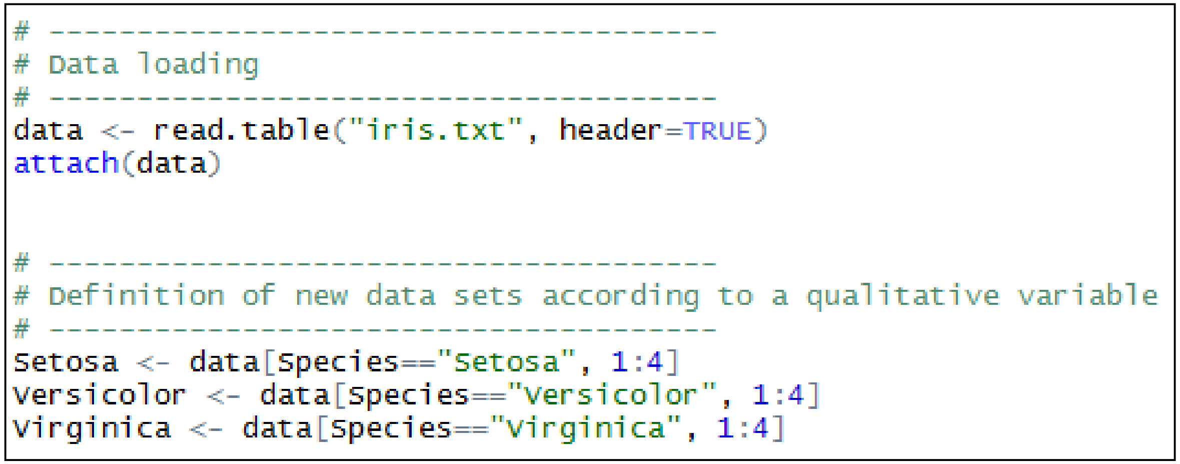
High-dimensional data analysis

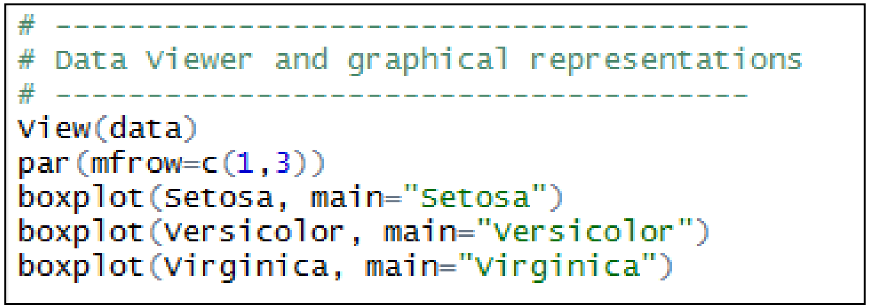
Script 4: Principal component analysis

Data loading: Iris

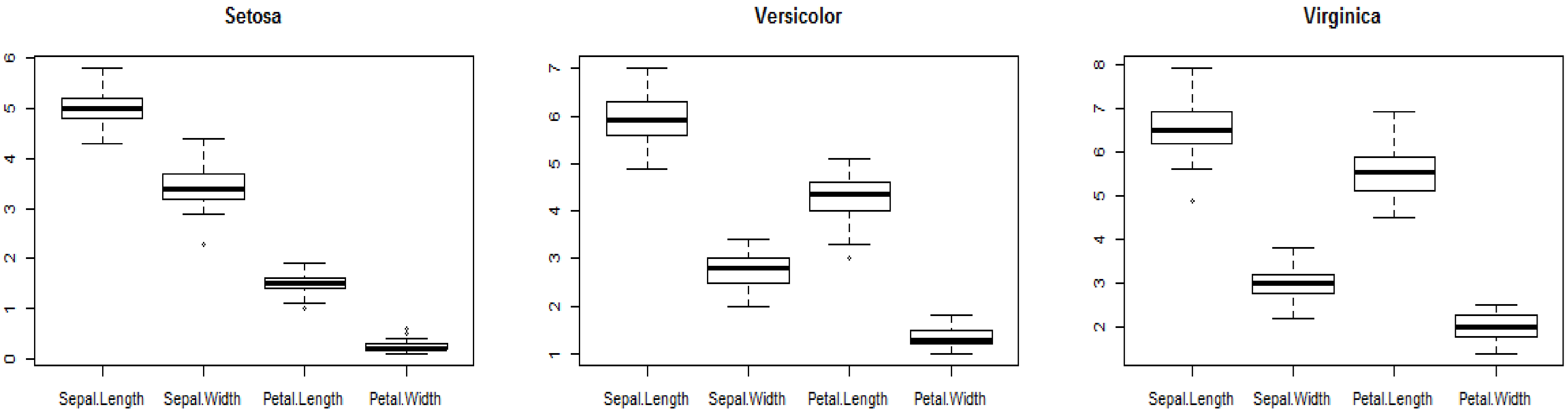


```
# --------------------------------------
# Data loading
# --------------------------------------
data <- read.table("iris.txt", header=TRUE)
attach(data)


# --------------------------------------
# Definition of new data sets according to a qualitative variable
# --------------------------------------
Setosa <- data[Species=="Setosa", 1:4]
Versicolor <- data[Species=="Versicolor", 1:4]
Virginica <- data[Species=="Virginica", 1:4]
```

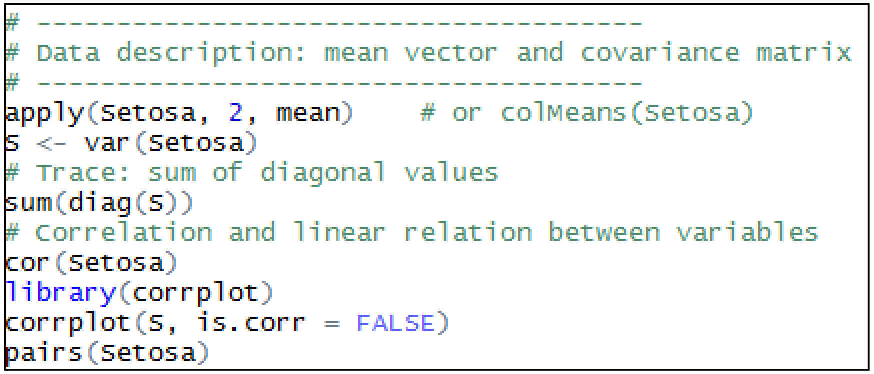
Data viewer



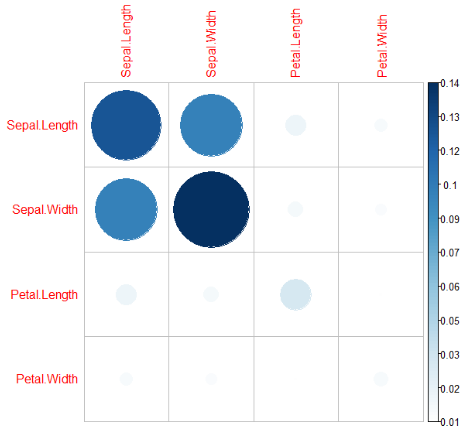
```
# -------------------------------------
# Data Viewer and graphical representations
# -------------------------------------
View(data)
par(mfrow=c(1,3))
boxplot(Setosa, main="Setosa")
boxplot(Versicolor, main="Versicolor")
boxplot(Virginica, main="Virginica")
```



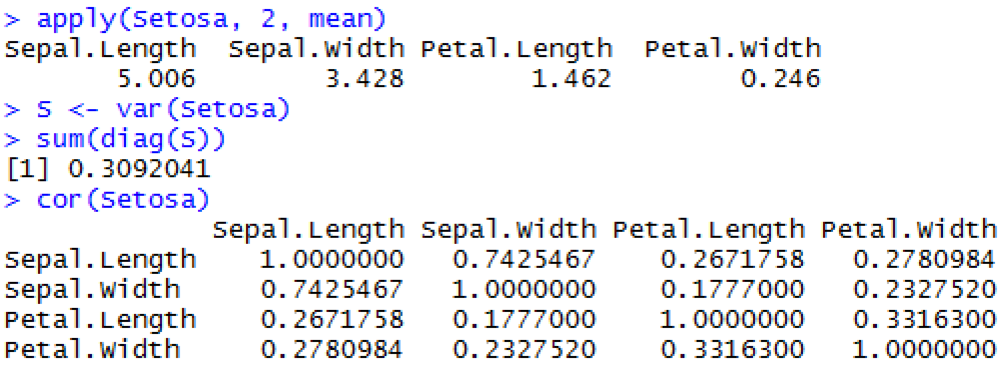
Mean vector, covariance and correlation matrices



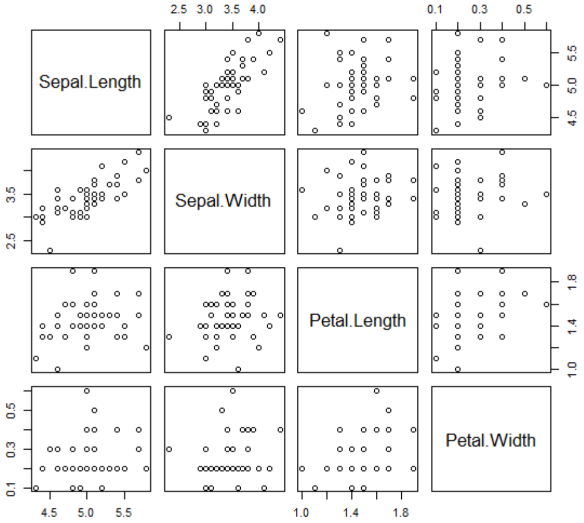
```
# ----------------------------------------
# Data description: mean vector and covariance matrix
# ----------------------------------------
apply(Setosa, 2, mean)    # or colMeans(Setosa)
S <- var(Setosa)
# Trace: sum of diagonal values
sum(diag(S))
# Correlation and linear relation between variables
cor(Setosa)
library(corrplot)
corrplot(S, is.corr = FALSE)
pairs(Setosa)
```



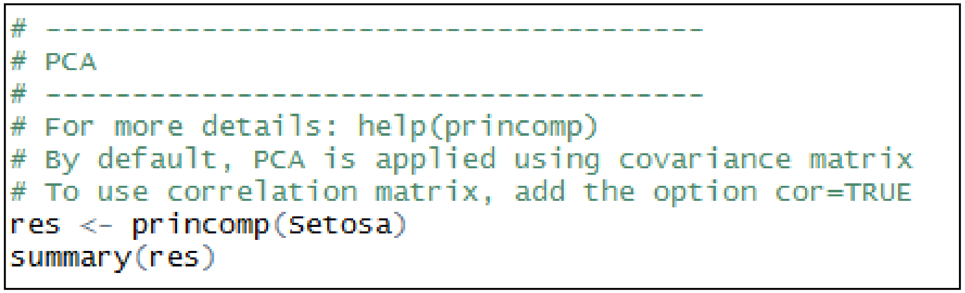
Output:



```
> apply(Setosa, 2, mean)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.006        3.428        1.462        0.246
> S <- var(Setosa)
> sum(diag(S))
[1] 0.3092041
> cor(Setosa)
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000   0.7425467    0.2671758   0.2780984
Sepal.Width     0.7425467   1.0000000    0.1777000   0.2327520
Petal.Length    0.2671758   0.1777000    1.0000000   0.3316300
Petal.Width     0.2780984   0.2327520    0.3316300   1.0000000
```

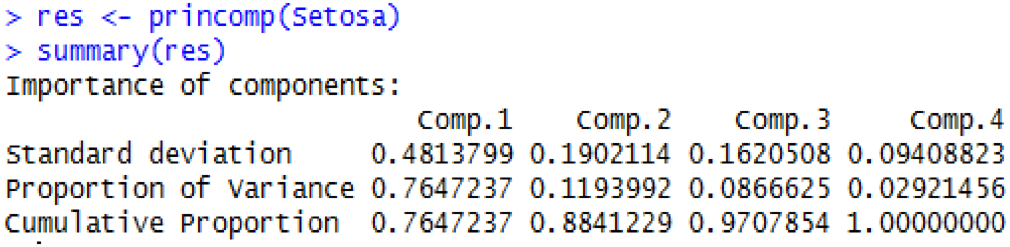


PCA



```
# --------------------------------------
# PCA
# --------------------------------------
# For more details: help(princomp)
# By default, PCA is applied using covariance matrix
# To use correlation matrix, add the option cor=TRUE
res <- princomp(Setosa)
summary(res)
```

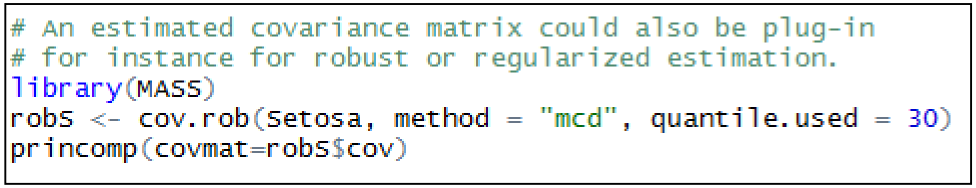
Output:



```
> res <- princomp(Setosa)
> summary(res)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4
Standard deviation     0.4813799 0.1902114 0.1620508 0.09408823
Proportion of Variance 0.7647237 0.1193992 0.0866625 0.02921456
Cumulative Proportion  0.7647237 0.8841229 0.9707854 1.00000000
```

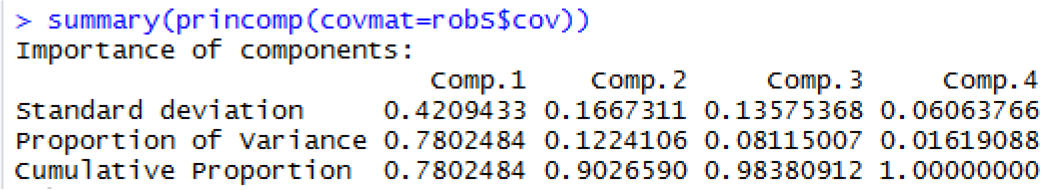
PCA with plug-in covariance matrix

An estimated covariance matrix could also be plug-in the PCA procedure. For instance, if we consider a robust or a regularized estimation.



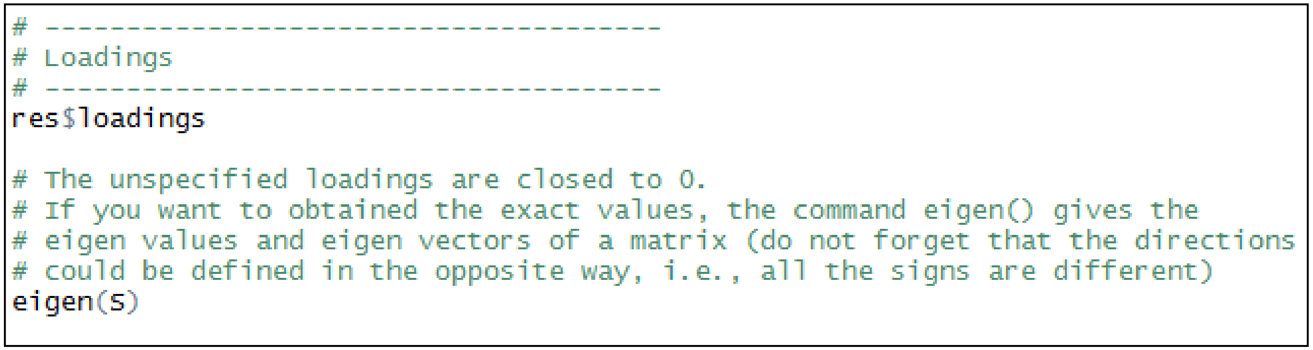
```
# An estimated covariance matrix could also be plug-in
# for instance for robust or regularized estimation.
library(MASS)
robS <- cov.rob(Setosa, method = "mcd", quantile.used = 30)
princomp(covmat=robS$cov)
```

Output:



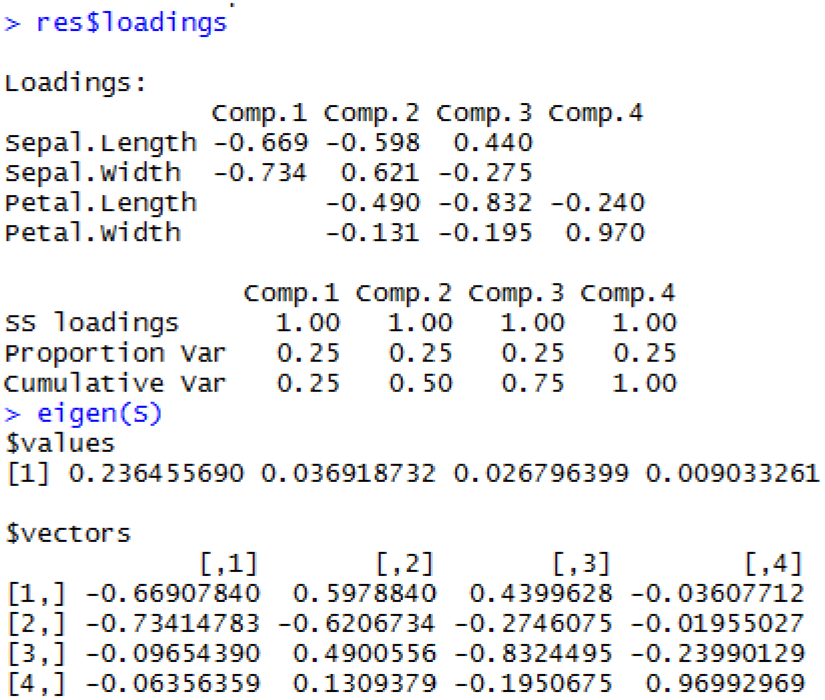
```
> summary(princomp(covmat=robS$cov))
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4
Standard deviation     0.4209433 0.1667311 0.13575368 0.06063766
Proportion of Variance 0.7802484 0.1224106 0.08115007 0.01619088
Cumulative Proportion  0.7802484 0.9026590 0.98380912 1.00000000
```

PCA loadings



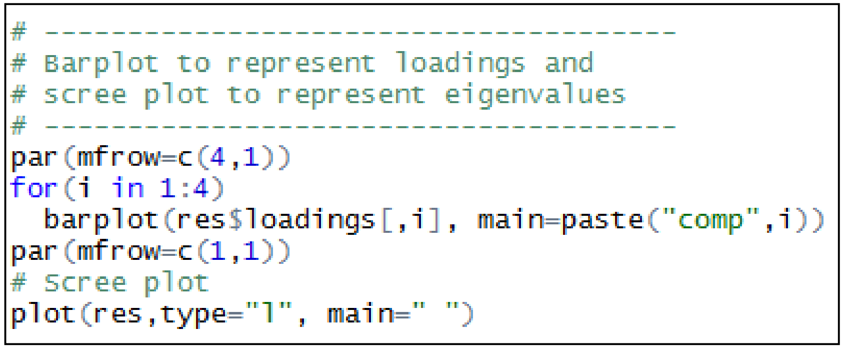
```
# -------------------------------------
# Loadings
# -------------------------------------
res$loadings

# The unspecified loadings are closed to 0.
# If you want to obtained the exact values, the command eigen() gives the
# eigen values and eigen vectors of a matrix (do not forget that the directions
# could be defined in the opposite way, i.e., all the signs are different)
eigen(S)
```

Output:

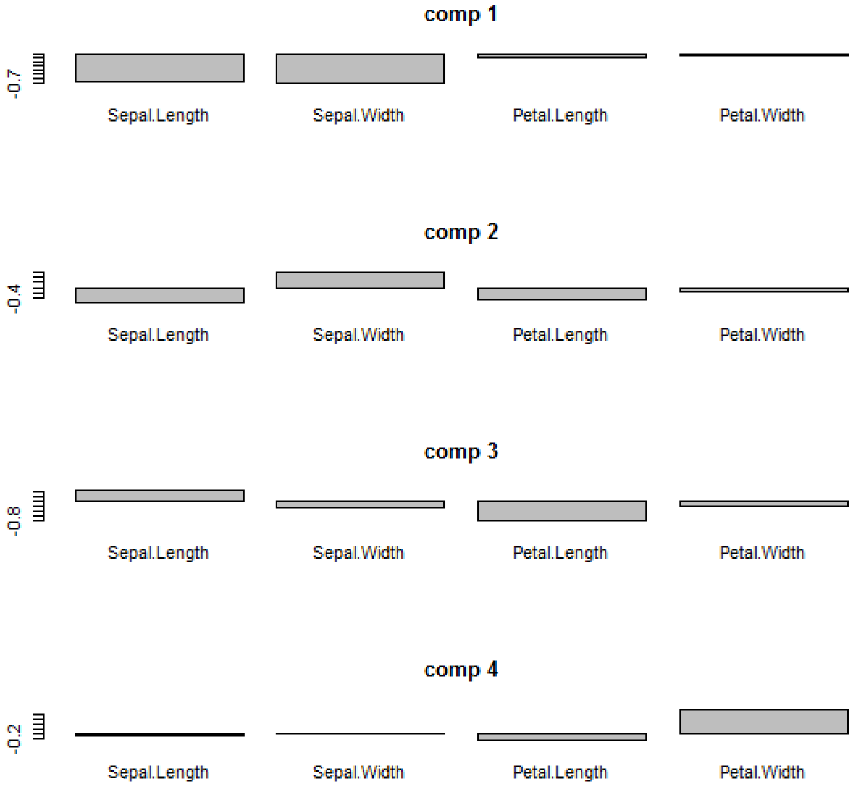


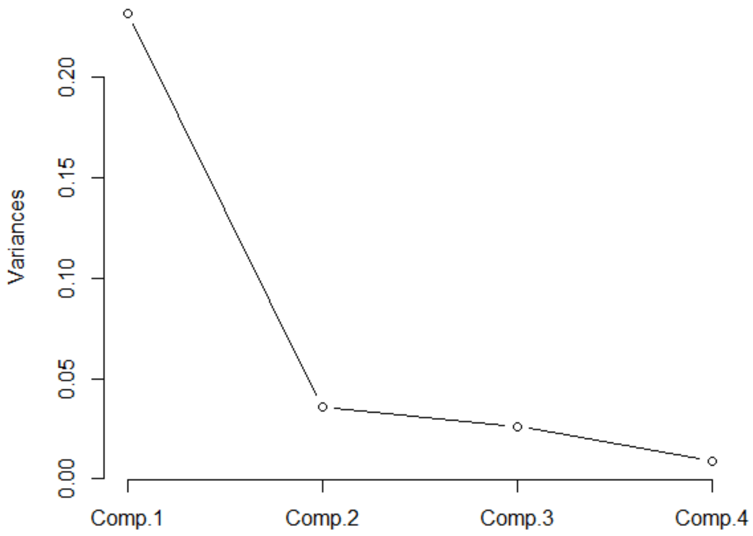
```
> res$loadings

Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4
Sepal.Length -0.669 -0.598  0.440
Sepal.Width  -0.734  0.621 -0.275
Petal.Length        -0.490 -0.832 -0.240
Petal.Width         -0.131 -0.195  0.970

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
> eigen(S)
$values
[1] 0.236455690 0.036918732 0.026796399 0.009033261

$vectors
            [,1]       [,2]       [,3]        [,4]
[1,] -0.66907840  0.5978840  0.4399628 -0.03607712
[2,] -0.73414783 -0.6206734 -0.2746075 -0.01955027
[3,] -0.09654390  0.4900556 -0.8324495 -0.23990129
[4,] -0.06356359  0.1309379 -0.1950675  0.96992969
```

Graphical representations

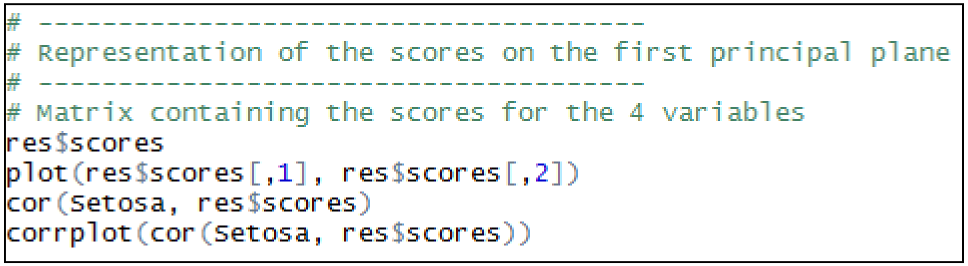


```
# ----------------------------------------
# Barplot to represent loadings and
# scree plot to represent eigenvalues
# ----------------------------------------
par(mfrow=c(4,1))
for(i in 1:4)
  barplot(res$loadings[,i], main=paste("comp",i))
par(mfrow=c(1,1))
# Scree plot
plot(res,type="l", main=" ")
```



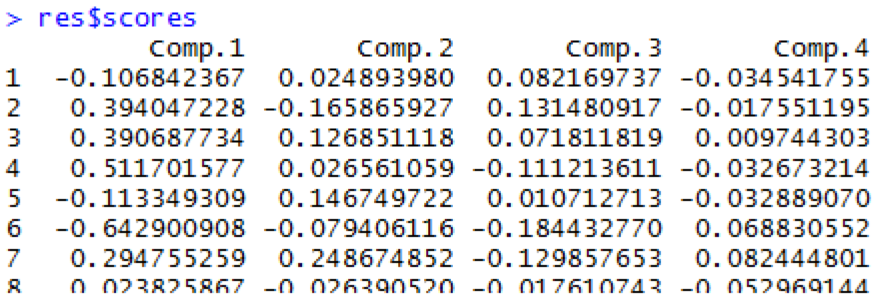


PCA scores

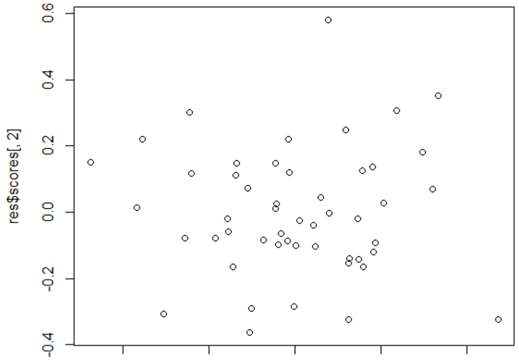


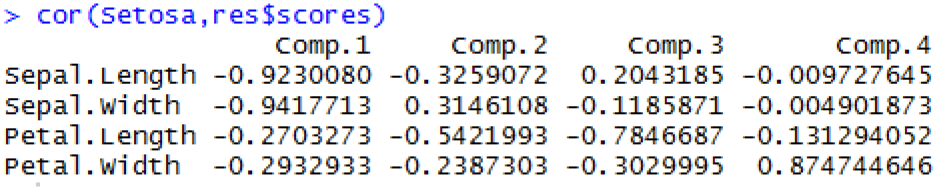
```
# ---------------------------------------
# Representation of the scores on the first principal plane
# ---------------------------------------
# Matrix containing the scores for the 4 variables
res$scores
plot(res$scores[,1], res$scores[,2])
cor(Setosa, res$scores)
corrplot(cor(Setosa, res$scores))
```

Output:

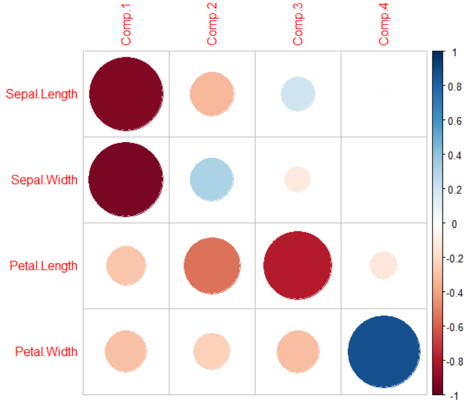


```
> res$scores
        Comp.1      Comp.2      Comp.3       Comp.4
1 -0.106842367  0.024893980  0.082169737 -0.034541755
2  0.394047228 -0.165865927  0.131480917 -0.017551195
3  0.390687734  0.126851118  0.071811819  0.009744303
4  0.511701577  0.026561059 -0.111213611 -0.032673214
5 -0.113349309  0.146749722  0.010712713 -0.032889070
6 -0.642900908 -0.079406116 -0.184432770  0.068830552
7  0.294755259  0.248674852 -0.129857653  0.082444801
8  0.023825867 -0.026390520 -0.017610743 -0.052969144
```

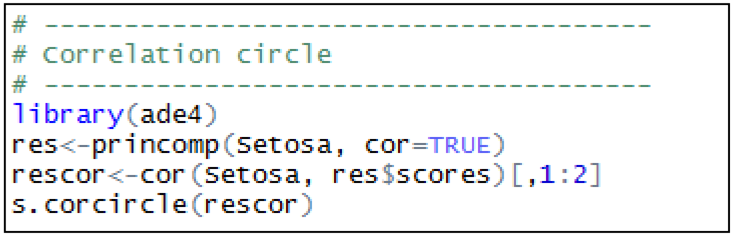




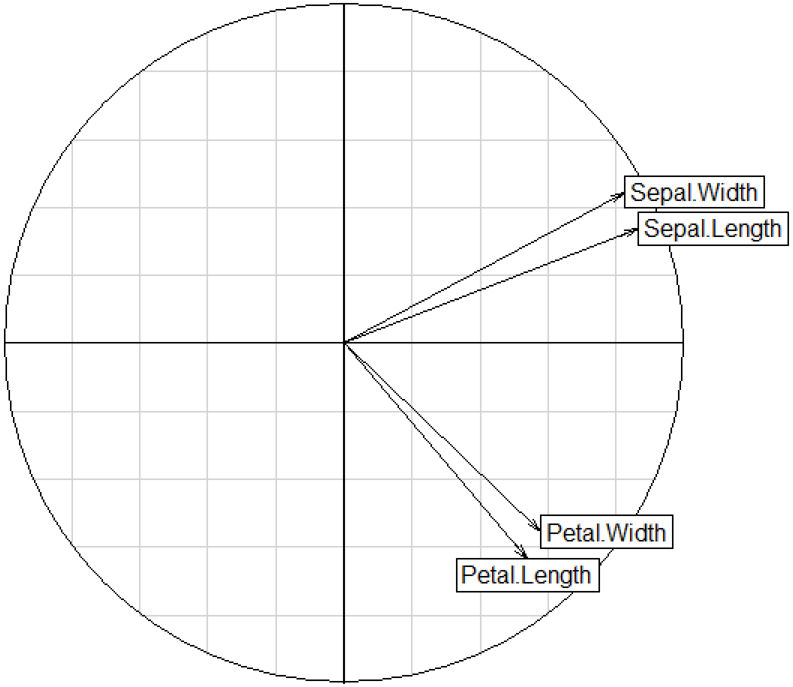
```
> cor(Setosa,res$scores)
                 Comp.1     Comp.2     Comp.3       Comp.4
Sepal.Length -0.9230080 -0.3259072  0.2043185 -0.009727645
Sepal.Width  -0.9417713  0.3146108 -0.1185871 -0.004901873
Petal.Length -0.2703273 -0.5421993 -0.7846687 -0.131294052
Petal.Width  -0.2932933 -0.2387303 -0.3029995  0.874744646
```



Correlation circle



```
# ----------------------------------------
# Correlation circle
# ----------------------------------------
library(ade4)
res<-princomp(Setosa, cor=TRUE)
rescor<-cor(Setosa, res$scores)[,1:2]
s.corcircle(rescor)
```


Sepal.Width
Sepal.Length
Petal.Width
Petal.Length