Practice: PCA

Jay Wei

2023-10-04

Contents

1 Exercise 1

The purpose of this problem is to examine the effect that different correlations have on the outcome of the PCA. To make this easier, suppose x has a bivariate normal distribution with $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\sigma_{11} = 1$, and $\sigma_{22} = 1$.

For $\sigma_{12} = -0.99, -0.9, -0.5, 0, 0.5, 0.9$, and 0.99 (remember that $\sigma_{12} = \rho_{12}$ because the variances are equal to 1), complete the following:

1.1 Question a

Simulate 1,000 observations from the bivariate normal where a seed number of 8128 is set right before each data simulation.

In Exercise 1, we use $\rho_{12} = 0.99$ for illustration purpose. For more, please check Exercise 2.

```
library(mvtnorm)

mu <- c(0, 0)
rho12 <- 0.99
sigma <- matrix(data = c(1, rho12,
                         rho12, 1),
                nrow = 2, ncol = 2,
                byrow = TRUE)
P <- cov2cor(V = sigma)
P
```

```
##      [,1] [,2]
## [1,] 1.00 0.99
## [2,] 0.99 1.00
```

```
N <- 1000
set.seed(8128)
X <- rmvnorm(n = N, mean = mu, sigma = sigma)
head(X)
```

```
##             [,1]       [,2]
## [1,]  0.08789969  0.1239771
## [2,]  0.64654881  0.5685544
## [3,] -1.51649613 -1.6042760
## [4,]  0.13794791  0.2297071
## [5,] -1.14282093 -1.0670453
## [6,]  2.41497659  2.2771057
```

1.2 Question b

Use `princomp()` with `cor = TRUE` to find the estimated eigenvalues and eigenvectors from the correlation matrix.

```r
pca.save <- princomp(x = X, cor = TRUE, scores = FALSE)
summary(pca.save, loadings = TRUE, cutoff = 0.0)
```

```
## Importance of components:
##                            Comp.1      Comp.2
## Standard deviation     1.4105423 0.101834613
## Proportion of Variance 0.9948149 0.005185144
## Cumulative Proportion  0.9948149 1.000000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,]  0.707 -0.707
```

1.3 Question c

Interpret the PCs.

PC #1 can be interpreted as a combination of variable one and two; while PC #2 can be interpreted as a contrast between variable one and two.

1.4 Question d

How many PCs are necessary?

Only one – PC #1 already accounts for 99% of the total variance! You shouldn't be surprised at the results as the we set $\rho_{12} = 0.99$ in our simulation.

1.5 Question e

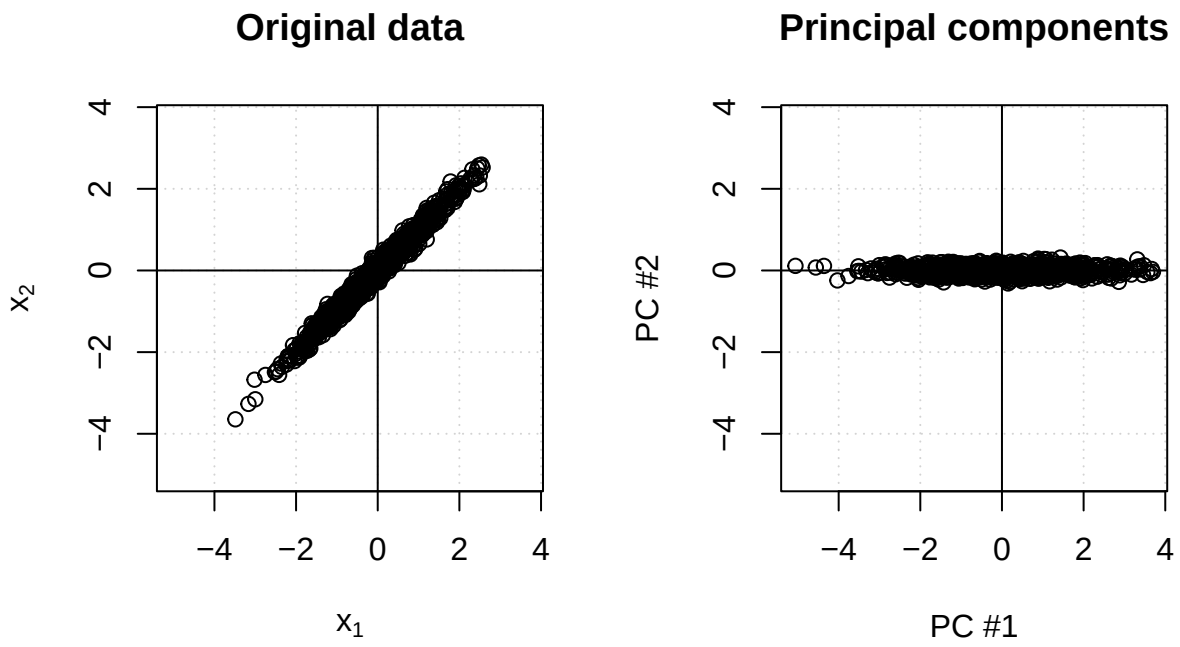
Create separate scatter plots of the data and the PC scores, but use one overall x-axis and y-axis set of limits. Describe the relationship between these plots for each ρ_{12}.

```r
pca.save$scale <- apply(X = X, MARGIN = 2, FUN = sd)
score.save <- predict(pca.save, newdata = X)
head(score.save)
```

```
##          Comp.1      Comp.2
## [1,]  0.1846096 -0.02391515
## [2,]  0.9007901  0.05776389
## [3,] -2.1943678  0.06367319
## [4,]  0.2957979 -0.06362483
## [5,] -1.5441590 -0.05285472
## [6,]  3.3828429  0.10136960
```

```r
par(mfrow = c(1,2)) #One row and two columns of plots
par(pty = "s")
common.limits <- c(min(score.save, X), max(score.save, X))
```

```r
plot(x = X[,1], y = X[,2],
     xlab = expression(x[1]),
     ylab = expression(x[2]),
     main = "Original data",
     xlim = common.limits, ylim = common.limits,
     panel.first = grid(col = "lightgray", lty = "dotted"))
abline(h = 0)
abline(v = 0)

plot(x = score.save[,1], y = score.save[,2],
     xlab = "PC #1", ylab = "PC #2",
     main = "Principal components",
     xlim = common.limits, ylim = common.limits,
     panel.first = grid(col = "lightgray", lty = "dotted"))
abline(h = 0)
abline(v = 0)
```



It looks like the x and y-axes have been rotated so that all of the variability in the data in represented in only one dimension!

1.6 Question f

Relation your answers in (c) to (e) to the values of σ_{12}

One shouldn't be surprised as we set a strong positive linear relationship with $\sigma_{12} = 0.99$ in our simulation.

In question (c), we find that a combination of variable one and two accounts for a much more proportion of total variance than a contrast of varable one and two.

In question (d), we find only one PC is necessary as a strong correlation exists for these two variables.

In question (e), we find all of the variability in the data in represented in only one dimension. One can sensiblly relate these results to the choice of σ_{12}

2 Exercise 2

2.1 Question a

```r
mu <- c(0, 0)
rho12 <- c(-0.99, -0.9, -0.5, 0, 0.5, 0.9, 0.99)
sigma.list <- list()

for (i in 1:length(rho12)) {
  rho <- rho12[i]
  cat("Correlation Matrix for rho =", rho, ":\n")
  sigma <- matrix(data = c(1, rho, rho, 1),
                  nrow = 2, ncol = 2, byrow = TRUE)
  P <- cov2cor(V = sigma)
  print(P)
  cat("-------------\n")

  # Store the current correlation matrix in the list
  sigma.list[[i]] <- P
}
```

```
## Correlation Matrix for rho = -0.99 :
##       [,1]  [,2]
## [1,]  1.00 -0.99
## [2,] -0.99  1.00
## -------------
## Correlation Matrix for rho = -0.9 :
##      [,1] [,2]
## [1,]  1.0 -0.9
## [2,] -0.9  1.0
## -------------
## Correlation Matrix for rho = -0.5 :
##      [,1] [,2]
## [1,]  1.0 -0.5
## [2,] -0.5  1.0
## -------------
## Correlation Matrix for rho = 0 :
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    1
## -------------
## Correlation Matrix for rho = 0.5 :
##      [,1] [,2]
## [1,]  1.0  0.5
## [2,]  0.5  1.0
## -------------
## Correlation Matrix for rho = 0.9 :
##      [,1] [,2]
## [1,]  1.0  0.9
## [2,]  0.9  1.0
```

```
## -------------
## Correlation Matrix for rho = 0.99 :
##      [,1] [,2]
## [1,] 1.00 0.99
## [2,] 0.99 1.00
## -------------
```

```r
N <- 1000
set.seed(8128)
X <- list()
for (i in 1:length(rho12)){
  X[[i]] <- rmvnorm(n = N, mean = mu, sigma = sigma.list[[i]])
}
```

2.2 Question b

```r
pca.store <- list()

for (i in 1:length(rho12)){
  rho <- rho12[i]
  pca.store[[i]] <- princomp(x= X[[i]], cor = TRUE, scores = FALSE)
  cat("PCA results for rho =", rho, ":\n")
  summary(pca.store[[i]], loadings=TRUE, cutoff = 0.0)|> print()
  cat("=============================================\n")
}
```

```
## PCA results for rho = -0.99 :
## Importance of components:
##                            Comp.1      Comp.2
## Standard deviation     1.4108003 0.098195845
## Proportion of Variance 0.9951788 0.004821212
## Cumulative Proportion  0.9951788 1.000000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,] -0.707  0.707
## =============================================
## PCA results for rho = -0.9 :
## Importance of components:
##                            Comp.1     Comp.2
## Standard deviation     1.3768049 0.32312280
## Proportion of Variance 0.9477958 0.05220417
## Cumulative Proportion  0.9477958 1.00000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,] -0.707  0.707
## =============================================
```

```
## PCA results for rho = -0.5 :
## Importance of components:
##                           Comp.1    Comp.2
## Standard deviation     1.2283430 0.7008377
## Proportion of Variance 0.7544133 0.2455867
## Cumulative Proportion  0.7544133 1.0000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,] -0.707  0.707
## ==========================================
## PCA results for rho = 0 :
## Importance of components:
##                           Comp.1    Comp.2
## Standard deviation     1.0140782 0.9857208
## Proportion of Variance 0.5141773 0.4858227
## Cumulative Proportion  0.5141773 1.0000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,] -0.707  0.707
## ==========================================
## PCA results for rho = 0.5 :
## Importance of components:
##                           Comp.1    Comp.2
## Standard deviation     1.233707 0.6913509
## Proportion of Variance 0.761017 0.2389830
## Cumulative Proportion  0.761017 1.0000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,]  0.707 -0.707
## ==========================================
## PCA results for rho = 0.9 :
## Importance of components:
##                           Comp.1     Comp.2
## Standard deviation     1.3796330 0.31082584
## Proportion of Variance 0.9516936 0.04830635
## Cumulative Proportion  0.9516936 1.00000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,]  0.707 -0.707
## ==========================================
## PCA results for rho = 0.99 :
## Importance of components:
##                           Comp.1      Comp.2
## Standard deviation     1.4109334 0.096264880
```

```
## Proportion of Variance 0.9953665 0.004633464
## Cumulative Proportion  0.9953665 1.000000000
##
## Loadings:
##      Comp.1 Comp.2
## [1,]  0.707  0.707
## [2,]  0.707 -0.707
## ========================================
```

2.3 Question c & d

There are some interesting findings here:

1. With a strong correlation (e.g. $\rho_{12} = -0.99, -0.9, 0.9, 0.99$), we only need one PC. However as the correlation decreases, we may need more than one PC to account for a large amount of variety. For example, when $\rho_{12} = \pm 0.5$, PC #1 only accounts for about 75% of the total variance. When $\rho_{12} = 0$, PCA doesn't work well. Again, PCA tries to find a combination of the original correlated variables such that the new variables (PCs) are uncorrelated. What's the meaning of doing PCA when the original variables are already uncorrelated?

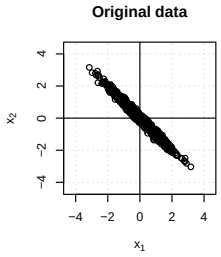
2. For positive correlation, PC #1 is a combination of the original variables; while for negative correltaion, PC #1 is a contrast of the original variables.

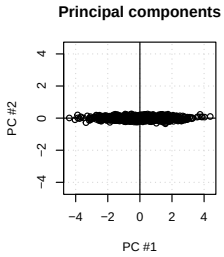
2.4 Question e

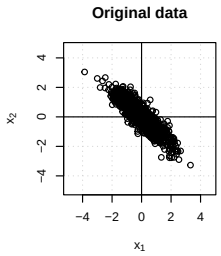
```
score.save.list <- list()
for (i in 1:length(rho12)){
  pca.store[[i]]$scale <- apply(X = X[[i]], MARGIN = 2, FUN = sd)
  score.save.list[[i]] <- predict(pca.store[[i]], newdata = X[[i]])
}
```

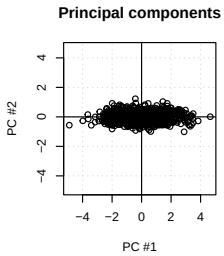
```
par(mfrow = c(length(rho12),2))
par(pty = "s")

for (i in 1:length(rho12)){
  rho <- rho12[i]
  common.limits <- c(min(score.save.list[[i]], X[[i]]),
                     max(score.save.list[[i]], X[[i]]))

  plot(x = X[[i]][,1], y = X[[i]][,2],
       xlab = expression(x[1]),
       ylab = expression(x[2]),
       main = "Original data",
       xlim = common.limits, ylim = common.limits,
       panel.first = grid(col = "lightgray", lty = "dotted"))
  abline(h = 0)
  abline(v = 0)

  plot(x = score.save.list[[i]][,1], y = score.save.list[[i]][,2],
```

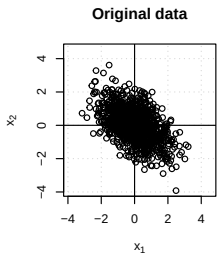
```
        xlab = "PC #1", ylab = "PC #2",
        main = "Principal components",
        xlim = common.limits, ylim = common.limits,
        panel.first = grid(col = "lightgray", lty = "dotted"))
  abline(h = 0)
  abline(v = 0)

}
```

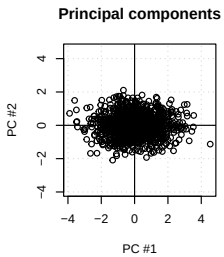

Original data

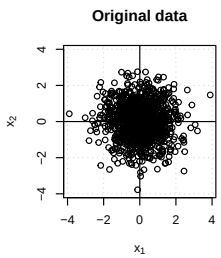

Principal components

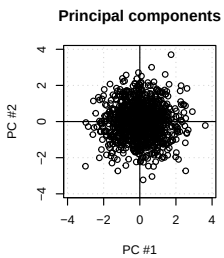

Original data

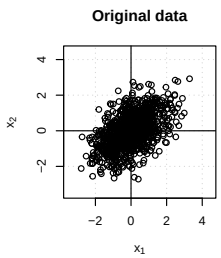

Principal components

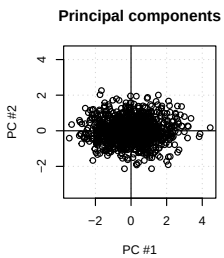

Original data

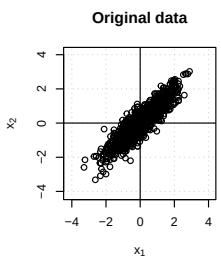

Principal components

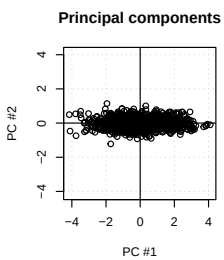

Original data

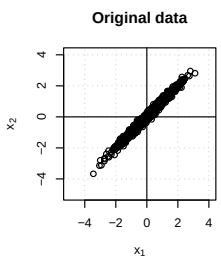

Principal components

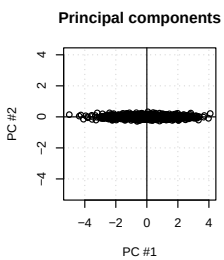

Original data


Principal components


Original data


Principal components


Original data


Principal components