PCA-Lectura

Fernando Anorve

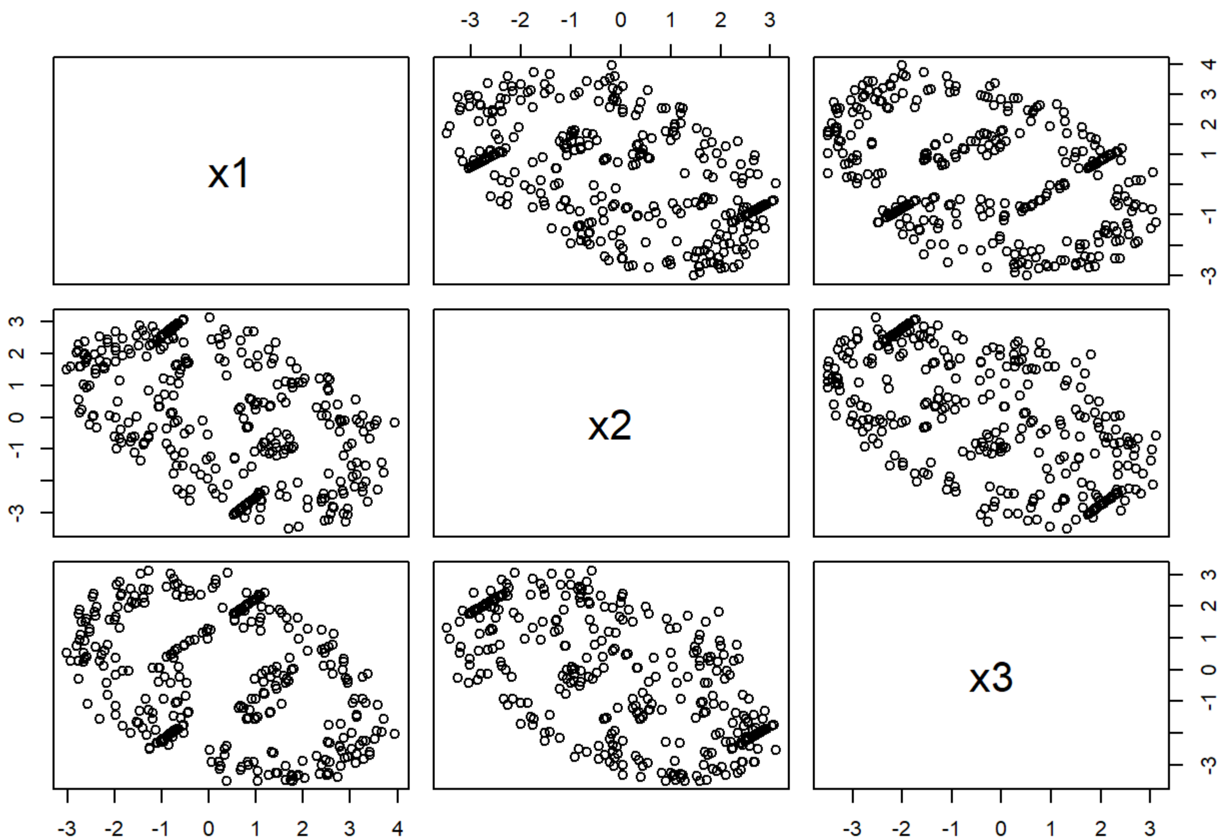
1/30/2020

Un ejemplo a mano

```
set.seed(123)

# setwd("D:/Documents/Escuela/Seminario Estadistica 2020-2/Lectures/PCA")

X.table = read.table("ejemplo3.txt")
names(X.table) <- c("x1","x2","x3")
pairs(X.table)
```



Convertimos el data.frame en una matriz con las coordenadas correctas

```
X = t(as.matrix(X.table))
```

Estandarizamos las variables

```
X[1,] = (X[1,] - mean(X[1,]))/sd(X[1,])
X[2,] = (X[2,] - mean(X[2,]))/sd(X[2,])
X[3,] = (X[3,] - mean(X[3,]))/sd(X[3,])
```

Matriz de varianza-covarianza

```
C = var(t(X))
```

Notad que si multiplicamos los vectores por -1, siguen siendo eigenvectores, siguen teniendo norma 1 y siguen siendo ortogonales

```
eig.matrix <- eigen(C)$vectors
eig.value  <- eigen(C)$values
```

Elegimos los primeros dos vectores de ponderaciones/cargas

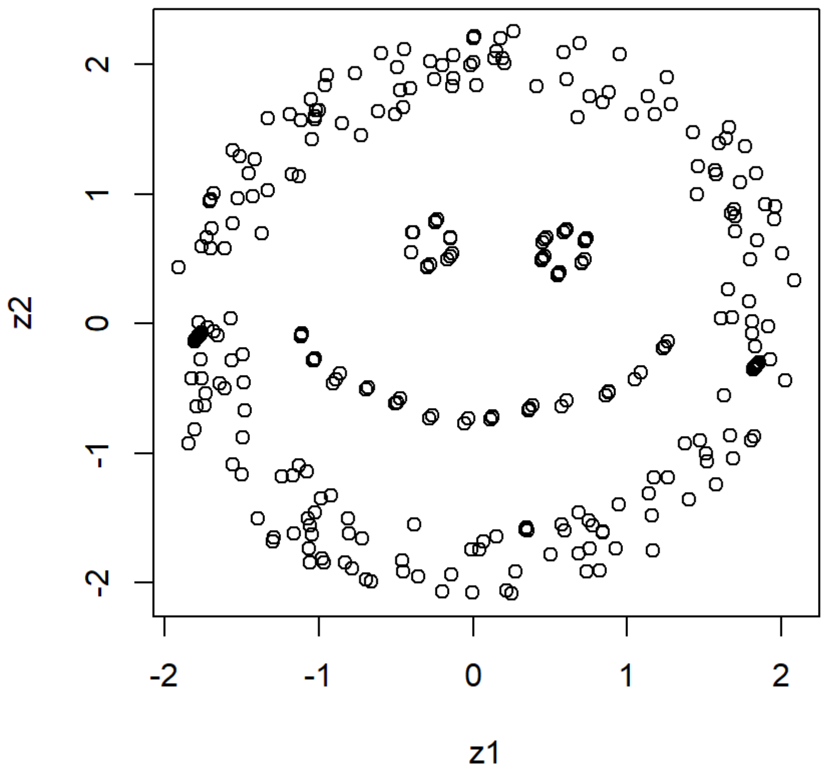
```
U = eig.matrix[,1:2]
```

Calculamos los primeros dos componentes principales

```
Z = t(U) %*% X
```

Asi lucen nuestros datos en 2d

```
plot(t(Z),xlab = "z1",ylab = "z2")
```



Y si lo queremos expresar de vuelta en 3d?

```
X.approx = U %*% Z
```

La libreria rgl tiene herramientas para visualizar en 3d

```
library(rgl)
```

```
## Warning: package 'rgl' was built under R version 3.6.2
```

```
# plot3d(t(X.approx), type='s', size=1,xlab = "x1" , ylab = "x2" , zlab = "x3")
# plot3d(t(cbind(X,X.approx)), type='s', size=0.5, xlab = "x1" , ylab = "x2" , zlab = "x3",
#       col = c(rep(1,405),rep(3,405)))
```

El mismo ejemplo, usando las librerias que vimos ayer

```
library("FactoMineR")
```

```
## Warning: package 'FactoMineR' was built under R version 3.6.2
```

```
library("factoextra")
```

```
## Warning: package 'factoextra' was built under R version 3.6.2
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
ej1.pca <- PCA(X.table, scale.unit = TRUE, ncp = 5, graph = F)
ej1.pca$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1 1.70977391              56.992464                          56.99246
## comp 2 1.25350704              41.783568                          98.77603
## comp 3 0.03671905               1.223968                         100.00000
```

Esto es lo que habiamos calculado antes

```
eig.value
```

```
## [1] 1.70977391 1.25350704 0.03671905
```

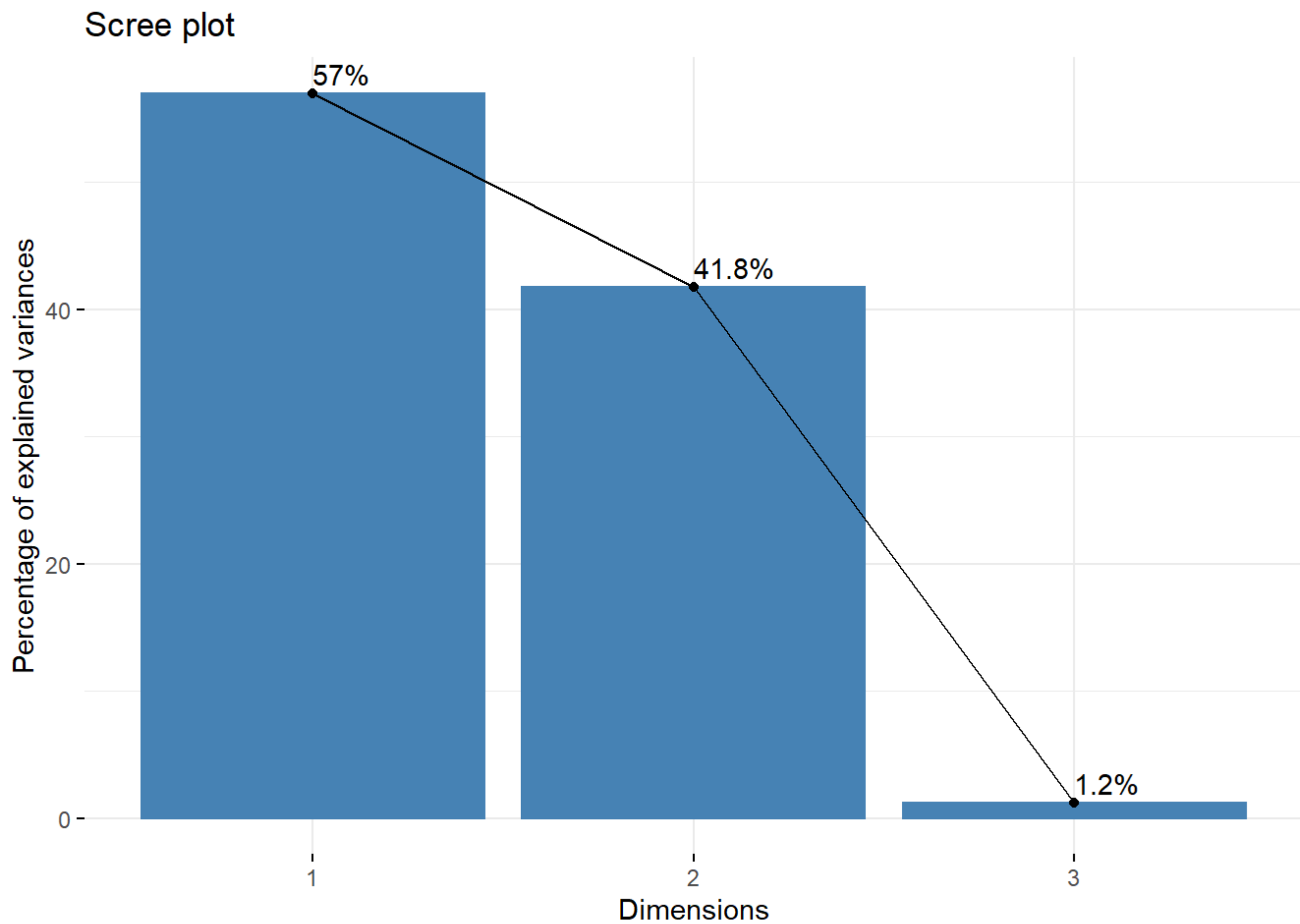
Notamos que a partir de los eigenvalores se puede calcular la proporcion de varianzas!!

```
100*eig.value/sum(eig.value)
```

```
## [1] 56.992464 41.783568  1.223968
```

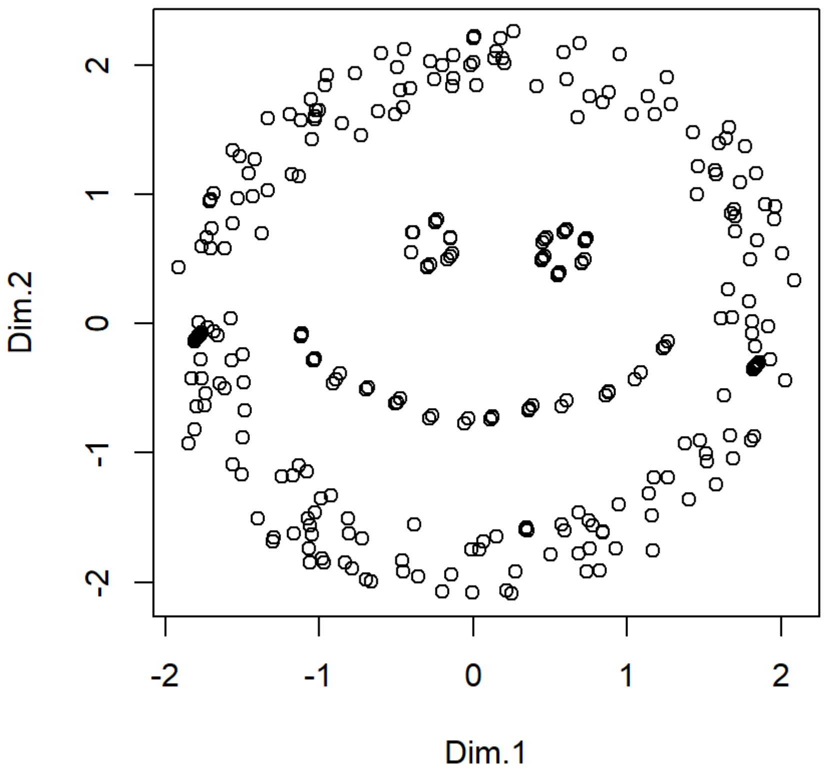
Scree plot de este ejemplo. Notamos que coinciden con la proporcion de eigenvalores y que los primeros dos componentes explican la mayor parte de la varianza

```
fviz_eig(ej1.pca, addlabels = TRUE)
```



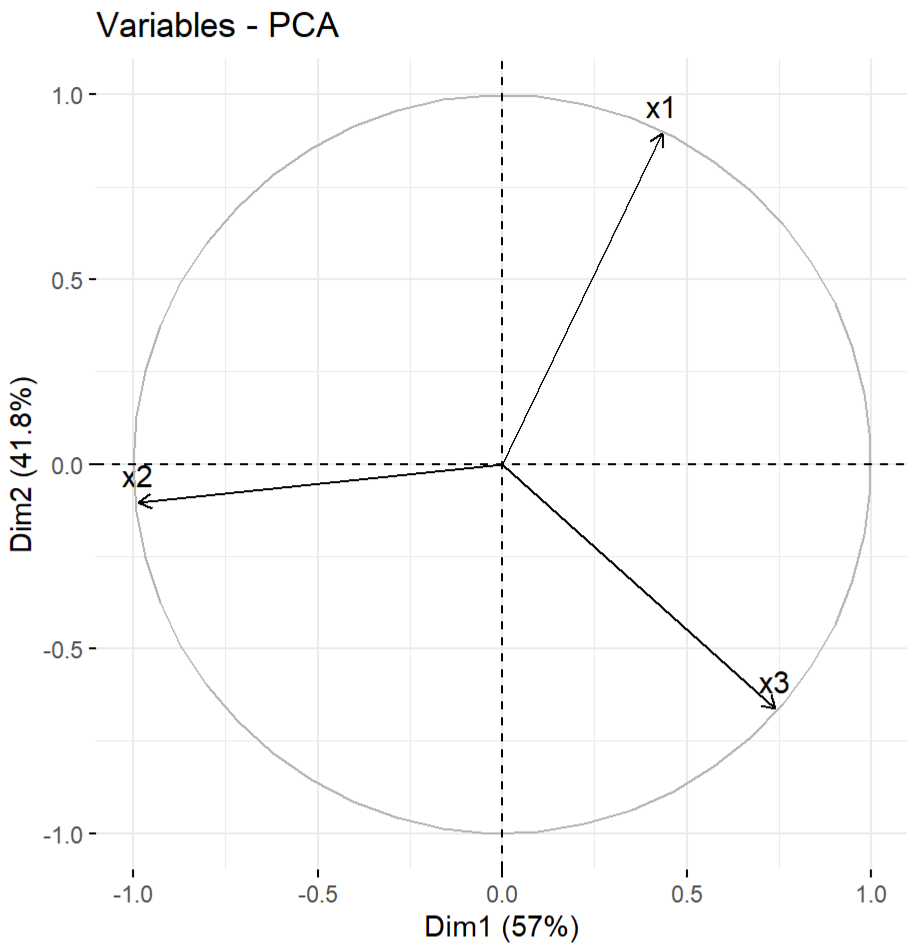
Como lucen los puntos en 2d?

```
coordenadas = ej1.pca$ind$coord[,1:2]
plot(coordenadas)
```



Por ultimo, podemos observar el peso de cada variable sobre los componentes principales

```
fviz_pca_var(ej1.pca, col.var = "black")
```



Como interpretar?

- La variable x1 tiene un gran peso en el componente 2, y relativamente poco peso en el componente 1
- La variable x2 tiene muy poco peso en el componente 2, y gran peso en el componente 1
- La variable x1 tiene relativamente buen peso en ambos