Principal Component Analysis of Simulated Data

P. Anipa

Simulating 100 observations from a bivariate normal distribution.

The bivariate normal distribution has the following parameters:

$$\mu = [47]$$

$$\Sigma = \begin{bmatrix} 10 & 6 \\ 6 & 8 \end{bmatrix}$$

We want to simulate 100 observations from this distribution. We will then plot the data and label the data points with the corresponding observation number.

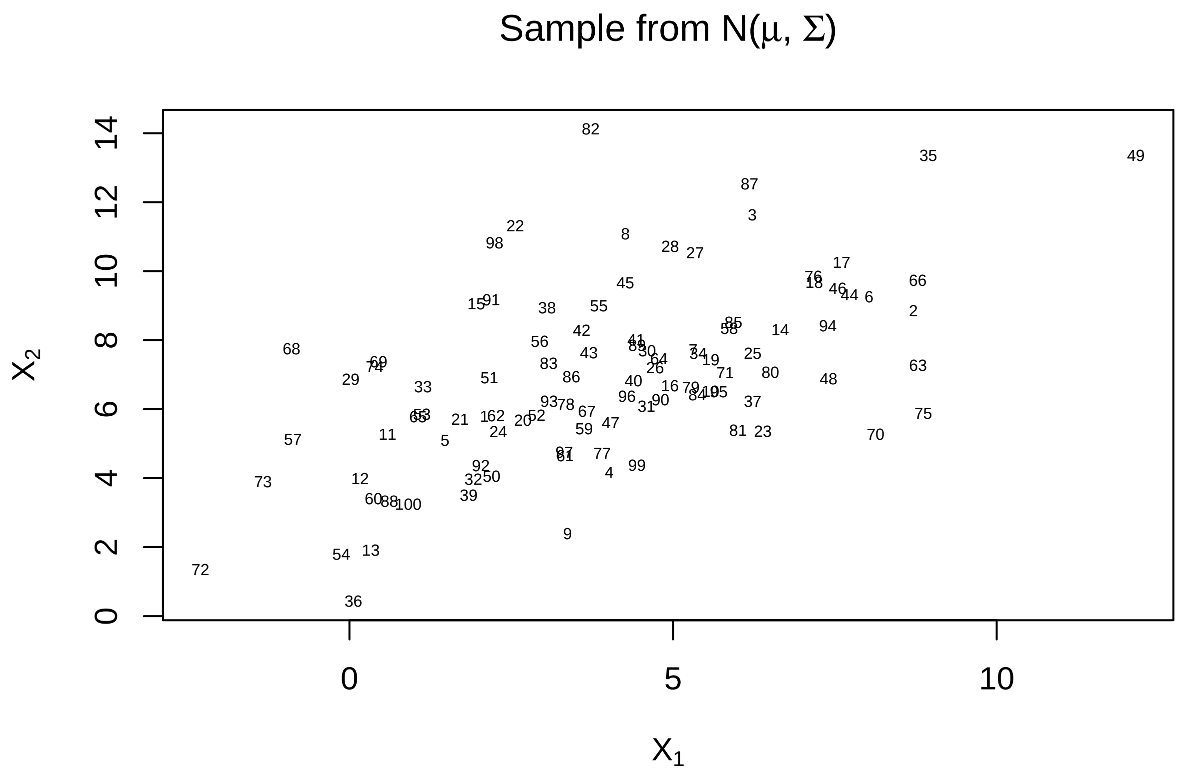
```
mu = c(4, 7)
sigma = matrix(c(10, 6, 6, 8), byrow = TRUE, ncol = 2)

#This library contains the function "rmvnorm" which generates random samples
#from a multivariate normal distribution.
library(mvtnorm)

#Set the seed for generating the same sequence of random variables for
#n(in our case 100) observations.
set.seed(123)
n = 100

#Use of function 'rmvnorm' to generate the data
x = rmvnorm(n, mu, sigma)
head(x)
```

```
##          [,1]      [,2]
## [1,] 2.085788  5.796222
## [2,] 8.712286  8.857059
## [3,] 6.224902 11.627028
## [4,] 4.013622  4.183931
## [5,] 1.478637  5.096696
## [6,] 8.027319  9.255103
```

```
dim(x)
```

```
## [1] 100   2
```

```
lbs = as.character(1:100)
plot(x, pch = 20, xlab = expression("X"[1]), ylab = expression("X"[2]),
     main = expression(paste("Sample from ", "N(", mu, ", ", Sigma, ")")),
     type="n")
text(x, labels = lbs, cex = 0.5)
```



We now perform the covariance based PCA transformation to the data set.

```
x_pca = princomp(x, cor = FALSE)
summary(x_pca)
```

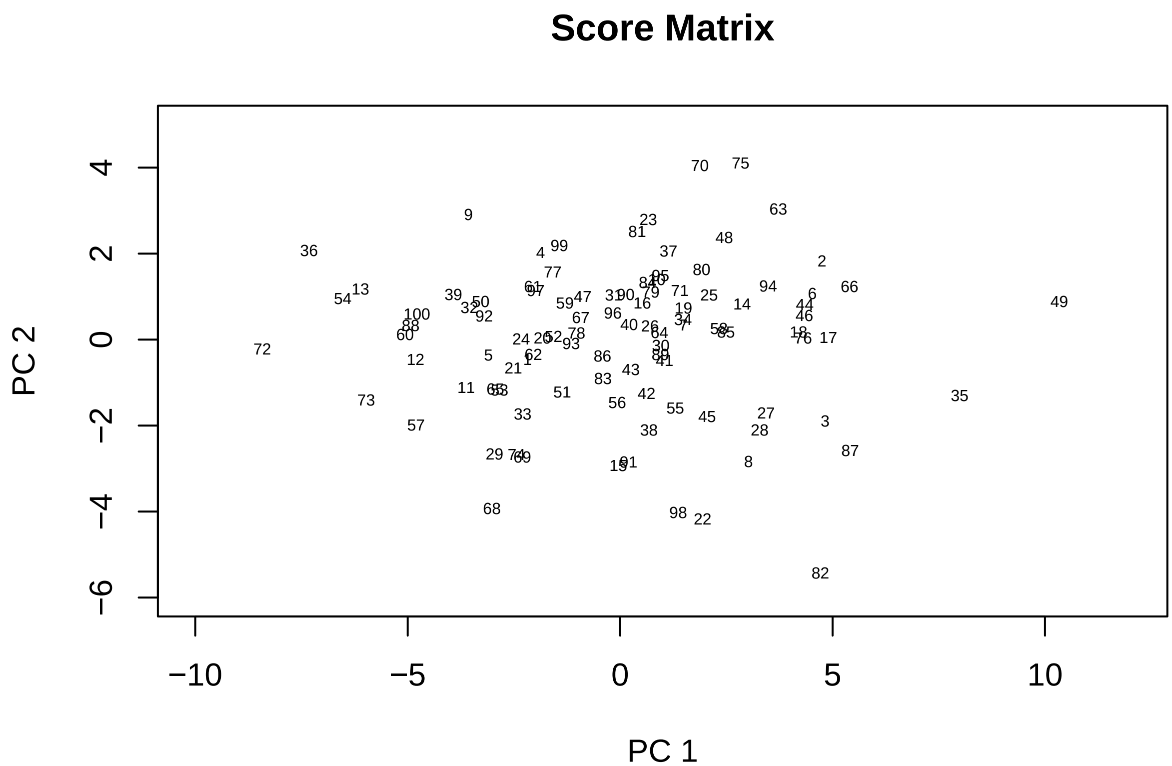
```
## Importance of components:
##                           Comp.1    Comp.2
## Standard deviation     3.3081457 1.7409557
## Proportion of Variance 0.7831141 0.2168859
## Cumulative Proportion  0.7831141 1.0000000
```

Plot of the score matrix.

```
#We store the data in variable "score"
score = x_pca$scores
head(score)
```

```
##         Comp.1     Comp.2
## [1,] -2.178272 -0.4622198
## [2,]  4.752625  1.8272928
## [3,]  4.827310 -1.8948341
## [4,] -1.871092  2.0321071
## [5,] -3.099513 -0.3658647
## [6,]  4.524100  1.0687446
```

```r
#Plot of score matrix
lbss = as.character(1:100)
plot(score, pch = 20, xlab = expression("PC 1"), ylab = expression("PC 2"),
     xlim = c(-10,12), ylim = c(-6,5), main = "Score Matrix", type="n")
text(score, labels = lbss, cex = 0.5)
```



We compare the plots of the original data and the score matrix data and describe the differences.

The data points were rotated and the data was centered by the y-axis.

Manual calculation of the score matrix.

Scores, that is, transformed variables are given by:

$Y = (X - \mathbf{1}_n \bar{x}^\top)G$

where G is the matrix of eigenvectors of the sample covariance and $\bar{x}$ is the sample mean vector.

We wish to calculate the G and Y matrices without using any existing PCA functions.

```r
n = nrow(x)
eig = eigen((n - 1) / n * cov(x)) #eigen of sample covariance
G = eig$vectors #eigen vectors
Y = as.matrix(sweep(x, 2, colMeans(x), "-")) %*% G #score matrix
head(G)
```

```
##             [,1]       [,2]
```

```
## [1,] -0.7304868  0.6829268
## [2,] -0.6829268 -0.7304868
```

```r
head(Y)
```

```
##           [,1]       [,2]
## [1,]  2.178272 -0.4622198
## [2,] -4.752625  1.8272928
## [3,] -4.827310 -1.8948341
## [4,]  1.871092  2.0321071
## [5,]  3.099513 -0.3658647
## [6,] -4.524100  1.0687446
```

Verifying that the G and Y matrices are calculated correctly.

We verify that the estimated scores and the loadings are equal (up to signs) in parts b) and e).

```r
#Matrix G
#We store the data in variable "load"
load = x_pca$loadings
all(abs(round(G, 2)) == abs(round(load, 2)))
```

```
## [1] TRUE
```

```r
#Matrix Y
all(abs(round(Y, 2)) == abs(round(score, 2)))
```

```
## [1] TRUE
```

PCA plot in the original data

Now we plot the directions of the first and second principal components to the original data.

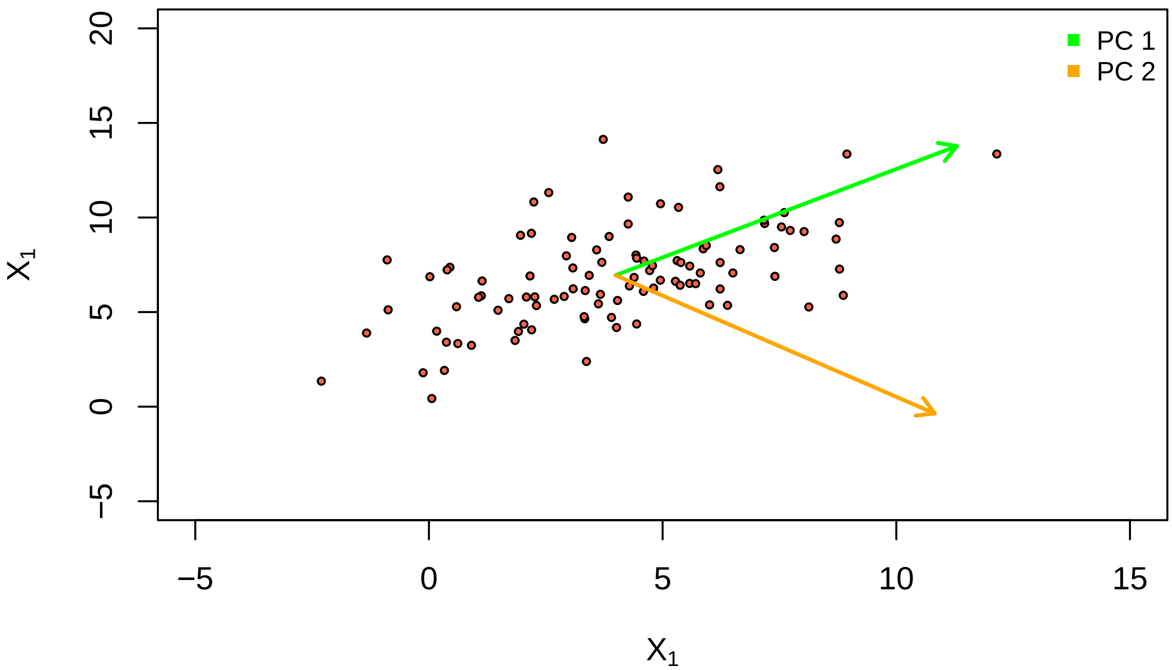
```r
center = x_pca$center
load = x_pca$loadings[]
load
```

```
##         Comp.1     Comp.2
## [1,] 0.7304868  0.6829268
## [2,] 0.6829268 -0.7304868
```

```r
arrows_xy = 10 * load + rep(1, 2) %*% t(x_pca$center)
arrows_xy
```

```
##        Comp.1     Comp.2
## [1,] 11.29752 13.7754456
## [2,] 10.82192 -0.3586906
```

```r
plot(x, xlim = c(-5, 15), ylim = c(-5, 20), pch = 21, bg = "tomato", cex = 0.5,
     xlab = expression("X"[1]), ylab = expression("X"[1]))

arrows(center[1], center[2], arrows_xy[, 1], arrows_xy[, 2], lwd = 2, col = c("green", "orange"), length

legend("topright", legend = c("PC 1", "PC 2"), col = c("green", "orange"), pch = 15, bty = "n", cex = 0
```


PC 1
PC 2
X_1
X_1
−5
0
5
10
15
20