

PCA

Omar Sanchez Hernandez

2022-03-28

Análisis de componentes principales

Introducción

El análisis de componentes o (PCA) es una técnica estadística que se encarga de describir un conjunto de datos con nuevas variables, por lo general los componentes se ordenan por sus respectivas varianzas, por lo que esta técnica es útil cuando es necesario reducir la dimensionalidad de nuestros datos.

Selección de los datos para la realización del ejemplo

1-. Se seleccionó la base llamada “*flores*”, datos extraídos de la paquetería “*datos*”

```
#install.packages("datos",dependencies=T)
library(datos)
x <- flores
```

Exploracion de la matriz

1-. Dimensión de la matriz

```
dim(x)
```

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

2-. Exploracion de las variables

```
str(x)
```

```
## '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(x)
```

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

4-.Se verifica que no existan datos perdidos

```
anyNA(x)
```

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

Creación de un nuevo data frame solo con las variables cuantitativas

Instalación de un paquete para la manipulación de dataframes

```
#install.packages("dplyr")  
library(dplyr)
```

1-. Creacion de un nuevo data frame

```
datos2 <- select(flores, Largo.Sepalo, Ancho.Sepalo, Largo.Petalo, Ancho.Petalo)  
x<-as.data.frame(datos2)
```

Tratamiento de matriz

Se genera una nueva matriz **x1** con los datos de solo una especie, en este caso la especie “Versicolor”..

1 Seleccin de las nuevas variables.

```
x1 <- x[51:100,1:4]
```

ACP paso a paso

1 se transforma la nueva matriz a un data frame

```
x1 <- as.data.frame(x1)
```

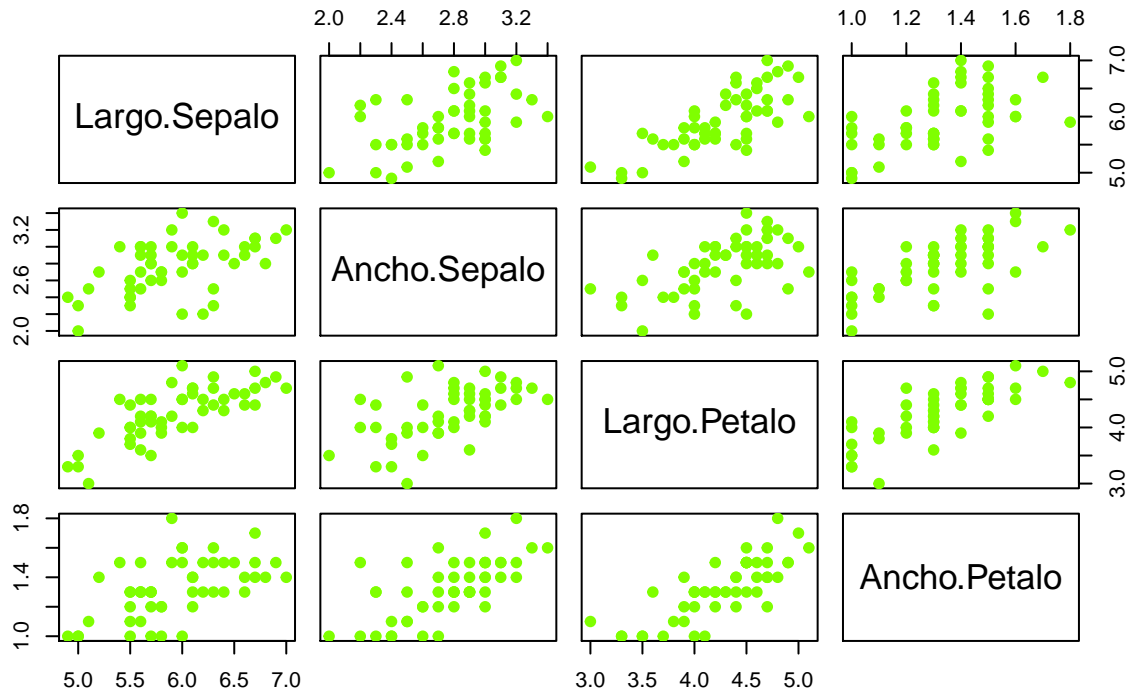
2- Definir n (individuos) y p (variables)

```
n<-dim(x)[1]  
p<-dim(x)[2]
```

3- Generar grafico

```
pairs(x1,col="chartreuse", pch=19,  
      main="Variables originales")
```

Variables originales



4.- Obtención de la media por columna

```
mu<-colMeans(x1)
mu
```

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

5.- la matriz de covarianza muestral

```
s<-cov(x1)
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
```

6.- 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
```

6.1-. Separación de la matriz de valores propios

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

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

6.2-. Separacion de matrices 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
```

7- Calcular la proporcion de la variabilidad

7.1- Para la matriz de valores propios

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

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

7.2- variabilidad acumulada

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

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

8-. obtencion de la matriz de correlaciones

```
R<-cor(x1)
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
```

9-. Obtencion 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
```

10-. Separacion de la matriz de valores propios

10.1- Separación de la matriz de valores propios

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

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

10.2- Separacion de matrices 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
```

11- Cálculo de la proporcion de variabilidad

11.1- Para la matriz de valores propios

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

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

11.2- Acumulada

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

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

Una vez observados los valores anteriores, podemos seleccionar los primeros 2 valores, ya que cumplen con el criterio del 80%

12-. Calcular la media de los valores propios

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

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

Obtencion de coeficientes

13-. Centrar los datos con respecto a la media

13.1 Construcción de la matriz centrada

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

13.2 Construcción de la matriz centrada

```
X.cen<-as.matrix(x1-ones%*%mu)
```

14-. Construcción de la matriz diagonal de las covarianzas.

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

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.2664327 0.00000000 0.00000000 0.00000000
## [2,] 0.0000000 0.09846939 0.00000000 0.00000000
## [3,] 0.0000000 0.00000000 0.2208163 0.00000000
## [4,] 0.0000000 0.00000000 0.00000000 0.03910612
```

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

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

16-. Construcción de los coeficientes o scores eigen.vec.R

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

```
## [1] 8.034844 8.562404 7.788330 11.826546 8.947754 10.330243 8.019619
## [8] 13.817098 9.227369 11.367486
```

17-. Se nombran las columnas

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

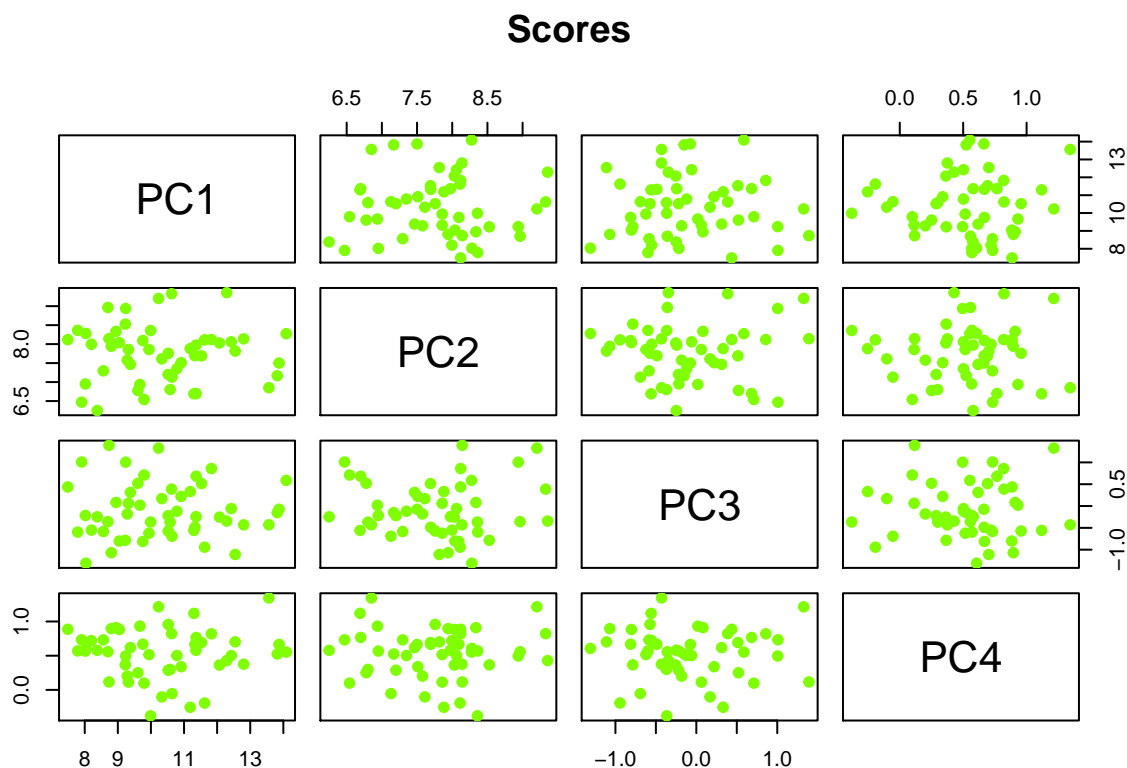
18-. Visualización de los scores

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

```
## [1] 8.034844 8.562404 7.788330 11.826546 8.947754 10.330243 8.019619
## [8] 13.817098 9.227369 11.367486
```

19-. Gráfico de los scores

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



Via sintetizada

A continuación se presenta la vía rápida para la visualización de los componentes principales y el screeplot.

1-. Aplicar el cálculo de la varianza a las columnas *1=filas*, *2=columnas*

```
apply(x, 2, var)
```

```
## Largo.Sepalo Ancho.Sepalo Largo.Petalo Ancho.Petalo
## 0.6856935 0.1899794 3.1162779 0.5810063
```

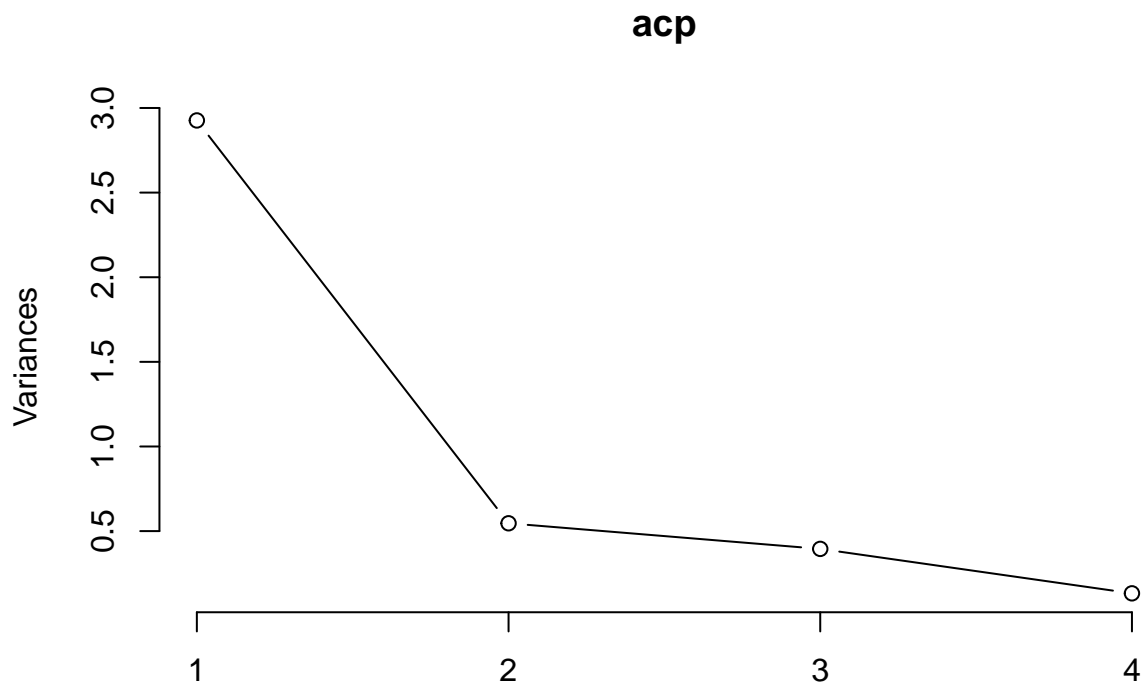
2-. Centrado por la media y escalada por la desviación estándar (dividir entre sd).

```
acp<-prcomp(x1, 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 screeplot

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



En el gráfico se muestra como solo toma un componente principal.