Actividad1.6

2022-11-10

Parte A

De los siguientes datos:

x1: 2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1

x2: 2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9

- Obtenga una matriz de datos centrados en sus medias.

```
x1 = c(2.5, 0.5, 2.2, 1.9, 3.1, 2.3, 2, 1, 1.5, 1.1)
x2 = c(2.4, 0.7, 2.9, 2.2, 3.0, 2.7, 1.6, 1.1, 1.6, 0.9)
M = data.frame(x1,x2)
M1 = data.frame(c(rep(mean(x1), 10)),c(rep(mean(x2), 10)))
M2 = M-M1
M2
```

```
##       x1    x2
## 1   0.69  0.49
## 2  -1.31 -1.21
## 3   0.39  0.99
## 4   0.09  0.29
## 5   1.29  1.09
## 6   0.49  0.79
## 7   0.19 -0.31
## 8  -0.81 -0.81
## 9  -0.31 -0.31
## 10 -0.71 -1.01
```

- Obtenga la matriz de varianza-covarianza de la matriz de datos centrados

```
mcov = cov(M2)
mcov
```

```
##           x1        x2
## x1 0.6165556 0.6154444
## x2 0.6154444 0.7165556
```

- Obtenga los valores propios y vectores propios de la matriz de varianza-covarianza de la matriz de datos centrados.

```
Mvalues = eigen(mcov)$values
Mvectors = eigen(mcov)$vectors
Mvalues
```

```
## [1] 1.2840277 0.0490834
```

```
Mvectors
```

```
##           [,1]       [,2]
## [1,] 0.6778734 -0.7351787
## [2,] 0.7351787  0.6778734
```

$Y_1 = 0.68 X_1$ -0.74 X_2

$Y_2 = 0.74 X_1$ + 0.68 X_2

- Correlación entre Y_1, y X_1, X_2

```
# Raiz cuadrada de lambda 1 * el primer eigen vector/varianza de 1,1 de los datos centrada
sqrt(Mvalues[1])*Mvectors[1,1] / sqrt(mcov[1,1])
```

```
## [1] 0.9782496
```

```
sqrt(Mvalues[1])*Mvectors[2,1] / sqrt(mcov[2,2])
```

```
## [1] 0.9841361
```

- Correlación entre Y_2, y X_1, X_2

```
# Raiz cuadrada de lambda 1 * el primer eigen vector/varianza de 1,1 de los datos centrada
sqrt(Mvalues[2])*Mvectors[1,2] / sqrt(mcov[1,1])
```

```
## [1] -0.2074312
```

```
sqrt(Mvalues[2])*Mvectors[2,2] / sqrt(mcov[2,2])
```

```
## [1] 0.1774153
```

- Obtenga las matrices transpuestas de los vectores propios y la traspuesta de la matriz de datos centrados.

```
tVec = t(Mvectors)
tM2 = t(M2)
tVec
```

```
##            [,1]      [,2]
## [1,]  0.6778734 0.7351787
## [2,] -0.7351787 0.6778734
```

```
tM2
```

```
##    [,1]  [,2] [,3] [,4] [,5] [,6]  [,7]  [,8]  [,9] [,10]
## x1 0.69 -1.31 0.39 0.09 1.29 0.49  0.19 -0.81 -0.31 -0.71
## x2 0.49 -1.21 0.99 0.29 1.09 0.79 -0.31 -0.81 -0.31 -1.01
```

- Multiplique la matriz transpuesta de los vectores propios con la transpuesta de la matriz de datos centrados.

```
CP = tVec%*%tM2
rownames(CP)= c("CP1", "CP2")
CP = t(CP)
CP
```

```
##               CP1         CP2
##  [1,]  0.82797019 -0.17511531
##  [2,] -1.77758033  0.14285723
##  [3,]  0.99219749  0.38437499
##  [4,]  0.27421042  0.13041721
##  [5,]  1.67580142 -0.20949846
##  [6,]  0.91294910  0.17528244
##  [7,] -0.09910944 -0.34982470
##  [8,] -1.14457216  0.04641726
##  [9,] -0.43804614  0.01776463
## [10,] -1.22382056 -0.16267529
```

- Comprobando el primer resultado:

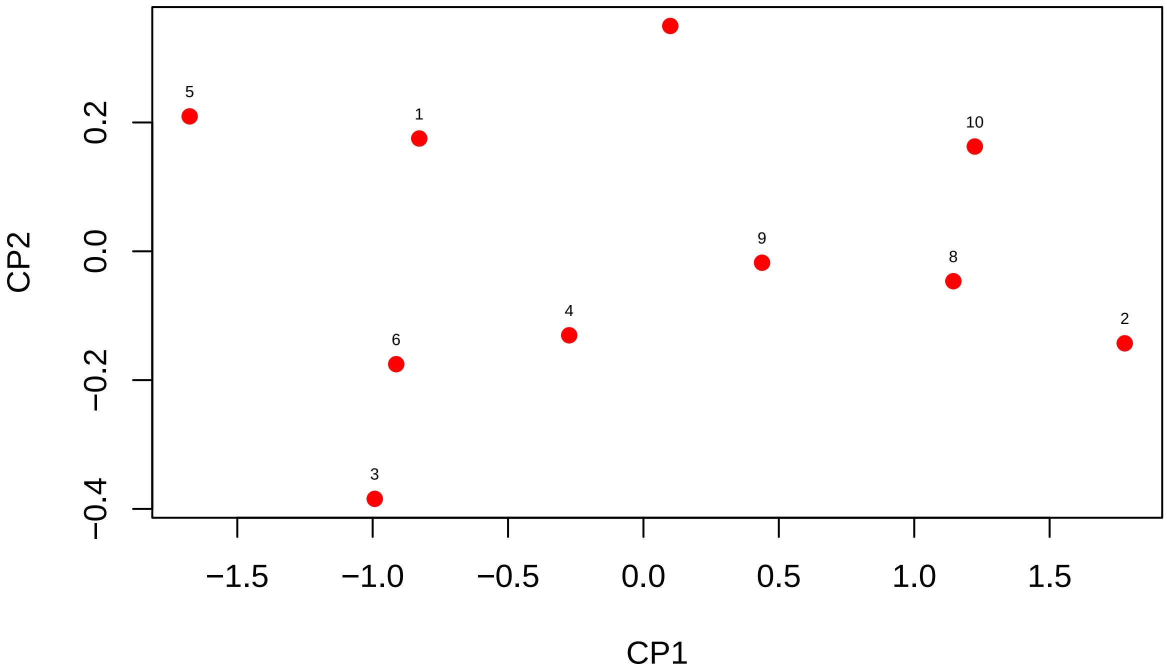
$Y_1 = 0.68\ X_1 + 0.74\ X_2$

```
Y1 = (Mvectors[1,1]*M2[1,1]) + (Mvectors[2,1]*M2[1,2]); Y1
```

```
## [1] 0.8279702
```

- Visualización

```
CP2 = -CP
plot(CP2,pch = 19,col = "red")
text(CP2[,1],CP2[,2],1:nrow(CP2),cex = .5, pos = 3)
```



- Interprete los resultados.

PARTE B

Aplique a los mismos datos las fórmulas de R para Componentes principales e interpreta resultados.

- Use el comando:

```
cpa <- prcomp(M, scale=FALSE)
names(cpa)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

- Explora las opciones del comando:

```
print("desviaciones estándar: ")
```

```
## [1] "desviaciones estándar: "
```

```
cpa$sdev
```

```
## [1] 1.1331495 0.2215477
```

```
print("medias: ")
```

```
## [1] "medias: "
```

```r
print("center da el centroide (medias de cada variable) previa estandarización: ")
```

```
## [1] "center da el centroide (medias de cada variable) previa estandarización: "
```

```r
cpa$center
```

```
##   x1   x2
## 1.81 1.91
```

```r
print("scale informa si la variable se estandarizó: ")
```

```
## [1] "scale informa si la variable se estandarizó: "
```

```r
cpa$scale
```

```
## [1] FALSE
```

```r
print("Los coeficientes de la combinación lineal normalizada de componete")
```

```
## [1] "Los coeficientes de la combinación lineal normalizada de componete"
```

```r
cpa$rotation
```

```
##           PC1        PC2
## x1 -0.6778734  0.7351787
## x2 -0.7351787 -0.6778734
```

```r
print("Los datos por sustituidos en la combinación lineal de vectores propios:")
```

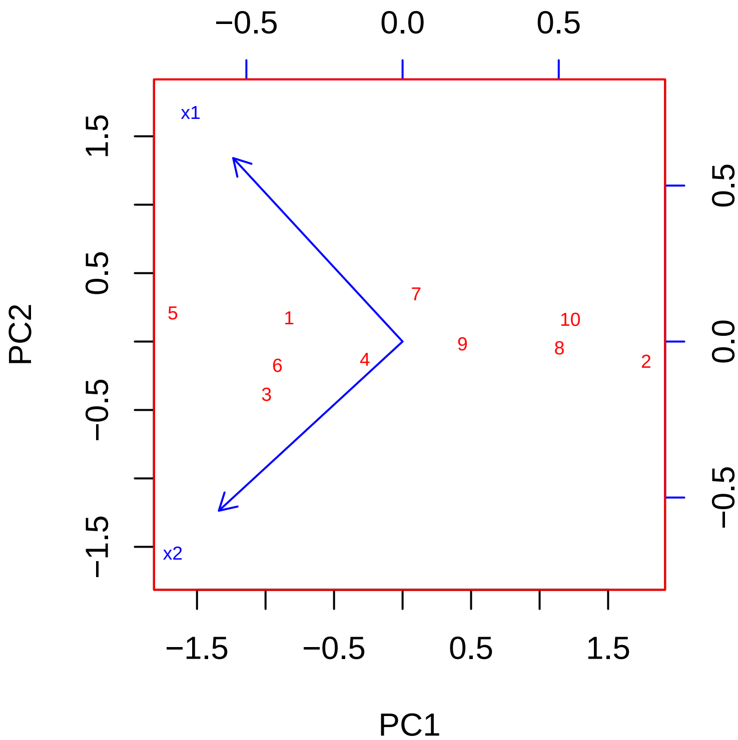
```
## [1] "Los datos por sustituidos en la combinación lineal de vectores propios:"
```

```r
cpa$x
```

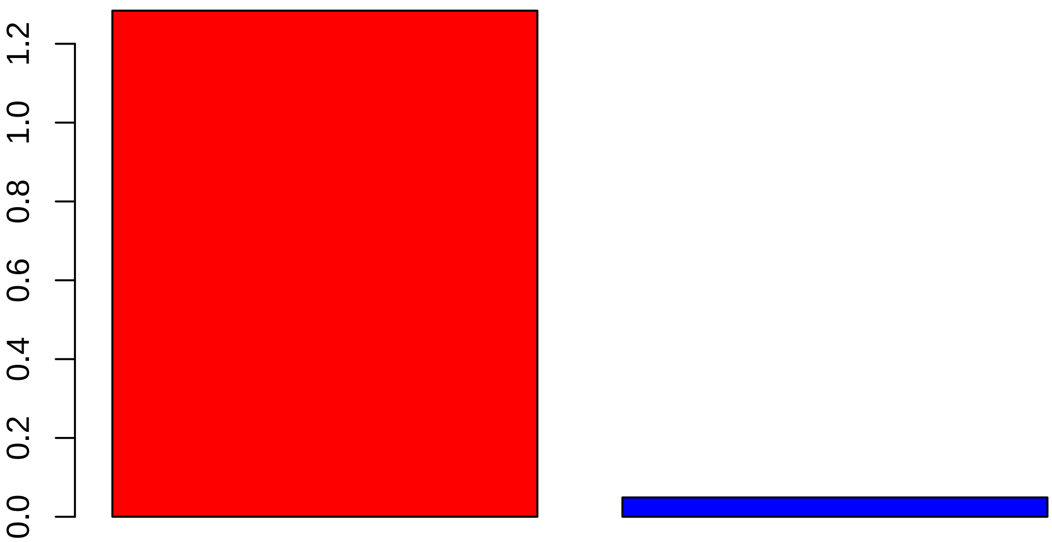
```
##               PC1         PC2
##  [1,] -0.82797019  0.17511531
##  [2,]  1.77758033 -0.14285723
##  [3,] -0.99219749 -0.38437499
##  [4,] -0.27421042 -0.13041721
##  [5,] -1.67580142  0.20949846
##  [6,] -0.91294910 -0.17528244
##  [7,]  0.09910944  0.34982470
##  [8,]  1.14457216 -0.04641726
##  [9,]  0.43804614 -0.01776463
## [10,]  1.22382056  0.16267529
```

- Grafica:

```r
biplot(x = cpa, scale = 0, cex =0.6, col = c("red", "blue"))
```




```r
barplot(cpa$sdev^2, col =c( "red", "blue"))
```



Observando la gráfica podemos ver que ambas variables son independientes ya que sus vectores ortogonales y podriamos despreciar los componentes 2.

- Importancia de los componentes:

```r
summary(cpa)
```

```
## Importance of components:
##                           PC1     PC2
## Standard deviation     1.1331 0.22155
## Proportion of Variance 0.9632 0.03682
## Cumulative Proportion  0.9632 1.00000
```