PCA

Lino Oswaldo Sánchez Juárez

24/3/2022

Analisis de comoponentes principales

Introdución

El análisis de componentes principales (**ACP**) es un método de reducción de la dimensionalidad de las variables originales es una técnica utilizada para describir un conjunto de datos en términos de nuevas variables («componentes») no correlacionadas,en donde los componentes se ordenan por la cantidad de varianza original que tienen, así que esta técnica muy útil para reducir la dimensionalidad de un conjunto de datos.

Principalmente lo que buscamos con el **ACP** es la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados, pues esta convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación lineal llamadas componentes principales.

Las aplicaciones son en el análisis exploratorio de un conjunto de datos y con ello poder crear un modelo predictivo pues con el **ACP** conjunta el cálculo de la descomposición en auto valores de la matriz de covarianza, normalmente tras centrar los datos en la media de cada atributo.

Matriz de trabajo

1.- Se tarbajo conla matriz () extraiad del paquete **datos** que see necurntra en el paquete precargado de r

```
install.packages("datos",repos = "http://cran.us.r-project.org")
```

```
## package 'datos' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Usuario\AppData\Local\Temp\Rtmpg1ctOu\downloaded_packages
```

```
library(datos)
```

2.- Se seleciono la matriz **atmosfera**

```
BD<-(datos::atmosfera)
```

Exploracion de matriz

La matriz cuenta con 41472 observaciones y 11 variables

```
dim(BD)
```

```
## [1] 41472    11
```

2.- Tipos de variables

```
str(BD)
```

```
## tibble [41,472 x 11] (S3: tbl_df/tbl/data.frame)
##  $ latitud        : num [1:41472] 36.2 33.7 31.2 28.7 26.2 ...
##  $ longitud       : num [1:41472] -114 -114 -114 -114 -114 ...
##  $ anio           : int [1:41472] 1995 1995 1995 1995 1995 1995 1995 1995 1995 1995 ...
##  $ mes            : int [1:41472] 1 1 1 1 1 1 1 1 1 1 ...
##  $ temp_superficie: num [1:41472] 273 280 285 289 292 ...
##  $ temperatura    : num [1:41472] 272 282 285 291 293 ...
##  $ presion        : num [1:41472] 835 940 960 990 1000 1000 1000 1000 1000 1000 ...
##  $ ozono          : num [1:41472] 304 304 298 276 274 264 258 252 250 250 ...
##  $ nube_baja      : num [1:41472] 7.5 11.5 16.5 20.5 26 30 29.5 26.5 27.5 26 ...
##  $ nube_media     : num [1:41472] 34.5 32.5 26 14.5 10.5 9.5 11 17.5 18.5 16.5 ...
##  $ nube_alta      : num [1:41472] 26 20 16 13 7.5 8 14.5 19.5 22.5 21 ...
```

3.- Nombres de las variables

```
colnames(BD)
```

```
##  [1] "latitud"         "longitud"        "anio"            "mes"
##  [5] "temp_superficie" "temperatura"     "presion"         "ozono"
##  [9] "nube_baja"       "nube_media"      "nube_alta"
```

4.- Enbusca de datos perdidos

```
anyNA(BD)
```

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

Trtamiento de la matriz

El head se utiliza par copiar el nombre de la variable que no quieres

```
head(BD)
```

```
## # A tibble: 6 x 11
##   latitud longitud  anio   mes temp_superficie temperatura presion ozono
##     <dbl>    <dbl> <int> <int>           <dbl>       <dbl>   <dbl> <dbl>
## 1    36.2    -114.  1995     1            273.        272.     835   304
## 2    33.7    -114.  1995     1            280.        282.     940   304
```

```
## 3    31.2    -114.  1995     1            285.        285.     960   298
## 4    28.7    -114.  1995     1            289.        291.     990   276
## 5    26.2    -114.  1995     1            292.        293.    1000   274
## 6    23.7    -114.  1995     1            294.        294.    1000   264
## # ... with 3 more variables: nube_baja <dbl>, nube_media <dbl>, nube_alta <dbl>
```

filtara las variables,quedarse con las cuantitativas

```
BD[c("anio","mes","nube_baja","nube_alta","nube_media","longitud")]<-NULL
```

Se eliminan las cualitativas y en el proceso se observa que otras variables que no son cualitativas en el scaterplot se puedo observar que no hay relación por eso se decide quitar desde este punto para trabajar mejor.

Al tener muchas observaciones podemos filtrar y trabajar con menos

```
BD1 <- BD[1:1000,]
dim(BD1)
```

```
## [1] 1000    5
```

se vuelve a visualizar para asegurarse que se removieron las variables no deseadas y la dimensión de la matriz es menor

```
str(BD1)
```

```
## tibble [1,000 x 5] (S3: tbl_df/tbl/data.frame)
##  $ latitud        : num [1:1000] 36.2 33.7 31.2 28.7 26.2 ...
##  $ temp_superficie: num [1:1000] 273 280 285 289 292 ...
##  $ temperatura    : num [1:1000] 272 282 285 291 293 ...
##  $ presion        : num [1:1000] 835 940 960 990 1000 1000 1000 1000 1000 1000 ...
##  $ ozono          : num [1:1000] 304 304 298 276 274 264 258 252 250 250 ...
```

ACP pasos a paso

1.- transformar la matriz en un data frame

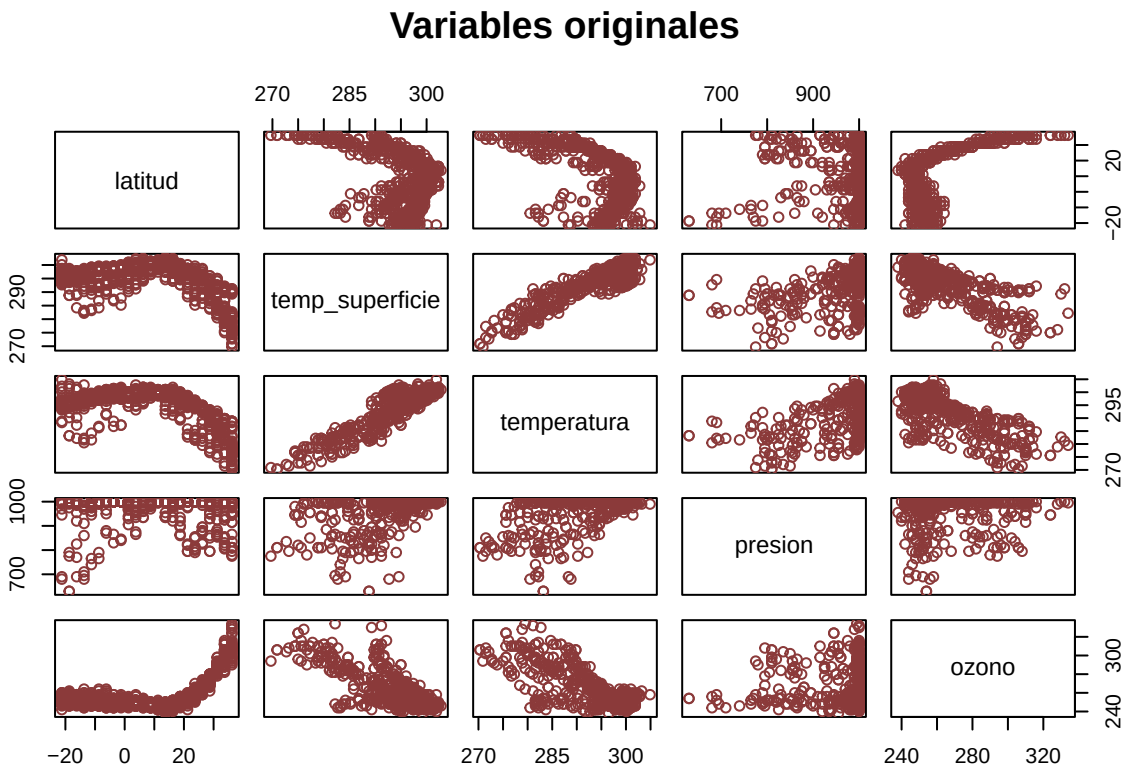
```
BD_1<-data.frame(BD1)
```

2.- definir n (individuos) y p (variables)

```
n<-dim(BD_1)[1]
p<-dim(BD_1)[2]
```

3.- Generación de grafico **scaterplot**

```
pairs(BD_1,col="indianred4", pch=1,
      main="Variables originales")
```



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

```
mu<-colMeans(BD_1)
s<-cov(BD_1)
```

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

```
es<-eigen(s)
es
```

```
## eigen() decomposition
## $values
## [1] 2761.763649  497.175304  120.404318    8.700765    2.569059
##
## $vectors
##             [,1]        [,2]        [,3]        [,4]         [,5]
## [1,]  0.04409286  0.67637736  0.73371967 -0.02053513 -0.042464673
## [2,] -0.06238732 -0.15904116  0.20807185  0.70482050  0.656313873
## [3,] -0.07671687 -0.18311455  0.14689443  0.60979398 -0.753098210
## [4,] -0.99200312  0.09924033 -0.03296757 -0.06916995  0.014485299
## [5,]  0.06486889  0.68836037 -0.62904429  0.35520866 -0.009061681
```

5.1.- Separción de la matriz de valores propios

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

```
## [1] 2761.763649  497.175304  120.404318    8.700765    2.569059
```

5.2.- Separción de la matriz de vectores propios

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

```
##             [,1]        [,2]        [,3]        [,4]         [,5]
## [1,]  0.04409286  0.67637736  0.73371967 -0.02053513 -0.042464673
## [2,] -0.06238732 -0.15904116  0.20807185  0.70482050  0.656313873
## [3,] -0.07671687 -0.18311455  0.14689443  0.60979398 -0.753098210
## [4,] -0.99200312  0.09924033 -0.03296757 -0.06916995  0.014485299
## [5,]  0.06486889  0.68836037 -0.62904429  0.35520866 -0.009061681
```

6.- Calcular la Proporcion de variabilidad

6.1.- Para la matriz de valores propios

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

```
## [1] 0.8145322311 0.1466328627 0.0355110756 0.0025661332 0.0007576974
```

6.2.- Acumulada

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

```
## [1] 0.8145322 0.9611651 0.9966762 0.9992423 1.0000000
```

7.- Obtención de la **matriz de correlaciones**

```
R<-cor(BD_1)
```

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

```
eR<-eigen(R)
eR
```

```
## eigen() decomposition
## $values
## [1] 3.29716886 1.04457840 0.49065573 0.10244023 0.06515677
##
## $vectors
##            [,1]       [,2]        [,3]       [,4]       [,5]
## [1,]  0.3602959 0.5345910  0.74254660 -0.1436620  0.1112885
## [2,] -0.5144367 0.1249429  0.34855729  0.5795488 -0.5122256
## [3,] -0.5322367 0.1258595  0.07901102  0.1734179  0.8152098
## [4,] -0.3077955 0.7469260 -0.42694715 -0.3538861 -0.1996101
## [5,]  0.4769988 0.3533590 -0.37229793  0.6986954  0.1443211
```

9.- Separacionnde la matriz de valores propiosapartir de la **matriz de correlaciones**

9.1.- Separccion de la matriz d evalores propios

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

```
## [1] 3.29716886 1.04457840 0.49065573 0.10244023 0.06515677
```

9.2.- separacion de la matriz de vectores propios

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

```
##            [,1]      [,2]        [,3]       [,4]       [,5]
## [1,]  0.3602959 0.5345910  0.74254660 -0.1436620  0.1112885
## [2,] -0.5144367 0.1249429  0.34855729  0.5795488 -0.5122256
## [3,] -0.5322367 0.1258595  0.07901102  0.1734179  0.8152098
## [4,] -0.3077955 0.7469260 -0.42694715 -0.3538861 -0.1996101
## [5,]  0.4769988 0.3533590 -0.37229793  0.6986954  0.1443211
```

10.- calculo de la proporción de variabilidad

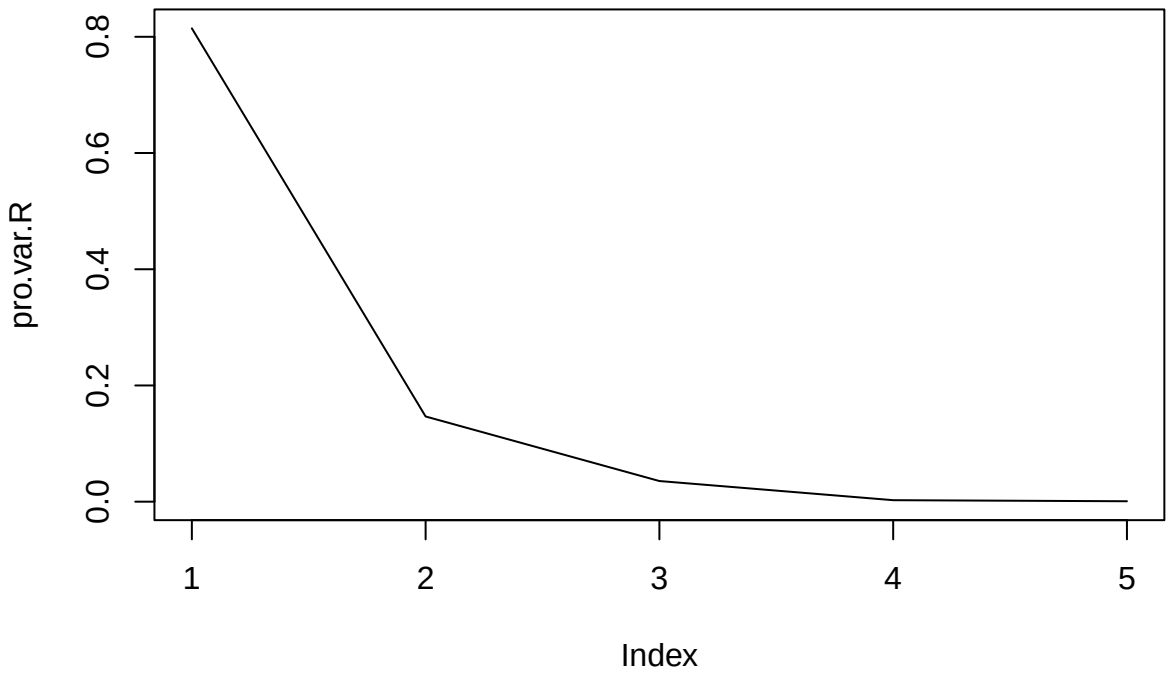
10.1.- para la matriz de valores propios

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

```
## [1] 0.8145322311 0.1466328627 0.0355110756 0.0025661332 0.0007576974
```

Grafíco de la proporción de variabilidad

```
plot(pro.var.R, type="l")
```



10.2.- Acumulada

En este punto se seleccionan el numero de componentes, siguiendo el criterio del 80% de la varíanza explicada.

para este ejemplo se van a seleccionar **1** factores de **0.8145322** varianza explicada aunque con el grafíco vemos la inflexion (codo) en el segundo factor con el primero estamos cubiertos

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

```
## [1] 0.8145322 0.9611651 0.9966762 0.9992423 1.0000000
```

11.- Calcularc 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 unos

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

12.2.- Construccion de la matriz centrada

```
X.cen<-as.matrix(BD_1)-ones%*%mu
```

13.- Construcción de la matriz diagonal de las varianzas

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

```
##          [,1]     [,2]     [,3]     [,4]     [,5]
## [1,] 297.6476  0.00000  0.00000    0.000   0.0000
## [2,]   0.0000 33.96656  0.00000    0.000   0.0000
## [3,]   0.0000  0.00000 40.21556    0.000   0.0000
## [4,]   0.0000  0.00000  0.00000 2722.839   0.0000
## [5,]   0.0000  0.00000  0.00000    0.000 295.9446
```

14.- Construccion de la matriz centrada multiplicada por Dx^1/2

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

15.- Construccion de los coeficientes o scores **eigen.vec matriz de autovectores* Semuestran las primeras 10 observaciones

```
scores<-Y%*%eigen.vec.R
scores[1.10,]
```

```
## [1]  6.6591092 -1.2047742 -0.2072927 -0.2477616  0.1564832
```

16.- Nombramos las columnas de acurdo a los componentes

```
colnames(scores)<-c("PC1","PC2","PC3","PC4","PC5" )
```

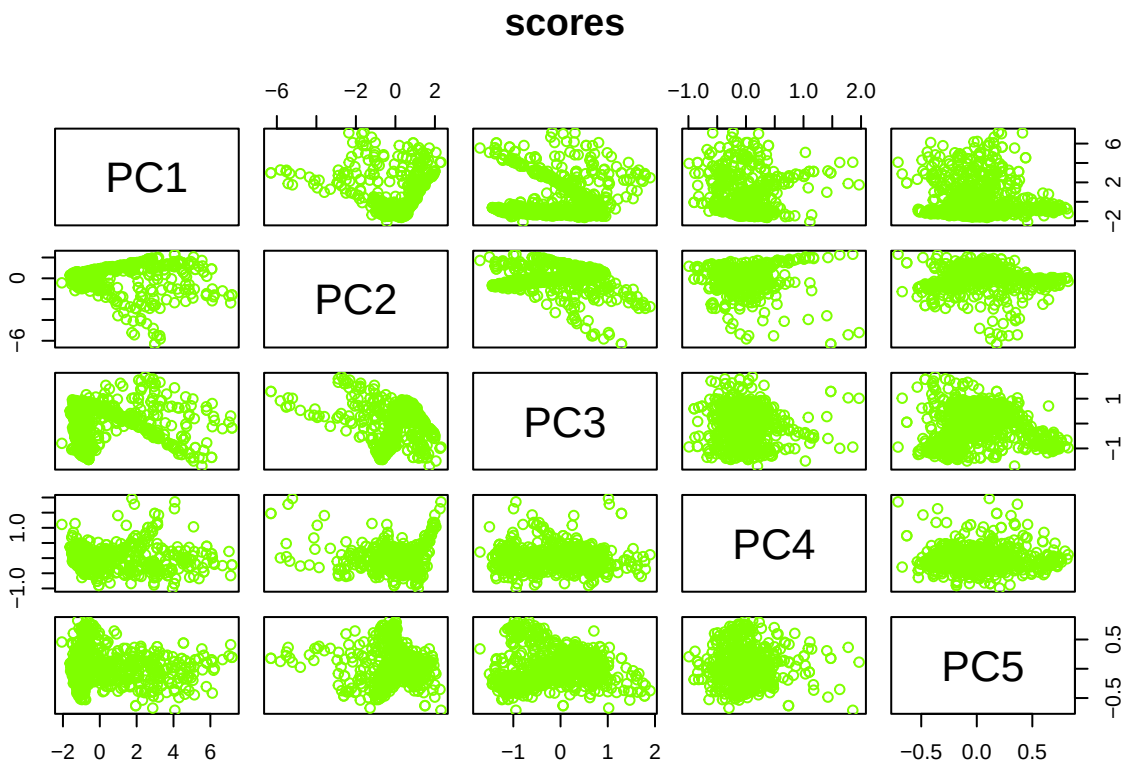
17.- Visualizar los scores

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

```
##              PC1        PC2         PC3           PC4          PC5
##  [1,]  6.6591092 -1.2047742 -0.20729269 -0.2477616375  0.156483219
##  [2,]  4.5397335  0.5671151 -0.64130063  0.0133118229  0.439431826
##  [3,]  3.4924956  0.8238429 -0.43413507  0.2538972970  0.225116997
##  [4,]  1.7857708  0.9318185  0.03273638 -0.2144697202  0.212428882
##  [5,]  1.1953756  1.0584121  0.08514284  0.0003336745  0.143519807
##  [6,]  0.6227355  0.8342704  0.31898959 -0.1714824664 -0.007765833
##  [7,]  0.2408816  0.6728372  0.40771014 -0.2775461171 -0.024750144
##  [8,] -0.4619236  0.5886562  0.65616328 -0.1094019872 -0.085553467
##  [9,] -0.8039161  0.5266947  0.71089815  0.0337549818 -0.160935959
## [10,] -0.9315700  0.4672256  0.61469850  0.0791479501 -0.061339317
```

18.- Generacion del grafico de los scores

```
pairs(scores, main="scores", col="chartreuse", pch=1)
```



ACP via sintetizada

1.- calculo de la varianza a las coolumnas 1= filas, 2= columnas

```
apply(BD_1, 2, var)
```

```
##         latitud temp_superficie     temperatura         presion          ozono
##       297.64758        33.96656        40.21556      2722.83881       295.94458
```

2.- aplicar la funcion **prcom** pra reducir la dimencinalidad y centrado por la nedia y escalada por la desviacion estandar (dividir entre sd)

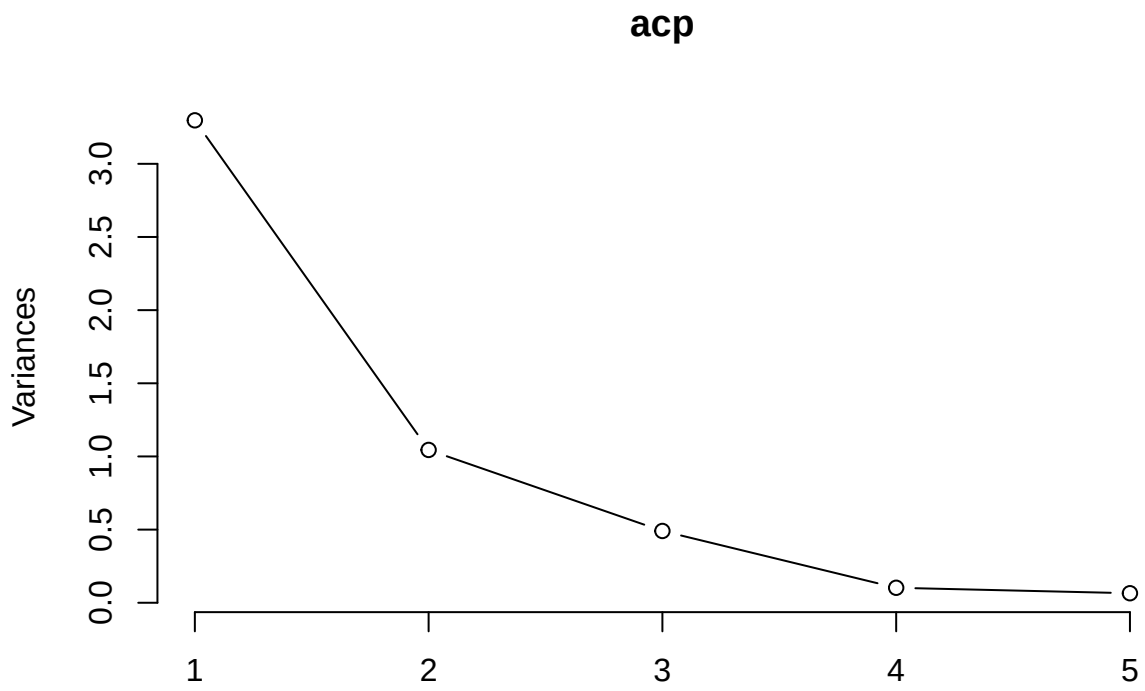
```
acp<-prcomp(BD_1, center=TRUE, scale=TRUE)
acp
```

```
## Standard deviations (1, .., p=5):
## [1] 1.8158108 1.0220462 0.7004682 0.3200629 0.2552583
##
## Rotation (n x k) = (5 x 5):
##                          PC1         PC2         PC3         PC4         PC5
```

```
## latitud         -0.3602959 -0.5345910 -0.74254660 -0.1436620 -0.1112885
## temp_superficie  0.5144367 -0.1249429 -0.34855729  0.5795488  0.5122256
## temperatura      0.5322367 -0.1258595 -0.07901102  0.1734179 -0.8152098
## presion          0.3077955 -0.7469260  0.42694715 -0.3538861  0.1996101
## ozono           -0.4769988 -0.3533590  0.37229793  0.6986954 -0.1443211
```

3.- generacion del grafico **screeplot**

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



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

```
summary(acp)
```

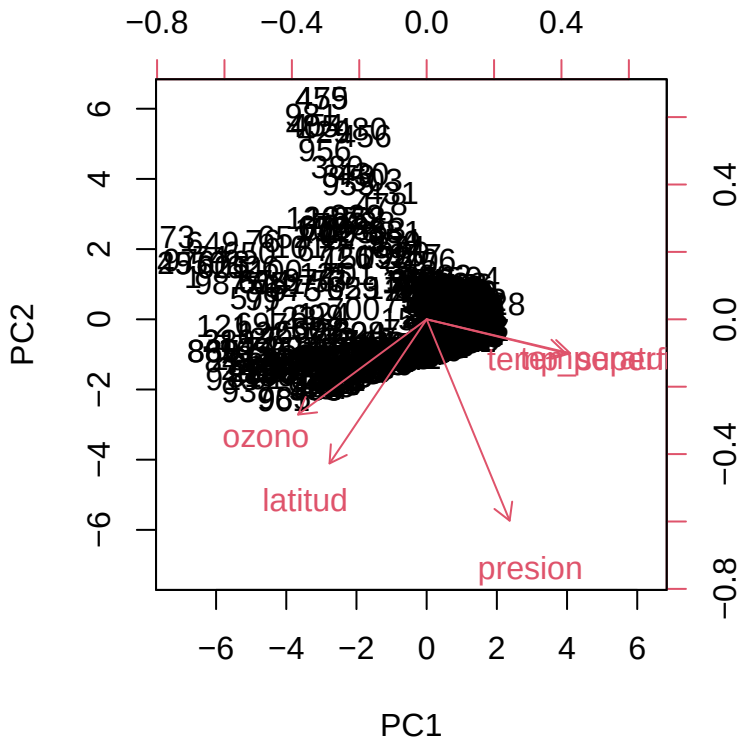
```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     1.8158 1.0220 0.70047 0.32006 0.25526
## Proportion of Variance 0.6594 0.2089 0.09813 0.02049 0.01303
## Cumulative Proportion  0.6594 0.8683 0.96648 0.98697 1.00000
```

4.1.- En este punto se seleccionan el numero de componentes, siguiendo el criterio del 80% de la varíanza explicada.

para este ejemplo se van a seleccionar **2** factores de **0.8683** varianza explicada

5.- Construcción del Biplot

```
biplot(acp, scale=0)
```



Construción de los componentes principales con las variables originales

combinación lineal de las variables originales

Primer componente

Z1 = -0.360(latitud) + 0.514(temp_superficie) + 0.532(temperatura) + 0.307(presión) - 0.476(ozono)

Segundo componente

Z2 = -0.534(latitud) - 0.124(temp_superficie) - 0.125(temperatura) - 0.746(presión) - 0.353(ozono)