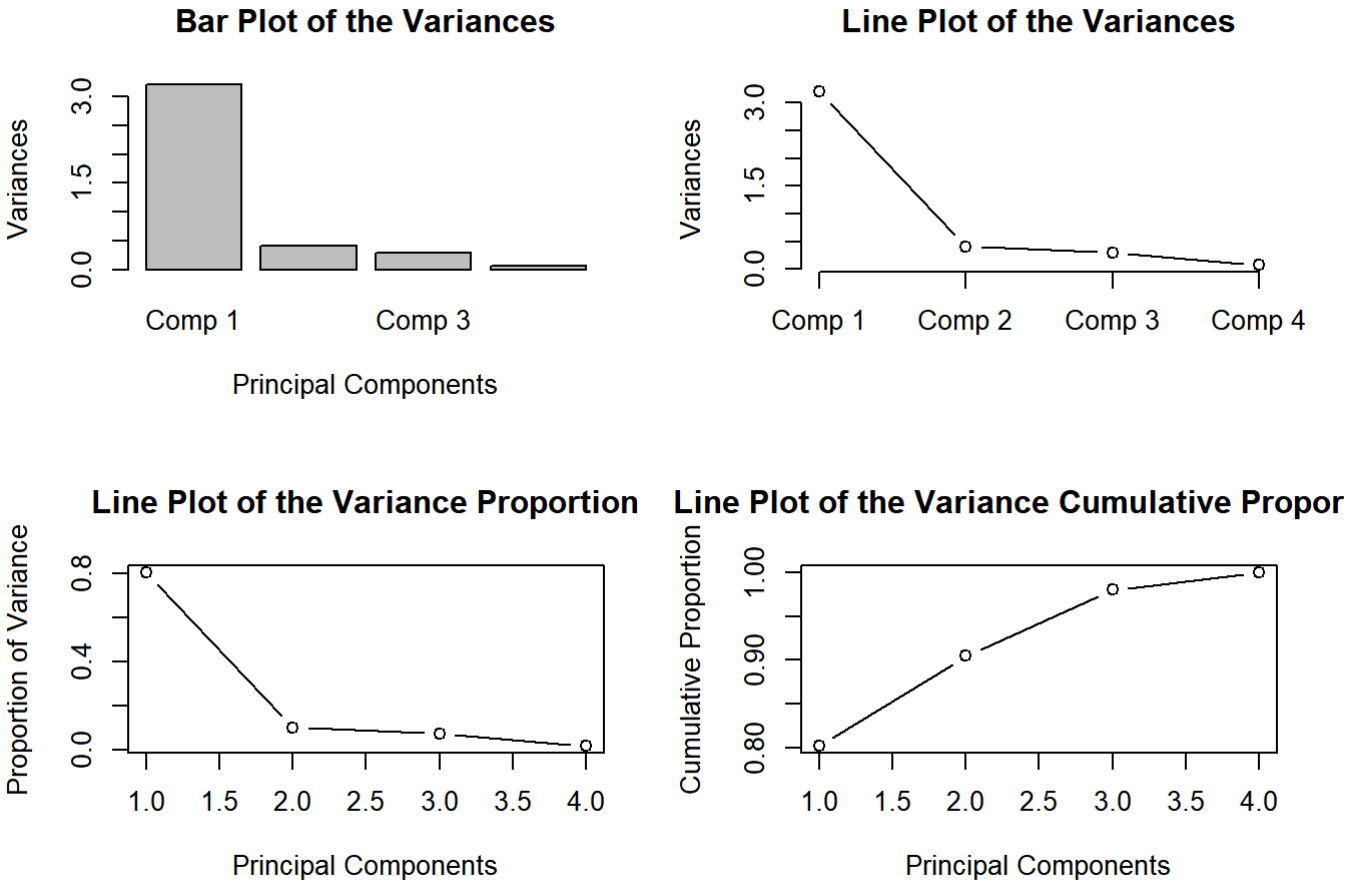
```
## [1] "Importance of components:"
```

```
print(PCAvalues1$summary.variance)
```

```
##                           Comp 1    Comp 2    Comp 3    Comp 4
## Standard deviation     1.7892940 0.6409162 0.5483029 0.2821034
## Proportion of Variance 0.8018838 0.1028846 0.0752990 0.0199326
## Cumulative Proportion  0.8018838 0.9047684 0.9800674 1.0000000
```

The variation proportion of the first component is 80% compared to the other components. So, Principal Component 1 is the largest and explains 80% of the variations in the dataset. The **first and second principal component comprises of ninety percentage of variation of the dataset**.

```
par(mfrow = c(2, 2))
screeplot(PCAvalues1,xlab="Principal Components", main = "Bar Plot of the Variances")
screeplot(PCAvalues1,type="lines",main = "Line Plot of the Variances")
plot(PCAvalues1$summary.variance[2,],type = "b",xlab = "Principal Components",ylab = "Proportio
n of Variance", main = "Line Plot of the Variance Proportion")
plot(PCAvalues1$summary.variance[3,],type = "b",xlab = "Principal Components",ylab = "Cumulativ
e Proportion", main = "Line Plot of the Variance Cumulative Proportion")
```



From the above plots of variation of the Principal components,we can see that the first principal component explains about 80% of variation and 10% of variation by PC2.This shows that the components one and two are enough to summarise the donkey dataset.

1f - Interpret the first column of the loadings matrix resulting from the application of your PCA function to the appropriate variables from the modified donkeys data.

```
#plot(newDataset$Length)
#plot(PCAvalues1$scores)
#par(new=TRUE)
#plot(PCAvalues1$loadings[1,1],col="red")

PCAvalues1$loadings[,1]
```

```
## Length  Girth Height Weight
##  0.477  0.517  0.468  0.535
```

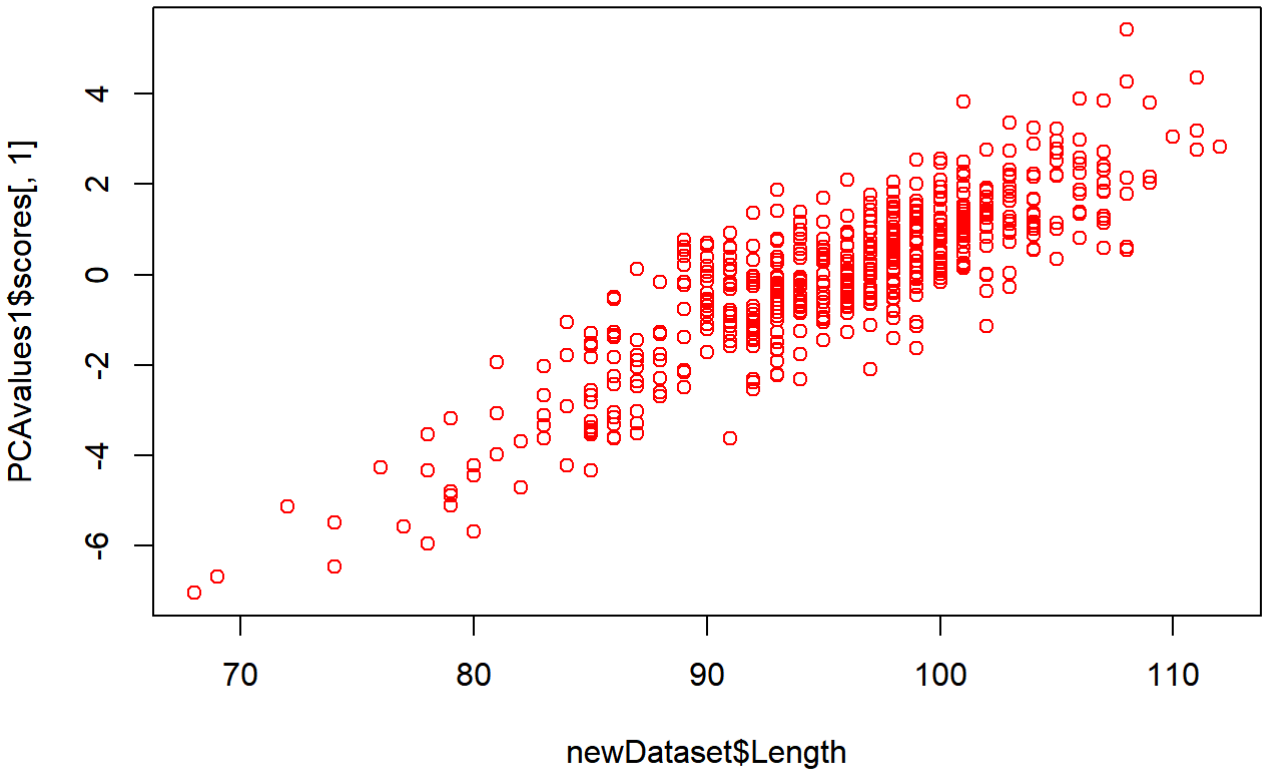
From the values of Principal Component 1, we see that the all four variables contribute to donkey characters. Out of the these four variables, weight variable contributes more compared with other variables.

1g - Plot the first principal component scores of the donkeys resulting from the application of your PCA function to the appropriate variables from the modified donkeys data (from 1(e)). Why is such a plot useful in PCA? Comment on the principal component scores in the context of the available data.

From part e, we came to the conclusion that the principal component 1 provides eighty percentage of variations in the dataset and first two principal component scores corresponds to 90% of the variation of dataset.

Now plotting the principal component 1 values with the donkey dataset, we can check the relationship between the data points and the scores, i.e, the score approximation of the respective dataset variables.

```
#Principal Component Score 1 with the Length variable of donkey dataset
plot(newDataset$Length,PCAvalues1$scores[,1],col="red")
```




```
cor(newDataset$Length,PCAvalues1$scores[,1])
```

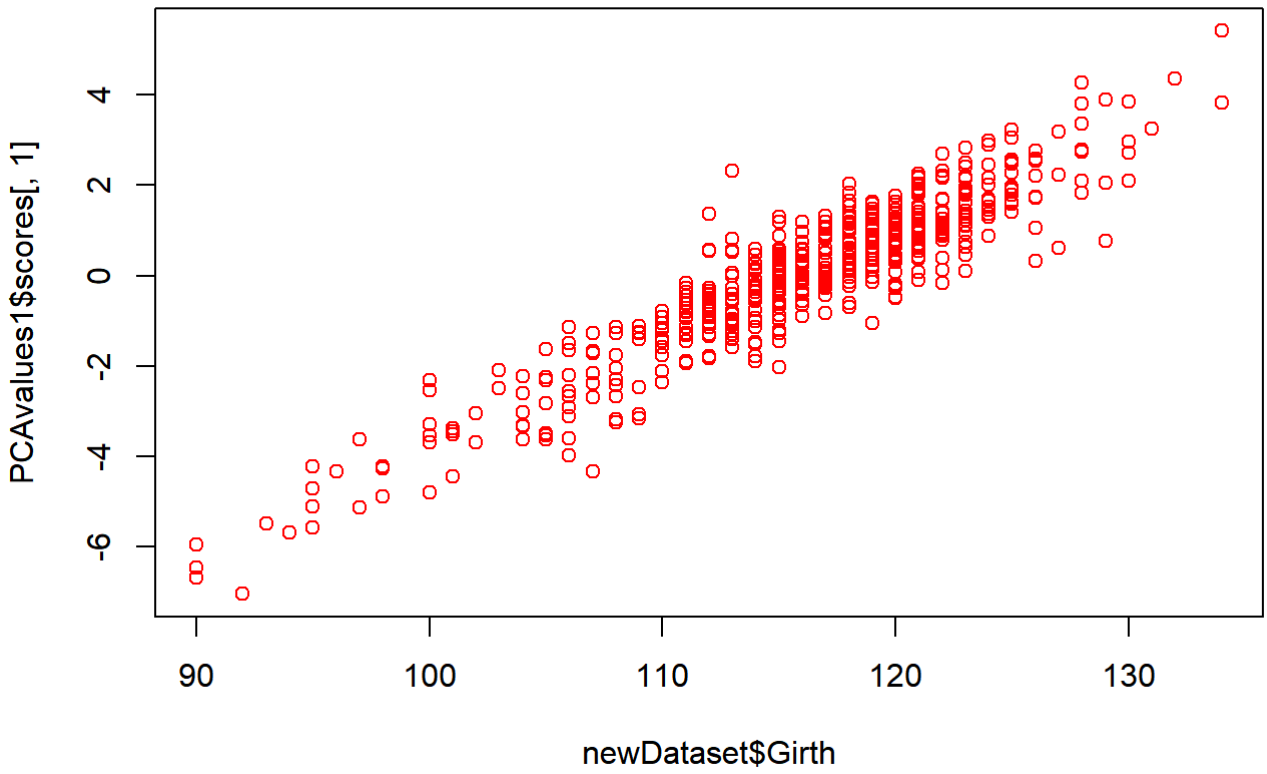
```
## [1] 0.8550794
```

From the above plot, we can see that as the Length variable of the donkey dataset increases, the principal score corresponding to the length also increases. This shows that there is a linear relationship between the original data variable and the corresponding score of the first principal component. There is a positive correlation among the variables. The correlation output is around 0.85 which shows that it is highly correlated with the variables.

For score with the other three numeric variables also shows a linear relationship among each other.

These plots are used for evaluating the relationships between the principal score of the first component with the dataset variables. These plots can also be used to verify if the principal components summarises the data points of the dataset.

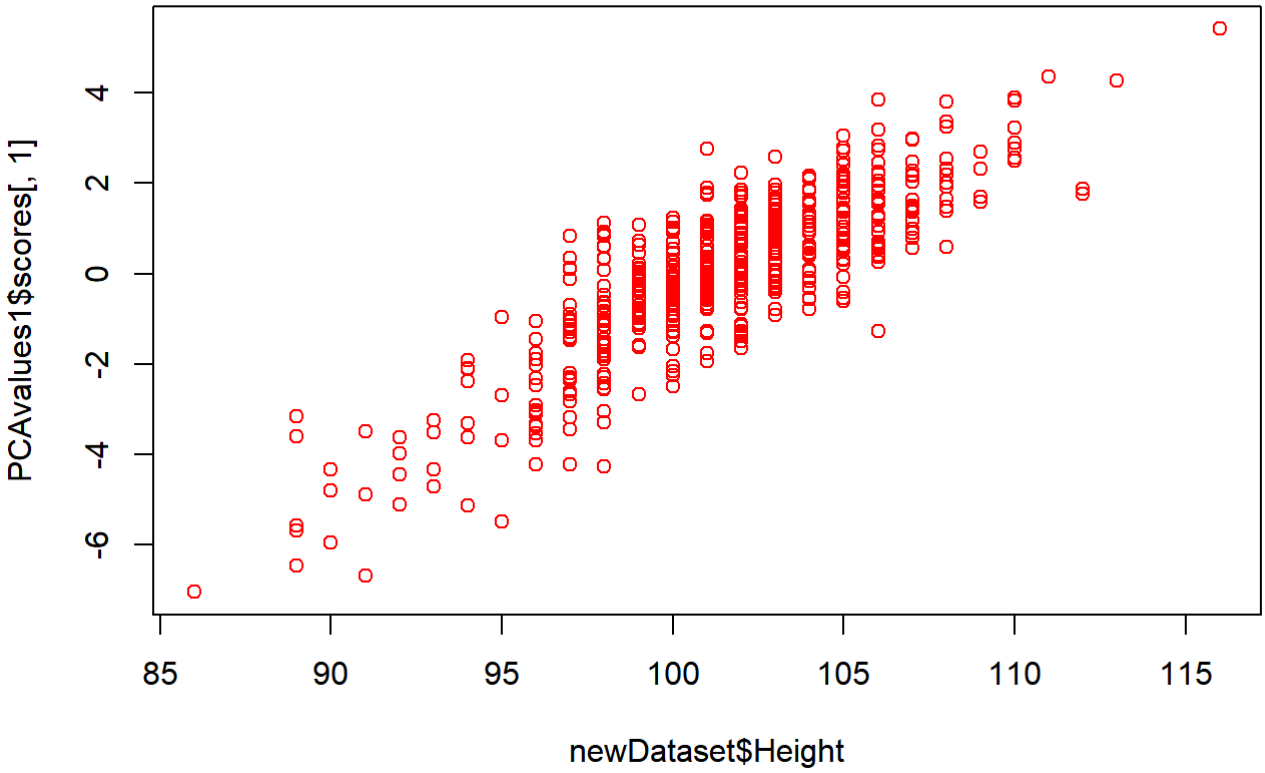
```
plot(newDataset$Girth,PCAvalues1$scores[,1],col="red")
```




```
cat("The correlation value of this plot is :",cor(newDataset$Girth,PCAvalues1$scores[,1]))
```

```
## The correlation value of this plot is : 0.9255787
```

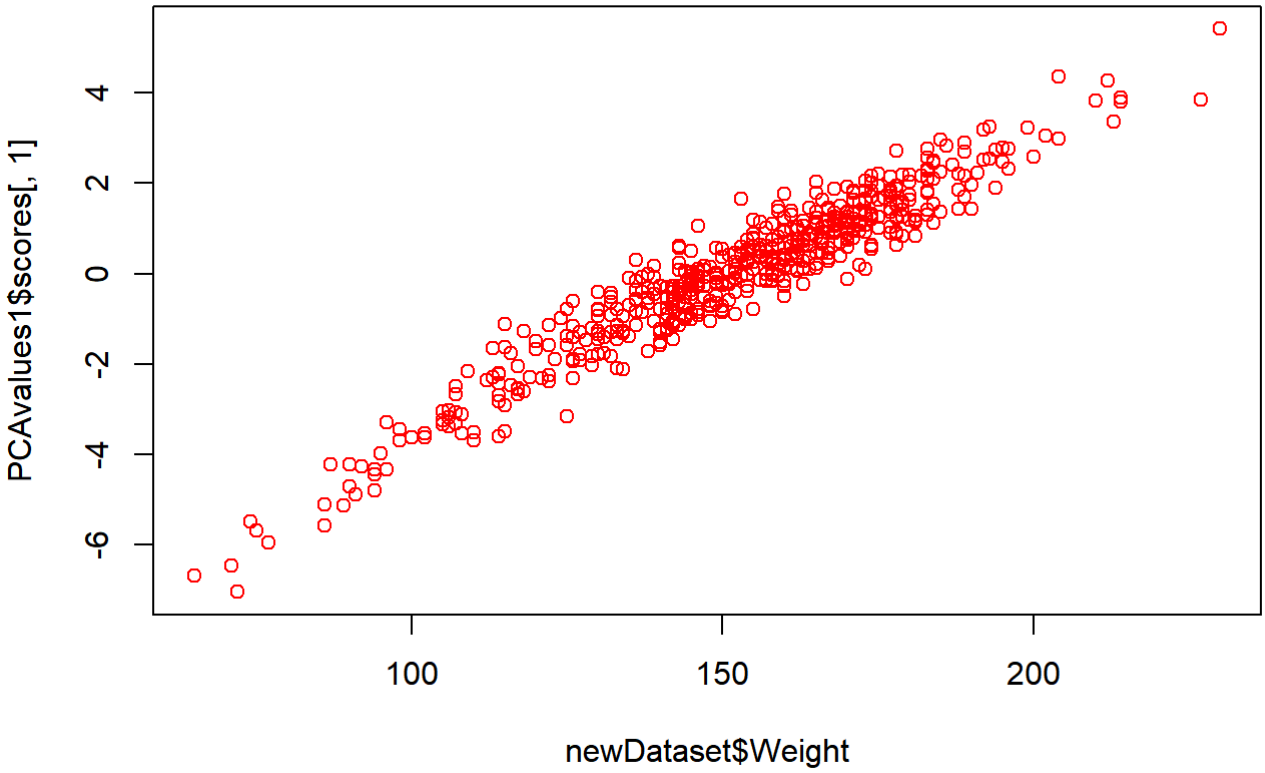
```
plot(newDataset$Height,PCAvalues1$scores[,1],col="red")
```




```
cat("The correlation value of this plot is :",cor(newDataset$Height,PCAvalues1$scores[,1]))
```

```
## The correlation value of this plot is : 0.8385734
```

```
plot(newDataset$Weight,PCAvalues1$scores[,1],col="red")
```




```
cat("The correlation value of this plot is :",cor(newDataset$Weight,PCAvalues1$scores[,1]))
```

```
## The correlation value of this plot is : 0.957326
```

From the above plots, we see that the variables of the dataset are linearly related to the principal component score.

This proves the property of PCA that the first principal component of the dataset is the linear combination of the variables which has the greatest variance.

1h - Jackknife Validation - Use your code to validate the results obtained from applying your PCA function to the appropriate variables from the modified donkeys data (from 1(e)).

Jackknife is a cross tabulation method using for validating the Principle Component Analysis data.This method is used for computing the variances. We will calculate the Pc using jackknife and it will be compared with th ecalculated highest eigen value suggesting that the first PC accounts for variance which is systematic and generalizable beyond the sample.

Jackknife validation can be carried out by repeated comparison of PCA values with calculated values of ‘left out’ data point and variances of both output is verified.The steps of Jackknife are,
-One observation is removed and PCA is applied to the remaining n-1 observations.
-The resulting PC1 is used to compute the values of PC1 for the ‘left out’ data point.
-The process is repeated for all n points in the sample,thus conducting a PCA analysis for all possible subsets of size n-1.
-The variance of the jackknifed PC1 values can be compared to the variance of the first PC determined from the entire set of n data points.

```
 stddata<-newDataset[4:7]

 #Standardizing the data before calculating the Jackknife on the dataset.
 m<-apply(stddata,2,mean)
 stddata<-sweep(stddata,2,m,"-")
 s<-apply(stddata,2,sd)
 stddata<-sweep(stddata,2,s,"/")

 #A variable for storing the calculated scores of jackknife method
 jackScores<-c()

 #the loop runs for n data points, leaving out one datapoint a a time and considering the rest
 for calculating the PCA.
 for(i in 1:nrow(newDataset)){

   #variable for storing the n-1 datapoints
   jackPCA<-princompFunc(newDataset[-i,4:7])

   #Jackknife scores are calculated by multiplying the leftout data with the loadings calculate
d using PCA on n-1 datapoints
   score<-as.matrix(stddata[i,])%*%jackPCA$loadings

   #The calculated score is binded to a variable
   jackScores<-rbind(jackScores,score)
 }

jackknifeVariance<-diag(var(jackScores))

cat("The variances of PC of the entire dataset are \n", PCAvalues1$sdev^2)
```

```
## The variances of PC of the entire dataset are
##  3.201573 0.4107735 0.3006361 0.07958232
```

```
cat("The variances of jackknife are \n", round(jackknifeVariance,6))
```

```
## The variances of jackknife are
##  3.20337 0.408199 0.304175 0.080637
```

```
cat("The first variance of Jackknife and Principal Component is ",jackknifeVariance[1]," & ",PC
Avalues1$sdev[1]^2," respectively.These values are approximately equal which proves jackknife v
alidation.")
```

```
## The first variance of Jackknife and Principal Component is  3.20337  &  3.201573  respective
ly.These values are approximately equal which proves jackknife validation.
```

From these values, we see that the variance of the jackknifed PC1values is approximately equal to the variances of the PC determined from the entire set of n data points,suggesting that the first PC accounts for variance which is systematic and generalizable beyond the sample.This validates the results obtained from the PCA function on the modified dataset.

Question 2

2a - Proof on Factor Analysis model.

Question 2. a) Show that $x_i \sim MVN_p(\mu, \Lambda\Lambda^T + \Psi)$.

The Factor Analysis is a statistical model which attempts to explain the correlation between a large set of variables interms of a small number of underlying factors.
This is typically used to draw inferences about unobservable quantities in the social sciences, eg. intelligence, musical ability, consumer attitudes, etc. Normally, FA is based on statistical model.

Under the factor analysis model, a p-dimensional observation x_i ($i = 1, \ldots, N$) is modeled as,

$$x_i = \mu + \Lambda f_i + \varepsilon_i$$

where,

$$f_i \sim MVN_q(0, I)$$
$$\varepsilon_i \sim MVN_p(0, \Psi)$$

with f_i & ε_i are assumed as independent & $q << p$.
& f is an underlying common factor,
ε are specific factors.

Under the orthogonal factor model, it is assumed that

$$\mathbb{E}[f_i] = \underline{0}, \quad Cov(f_i) = I$$
$$\mathbb{E}[\varepsilon_i] = \underline{0}, \quad cov(\varepsilon_i) = \Psi$$

The model is given by,

$$x_i = \mu + \Lambda f_i + \varepsilon_i$$
$$\mathbb{E}[x_i] = E[\mu] + \Lambda E[F_i] + E[\varepsilon_i]$$
$$= \mu + 0 + 0$$
$$\boxed{\mathbb{E}[x_i] = \underline{\mu}.}$$

The parsimonious covariance structure for the data x_i or $\underline{X}$,

$$\Sigma = Cov(\underline{x})$$

$$= \mathbb{E}[(x_i - \underline{\mu})(x_i - \underline{\mu})^T]$$

$$= \mathbb{E}[(\Lambda f_i + \varepsilon_i)(\Lambda f_i + \varepsilon_i)^T] \quad [\because x_i = \mu + \Lambda f_i + \varepsilon_i]$$

$$= \mathbb{E}[(\Lambda f_i)(\Lambda f_i)^T + \varepsilon_i(\Lambda f_i)^T + \Lambda f_i(\varepsilon_i)^T + \varepsilon_i\varepsilon_i^T]$$

$$\Sigma = \Lambda \mathbb{E}[f_i f_i^T]\Lambda^T + \mathbb{E}[\varepsilon_i f_i^T]\Lambda^T + \Lambda \mathbb{E}[f_i \varepsilon_i^T] + \mathbb{E}[\varepsilon_i \varepsilon_i^T]$$

Since f_i & ε_i are assumed as independent,

$$cov(f_i, \varepsilon_i) = \mathbb{E}[f_i \varepsilon_i^T] = \underline{O}$$

So,

$$\Sigma = \Lambda\Lambda^T + O + O + \Psi$$

$$\boxed{\Sigma = \Lambda\Lambda^T + \Psi}$$

The multivariate normal distribution of the p-dimensional observation x_i can be written as,

$$x_i \sim MVN_p(\underline{\mu}, \Sigma)$$

So, mean, $\mu \Rightarrow \mathbb{E}[x_i] = \underline{\mu}$

& covariance matrix, $\Sigma = \Lambda\Lambda^T + \Psi$.

$\therefore$ $x_i \sim MVN_p(\mu, \Lambda\Lambda^T + \Psi)$

2b - Describe why a factor rotation is often employed when fitting a factor analysis model.

The factor rotation is used for making the structure simpler to interpret, to minimize the complexity of the factor loading.The original loading from the function may not be readily interpretable. Thus it is usual practice to rotate them until a simpler structure is achieved. This is generally done by transforming the initial loadings by an orthogonal transformation,given the covariance matrix can still be reproduced. This orthogonal transformation corresponds to a rigid rotation or reflection of the coordinate axes. Another method of factor rotation is Oblique rotation. This is also called nonrigid rotation of the coordinate system i.e, the resulting axes need no longer be perpendicular. The degree of correlation allowed among factors is generally small. A difference between these two factor rotation methods is that the orthogonal rotation does not change the position of variables relative to each other in the space of the factors, i.e. correlations between variables are being preserved.Where as in Oblique rotation,factors are allowed to lose their uncorrelatedness if that will produce a clearer “simple structure”.

2c

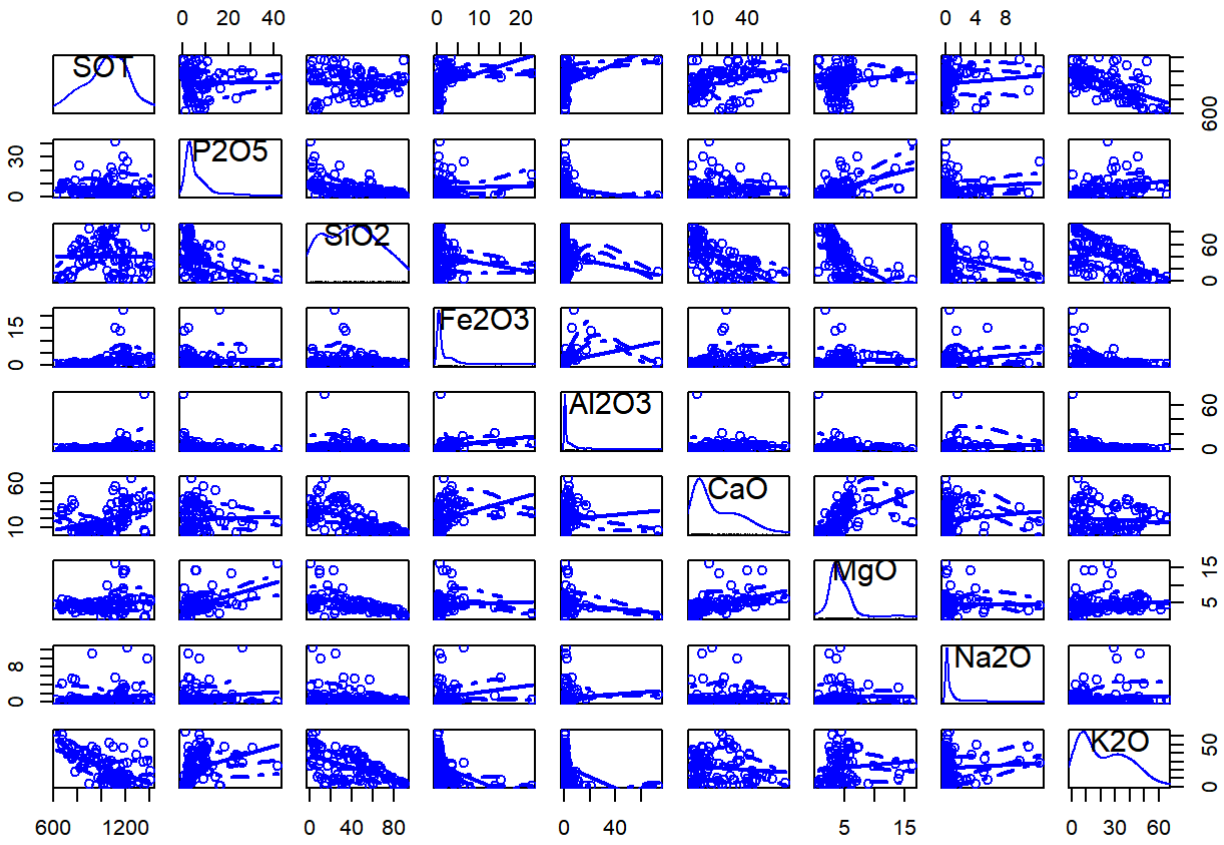
The biomasses data is loaded and five random values are dropped from the dataset for further analysis on the dataset.

```
ashdata<-read.csv("AshData.csv")
set.seed(19200349)
sampleashval<- sample(1:99,5)
ashdata1<-ashdata[-sampleashval,]
```

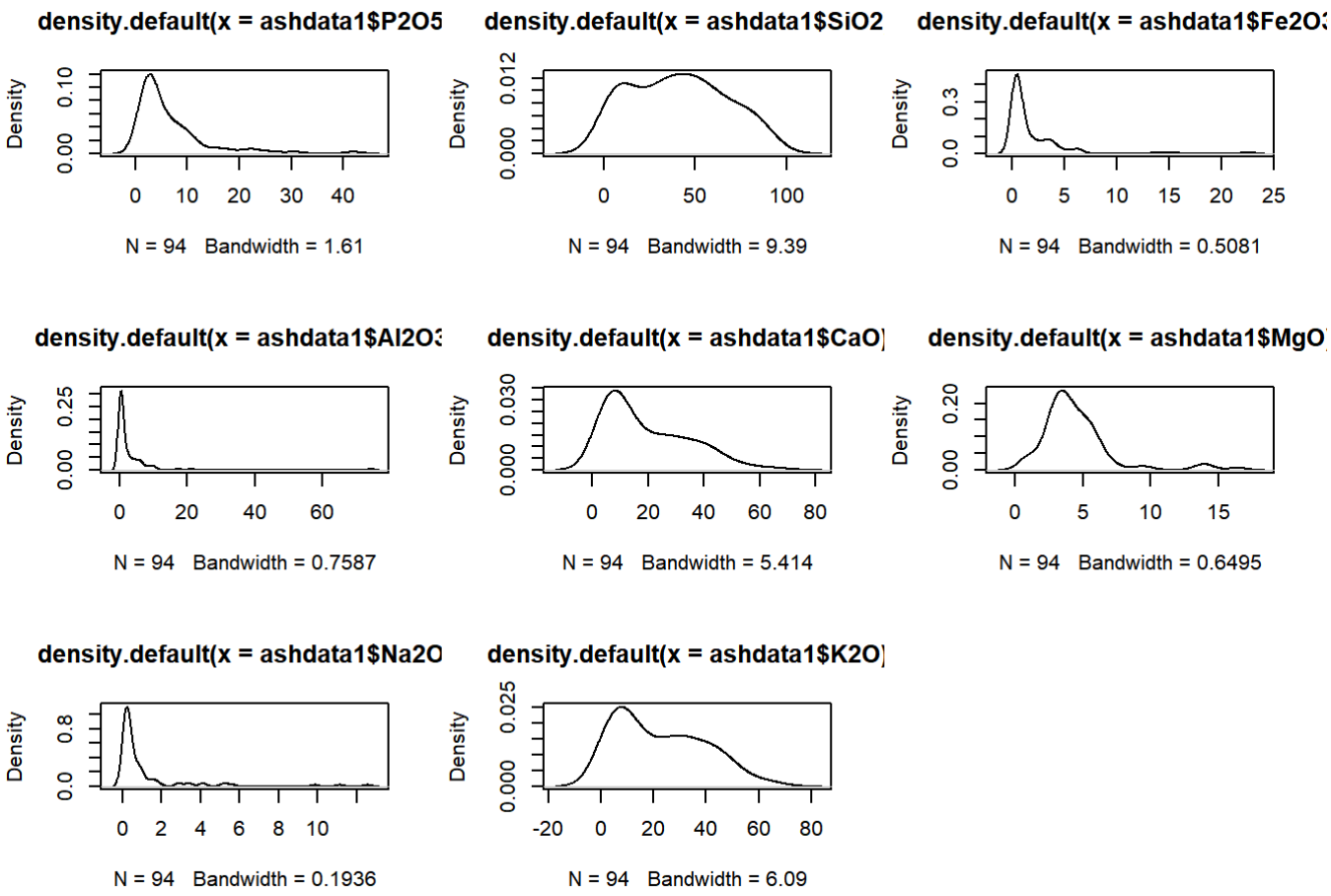
2cI - Plot the mass concentration data for the ash samples. Would you advise transforming the data prior to the application of factor analysis? Explain your reasoning.

From the concentrations of each elements, we see that the data points are recorded in different range. Few elements such as SiO2, CaO & Ka2O are recorded with values closer to each other where as elements such as P2O5 & Fe2O3 are recorded with values of far range. So, when these values are plotted, there will be sharp skew in the graph which can be seen in the scatterplotmatrix shown below. By this initial analysis of records, we can consider transforming the data before use. For confirming this, we can plot the values and compare the datapoints to see if the data needs to be transformed.

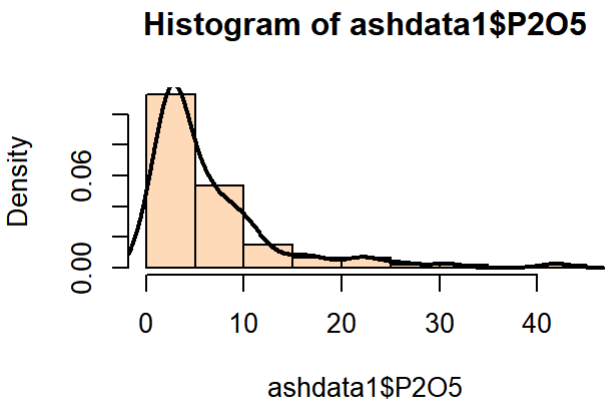
```
scatterplotMatrix(ashdata1)
```

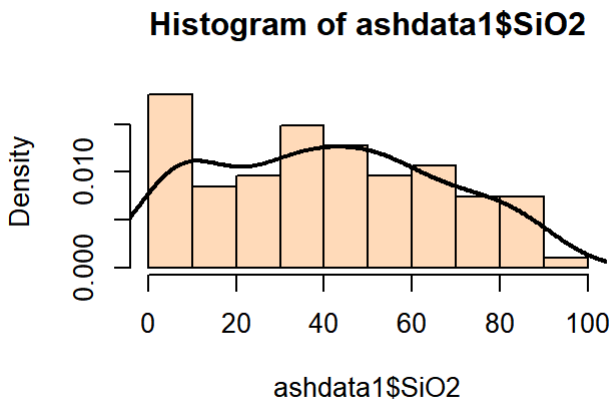


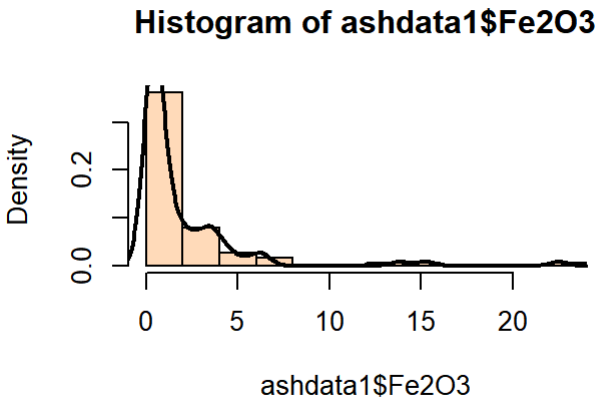

```
par(mfrow=c(3,3))
plot(density(ashdata1$P2O5))
plot(density(ashdata1$SiO2))
plot(density(ashdata1$Fe2O3))
plot(density(ashdata1$Al2O3))
plot(density(ashdata1$CaO))
plot(density(ashdata1$MgO))
plot(density(ashdata1$Na2O))
plot(density(ashdata1$K2O))


par(mfrow = c(2, 2))
```

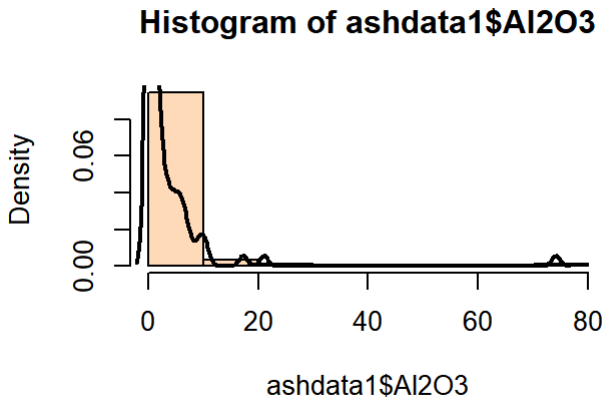



```
hist(ashdata1$P2O5,prob = TRUE,col="peachpuff")
lines(density(ashdata1$P2O5),lwd = 2)
hist(ashdata1$SiO2,prob = TRUE,col="peachpuff")
lines(density(ashdata1$SiO2),lwd = 2)
hist(ashdata1$Fe2O3,prob = TRUE,col="peachpuff")
lines(density(ashdata1$Fe2O3),lwd = 2)
hist(ashdata1$Al2O3,prob = TRUE,col="peachpuff")
lines(density(ashdata1$Al2O3),lwd = 2)
```

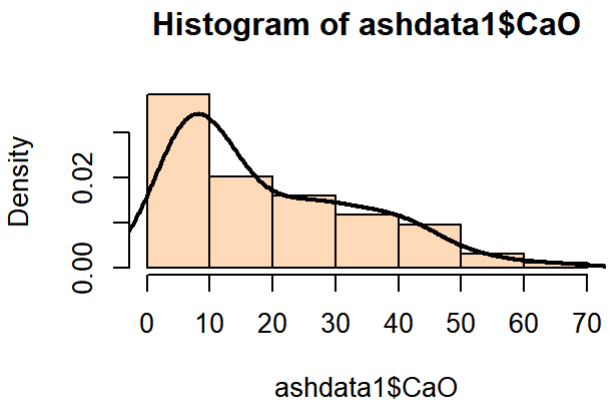


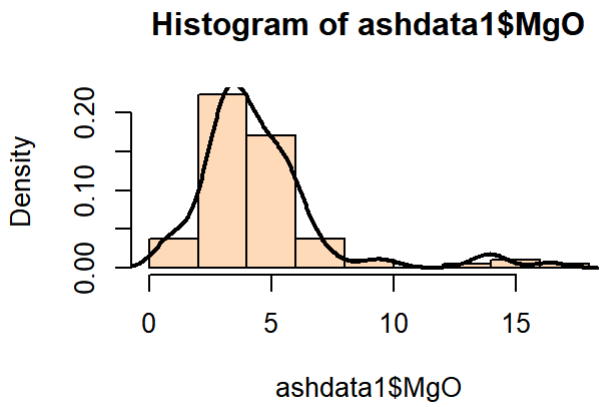


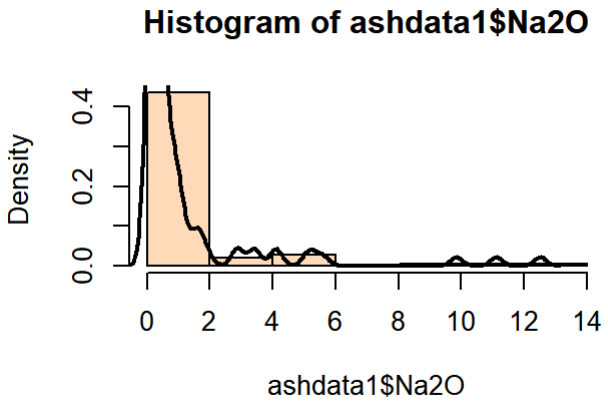


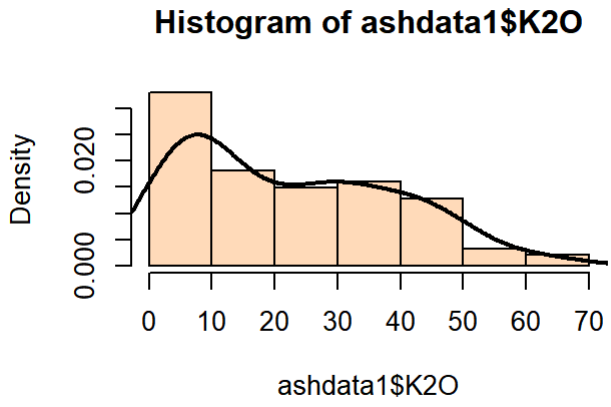



```
hist(ashdata1$CaO,prob = TRUE,col="peachpuff")
lines(density(ashdata1$CaO),lwd = 2)
hist(ashdata1$MgO,prob = TRUE,col="peachpuff")
lines(density(ashdata1$MgO),lwd = 2)
hist(ashdata1$Na2O,prob = TRUE,col="peachpuff")
lines(density(ashdata1$Na2O),lwd = 2)
hist(ashdata1$K2O,prob = TRUE,col="peachpuff")
lines(density(ashdata1$K2O),lwd = 2)
```



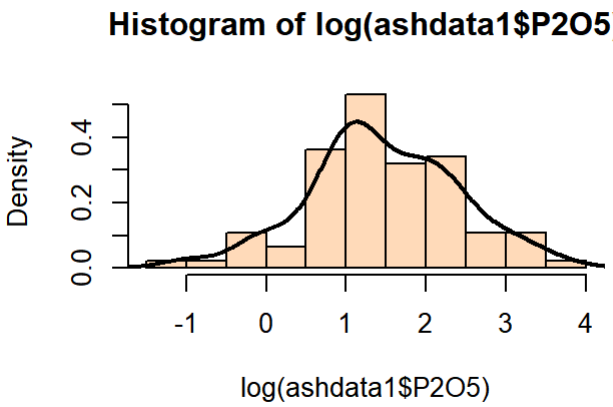


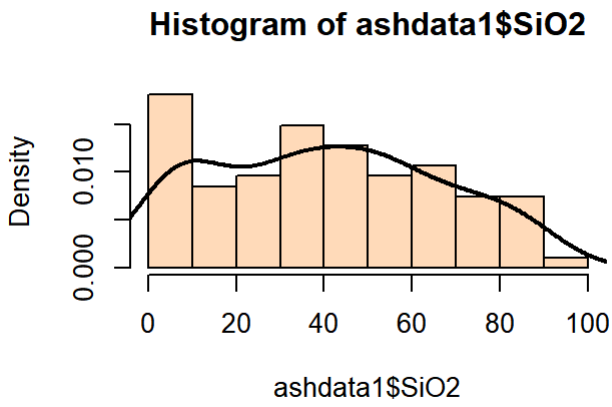


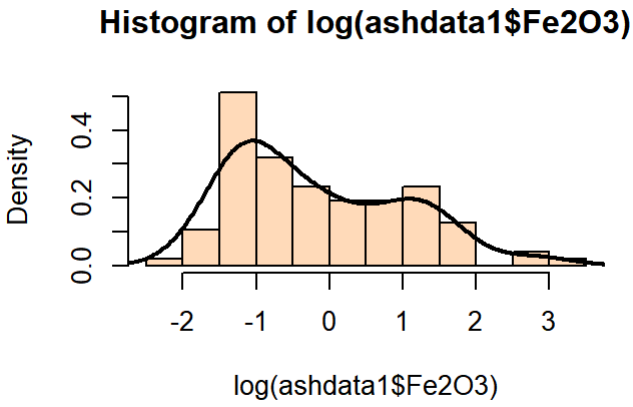


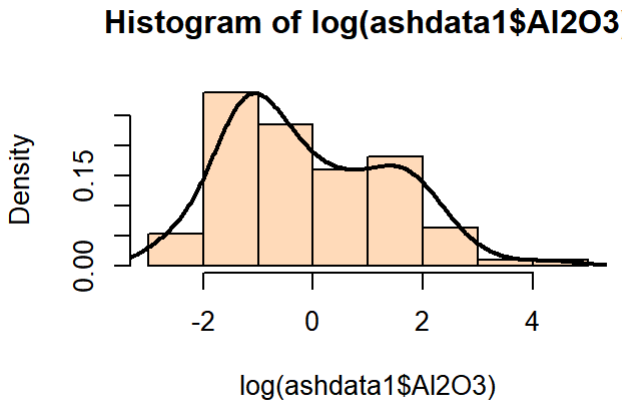
From the plots we see that there is a positive skew for most of the elements which is a tell-tale sign that the data needs to be transformed. This dataset needs to be transformed for changing it to be in a normally distributed form with less skewness on the values.

```
par(mfrow = c(2, 2))
hist(log(ashdata1$P2O5),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$P2O5)),lwd = 2)
hist(ashdata1$SiO2,prob = TRUE,col="peachpuff")
lines(density(ashdata1$SiO2),lwd = 2)
hist(log(ashdata1$Fe2O3),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$Fe2O3)),lwd = 2)
hist(log(ashdata1$Al2O3),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$Al2O3)),lwd = 2)
```

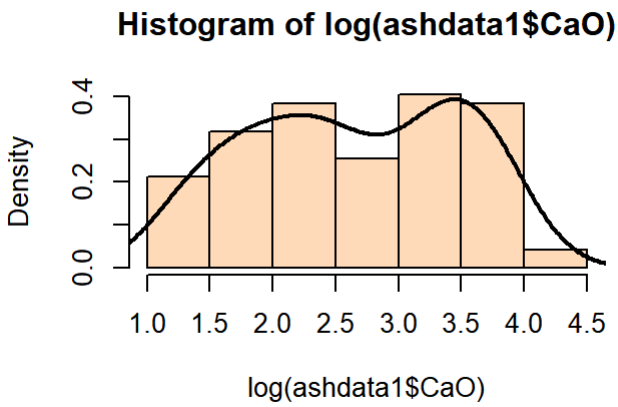


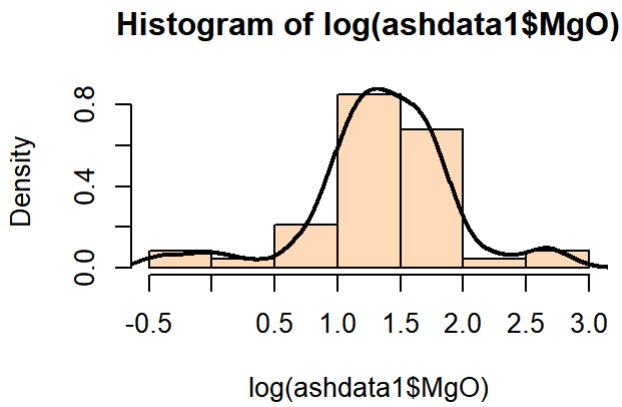


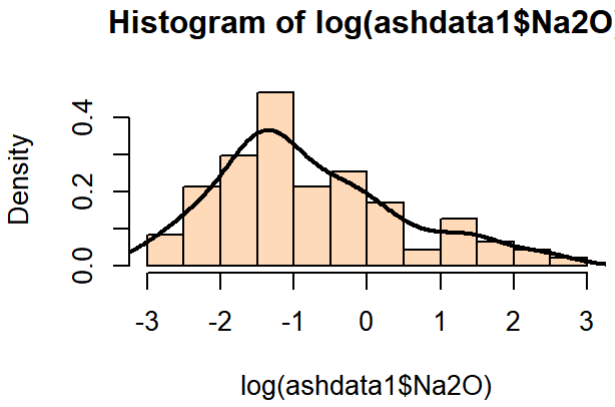


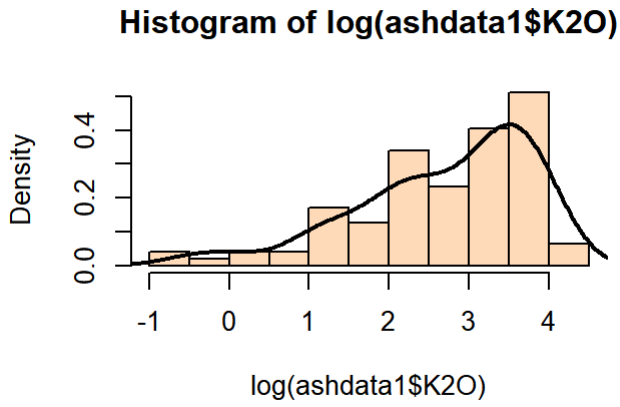



```
hist(log(ashdata1$CaO),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$CaO)),lwd = 2)
hist(log(ashdata1$MgO),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$MgO)),lwd = 2)
hist(log(ashdata1$Na2O),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$Na2O)),lwd = 2)
hist(log(ashdata1$K2O),prob = TRUE,col="peachpuff")
lines(density(log(ashdata1$K2O)),lwd = 2)
```









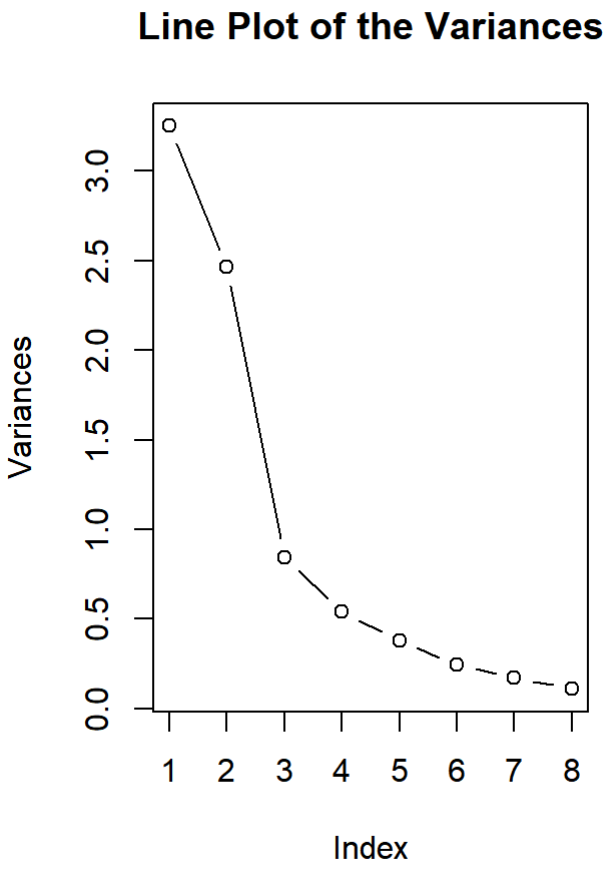
After transforming, we can see that the data points are adjusted to a normally distributed form.For transforming the data, we can use log function on the data. This brings the values closer and there by reducing the skewness of the data points. From the plots, we see that the variables except variable SiO2 are skewed and can be transformed before carrying out the Factor analysis on the dataset.

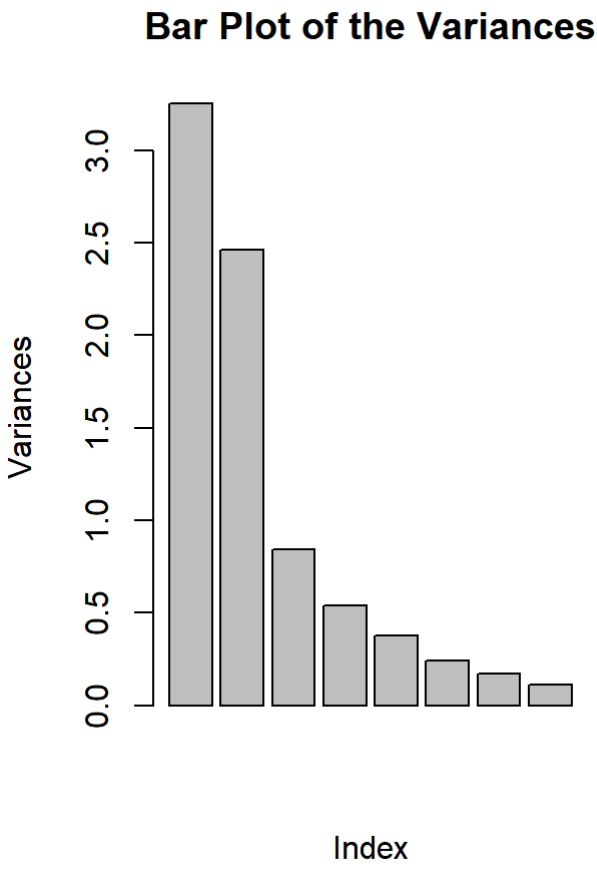
```
ashdata2<-ashdata1
ashdata2$P2O5<-log(ashdata2$P2O5)
ashdata2$Fe2O3<-log(ashdata2$Fe2O3)
ashdata2$Al2O3<-log(ashdata2$Al2O3)
ashdata2$MgO<-log(ashdata2$MgO)
ashdata2$Na2O<-log(ashdata2$Na2O)
ashdata2$CaO<-log(ashdata2$CaO)
ashdata2$K2O<-log(ashdata2$K2O)
```

2cII-Apply factor analysis, employing a varimax rotation, to the (possibly transformed based on your answer to 2(c)(i)) mass concentrations of the eight elements. How many factors do you think are required to capture the correlation structure in the variables? Explain your reasoning. Interpret the first two columns of the loadings matrix.

Choosing the number of factors plays a critical role in factor analysis. The fewer the factors, the higher is the loading values. If there are many factors, then it is fragmented and difficult to interpret.

```
# Determine Number of Factors to Extract
ev <- eigen(cor(ashdata2[,2:9])) # get eigenvalues
eval<-ev$values
par(mfrow=c(1,2))
plot(ev$values,type="b",ylab = "Variances",main = "Line Plot of the Variances")
barplot(eval,xlab = "Index",ylab = "Variances",main = "Bar Plot of the Variances")
```





From the line plot, we see that the elbow is achieved from factor three. This can be seen in the bar plot aswell, where the values after bar three are all similar. So, we can take the number of factors as 3 or 4 for factor Anlaysis based on the significance.

The iteration is repeated until p-value>0.05. This is the advantage of Maximum Likelihood Estimation where formal hypothesis testing procedures for q are available. Initially,q value is started from 1 for estimating teh factor analysis model parameters and the corresponding p-value is verified.

```
for(q in 1:4){
  favalues<-factanal(ashdata2[,2:9],q,"varimax")
  cat("P-value of ",q," factors is ",favalues$PVAL,"\n")
}
```

```
## P-value of  1  factors is  1.960444e-41
## P-value of  2  factors is  9.097335e-08
## P-value of  3  factors is  2.953562e-05
## P-value of  4  factors is  0.03928631
```

Factor 5 cannot be used as 5 factors are too many for 8 variables. So, from the p values, we can use four as the number of factors at 95% confidence level while calculating the loadings and scores of the dataset.

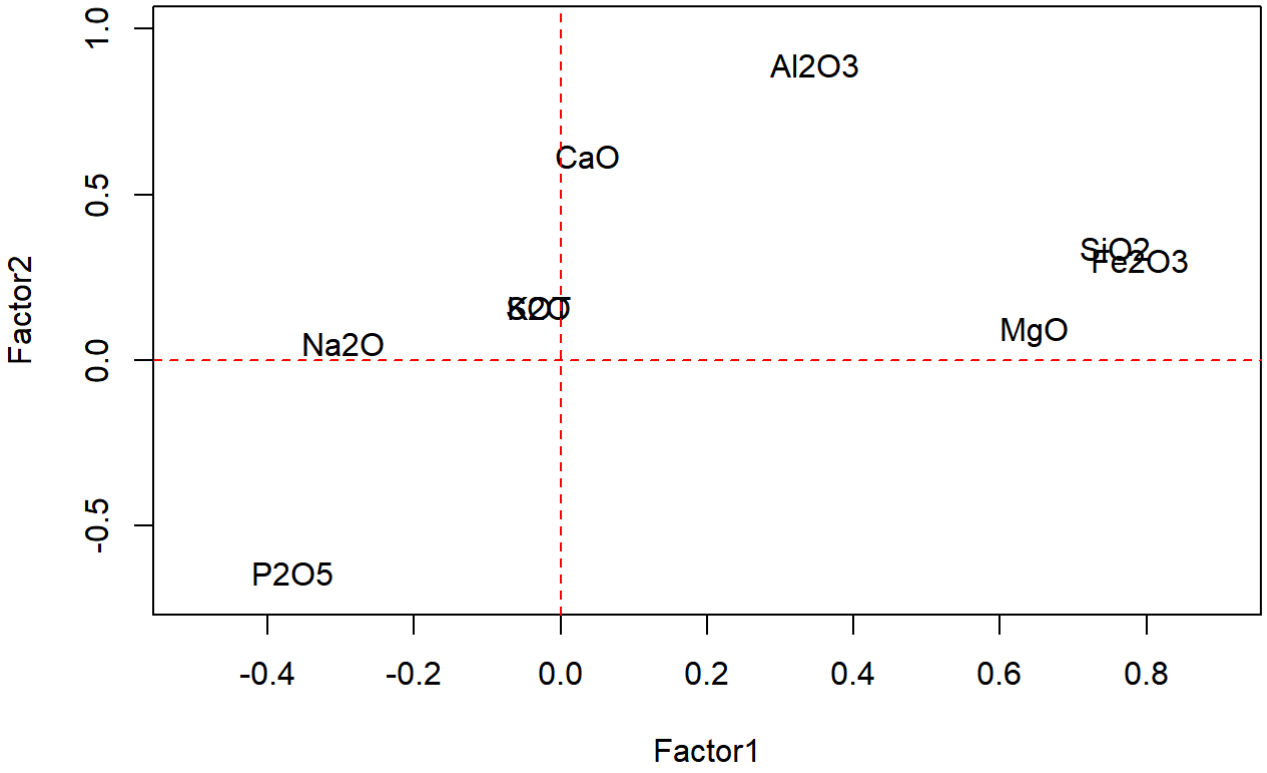
```
factorAnalysis4<-factanal(ashdata2[,2:9],4,"varimax", scores="regression")
print(factorAnalysis4)
```

```
##
## Call:
## factanal(x = ashdata2[, 2:9], factors = 4, data = "varimax",     scores = "regression")
##
## Uniquenesses:
##  P2O5  SiO2 Fe2O3 Al2O3   CaO   MgO  Na2O   K2O
## 0.371 0.073 0.005 0.169 0.083 0.382 0.569 0.328
##
## Loadings:
##       Factor1 Factor2 Factor3 Factor4
## P2O5           0.160   0.774
## SiO2  -0.366  -0.643  -0.549   0.278
## Fe2O3  0.758   0.340  -0.159   0.529
## Al2O3  0.791   0.301  -0.317   0.120
## CaO    0.347   0.891
## MgO            0.618   0.450   0.177
## Na2O   0.647
## K2O   -0.296           0.694  -0.317
##
##                Factor1 Factor2 Factor3 Factor4
## SS loadings      1.963   1.832   1.713   0.511
## Proportion Var   0.245   0.229   0.214   0.064
## Cumulative Var   0.245   0.474   0.688   0.752
##
## Test of the hypothesis that 4 factors are sufficient.
## The chi square statistic is 6.47 on 2 degrees of freedom.
## The p-value is 0.0393
```

Factor analysis creates linear combinations of factors to abstract the variable's underlying communality. To the extent that the variables have an underlying communality, fewer factors capture most of the variance in the data set. This allows us to aggregate a large number of observable variables in a model to represent an underlying concept, making it easier to understand the data.

The uniquenesses ranges from 0 to 1.It corresponds to the proportion of variability, which can not be explained by a linear combination of the factors. A high uniqueness for a variable indicates that the factors do not account well for its variance.

```
plot(factorAnalysis4$loadings,type = "n",xlim = c(-0.5,0.9),ylim = c(-0.7,1))
text(factorAnalysis4$loadings[,1],factorAnalysis4$loadings[,2],colnames(ashdata2))
abline(v=0,lty=2,col=2)
abline(h=0,lty=2,col=2)
```

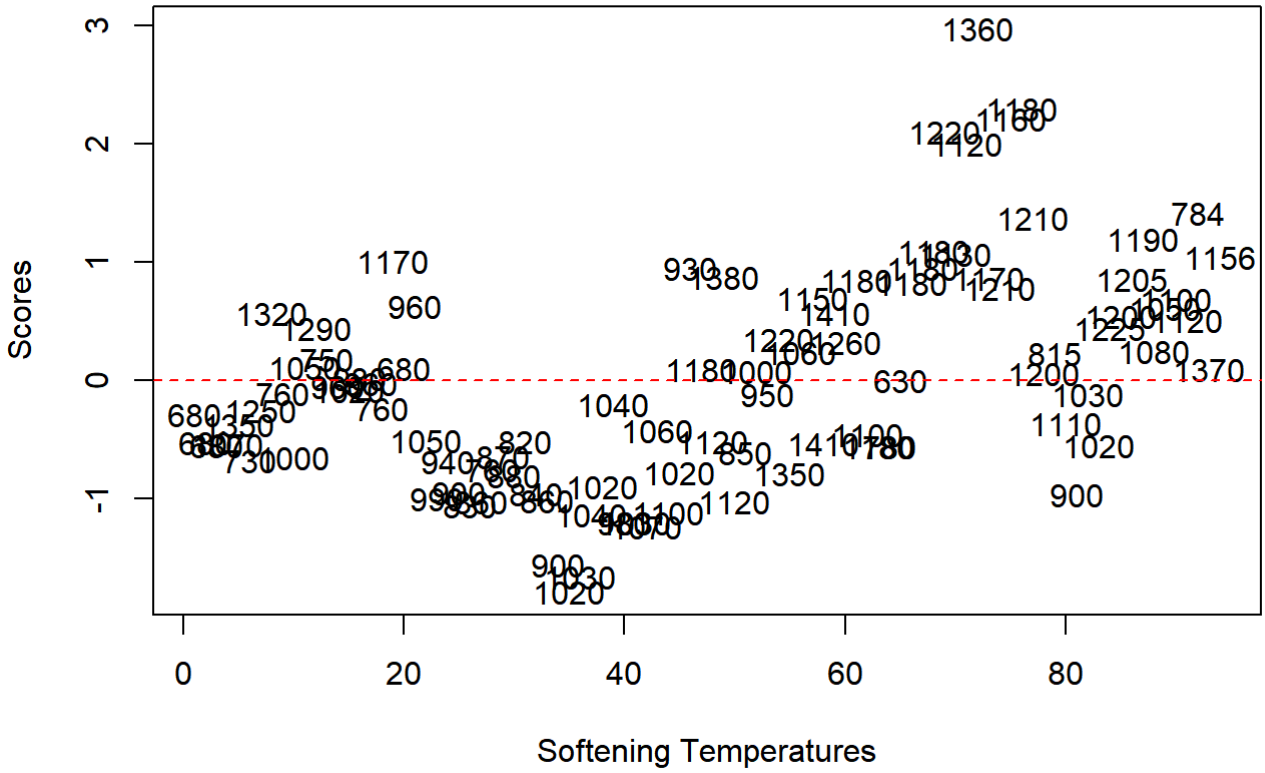


The loadings ranges from −1 to 1.It is the contribution of each original variable to the factor. Variables with a high loading are well explained by the factor. No value for certain variables indicates that its loadings is less than 0.1. From the Loadings of this dataset, we see that element Al2O3 has the strongest association to factor 1. Elements Fe2O3 and Na2O also have good association with factor 1. The empty value of MgO & P2O5 indicates that such item may not be an effective measure of its construct and it is discarded. In factor 2, elements Cao,MgO & SiO2 have a strong association with the variable. It represents that the factor has extracted significant variance from the corresponding variables. The loadings has a lower correlation coefficient for the variable P2O3 and the factor. So, all the variables except P2O3,Na2O and K2O explains a good association of variances on the factor 2. From the above plot,we can predict that the factor 1 accounts for softening temperature of Al2O3,Fe2O3 and SiO2 are higher compared to the other elements. Where as factor 2 clubs the othe elements such as MgO and CaO with higher softening temperature compared with other elements. In all, the four factors account for 75.2% of the total variance ofthe data.

2cIII- Plot the ash samples' factor scores on the first latent factor, illustrating the SOT variable for each sample. Comment on any observed patterns.

From this plot, we can see that the temperatures of these elements are distributed based on the factor scores on the first latent factor. Some of the temeratures are below zero and others are above zero.

```
plot(factorAnalysis4$scores[,1],type="n",xlab = "Softening Temperatures",ylab = "Scores")
text(factorAnalysis4$scores[,1],labels =ashdata2$SOT)
abline(h=0,lty=2,col=2)
```



We can see that the sample whose softening temperature are about 1200 centigrate are mostly above the zero score of the first latent factor. Only few of the samples with softening temperature above 1200 centigrate are below zero. The location of each original observation in the reduced factor space is often used as input to a subsequent analysis, or for visualisation purpose. Similarly, most of the sample with softening temperature less than 1200 are below zero factor score and only a few are above this zero margin.This indicates that the softening temperature of these sample elements are less when the factor score is less than zero. The large and positive loadings on the first factor led us to believe that the first factor measures the overall Softening temperature of the elements in the samples.So, if a sample has good overall temperature, they should have a positive score on the first latent factor i.e, the value will be above zero margin in the plot.