

# PCA

Horacio Alberto

2022-06-07

## Análisis de Componentes Principales

### Introducción

El Análisis de componentes principales (**ACP**) es un método de reducción de la dimensionalidad de las variables originales.

### Matriz de trabajo

1.- Se trabajó con la matriz flores, extraída del paquete **datos** que se encuentra precargado en R.

```
install.packages("datos")
```

```
library(datos)
```

2.- Se selecciona la matriz flores

```
flo<-datos::flores
```

### Exploración de la matriz

1.- Dimensión de la matriz. La matriz cuenta con 150 observaciones y 5 variables.

```
dim(flo)
```

```
## [1] 150 5
```

2.- Tipo de variables.

```
str(flo)
```

```
## 'data.frame': 150 obs. of 5 variables:
## $ Largo.Sepalo: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Ancho.Sepalo: num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Largo.Petalo: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Ancho.Petalo: num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Especie : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

3.- Nombre de las variables.

```
colnames(flo)
```

```
## [1] "Largo.Sepalo" "Ancho.Sepalo" "Largo.Petalo" "Ancho.Petalo" "Especie"
```

4.- En busca de datos perdidos.

```
anyNA(flo)
```

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

## Tratamiento de la matriz

Se genera una nueva matriz **flor** filtrada.

2.- Selección de las variables cuantitativas de la especie versicolor.

```
flor<-flo[51:100,1:4]
```

## ACP paso a paso

1.- Transformar la matriz en un data frame

```
flo<-as.data.frame(flor)
```

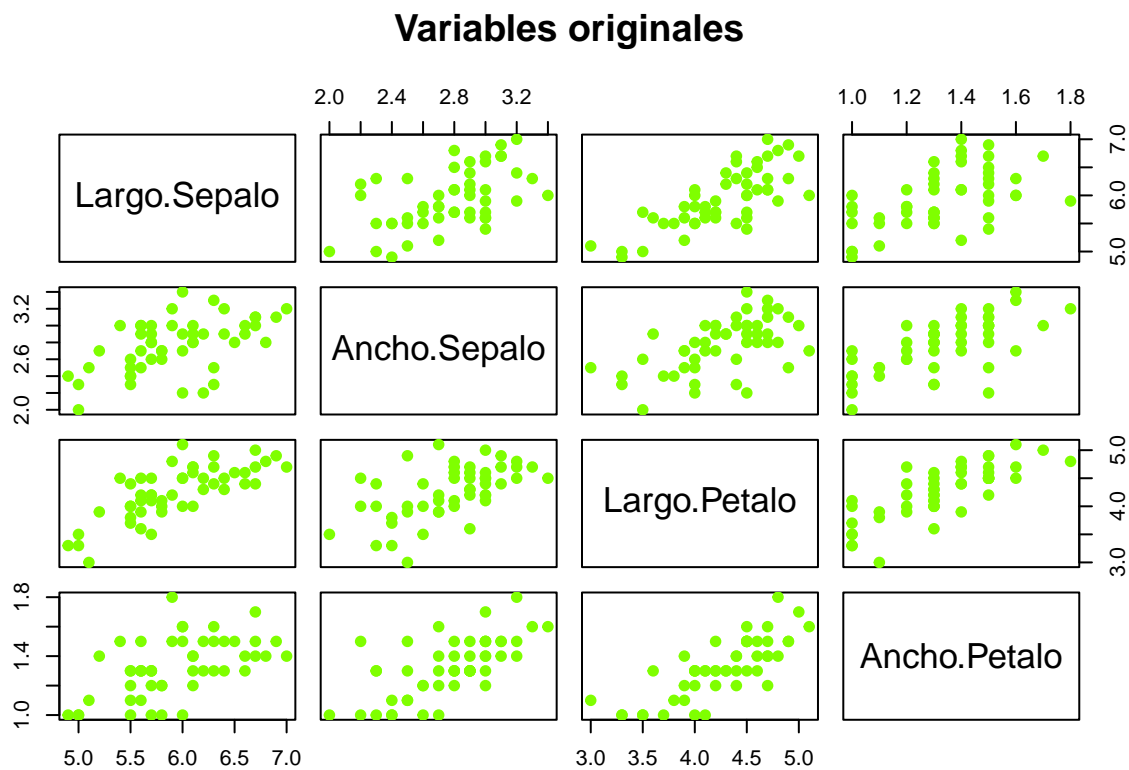
2.- Definir  $n$  (individuos) y  $p$  (variables).

```
n<-dim(flo)[1]
```

```
p<-dim(flo)[2]
```

3.- Generación del gráfico **scatterplot**.

```
pairs(flor,col="chartreuse", pch=19,  
      main="Variables originales")
```



4.- Obtención de la media por columna y la matriz de covarianza muestral.

```
mu<-colMeans(flor)
```

```
mu
```

```
## Largo.Sepalo Ancho.Sepalo Largo.Petalo Ancho.Petalo  
##      5.936      2.770      4.260      1.326
```

```
s<-cov(flor)
```

```
s
```

```
##           Largo.Sepalo Ancho.Sepalo Largo.Petalo Ancho.Petalo
## Largo.Sepalo  0.26643265  0.08518367  0.18289796  0.05577959
## Ancho.Sepalo  0.08518367  0.09846939  0.08265306  0.04120408
## Largo.Petalo  0.18289796  0.08265306  0.22081633  0.07310204
## Ancho.Petalo  0.05577959  0.04120408  0.07310204  0.03910612
```

5.- Obtención de los valores y vectores propios desde la matriz de covarianza muestral.

```
es<-eigen(s)
es

## eigen() decomposition
## $values
## [1] 0.487873944 0.072384096 0.054776085 0.009790365
##
## $vectors
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.6867238  0.6690891 -0.26508336  0.1022796
## [2,] 0.3053470 -0.5674653 -0.72961786 -0.2289194
## [3,] 0.6236631 -0.3433270  0.62716496 -0.3159668
## [4,] 0.2149837 -0.3353051  0.06366081  0.9150409
```

5.1.- Separación de la matriz de valores propios.

```
eigen.val<-es$values
eigen.val

## [1] 0.487873944 0.072384096 0.054776085 0.009790365
```

5.2.- Separación de la matriz de vectores propios.

```
eigen.vec<-es$vectors
eigen.vec

##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.6867238  0.6690891 -0.26508336  0.1022796
## [2,] 0.3053470 -0.5674653 -0.72961786 -0.2289194
## [3,] 0.6236631 -0.3433270  0.62716496 -0.3159668
## [4,] 0.2149837 -0.3353051  0.06366081  0.9150409
```

6.- Calcular la proporción de variabilidad

6.1.- Para la matriz de valores propios.

```
pro.var<-eigen.val/sum(eigen.val)
pro.var

## [1] 0.78081758 0.11584709 0.08766635 0.01566898
```

6.2.- Acumulada.

```
pro.var.acum<-cumsum(eigen.val)/sum(eigen.val)
pro.var.acum

## [1] 0.7808176 0.8966647 0.9843310 1.0000000
```

7.- Obtención de la matriz de correlaciones

```
R<-cor(flor)
R

##           Largo.Sepalo Ancho.Sepalo Largo.Petalo Ancho.Petalo
```

```
## Largo.Sepalo    1.0000000    0.5259107    0.7540490    0.5464611
## Ancho.Sepalo    0.5259107    1.0000000    0.5605221    0.6639987
## Largo.Petalo    0.7540490    0.5605221    1.0000000    0.7866681
## Ancho.Petalo    0.5464611    0.6639987    0.7866681    1.0000000
```

8.- Obtención de los valores y vectores propios a partir de la **matriz de correlaciones**.

```
eR<-eigen(R)
eR

## eigen() decomposition
## $values
## [1] 2.9263407 0.5462747 0.3949976 0.1323871
##
## $vectors
##          [,1]      [,2]      [,3]      [,4]
## [1,] -0.4823284  0.6107980 -0.4906296  0.3918772
## [2,] -0.4648460 -0.6727830 -0.5399025 -0.1994658
## [3,] -0.5345136  0.3068495  0.3402185 -0.7102042
## [4,] -0.5153375 -0.2830765  0.5933290  0.5497778
```

9.- Separación de la matriz de valores propios a partir de la matriz de correlaciones.

9.1.- Separación de la matriz de vectores propios.

```
eigen.val.R<-eR$values
eigen.val.R

## [1] 2.9263407 0.5462747 0.3949976 0.1323871
```

9.2.- Separación de la matriz de vectores propios.

```
eigen.vec.R<-eR$vectors
eigen.vec.R

##          [,1]      [,2]      [,3]      [,4]
## [1,] -0.4823284  0.6107980 -0.4906296  0.3918772
## [2,] -0.4648460 -0.6727830 -0.5399025 -0.1994658
## [3,] -0.5345136  0.3068495  0.3402185 -0.7102042
## [4,] -0.5153375 -0.2830765  0.5933290  0.5497778
```

10.- Cálculo de la proporción de variabilidad.

10.1.- Para la matriz de valores propios.

```
pro.var.R<-eigen.val.R/sum(eigen.val.R)
pro.var.R

## [1] 0.73158517 0.13656866 0.09874939 0.03309677
```

10.2.- Acumulada. En este punto se seleccionan en número de componentes, siguiendo el criterio del 80% de la varianza explicada. Para este ejemplo se van a seleccionar 2 factores (0.868% de varianza explicada).

```
pro.var.acum.R<-cumsum(eigen.val.R)/sum(eigen.val.R)
pro.var.acum.R

## [1] 0.7315852 0.8681538 0.9669032 1.0000000
```

11.- Calcular la media de los valores propios.

```
mean(eigen.val.R)
```

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

## Obtención de coeficientes

12.- Centrar los datos con respecto a la media 12.1.- Construcción de matriz de 1

```
ones<-matrix(rep(1,n),nrow=n, ncol=1)
```

12.2.- Construcción de la matriz centrada

```
X.cen<-as.matrix(flor)-ones%*%mu
```

13.- Construcción de la matriz diagonal de las covarianzas

```
Dx<-diag(diag(s))
```

Dx

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.2664327 0.0000000 0.0000000 0.0000000
## [2,] 0.0000000 0.09846939 0.0000000 0.0000000
## [3,] 0.0000000 0.0000000 0.2208163 0.0000000
## [4,] 0.0000000 0.0000000 0.0000000 0.03910612
```

14.- Construcción de la matriz centrada multiplicada por  $Dx^{1/2}$

```
Y<-X.cen%*%solve(Dx)^(1/2)
```

15.- Construcción de los coeficientes o scores eigen.vec.R matriz de autovectores Se muestran las primeras 10 observaciones.

```
scores<-Y%*%eigen.vec.R
scores[1:10,]
```

```
##           [,1]      [,2]      [,3]      [,4]
## 51 -2.32455278  0.5185273 -1.21059316  0.075191200
## 52 -1.79699308 -0.4652131 -0.48504815  0.199955742
## 53 -2.57106666  0.6020469 -0.49865033  0.038577169
## 54  1.46714905  0.3591890  0.95682822  0.288414020
## 55 -1.41164332  0.5760181  0.18051660  0.378999671
## 56 -0.02915352 -0.1496476  0.26845808 -0.633250224
## 57 -2.33977751 -0.8104931 -0.11721324  0.036211804
## 58  3.45770058 -0.5928617 -0.05182738 -0.006758222
## 59 -1.13202813  0.7662442 -0.68666085 -0.164670936
## 60  1.00808930 -1.0618537  0.78140281  0.235542894
```

16.- Nombramos las columnas PC1...PC8

```
colnames(scores)<-c("PC1", "PC2", "PC3", "PC4")
```

17.- Visualización de los scores

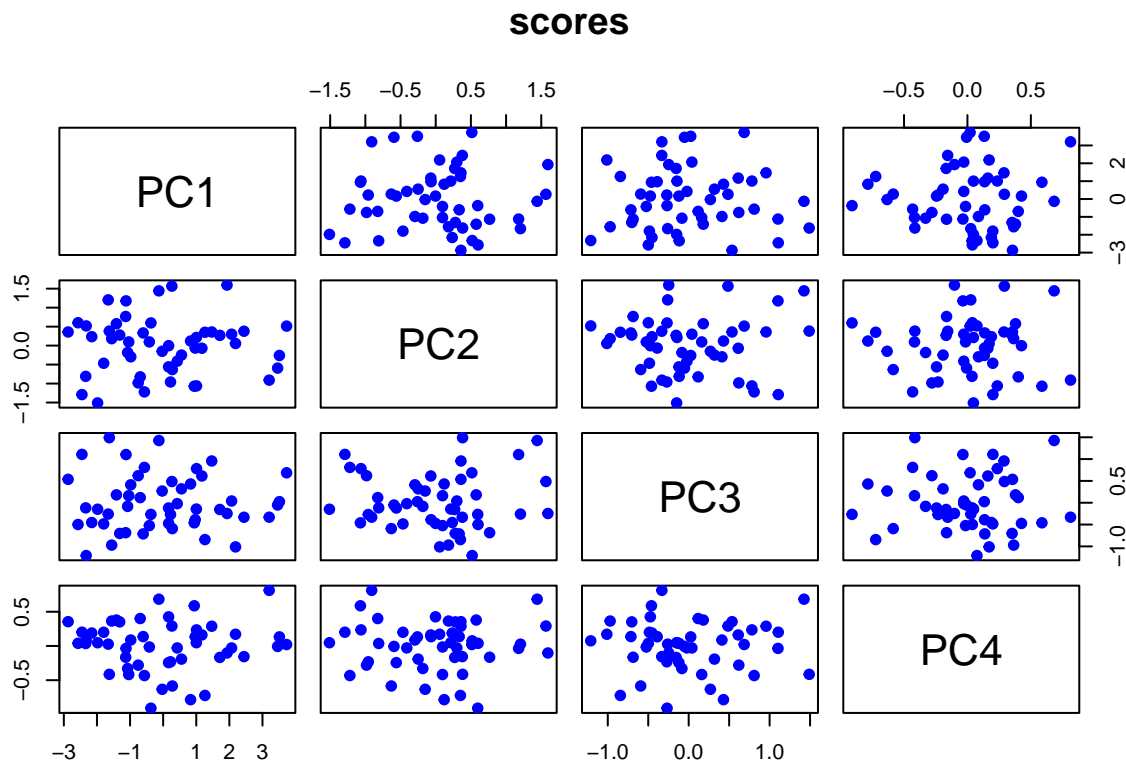
```
scores[1:10,]
```

```
##           PC1      PC2      PC3      PC4
## 51 -2.32455278  0.5185273 -1.21059316  0.075191200
## 52 -1.79699308 -0.4652131 -0.48504815  0.199955742
## 53 -2.57106666  0.6020469 -0.49865033  0.038577169
## 54  1.46714905  0.3591890  0.95682822  0.288414020
## 55 -1.41164332  0.5760181  0.18051660  0.378999671
## 56 -0.02915352 -0.1496476  0.26845808 -0.633250224
## 57 -2.33977751 -0.8104931 -0.11721324  0.036211804
```

```
## 58  3.45770058 -0.5928617 -0.05182738 -0.006758222
## 59 -1.13202813  0.7662442 -0.68666085 -0.164670936
## 60  1.00808930 -1.0618537  0.78140281  0.235542894
```

18.-Generacion del grafico de los scores

```
pairs(scores, main="scores", col="blue", pch=19)
```



## ACP VIA SINTETIZADA

1.- Cálculo de la varianza a las columnas (1=filas, 2=columnas)

```
apply(flor, 2, var)
```

```
## Largo.Sepalo Ancho.Sepalo Largo.Petalo Ancho.Petalo
## 0.26643265 0.09846939 0.22081633 0.03910612
```

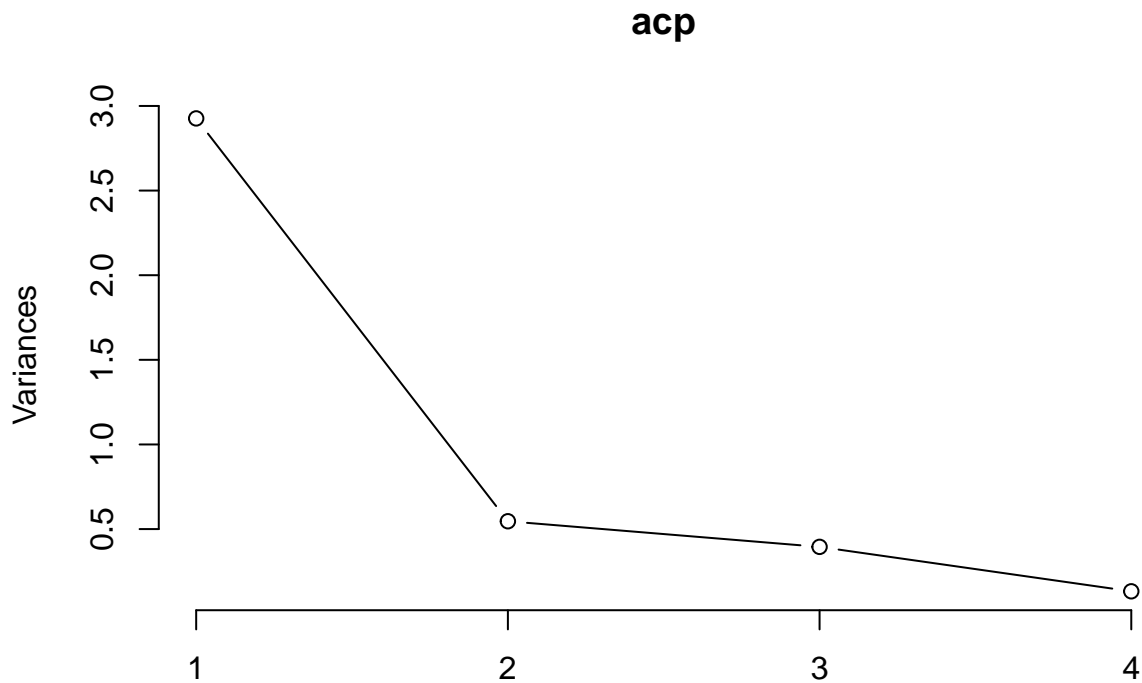
2.- Aplicar la función **prcomp** para reducir la dimensionalidad y centrado por la media y escalada por la desviación estandar (dividir entre sd).

```
acp<-prcomp(flor, center = TRUE, scale=TRUE)
acp
```

```
## Standard deviations (1, ..., p=4):
## [1] 1.7106550 0.7391040 0.6284883 0.3638504
##
## Rotation (n x k) = (4 x 4):
##           PC1      PC2      PC3      PC4
## Largo.Sepalo -0.4823284 -0.6107980  0.4906296  0.3918772
## Ancho.Sepalo -0.4648460  0.6727830  0.5399025 -0.1994658
## Largo.Petalo -0.5345136 -0.3068495 -0.3402185 -0.7102042
## Ancho.Petalo -0.5153375  0.2830765 -0.5933290  0.5497778
```

3.- Generación del gráfico **screeploy**.

```
plot(acp, type="l")
```



4.- Resumen de la matriz **acp**

```
summary(acp)
```

```
## Importance of components:
##               PC1    PC2    PC3    PC4
## Standard deviation  1.7107 0.7391 0.62849 0.3639
## Proportion of Variance 0.7316 0.1366 0.09875 0.0331
## Cumulative Proportion 0.7316 0.8681 0.96690 1.0000
```

## Construcción de los CP con las variables originales

Combinación lineal de las variables originales

$$z1 = -0.0482(\text{var1}) - 0.464(\text{var2}) - 0.534(\text{var3}) - 0.515(\text{var4})$$

El primer componente distingue entre flores grandes y pequeñas

- Sépalo corto
- Sépalo angosto
- Sépalo corto
- Sépalo angosto

$$z2 = -0.610(\text{var1}) + 0.672(\text{var2}) - 0.306(\text{var3}) - 0.515(\text{var4})$$

El segundo componente distingue flores por especie

- Sépalo corto
- Sépalo ancho
- Pétalo corto
- Pétalo ancho