PRÁCTICA 4: ANÁLISIS DE COMPONENTES PRINCIPALES

ANÁLISIS ESTADÍSTICO MULTIVARIANTE

GRADO EN CIENCIA E INGENIERÍA DE DATOS

Sumario: En esta práctica mostramos cómo resumir la información contenida en muchas variables aleatorias (relacionadas) en unas pocas variables denominadas componentes principales. Al pasar de muchas variables a unas pocas, estamos reduciendo la dimensión del problema. Para ello necesitaremos disponer de una muestra de las variables en estudio. El Análisis de Componentes Principales (ACP o PCA por sus siglas en inglés) se podrá usar para estudiar (resumir) la información contenida en la muestra y las principales características de sus individuos. También se podrá usar para detectar grupos.

1. Estudio descriptivo inicial

Como en el resto de técnicas vistas anteriormente, debemos comenzar con un análisis descriptivo previo de nuestros datos. En particular, para dar respuesta a las cuestiones:

1) ¿Tiene sentido plantearse un Análisis de Componentes Principales para estos datos? La respuesta será afirmativa si las variables presentan correlación.

2) ¿Todas las variables se miden en magnitudes similares y presentan dispersión similar? Recordar que si no es así, las que tengan mayor dispersión tendrán más importancia a la hora de obtener las componentes principales.

3) ¿Conviene usar la matriz de covarianzas, o es preferible la matriz de correlaciones a la hora de extraer las componentes principales? Recordar que esto equivale, respectivamente, a trabajar con los datos originales, o bien con los datos tipificados (estandarizados).

Comenzamos cargando el conjunto de datos *LifeCycleSavings*, que está disponible en R. Podemos ver su descripción en la ayuda de R. El fichero contiene 5 variables medidas en 50 países diferentes. Las variables son:

sr: incremento de los ahorros personales 1960-1970.

pop15: porcentaje población menor de 15 años.

pop75: porcentaje población mayor de 75.

dpi: ingresos per-capita.

ddpi: crecimiento del dpi 1960-1970.

```
d <- LifeCycleSavings
#help(LifeCycleSavings)
```

Mostramos la estructura del conjunto de datos y hacemos un resumen numérico:

```
str(d)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ sr   : num  11.43 12.07 13.17 5.75 12.88 ...
##  $ pop15: num  29.4 23.3 23.8 41.9 42.2 ...
##  $ pop75: num  2.87 4.41 4.43 1.67 0.83 2.85 1.34 0.67 1.06 1.14 ...
##  $ dpi  : num  2330 1508 2108 189 728 ...
```

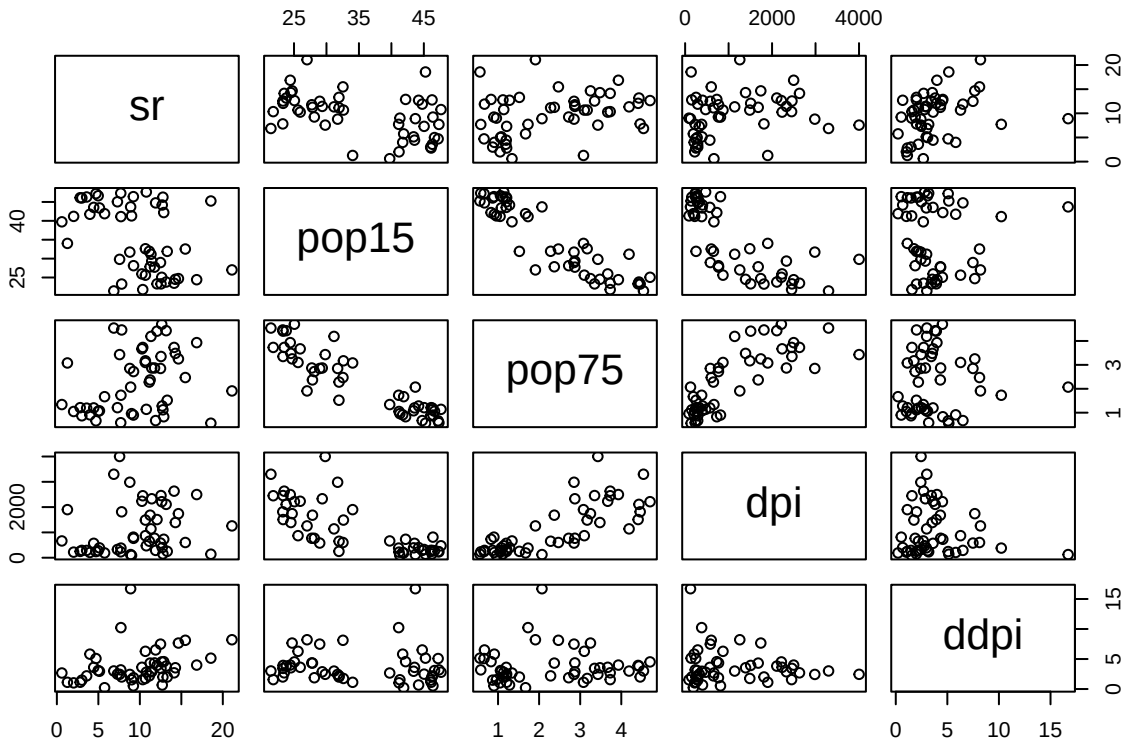
```
##  $ ddpi : num  2.87 3.93 3.82 0.22 4.56 2.43 2.67 6.51 3.08 2.8 ...
```

```
summary(d)
```

```
##        sr             pop15           pop75            dpi
##  Min.   : 0.600   Min.   :21.44   Min.   :0.560   Min.   :  88.94
##  1st Qu.: 6.970   1st Qu.:26.21   1st Qu.:1.125   1st Qu.: 288.21
##  Median :10.510   Median :32.58   Median :2.175   Median : 695.66
##  Mean   : 9.671   Mean   :35.09   Mean   :2.293   Mean   :1106.76
##  3rd Qu.:12.617   3rd Qu.:44.06   3rd Qu.:3.325   3rd Qu.:1795.62
##  Max.   :21.100   Max.   :47.64   Max.   :4.700   Max.   :4001.89
##       ddpi
##  Min.   : 0.220
##  1st Qu.: 2.002
##  Median : 3.000
##  Mean   : 3.758
##  3rd Qu.: 4.478
##  Max.   :16.710
```

Realizamos la matriz de nube de puntos para ver si existen variables correladas:

```
plot(d)
```




```
# plot(d, pch=20, cex=0.8)
```

Podemos observar algunos pares de variables correlados (relacionados linealmente), aunque no de manera muy estrecha.

Calculamos las matrices de covarianzas y de correlaciones del conjunto de datos:

```
cov(d)
```

```
##               sr        pop15         pop75         dpi          ddpi
## sr     20.074046   -18.678638    1.83049898    978.2825    3.91901061
## pop15 -18.678638    83.754110  -10.73166612  -6857.2360   -1.25610710
## pop75   1.830499   -10.731666    1.66609082   1006.5607    0.09379918
## dpi   978.282487 -6857.235988 1006.56074980 981821.1551 -368.21350800
## ddpi    3.919011    -1.256107    0.09379918   -368.2135    8.23615739
```

```
cor(d)
```

```
##               sr       pop15       pop75        dpi        ddpi
## sr     1.0000000 -0.45553809  0.31652112  0.2203589  0.30478716
## pop15 -0.4555381  1.00000000 -0.90847871 -0.7561881 -0.04782569
## pop75  0.3165211 -0.90847871  1.00000000  0.7869995  0.02532138
## dpi    0.2203589 -0.75618810  0.78699951  1.0000000 -0.12948552
## ddpi   0.3047872 -0.04782569  0.02532138 -0.1294855  1.00000000
```

Para saber si las correlaciones anteriores son significativas, podemos usar la función *rcorr()* del paquete *Hmisc*, que proporciona tanto el valor de las correlaciones como el p-valor del contraste de significación de cada correlación.

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.1.3
```

```
rcorr(as.matrix(d))
```

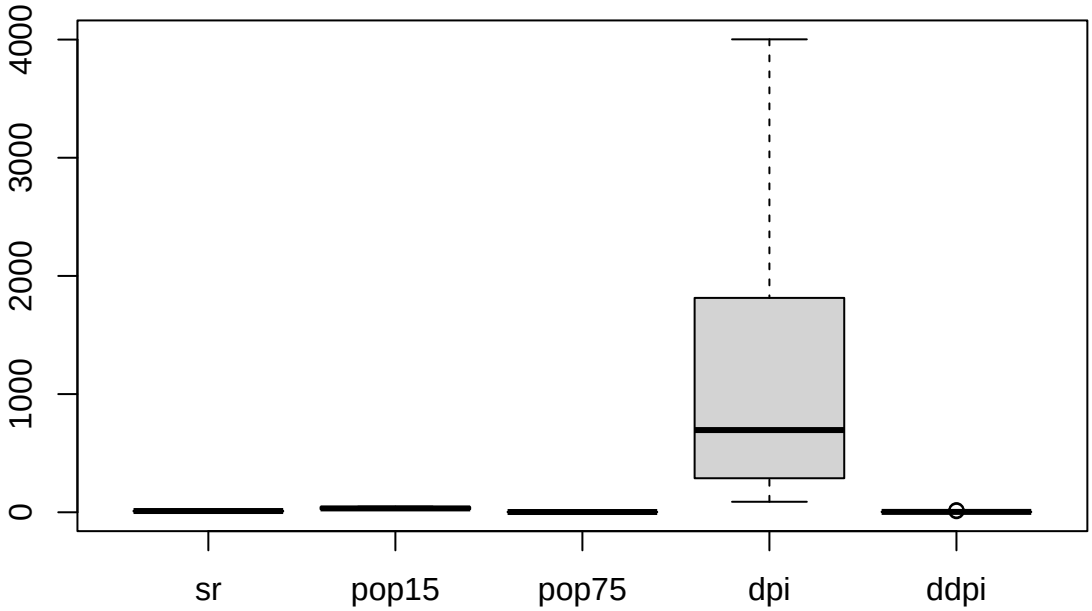
```
##          sr pop15 pop75   dpi  ddpi
## sr     1.00 -0.46  0.32  0.22  0.30
## pop15 -0.46  1.00 -0.91 -0.76 -0.05
## pop75  0.32 -0.91  1.00  0.79  0.03
## dpi    0.22 -0.76  0.79  1.00 -0.13
## ddpi   0.30 -0.05  0.03 -0.13  1.00
##
## n= 50
##
##
## P
##       sr     pop15  pop75  dpi    ddpi
## sr           0.0009 0.0251 0.1241 0.0314
## pop15 0.0009        0.0000 0.0000 0.7415
## pop75 0.0251 0.0000        0.0000 0.8614
## dpi   0.1241 0.0000 0.0000        0.3701
## ddpi  0.0314 0.7415 0.8614 0.3701
```

Observamos varios p-valores inferiores a 0.05, lo que indica que las correspondiente correlaciones son significativas.

Conclusión 1: Como existe correlación entre algunas variables, sí tiene sentido intentar reducir la dimensión haciendo ACP.

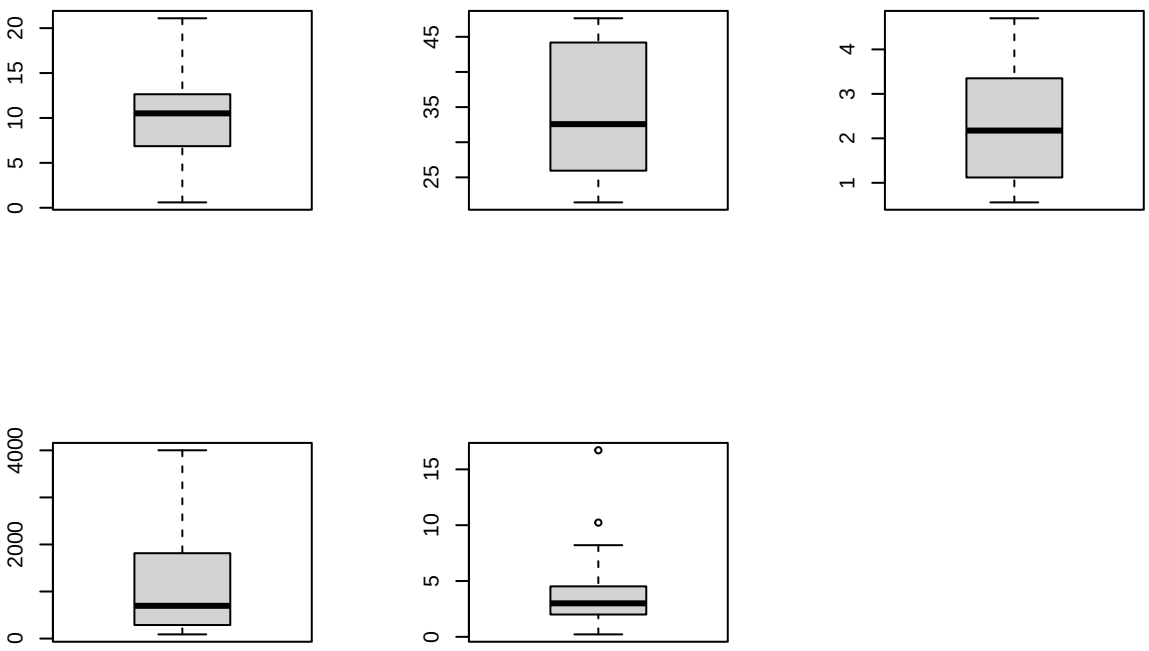
La forma en la que se distribuye cada variable y la posible existencia de atípicos puede visualizarse a través de los diagramas de caja-bigotes:

```
boxplot(d)
```



Como las variables se mueven en magnitudes muy distintas, conviene hacer cada diagrama de caja por separado:

```
par(mfrow = c(2, 3))
boxplot(d[, 1])
boxplot(d[, 2])
boxplot(d[, 3])
boxplot(d[, 4])
boxplot(d[, 5])
```



Calculemos la media y desviación típica de cada variable:

```
apply(d, 2, mean)
```

```
##        sr     pop15     pop75       dpi      ddpi
##    9.6710   35.0896    2.2930 1106.7584    3.7576
```

```
apply(d, 2, sd)
```

```
##         sr      pop15      pop75        dpi       ddpi
##   4.480407   9.151727   1.290771 990.868889   2.869871
```

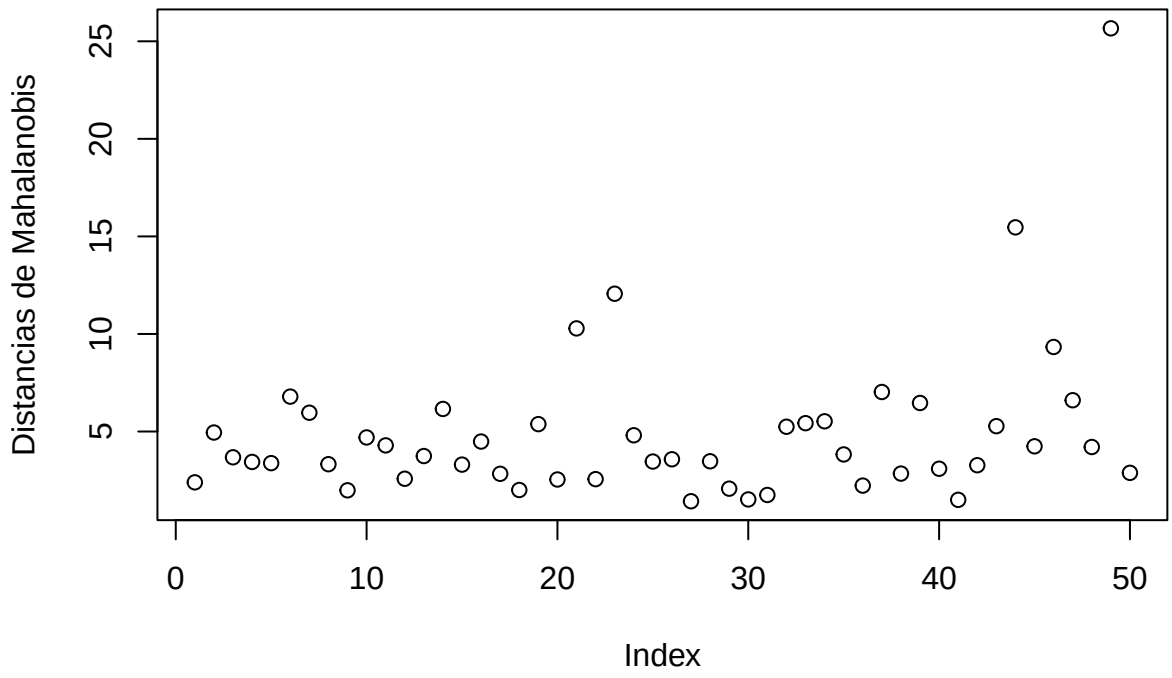
```
# Otra alternativa:
# colMeans(d)
# sapply(d, mean)
# sapply(d, sd)
```

Observamos que las variables se mueven en magnitudes muy distintas y con diferente dispersión.

Conclusión 2: La variable *dpi* tendrá mucho más peso que el resto a la hora de extraer las componentes principales. Para evitarlo, debemos trabajar con las variables estandarizadas (o equivalentemente, trabajar con la matriz de correlaciones).

Por último, veamos si hay observaciones que se alejan especialmente del resto. Para ello usaremos la distancia de Mahalanobis al cuadrado:

```
md <- mahalanobis(d, colMeans(d), cov(d))
plot(md, ylab = 'Distancias de Mahalanobis')
```




```
sort(md)
```

```
##         Norway         Turkey      Nicaragua         Panama       Colombia
##       1.425905       1.497556       1.518641       1.746897       1.986479
##       Honduras    New Zealand   South Africa      Australia          India
##       2.001921       2.070036       2.227677       2.394116       2.540639
##          Italy        Ecuador      Guatamala          Spain       Malaysia
##       2.558624       2.578377       2.831859       2.843018       2.881037
##    Switzerland        Tunisia        Germany          China         Brazil
##       3.091808       3.271746       3.302971       3.324294       3.378909
##        Bolivia     Luxembourg    Netherlands          Malta        Belgium
##       3.440807       3.461258       3.473722       3.577171       3.677412
##        Finland       Portugal        Uruguay      Venezuela        Denmark
##       3.742888       3.824790       4.205328       4.241523       4.289174
##         Greece     Costa Rica          Korea        Austria       Paraguay
##       4.485636       4.695837       4.807564       4.947422       5.244924
## United Kingdom        Iceland           Peru    Philippines          Chile
##       5.275044       5.378798       5.427633       5.523590       5.963195
##         France         Sweden        Jamaica         Canada South Rhodesia
##       6.155482       6.462193       6.603365       6.789279       7.025241
##         Zambia        Ireland          Japan  United States          Libya
##       9.331990      10.285536      12.062666      15.463758      25.664265
```

Observamos que Libia es el país más alejado del conjunto de datos.

2. Cálculo de las componentes principales

La obtención de las componentes principales puede realizarse con las funciones *princomp()* y *prcomp()* . Los resultados pueden variar ligeramente debido a los métodos usados en cada caso.

```
PCA <- princomp(d, cor = TRUE)
summary(PCA, loadings = TRUE)
```

```
## Importance of components:
##                            Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     1.6799041 1.1207437 0.7775124 0.48953545 0.27872068
## Proportion of Variance 0.5644156 0.2512133 0.1209051 0.04792899 0.01553704
## Cumulative Proportion  0.5644156 0.8156289 0.9365340 0.98446296 1.00000000
##
## Loadings:
##       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## sr     0.308  0.554  0.750  0.130  0.134
## pop15 -0.571                0.416  0.707
## pop75  0.560 -0.101 -0.212 -0.390  0.692
## dpi    0.514 -0.266 -0.145  0.801
## ddpi          0.782 -0.609  0.123
```

```
# PCA$center
# PCA$scale
# observar que en esta estandarización, R divide entre la desv típica,
# no entre la cuasidesviación típica
```

Standard deviation: indica la desviación típica de cada componente, es decir, la raíz cuadrada de cada valor propio. Por tanto, dichos valores al cuadrado se corresponden con los valores propios.

Proportion of Variance: indica la proporción de varianza explicada por cada componente.

Cumulative Proportion: indica cómo se acumula la varianza explicada conforme consideramos nuevas componentes. En este caso, vemos que las tres primeras componentes ya explican más del 93% de variabilidad.

Las **cargas (loadings)** son los vectores propios unitarios correspondientes a cada valor propio (en este caso de la matriz de correlaciones). Observamos que algunas cargas no aparecen, por tener valores bajos (aunque no sean nulos). Para poder visualizarlos todos haremos:

```
L <- PCA$loadings
L[ , ]
```

```
##            Comp.1     Comp.2      Comp.3     Comp.4      Comp.5
## sr     0.30846174  0.5542456  0.75014389  0.1301489  0.13419615
## pop15 -0.57065322  0.0126733  0.02654937  0.4162082  0.70729072
## pop75  0.56043119 -0.1009938 -0.21182951 -0.3904265  0.69167374
## dpi    0.51350640 -0.2663021 -0.14501556  0.8013443 -0.04703435
## ddpi   0.03787232  0.7820068 -0.60883333  0.1234770 -0.03326304
```

Podemos comprobar que las cargas (loadings) obtenidas con la función *princomp()* coinciden con los vectores propios unitarios de la matriz de correlaciones.

```
eigen(cor(d))
```

```
## eigen() decomposition
## $values
## [1] 2.82207781 1.25606655 0.60452546 0.23964496 0.07768522
##
## $vectors
```

```
##             [,1]       [,2]        [,3]       [,4]        [,5]
## [1,]  0.30846174  0.5542456  0.75014389 -0.1301489 -0.13419615
## [2,] -0.57065322  0.0126733  0.02654937 -0.4162082 -0.70729072
## [3,]  0.56043119 -0.1009938 -0.21182951  0.3904265 -0.69167374
## [4,]  0.51350640 -0.2663021 -0.14501556 -0.8013443  0.04703435
## [5,]  0.03787232  0.7820068 -0.60883333 -0.1234770  0.03326304
```

También podemos comprobar que, al utilizar la matriz de correlaciones, la suma de todos los valores propios coincide con el número total de variables (k = 5).

```
sum(eigen(cor(d))$values)
```

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

Las **puntuaciones (scores)** se corresponden con las coordenadas que tendrían los individuos de la muestra (en esta caso países) en el nuevo sistema de referencia formado por las componentes principales. Es decir, los valores que se obtienen al sustituir los datos originales de cada país en la expresión que define cada componente principal. Para obtener los scores haremos:

```
PCA$scores
```

```
##                    Comp.1      Comp.2      Comp.3       Comp.4       Comp.5
## Australia       1.36528994 -0.41014926  0.19444121  0.5721114631 -0.130783981
## Austria         2.04902157  0.05451922 -0.07596611 -0.7818715755  0.278413095
## Belgium         2.41694452 -0.00223304  0.04296459 -0.2478784163  0.332489024
## Bolivia        -1.50180971 -1.15583585  0.35382809 -0.5156855515  0.160459153
## Brazil         -1.05306237  0.85012549  0.69003979  0.5932072262 -0.131762954
## Canada          1.35972099 -1.03359418 -0.24406991  1.1241361456 -0.062629754
## Chile          -1.58862456 -1.23054191 -1.06384364 -0.1712545480 -0.393237592
## China          -1.55643632  1.39980697  0.20532875  0.4571009726 -0.049947032
## Colombia       -2.03815052 -0.43372897 -0.28719438  0.0621201586  0.140117239
## Costa Rica     -1.56455012  0.15621651  0.71464948  0.4006072269  0.430922666
## Denmark         2.61994057  0.43944747  0.65625794  0.3657714010  0.200985456
## Ecuador        -2.06305917 -0.86615597 -0.35571615 -0.0631453495  0.152554781
## Finland         0.90776832  0.17867324  0.02589724  0.1832313218 -0.510952948
## France          2.48362237  0.07603810 -0.25332534 -0.1715549583  0.547725215
## Germany         2.10846430 -0.19428353  0.14559926  0.3097181703 -0.321397489
## Greece          0.93001508  0.80615418 -0.49824935 -0.7353928103 -0.290452905
## Guatamala      -2.23581716 -1.10966237 -0.24965686 -0.0236074421 -0.050288335
## Honduras       -2.12401901 -0.01266767  0.23736077  0.2884996784  0.016507941
## Iceland         0.20781271 -2.05486798 -1.10645823 -0.0021804333  0.081942726
## India          -1.58550596 -0.30387338  0.75142950 -0.2545064257 -0.181464049
## Ireland         1.20273945 -0.16716892  0.11586932 -0.7174203317  0.777958625
## Italy           1.65227597  0.33054767  0.55651519 -0.4910772342 -0.054224229
## Japan           1.27392504  2.63182027  0.99636899  0.3976873733 -0.551641676
## Korea          -1.86455847  0.21648242 -1.02065844 -0.0841918925 -0.382530785
## Luxembourg      2.18826505 -1.01397068  0.10798324 -0.0279038206 -0.277886233
## Malta           0.43563038  2.04640214  0.08554582 -0.2242897501  0.045693536
## Norway          1.80661188 -0.39236509 -0.29385546  0.0888575555 -0.002932437
## Netherlands     1.80367736  1.43406814 -0.27698668  0.0643486388 -0.217748948
## New Zealand     0.78300484 -0.60116940  0.38810116 -0.1283244403  0.316496052
## Nicaragua      -1.69263959 -0.33634093  0.19697846  0.0247231532  0.171275283
## Panama         -1.66042013 -0.44994515 -0.56749239  0.1235096470 -0.061095381
## Paraguay       -1.96116942 -1.35950370 -0.35454612 -0.4074346178 -0.354421940
## Peru           -1.21796836 -0.18672000  1.47303829  0.1049517098  0.323807347
## Philippines    -1.52508855  0.27224933  1.27081045  0.1064458909  0.397545818
```

```
## Portugal        0.60004161  1.46741361 -0.35329486 -0.6380131251 -0.110040161
## South Africa    0.03541852 -0.12759994  0.64467290 -0.5379846576 -0.168277176
## South Rhodesia -0.35844344  0.25869067  1.23579439 -0.5784628240 -0.494463129
## Spain           0.69294913  0.46122019  0.16082031 -0.7027045876 -0.188672802
## Sweden          2.78770795 -1.34902763 -1.05187533  0.3627225706 -0.030752066
## Switzerland     2.45481415 -0.27751238  0.48323508  0.3580972094 -0.053517853
## Turkey         -1.75856517 -0.48478109 -0.26556802 -0.0004103802 -0.099916239
## Tunisia        -2.13053636 -1.24695247 -0.25874636 -0.1779046015  0.139175893
## United Kingdom  1.91240048 -1.09338068 -0.43865703 -0.7579614002  0.180494516
## United States   2.18258748 -1.50694925 -0.70877865  1.6564196623  0.015901885
## Venezuela      -1.55134335 -0.73920434  0.92284265  0.5520534885  0.167176377
## Zambia         -1.27041105  1.90538194  1.66736141  0.5260518978  0.154338320
## Jamaica        -1.05638365  1.78787286 -1.49862861  0.0797828334  0.066054864
## Uruguay         0.39254017 -0.52179694  0.28859202 -0.8223841254 -0.286941263
## Libya          -1.03587286  3.76419265 -2.70027114  0.2000664334  0.422584706
## Malaysia       -2.25875456  0.12465965 -0.68848726  0.2613234709 -0.062641161
```

Veamos cómo se haría el ACP con la función *prcomp()* :

```
PCA_bis <- prcomp(d, scale. = TRUE)
summary(PCA_bis)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5
## Standard deviation     1.6799 1.1207 0.7775 0.48954 0.27872
## Proportion of Variance 0.5644 0.2512 0.1209 0.04793 0.01554
## Cumulative Proportion  0.5644 0.8156 0.9365 0.98446 1.00000
```

```
PCA_bis$rotation  # loadings
```

```
##               PC1        PC2         PC3        PC4         PC5
## sr    -0.30846174 -0.5542456  0.75014389 -0.1301489 -0.13419615
## pop15  0.57065322 -0.0126733  0.02654937 -0.4162082 -0.70729072
## pop75 -0.56043119  0.1009938 -0.21182951  0.3904265 -0.69167374
## dpi   -0.51350640  0.2663021 -0.14501556 -0.8013443  0.04703435
## ddpi  -0.03787232 -0.7820068 -0.60883333 -0.1234770  0.03326304
```

```
PCA_bis$x         # scores
```

```
##                       PC1          PC2         PC3           PC4          PC5
## Australia      -1.35156809  0.406027056  0.19248698 -0.5663614533  0.129469535
## Austria        -2.02842787 -0.053971271 -0.07520262  0.7740133703 -0.275614902
## Belgium        -2.39265301  0.002210597  0.04253278  0.2453871127 -0.329147341
## Bolivia         1.48671577  1.144219119  0.35027194  0.5105026506 -0.158846457
## Brazil          1.04247856 -0.841581302  0.68310454 -0.5872451933  0.130438670
## Canada         -1.34605510  1.023206037 -0.24161688 -1.1128380081  0.062000293
## Chile           1.57265808  1.218174336 -1.05315147  0.1695333530  0.389285355
## China           1.54079334 -1.385738197  0.20326509 -0.4525068764  0.049445039
## Colombia        2.01766607  0.429369771 -0.28430793 -0.0614958196 -0.138708990
## Costa Rica      1.54882559 -0.154646452  0.70746689 -0.3965809214 -0.426591675
## Denmark        -2.59360884 -0.435030804  0.64966222 -0.3620952133 -0.198965450
## Ecuador         2.04232438  0.857450669 -0.35214102  0.0625107067 -0.151021529
## Finland        -0.89864479 -0.176877487  0.02563696 -0.1813897542  0.505817612
## France         -2.45866071 -0.075273880 -0.25077929  0.1698307441 -0.542220299
## Germany        -2.08727316  0.192330878  0.14413591 -0.3066053459  0.318167282
## Greece         -0.92066795 -0.798051923 -0.49324169  0.7280017402  0.287533706
## Guatamala       2.21334606  1.098509699 -0.24714769  0.0233701754  0.049782911
```

```
## Honduras        2.10267155  0.012540353  0.23497518 -0.2856001105 -0.016342028
## Iceland        -0.20572409  2.034215512 -1.09533776  0.0021585189 -0.081119160
## India           1.56957082  0.300819300  0.74387726  0.2519485073  0.179640244
## Ireland        -1.19065131  0.165488790  0.11470478  0.7102098941 -0.770139747
## Italy          -1.63566976 -0.327225496  0.55092193  0.4861416594  0.053679248
## Japan          -1.26112145 -2.605369140  0.98635497 -0.3936904138  0.546097398
## Korea           1.84581871 -0.214306660 -1.01040031  0.0833457214  0.378686157
## Luxembourg     -2.16627188  1.003779763  0.10689795  0.0276233731  0.275093335
## Malta          -0.43125207 -2.025834761  0.08468604  0.2220355246 -0.045234293
## Norway         -1.78845451  0.388421624 -0.29090207 -0.0879644921  0.002902964
## Netherlands    -1.78554949 -1.419655030 -0.27420283 -0.0637019024  0.215560460
## New Zealand    -0.77513525  0.595127348  0.38420055  0.1270347147 -0.313315106
## Nicaragua       1.67562770  0.332960535  0.19499873 -0.0244746730 -0.169553879
## Panama          1.64373207  0.445422976 -0.56178880 -0.1222683125  0.060481341
## Paraguay        1.94145867  1.345840004 -0.35098275  0.4033396936  0.350859820
## Peru            1.20572716  0.184843374  1.45823351 -0.1038968920 -0.320552919
## Philippines     1.50976063 -0.269513085  1.25803816 -0.1053760558 -0.393550281
## Portugal       -0.59401089 -1.452665362 -0.34974407  0.6316007701  0.108934202
## South Africa   -0.03506255  0.126317497  0.63819361  0.5325776394  0.166585905
## South Rhodesia  0.35484090 -0.256090697  1.22337403  0.5726489798  0.489493524
## Spain          -0.68598464 -0.456584698  0.15920398  0.6956420507  0.186776545
## Sweden         -2.75969008  1.335469215 -1.04130345 -0.3590770252  0.030442992
## Switzerland    -2.43014203  0.274723238  0.47837832 -0.3544981511  0.052979972
## Turkey          1.74089070  0.479908794 -0.26289893  0.0004062557  0.098912030
## Tunisia         2.10912339  1.234419971 -0.25614583  0.1761165702 -0.137777105
## United Kingdom -1.89317989  1.082391646 -0.43424831  0.7503435044 -0.178680455
## United States  -2.16065137  1.491803649 -0.70165506 -1.6397718060 -0.015742063
## Venezuela       1.53575156  0.731774963  0.91356762 -0.5465050714 -0.165496170
## Zambia          1.25764277 -1.886231887  1.65060358 -0.5207648099 -0.152787141
## Jamaica         1.04576646 -1.769903829 -1.48356663 -0.0789809756 -0.065390979
## Uruguay        -0.38859494  0.516552613  0.28569152  0.8141187485  0.284057358
## Libya           1.02546181 -3.726360606 -2.67313205 -0.1980556644 -0.418337516
## Malaysia        2.23605293 -0.123406759 -0.68156762 -0.2586970377  0.062011586
```

Nota: Obsérvese que las puntuaciones (scores) no coinciden al aplicar las funciones *princomp()* y *prcomp()*. Esto se debe a que en el primer caso, se estandariza dividiendo entre la desviación típica de cada variable y en el segundo caso se estandariza dividiendo entre la cuasi-desviación típica. Tampoco coinciden las cargas (loadings), observamos que aparecen cambiadas de signo: como sabemos, las componentes principales son únicas salvo signo.

3. Análisis de las componentes principales

En esta sección veremos cómo:

1) A partir de las cargas (loadings), dar una interpretación de las componentes principales.

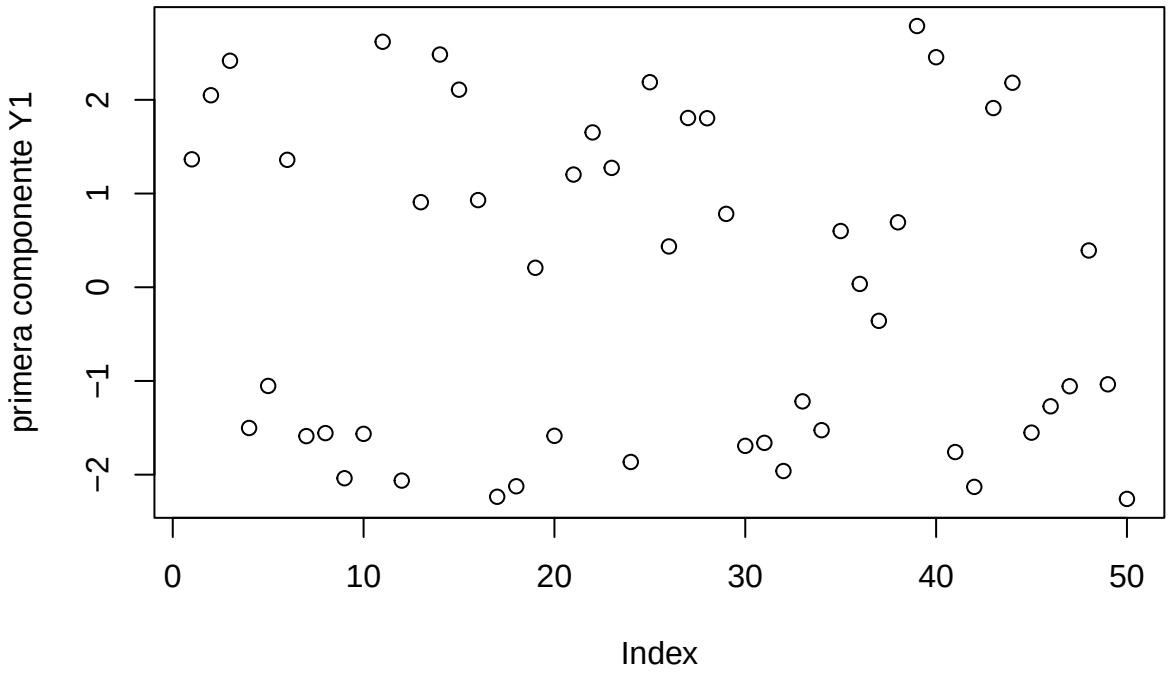
2) Representar los individuos de la muestra (países en este caso) en el nuevo sistema de referencia dado por las primeras componentes principales. Concretamente, los gráficos de las puntuaciones (scores) estandarizadas y sin estandarizar.

3) Realizar el gráfico de saturaciones, para facilitar la interpretación de las componentes.

Para **interpretar las componentes principales**, miraremos las cargas (loadings), es decir, los coeficientes de las componentes que queremos analizar. Si miramos las cargas de la primera componente principal, teniendo en cuenta que las variables están estandarizadas (y tendrán valores similares), podemos afirmar que las variables que más influyen son (por orden de influencia): pop15 (negativa), pop75 (positiva), dpi

(positiva) y sr (positiva). Por lo tanto, la primera componente tomará valores grandes en los países con valores pequeños en pop15 y grandes en las otras tres. Por lo tanto, la primera componente nos indicará los países que tienen poblaciones envejecidas (alta pop75 y baja pop15) y ricos (altos valores en dpi y sr). En resumen, la primera componente principal podría interpretarse como el nivel de desarrollo económico, de manera que los países muy desarrollados económicamente tomarán valores altos (scores altos) en la primera componente principal.

```
S <- PCA$scores
plot(S[, 1], ylab = 'primera componente Y1')
```




```
which.max(S[, 1])
```

```
## Sweden
##     39
```

```
which.min(S[, 1])
```

```
## Malaysia
##       50
```

```
sort(S[, 1])
```

```
##       Malaysia      Guatamala        Tunisia       Honduras        Ecuador
##    -2.25875456    -2.23581716    -2.13053636    -2.12401901    -2.06305917
##       Colombia       Paraguay          Korea         Turkey      Nicaragua
##    -2.03815052    -1.96116942    -1.86455847    -1.75856517    -1.69263959
##         Panama          Chile          India     Costa Rica          China
##    -1.66042013    -1.58862456    -1.58550596    -1.56455012    -1.55643632
##      Venezuela    Philippines        Bolivia         Zambia           Peru
```

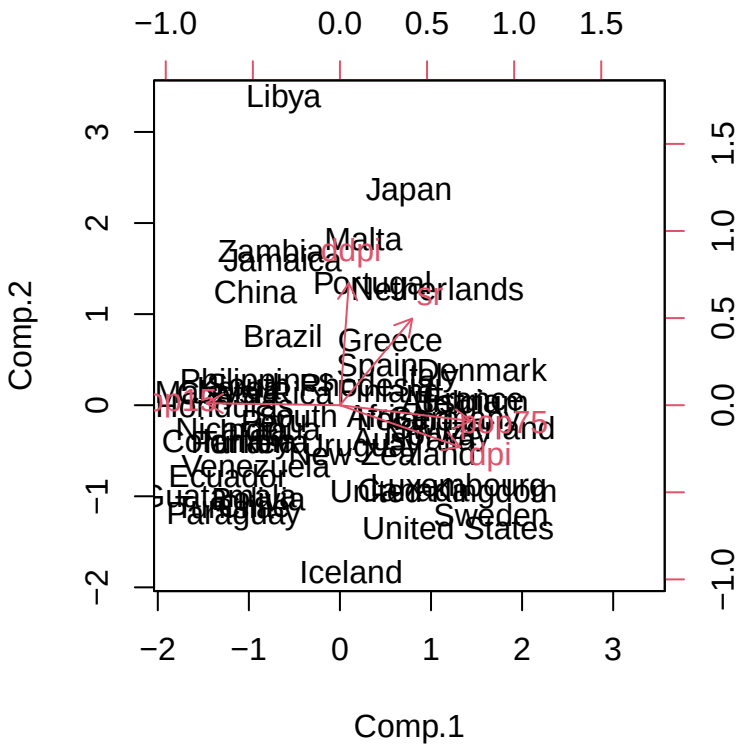
```
##    -1.55134335    -1.52508855    -1.50180971    -1.27041105    -1.21796836
##        Jamaica         Brazil          Libya South Rhodesia   South Africa
##    -1.05638365    -1.05306237    -1.03587286    -0.35844344     0.03541852
##        Iceland        Uruguay         Malta       Portugal          Spain
##     0.20781271     0.39254017     0.43563038     0.60004161     0.69294913
##    New Zealand        Finland        Greece        Ireland          Japan
##     0.78300484     0.90776832     0.93001508     1.20273945     1.27392504
##         Canada      Australia          Italy    Netherlands         Norway
##     1.35972099     1.36528994     1.65227597     1.80367736     1.80661188
## United Kingdom        Austria        Germany  United States     Luxembourg
##     1.91240048     2.04902157     2.10846430     2.18258748     2.18826505
##        Belgium    Switzerland         France        Denmark         Sweden
##     2.41694452     2.45481415     2.48362237     2.61994057     2.78770795
```

Observamos que Suecia es el país con mayor score en la componente 1 (el más desarrollado económicamente) y Malasia el país con menor score en dicha componente (el menos desarrollado econónimcamente).

De forma similar, se podría intentar dar un significado al resto de componentes principales, aunque no siempre será fácil.

Vamos ahora a **representar los individuos de la muestra** (países en este caso) **en el nuevo sistema de referencia** dado por las 2 primeras componentes principales. Indicar que R estandariza las puntuaciones (scores) en el gráfico siguiente:

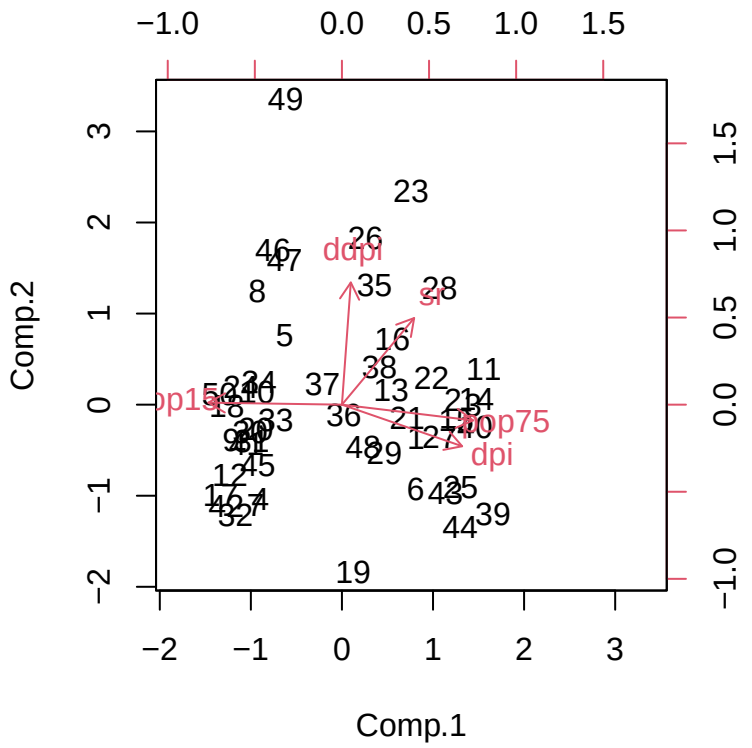
```
biplot(PCA, pc.biplot = TRUE)
```




```
# biplot(PCA)
```

Para una mejor visualización, pondremos que use numeración en lugar del nombre del país.

```
# biplot(PCA, pc.biplot=TRUE, xlabs=row.names(d))
biplot(PCA, pc.biplot = TRUE, xlabs = 1:dim(d)[1])
```



Obsérvese que el país 39 (Suecia), es el que tiene mayor valor en la Comp.1 (componente 1), indicando que es el país con más desarrollo económico.

Los vectores en rojo se corresponden con las **saturaciones**, con las escalas en la derecha y arriba del gráfico. Las puntuaciones estandarizadas aparecen con las etiquetas de los datos en negro, con las escalas abajo (Y1) y en la izquierda (Y2). Este gráfico es la 'mejor' proyección bidimensional de nuestros datos muestrales (que vienen dados por 5 variables, es decir, dimensión 5).

Las saturaciones de este gráfico se pueden usar para interpretar las componentes. Las variables con vectores largos (norma cercana a 1) estarán bien representadas por las dos primeras componentes, mientras que las que tengan vectores cortos estarán mal representadas (se pierden al proyectar por ser casi perpendiculares). En este ejemplo, todas las variables están bien representadas en este gráfico. Además, se aprecia que pop75, dpi y, en menor medida, sr, hacen crecer la primera componente mientras que pop15 la hace disminuir y ddpi no influye en ella. Lógicamente, la interpretación es la misma que antes. También podemos observar que la segunda componente crece cuando crece ddpi (incremento ingresos per-capita) y, en menor medida, sr (incremento de los ahorros personales), decrece un poco si crece dpi y casi no se ve afectada ni por pop15 ni por pop75. Por lo tanto, se podría interpretar como un índice del crecimiento en la década 1960-1970.

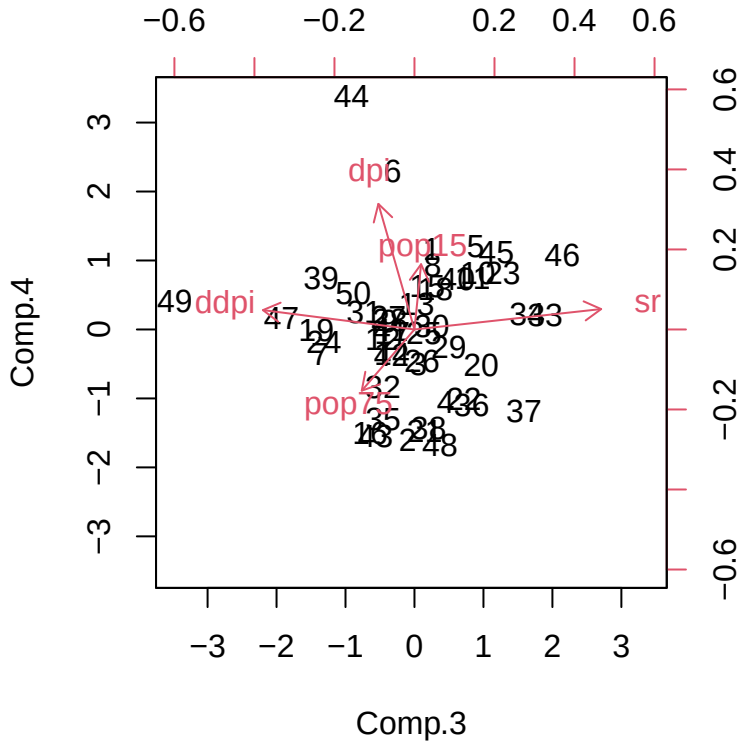
En la siguiente sección veremos con más detalle las saturaciones.

Las puntuaciones se usarán para decir cómo serán (aproximadamente) los individuos de la muestra (países) en esas características. A la derecha tendremos a los países más desarrollados y con poblaciones envejecidas (Suecia, US, etc.) a la izquierda lo contrario (Honduras, Guatemala, etc.), arriba a los países que más se desarrollaron durante esa época (Líbia, Japón, etc.) y debajo los que menos (Islandia, US, Paraguay, etc.).

También nos podemos fijar en una variable concreta. Por ejemplo, con respecto a sr podríamos decir que los países con mayores incrementos de los ahorros personales (valores sr) deberían ser Japón, Malta y Holanda. Si vemos los datos de sr, podemos comprobar que efectivamente Japón es el que tiene un valor mayor (21.10) pero que el segundo es Zambia (18.56). Es lógico que al proyectar las variables originales, se pierda algo de la información contenida en ellas.

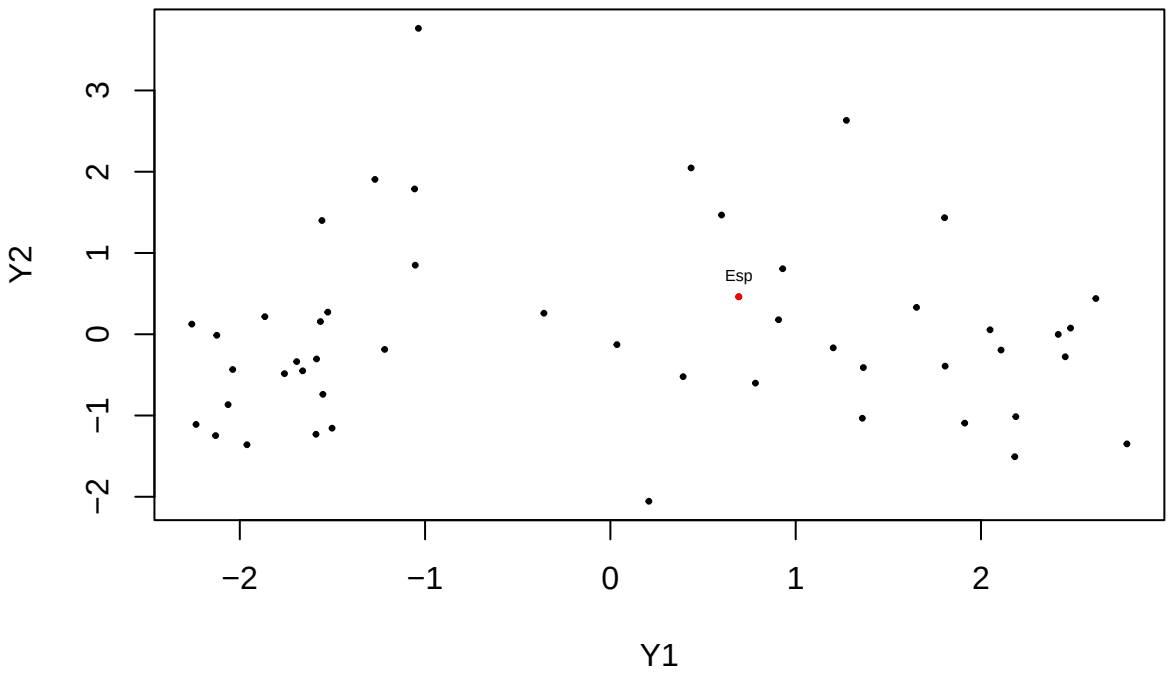
Si queremos hacer un gráfico de las componentes tercera y cuarta haremos:

```
biplot(PCA, pc.biplot = TRUE, choices = c(3, 4), xlabs = 1:dim(d)[1])
```



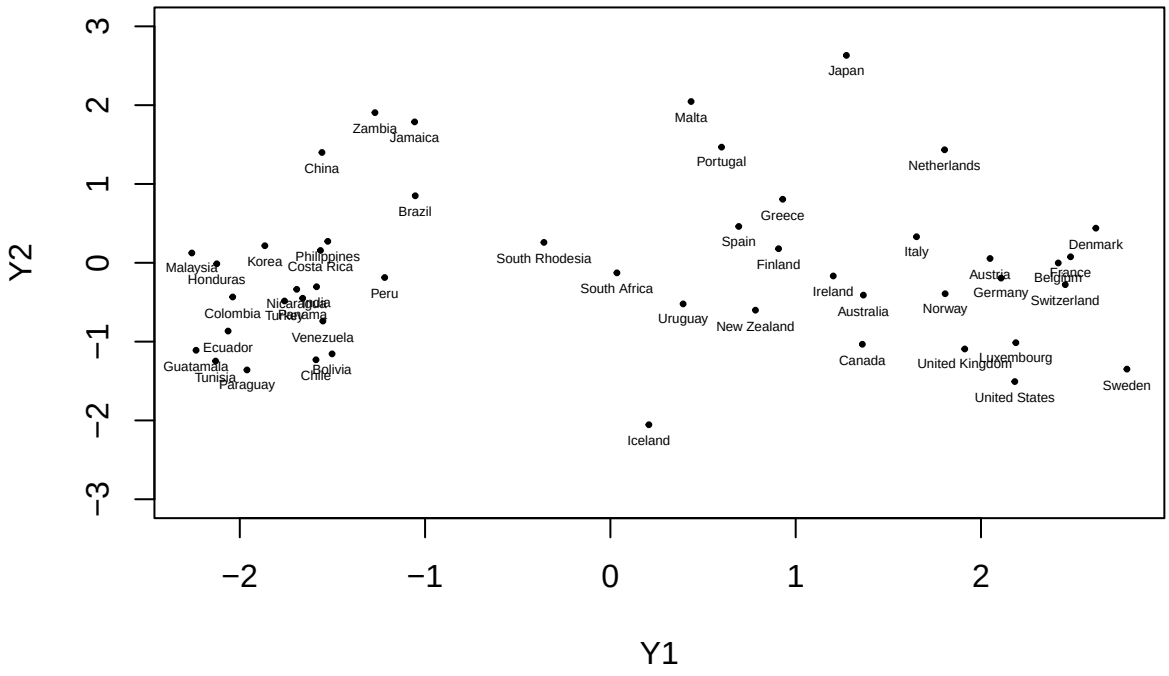
Podemos hacer un gráfico solo de las puntuaciones (sin estandarizar) para las dos primeras componentes. Además, vamos a identificar a España:

```
plot(S[, 1], S[, 2], xlab = 'Y1', ylab = 'Y2', pch = 20, cex = 0.5)
text(S[38, 1], S[38, 2] + 0.25, labels = 'Esp', cex = 0.5)
points(S[38, 1], S[38, 2], col = 'red', pch = 20, cex = 0.5)
```



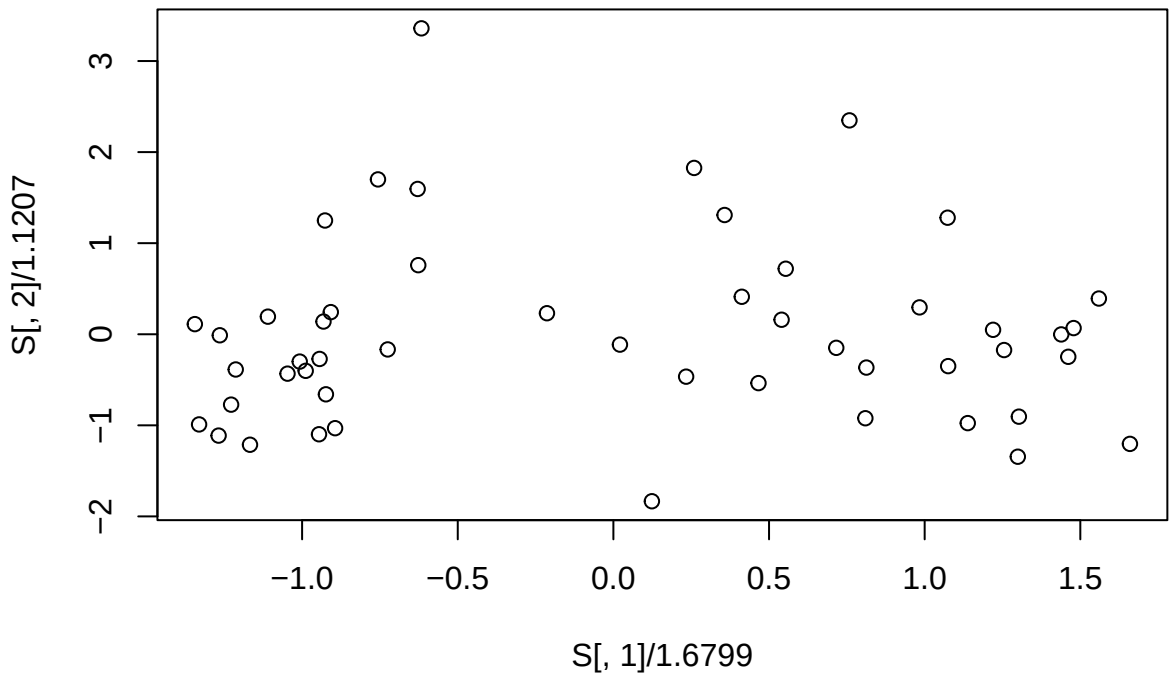
Para que pongan las etiquetas (debajo) en todos los países haremos:

```
plot(S[, 1], S[, 2], xlab = 'Y1', ylab = 'Y2', pch = 20, cex = 0.5, ylim = c(-3, 3))
text(S[, 1], S[, 2] - 0.2, xlab = 'Y1', ylab = 'Y2', labels = row.names(d), cex = 0.4)
```



Note que estos gráficos no coinciden con el realizado automáticamente por R con la función *biplot()*, ya que esa función dibuja las puntuaciones estandarizadas, es decir, hace:

```
plot(S[, 1]/1.6799, S[, 2]/1.1207)
```




```
# pairs(PCA$scores[,1:3])
```

En el gráfico de puntuaciones sin estandarizar, la mayor dispersión de la primera componente nos indica que esta componente es más importante (tiene más información) a la hora de distinguir los datos.

4. Las saturaciones

Recordar que las saturaciones miden las correlaciones (relaciones lineales) entre las variables iniciales y las componentes principales, así que son de utilidad para interpretar las componentes.

Para calcular las saturaciones bastará con hacer:

```
cor(d, S)
```

```
##            Comp.1      Comp.2      Comp.3      Comp.4       Comp.5
## sr     0.51818615  0.62116726  0.58324614  0.06371249  0.037403244
## pop15 -0.95864269  0.01420352  0.02064246  0.20374866  0.197136552
## pop75  0.94147065 -0.11318816 -0.16470006 -0.19112760  0.192783778
## dpi    0.86264152 -0.29845643 -0.11275139  0.39228642 -0.013109446
## ddpi   0.06362187  0.87642926 -0.47337543  0.06044637 -0.009271098
```

En el caso de trabajar con variables estandarizadas (equivalentemente, matriz de correlaciones), las saturaciones coinciden con el producto de las cargas por la raíz del correspondiente valor propio:

```
S1 <- L[ , 1] * PCA$sdev[1]
S1
```

```
##          sr       pop15       pop75         dpi        ddpi
##  0.51818615 -0.95864269  0.94147065  0.86264152  0.06362187
```

Las saturaciones al cuadrado nos indicarán cuanta información (en tanto por 1) tendrá cada componente de cada variable:

```
S1^2
```

```
##          sr       pop15       pop75         dpi        ddpi
## 0.268516885 0.918995810 0.886366987 0.744150387 0.004047742
```

De esta forma, comprobamos que la variable mejor representada en la primera componente Y1 es pop15, con un 91.89958%, y que de la última variable ddpi Y1 prácticamente no tiene ninguna información (0.4047742%).

Como las componentes son incorreladas, las correlaciones múltiples al cuadrado o **comunalidades** serán la suma de las correlaciones al cuadrado:

$$Corr^2(X_i, (Y_1, ..., Y_p)) = \sum_{j=1}^{p} Corr^2(X_i, Y_j).$$

Estos valores nos indicarán la información (en tanto por 1) que mantienen las p primeras componentes sobre cada variable. Por ejemplo, si decidimos usar las dos primeras componentes, las comunalidades se calcularán mediante:

```
SAT <- cor(d, S)
COM2 <- SAT[, 1]^2 + SAT[, 2]^2
COM2
```

```
##        sr     pop15     pop75       dpi      ddpi
## 0.6543657 0.9191976 0.8991785 0.8332266 0.7721760
```

Se observa que la variable mejor representada en las dos primeras componentes es pop15 de la que se mantiene un 91.91976% de su información y que la peor representada es sr con un 65.43657%. Esto también se vio en el gráfico biplot.

Se puede comprobar que la media de las comunalidades que acabamos de calcular es 0.8156289, coincidiendo con la información que (en promedio) mantienen las dos primeras componentes Y1 y Y2.

```
mean(COM2)
```

```
## [1] 0.8156289
```

Podemos comprobar que ocurre lo mismo con las informaciones individuales de cada componente.

```
mean(SAT[ , 1]^2)
```

```
## [1] 0.5644156
```

```
mean(SAT[ , 2]^2)
```

```
## [1] 0.2512133
```

Si en lugar de hacer la media de las saturaciones al cuadrado, hacemos la suma, podemos comprobar que se obtienen los valores propios (informaciones) de cada componente principal (solo si usamos la matriz de correlaciones). Por ejemplo, comprobamos que sumando los valores de la primera columna de las saturaciones al cuadrado obtenemos 2.822078, es decir, el mayor valor propio de la matriz de correlaciones.

```
sum(SAT[ , 1]^2)
```

```
## [1] 2.822078
```

La correlación múltiple al cuadrado $Corr^2(X_i, (Y_1, ..., Y_p))$ es el máximo de las correlaciones que se pueden obtener con combinaciones lineales de las componentes $Y_1, ..., Y_p$. Además, el máximo de esas correlaciones se obtiene con los coeficientes incluidos en la matriz de cargas L. Por ejemplo, la mejor combinación lineal de las dos primeras componentes para estimar (linealmente) sr es la que se obtiene cortando L[1,], es decir, $Z_1 = 0.3084617 * Y_1 + 0.5542456 * Y_2$. Si calculamos Z1 con la expresión anterior y luego calculamos la

correlación al cuadrado entre Z1 y la variable sr, obtenemos 0.6543657 (recordar que sr viene representada en un 65.43657% si consideramos las dos primeras componentes):

```
Z1 <- 0.3084617 * S[ , 1]+ 0.5542456 * S[ , 2]
cor(d[ , 1], Z1)^2
```

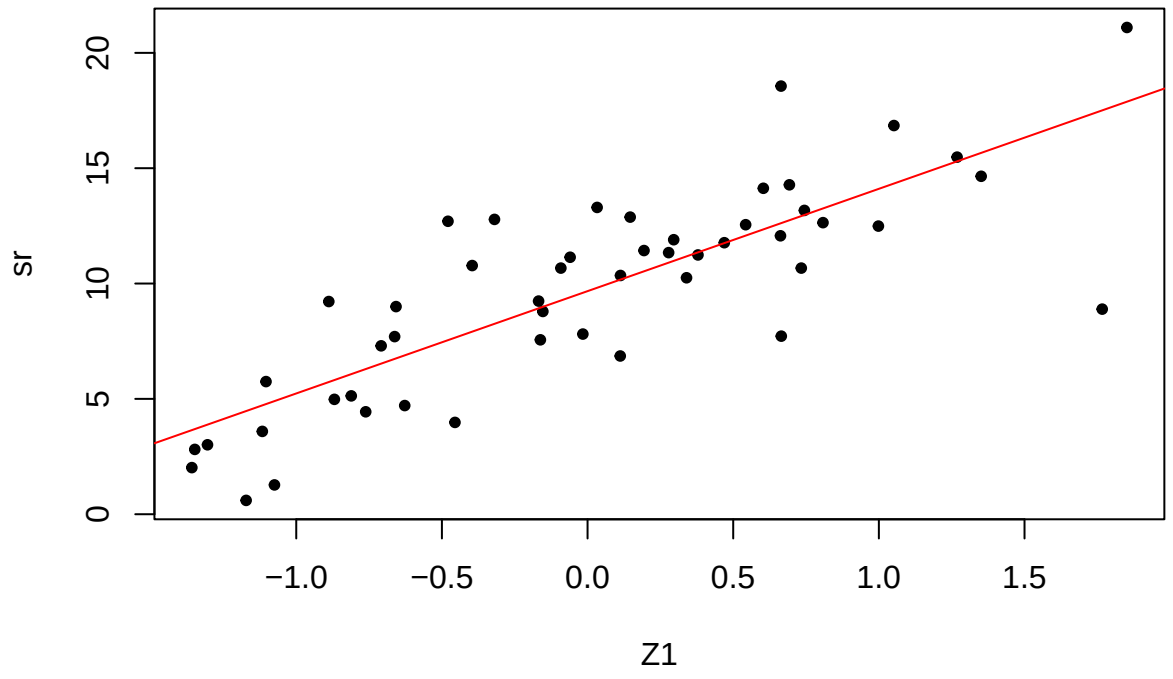
```
## [1] 0.6543657
```

La variable Z1 se podría usar para predecir sr usando las técnicas de regresión lineal mediante:

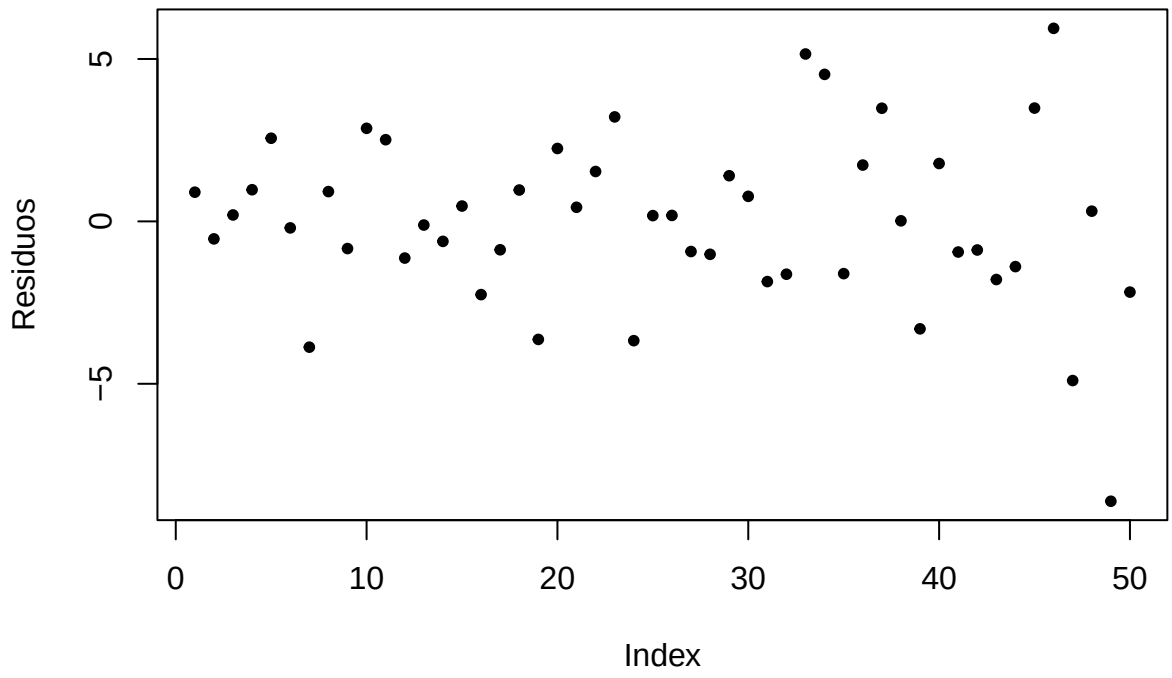
```
lm(d$sr ~ Z1)
```

```
##
## Call:
## lm(formula = d$sr ~ Z1)
##
## Coefficients:
## (Intercept)           Z1
##       9.671        4.435
```

```
plot(Z1, d$sr, pch = 20, ylab = 'sr')
abline(lm(d$sr ~ Z1), col = 'red')
```




```
plot(d$sr - 9.671 - 4.435 * Z1, pch = 20, ylab = 'Residuos') # gráfico de residuos
```




```
9.671 + 4.435 * Z1  # valores ajustados
```

```
##      Australia        Austria        Belgium        Bolivia         Brazil
##      10.530575      12.608131      12.971958       4.775347      10.320057
##         Canada          Chile          China       Colombia     Costa Rica
##       8.990479       4.472948      10.982589       5.816614       7.914645
##        Denmark        Ecuador        Finland         France        Germany
##      14.335348       4.719598      11.352045      13.255572      12.077873
##         Greece      Guatamala       Honduras        Iceland          India
##      12.924877       3.884702       6.734145       4.904265       6.755039
##        Ireland          Italy          Japan          Korea     Luxembourg
##      10.905466      12.743872      17.882987       7.652363      10.172187
##          Malta         Norway    Netherlands    New Zealand      Nicaragua
##      15.297173      11.178030      15.663534       9.264450       6.528670
##         Panama       Paraguay           Peru    Philippines       Portugal
##       6.293499       3.646298       7.545813       8.253847      14.098892
##   South Africa South Rhodesia          Spain         Sweden    Switzerland
##       9.405803       9.816522      11.752689      10.168645      12.347106
##         Turkey        Tunisia United Kingdom  United States      Venezuela
##       6.073604       3.691259       9.599600       8.952639       5.731697
##         Zambia        Jamaica        Uruguay          Libya       Malaysia
##      12.616622      12.620571       8.925388      17.506581       6.887385
```

Así observamos que la mejor manera de recuperar sr usando Z1 es mediante el modelo lineal $sr = 9.671 + 4.435 * Z_1$. Por ejemplo, para Australia obtendríamos como valor ajustado 10.530575, cuando su verdadero valor de la variable *sr* es 11.43. Para obtener mejores aproximaciones debemos aumentar el número de componentes. Lógicamente, si retenemos todas las componentes, obtendremos los valores exactos.

5. Número de componentes a retener

Una vez realizado un PCA podemos preguntarnos con cuántas componentes principales debemos quedarnos. Las respuesta no es única y puede depender de factores subjetivos. Todas las soluciones serán correctas ya que lo que estamos haciendo es perder algo de información (la menor posible) a cambio de reducir la dimensión inicial (número de variables). A continuación comentamos algunas de las técnicas más usadas. En todos ellas el número de componentes elegidas se representará por p y, lógicamente, siempre se tomarán las p primeras componentes principales (ya que son las que más información tienen).

5.1. Fijar un número concreto de componentes

Una opción válida es fijar un número de componentes concreto. Por ejemplo, si queremos hacer una única gráfica bidimensional, evidentemente, debemos tomar p = 2, con lo que únicamente analizaremos Y1 e Y2. En esta opción es fundamental incluir la información total mantenida por las componentes elegidas y advertir si ese número es bajo. Se suelen tomar números pares de componentes para poder realizar gráficas bidimensionales y el valor más usual es p = 2. Tecleando **summary(PCA)** (ver sección 2) comprobamos que, en nuestro ejemplo, si tomamos p = 2, mantendríamos un 81.56% de la información inicial lo que podemos considerar como aceptable al reducir la dimensión de 5 a 2.

También podríamos informar sobre las comunalidades, es decir, sobre la información mantenida por esas componentes de cada variable (ver sección anterior). En nuestro ejemplo, para p = 2, la variable peor representada es sr de la que mantienen un 65.44%. Por lo tanto, todas las variables están bien representadas. En otros ejemplos nos podremos encontrar con variables que no están representadas en las componentes elegidas. En estos casos es importante señalarlo y, si fuera necesario, aumentar p.

5.2. Fijar un porcentaje mínimo de información mantenida

Si queremos mantener al menos un porcentaje concreto de la variabilidad inicial, deberemos quedarnos con las primeras p componentes que cumplan dicha condición. En nuestro ejemplo, si queremos mantener al menos un 90% de información, debemos tomar p = 3, con lo que mantendríamos un 93.65%.

Otra regla (diferente) podría ser el fijar un porcentaje mínimo para las comunalidades. De esta forma nos aseguramos de que todas las variables originales (sean importantes o no), estén representadas en las componentes. En nuestro ejemplo, si queremos que las comunalidades sean mayores que 0.5 (es decir queremos mantener al menos un 50% de todas las variables), debemos tomar p = 2. Nótese que con esta regla, en nuestro ejemplo, nunca obtendríamos p = 1 a pesar de que Y1 tiene un 56% de la información total.

5.3. Regla de Rao

Esta regla establece que solo serán relevantes las componentes que tengan una variabilidad (varianza o valor propio) mayor que la variabilidad mínima de las variables originales. Si las componentes se calculan usando la matriz de correlaciones, como esto es equivalente a usar las variables estandarizadas, se entiende que las varianzas son 1 y, por lo tanto, se toman solo las componentes con valores propios (varianzas o desviaciones típicas) mayores que uno. En nuestro ejemplo, esta regla nos conduce a p = 2.

Si calculásemos las componentes con la matriz de covarianzas (aunque ya hemos comentado que esto no sería correcto en este ejemplo), el mínimo de las cuasivarianzas muestrales corresponde a la variable pop75 y vale 1.66609082 (hacer var(d$pop75) o cov(d)) y los valores propios de la matriz de covarianzas valen: 981871.2, 43.14338, 13.68328, 6.629537 y 0.2351568 por lo que, con este criterio, tomaríamos p = 4.

```
cov(d)
```

```
##                sr        pop15         pop75         dpi          ddpi
## sr     20.074046   -18.678638    1.83049898    978.2825    3.91901061
## pop15 -18.678638    83.754110  -10.73166612  -6857.2360   -1.25610710
## pop75   1.830499   -10.731666    1.66609082   1006.5607    0.09379918
## dpi   978.282487 -6857.235988 1006.56074980 981821.1551 -368.21350800
```

```
## ddpi    3.919011    -1.256107    0.09379918   -368.2135    8.23615739
```

```
eigen(cov(d))$values
```

```
## [1] 9.818712e+05 4.314338e+01 1.368328e+01 6.629537e+00 2.351568e-01
```

5.4. Regla de Kaiser

Esta regla es similar a la anterior y establece que solo serán relevantes las componentes que tengan una variabilidad mayor que la variabilidad media de las variables originales. Si usamos la matriz de correlaciones para calcular las componentes, como las varianzas iniciales son 1, su media es 1 y este criterio coincide con el de Rao por lo que, en nuestro ejemplo, obtenemos el mismo resultado p = 2.

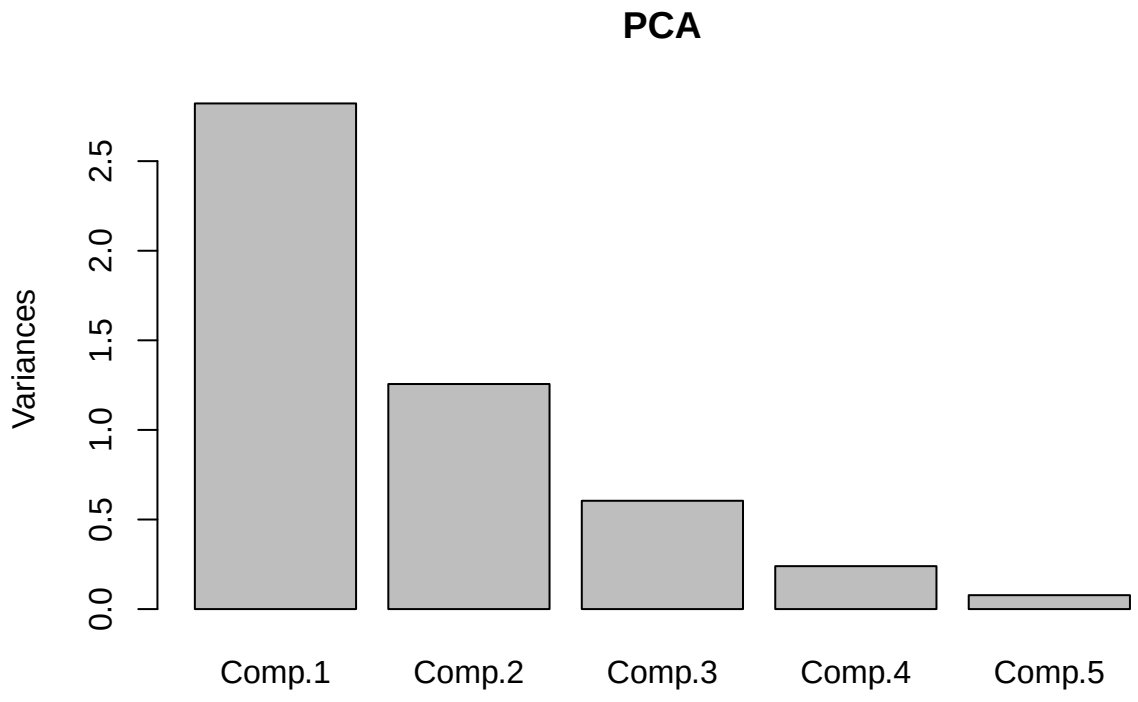
Si calculásemos las componentes con la matriz de covarianzas (aunque no sea correcto), la media de las cuasivarianzas muestrales es 196387 y los valores propios de la matriz de covarianzas valen: 981871.2, 43.14338, 13.68328, 6.629537 y 0.2351568 por lo que, con este criterio, tomaríamos p = 1.

5.5 Regla del codo o del gráfico de sedimentación

Es uno de los métodos más usados y suele ir incluido en casi todos los programas de estadística. El método consiste en representar j (eje x) frente a los valores propios estimados obteniéndose el denominado gráfico de sedimentación o desmoronamiento (scree graph). El gráfico será similar a la acumulación de sedimentos en la ladera de una montaña (cono de desmoronamiento). Se trataría de separar "la montaña" de los "sedimentos".

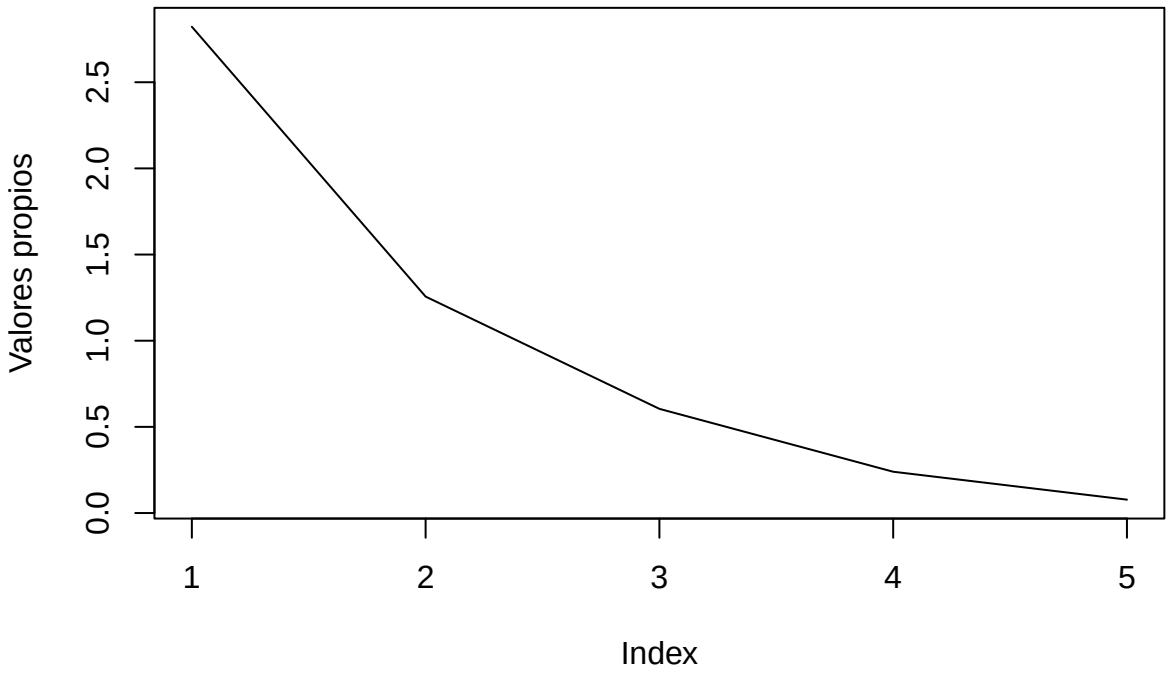
La regla establece que serán representativas las componentes hasta el primer "codo" (sin incluirlo) de la gráfica o hasta que comience la línea recta aproximada final (separando los sedimentos de la montaña). Para realizar este gráfico de forma automática en R haremos:

```
screeplot(PCA)
```



Se puede obtener un gráfico similar mediante:

```
plot(eigen(cor(d))$values, type = 'l', ylab = 'Valores propios')
```



En estos gráficos, aunque no está muy claro, parece que el codo (los sedimentos) se encuentra en j = 3, por lo que tomaríamos las dos primeras componentes (p = 2). Las soluciones p = 1 y p = 3 también serían aceptables. En otras ocasiones el "codo" aparece más claro y solo hay una opción (especialmente cuando k es más grande).