

PCA

Yoselin Merari Medel Colorado

2022-03-24

Análisis de Componentes Principales

Introducción

El análisis de Componentes Principales (**ACP**) es un método de reducción de la dimensión de las variables originales. Con el objetivo que estén relacionadas o que midan lo mismo bajo distintos puntos de vista. Se puede transformar el conjunto original de variables en otro conjunto de nuevas variables incorreladas entre sí. Las nuevas variables son combinaciones lineales de las anteriores y se van construyendo según el orden de importancia en cuanto a la variabilidad total que recogen de la muestra. Si las variables originales están incorreladas de partida, entonces no tiene sentido realizar un análisis de componentes principales.

Matriz de trabajo

1.- Setrabajo con la matriz (nombre la matriz), extraída del paquete **datos** que se encuentra recargado en R.

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

```
library(datos)
```

2.- Se selecciona la matriz (nombre la matriz)

```
x<-datos::flores
```

Exploración de la matriz

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

```
dim(x)
```

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

2.- Tipo de 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.- En busca de datos perdidos

```
anyNA(x)
```

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

Tratamiento de la matriz

Se genera una nueva matriz **x1** filtrada

1.- Selección del bloque por especie

```
x$Especie
```

```
## [1] setosa      setosa      setosa      setosa      setosa      setosa
## [7] setosa      setosa      setosa      setosa      setosa      setosa
## [13] setosa      setosa      setosa      setosa      setosa      setosa
## [19] setosa      setosa      setosa      setosa      setosa      setosa
## [25] setosa      setosa      setosa      setosa      setosa      setosa
## [31] setosa      setosa      setosa      setosa      setosa      setosa
## [37] setosa      setosa      setosa      setosa      setosa      setosa
## [43] setosa      setosa      setosa      setosa      setosa      setosa
## [49] setosa      setosa      versicolor  versicolor  versicolor  versicolor
## [55] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [61] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [67] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [73] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [79] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [85] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [91] versicolor  versicolor  versicolor  versicolor  versicolor  versicolor
## [97] versicolor  versicolor  versicolor  versicolor  virginica   virginica
## [103] virginica   virginica   virginica   virginica   virginica   virginica
## [109] virginica   virginica   virginica   virginica   virginica   virginica
## [115] virginica   virginica   virginica   virginica   virginica   virginica
## [121] virginica   virginica   virginica   virginica   virginica   virginica
## [127] virginica   virginica   virginica   virginica   virginica   virginica
## [133] virginica   virginica   virginica   virginica   virginica   virginica
## [139] virginica   virginica   virginica   virginica   virginica   virginica
## [145] virginica   virginica   virginica   virginica   virginica   virginica
## Levels: setosa versicolor virginica
```

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

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

PCA paso a paso

1.- Convertir la matriz en data frame

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

2.- Se define n (individuos) y p (variables)

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

3.-generación del gráfico **scatterplot**

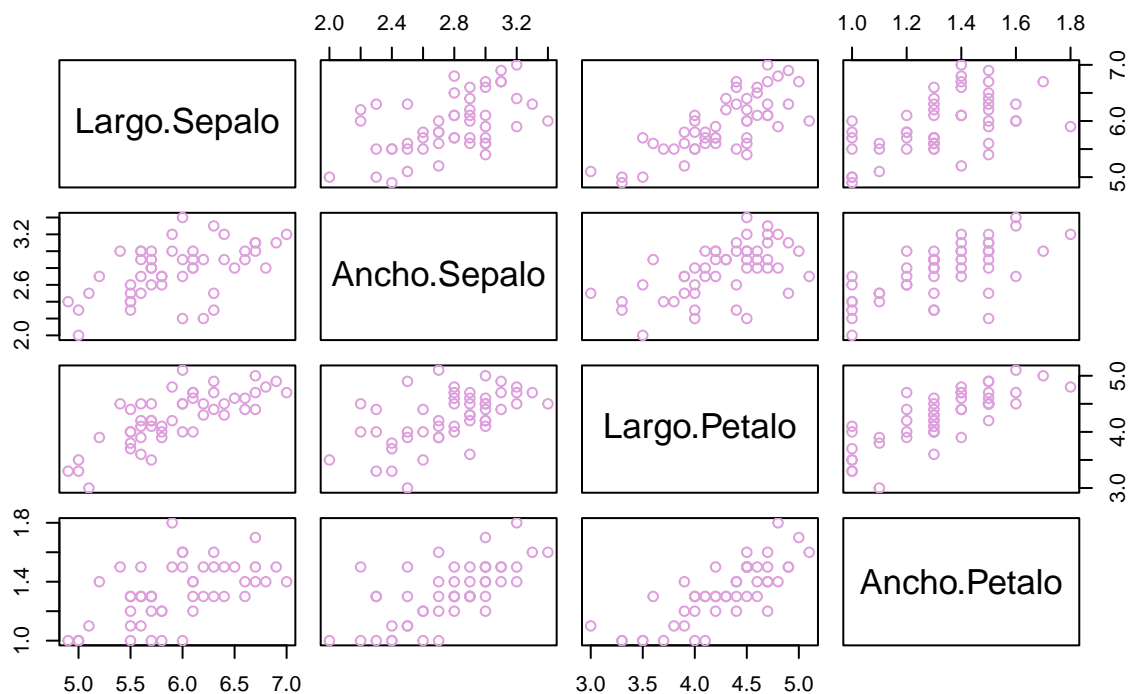
```
pairs(x1,col="plum", ph=19,  
      main="Variables originales")
```

```
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter  
## Warning in title(...): "ph" is not a graphical parameter  
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter  
## Warning in title(...): "ph" is not a graphical parameter  
## Warning in axis(side = side, at = at, labels = labels, ...): "ph" is not a  
## graphical parameter  
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "ph" is not a graphical  
## parameter  
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter  
## Warning in title(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "ph" is not a graphical  
## parameter  
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter  
## Warning in title(...): "ph" is not a graphical parameter  
## Warning in axis(side = side, at = at, labels = labels, ...): "ph" is not a  
## graphical parameter  
## Warning in axis(side = side, at = at, labels = labels, ...): "ph" is not a  
## graphical parameter  
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "ph" is not a graphical  
## parameter  
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter  
## Warning in title(...): "ph" is not a graphical parameter  
## Warning in axis(side = side, at = at, labels = labels, ...): "ph" is not a  
## graphical parameter  
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "ph" is not a graphical  
## parameter  
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter  
## Warning in title(...): "ph" is not a graphical parameter  
## Warning in plot.window(...): "ph" is not a graphical parameter  
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter
```



```
## Warning in title(...): "ph" is not a graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "ph" is not a graphical
## parameter
## Warning in plot.window(...): "ph" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter
## Warning in title(...): "ph" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "ph" is not a
## graphical parameter
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "ph" is not a graphical
## parameter
## Warning in plot.window(...): "ph" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "ph" is not a graphical parameter
## Warning in title(...): "ph" is not a graphical parameter
```

Variables originales



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

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

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

```
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
```

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(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
```

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 a partir de la matriz de correlaciones.

9.1.- Separación de la matriz de valores propios

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

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

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 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] 0.78081758 0.11584709 0.08766635 0.01566898
```

10.2.- Acumulada En este punto se selecciona el 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)/sum(eigen.val.R)
pro.var.acum.R

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

11.- Calcular la media de los valores propios

```
mean(eigen.val.R)

## [1] 0.1562061
```

Obtención de coeficientes

12.- Centrar los datos con respecto a la media

12.1.- Construcción de matriz 1

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

12.2.- Construcción de la matriz centrada

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

13.- 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
```

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 10 primeras observaciones

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

```
##           [,1]      [,2]      [,3]      [,4]  
## 51  8.034844  8.279044 -1.30931730  0.6061900  
## 52  8.562404  7.295304 -0.58377228  0.7309545  
## 53  7.788330  8.362564 -0.59737447  0.5695760  
## 54 11.826546  8.119706  0.85810409  0.8194128  
## 55  8.947754  8.336535  0.08179247  0.9099985  
## 56 10.330243  7.610869  0.16973395 -0.1022514  
## 57  8.019619  6.950024 -0.21593738  0.5672106  
## 58 13.817098  7.167655 -0.15055152  0.5242406  
## 59  9.227369  8.526761 -0.78538498  0.3663279  
## 60 11.367486  6.698663  0.68267868  0.7665417
```

16.- Nombramos las columnas de PC1 a PC8

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

17.- Visualización de los scores

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

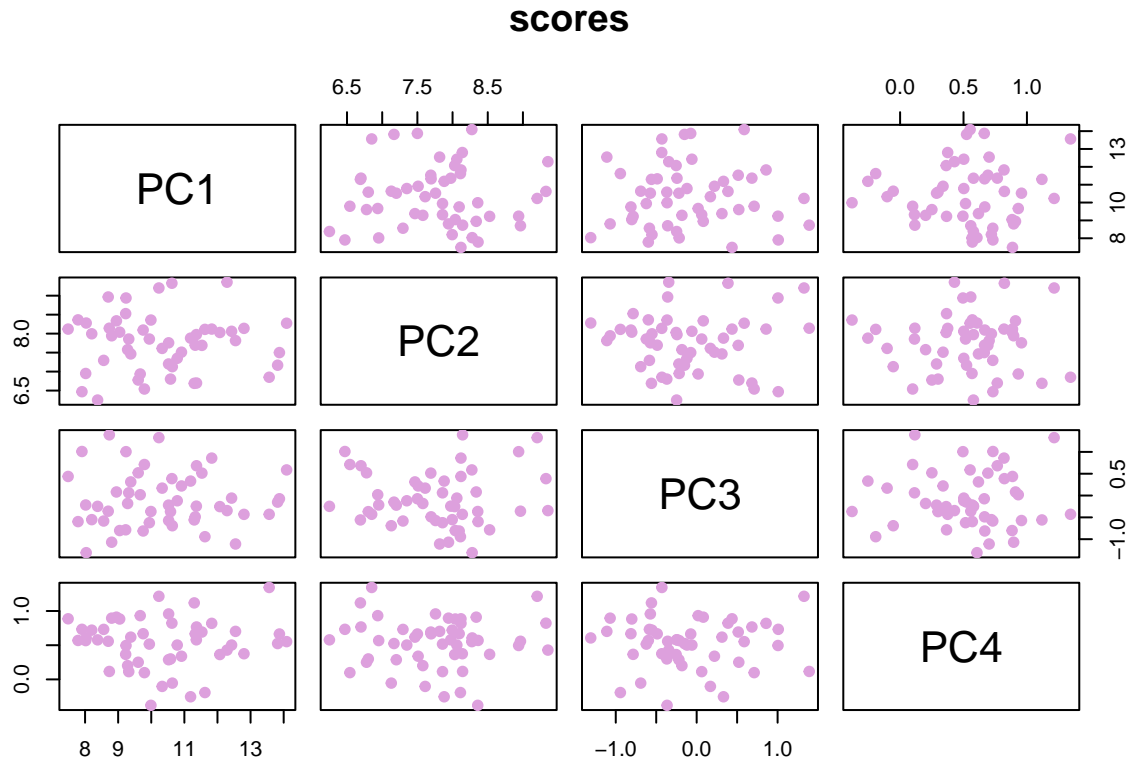
```
##           PC1      PC2      PC3      PC4  
## 51  8.034844  8.279044 -1.30931730  0.6061900  
## 52  8.562404  7.295304 -0.58377228  0.7309545  
## 53  7.788330  8.362564 -0.59737447  0.5695760  
## 54 11.826546  8.119706  0.85810409  0.8194128  
## 55  8.947754  8.336535  0.08179247  0.9099985  
## 56 10.330243  7.610869  0.16973395 -0.1022514  
## 57  8.019619  6.950024 -0.21593738  0.5672106  
## 58 13.817098  7.167655 -0.15055152  0.5242406
```



```
## 59  9.227369 8.526761 -0.78538498  0.3663279
## 60 11.367486 6.698663  0.68267868  0.7665417
```

18.- Generación del gráfico de los scores

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



ACP via sintetizada

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

```
apply(x1, 2, var)
```

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

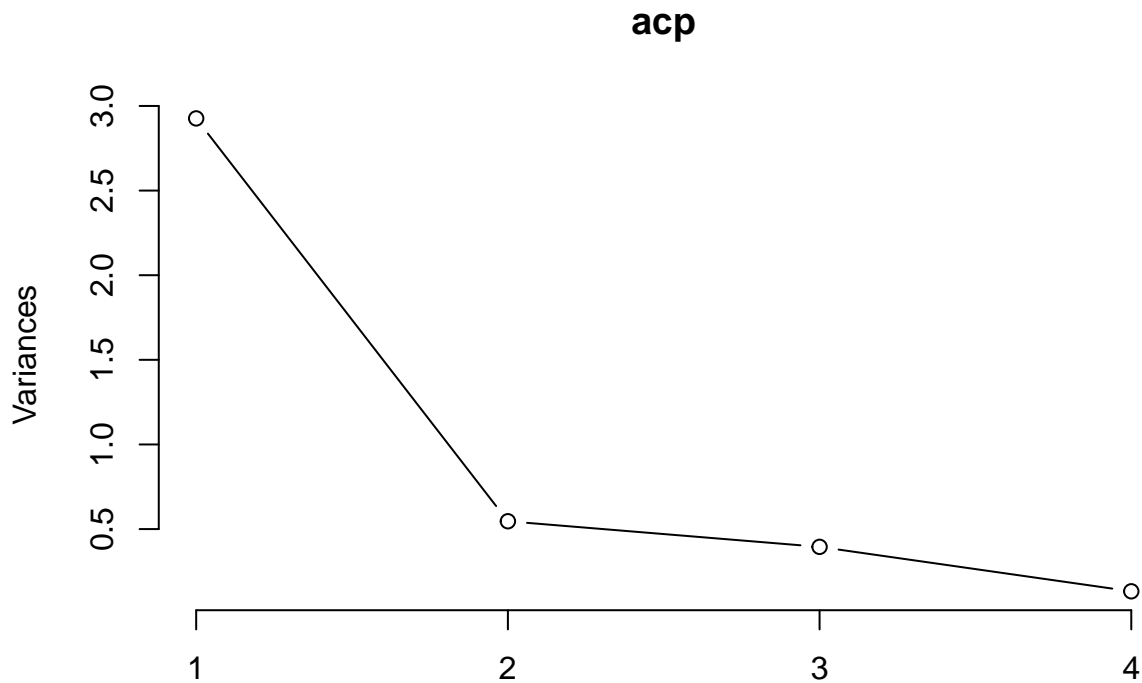
2.-Aplicar la función **prcomp** para hacer reducción de la dimensionalidad y centrado por la media y escalada por la desviación estandar

```
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.-Generar el gráfico de **screeplot**

```
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
```

Contrunscción de los CP con las varibales originales

Combinación ñineal de las variables originales

$$z1 = -0.482(\text{var1}) - 0.464(\text{var2}) - 0.534(\text{var3}) - 0.515(\text{var4})$$

El primer componente distingue entre flores grandes y flores pequeñas

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

$$z2 = -0.610(\text{var1}) + 0.672(\text{var2}) - 0.306(\text{var3}) + 0.238(\text{var4})$$

El segundo componente distingue flores por especie

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