HW-11

Dennis Goldenberg

2024-04-25

1. PCA

```
#read in data
data = as.data.frame(cbind(c(2.7,0.75, 2.25, 1.9, 4.0, 2.3),
                           c(2.5,1,4,2.25,2.5, 2.7)))
colnames(data) <- c("X1", "X2")
```

a. Contructing covariance matrix

I first find the means of the two variables:

```
X_1m = mean(data$X1)
X_2m = mean(data$X2)
sprintf("mean of X_1: %.4f", X_1m)
```

```
## [1] "mean of X_1: 2.3167"
```

```
sprintf("mean of X_2: %.4f", X_2m)
```

```
## [1] "mean of X_2: 2.4917"
```

I subtract these means from the original data to get the centered data:

```
data$X1 <- data$X1 - X_1m
data$X2 <- data$X2 - X_2m
data
```

```
##            X1           X2
## 1  0.38333333  0.008333333
## 2 -1.56666667 -1.491666667
## 3 -0.06666667  1.508333333
## 4 -0.41666667 -0.241666667
## 5  1.68333333  0.008333333
## 6 -0.01666667  0.208333333
```

```
var_covar <- cov(data)
var_covar
```

```
##           X1        X2
## X1 1.1226667 0.4701667
## X2 0.4701667 0.9204167
```

b. Finding eigenvectors, eigenvalues

I find the eigenvalues by solving the following equation:

$$\det\{\Sigma_x - \lambda I\} = 0$$

Therefore:

$$\begin{aligned}
\det\{\Sigma_x - \lambda I\} = 0 &\rightarrow (1.123 - \lambda)(.920 - \lambda) - .47^2 = 0 \\
&\rightarrow 1.12267(.92042) - (1.12267 + .92042)\lambda + \lambda^2 - .47017^2 = 0 \\
&\rightarrow \lambda^2 - 2.043\lambda + .8124 = 0 \\
&\rightarrow \lambda = \frac{2.04309 \pm \sqrt{2.04309^2 - 4 * .81227}}{2} \\
&\rightarrow \lambda = \mathbf{1.5025}, \mathbf{0.5406}
\end{aligned}$$

For the first eigenvector, I know that $(A - 1.5025I)v_1 = \overrightarrow{0}$. Therefore, I solve for the vector v_1:

$$\begin{aligned}
&(A - 1.5025I)v_1 = \overrightarrow{0} \\
&\rightarrow \begin{pmatrix} -.37983 & .47017 \\ .47017 & -.58208 \end{pmatrix} v_1 = \overrightarrow{0} \\
&\rightarrow -.37983v_{1,1} + .47017v_{1,2} = 0 \text{ and } .47017v_{1,1} - .58208v_{1,2} = 0 \\
&\rightarrow -.37983v_{1,1} + .47017v_{1,2} = 0 \text{ and } .47017\left(-.37983v_{1,1} + .47017v_{1,2}\right) + .37983\left(.47017v_{1,1} - .58208v_{1,2}\right) = 0 \\
&\rightarrow -.37983v_{1,1} + .47017v_{1,2} = 0 \text{ and } 0 + 0 = 0 \\
&\rightarrow v_{1,2} = \frac{.37983}{.47017}v_{1,1}
\end{aligned}$$

I choose $v_1 = \begin{bmatrix} .47017 & .37983 \end{bmatrix}$ to fit that description. However, it has to be a unit eigenvector, so:

$$u_1 = \begin{bmatrix} \frac{.47017}{\sqrt{.47017^2 + .37983^2}} & \frac{.37983}{\sqrt{.47017^2 + .37983^2}} \end{bmatrix}^T = \begin{bmatrix} .7779 & .6284 \end{bmatrix}^T$$

Via the spectral decomposition theorem, I know that u_2 will be perpendicular to u_1. It has the same unit norm condition as u_1. I use a similar process as before to solve for u_2:

$$\begin{aligned}
&(A - .5406I)v_2 = \overrightarrow{0} \\
&\rightarrow \begin{pmatrix} .58207 & .47017 \\ .47017 & .37982 \end{pmatrix} v_2 = \overrightarrow{0} \\
&\rightarrow .58207v_{2,1} + .47017v_{2,2} = 0 \\
&\rightarrow v_{2,2} = \frac{-.58207}{.47017}v_{2,1}
\end{aligned}$$

I choose $v_2 = \begin{bmatrix} -.47017 & .58207 \end{bmatrix}$ to match this condition. I apply the unit norm condition to get the unit eigenvector:

$$u_2 = \begin{bmatrix} \frac{-.47017}{\sqrt{.47017^2 + .58207^2}} & \frac{.58207}{\sqrt{.47017^2 + .58207^2}} \end{bmatrix}^T = \begin{bmatrix} -.6284 & .7779 \end{bmatrix}^T$$

c. Finding principal components

Note the principal components are linear combinations of the original variables, or $Z_1 = .7779(X_1 - \bar{x}_1) + .6284(X_2 - \bar{X}_2)$, and $Z_2 = -.6284(X_1 - \bar{x}_1) + .7779(X_2 - \bar{x}_2)$. I calculate the principal component scores:

```
Z_1 <- .7779 * data$X1 + .6284 * data$X2
Z_2 <- -.6284 * data$X1 + .7779 * data$X2
pc_scores <- as.data.frame(cbind(Z_1, Z_2))
pc_scores
```

```
##          Z_1         Z_2
## 1  0.3034317 -0.23440417
## 2 -2.1560733 -0.17587417
## 3  0.8959767  1.21522583
## 4 -0.4759883  0.07384083
## 5  1.3147017 -1.05132417
## 6  0.1179517  0.17253583
```

If I calculate the variance of each of the pc_scores:

```
sprintf("Variance, Z_1: %.4f", var(Z_1))
```

```
## [1] "Variance, Z_1: 1.5025"
```

```
sprintf("Variance, Z_: %.4f", var(Z_2))
```

```
## [1] "Variance, Z_: 0.5406"
```

The variance is just equal to the original eigenvalues.

d. Calculating PVE (percent of variation explained)

THe percent of the variation explained can be calcualted for both principal components:

$$PVE_{\text{PC1}} = \frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{1.5025}{1.5025 + .5406} = .7354 \text{ or } 73.54\%$$
$$PVE_{\text{PC2}} = \frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{.5406}{1.5025 + .5406} = .2646 \text{ or } 26.46\%$$

e. Repeat steps b-d in R

I get the eigenvalues and eigenvectors using the 'eigen' function:

```
eval_1 <- eigen(var_covar)$values
eval_1
```

```
## [1] 1.5024605 0.5406228
```

```
evec_1 <- eigen(var_covar)$vectors
colnames(evec_1) <- c("PC1", "PC2")
evec_1
```

```
##             PC1        PC2
## [1,] -0.7779057  0.6283810
## [2,] -0.6283810 -0.7779057
```

Now, I do the same thing using prcomp:

```
eval_2 <- prcomp(data)$sdev^2
eval_2
```

```
## [1] 1.5024605 0.5406228
```

```
evec_2 <- prcomp(data)$rotation
evec_2
```

```
##           PC1        PC2
## X1 -0.7779057 -0.6283810
## X2 -0.6283810  0.7779057
```

Then, I accomplish it using SVD:

```r
eval_3 <- svd(var_covar)$d
evec_3 <- svd(var_covar)$v
eval_3
```

```
## [1] 1.5024605 0.5406228
```

```r
colnames(evec_3) <- c("PC1", "PC2")
evec_3
```

```
##             PC1        PC2
## [1,] -0.7779057 -0.6283810
## [2,] -0.6283810  0.7779057
```

In each of these cases, the pc-scores are the following:

```r
Z_1_alt <- evec_3[1,1]*data$X1 + evec_3[1,2]*data$X2
Z_2_alt <- evec_3[2,1]*data$X1 + evec_3[2,2]*data$X2
pc_scores_alt <- as.data.frame(cbind(Z_1_alt, Z_2_alt))
pc_scores_alt
```

```
##      Z_1_alt     Z_2_alt
## 1 -0.3034337 -0.23439685
## 2  2.1560540 -0.17591238
## 3 -0.8959477  1.21523316
## 4  0.4759861  0.07383155
## 5 -1.3147111 -1.05129219
## 6 -0.1179476  0.17253670
```

The proportion of variance explained is:

```r
pVE <- eval_3/sum(eval_3)
pVE
```

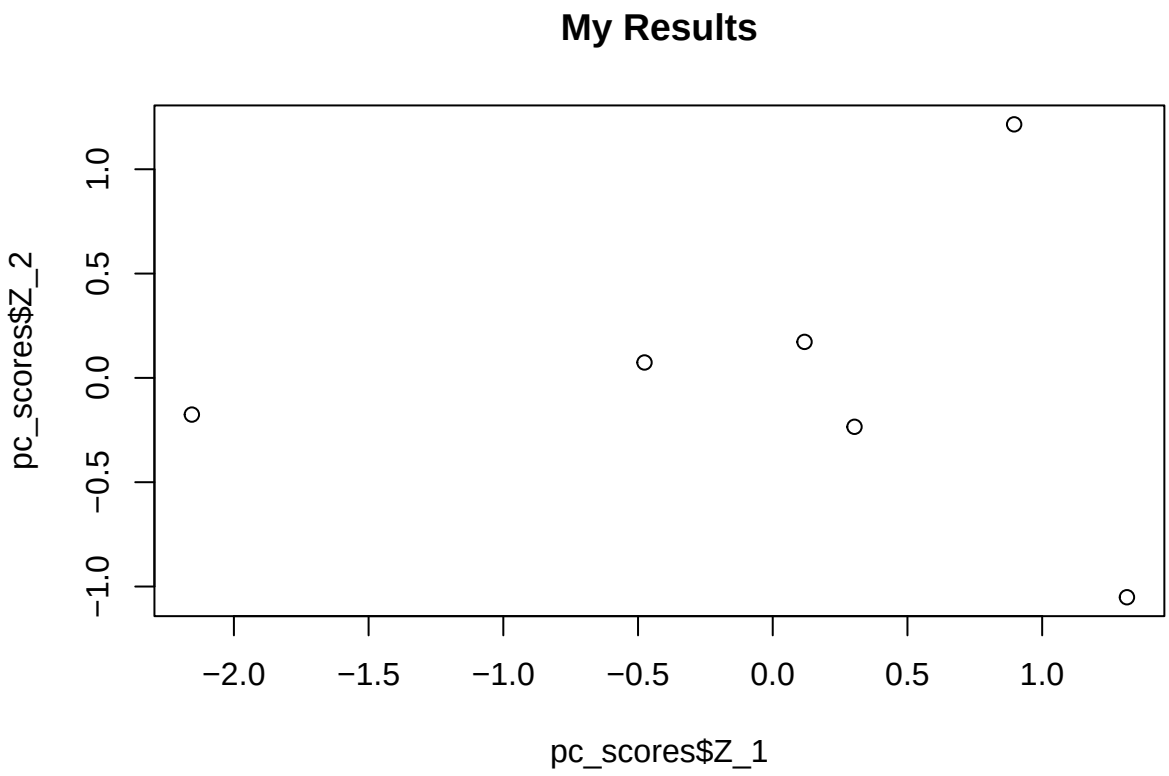
```
## [1] 0.7353888 0.2646112
```

f. Compare b-d, e results

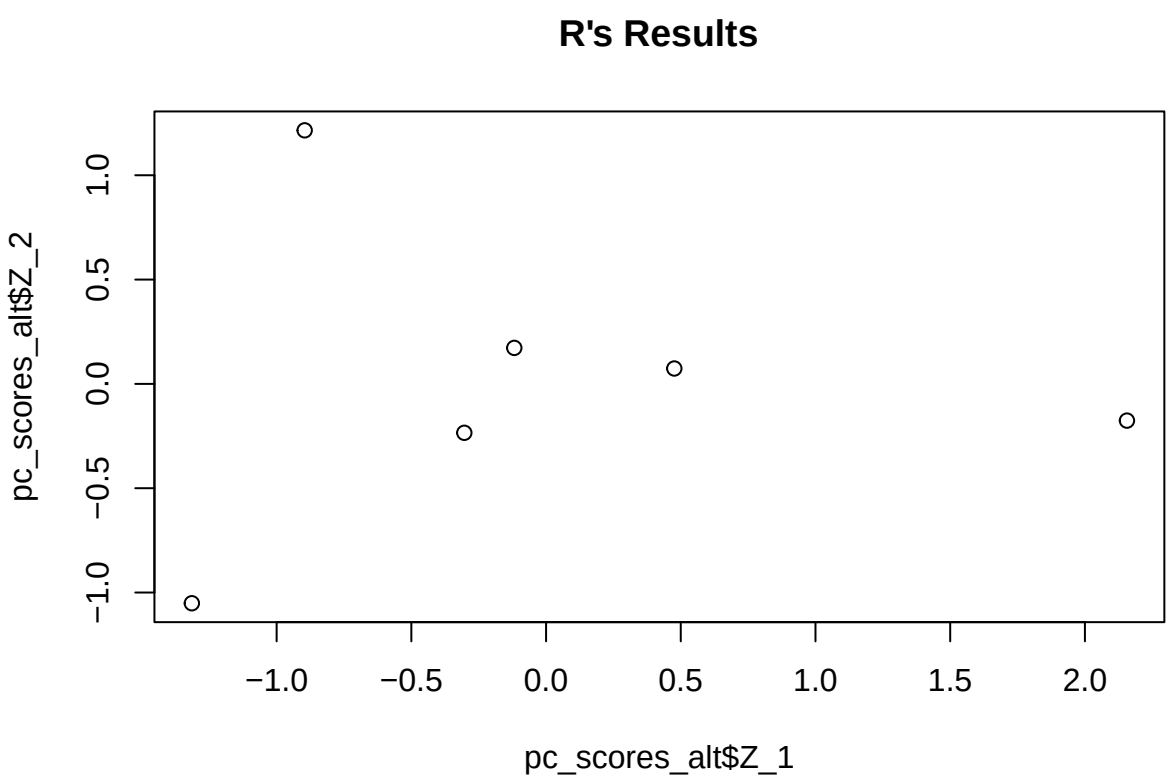
My results and the results from R are exactly the same, with the one exception that the signs of all of the principal component loadings are of opposite parity. This does not effect the eigenvalues or proportion of variance explained, and actually, either is correct.

g. Plotting two PCs

```r
plot(pc_scores$Z_1, pc_scores$Z_2)
title("My Results")
```




```
plot(pc_scores_alt$Z_1, pc_scores_alt$Z_2)
title("R's Results")
```



2. Hierarchical Clustering

a. Building dendogram

b. Proportion of variation for 2 groups

c. Use R for a-b

3. K-means Clustering

a. Find two clusters manually

b. Use R to randomly perform K-means using 2 clusters

4. Fun with PCA

a. Calculating 1st principal component score, Obs. 1

I follow the formula for principal component score:

$$\begin{aligned} z_{1,1} &= \sum_{j=1}^{4} x_{1,j} u_{j,1} \\ &= 1.2426 * .5359 + .7828 * .5832 - .5209 * .2782 - .0034 * .5434 \\ &= \mathbf{.9757} \end{aligned}$$

b. Calcualting 2nd principal component score, Obs. 2

I follow the same formula from before:

$$\begin{aligned} z_{2,2} &= \sum_{j=1}^{4} x_{2,j} u_{j.2} \\ &= .5079 * -.4182 + 1.1068 * -.1880 - 1.2118 * .8728 + 2.4842 * .1673 \\ &= \mathbf{-1.0625} \end{aligned}$$

c. Approximating $x_{1,4}$ by first two PCs

I incorporate the 1st 2 principal components for the approximation of the 4th feature on the 1st observation:

$$\begin{aligned} x_{1,4} &\approx z_{1,1} * u_{4,1} + z_{1,2} * u_{4,2} \\ &= .9757 * .5434 + z_{1,2} * .1673 \end{aligned}$$

I calculate the pc 2 score for the 1st observation:

$$\begin{aligned} z_{1,2} &= \sum_{j=1}^{4} x_{1,j} u_{j,2} \\ &= 1.2426 * -.4182 + .7828 * -.188 - .5209 * .8728 + -.0034 * .1673 \\ &= -1.122 \end{aligned}$$

Therefore, $x_{1,4} \approx .9757 * .5434 - 1.122 * .1673 = \mathbf{.3425}$.

d. Approximation error of $x_{1,3}$ by first two PCs

e. Distance between first observation, approximation by first two Pcs