Assignment 2

Nguyen Xuan Binh

Homework Problem 2: PCA for Simulated Data

Simulate 100 observations from bivariate normal distribution with parameters:

$$\mu = \begin{pmatrix} 4 \\ 7 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 10 & 6 \\ 6 & 8 \end{pmatrix}.$$

```r
setwd(getwd())
```

---

a) Plot the data. Label the data points with the corresponding observation number.

```r
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

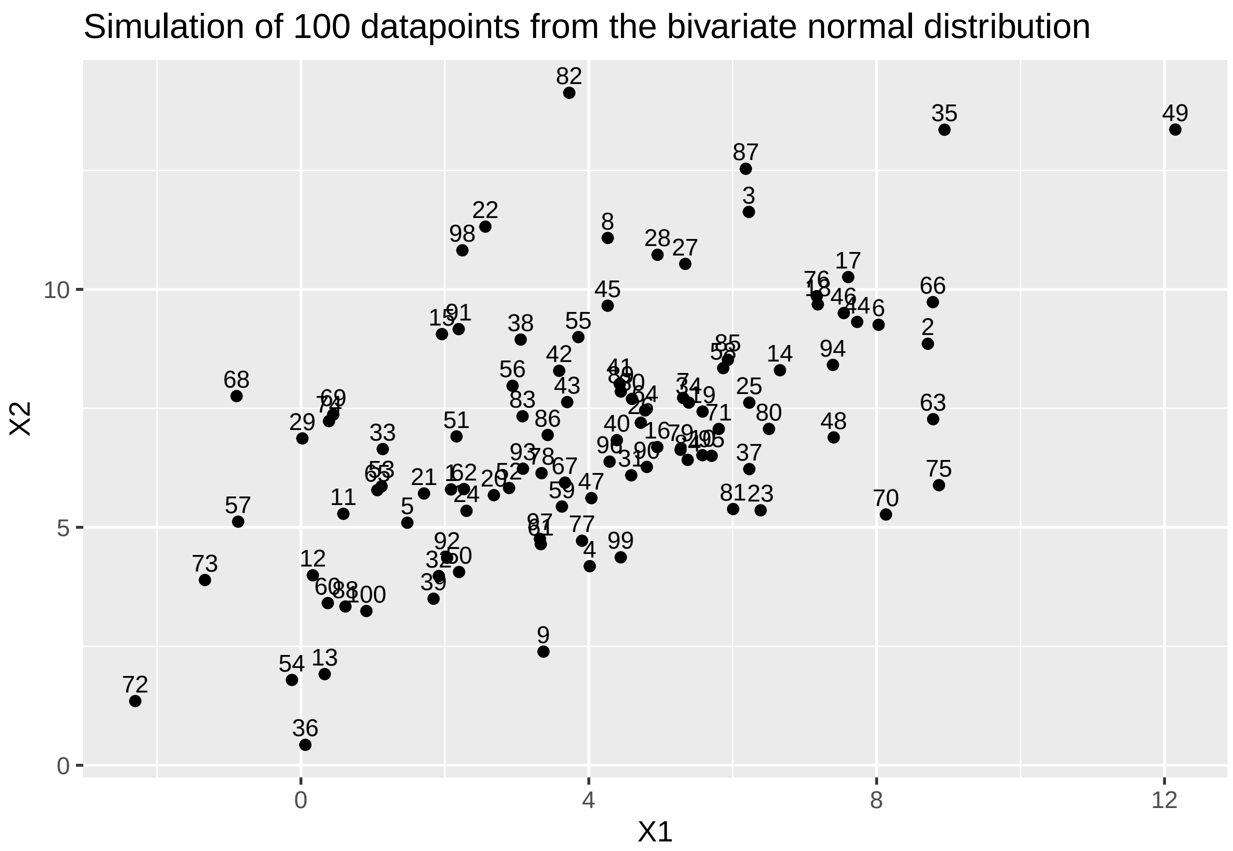
```r
library("mvtnorm")
```

```
## Warning: package 'mvtnorm' was built under R version 4.3.2
```

```r
# Set parameters for the bivariate normal distribution
mu <- c(4, 7)
sigma <- matrix(c(10, 6, 6, 8), nrow = 2, ncol = 2)

# Simulate 100 observations
set.seed(123)
multinormData <- data.frame(rmvnorm(n = 100, mean = mu, sigma = sigma))

# Plot data and label points with observation number
library(ggplot2)
ggplot(multinormData, aes(x = X1, y = X2)) +
  geom_point() + geom_text(hjust= 0.5, vjust=-0.5, label = 1:100, size=3) +
  ggtitle("Simulation of 100 datapoints from the bivariate normal distribution") +
  xlab("X1") + ylab("X2")
```



---

b) Perform the covariance based PCA transformation to the data set.

```
multinormData.PCA <- princomp(multinormData,cor=FALSE)

# names(multinormData.PCA)

cat("sdev\n")
```

```
## sdev
```

```
multinormData.PCA$sdev
```

```
##   Comp.1   Comp.2
## 3.308146 1.740956
```

```
cat("\nThe G matrix (columns are eigenvectors)")
```

```
##
## The G matrix (columns are eigenvectors)
```

```r
multinormData.PCA$loadings
```

```
##
## Loadings:
##    Comp.1 Comp.2
## X1  0.730  0.683
## X2  0.683 -0.730
##
##                Comp.1 Comp.2
## SS loadings       1.0    1.0
## Proportion Var    0.5    0.5
## Cumulative Var    0.5    1.0
```

```r
cat("\nThe Y matrix\n")
```

```
##
## The Y matrix
```

```r
multinormData.PCA$scores
```

```
##             Comp.1      Comp.2
##   [1,] -2.17827246 -0.46221980
##   [2,]  4.75262497  1.82729277
##   [3,]  4.82730975 -1.89483414
##   [4,] -1.87109223  2.03210706
##   [5,] -3.09951296 -0.36586466
##   [6,]  4.52410022  1.06874460
##   [7,]  1.49123853  0.33524180
##   [8,]  3.02122043 -2.83432943
##   [9,] -3.56750067  2.90523250
##  [10,]  0.86452524  1.39810492
##  [11,] -3.62112631 -1.10959379
##  [12,] -4.81319762 -0.45511335
##  [13,] -6.11064265  1.17427376
##  [14,]  2.87085593  0.83015426
##  [15,] -0.04140454 -2.93237148
##  [16,]  0.52268395  0.84562948
##  [17,]  4.90085754  0.04729255
##  [18,]  4.19996010  0.17892353
##  [19,]  1.49249564  0.72978018
##  [20,] -1.82513211  0.03249809
##  [21,] -2.51108335 -0.65593438
##  [22,]  1.94167798 -4.16997989
##  [23,]  0.66460325  2.79680870
##  [24,] -2.32834747  0.01317426
##  [25,]  2.09416553  1.03710195
##  [26,]  0.70481207  0.31540707
##  [27,]  3.43629048 -1.70119071
##  [28,]  3.28500283 -2.10370292
##  [29,] -2.95509685 -2.65577047
##  [30,]  0.95721105 -0.13433844
##  [31,] -0.14623075  1.03153253
##  [32,] -3.54489210  0.75074988
##  [33,] -2.29184702 -1.72932982
##  [34,]  1.48078864  0.46242341
##  [35,]  7.99286146 -1.30042475
##  [36,] -7.32165972  2.07528696
##  [37,]  1.14116684  2.05656183
##  [38,]  0.67973242 -2.10170256
##  [39,] -3.92387042  1.04947859
##  [40,]  0.21185361  0.35564027
##  [41,]  1.04944810 -0.48130137
##  [42,]  0.62180026 -1.25638523
##  [43,]  0.25447297 -0.70051348
##  [44,]  4.34813784  0.82114233
##  [45,]  2.04868382 -1.79523985
##  [46,]  4.33895340  0.55875050
##  [47,] -0.87855189  1.00377597
##  [48,]  2.45255079  2.37093962
##  [49,] 10.33837913  0.88795977
##  [50,] -3.28173211  0.88165260
##  [51,] -1.36167924 -1.22346275
##  [52,] -1.56824563  0.06762844
##  [53,] -2.83880849 -1.16849400
##  [54,] -6.52696056  0.95271376
##  [55,]  1.30006233 -1.59206270
##  [56,] -0.06658641 -1.46883587
##  [57,] -4.80189025 -1.98733456
##  [58,]  2.32567267  0.25871621
##  [59,] -1.29808735  0.85285978
##  [60,] -5.05998178  0.11289337
##  [61,] -2.05201952  1.22943550
##  [62,] -2.04173537 -0.34443680
##  [63,]  3.72373950  3.03474839
##  [64,]  0.92654191  0.16747559
##  [65,] -2.93612569 -1.15028626
##  [66,]  5.40046793  1.23571999
##  [67,] -0.92431835  0.51521694
##  [68,] -3.01538270 -3.92971589
##  [69,] -2.29969344 -2.72895597
##  [70,]  1.87719531  4.04974630
##  [71,]  1.40721612  1.15037616
##  [72,] -8.42040845 -0.21017026
##  [73,] -5.97652244 -1.40656950
##  [74,] -2.43670809 -2.66901014
##  [75,]  2.83551004  4.10313157
##  [76,]  4.30833993  0.04191688
##  [77,] -1.58515486  1.56892020
##  [78,] -1.02588149  0.14721945
##  [79,]  0.72047797  1.10717030
##  [80,]  1.91790458  1.62619918
##  [81,]  0.40356508  2.51595575
##  [82,]  4.71328101 -5.42729074
##  [83,] -0.40240209 -0.90595789
##  [84,]  0.64801884  1.33022979
##  [85,]  2.49262247  0.17560007
##  [86,] -0.41584453 -0.37998659
##  [87,]  5.41547621 -2.58597297
##  [88,] -4.92986497  0.33282071
##  [89,]  0.95018939 -0.35316740
##  [90,]  0.13207819  1.05068239
##  [91,]  0.20006836 -2.85008072
##  [92,] -3.19914110  0.54979377
##  [93,] -1.15139083 -0.09664384
##  [94,]  3.48514177  1.25059196
##  [95,]  0.94951798  1.49509384
##  [96,] -0.16953020  0.61601742
##  [97,] -1.98505480  1.13922854
##  [98,]  1.36818432 -4.02416677
##  [99,] -1.42943533  2.19026170
## [100,] -4.78168954  0.60071646
```

```
cat("\nThe sample mean\n")
```

```
##
## The sample mean
```

```
multinormData.PCA$center
```

```
##       X1       X2
## 3.992649 6.946177
```

```
cat("\nRelevant when cor=TRUE\n")
```

```
##
## Relevant when cor=TRUE
```

```
multinormData.PCA$scale
```

```
## X1 X2
##  1  1
```

```
cat("\nnumber of observations\n")
```

```
##
## number of observations
```

```
multinormData.PCA$n.obs
```

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

```
cat("\nfunction input\n")
```

```
##
## function input
```

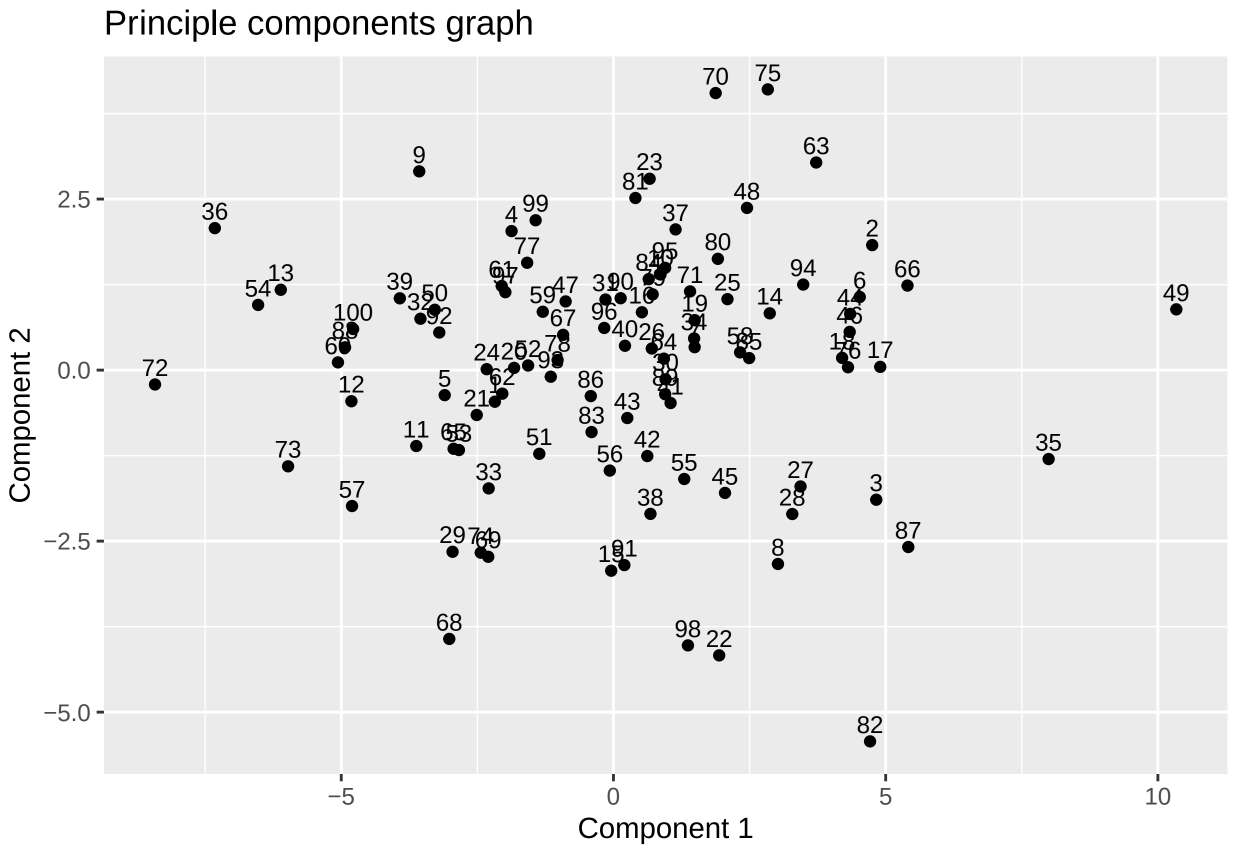
```
multinormData.PCA$calls
```

```
## NULL
```

---

c) Plot the score matrix. Use the same scale as in a) and label the data points with the corresponding observation number. Choose your scale (limits for the x- and y-axis) in a way that all the observations are visible in the figure.

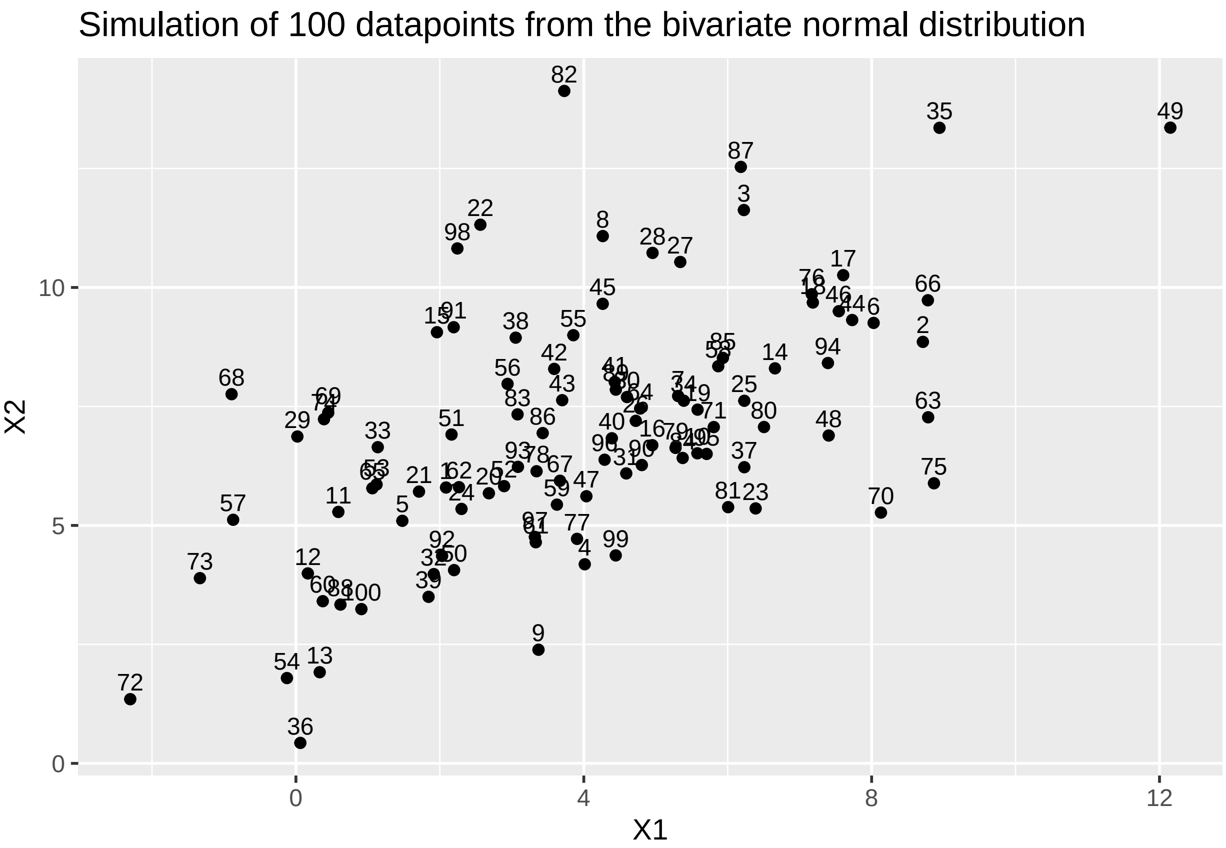
```
library(ggplot2)
ggplot(data.frame(multinormData.PCA$scores), aes(x = Comp.1, y = Comp.2)) +
  geom_point() + geom_text(hjust= 0.5, vjust=-0.5, label = 1:100, size=3) +
  ggtitle("Principle components graph") +
  xlab("Component 1") + ylab("Component 2")
```



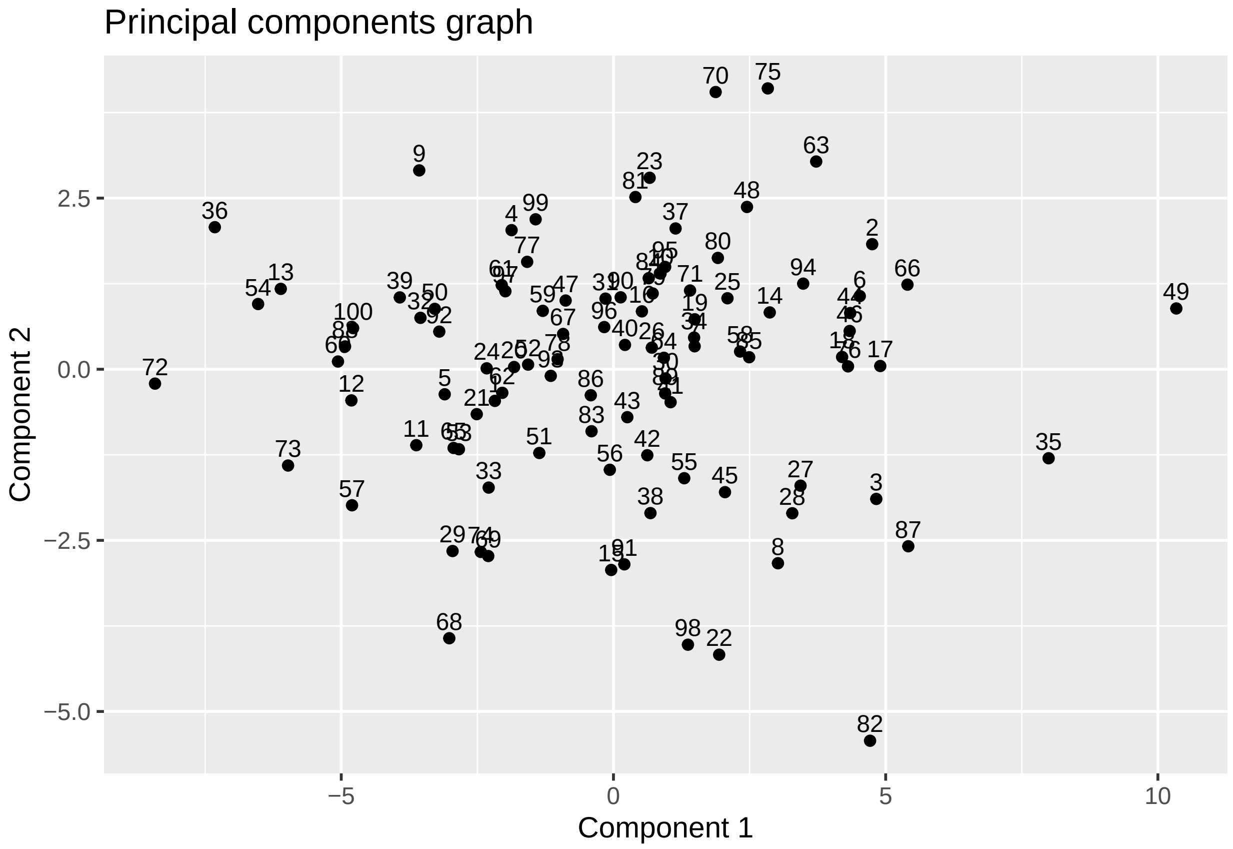
---

d) Compare the plots of a) and c) and describe the differences.

```
# Plot data and label points with observation number
ggplot(multinormData, aes(x = X1, y = X2)) +
  geom_point() + geom_text(hjust= 0.5, vjust=-0.5, label = 1:100, size=3) +
  ggtitle("Simulation of 100 datapoints from the bivariate normal distribution") +
  xlab("X1") + ylab("X2")
```



```
ggplot(data.frame(multinormData.PCA$scores), aes(x = Comp.1, y = Comp.2)) +
  geom_point() + geom_text(hjust= 0.5, vjust=-0.5, label = 1:100, size=3) +
  ggtitle("Principal components graph") +
  xlab("Component 1") + ylab("Component 2")
```



We can observe that the PCA transform has rotated the original data to be orthogonal to the axis. This is because the principal components are the eigenvectors of the covariance matrix of the data, which tries to maximize the difference between the datapoints. The first component has the largest eigenvalue, the second component has the second largest eigenvalue and so on. So in short, the difference between them is that the covariance between the original data is nonzero, while the covariance under the PCA transformed graph is zero or nearly zero.

---

e) Calculate the G and Y matrices without using any existing PCA functions. Note that the function princomp scales the covariance matrix with $1/n$ (instead of the usual $1/(n-1)$). Attach the R code to your solution.

The matrix G is the eigenvector matrix of the sample covariance matrix Σ The matrix Y is the sample PCA transformation: $Y = (X - 1_n \overline{x}^T)G$

Model reference

```
cat("\nThe G matrix (columns are eigenvectors)")
```

```
##
## The G matrix (columns are eigenvectors)
```

```
multinormData.PCA$loadings
```

```
##
## Loadings:
##    Comp.1 Comp.2
## X1  0.730  0.683
## X2  0.683 -0.730
##
##                Comp.1 Comp.2
## SS loadings       1.0    1.0
## Proportion Var    0.5    0.5
## Cumulative Var    0.5    1.0
```

```
cat("\nThe Y matrix\n")
```

```
##
## The Y matrix
```

```
multinormData.PCA$scores
```

```
##             Comp.1      Comp.2
##   [1,] -2.17827246 -0.46221980
##   [2,]  4.75262497  1.82729277
##   [3,]  4.82730975 -1.89483414
##   [4,] -1.87109223  2.03210706
##   [5,] -3.09951296 -0.36586466
##   [6,]  4.52410022  1.06874460
##   [7,]  1.49123853  0.33524180
##   [8,]  3.02122043 -2.83432943
##   [9,] -3.56750067  2.90523250
##  [10,]  0.86452524  1.39810492
##  [11,] -3.62112631 -1.10959379
##  [12,] -4.81319762 -0.45511335
##  [13,] -6.11064265  1.17427376
##  [14,]  2.87085593  0.83015426
##  [15,] -0.04140454 -2.93237148
##  [16,]  0.52268395  0.84562948
##  [17,]  4.90085754  0.04729255
##  [18,]  4.19996010  0.17892353
##  [19,]  1.49249564  0.72978018
##  [20,] -1.82513211  0.03249809
##  [21,] -2.51108335 -0.65593438
##  [22,]  1.94167798 -4.16997989
##  [23,]  0.66460325  2.79680870
##  [24,] -2.32834747  0.01317426
##  [25,]  2.09416553  1.03710195
##  [26,]  0.70481207  0.31540707
##  [27,]  3.43629048 -1.70119071
##  [28,]  3.28500283 -2.10370292
##  [29,] -2.95509685 -2.65577047
##  [30,]  0.95721105 -0.13433844
##  [31,] -0.14623075  1.03153253
##  [32,] -3.54489210  0.75074988
##  [33,] -2.29184702 -1.72932982
##  [34,]  1.48078864  0.46242341
##  [35,]  7.99286146 -1.30042475
##  [36,] -7.32165972  2.07528696
##  [37,]  1.14116684  2.05656183
##  [38,]  0.67973242 -2.10170256
##  [39,] -3.92387042  1.04947859
##  [40,]  0.21185361  0.35564027
##  [41,]  1.04944810 -0.48130137
##  [42,]  0.62180026 -1.25638523
##  [43,]  0.25447297 -0.70051348
##  [44,]  4.34813784  0.82114233
##  [45,]  2.04868382 -1.79523985
##  [46,]  4.33895340  0.55875050
##  [47,] -0.87855189  1.00377597
##  [48,]  2.45255079  2.37093962
##  [49,] 10.33837913  0.88795977
##  [50,] -3.28173211  0.88165260
##  [51,] -1.36167924 -1.22346275
##  [52,] -1.56824563  0.06762844
##  [53,] -2.83880849 -1.16849400
##  [54,] -6.52696056  0.95271376
##  [55,]  1.30006233 -1.59206270
##  [56,] -0.06658641 -1.46883587
##  [57,] -4.80189025 -1.98733456
##  [58,]  2.32567267  0.25871621
##  [59,] -1.29808735  0.85285978
##  [60,] -5.05998178  0.11289337
##  [61,] -2.05201952  1.22943550
##  [62,] -2.04173537 -0.34443680
##  [63,]  3.72373950  3.03474839
##  [64,]  0.92654191  0.16747559
##  [65,] -2.93612569 -1.15028626
##  [66,]  5.40046793  1.23571999
##  [67,] -0.92431835  0.51521694
##  [68,] -3.01538270 -3.92971589
##  [69,] -2.29969344 -2.72895597
##  [70,]  1.87719531  4.04974630
##  [71,]  1.40721612  1.15037616
##  [72,] -8.42040845 -0.21017026
##  [73,] -5.97652244 -1.40656950
##  [74,] -2.43670809 -2.66901014
##  [75,]  2.83551004  4.10313157
##  [76,]  4.30833993  0.04191688
##  [77,] -1.58515486  1.56892020
##  [78,] -1.02588149  0.14721945
##  [79,]  0.72047797  1.10717030
##  [80,]  1.91790458  1.62619918
##  [81,]  0.40356508  2.51595575
##  [82,]  4.71328101 -5.42729074
##  [83,] -0.40240209 -0.90595789
##  [84,]  0.64801884  1.33022979
##  [85,]  2.49262247  0.17560007
##  [86,] -0.41584453 -0.37998659
##  [87,]  5.41547621 -2.58597297
##  [88,] -4.92986497  0.33282071
##  [89,]  0.95018939 -0.35316740
##  [90,]  0.13207819  1.05068239
##  [91,]  0.20006836 -2.85008072
##  [92,] -3.19914110  0.54979377
##  [93,] -1.15139083 -0.09664384
##  [94,]  3.48514177  1.25059196
##  [95,]  0.94951798  1.49509384
##  [96,] -0.16953020  0.61601742
##  [97,] -1.98505480  1.13922854
##  [98,]  1.36818432 -4.02416677
##  [99,] -1.42943533  2.19026170
## [100,] -4.78168954  0.60071646
```

```
calculateG <- function(multinormData){
  covarianceMatrix <- cov(multinormData)
  G <- eigen(covarianceMatrix)$vectors
  return(G)
}

calculateY <- function(multinormData, G){
  n <- nrow(multinormData)
  meanData <- colMeans(multinormData)
  centered <- sweep(multinormData, 2, meanData, "-")
  Y <- as.matrix(centered) %*% (G)
  return(Y)
}

G <- calculateG(multinormData)
cat("The matrix G is\n")
```

```
## The matrix G is
```

```
print(G)
```

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

```
Y <- calculateY(multinormData, G)
cat("\nThe matrix Y is\n")
```

```
##
## The matrix Y is
```

```
print(Y)
```

```
##               [,1]        [,2]
##   [1,]  2.17827246 -0.46221980
##   [2,]  -4.75262497  1.82729277
##   [3,]  -4.82730975 -1.89483414
##   [4,]   1.87109223  2.03210706
##   [5,]   3.09951296 -0.36586466
##   [6,]  -4.52410022  1.06874460
##   [7,]  -1.49123853  0.33524180
##   [8,]  -3.02122043 -2.83432943
##   [9,]   3.56750067  2.90523250
##  [10,]  -0.86452524  1.39810492
##  [11,]   3.62112631 -1.10959379
##  [12,]   4.81319762 -0.45511335
##  [13,]   6.11064265  1.17427376
##  [14,]  -2.87085593  0.83015426
##  [15,]   0.04140454 -2.93237148
##  [16,]  -0.52268395  0.84562948
##  [17,]  -4.90085754  0.04729255
##  [18,]  -4.19996010  0.17892353
##  [19,]  -1.49249564  0.72978018
##  [20,]   1.82513211  0.03249809
##  [21,]   2.51108335 -0.65593438
##  [22,]  -1.94167798 -4.16997989
##  [23,]  -0.66460325  2.79680870
##  [24,]   2.32834747  0.01317426
##  [25,]  -2.09416553  1.03710195
##  [26,]  -0.70481207  0.31540707
##  [27,]  -3.43629048 -1.70119071
##  [28,]  -3.28500283 -2.10370292
##  [29,]   2.95509685 -2.65577047
##  [30,]  -0.95721105 -0.13433844
##  [31,]   0.14623075  1.03153253
##  [32,]   3.54489210  0.75074988
##  [33,]   2.29184702 -1.72932982
##  [34,]  -1.48078864  0.46242341
##  [35,]  -7.99286146 -1.30042475
##  [36,]   7.32165972  2.07528696
##  [37,]  -1.14116684  2.05656183
##  [38,]  -0.67973242 -2.10170256
##  [39,]   3.92387042  1.04947859
##  [40,]  -0.21185361  0.35564027
##  [41,]  -1.04944810 -0.48130137
##  [42,]  -0.62180026 -1.25638523
##  [43,]  -0.25447297 -0.70051348
##  [44,]  -4.34813784  0.82114233
##  [45,]  -2.04868382 -1.79523985
##  [46,]  -4.33895340  0.55875050
##  [47,]   0.87855189  1.00377597
##  [48,]  -2.45255079  2.37093962
##  [49,] -10.33837913  0.88795977
##  [50,]   3.28173211  0.88165260
##  [51,]   1.36167924 -1.22346275
##  [52,]   1.56824563  0.06762844
##  [53,]   2.83880849 -1.16849400
##  [54,]   6.52696056  0.95271376
##  [55,]  -1.30006233 -1.59206270
##  [56,]   0.06658641 -1.46883587
##  [57,]   4.80189025 -1.98733456
##  [58,]  -2.32567267  0.25871621
##  [59,]   1.29808735  0.85285978
##  [60,]   5.05998178  0.11289337
##  [61,]   2.05201952  1.22943550
##  [62,]   2.04173537 -0.34443680
##  [63,]  -3.72373950  3.03474839
##  [64,]  -0.92654191  0.16747559
##  [65,]   2.93612569 -1.15028626
##  [66,]  -5.40046793  1.23571999
##  [67,]   0.92431835  0.51521694
##  [68,]   3.01538270 -3.92971589
##  [69,]   2.29969344 -2.72895597
##  [70,]  -1.87719531  4.04974630
##  [71,]  -1.40721612  1.15037616
##  [72,]   8.42040845 -0.21017026
##  [73,]   5.97652244 -1.40656950
##  [74,]   2.43670809 -2.66901014
##  [75,]  -2.83551004  4.10313157
##  [76,]  -4.30833993  0.04191688
##  [77,]   1.58515486  1.56892020
##  [78,]   1.02588149  0.14721945
##  [79,]  -0.72047797  1.10717030
##  [80,]  -1.91790458  1.62619918
##  [81,]  -0.40356508  2.51595575
##  [82,]  -4.71328101 -5.42729074
##  [83,]   0.40240209 -0.90595789
##  [84,]  -0.64801884  1.33022979
##  [85,]  -2.49262247  0.17560007
##  [86,]   0.41584453 -0.37998659
##  [87,]  -5.41547621 -2.58597297
##  [88,]   4.92986497  0.33282071
##  [89,]  -0.95018939 -0.35316740
##  [90,]  -0.13207819  1.05068239
##  [91,]  -0.20006836 -2.85008072
##  [92,]   3.19914110  0.54979377
##  [93,]   1.15139083 -0.09664384
##  [94,]  -3.48514177  1.25059196
##  [95,]  -0.94951798  1.49509384
##  [96,]   0.16953020  0.61601742
##  [97,]   1.98505480  1.13922854
##  [98,]  -1.36818432 -4.02416677
##  [99,]   1.42943533  2.19026170
## [100,]   4.78168954  0.60071646
```

Note that the sign of the eigenvectors does not matter. Therefore, the matrix G produced here matches the referenced G matrix above by putting minus sign on the first component

---

f) Verify that the estimated scores and the loadings are equal (up to signs) in parts b) and e). Hint: If parts b) and e) are done correctly, the scores and loadings should be the same up to

heterogeneous sign changes.

Yes, they are equal, please scroll up to check part (b) and (e) again. The matrix G is the loadings matrix and the matrix Y is the scores matrix. Together, they make up the PCA transformation matrix, where the G * Y = PCA.

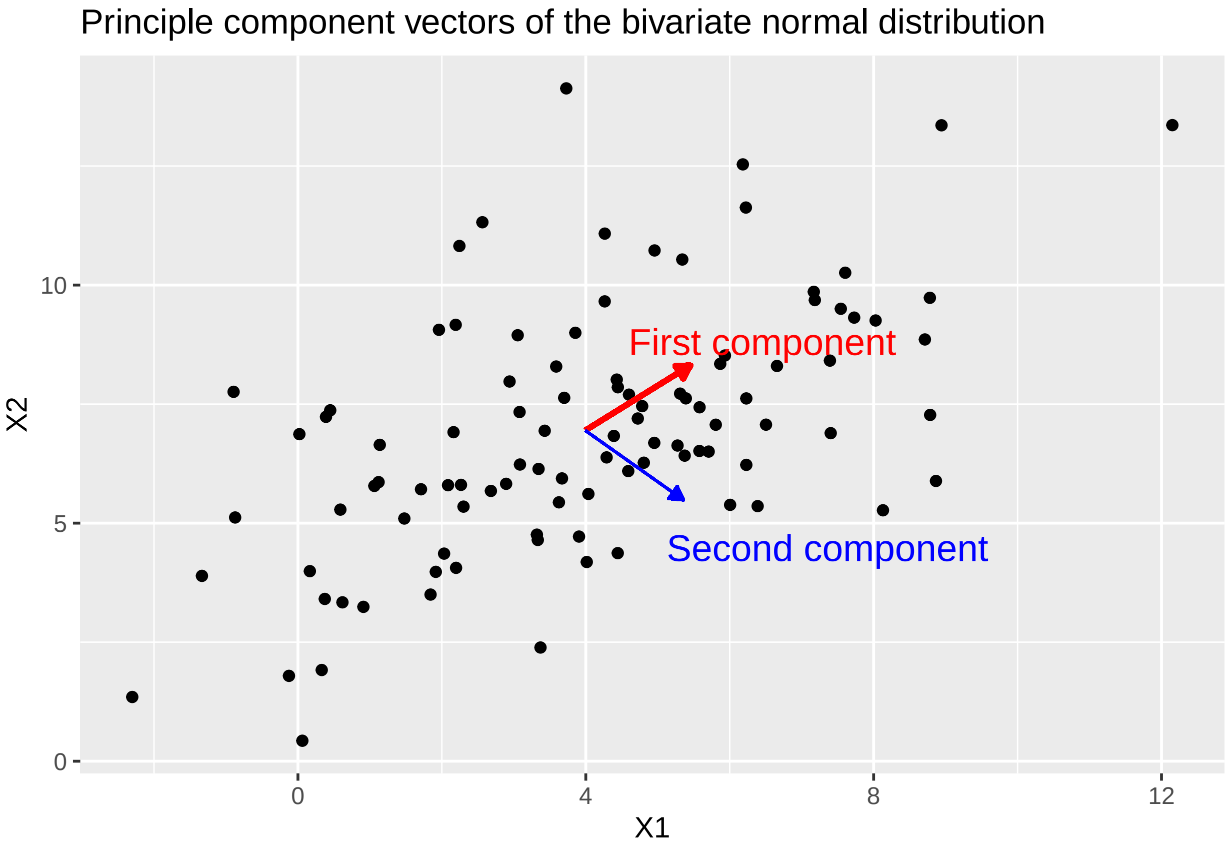
---

g) Plot the directions of the first and second principal component to the original data. The function arrows might be useful.

```
x <- multinormData.PCA$center["X1"]
y <- multinormData.PCA$center["X2"]
eigenVectorComponent1 <- loadings(multinormData.PCA)[1:2, 1]
eigenVectorComponent2 <- loadings(multinormData.PCA)[1:2, 2]

# Plot data and label points with observation number
ggplot(multinormData, aes(x = X1, y = X2)) +
  geom_point() +
  ggtitle("Principle component vectors of the bivariate normal distribution") +
  xlab("X1") + ylab("X2") +
  geom_segment(aes(x = x, y = y, xend = x + 2 * eigenVectorComponent1[1],
                   yend = y + 2 * eigenVectorComponent1[2]),
                arrow = arrow(length = unit(0.2, "cm"), type = "closed"),
                color = "red", linewidth = 1) +
  annotate("text", x = x + 2 * eigenVectorComponent1[1] + 1, y = y + 2 *
             eigenVectorComponent1[2] + 0.5, label = "First component",
           color = "red", size = 5) +
  geom_segment(aes(x = x, y = y, xend = x + 2 * eigenVectorComponent2[1],
                   yend = y + 2 * eigenVectorComponent2[2]),
                arrow = arrow(length = unit(0.2, "cm"), type = "closed"),
                color = "blue", lindwidth = 1) +
  annotate("text", x = x + 2 * eigenVectorComponent2[1] + 2, y = y + 2 *
             eigenVectorComponent2[2] - 1, label = "Second component",
           color = "blue", size = 5)
```

```
## Warning in geom_segment(aes(x = x, y = y, xend = x + 2 *
## eigenVectorComponent2[1], : Ignoring unknown parameters: ‘lindwidth‘
```


Principle component vectors of the bivariate normal distribution
First component
Second component
X2
10
5
0
X1
0
4
8
12