```
#Read the data from GeneData text file and put it in a table
GeneData.table = read.delim("GeneData.txt")
GeneData=GeneData.table[,-c(1,2)]
#Display the data
GeneData
```

```
##       A    C    G    T
## 1  0.01 0.02 0.51 0.46
## 2  0.02 0.04 0.52 0.42
## 3  0.10 0.20 0.07 0.63
## 4  0.12 0.24 0.35 0.29
## 5  0.16 0.32 0.15 0.37
## 6  0.17 0.34 0.02 0.47
## 7  0.21 0.42 0.01 0.36
## 8  0.22 0.44 0.25 0.09
## 9  0.23 0.46 0.26 0.05
## 10 0.24 0.48 0.03 0.25
```

The result of standardization (or Z-score normalization) is that the features will be rescaled so that they'll have the properties of a standard normal distribution with

μ=0 and σ=1 where μ is the mean (average) and σ is the standard deviation from the mean; standard scores (also called z scores) of the samples are calculated as follows:

z=(x−μ)/σ

```
#pre-process data using standardization
GeneData.s=scale(GeneData,center = TRUE,scale = TRUE)
#display standardized data
GeneData.s
```

```
##                A          C          G          T
##  [1,] -1.6447230 -1.6447230  1.5052385  0.6856505
##  [2,] -1.5255401 -1.5255401  1.5566118  0.4589892
##  [3,] -0.5720776 -0.5720776 -0.7551879  1.6489612
##  [4,] -0.3337119 -0.3337119  0.6832653 -0.2776601
##  [5,]  0.1430194  0.1430194 -0.3442013  0.1756625
##  [6,]  0.2622022  0.2622022 -1.0120546  0.7423159
##  [7,]  0.7389335  0.7389335 -1.0634279  0.1189972
##  [8,]  0.8581163  0.8581163  0.1695320 -1.4109668
##  [9,]  0.9772992  0.9772992  0.2209053 -1.6376282
## [10,]  1.0964820  1.0964820 -0.9606812 -0.5043215
## attr(,"scaled:center")
##     A     C     G     T
## 0.148 0.296 0.217 0.339
## attr(,"scaled:scale")
##          A          C          G          T
## 0.08390471 0.16780942 0.19465354 0.17647474
```

A correlation is a statistical measure of the relationship between two variables. The measure is best used in variables that demonstrate a linear relationship between each other.

The correlation coefficient is a value that indicates the strength of the relationship between variables. The coefficient can take any values from -1 to 1. The interpretations of the values are:

-1: Perfect negative correlation. The variables tend to move in opposite directions (i.e., when one variable increases, the other variable decreases). 0: No correlation. The variables do not have a relationship with each other. 1: Perfect positive correlation. The variables tend to move in the same direction (i.e., when one variable increases, the other variable also increases).

The correlation coefficient that indicates the strength of the relationship between two variables can be found using the following formula:

Correlation - Formula

rxy = ∑(x(i)- mean(x))*(y(i)-mean(y)) / √ (∑(x(i)-mean(x))2* ∑(y(i)-mean(y))2)

Where:

rxy – the correlation coefficient of the linear relationship between the variables x and y xi – the values of the x-variable in a sample x̄ – the mean of the values of the x-variable yi – the values of the y-variable in a sample ȳ – the mean of the values of the y-variable

```
GeneData.cor=cor(GeneData)
```

Eigenvalues are the special set of scalars associated with the system of linear equations. Eigenvectors are the vectors (non-zero) that do not change the direction when any linear transformation is applied.

The basic equation is Ax = λx The number or scalar value "λ" is an eigenvalue of A. x is an eigenvector of A corresponding to eigenvalue, λ.

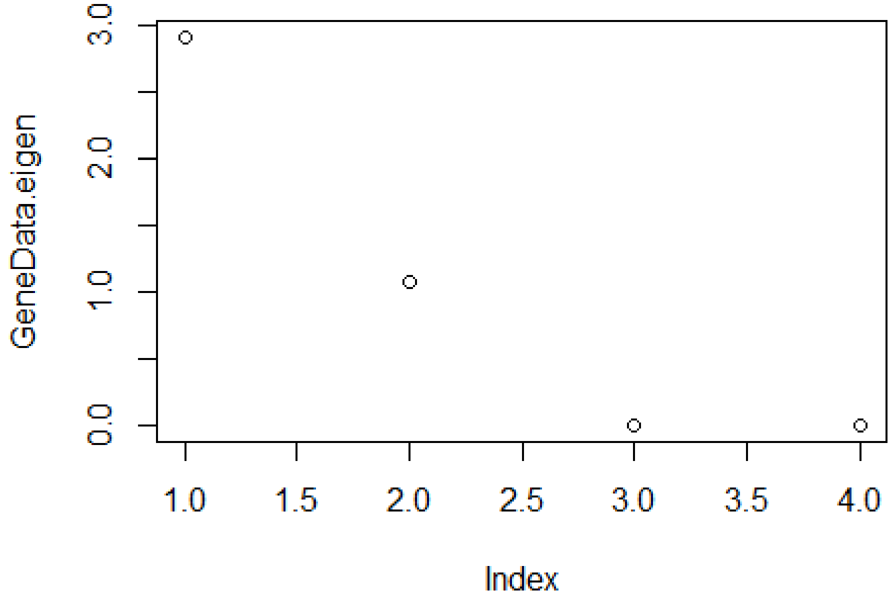
```
GeneData.eig=eigen(GeneData.cor)
GeneData.eigen=cbind(GeneData.eig$values)
GeneData.eigen

##               [,1]
## [1,]  2.917930e+00
## [2,]  1.082070e+00
## [3,] -8.140250e-17
## [4,] -1.692014e-16

PrGeneData.eignen=GeneData.eigen/sum(GeneData.eigen)
PrGeneData.eignen
```

```
##                 [,1]
## [1,]  7.294826e-01
## [2,]  2.705174e-01
## [3,] -2.035063e-17
## [4,] -4.230035e-17
```

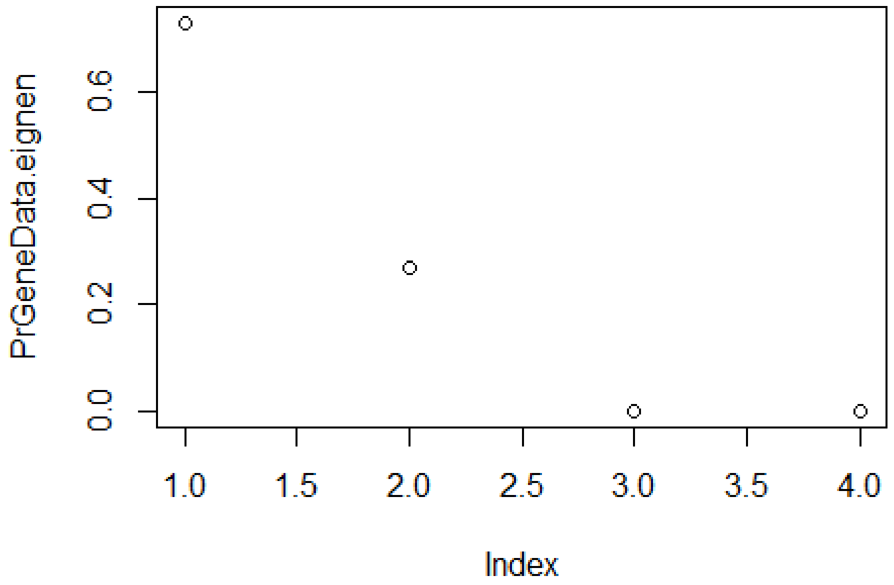
```
cumsum(PrGeneData.eignen)
```

```
## [1] 0.7294826 1.0000000 1.0000000 1.0000000
```

```
plot(GeneData.eigen)
```

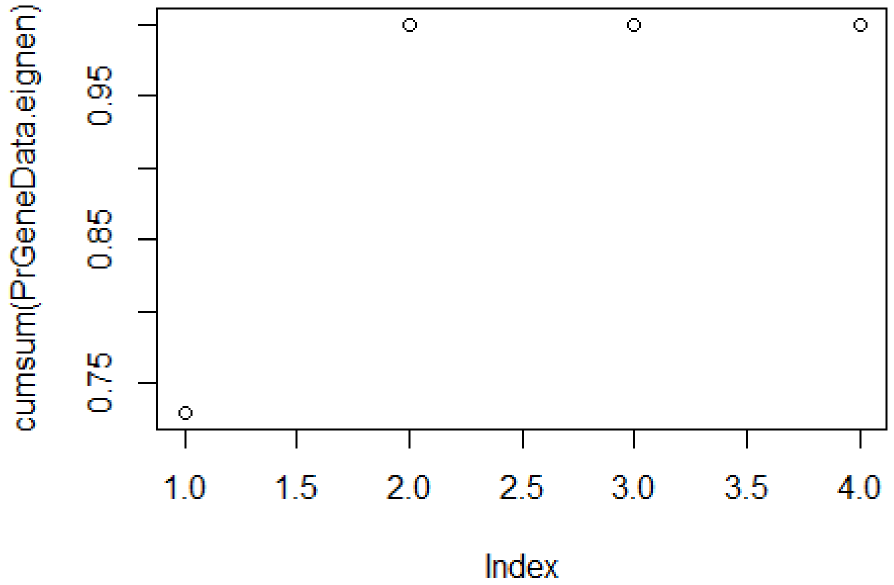


```
plot(PrGeneData.eignen)
```



### Provides variance of principal components

```
plot(cumsum(PrGeneData.eignen))
```



### Principal component analysis (PCA) is a technique used to emphasize variation and bring out strong

patterns in a dataset. It’s often used to make data easy to explore and visualize. Produce principal components

```
GeneData.new=GeneData.s%*%GeneData.eig$vectors
colnames(GeneData.new)=c("pc1","pc2","pc3","pc4")
GeneData.new
```

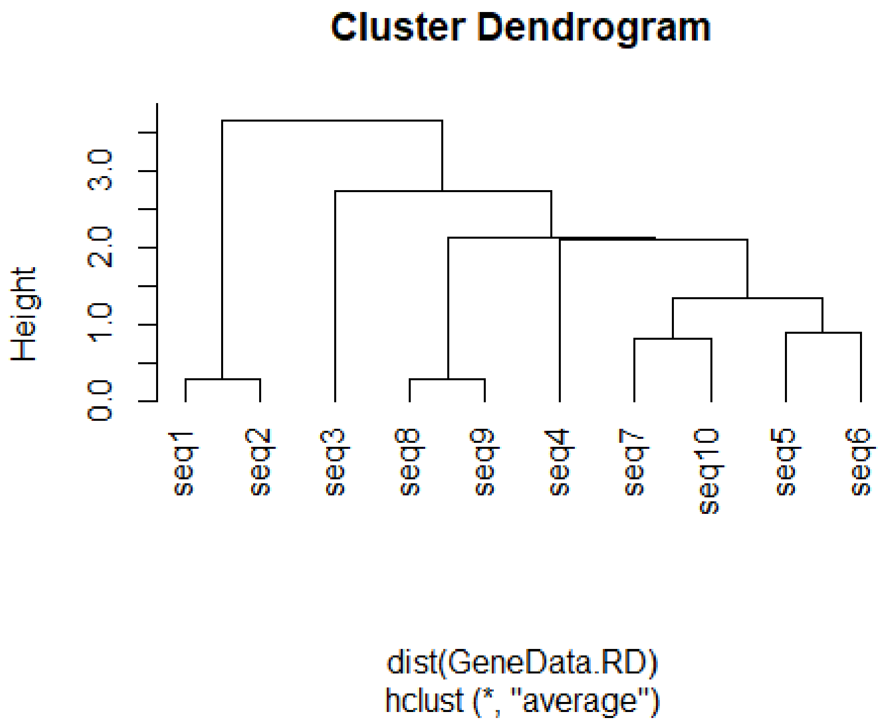
```
##              pc1        pc2           pc3           pc4
##  [1,]  2.8133100 -0.4810131  1.665335e-16 -5.689893e-16
##  [2,]  2.6113194 -0.6850314  1.110223e-16 -4.510281e-16
##  [3,]  0.9636198  1.7364804  3.330669e-16 -3.053113e-16
##  [4,]  0.5752260 -0.6601431 -5.551115e-17 -8.326673e-17
##  [5,] -0.2471250  0.3594026  2.220446e-16  5.551115e-17
##  [6,] -0.4575089  1.2261619  2.220446e-16  3.469447e-17
##  [7,] -1.2684690  0.7925093  1.665335e-16  3.044440e-16
##  [8,] -1.4566707 -1.1706435  1.110223e-16  4.718448e-16
##  [9,] -1.6586613 -1.3746618  1.110223e-16  5.828671e-16
## [10,] -1.8750404  0.2569386  2.775558e-16  3.642919e-16
```

Produce a dendogram for clustering

```
GeneData.RD=GeneData.new[,c(1,2)]
geneclust=hclust(dist(GeneData.RD),method = "average")
x=c("seq1","seq2","seq3","seq4","seq5","seq6","seq7","seq8","seq9","seq10")
```

Visualization of the dendrogram

```
plot(geneclust,labels=x,cex=1,hang = -2)
```


Cluster Dendrogram
Height
0.0
1.0
2.0
3.0
seq1
seq2
seq3
seq8
seq9
seq4
seq7
seq10
seq5
seq6
dist(GeneData.RD)
hclust (*, "average")