Modelos para crear mapas de posiconamiento

Jordi López Sintas

13 de enero de 2015

Introducción

Supongamos que una muestra de consumidores ha valorado su percepción acerca de p atributos o características de un conjunto de n marcas. Cuando el número de atributos o características es elevado, es difícil hacerse una idea de cuán parecidas o diferentes son las marcas según la percepción que de ellas tienen los consumidores. Por ello nos preguntamos, ¿es posible representar visualmente n marcas en un espacio de reducidas dimensiones, por ejemplo, k dimensiones, siendo $k < p$? ¿Y, además, que las marcas similares (según la percepción de los consumidores) se sitúen cerca unas de otras? Para ello disponemos de una familia de modelos conocida con el nombre de análisis o escalado multidimensional (MDS, muldimensional scaling). Esta familia de modelos, también, nos proporciona una medida de la discrepancia entre las percepciones observadas y las estimadas por el modelo, un índice de la medida en la que distancias observadas difieren de las estimadas.

Interpretar la estructura de los mercados

Cuando las medidas acerca de las percepciones se han recogido en una escala métrica o intervalo el análisis multidimensional utiliza el análisis de los componentes principales para su representación visual (Gower y Hand, 1996).

El análisis de los componentes principales es una técnica para la reducción de datos multivariantes. Su propósito es encontrar un conjunto de nuevas variables, igual en número a las variables originales, aunque con ciertas propiedades: 1) las nuevas variables, llamadas componentes principales, están incorrelacionadas entre ellas (son independientes en el espacio formado por los componente principales), 2) los componentes principales están formados a partir de las variables originales, 3) los componentes principales explican la varianza de las variables originales, de manera que el primer componente explica la mayor parte posible de variación, el segundo la mayor parte de la varianza residual, y así sucesivamente.

Durante el proceso de segmentación hemos utilizado los componentes principales para transformar las bases de segmentación en unas variables artificiales, componentes principales, incorrelacionadas entre ellas con el objeto de poder utilizar los modelos de segmentación tradicionales o para escoger un número reducido de bases de segmentación originales según su correlación con los componentes principales. En cambio cuando queremos estudiar la estructura de los mercados los componentes principales se utilzan para obtener una representación bidimensional, o tridimensional de la estructura de los mercados.

Para ello partiremos de una matriz de percepciones, $\mathbf{X}$, que contendrá tantas filas, n, como marcas y tantas columnas, p, como atributos o características describan a los productos. El subespacio de k dimensiones proporcionado por los componentes principales normalmente se obtiene de los datos estandarizados con el objeto de eliminar las diferencias de escala en la medida de las percepciones y a ayudarnos en la interpretación del la estructura de los mercados y en la valoración de la calidad de la representación visual. Por esa razón transformaremos la matriz de datos original, $\mathbf{X}$, sustrayéndole la media de cada una de las medidas y dividiremos las observaciones centradaspor su desviación típica, resultando una matriz de datos estandarizados.

La descomposición de la matriz de covarianzas, o correlaciones en la mayoría de los casos, $X'X$, nos proporciona los vectores y valores propios necesarios para construir e interpretar el espacio reducido, $X'X = A\Lambda A'$, donde A es una matriz ortogonal que proporciona los vectores propios, $A'A = I$. Además, los valores de la diagonal de la matriz Λ nos proporcionan la varianza explicada por cada uno de los componentes principales (Gower y Hand, 1996). Los vectores propios nos proporcionan un espacio de $k<p$ dimensiones alternativo al espacio de las características o atributos en los que se midieron las marcas que estamos analizando. Pero,

también, podemos describir el significado de los nuevos componentes o beneficios que estructuran el mercado (o, alternativamente, de los vectores propios) analizando la correlación de los atributos con los componentes o factores del nuevo espacio.

El mejor espacio reducido para representar al conjunto de marcas viene dado por los ejes formados por los k primeros componentes principales estimados. Buscamos una combinación lineal de las variables originales como la siguiente:

$$C_1 = a_{11}x1 + ... + a_{1p}xp$$
$$C_2 = a_{21}x2 + ... + a_{2p}xp$$

. . .

$$C_k = a_{k1}xk + ... + a_{kp}xp$$

Con ciertas propiedades:

1. Que la varianza de C_i sea tan grande como sea posible
2. Que los valores de $C_1, ..., C_p$ para todos los individuos de la muestra sean independientes, $S_{ij} = 0$, para todos y cada uno de los componentes
3. Para cada componente la suma del cuadrado de los coeficientes de su combinación lineal sea igual a la unidad, $a_{11}^2 + a_{12}^2 + ... + a_{1p}^2 = 1$.

De hecho la proyección de las marcas en el espacio reducido a k ejes principales, vendrá dada por predicción del conjunto inicial de medidas en el nuevo subespacio definido por los k<p primeros componentes principales $C = XA_k$, por ejemplo k=*2*.

$$\begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ \vdots & \vdots \\ c_{n1} & c_{n2} \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{pmatrix}$$

Así la predicción de la primera marca en el primer componente principal (primer elemento de la primera columna de la matriz C) vendrá dada por

$$C_{11} = a_{11}x11 + a_{21}x21 + ... + a_{p1}xp1$$

y así sucesivamente con el resto de los factores o beneficios que estructuran el mercado.

¿Cuál es la calidad de la representación visual? Nuestro propósito es encontrar un subespacio k que minimice la diferencia entre las percepciones medidas y las que visualizamos en el mapa de percepciones. Para conocer la calidad de la representación visual tendremos, primero, que estimar las percepciones que se derivan del modelo visual, $\hat{X} = C_{k=2}A'_{k=2}$,

$$\begin{pmatrix} \hat{x_{11}} & \hat{x_{12}} & \cdots & \hat{x_{1p}} \\ \hat{x_{21}} & \hat{x_{22}} & \cdots & \hat{x_{2p}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x_{n1}} & \hat{x_{n2}} & \cdots & \hat{x_{np}} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ \vdots & \vdots \\ c_{n1} & c_{n2} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{p1} \\ a_{12} & a_{22} & \cdots & a_{p2} \end{pmatrix}$$

y, después, verificar la diferencia con los datos originales, $X - \hat{X}$. La suma cuadrado de las diferencias entre lo observado y lo predicho por el modelo, $traza(X - \hat{X})'(X - \hat{X})$ resulta que es idéntica a la varianza no explicada por los componentes principales utilizados para formar el mapa de percepciones, $traza(\Lambda - \Lambda_k)$. Por ello, cuanto menor sea esa diferencia, mejor será la representación visual de las marcas en el espacio reducido mostrado en el mapa de percepciones. Si lo expresamos en relación a la variación total de la muestra,

la $traza(\Lambda)$, la calidad de la representación tomará valores entre cero y uno, de tal manera que cuanto más se acerque a cero, mejor será la representación.

También es de interés conocer la calidad de la representación las variables originales en los componentes principales. Una medida de esa calidad nos la ofrece la suma del cuadrado de los coeficientes de las funciones de los componentes principales, $A_k A_k' 1$, cuyo resultado es un vector columna con tantas filas como variables originales –donde 1 es el vector identidad. Si una variable está perfectamente representada, su valor del producto anterior será la unidad. Otra medida utilizada es el coeficiente de correlación entre las variables originales y los componentes principales, que cuando ambas variables están estandarizadas es igual al coeficiente de la función la función principal multiplicado por la desviación estándar del componente, $r_{ik} = a_{ik}\sqrt{\lambda_k} = l_{ik}$ (en la literatura anglosajona recibe el nombre de loadings), y su cuadrado nos da la medida en la que el componente k contribuye a explicar la varianza de la variable i. La calidad de representación visual de una variable original, entonces, será proporcional a la varianza total explicada por los componentes utilizados en la formación del mapa de percepciones, $h_i^2 = \sum_k^K r_{ik}^2 = \sum_k^K a_{ik}^2 \lambda_k$.

En ocasiones es necesario rotar los componentes sobre su propio eje con el objeto de facilitar la interpretación del significado de los componentes o beneficios que estructuran el mercado. En tales casos procedemos, primero, a normalizar los componentes, que entonces reciben el nombre de factores principales, $\hat{X} = C_k \sum_k m1 \sum_k A_k' = F_k L_k$, y $F_j = \frac{C_j}{\sqrt{Var(C_j)}}$. Así la relación entre las variables originales y los, ahora, factores principales, vendrá dada por $\hat{X} = F_k L_k$ (Afifi y Azen, 1979).

$$\begin{pmatrix} \hat{x_{11}} & \hat{x_{12}} & \cdots & \hat{x_{1p}} \\ \hat{x_{21}} & \hat{x_{22}} & \cdots & \hat{x_{2p}} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x_{n1}} & \hat{x_{n2}} & \cdots & \hat{x_{np}} \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \\ \vdots & \vdots \\ f_{n1} & f_{n2} \end{pmatrix} = \begin{pmatrix} l_{11} & l_{21} & \cdots & l_{p1} \\ l_{12} & l_{22} & \cdots & l_{p2} \end{pmatrix}$$

Donde $l_{ik} = a_{ik}\sqrt{\lambda_k}$ y a la matriz L_k se la conoce con el nombre de *matriz de correlaciones* (*pattern matrix*), a las cuales también se las denomina cargas (*loadings*). Ahora la rotación de factores principales (la puntuación de las marcas en ellos) no altera su propiedad de indepencia entre los factores principales y corresponden a rotaciones ortogonales de las correlaciones (Venables y Ripley, 2000: 47-48). Una vez realizada la rotación, la nueva matriz de correlaciones , L^0, recibe el nombre de *matriz de la estructura* de los factores principales y trata de mostrar una estructura simple, fácilmente interpretable. Si la matriz de correlaciones no se ha rotado, coincide con la matriz de estructura de los factores.

Veamos cómo representar visualmente las percepciones que una muestra de consumidores mostraron acerca de un conjunto de marcas de coches.

```
g20per<-read.table("g20per.txt", header=TRUE)
names(g20per)
```

```
##  [1] "G20"     "FORD"    "AUDI"    "TOYOTA"  "EAGLE"   "HONDA"   "SAAB"
##  [8] "PONTIAC" "BMW"     "MERCURY"
```

Para hacer el análisis tenemos que transponer la tabla de datos que acabamos de leer, de tal manera que las filas ahora serán las marcas, y las columnas, los atributos.

```
g20<-data.frame(t(g20per))
names(g20)
```

```
##  [1] "Atract"     "Silec"      "Nofiab"     "MalaCons"   "Interesan"
##  [6] "Deport"     "Incomod"    "Espacioso"  "FacilMan"   "Prestigio"
## [11] "Ordinar"    "Economico"  "Existos"    "Vanguard"   "Malacompra"
```

```r
options(digits=1)
t(head(cor(g20)))
```

```
##            Atract Silec Nofiab MalaCons Interesan Deport
## Atract       1.00  0.79  -0.87    -0.95      -0.6  -0.25
## Silec        0.79  1.00  -0.52    -0.87      -0.7  -0.63
## Nofiab      -0.87 -0.52   1.00     0.84       0.4  -0.09
## MalaCons    -0.95 -0.87   0.84     1.00       0.6   0.39
## Interesan   -0.64 -0.70   0.37     0.60       1.0   0.54
## Deport      -0.25 -0.63  -0.09     0.39       0.5   1.00
## Incomod     -0.57 -0.76   0.21     0.64       0.5   0.67
## Espacioso    0.36  0.53   0.03    -0.41      -0.5  -0.74
## FacilMan    -0.17 -0.31  -0.06     0.29       0.4   0.74
## Prestigio    0.87  0.75  -0.76    -0.92      -0.5  -0.46
## Ordinar     -0.68 -0.42   0.46     0.60       0.5   0.32
## Economico   -0.06  0.09   0.25     0.16      -0.6  -0.22
## Existos      0.93  0.74  -0.84    -0.90      -0.5  -0.24
## Vanguard     0.18 -0.15  -0.18    -0.03      -0.1   0.48
## Malacompra  -0.67 -0.54   0.43     0.49       0.8   0.08
```

```r
cor(g20)
```

```
##            Atract Silec Nofiab MalaCons Interesan Deport Incomod Espacioso
## Atract       1.00  0.79  -0.87    -0.95      -0.6  -0.25  -0.570      0.36
## Silec        0.79  1.00  -0.52    -0.87      -0.7  -0.63  -0.755      0.53
## Nofiab      -0.87 -0.52   1.00     0.84       0.4  -0.09   0.214      0.03
## MalaCons    -0.95 -0.87   0.84     1.00       0.6   0.39   0.639     -0.41
## Interesan   -0.64 -0.70   0.37     0.60       1.0   0.54   0.542     -0.48
## Deport      -0.25 -0.63  -0.09     0.39       0.5   1.00   0.671     -0.74
## Incomod     -0.57 -0.76   0.21     0.64       0.5   0.67   1.000     -0.90
## Espacioso    0.36  0.53   0.03    -0.41      -0.5  -0.74  -0.904      1.00
## FacilMan    -0.17 -0.31  -0.06     0.29       0.4   0.74   0.720     -0.85
## Prestigio    0.87  0.75  -0.76    -0.92      -0.5  -0.46  -0.711      0.57
## Ordinar     -0.68 -0.42   0.46     0.60       0.5   0.32   0.640     -0.72
## Economico   -0.06  0.09   0.25     0.16      -0.6  -0.22   0.003      0.07
## Existos      0.93  0.74  -0.84    -0.90      -0.5  -0.24  -0.602      0.39
## Vanguard     0.18 -0.15  -0.18    -0.03      -0.1   0.48   0.114     -0.11
## Malacompra  -0.67 -0.54   0.43     0.49       0.8   0.08   0.342     -0.22
##            FacilMan Prestigio Ordinar Economico Existos Vanguard
## Atract        -0.17       0.9    -0.7    -0.055    0.93    0.177
## Silec         -0.31       0.7    -0.4     0.086    0.74   -0.151
## Nofiab        -0.06      -0.8     0.5     0.252   -0.84   -0.185
## MalaCons       0.29      -0.9     0.6     0.164   -0.90   -0.033
## Interesan      0.36      -0.5     0.5    -0.645   -0.52   -0.131
## Deport         0.74      -0.5     0.3    -0.217   -0.24    0.475
## Incomod        0.72      -0.7     0.6     0.003   -0.60    0.114
## Espacioso     -0.85       0.6    -0.7     0.070    0.39   -0.105
## FacilMan       1.00      -0.5     0.5     0.022   -0.22    0.176
## Prestigio     -0.51       1.0    -0.7    -0.240    0.93   -0.113
## Ordinar        0.52      -0.7     1.0     0.099   -0.63   -0.258
## Economico      0.02      -0.2     0.1     1.000   -0.12   -0.007
## Existos       -0.22       0.9    -0.6    -0.121    1.00   -0.058
```

```
## Vanguard       0.18      -0.1    -0.3    -0.007   -0.06    1.000
## Malacompra    -0.08      -0.4     0.4    -0.589   -0.58   -0.340
##            Malacompra
## Atract          -0.67
## Silec           -0.54
## Nofiab           0.43
## MalaCons         0.49
## Interesan        0.79
## Deport           0.08
## Incomod          0.34
## Espacioso       -0.22
## FacilMan        -0.08
## Prestigio       -0.37
## Ordinar          0.41
## Economico       -0.59
## Existos         -0.58
## Vanguard        -0.34
## Malacompra       1.00
```

Podemos hacer la descomposición de la matrix de correlaciones de la siguiente manera:

```
#centrar y dividir por la desviación estándar
X<-scale(g20)
##matriz de correlaciones, X'X
cor=X%*%t(X)
cor
```

```
##         G20 FORD AUDI TOYOTA EAGLE HONDA  SAAB PONTIAC BMW MERCURY
## G20       9   -7    3    3.3  -7.3   3.3   4.2    -8.7   7      -8
## FORD     -7   12   -1   -4.6   6.2  -4.9  -9.4    11.9  -9       6
## AUDI      3   -1   15   -6.9  -6.1  -3.9   6.9    -5.1   5      -7
## TOYOTA    3   -5   -7   15.4  -6.5   0.4  -0.6     0.7   2      -3
## EAGLE    -7    6   -6   -6.5  12.9  -0.8  -6.6     7.1 -10      11
## HONDA     3   -5   -4    0.4  -0.8   9.9   0.9    -7.1   4      -2
## SAAB      4   -9    7   -0.6  -6.6   0.9  18.1   -13.5   7      -7
## PONTIAC  -9   12   -5    0.7   7.1  -7.1 -13.5    18.3 -12       8
## BMW       7   -9    5    2.4 -10.0   4.4   6.9   -11.8  14      -9
## MERCURY  -8    6   -7   -3.5  11.1  -2.3  -6.8     8.3  -9      11
```

```
#Calcular los componentes principales P o t(P)
E=eigen(cor,TRUE)
A<-E$vectors
I=A%*%t(A)
var<-E$values
#La puntuación en los componentes o factors principales
C = A %*% X
options(digits=6)
#La desviación estandar de cada atributo de la matriz rotada
sdev = sqrt(diag((1/(dim(X)[2]-1)* A %*% cor %*% t(A))))
sdev
```

```
##  [1] 0.461311 0.979078 0.711939 0.659651 1.191503 1.379996 0.843901
##  [8] 1.340482 1.198730 0.509292
```

```
#######
```

No obstante podemos obtener los componentes principales utilizando las funciones `prcomp()` y `princomp()` de R. la equivalencia de resultados es la siguientes.

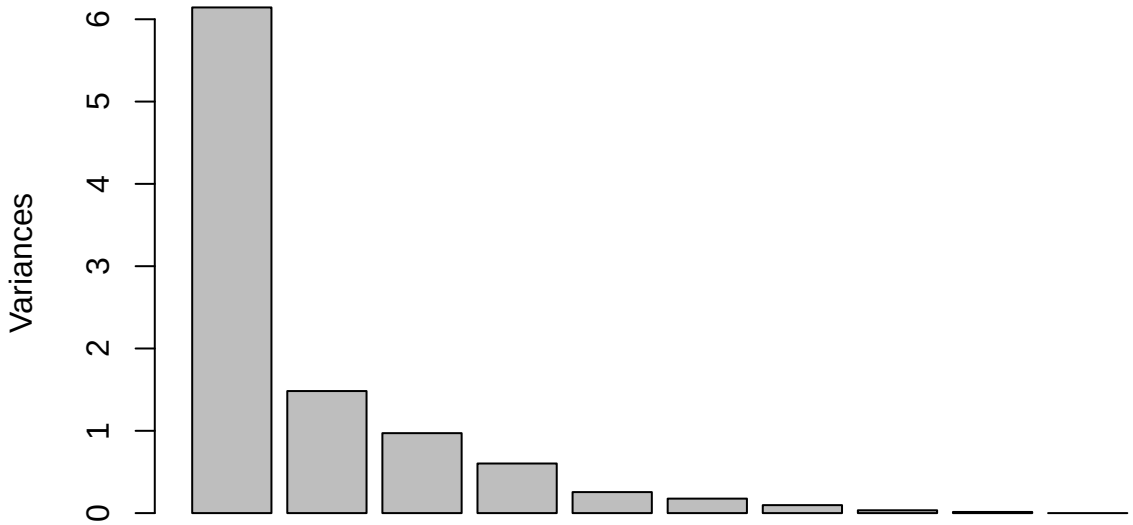
prcomp()	princomop()	Manualmente	Interpretación
sdev	sdev	sdev	Desviaciones estandr de cada columna de la matriz rotada
rotation	loadings	A	Los coeficients de los componentes principales
center	center	scale()	Dactos centrados en la media
scale	scale	scale()	Datos centrados y divididos por la desviación standar
x	scores	newdata	La puntuación en los componentes

Calculamos los componentes principales de la matriz de datos g20 con la función `prcomp`. El resultado queda alamacenado en el objeto `g20.pca`. La información está estructurada en diferentes componentes, `sdev` (las desviaciones estándares de los componentes principales), `rotation` (loadings), `center`, `scale` (la escala y centros utilizados), y `x` ($\hat{X}$) si hemos pedido que calcule la puntuación de las marcas en los components principales (retx=TRUE).

```
g20.pca<-prcomp(g20, cor=T, retx=TRUE)
g20.puntos<-g20.pca$x
summary(g20.pca)
```

```
## Importance of components:
##                           PC1   PC2    PC3    PC4    PC5   PC6     PC7
## Standard deviation     2.479 1.218 0.9855 0.7763 0.5053 0.419 0.31145
## Proportion of Variance 0.628 0.152 0.0993 0.0616 0.0261 0.018 0.00992
## Cumulative Proportion  0.628 0.780 0.8794 0.9410 0.9672 0.985 0.99507
##                           PC8     PC9     PC10
## Standard deviation     0.18515 0.11790 5.07e-16
## Proportion of Variance 0.00351 0.00142 0.00e+00
## Cumulative Proportion  0.99858 1.00000 1.00e+00
```

```
plot(g20.pca)
```



Como dijimos anteriormente, cada componente principal explica la mayor parte posible de la variación residual. Así vemos que el primer componentes explica un 63%, el segundo un 15%, el tercero un 10 %, el cuarto un 6% y los sucesivos componentes van reduciendo su contribución a la variación de la muestra.

Podemos saber cómo se forman los componentes principales a partir de las variables originales examinando las cargas o rotaciones. Estos son los vectores singulares que multiplican las variables originales para producir los components principales.

```
# pc loadings
head(g20.pca$rotation)
```

```
##                PC1       PC2         PC3        PC4        PC5        PC6
## Atract    -0.291889  0.194684 -0.10726897 -0.0914134  0.0736472  0.0146376
## Silec     -0.397015 -0.123137 -0.22158169  0.4664042  0.4493927 -0.0580889
## Nofiab     0.231296 -0.436302  0.01711390  0.0283078  0.2068075 -0.1867461
## MalaCons   0.430711 -0.203274  0.00335342 -0.1636509 -0.1028254 -0.3081835
## Interesan  0.196025  0.161733  0.43305407  0.1125605  0.1645701 -0.3086490
## Deport     0.164648  0.462993 -0.04106408 -0.2905780  0.2311660 -0.3083159
##                 PC7        PC8        PC9      PC10
## Atract    -0.17552204 -0.1043880  0.4922212  0.560310
## Silec     -0.12098389  0.0265838 -0.3979857  0.127730
## Nofiab    -0.08615341 -0.3127262 -0.0532749  0.415954
## MalaCons  -0.00751391 -0.2104999 -0.2249212  0.134376
## Interesan -0.17633906 -0.0834722  0.3076260 -0.205296
## Deport     0.35558389  0.3245623 -0.3133913  0.355113
```

```
tail(g20.pca$rotation)
```

```
##                  PC1        PC2         PC3        PC4        PC5
## Prestigio  -0.45725890  0.0981032  0.28267264 -0.0187148 -0.3802080
## Ordinar     0.21604211  0.0595714 -0.16582852  0.5781448 -0.0130735
## Economico   0.01213193 -0.2334333 -0.59086516 -0.1237918 -0.3521721
## Existos    -0.28404847  0.1823498  0.00461998 -0.0316698 -0.1616456
## Vanguard    0.00529025  0.1215812 -0.10439675 -0.3870939  0.4448913
```

```
## Malacompra  0.13161869 -0.0318702  0.43145462  0.2418743 -0.0810435
##                   PC6       PC7       PC8        PC9       PC10
## Prestigio  -0.0808296 0.1935189 -0.260787 -0.2838078  0.1209051
## Ordinar    -0.0569748 0.5829864 -0.249168  0.1298428 -0.0841885
## Economico  -0.0952982 0.0596020  0.300127  0.0569406 -0.0870917
## Existos    -0.4840675 0.0426210 -0.197524 -0.0772476 -0.2466278
## Vanguard    0.4268468 0.2437228 -0.350912 -0.0890720 -0.2684085
## Malacompra  0.2952736 0.0252671  0.458262 -0.2104587  0.1123806
```

Estos coeficientes que multiplican a las variables origianles para producir los components principales nos indica la contribución de cada variable original (estandarizada) a la construcción de los components principales. Así el primer componente es una combinación lineal de asesinatos, asaltos y violaciones (PC1=-0,29Atrac-0,39Silen+...+ 0,29Malacompra), mientras que para el resto sólo tenemos que leer la matriz de datos verticalmente.

Puntuación en las marcas en los componentes principales que estructuran el mercado (presentamos las seis primeras ciudades y las 6 últimas de la base de datos), la correlación entre los componentes principales contruidos (que debería ser cero) y la correlación de las variables originales con los factores

```
head(g20.pca$x[, c(1,2)])
```

```
##              PC1       PC2
## G20    -2.522887  0.247712
## FORD    2.426923 -0.312052
## AUDI   -1.565987 -1.613781
## TOYOTA -1.047514  2.575236
## EAGLE   2.890005 -0.764345
## HONDA  -0.985592  0.269983
```

```
tail(g20.pca$x[, c(1,2)])
```

```
##               PC1       PC2
## EAGLE    2.890005 -0.764345
## HONDA   -0.985592  0.269983
## SAAB    -2.099931 -1.494307
## PONTIAC  3.122505  0.902919
## BMW     -2.935520  0.431988
## MERCURY  2.717998 -0.243354
```

```
cor(g20.pca$x[, c(1,2)])
```

```
##             PC1         PC2
## PC1 1.00000e+00 2.18172e-16
## PC2 2.18172e-16 1.00000e+00
```

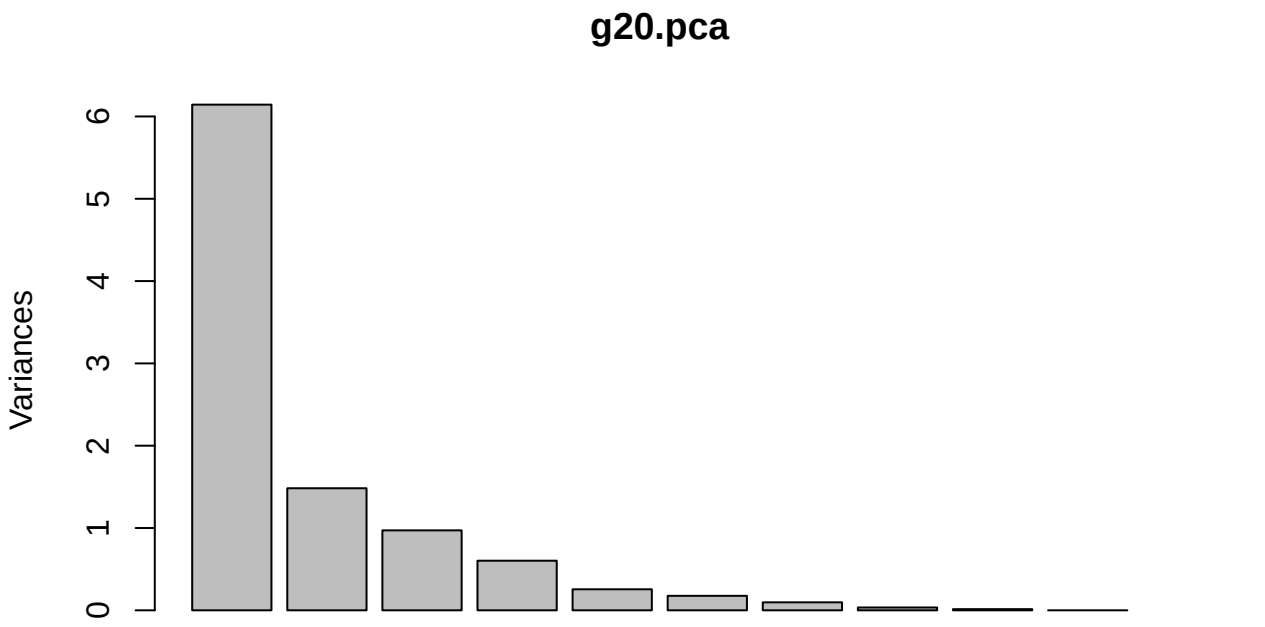
```
cor(scale(g20), g20.pca$x[, c(1,2)])
```

```
##                   PC1        PC2
## Atract     -0.9309176  0.3050462
## Silec      -0.8889402 -0.1354558
## Nofiab      0.7206381 -0.6678491
```

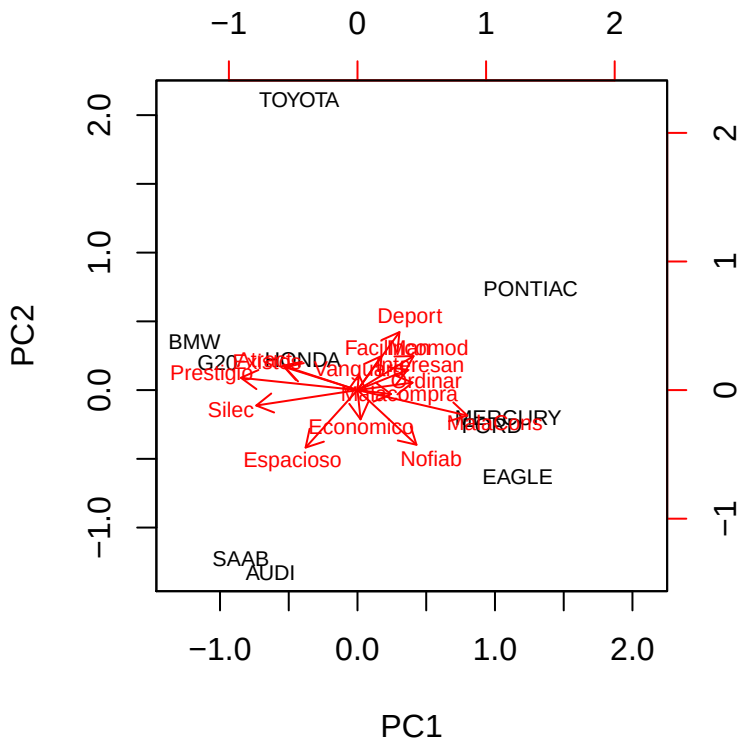
```
## MalaCons    0.9591839 -0.2224028
## Interesan   0.6920673  0.2805296
## Deport      0.5337704  0.7374219
## Incomod     0.7873233  0.4768687
## Espacioso  -0.6205666 -0.6872268
## FacilMan    0.4632175  0.6582476
## Prestigio  -0.9505139  0.1001895
## Ordinar     0.7180750  0.0972773
## Economico   0.0440136 -0.4160663
## Existos    -0.9126528  0.2878466
## Vanguard    0.0282356  0.3188085
## Malacompra  0.5523141 -0.0657047
```

Podemos observarlo si graficamos la contribución de los componentes a la variación de los datos originales estandarizados y situamos los datos originales en el plano de los componentes principales.

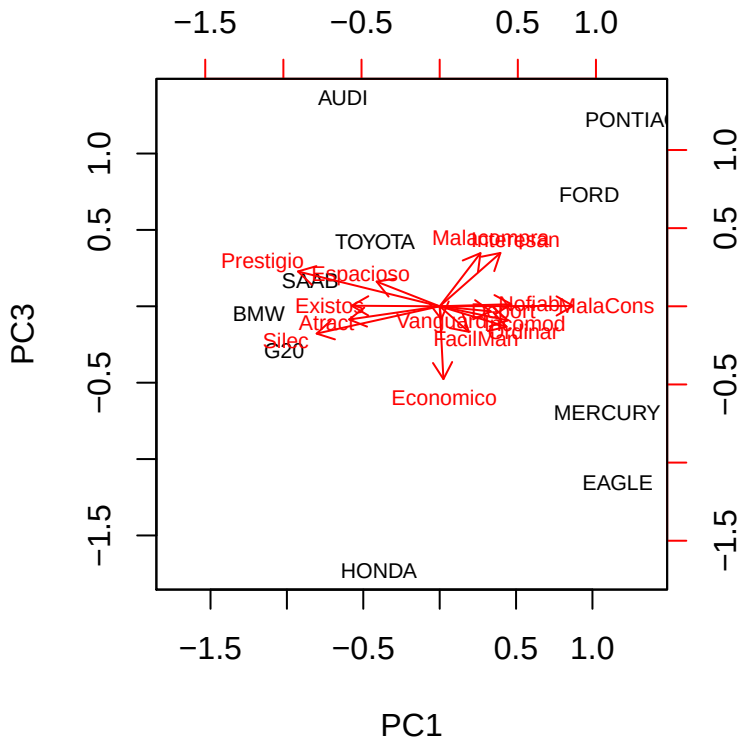
```
# scree plot
plot(g20.pca)
```



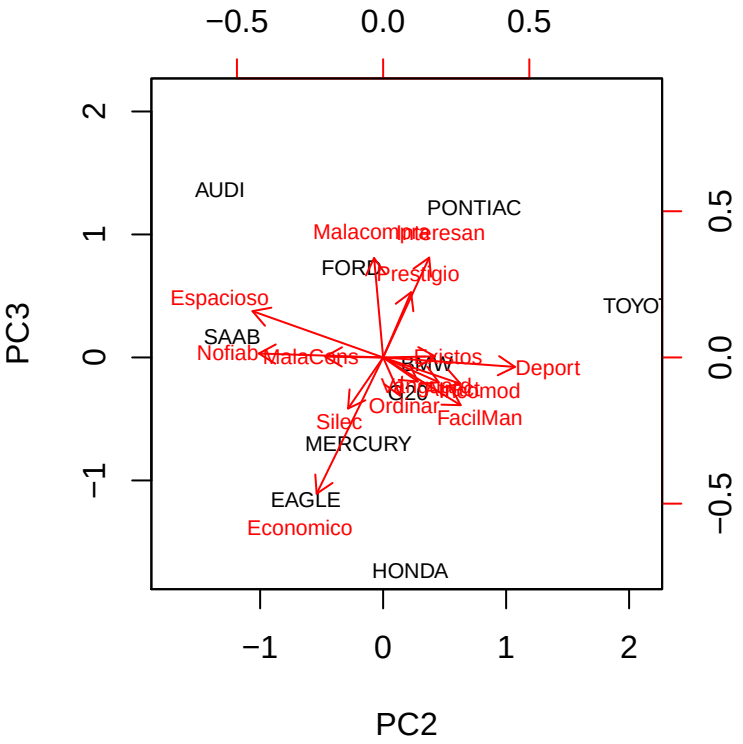

```
biplot(g20.pca, pc.biplot=T, cex=0.7, ex=0.8)
```




```
biplot(g20.pca, choices=c(1,3),pc.biplot=T, cex=0.7, ex=0.8)
```




```
biplot(g20.pca, choices=c(2,3),pc.biplot=T, cex=0.7, ex=0.8)
```



La representación simultánea de la puntuación de las marcas en el nuevo espacio reducido formado por los componentes principales y las variables originales se realiza por medio de la función `biplot` del paquete `MASS`. Las variables originales se representan por medio de unos vectores cuya longitud viene dada por su correlación con los factores principales, mayor cuanto mayor es la correlación. Nótese, no obstante, que la correlación de cada una de los atributos de las marcas con los factores y la puntuación de las marcas en ellos forman dos grupos de datos medidos en diferentes escalas. Esa es la razón por la que en los gráficos aparecen dos escalas, una para cada grupo de datos. No obstante es habitual multiplicar las correlaciones por algún factor de escala como la varianza de los factores principales [insertar referencia a MDPREF]. De esa manera la proyección perpendicular de las marcas sobre los vectores de los atributos nos ofrece una medida relativa de la cantidad de ese atributo que los consumidores han percibido en la marca.

Mapas conjuntos: De perceptiones y preferencias

Análisis interno

Si queremos introducir las preferencias en el mapa y construir un mapa conjunto, de percepciones y preferencias, entonces necesitamos añadir las preferencias de los consumidores. Para ello debemos definir el modelo de preferencias, vectoriales o ideales. En el primer caso pedimos a los consumidores que nos digan su preferencias por las marcas, segmentamos la matriz de preferencias y el resultado lo incorporamos a la matriz de de percepciones como nuevos atributos que muestran las preferencias por las marcas. En el caso de utilizar un modelo ideal de preferencias, pediríamos a los consumidores que nos mostraran sus preferencias por cada uno de los atributos de forman las marcas, segmentaríamos e introduciríamos el resultado en la matriz de percepciones como nuevas marcas.

En el caso de las preferencias vectoriales, primero accedemos a la matriz de preferencias y analizamos la heterogeneidad de las preferencias de los consumidores.

```
## Loading required package: XLConnect
## Loading required package: XLConnectJars
```

```
## XLConnect 0.2-9 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI, Apache Commons
##     Codec, XML Commons External Components XML APIs),
##   Stephen Colebourne [ctb, cph] (Joda-Time Java library),
##   Metastuff, Ltd. [ctb, cph] (dom4j)
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com
```

```
g20 <- loadWorkbook("g20_completo.xlsx")
g20.prefs <- readWorksheet(g20, rownames=1, sheet = "prefs", header = TRUE)
head(g20.prefs)
```

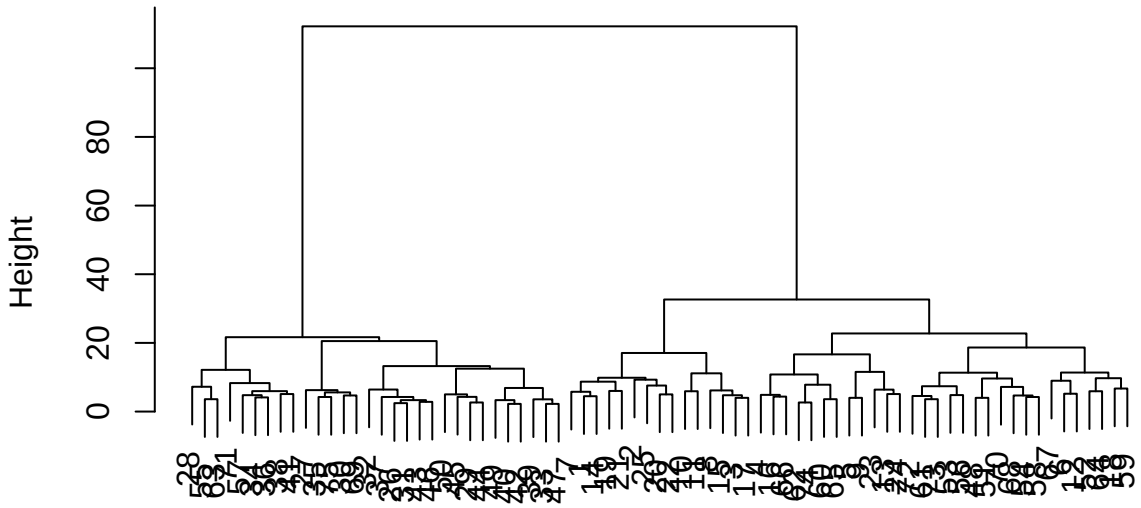
```
##   G20 FORD AUDI TOYOTA EAGLE HONDA SAAB PONTIAC BMW MERCURY
## 1   4    7    8      3     4     5    5       1   4       5
## 2   4    8    6      5     8     7    3       1   5       2
## 3   8    5    9      4     1     7    7       2   4       4
## 4   7    1    8      1     4     6    5       5   7       3
## 5   9    8    8      3     5     4    3       2   8       6
## 6   5    6    5      5     2     4    8       4   4       7
```

Para analizar la heterogeneidad en las preferencias, realizamos un análisis de cluster en fos fases: 1) utilizamos un modelo de la familia de los procedimientos de clasificación jerárquica y, después, 2) utilizamos un modelo de la familia de los procedimientos de partición de una muestra.

```
g20.prefs.hclust<-hclust(dist(g20.prefs, method="euclidean"), method="ward")
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
#Mostramos el resultado de la agrupación
plot(g20.prefs.hclust)
```



dist(g20.prefs, method = "euclidean")
hclust (*, "ward.D")

```
g20.prefs.hclust.centros<-tapply(as.matrix(g20.prefs), list(rep(cutree(g20.prefs.hclust, 3), ncol(as.ma
#Visualizamos el resultado
t(g20.prefs.hclust.centros)
```

```
##          1       2       3
## 1  5.60000 6.56667 8.13333
## 2  6.80000 4.16667 2.10000
## 3  7.80000 6.40000 3.63333
## 4  3.33333 5.10000 7.83333
## 5  3.06667 3.40000 5.90000
## 6  6.06667 5.80000 8.10000
## 7  5.86667 6.30000 8.16667
## 8  2.00000 3.30000 4.70000
## 9  4.33333 7.13333 7.16667
## 10 5.00000 4.06667 3.16667
```

```
g20.prefs.kmeans3<-kmeans(g20.prefs, g20.prefs.hclust.centros)
names(g20.prefs.kmeans3)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
t(g20.prefs.kmeans3$centers)
```

```
##               1    2       3
```

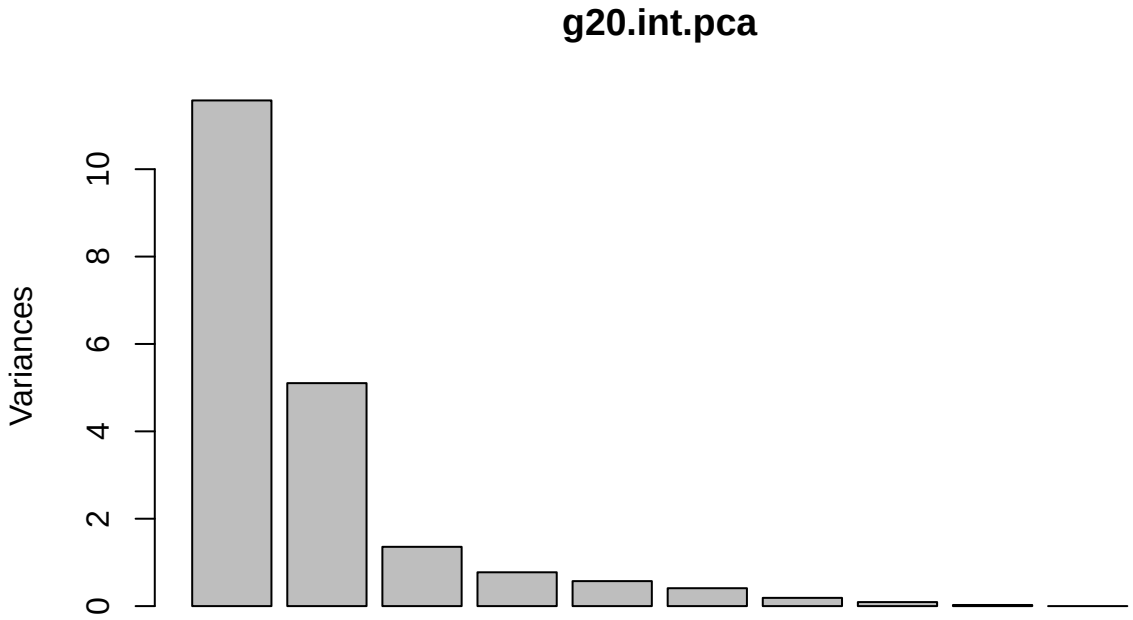
```
## G20     5.72222 6.40 8.18750
## FORD    6.94444 3.84 2.15625
## AUDI    7.61111 6.44 3.75000
## TOYOTA  3.50000 5.00 7.81250
## EAGLE   3.05556 3.56 5.65625
## HONDA   5.72222 5.84 8.09375
## SAAB    5.88889 6.16 8.18750
## PONTIAC 2.27778 3.12 4.71875
## BMW     4.61111 7.20 7.21875
## MERCURY 4.94444 4.12 3.12500
```

Después introducimos el resultado de la segmentación en la base de datos de las percepciones. Para ello añadir tres filas (correspondientes a los tres segmentos obtenidos) mediante la función `rbind`.

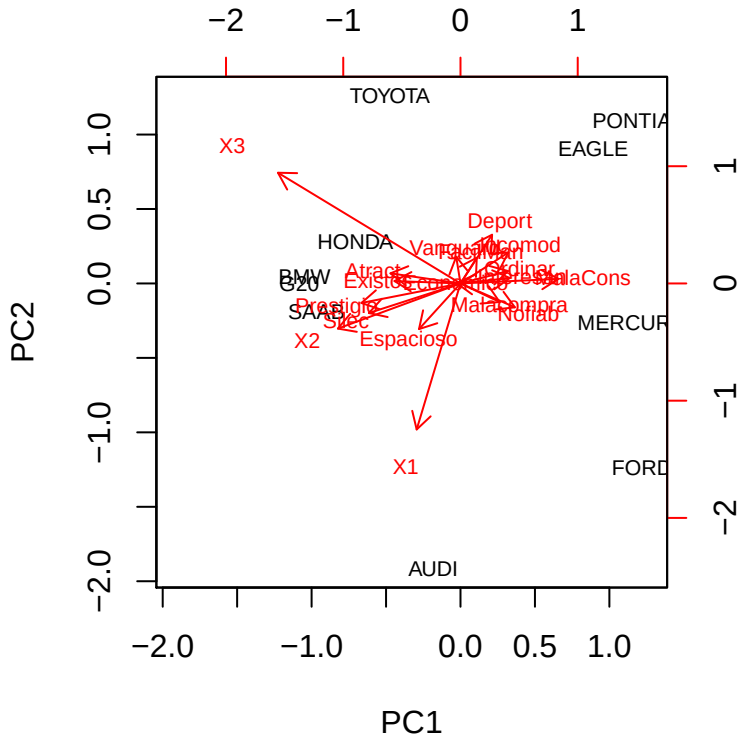
```
g20.int.seg<-rbind(g20per, g20.prefs.kmeans3$centers)
tail(g20.int.seg)
```

```
##                G20    FORD    AUDI TOYOTA   EAGLE   HONDA    SAAB PONTIAC
## Existos    5.30000 4.20000 5.00000 5.5000 3.70000 5.60000 5.30000 4.40000
## Vanguard   4.30000 3.60000 3.60000 4.9000 4.40000 3.90000 4.70000 4.10000
## Malacompra 3.40000 4.30000 4.30000 3.5000 3.60000 2.60000 2.90000 4.30000
## 1          5.72222 6.94444 7.61111 3.5000 3.05556 5.72222 5.88889 2.27778
## 2          6.40000 3.84000 6.44000 5.0000 3.56000 5.84000 6.16000 3.12000
## 3          8.18750 2.15625 3.75000 7.8125 5.65625 8.09375 8.18750 4.71875
##                BMW MERCURY
## Existos    5.90000 3.90000
## Vanguard   3.70000 4.50000
## Malacompra 3.30000 3.80000
## 1          4.61111 4.94444
## 2          7.20000 4.12000
## 3          7.21875 3.12500
```

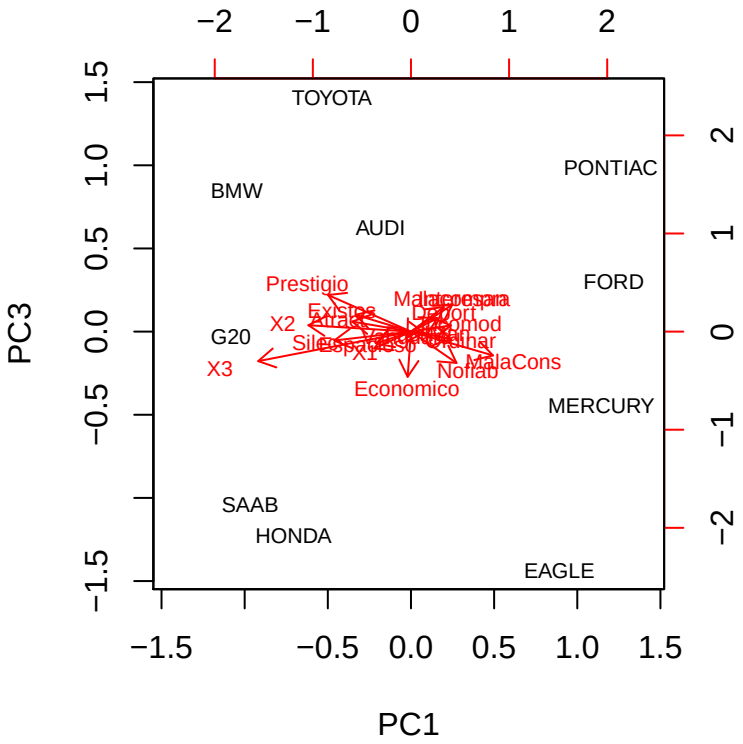
```
g20.int.seg.t<-data.frame(t(g20.int.seg))
g20.int.pca <- prcomp(g20.int.seg.t, cor=TRUE)

plot(g20.int.pca)
```



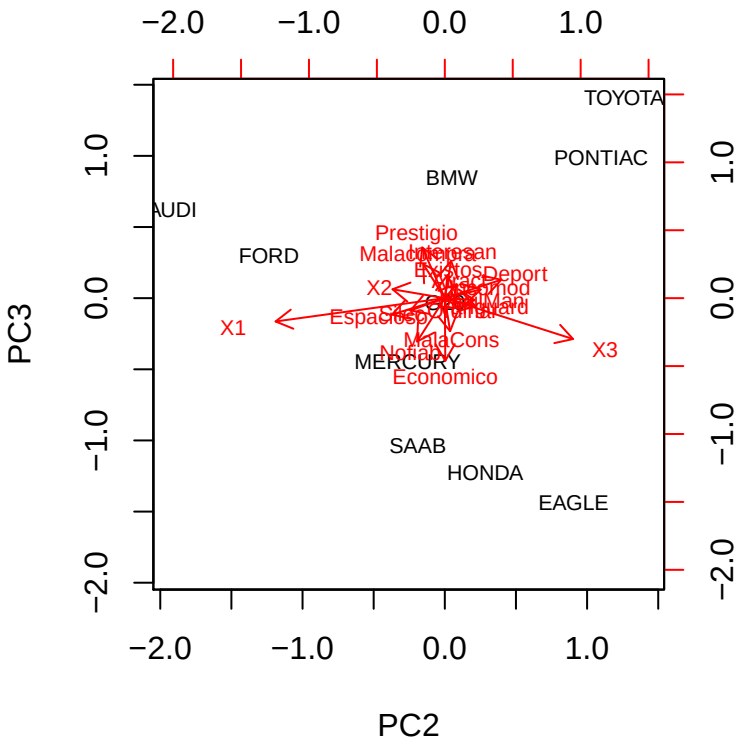

```
biplot(g20.int.pca , pc.biplot=T, cex=0.7, ex=0.8)
```




```
biplot(g20.int.pca , choices=c(1,3), pc.biplot=T, cex=0.7, ex=0.8)
```




```
biplot(g20.int.pca , choices=c(2,3), pc.biplot=T, cex=0.7, ex=0.8)
```

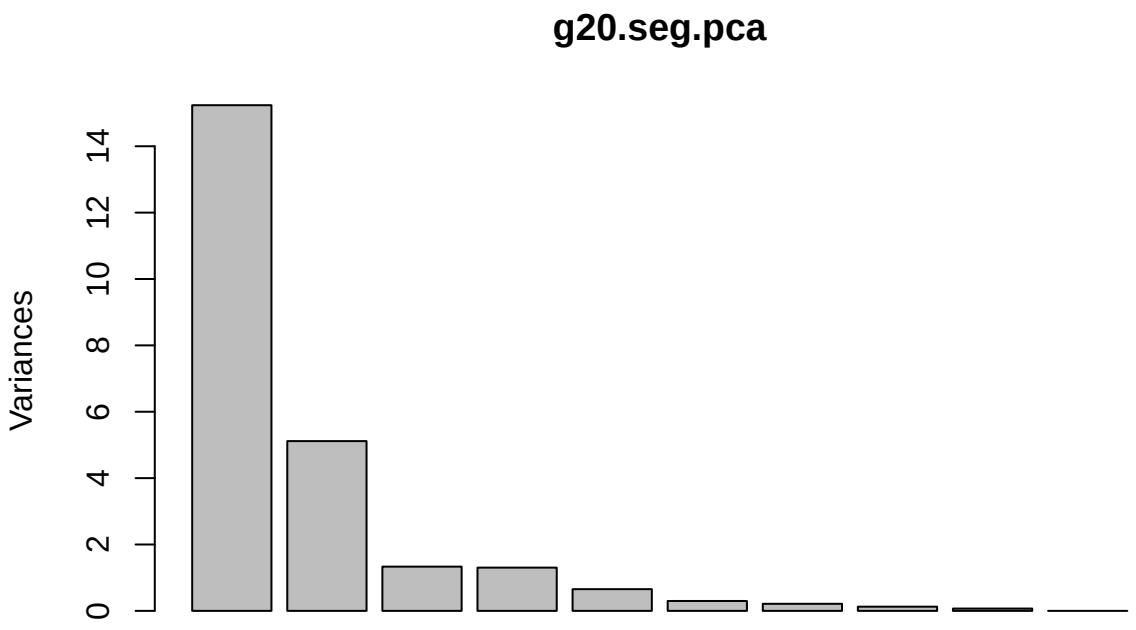


Alternativamente podemos leer el fichero de datos "g20seg" que ya incorpora el resultado de la segmentación de las preferencias.

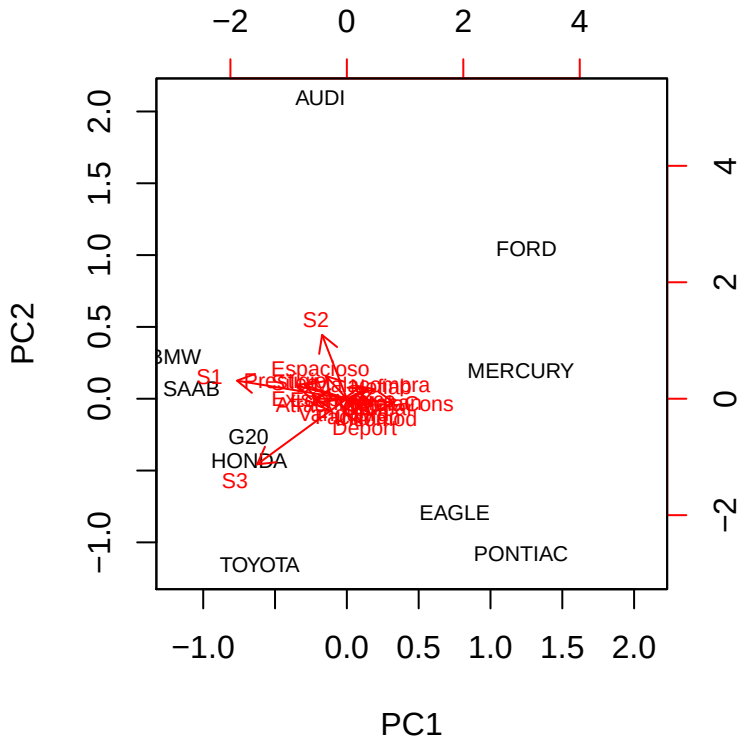
```
g20seg<-read.table("g20seg.txt", header=T)
tail(g20seg)
```

```
##            G20 FORD AUDI TOYOTA EAGLE HONDA SAAB PONTIAC BMW MERCURY
## Existos    5.3  4.2  5.0    5.5   3.7   5.6  5.3     4.4 5.9     3.9
## Vanguard   4.3  3.6  3.6    4.9   4.4   3.9  4.7     4.1 3.7     4.5
## Malacompra 3.4  4.3  4.3    3.5   3.6   2.6  2.9     4.3 3.3     3.8
## S1         4.3  2.1  6.0    6.1   3.3   6.0  7.5     1.2 8.3     1.7
## S2         5.9  6.0  7.7    3.5   3.1   5.5  5.4     2.5 5.4     4.9
## S3         8.6  2.1  3.4    8.1   5.8   8.3  8.4     5.3 7.3     3.4
```

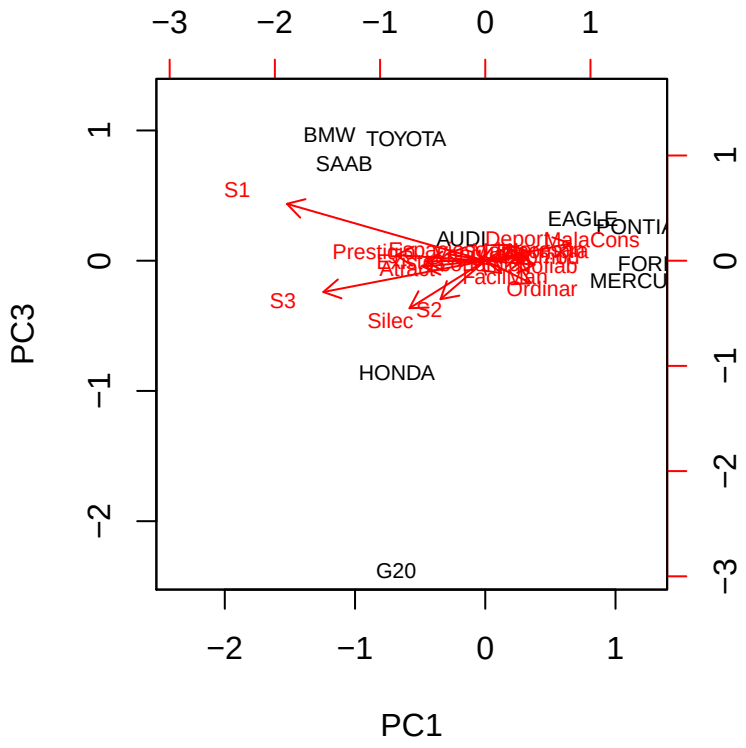
```
g20.seg<-data.frame(t(g20seg))
g20.seg.pca <- prcomp(g20.seg, cor=TRUE)

plot(g20.seg.pca)
```



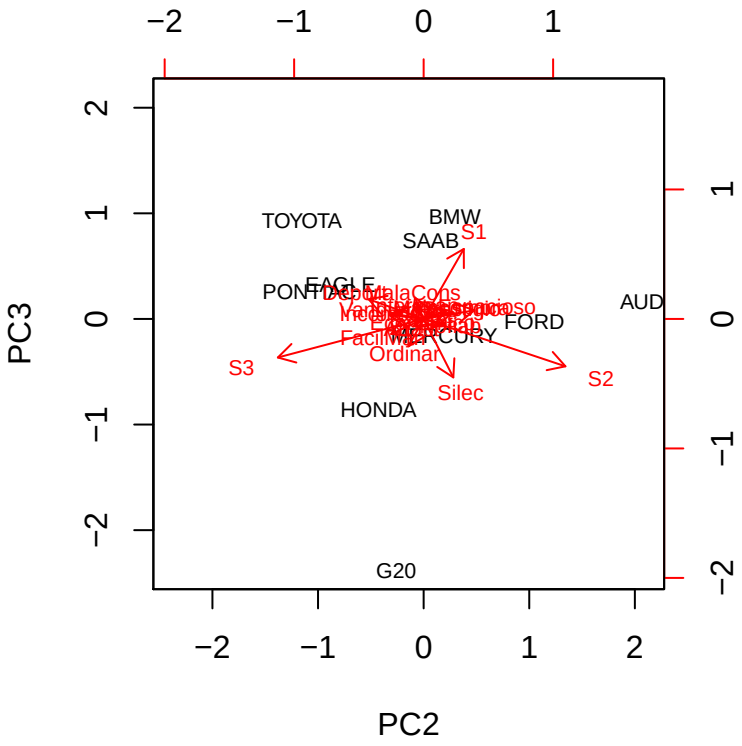

```
biplot(g20.seg.pca , pc.biplot=T, cex=0.7, ex=0.8)
```




```
biplot(g20.seg.pca , choices=c(1,3), pc.biplot=T, cex=0.7, ex=0.8)
```




```
biplot(g20.seg.pca , choices=c(2,3), pc.biplot=T, cex=0.7, ex=0.8)
```



El resultado no debería diferir.

Análisis externo

Cuando hacemos un análisis externo de las preferencias, en lugar de construir simultáneamente un mapa de percepcines y preferencias, lo que hacemos es construir un mapa de de percepciones y sobre él superponer los datos sobre preferencias, correlacionando la nueva base de datos con la estructura del espacio formado por los componentes principales.

```
g20 <- loadWorkbook("g20_completo.xlsx")
#leemos la hoja correspondiente a las percepciones
mydata <- readWorksheet(g20, rownames=1, sheet = "per", header = TRUE)
head(mydata)
```

```
##             G20 FORD AUDI TOYOTA EAGLE HONDA SAAB PONTIAC BMW MERCURY
## Attractive  5.6  4.0  4.6    5.6   4.0   5.2  5.3     3.9 5.7     3.9
## Quiet       6.3  3.6  5.2    4.2   3.5   5.4  4.8     2.8 5.0     3.3
## Unreliable  2.9  4.2  3.7    2.0   4.3   3.2  3.7     3.9 2.3     4.0
## Poorl_Built 1.6  4.2  2.6    2.1   4.3   2.8  2.8     4.4 1.8     4.3
## Interesting 3.6  5.0  4.0    4.3   3.9   3.4  3.4     5.4 3.3     3.9
## Sporty      4.1  4.9  3.8    6.2   4.9   5.1  4.3     5.7 4.1     5.2
```

```
g20.per = t(mydata) #transponemos los datos para tener los atributos en las columnas.
head(g20.per) #Vemos las seis primeras observaciones
```

```
##        Attractive Quiet Unreliable Poorl_Built Interesting Sporty
## G20           5.6   6.3        2.9         1.6         3.6    4.1
## FORD          4.0   3.6        4.2         4.2         5.0    4.9
## AUDI          4.6   5.2        3.7         2.6         4.0    3.8
## TOYOTA        5.6   4.2        2.0         2.1         4.3    6.2
```

```
## EAGLE         4.0   3.5        4.3         4.3         3.9    4.9
## HONDA         5.2   5.4        3.2         2.8         3.4    5.1
##        Unconfortable Roomy EasyService Hi_prestige Common Economical
## G20              3.2   4.2         4.6         5.4    3.5        3.6
## FORD             4.0   3.9         4.9         3.5    3.6        3.7
## AUDI             2.4   5.3         3.5         5.6    3.4        3.6
## TOYOTA           3.7   3.5         4.9         5.3    2.9        3.2
## EAGLE            4.0   3.6         4.6         2.8    4.3        4.9
## HONDA            3.3   3.9         5.0         4.7    3.9        5.0
##        Successful AvantGarde PoorValue  S1  S2  S3 Overall
## G20           5.3        4.3       3.4 4.3 5.9 8.6     6.3
## FORD          4.2        3.6       4.3 2.1 6.0 2.1     3.9
## AUDI          5.0        3.6       4.3 6.0 7.7 3.4     6.0
## TOYOTA        5.5        4.9       3.5 6.1 3.5 8.1     5.5
## EAGLE         3.7        4.4       3.6 3.3 3.1 5.8     4.0
## HONDA         5.6        3.9       2.6 6.0 5.5 8.3     6.5
```

```
# assignamos lo snombres de las marcas y atributos a un vector de marcas y atributos respectivamente  #

brdnames = rownames(g20.per);

attribnames = colnames(g20.per)

#Leemos la hoja excel que contiene las preferencias 'prefs'.

pref <- readWorksheet(g20, rownames=1, sheet = "prefs", header = TRUE)
head(pref)
```

```
##   G20 FORD AUDI TOYOTA EAGLE HONDA SAAB PONTIAC BMW MERCURY
## 1   4    7    8      3     4     5    5       1   4       5
## 2   4    8    6      5     8     7    3       1   5       2
## 3   8    5    9      4     1     7    7       2   4       4
## 4   7    1    8      1     4     6    5       5   7       3
## 5   9    8    8      3     5     4    3       2   8       6
## 6   5    6    5      5     2     4    8       4   4       7
```

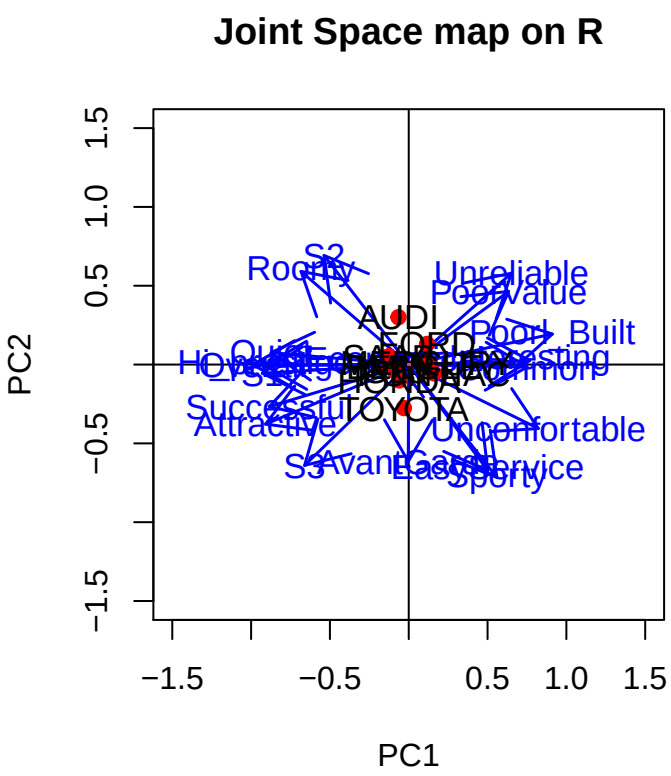
```
dim(pref) #comprobar la tabla de datos
```

```
## [1] 75 10
```

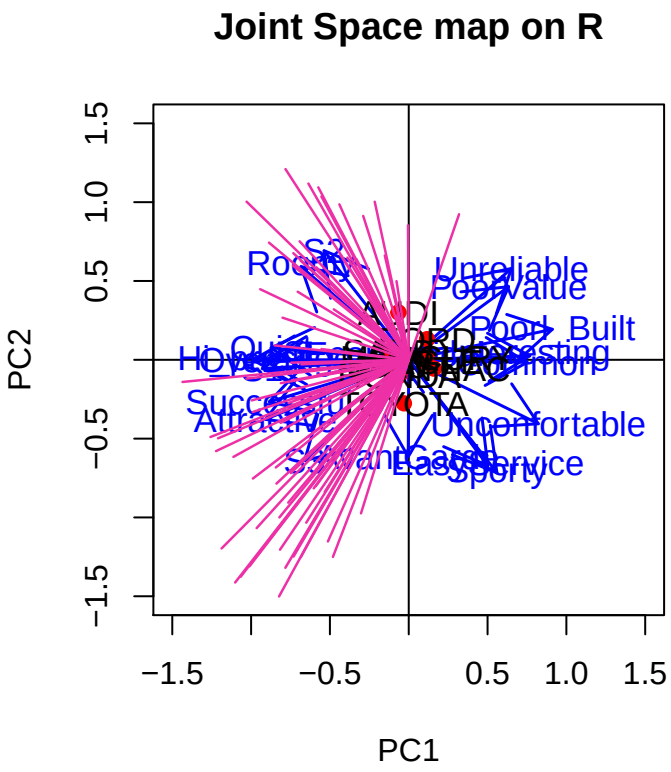
```
pref[1:10,] #ver las 10 primeras observaciones
```

```
##    G20 FORD AUDI TOYOTA EAGLE HONDA SAAB PONTIAC BMW MERCURY
## 1    4    7    8      3     4     5    5       1   4       5
## 2    4    8    6      5     8     7    3       1   5       2
## 3    8    5    9      4     1     7    7       2   4       4
## 4    7    1    8      1     4     6    5       5   7       3
## 5    9    8    8      3     5     4    3       2   8       6
## 6    5    6    5      5     2     4    8       4   4       7
## 7    3    9    7      4     4     3    6       4   3       6
## 8    4    7    9      3     1     7    9       3   6       6
## 9    8    6    6      4     5     5    1       2   8       7
## 10   6    4    6      3     2     8    7       3   1       8
```

```
## --- Leer la función JSM que construirá los mapas conjuntos --- ##
pref0 = pref*0; rownames(pref0) = NULL
source("JSM.R")
#Contruimos la visualización de la estructura de los mercados, un mapa de percepciones y después un map
JSM(g20.per, pref0)
```




```
JSM(g20.per, pref)
```



Análisis multidimensinal no métrico

El problema que resuelve el MDS no métrico es encontrar n puntos en k dimensiones cuyas coordenadas vendrán dadas por las filas de la matriz $\mathbf{Z}$ la cual generará una matriz cuadrada de distancias de dimensión nxn entre las marcas,$\{d_{ij}\}$, que deberán aproximarnos las distancias o similitudes originales, denominadas habitualmente proximidades, $\{p_{ij}\}$. En general, la construcción de mapas para representar gráficamente las percepciones que los consumidores tienen de las marcas se especifica mediante una función de representación visual,$f : p_{ij} \rightarrow d_{ij}(Z)$, la cual especifica cómo las proximidades deben relacionarse con las distancias. Normalmente una configuración en dos o tres dimensiones cuyas distancias derivadas, $d_{ij}(Z)$, satisfagan la función de representación visual, f, tan fielmente como sea posible. Para saber si el modelo ha alcanzado su propósito necesitamos alguna medida de la discrepancia entre las distancias derivadas del espacio de percepciones y las distancias originales. Esa medida es la expresión matemática que agrega los errores cometidos en la representación visual de cada una de las marcas, $e_{ij} = f(p_{ij}) - d_{ij}(Z)$, y se la conoce con el nombre de stress.

Coordenadas en el espacio reducido

En el caso de querer representar las proximidades originales en un espacio reducido de k dimensiones, Z_k, el modelo debe obtener las coordenadas de las n marcas en el espacio reducido, de tal manera que las distancias entre ellas en el espacio reducido, $d_{ij}(Z)$, satisfagan de forma satisfactoria la función de representación visual,

$$d_{ij}(Z) = \sqrt{\sum_{k=1}^{K}(z_{ik} - z_{jk})^2}$$

.

Una vez estimada la matriz de coordenadas de las marcas en el espacio reducido debemos conocer cuán fiel es la representación visual obtenida. Para ello, primero, debemos especificar una función concreta de la

representación visual, $f(p_{ij})$. Esta difiere en función de los datos originales, p_{ij}. Si los datos son métricos, una matriz de distancias o de correlaciones, por ejemplo, y no existe un argumento teórico que sugiera lo contrario, la función de representación visual viene dada por

$$p_{ij} = a + bp_{ij} = d_{ij}(Z)$$

,

para todos los pares *(i,j)*. En el caso de que las proximidades originales, p_{ij}, expresen sólo una relación de orden, como puede ser un orden de preferencias o de similitud percibida entre marcas, estaríamos ante un modelo MDS ordinal donde la relación entre las proximidades, p_{ij}, y las distancias derivadas en el espacio reducido, $d_{ij}(Z)$,vendría dada por

si $p_{ij} < p_{kj}$, entonces $d_{ij}(Z) \leq d_{kj}(Z)$,

para todas las marcas de la muestra. Esta es una especificación débil de la función de regresión monótona (ver Borg Groenen, [1997:32] para otras variantes).

La medida en la cual discrepan las distancias estimadas en el espacio reducido, $d_{ij}(Z_{k}), y las predichas a partir de las proximidades originales, $f(p_{ij})$, nos estiman los errores cometidos en la representación visual,

$$e_{ij}^2 = (f(p_{ij}) - d_{ij}(Z))^2$$

.

Su suma nos ofrece un indicador de la medida en la que las proximidades originales, p_{ij}, en su conjunto difieren de las derivadas en el espacio reducido, y se conoce con el nombre de *raw stress*,

$$\sigma_r = \sum_{\forall_{i \neq j}} (f(p_{ij}) - d_{ij}(Z))^2$$

.

Debido a que esta medida es sensible a la escala en la cual están medidas las proximidades originales el stress básico se suele dividir por la suma del cuadrado de las distancias derivas en el espacio reducido,

$$\sigma_1^2 = \frac{\sum_{\forall_{i \neq j}} (f(p_{ij}) - d_{ij}(Z))^2}{\sum_{\forall_{i \neq j}} (d_{ij}^2(Z))^2}$$

.

Su raíz cuadrada se conoce con el nombre de *Stress-1* de Kruskal,

$$\sigma_1 = \sqrt{\frac{\sum_{\forall_{i \neq j}} (f(p_{ij}) - d_{ij}(Z))^2}{\sum_{\forall_{i \neq j}} (d_{ij}^2(Z))^2}}$$

.

donde ahora hemos sustituido la función de representación visual por su predicción en el espacio reducido, Z_k. Cuanto menor sea su valor, mejor será la representación visual obtenida. Su especificación en el lenguaje **R** nos proporciona el valor del cuadrado del *stress-1* de Kruskal expresado en porcentaje.

También es posible analizar con detalle la formación de los errores que forman el Stress-1 de Kruskal a través del diagrama de Shepard el cual nos relaciona las proximidades originales p_{ij} con las derivadas del espacio reducido, $d_{ij}(Z)$, a través de las predicciones que se derivan de aplicar la función de representación visual, $\hat{d}_{ij}(Z)$. En el caso en el que los datos originales, p_{ij} pij, midan un orden de similitud o preferencia la relación entre las proximidades y las distancias que se derivan de aplicar la función de representación visual tiene una forma visual de escalera. La discrepancia entre las distancias en el espacio reducido y las estimadas vienen dadas por la distancia vertical y nos indican los errores de la representación visual $f(p_{ij}) - d_{ij}(Z)$. Veamos un ejemplo

```
#####MDS datos hospital
kcpref<-read.table("hospital.prefs.txt", header=T)
summary(kcpref)
```

```
##        HK              B               C               D
##  Min.   :1.00   Min.   :1.00   Min.   :1.00   Min.   :1.00
##  1st Qu.:7.00   1st Qu.:3.00   1st Qu.:2.00   1st Qu.:8.00
##  Median :9.00   Median :6.00   Median :5.00   Median :8.00
##  Mean   :7.41   Mean   :5.23   Mean   :4.44   Mean   :7.74
##  3rd Qu.:9.00   3rd Qu.:7.00   3rd Qu.:6.00   3rd Qu.:9.00
##  Max.   :9.00   Max.   :9.00   Max.   :8.00   Max.   :9.00
##        E               F               G               H
##  Min.   :1.00   Min.   :1.00   Min.   :1.00   Min.   :1.00
##  1st Qu.:1.00   1st Qu.:2.00   1st Qu.:3.00   1st Qu.:2.00
##  Median :3.00   Median :4.00   Median :3.00   Median :3.00
##  Mean   :3.41   Mean   :3.98   Mean   :3.57   Mean   :3.47
##  3rd Qu.:5.00   3rd Qu.:5.00   3rd Qu.:5.00   3rd Qu.:5.00
##  Max.   :8.00   Max.   :8.00   Max.   :7.00   Max.   :7.00
##        I
##  Min.   :1.00
##  1st Qu.:4.00
##  Median :7.00
##  Mean   :5.74
##  3rd Qu.:7.00
##  Max.   :8.00
```

```
str(kcpref)
```

```
## 'data.frame':    270 obs. of  9 variables:
##  $ HK: int  9 1 9 8 9 9 7 9 9 9 ...
##  $ B : int  1 5 1 2 1 2 9 6 7 1 ...
##  $ C : int  2 3 2 1 2 1 8 5 6 2 ...
##  $ D : int  8 9 8 9 6 8 5 8 4 8 ...
##  $ E : int  6 4 6 4 8 6 1 1 5 6 ...
##  $ F : int  5 2 3 3 3 5 6 4 8 3 ...
##  $ G : int  3 7 4 5 4 3 3 3 3 4 ...
##  $ H : int  4 6 5 6 5 4 2 2 2 5 ...
##  $ I : int  7 8 7 7 7 7 4 7 1 7 ...
```

Ahora calculamos las distancias entre las preferencias por las marcas, para ello trasponemos la tabla de datos y después calculamos las distancias euclidianas (la función `dist` opera sobre las filas de la tabla de datos).

```
kcpref.dist<-dist(t(kcpref))
kcpref.dist
```

```
##        HK       B       C       D       E       F       G       H
## B 70.8872
## C 71.9514 22.3159
## D 60.0000 65.5210 75.4122
## E 76.8830 72.3326 58.4808 90.3493
## F 70.1213 48.8057 32.7414 88.4703 43.5201
```

```
## G 86.6603 58.3866 52.1057 75.5381 44.9110 60.0750
## H 84.3267 66.3325 57.7581 79.6430 34.4093 58.0861 17.7482
## I 71.4843 61.0164 64.6297 39.8748 68.3301 76.9480 44.8999 49.6689
```

Cargamos el paquete `MASS` de la biblioteca de programas que tenemos instalados. La función `isoMDS` estima las coordenadas en un espacio reducido de dos dimensiones si no le indicamos lo contrario. El resultado lo asignamos a un objeto, `kcprefs.mds`, en este caso, y nos muestra el proceso iterativo para calcular las coordenadas de las marcas en el espacio reducido de dos dimensiones. Necesita cinco iteraciones para alcanzar la mejor configuración en el espacio reducido, $Z_{k=2}$. La medida de bondad de la representación visual en el espacio reducido es el cuadrado la fórmula del `stress-1` de Kruskal, expresada en porcentaje. Así, en el ejemplo que nos ocupa es de un 8,49%.

```
###MDS
library(MASS)
kcpref.mds<-isoMDS(kcpref.dist)
```

```
## initial  value 10.799238
## iter   5 value 8.667102
## final  value 8.483966
## converged
```

```
names(kcpref.mds)
```

```
## [1] "points" "stress"
```

```
kcpref.mds$stress
```

```
## [1] 8.48397
```

```
kcpref.mds$points
```

```
##         [,1]     [,2]
## HK  40.95243  27.8877
## B    3.04205  18.9440
## C   -8.64118  22.4593
## D   50.58884 -15.3233
## E  -38.67031  -6.3604
## F  -24.19746  27.6714
## G  -16.17744 -25.6838
## H  -25.85696 -22.3035
## I   18.96003 -27.2915
```

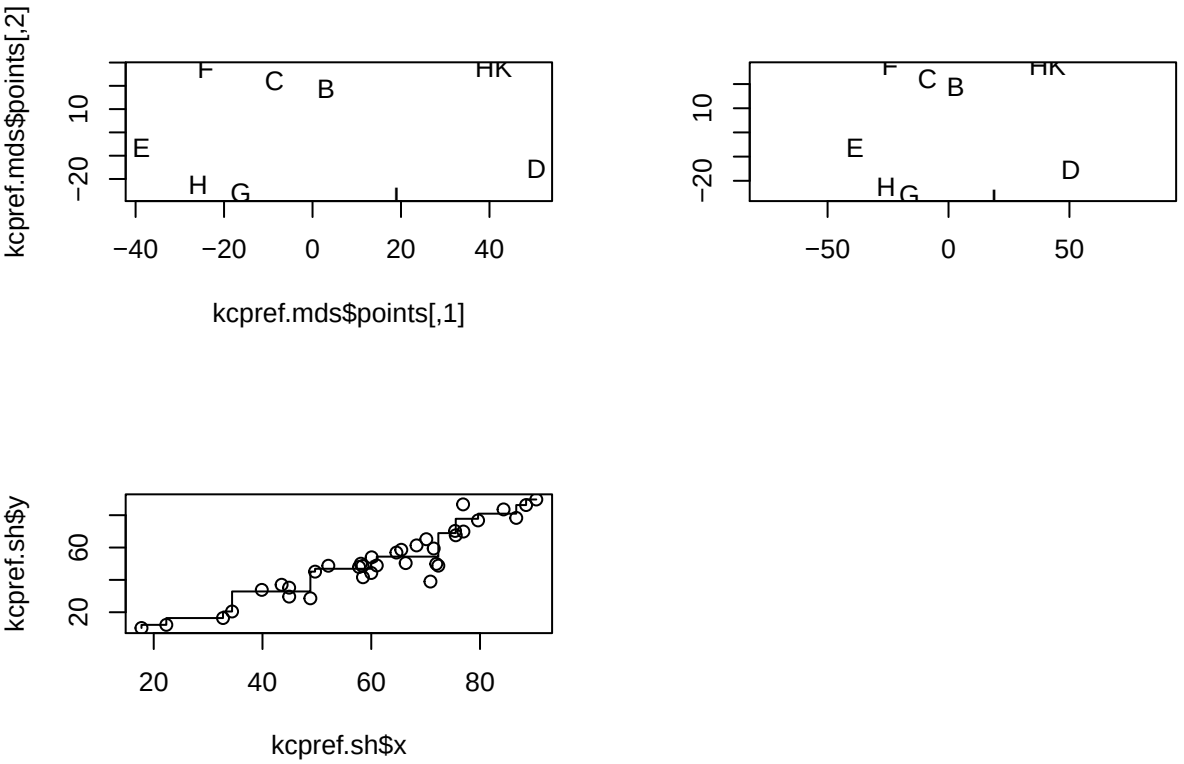
Para la prepresentación gráfica del espacio reducido de dos dimensiones podemos utilizar las funciones gráficas primitivas de **R**, `plot` y `text`. O bien las funciones gráficas que ofrece el paquete `MASS`, concretamente en este caso `eqscplot`, donde las escalas de ambos ejes son iguales `-eqscplot` significa *equal scale plot*.

```
par(mfrow=c(2,2))
##Plot
plot(kcpref.mds$points, type="n")
text(kcpref.mds$points, labels=names(kcpref))
```

```
##Plot
eqscplot(kcpref.mds$points, type="n")
text(kcpref.mds$point, labels=names(kcpref))
kcpref.sh <- Shepard(kcpref.dist, kcpref.mds$points)
names(kcpref.sh)
```

```
## [1] "x"  "y"  "yf"
```

```
#Plot
plot(kcpref.sh)
lines(kcpref.sh$x, kcpref.sh$yf, type="S")
par(mfrow=c(1,1))
```



El objeto `kcpref.sh` tiene tres atributos, las proximidades `iniciales`, p_{ij}, denominadas `x` en **R**, las distancias derivadas del espacio reducido, $d_{ij}(Z)$, y las estimadas, $\hat{d}_{ij}(Z) = f(p_{ij})$. El gráfico de Shepard nos muestra las proximidades originales, p_{ij}, en el eje de las x , y las distancias, tanto las derivadas del espacio reducido, $d_{ij}(Z)$, como las estimadas, $\hat{d}_{ij}(Z) = f(p_{ij})$ en el de las y. Una línea en forma de escalera conecta las proximidades, p_{ij}, con las distancias derivadas, $\hat{d}_{ij}(Z) = f(p_{ij})$. Las distancias verticales nos miden la contribución de cada distancia a la medida de stress minimizada.

Otros modelos

SMACOF

El paquete `SMACOF` (de Leeuw & Mair, 2009) implementado en **R** ofrece es un conjunto de funciones que implementan diferentes algoritmos de la familia MDS (Borg & Groenen, 2005). SMACOF minimiza un función de *stress* por medi de un algoritm de mayorización. El paquete smacof implemnta los siguientes

procedimientos: SMACOF simple en un matrices de disimilaritud simétricas, SMACOF con restriccines en las configuraciones, three-way SMACOF para diferencias individuales (INDSCAL, IDIOSCAL, etc.), matrices rectangulares (unfolding), SMACOF esférico, tanto con datos métricos como no métricos.

Breakfast rating for rectangular SMACOF As a metric unfolding example we use the breakfast dataset from Green and Rao (1972) which is also analyzed in Borg and Groenen (2005, Chapter 14). 42 individuals were asked to order 15 breakfast items due to their preference. These items are: toast = toast pop-up, butoast = buttered toast, engmuff = English muffin and margarine, jdonut = jelly donut, cintoast = cinnamon toast, bluemuff = blueberry muffin and margarine, hrolls = hard rolls and butter, toastmarm = toast and marmalade, butoastj = buttered toast and jelly, toastmarg = toast and margarine, cinbun = cinnamon bun, danpastry = Danish pastry, gdonut = glazed donut, cofcake = coffee cake, and cornmuff = corn muffin and butter. For this 42×15 matrix we compute a rectangular SMACOF solution.

```
dist(t(kcpref))
```

```
##        HK       B       C       D       E       F       G       H
## B 70.8872
## C 71.9514 22.3159
## D 60.0000 65.5210 75.4122
## E 76.8830 72.3326 58.4808 90.3493
## F 70.1213 48.8057 32.7414 88.4703 43.5201
## G 86.6603 58.3866 52.1057 75.5381 44.9110 60.0750
## H 84.3267 66.3325 57.7581 79.6430 34.4093 58.0861 17.7482
## I 71.4843 61.0164 64.6297 39.8748 68.3301 76.9480 44.8999 49.6689
```

The data set we provide for three-way SMACOF is described in Bro (1998). The raw data consist of ratings of 10 breads on 11 different attributes carried out by 8 raters. Note that the bread samples are pairwise replications: Each of the 5 different breads, which have a different salt content, was presented twice for rating. The attributes are bread odor, yeast odor, off-flavor, color, moisture, dough, salt taste, sweet taste, yeast taste, other taste, and total. First we fit an unconstrained solution followed by a model with identity restriction. ##SensoMineR El apquete SensoMineR ofrece un conjunto de funciones para realizar el trabajo de campo necesario para obtener datos de consumidores y marcas, manipularlos y analizarlos para construir mapas de percepciones y preferencias.

```
######
library(SensoMineR)
```

```
## Loading required package: FactoMineR
```

```
data(chocolates)
names(sensochoc)
```

```
##  [1] "Panelist"    "Session"     "Rank"        "Product"     "CocoaA"
##  [6] "MilkA"       "CocoaF"      "MilkF"       "Caramel"     "Vanilla"
## [11] "Sweetness"   "Acidity"     "Bitterness"  "Astringency" "Crunchy"
## [16] "Melting"     "Sticky"      "Granular"
```

```
head(sensochoc)
```

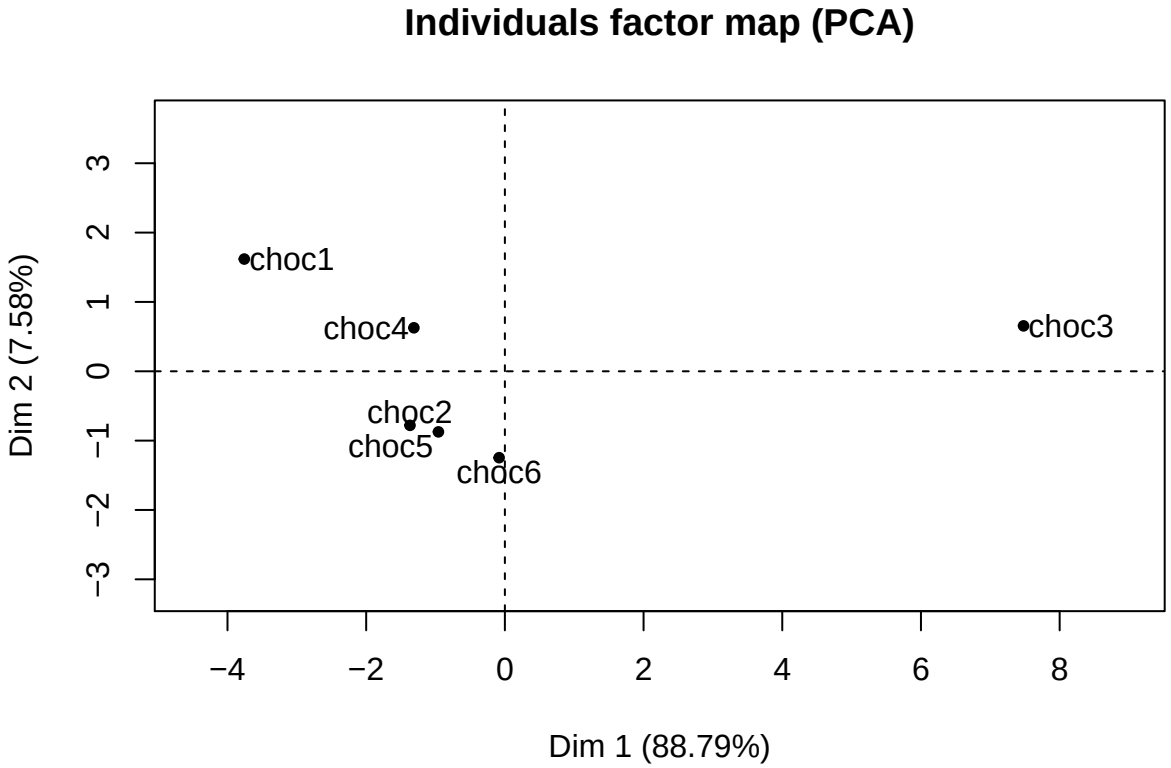
```
##      Panelist Session Rank Product CocoaA MilkA CocoaF MilkF Caramel
## I001        1       1    1   choc6      7     6      6     5       5
## I002        1       1    6   choc3      6     7      2     7       8
```

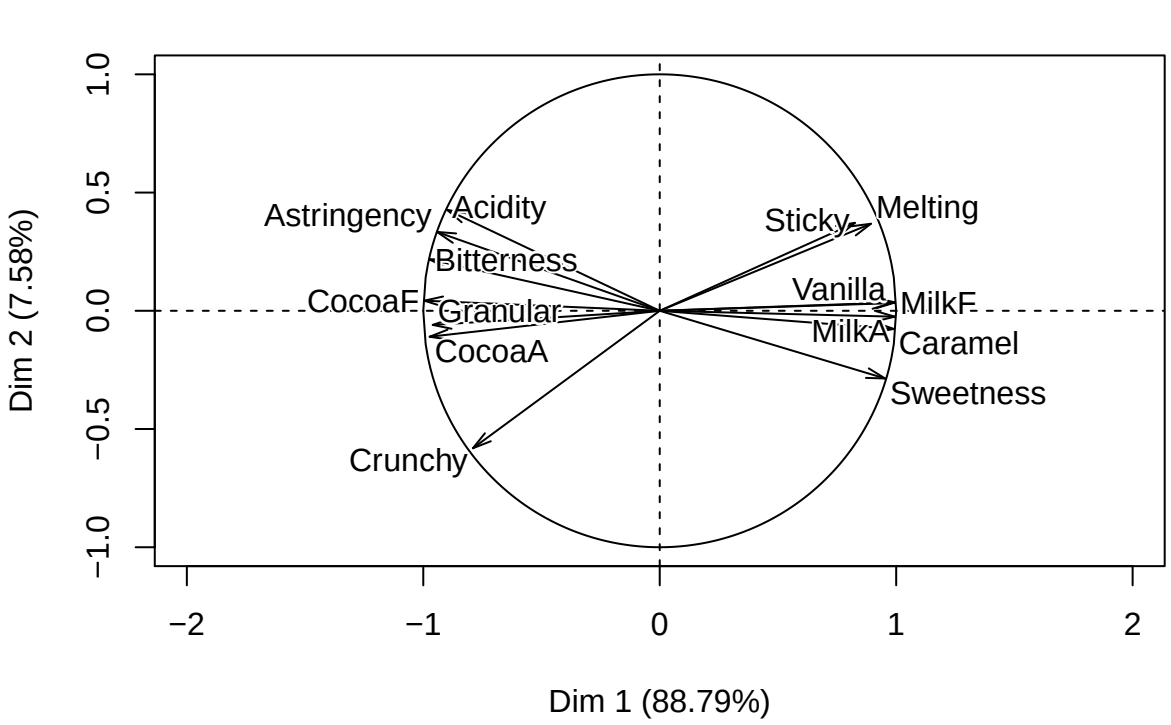
```
## I003        1       1    3   choc2      8     6      5     4       7
## I004        1       1    5   choc1      7     8      8     3       3
## I005        1       1    2   choc4      8     5      4     4       4
## I006        1       1    4   choc5      7     5      3     5       6
##      Vanilla Sweetness Acidity Bitterness Astringency Crunchy Melting
## I001       3         7       2          4           5       8       3
## I002       4         7       2          2           3       3       8
## I003       4         5       5          6           6       7       5
## I004       2         4       7          8           6       3       2
## I005       4         5       6          6           4       6       3
## I006       2         5       4          7           4       6       6
##      Sticky Granular
## I001      4        3
## I002      6        5
## I003      4        3
## I004      3        5
## I005      7        3
## I006      4        7
```

```
resaverage<-averagetable(sensochoc, formul = "~Product+Panelist",
                         firstvar = 5)
resaverage
```

```
##        CocoaA   MilkA  CocoaF   MilkF Caramel Vanilla Sweetness Acidity
## choc1 7.08621 3.58621 8.06897 1.56897 1.67241 1.10345   3.13793 4.65517
## choc2 6.55172 4.00000 6.91379 2.37931 2.77586 1.81034   4.62069 3.13793
## choc3 4.67241 6.05172 3.37931 7.70690 6.32759 3.67241   7.60345 1.56897
## choc4 6.25862 4.10345 6.68966 2.58621 2.67241 2.12069   4.29310 3.93103
## choc5 6.79310 4.17241 6.79310 3.12069 3.41379 1.79310   5.22414 3.08621
## choc6 6.36207 4.56897 6.22414 3.36207 3.25862 1.91379   5.62069 2.67241
##       Bitterness Astringency Crunchy Melting  Sticky Granular
## choc1    7.06897     4.75862 5.96552 4.74138 3.75862  3.44828
## choc2    4.94828     3.15517 7.70690 4.32759 3.82759  3.15517
## choc3    1.39655     1.20690 2.98276 7.31034 5.03448  1.60345
## choc4    5.18966     3.68966 6.10345 4.37931 4.10345  3.55172
## choc5    4.87931     3.10345 6.63793 4.74138 3.22414  3.06897
## choc6    4.18966     2.75862 7.32759 4.20690 3.93103  3.17241
```

```
res.pca = PCA(resaverage, scale.unit = TRUE)
```



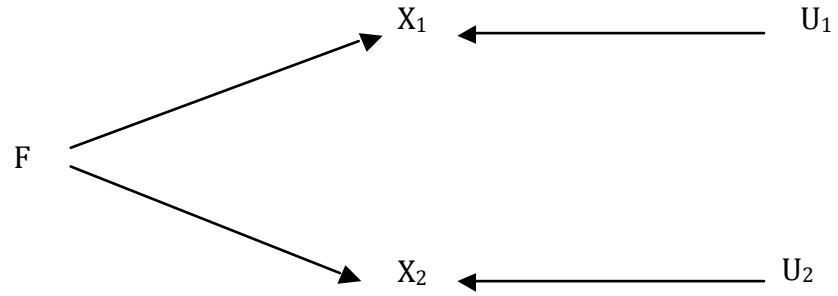



```
res.pca
```

```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 6 individuals, described by 14 variables
## *The results are available in the following objects:
```

```
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
## 4  "$var$cor"         "correlations variables - dimensions"
## 5  "$var$cos2"        "cos2 for the variables"
## 6  "$var$contrib"     "contributions of the variables"
## 7  "$ind"             "results for the individuals"
## 8  "$ind$coord"       "coord. for the individuals"
## 9  "$ind$cos2"        "cos2 for the individuals"
## 10 "$ind$contrib"     "contributions of the individuals"
## 11 "$call"            "summary statistics"
## 12 "$call$centre"     "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"      "weights for the individuals"
## 15 "$call$col.w"      "weights for the variables"
```

Anexo: Modelo factorial exploratorio



X1=b1F+d1U1 X2=b2F+d2U2 X1=0.8F+0.6U1 X2=0.6F+0.8U2

Cov(F,U1)=cov(F,U2)=cov(U1,U2)=0

Tabla 1 Ilustración de factores y variables: dos variables y un factor común*

Casos	F	U1	U2	X1=0.8F+0.6U1	X2=0.6F+0.8U2
1	1	1	1	1.4	1.4
2	1	1	-1	1.4	-0.2
3	1	-1	1	0.2	1.4
4	1	-1	-1	0.2	-0.2
5	-1	1	1	-0.2	0.2
6	-1	1	-1	-0.2	-1.4
7	-1	-1	1	-1.4	0.2
8	-1	-1	-1	-1.4	1.4

Kim y Mueller (1978:14)

Media=$\sum(Xi)/N = E(X) = \bar{X}$.

Varianza = $\sum[Xi - E(X)]^2/N = E(X - E(X))^2 = V_x$.

Cov(X,Y)= $\sum[(Xi - \bar{X})(Yi - \bar{Y})]/N = E((X - \bar{X})(Y - \bar{X})$.

Si las variables están estandarizadas,

$Cov(X,Y) = E(XY) = r_{xy}$ (coeficiente de correlación de Pearson).

$Var(X1) = E(X1 - \bar{X1})^2 = E(X1)^2 = E[X1 = b_1F + d_1U1]^2 =$(desarrollando el cuadrado).

$Var(X1) = E[b_1^2 + F^2 + d2_1^2U1^2 + 2b_1Fd_1U1] = b_1^2E(F^2) + d_1^2E(U1^2) + 2b_1d_1E(FU1)] =$

$Var(X1) = b_1^2Var(F) + d_1^2Var(U1) + 2b_1d_1Cov(FU1)]$ si $Cov(FU1) = 0$ entonces

$Var(X1) = b_1^2Var(F) + d_1^2Var(U1)]$ Si las variables están estandarizadas, entonces $Var(F) = Var(U1) = 1$.

$Var(X1) = b_1^2 + d_1^2 = 1$.

$Cov(F, X1) = E[(F - \bar{F})(X1 - \bar{X1})] = E[FX1] = E[F(b1_1F + d_1U1) = b_1E(F^2) + d_1E(F, U1)$.

$Cov(F, X1) = b_1Var(F) + d_1Cov(F, U1)$ pero $Cov(F, U1) = 0$ y $Var(F) = 0$.

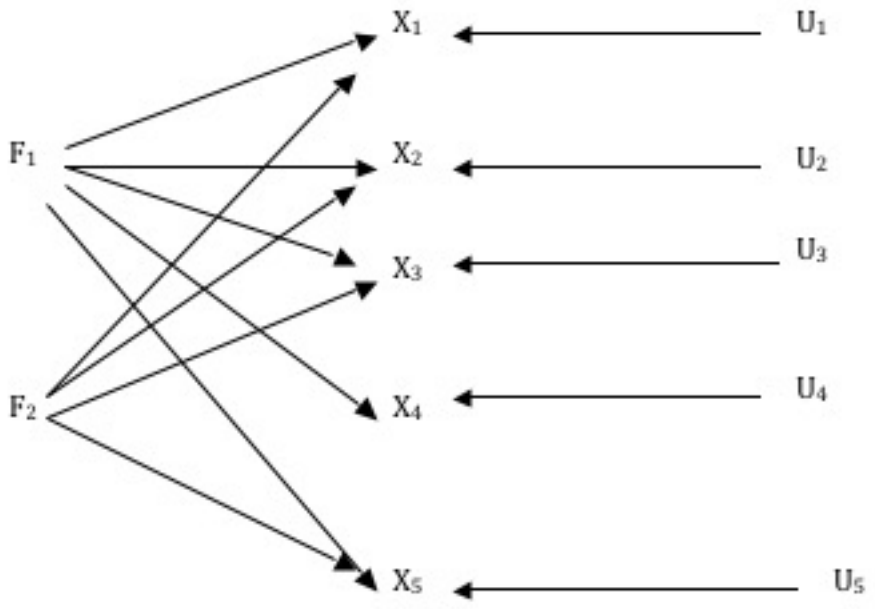
$Cov(F, X1) = r_{FX} = b_1$.

De igual manera

$Cov(X1, X2) = r_{X1,X2} = b_1b_2$.

$r_{ij.F} = 0$.

CON DOS FACTORES TENDRÍAMOS EL SIGUIENTE CASO



Hipótesis

$Cov(F1, F2) = cov(FiUj) = cov(UkUk) = 0$

$X1 = b_{11}F1 + b_{12}F2 + d_1U1$

...

...

...

$X5 = b_{51}F1 + b_{52}F2 + d_5U5$

$Var(Xi) = b_{i1}^2 + b_{i2}^2 + di^2$

La proporción de la varianza de una variable observable explicada por los factores comunes viene dada por:

$H_i^2 = b_{i1}^2 + b_{i2}2$

La covarianza entre dos variables observables:

$Cov(Xi, Xk) = r_{ik} = b_{i1}b_{k1} + b_{i2}b_{k2}$

Jae-On Kim y Charles W. Mueller (1978) Introduction to Factor Analysis. Sage University Paper series on Quantitative Application in the Social Sciences, series no. 07-013. London-Berverly Hills: Sage.

Cuando los componentes principales no se pueden interpretar bien, es posible utilizar otros paquetes que nos proporcionan la posibilidad de girar los componentes sobre su eje con el objeto de encontrar una matriz de correlaciones más interpretable. En el entorno **R** podemos utilizar los paquetes `psych` y `GPArotation`. El primer paquete realiza la extracción de los componentes principales y utitliza al segundo para encontrar una matriz de correlaciones más interpretable. Vamos a utilizar los mismos datos sobre arrestos para ilustrar su uso. En este ejemplo utilizamos una rotación habitual, `varimax`, pero también es posible utilizar otros procedimientos para rotar la matriz de estructura inicial: `quatimax`, `promax`, `oblimin`, `simplimax`, o `cluster`.

```
#Lectura de datos
g20per<-read.table("g20per.txt", header=TRUE)
names(g20per)
```

```
##  [1] "G20"     "FORD"    "AUDI"    "TOYOTA"  "EAGLE"   "HONDA"   "SAAB"
##  [8] "PONTIAC" "BMW"     "MERCURY"
```

Para hacer el análisis tenemos que transponer la tabla de datos que acabamos de leer, de tal manera que las filas ahora serán las marcas, y las columnas, los atributos.

```
g20<-data.frame(t(g20per))
names(g20)
```

```
##  [1] "Atract"     "Silec"      "Nofiab"     "MalaCons"   "Interesan"
##  [6] "Deport"     "Incomod"    "Espacioso"  "FacilMan"   "Prestigio"
## [11] "Ordinar"    "Economico"  "Existos"    "Vanguard"   "Malacompra"
```

Después cargamos los paquetes `psych` y `GPArotation`.

```
library(psych)
library(GPArotation)

g20.principal<- principal(g20, nfactors=2, rotate="varimax")
```

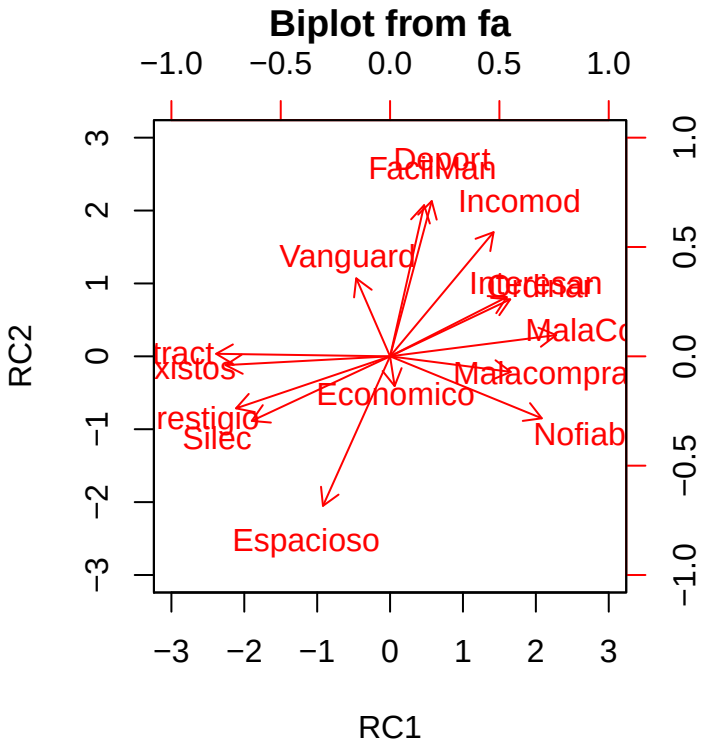
```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done

## In factor.stats, the correlation matrix is singular, an approximation is used

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs =
## np.obs, : In factor.stats, the correlation matrix is singular, and we
## could not calculate the beta weights for factor score estimates

## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
## I was unable to calculate the factor score weights, factor loadings used instead
```

```
g20.principal
```

```
## Principal Components Analysis
## Call: principal(r = g20, nfactors = 2, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##              RC1   RC2    h2    u2
## Atract     -0.99  0.01 0.989 0.011
## Silec      -0.79 -0.37 0.761 0.239
## Nofiab      0.87 -0.35 0.878 0.122
## MalaCons    0.95  0.12 0.916 0.084
## Interesan   0.67  0.34 0.560 0.440
## Deport      0.24  0.89 0.844 0.156
## Incomod     0.59  0.71 0.854 0.146
## Espacioso  -0.38 -0.85 0.878 0.122
## FacilMan    0.19  0.86 0.785 0.215
## Prestigio  -0.88 -0.30 0.864 0.136
## Ordinar     0.69  0.33 0.578 0.422
## Economico   0.03 -0.17 0.029 0.971
## Existos    -0.94 -0.05 0.886 0.114
## Vanguard   -0.19  0.45 0.236 0.764
## Malacompra  0.69 -0.09 0.484 0.516
##
##                        RC1  RC2
## SS loadings           6.95 3.59
## Proportion Var        0.46 0.24
## Cumulative Var        0.46 0.70
## Proportion Explained  0.66 0.34
## Cumulative Proportion 0.66 1.00
##
## Test of the hypothesis that 2 components are sufficient.
##
## The degrees of freedom for the null model are  105  and the objective function was  199.47
## The degrees of freedom for the model are 76  and the objective function was  183.44
## The total number of observations was  10  with MLE Chi Square =  336.3  with prob <  2e-34
##
## Fit based upon off diagonal values = 0.93
```

```
biplot(g20.principal )
```




```
g20.principal<- principal(g20, nfactors=3, rotate="varimax")
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

```
## In factor.stats, the correlation matrix is singular, an approximation is used
```

```
## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs =
## np.obs, : In factor.stats, the correlation matrix is singular, and we
## could not calculate the beta weights for factor score estimates
```

```
## In factor.scores, the correlation matrix is singular, an approximation is used
```

```
## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done
```

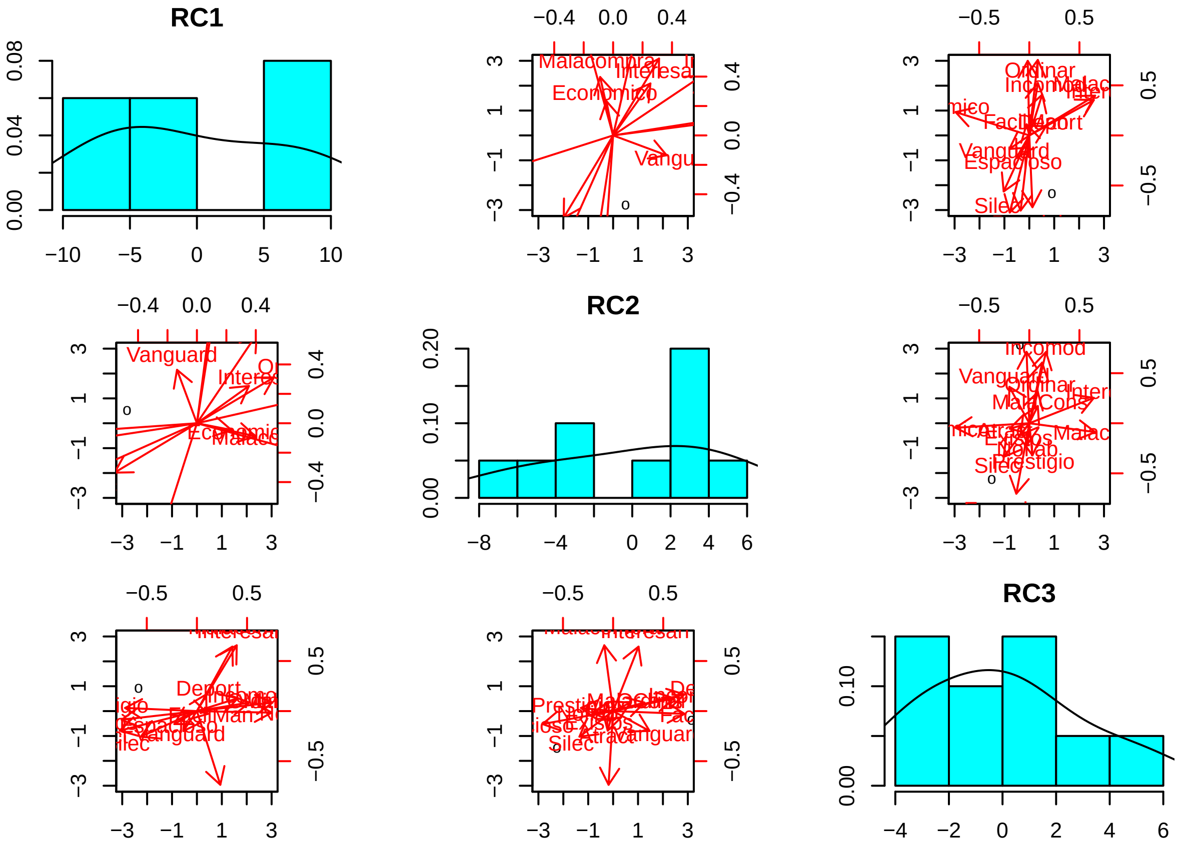
```
## I was unable to calculate the factor score weights, factor loadings used instead
```

```
g20.principal
```

```
## Principal Components Analysis
## Call: principal(r = g20, nfactors = 3, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##              RC1   RC2   RC3   h2    u2
## Atract     -0.96 -0.07 -0.24 0.99 0.010
## Silec      -0.70 -0.42 -0.32 0.77 0.233
## Nofiab      0.93 -0.26 -0.02 0.93 0.069
```

```
## MalaCons    0.94  0.22  0.11 0.94 0.055
## Interesan   0.44  0.32  0.80 0.94 0.057
## Deport      0.12  0.89  0.21 0.85 0.151
## Incomod     0.51  0.76  0.16 0.86 0.142
## Espacioso  -0.28 -0.88 -0.16 0.88 0.122
## FacilMan    0.14  0.89 -0.03 0.81 0.186
## Prestigio  -0.90 -0.40  0.04 0.96 0.035
## Ordinar     0.65  0.39  0.11 0.59 0.409
## Economico   0.29 -0.06 -0.92 0.93 0.067
## Existos    -0.94 -0.14 -0.11 0.91 0.089
## Vanguard   -0.17  0.45 -0.25 0.30 0.703
## Malacompra  0.50 -0.11  0.82 0.93 0.068
##
##                        RC1  RC2  RC3
## SS loadings           6.20 3.88 2.52
## Proportion Var        0.41 0.26 0.17
## Cumulative Var        0.41 0.67 0.84
## Proportion Explained  0.49 0.31 0.20
## Cumulative Proportion 0.49 0.80 1.00
##
## Test of the hypothesis that 3 components are sufficient.
##
## The degrees of freedom for the null model are  105  and the objective function was  199.47
## The degrees of freedom for the model are 63  and the objective function was  178.81
## The total number of observations was  10  with MLE Chi Square =  208.61  with prob <  1.7e-17
##
## Fit based upon off diagonal values = 0.98
```

```
biplot(g20.principal )
```


RC1
RC2
RC3
Vanguard