Estadística Aplicada III

Métodos de agrupación: Análisis de Correspondencias, Escalamiento Multidimensional y Conglomerados

Jorge de la Vega Góngora

Departamento de Estadística,
Instituto Tecnológico Autónomo de México

Semana 12: 7/9 de noviembre de 2018

Introducción

¿Qué veremos en esta sección?

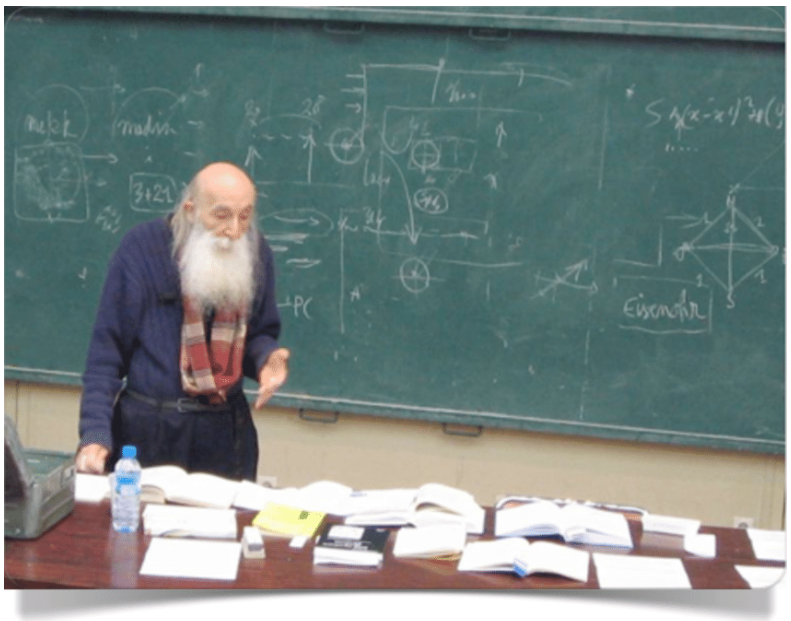
- En esta sección del curso veremos un conjunto de métodos exploratorios para datos multivariados en donde se enfatizarán más métodos de carácter no paramétrico, de *aprendizaje no supervisado*. Se revisarán los siguientes métodos:
  1. **Análisis de Correspondencias**, método para visualizar relaciones entre variables categóricas. Es un equivalente al caso de análisis de componentes principales aplicado a variables categóricas.
  2. **Escalamiento multidimensional**, busca representar N pares de objetos de los que se puede definir su *similaridad* o *distancia*, en un espacio de dimensión menor de tal manera que las similitudes o distancias en ese espacio sea cercana a la similitud o distancia original.
  3. **Conglomerados**, identifica de qué manera se pueden agrupar datos, a partir de ciertas similitudes o distancias, *sin conocer de antemano las definiciones de los grupos*. Aquí el número de grupos es desconocido.

Análisis de Correspondencias

Análisis de Correspondencias (AC)

- El Análisis de Correspondencias fue desarrollado por Jean-Paul Benzecri (1932-) en 1973 y desarrollado por sus estudiantes Lebart y Greenacre.
- Es un procedimiento gráfico para representar asociaciones en una tabla de frecuencias o conteos. Es la mejor representación bidimensional de los datos, y que proporciona una medida (llamada *inercia*) de la cantidad de información retenida en cada dimensión.
- Algebráicamente, AC busca scores f y g para los renglones y columnas con correlación *maximal*
- Versión de PCA para variables categóricas.

- Para revisar este método primero formalizaremos el concepto de *tabla de contingencia* como parte de los temas que revisaremos para análisis de datos categóricos. Ya hicimos uso de este concepto al estudiar los datos `German Credit`.



Tablas de contingencia I

Tablas de contingencia

- Supóngase que se tienen m variables categóricas $C_1, \ldots, C_m$, en donde la variable C_i tiene I_i categorías o niveles, y se tienen n observaciones que pueden tomar valores en las m posibles categorías.
- Una **tabla de contingencia** de dimensión $I_1 \times \cdots \times I_m = \prod_{i=1}^{m} I_i$ es un arreglo multidimensional donde cada celda contiene las frecuencias de las observaciones que son comunes a la intersección de los niveles de las variables categóricas.
- Se denota con $n_{i_1 i_2, \ldots, i_m}$ la frecuencia observada en la celda $(i_1, i_2, \ldots, i_m)$ y el total de observaciones es:

$$n = \sum_{i_1=1}^{I_1} \cdots \sum_{i_m=1}^{I_m} n_{i_1 i_2, \ldots, i_m}$$

Por ejemplo, Supongan que tienen n observaciones de $m = 3$ variables categóricas. Entonces la tabla de contingencia de dimensión $I_1 \times I_2 \times I_3$ se puede representar de la siguiente manera:

Tablas de contingencia II

Cuadro: Ejemplo de tabla de contingencia de $I_1 \times I_2 \times I_3$

		C_2			
C_3	C_1	1	2	...	I_2
1	1	n_{111}	n_{121}	...	n_{1I_21}
	2	n_{211}	n_{221}	...	n_{2I_21}
	⋮	⋮	⋮	⋮	⋮
	I_1	n_{I_111}	n_{I_121}	...	$n_{I_1I_21}$
2	1	n_{112}	n_{122}	...	n_{1I_22}
	2	n_{212}	n_{222}	...	n_{2I_22}
	⋮	⋮	⋮	⋮	⋮
	I_1	n_{I_112}	n_{I_122}	...	$n_{I_1I_22}$
⋮	⋮		⋮		
I_3	1	n_{11I_3}	n_{12I_3}	...	$n_{1I_2I_3}$
	2	n_{21I_3}	n_{22I_3}	...	$n_{2I_2I_3}$
	⋮	⋮	⋮	⋮	⋮
	I_1	$n_{I_11I_3}$	$n_{I_12I_3}$	...	$n_{I_1I_2I_3}$

Análisis de Correspondencias: motivación I

- Para dar seguimiento a la construcción de las ideas del AC, utilizaremos el siguiente ejemplo.
- En 7 sitios arqueológicos, se encuentran 4 tipos de cerámicas identificadas por los arqueólogos. La siguiente tabla de contingencia de dos vías de 7×4 muestra los casos de cada tipo de cerámica encontrados en cada sitio arqueológico:

Sitio	Tipo				Total
	A	B	C	D	
P_0	30	10	10	39	89
P_1	53	4	16	2	75
P_2	73	1	41	1	4
P_3	20	6	1	4	31
P_4	46	36	37	13	132
P_5	45	6	59	10	120
P_6	16	28	169	5	218
Total	283	91	333	74	781

- Es de interés saber si los sitios están relacionados con los tipos de cerámica, para asociar posibles culturas a los sitios.

Análisis de Correspondencias: motivación II

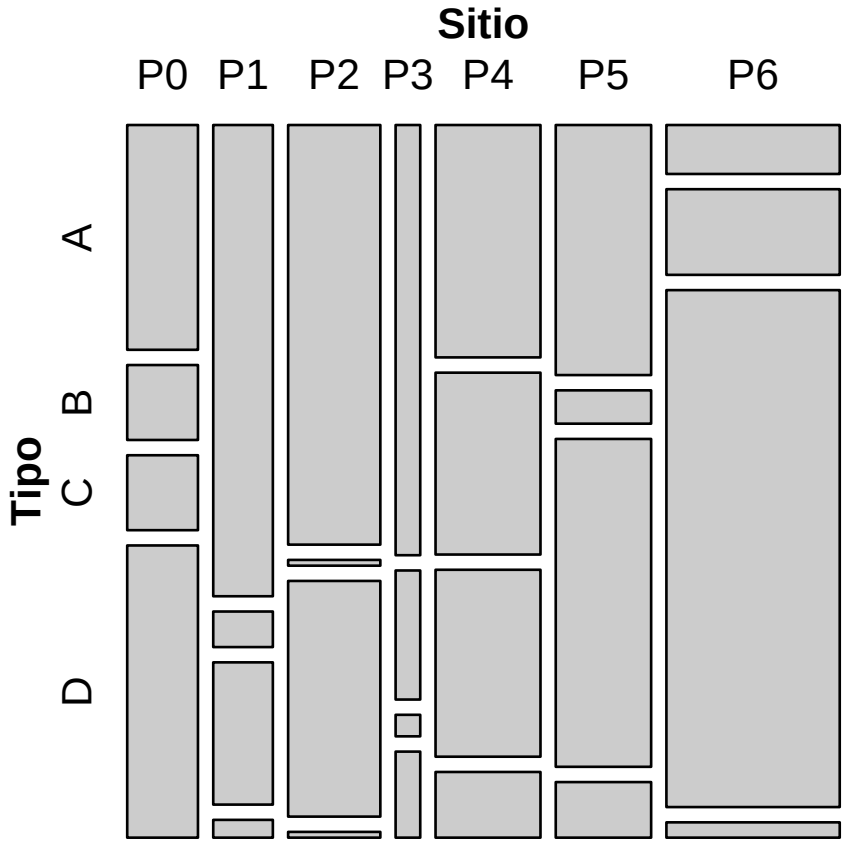
- Como ya hemos visto antes, podemos obtener la distribución conjunta y también las marginales de cada variable categórica.

```
library(vcd)
```

```
Loading required package: grid
```

```
datos <- array(
                        data = c(30,53,73,20,46,45,16,10,4,1,6,36,6,28,10,16,41,1,37,59,169,39,2,1,4,13,10,5),
                        dim = c(7,4), dimnames = list(Sitio=paste0("P",0:6),Tipo=c("A","B","C","D")))
arqueo <- as.data.frame.table(datos)
mosaic(datos,direction = c("v","h"))
```

Análisis de Correspondencias: motivación III

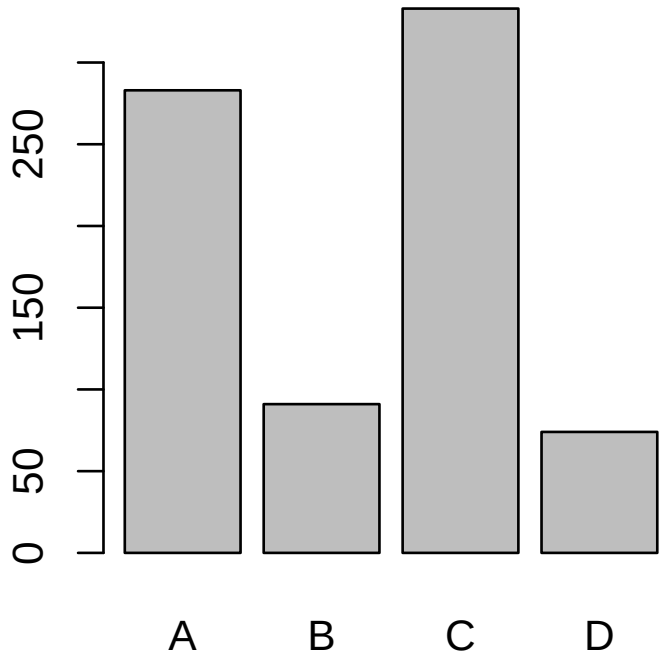


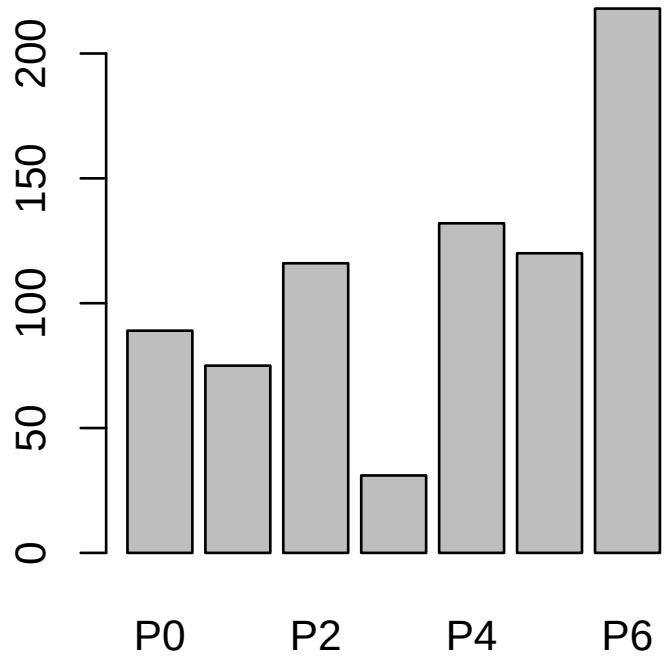
- ¿Podemos identificar sitios "parecidos" en la distribución del tipo de cerámica?

Análisis de Correspondencias: motivación IV

- Las distribuciones marginales se obtienen a continuación:

```
par(mfrow=c(1,2))
barplot(height=with(arqueo,tapply(Freq,Tipo,sum)))
barplot(height=with(arqueo,tapply(Freq,Sitio,sum)))
```





Análisis de Correspondencias: motivación V

- Nos interesa investigar si hay una relación entre el tipo de cerámica encontrado y los sitios arqueológicos.
- En general, se busca una representación de los I puntos (los $\mathrm{I} = 7$ sitios correspondientes a los renglones de la tabla de contingencia) en $\mathbb{R}^{\mathrm{J}}$, $(\mathrm{J} = 4)$ en un espacio de dimensión menor para visualizar sus distancias relativas.

Desarrollo del modelo para AC I

Definición

Sean:

- $\mathbf{X}$ = la matriz de frecuencias de rango k, donde $k = \text{mín}\{I, J\}$.
- $\mathbf{P} = \frac{1}{n}\mathbf{X}$ es la matriz con proporciones o frecuencias relativas.
- $\mathbf{r} = \mathbf{P}\mathbf{1}_J$ es el vector de sumas por columnas de los renglones con componentes $r_i = p_{i\cdot} = \sum_{j=1}^{J} p_{ij}$
- $\mathbf{c} = \mathbf{P}'\mathbf{1}_I$ es el vector de sumas por renglón de las columnas con componentes $c_j = p_{\cdot j} = \sum_{i=1}^{I} p_{ij}$
- $\mathbf{D}_r = diag(\mathbf{r})$ y $\mathbf{D}_c = diag(\mathbf{c})$

De nuestro ejemplo, $k = \text{mín}\{7, 4\} = 4$, y el resto de los conceptos aplicados nos dan:

Desarrollo del modelo para AC II

```
X <- datos; X
```

```
     Tipo
Sitio  A  B   C  D
   P0 30 10  10 39
   P1 53  4  16  2
   P2 73  1  41  1
   P3 20  6   1  4
   P4 46 36  37 13
   P5 45  6  59 10
   P6 16 28 169  5
```

```
n <- sum(X); n
```

```
[1] 781
```

```
P <- X/n; P #proporciones
```

```
     Tipo
Sitio          A           B          C           D
   P0 0.03841229 0.012804097 0.01280410 0.049935980
   P1 0.06786172 0.005121639 0.02048656 0.002560819
   P2 0.09346991 0.001280410 0.05249680 0.001280410
   P3 0.02560819 0.007682458 0.00128041 0.005121639
   P4 0.05889885 0.046094750 0.04737516 0.016645327
   P5 0.05761844 0.007682458 0.07554417 0.012804097
   P6 0.02048656 0.035851472 0.21638924 0.006402049
```

Desarrollo del modelo para AC III

```
r <- P %*% rep(1,4); r  # Vector con la distribución marginal por Sitio
```

```
Sitio       [,1]
   P0 0.11395647
   P1 0.09603073
   P2 0.14852753
   P3 0.03969270
   P4 0.16901408
   P5 0.15364917
   P6 0.27912932
```

```
c <- t(P) %*% rep(1,7); as.vector(c) # Vector con distribución marginal por Tipo
```

```
[1] 0.36235595 0.11651729 0.42637644 0.09475032
```

```
Dr <- diag(as.vector(r)); Dr
```

```
          [,1]       [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.1139565 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[2,] 0.0000000 0.09603073 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
[3,] 0.0000000 0.00000000 0.1485275 0.0000000 0.0000000 0.0000000 0.0000000
[4,] 0.0000000 0.00000000 0.0000000 0.0396927 0.0000000 0.0000000 0.0000000
[5,] 0.0000000 0.00000000 0.0000000 0.0000000 0.1690141 0.0000000 0.0000000
[6,] 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000 0.1536492 0.0000000
[7,] 0.0000000 0.00000000 0.0000000 0.0000000 0.0000000 0.0000000 0.2791293
```

```
Dc <- diag(as.vector(c)); Dc
```

```
         [,1]      [,2]      [,3]       [,4]
[1,] 0.362356 0.0000000 0.0000000 0.00000000
[2,] 0.000000 0.1165173 0.0000000 0.00000000
[3,] 0.000000 0.0000000 0.4263764 0.00000000
[4,] 0.000000 0.0000000 0.0000000 0.09475032
```

Desarrollo del modelo para AC I

- Los I puntos no tienen el mismo peso, ya que algunos renglones tienen más datos que otros, así que la distancia euclídea no es apropiada. Podemos ponderar las frecuencias relativas condicionadas al total de cada renglón:

$$\mathbf{R} = \mathbf{D}_r^{-1}\mathbf{P}$$

Esta matriz nos da las distribuciones condicionales de las columnas al respectivo nivel del renglón $R_{ij} = P(Tipo = j|Sitio = i)$ (así que cada rengón suma 1):

```
R <- solve(Dr) %*% P; R
```

```
      Tipo
               A          B          C          D
  [1,] 0.3370787 0.11235955 0.11235955 0.43820225
  [2,] 0.7066667 0.05333333 0.21333333 0.02666667
  [3,] 0.6293103 0.00862069 0.35344828 0.00862069
  [4,] 0.6451613 0.19354839 0.03225806 0.12903226
  [5,] 0.3484848 0.27272727 0.28030303 0.09848485
  [6,] 0.3750000 0.05000000 0.49166667 0.08333333
  [7,] 0.0733945 0.12844037 0.77522936 0.02293578
```

```
apply(R,1,sum)
```

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

Desarrollo del modelo para AC II

- Queremos definir una distancia entre dos renglones $\mathbf{r}_a$ y $\mathbf{r}_b$ de $\mathbf{R}$ que tome en cuenta las frecuencias relativas de las columnas:

$$D^2(\mathbf{r}_a, \mathbf{r}_b) = (\mathbf{r}_a - \mathbf{r}_b)' \mathbf{D}_c^{-1} (\mathbf{r}_a - \mathbf{r}_b)$$

  Esta distancia se conoce como la distancia χ^2.

- Bajo la transformación $\mathbf{y}_i = \mathbf{D}_c^{-1/2} \mathbf{r}_i$, notamos que la distancia χ^2 es equivalente a la distancia euclídea:

$$\begin{aligned}
D_E^2(\mathbf{y}_a, \mathbf{y}_b) &= (\mathbf{y}_a - \mathbf{y}_b)'(\mathbf{y}_a - \mathbf{y}_b) \\
&= (\mathbf{D}_c^{-1/2}\mathbf{r}_a - \mathbf{D}_c^{-1/2}\mathbf{r}_b)'(\mathbf{D}_c^{-1/2}\mathbf{r}_a - \mathbf{D}_c^{-1/2}\mathbf{r}_b) \\
&= (\mathbf{D}_c^{-1/2}(\mathbf{r}_a - \mathbf{r}_b)'(\mathbf{D}_c^{-1/2}(\mathbf{r}_a - \mathbf{r}_b)) \\
&= (\mathbf{r}_a - \mathbf{r}_b)'\mathbf{D}_c^{-1/2}\mathbf{D}_c^{-1/2}(\mathbf{r}_a - \mathbf{r}_b)) \\
&= (\mathbf{r}_a - \mathbf{r}_b)'\mathbf{D}_c^{-1}(\mathbf{r}_a - \mathbf{r}_b) \\
&= D_{\chi^2}^2(\mathbf{r}_a, \mathbf{r}_b)
\end{aligned}$$

Desarrollo del modelo para AC III

- Con esta transformación, y recordando que $\mathbf{R} = \mathbf{D}_r^{-1}\mathbf{P}$, podemos definir una matriz de datos transformados:

$$\mathbf{Y} = \mathbf{R}\mathbf{D}_c^{-1/2} = \mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1/2}$$

```
Y <- R %*% sqrt(solve(Dc)); Y
```

```
          [,1]       [,2]       [,3]       [,4]
[1,] 0.5599684 0.32916588 0.17207326 1.42358780
[2,] 1.1739427 0.15624407 0.32670977 0.08663201
[3,] 1.0454353 0.02525497 0.54128908 0.02800604
[4,] 1.0717675 0.56701478 0.04940168 0.41918714
[5,] 0.5789169 0.79897537 0.42927065 0.31994776
[6,] 0.6229649 0.14647882 0.75296392 0.27072503
[7,] 0.1219258 0.37627586 1.18722658 0.07451148
```

- Notemos que esta matriz tiene términos de la forma:

$$y_{ij} = \frac{p_{ij}}{r_i c_j^{1/2}}$$

y no suman 1 ni por renglones ni por columnas. Representan las frecuencias relativas condicionadas por renglones, pero estandarizadas por su variabilidad, que depende de la raíz cuadrada de la frecuencia relativa de la columna.

Desarrollo del modelo para AC IV

- La matriz $\mathbf{Y}$ se puede pensar como una matriz usual de datos, en donde los renglones son observaciones y las columnas son variables, y el problema es encontrar una proyección que preserve la distancia relativa entre las observaciones. Este problema es el mismo de encontrar un vector $\mathbf{e}$ unitario tal que $\mathbf{Ye}$ tenga variabilidad máxima.
- Este es el mismo problema de optimización que el de componentes principales. Sin embargo, las 'observaciones' no tienen las mismas frecuencias relativas, por lo que hay que ponderarlas nuevamente por las frecuencias de cada renglón. Entonces el problema a resolver es maximizar la suma de cuadrados ponderada:

$$\mathbf{e}'\mathbf{Y}'\mathbf{D}_r\mathbf{Ye}$$

- Entonces:

$$\begin{aligned}
\mathbf{e}'\mathbf{Y}'\mathbf{D}_r\mathbf{Ye} &= \mathbf{e}'(\mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1/2})'\mathbf{D}_r(\mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1/2})\mathbf{e} \\
&= \mathbf{e}'\mathbf{D}_c^{-1/2}\mathbf{P}'\mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1/2}\mathbf{e} \\
&= \mathbf{e}'(\mathbf{D}_c^{-1/2}\mathbf{P}'\mathbf{D}_r^{-1/2})'(\mathbf{D}_r^{-1/2}\mathbf{P}\mathbf{D}_c^{-1/2})\mathbf{e} \\
&= \mathbf{e}'\mathbf{A}'\mathbf{Ae}
\end{aligned}$$

Desarrollo del modelo para AC V

donde $\mathbf{A} = \mathbf{D}_r^{-1/2}\mathbf{P}\mathbf{D}_c^{-1/2}$. Esta matriz tiene entradas estandarizadas de la forma

$$A_{ij} = \frac{p_{ij}}{\sqrt{r_{i\cdot}c_{\cdot j}}}$$

- Finalmente, la solución al problema estará dada por los eigenvalores e eingenvectores de $\mathbf{A}'\mathbf{A}$. Pero siempre el primer eigenvalor de esta matriz es $\lambda = 1$.

```
A <- sqrt(solve(Dr)) %*% P %*% sqrt(solve(Dc))
eigen(t(A)%*% A)

eigen() decomposition
$values
[1] 1.00000000 0.28358759 0.17010675 0.05878625

$vectors
           [,1]         [,2]       [,3]        [,4]
[1,] -0.6019601  0.475182733  0.6415934  0.01425899
[2,] -0.3413463 -0.005105499 -0.2966575 -0.89188059
[3,] -0.6529751 -0.691726534 -0.1067572  0.28938014
[4,] -0.3078154  0.543773580 -0.6992533  0.34728205
```

Desarrollo del modelo para AC VI

Teorema

La matriz $\mathbf{A}'\mathbf{A}$ tiene como máximo eigenvalor $\lambda_{\text{máx}} = 1$ y vector propio $\mathbf{D}_c^{1/2}$.

Demostración.

Si $\mathbf{a}$ es un vector propio de $\mathbf{A}'\mathbf{A}$, entonces:

$$\mathbf{D}_c^{-1/2}\mathbf{P}'\mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1/2}\mathbf{a} = \lambda\mathbf{a}$$

Multiplicando por la izquierda por $\mathbf{D}_c^{-1/2}$:

$$\mathbf{D}_c^{-1}\mathbf{P}'\mathbf{D}_r^{-1}\mathbf{P}\mathbf{D}_c^{-1/2}\mathbf{a} = \lambda\mathbf{D}_c^{-1/2}\mathbf{a}$$

Pero por las definiciones previas, de la parte roja: $\mathbf{D}_c^{-1}\mathbf{P}'\mathbf{1} = \mathbf{1}$ y la parte azul: $\mathbf{D}_r^{-1}\mathbf{P}\mathbf{1} = \mathbf{1}$.
Lo anterior implica que la matriz $\mathbf{D}_c^{-1}\mathbf{P}'\mathbf{D}_r^{-1}\mathbf{P}$ tiene un valor propio igual a 1 y entonces si hacemos $\mathbf{D}_c^{-1/2}\mathbf{a} = \mathbf{1}$ se tiene que $\mathbf{A}'\mathbf{A}$ tiene un valor propio con valor 1 y vector propio: $\mathbf{a} = \mathbf{D}_c^{1/2}$

□

Desarrollo del modelo para AC VII

- Finalmente, los scores corresponderán a los dos siguientes eigenvectores que no son la unidad, para los renglones:

```
Cr <- Y %*% eigen(t(A) %*% A)$vectors[,2:3]; Cr

            [,1]        [,2]
[1,]  0.91948857 -0.75219596
[2,]  0.37815399  0.61138660
[3,]  0.13744880  0.58588253
[4,]  0.70016098  0.22103785
[5,]  0.14805337 -0.13514502
[6,] -0.07835769  0.08654646
[7,] -0.72470277 -0.21224557
```

Estas coordenadas son la mejor representación de los renglones de $\mathbf{P}$ en un espacio de dos dimensiones.

- El mismo desarrollo se puede hacer para las columnas y la solución final queda en términos de los eigenvectores de la matriz $\mathbf{AA}'$. Las coordenadas obtenidas son la mejor representación en el espacio de dos dimensiones.

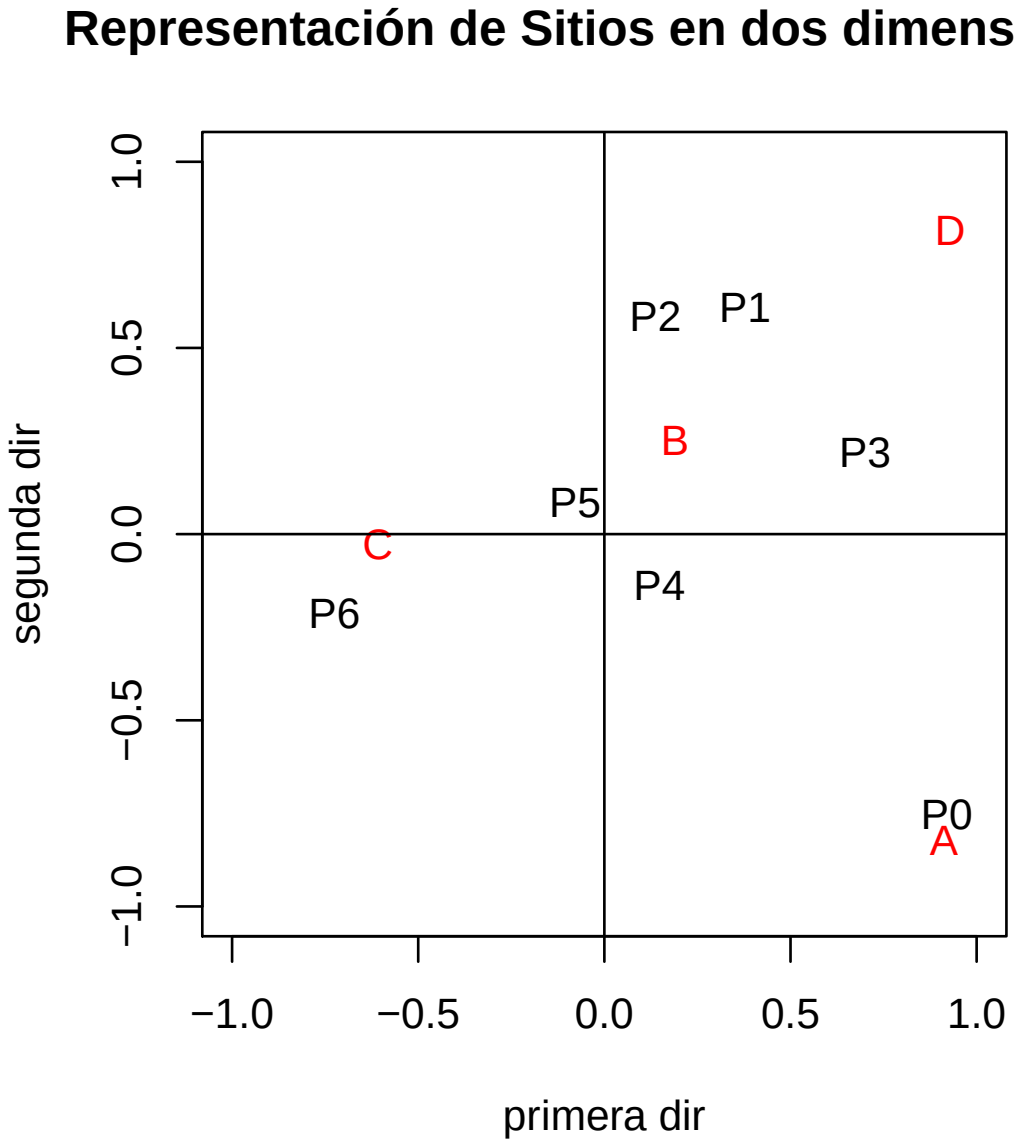
```
Cc <- t(Y) %*% eigen(A %*% t(A))$vectors[,2:3]; Cc

           [,1]       [,2]
[1,]  0.9120311 -0.8213983
[2,]  0.1896181  0.2543968
[3,] -0.6089868 -0.0270720
[4,]  0.9288030  0.8176481
```

Finalmente, los datos se pueden graficar de manera conjunta en los mismos ejes

Desarrollo del modelo para AC VIII

```
par(pty="s")
plot(Cr,main="Representación de Sitios en dos dimensiones",xlab="primera dir",ylab="segunda dir",
        xlim = c(-1,1), ylim = c(-1,1) ,pch="")
text(Cr,labels=row.names(X))
points(Cc,col="red",pch="")
text(Cc,labels=colnames(X),col="red")
abline(h=0); abline(v=0)
```

Desarrollo del modelo para AC IX



Interpretación gráfica:

- El resultado de este análisis es un par de gráficas bivariadas (o biplot):
  1. Una gráfica bivariada se basa en los primeros dos ejes principales de los renglones.
  2. La segunda se basa en los dos primeros dos ejes de las columnas.
- Las relaciones espaciales entre los dos conjuntos de categorías se puede estudiar usando las dos gráficas bivariadas, superimpuestas, mapeando sus respectivos ejes a ejes comunes. Las configuraciones de los puntos reflejan asociaciones entre los renglones y columnas de los datos:
  - Puntos renglones/columnas que se encuentran juntos indican renglones/columnas que tienen distribuciones condicionales (perfiles) similares a lo largo de las columnas/renglones.
  - Los puntos renglones que están cercanos a puntos columnas representan combinaciones que aparecen más frecuentemente de lo que se esperaría de un modelo de independencia: un modelo en el que las categorías renglones no se relacionan con las categorías columnas.

Análisis conjunto I

- Dado que el problema es simétrico, es conveniente resolverlo de manera simultánea para renglones y columnas.
- Para resolver simultáneamente el problema, podemos utilizar la descomposición en valor singular de la matriz $\mathbf{A}$ o de $\mathbf{A}'$. La descomposición en valor singular en $\mathbf{A}$ nos da:

$$\mathbf{A} = \mathbf{U}_k \mathbf{D}_k \mathbf{V}_k' = \sum_{i=1}^{k} \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i'$$

- La matriz $\mathbf{U}_k$ contiene las columnas de los vectores propios de $\mathbf{A}\mathbf{A}'$ y $\mathbf{V}$ los de $\mathbf{A}'\mathbf{A}$, y la matriz diagonal $\mathbf{D}_k$ tiene los valores singulares o raíces de los valores propios $\sqrt{\lambda_i}$.

Análisis conjunto II

```
svd(A)

$d
[1] 1.0000000 0.5325294 0.4124400 0.2424588

$u
           [,1]        [,2]        [,3]         [,4]
[1,] -0.3375744  0.58287068 -0.61565825  0.360032448
[2,] -0.3098883  0.22005447  0.45936749  0.002578257
[3,] -0.3853927  0.09947199  0.54746107  0.252330896
[4,] -0.1992303  0.26194471  0.10677294 -0.271619744
[5,] -0.4111132  0.11429735 -0.13471027 -0.795238966
[6,] -0.3919811 -0.05767706  0.08225336  0.307417153
[7,] -0.5283269 -0.71898370 -0.27188208  0.077533882

$v
           [,1]         [,2]       [,3]        [,4]
[1,] -0.6019601  0.475182733  0.6415934  0.01425899
[2,] -0.3413463 -0.005105499 -0.2966575 -0.89188059
[3,] -0.6529751 -0.691726534 -0.1067572  0.28938014
[4,] -0.3078154  0.543773580 -0.6992533  0.34728205
```

- La matriz $\mathbf{A}$ se puede aproximar, para alguna $h \leqslant k$ (típicamente $h = 2$) con la matriz $\hat{\mathbf{A}}_h = \mathbf{U}_h \mathbf{D}_h \mathbf{V}_h'$, tomando h columnas de la descomposición. Esta aproximación equivale a aproximar la tabla de contingencias observada con

$$\hat{\mathbf{P}}_h = \mathbf{D}_r^{1/2} \hat{\mathbf{A}}_h \mathbf{D}_c^{1/2}$$

Simplificación I

- Para eliminar el valor propio correspondiente a la unidad, se puede reemplazar la matriz $\mathbf{P}$ por $\mathbf{P} - \hat{\mathbf{P}}_e$, donde

$$\hat{\mathbf{P}}_e = \frac{1}{\mathrm{n}}\mathbf{r}\mathbf{c}'$$

- Este ajuste elimina el supuesto caso de independencia de las variables y la matriz ajustada $\mathbf{P} - \hat{\mathbf{P}}_e$ tiene rango $\mathrm{k} - 1$. Con esta matriz ajustada se realiza el cálculo de la matriz $\mathbf{A}_{\mathrm{aj}}$
- **Nota: es importante realizar el cálculo conjunto para evitar problemas de signos.**

Concepto de Inercia

- La *inercia total* es una medidad de la variación en los datos de frecuencias y se define como:

$$\text{tr}(\mathbf{A}_{aj}\mathbf{A}'_{aj}) = \sum_i \sum_j \frac{(p_{ij} - r_i c_j)^2}{r_i c_j} = \sum_{k=1}^{J-1} \lambda_k$$

  donde $\sqrt{\lambda_i}$ son los valores singulares de la descomposición de $\mathbf{A}_{aj}$
- La inercia de la solución con K componentes es $\sum_{k=1}^{K} \lambda_k$, todo lo mismo que en PCA.

Herramientas en R I

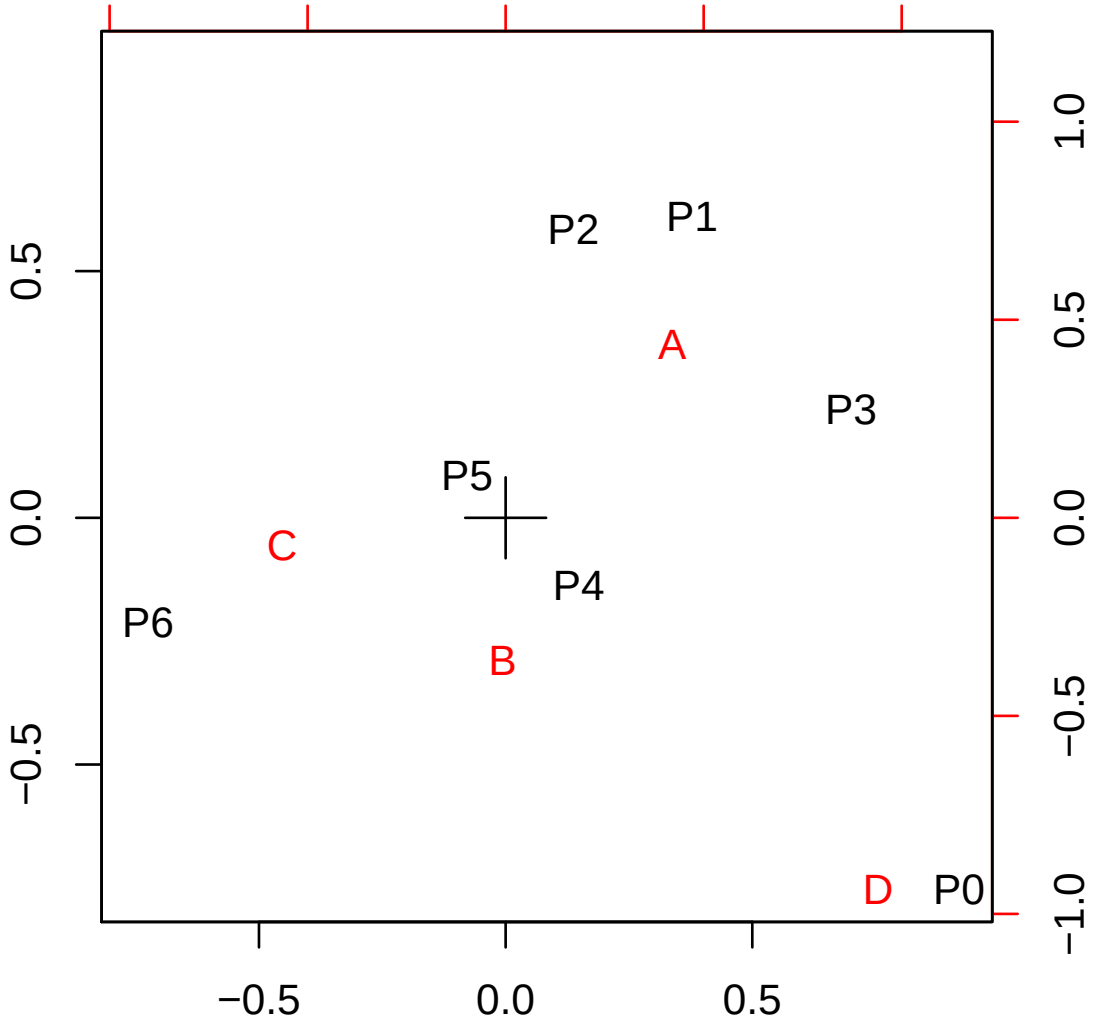
- la función básica `corresp` en el paquete `MASS`
- El paquete `ca`, escrito por Nenadic y Greenacre: inlcuye AC simple, múltiple y conjunto.
- `FactoMineR` para el análisis y `factoextra` para la visualización.
- La función `dudicoa` en el paquete `ade4`
- La función `epCA` en el paquete `ExPosition`

Ejemplo de aplicación I

- Para mostrar la aplicación, usando el paquete `MASS`:

```
library(MASS)
m1 <- corresp(X,nf=2); m1

First canonical correlation(s): 0.5325294 0.4124400

 Sitio scores:
         [,1]       [,2]
P0  1.7266437 -1.8237706
P1  0.7101091  1.4823650
P2  0.2581055  1.4205279
P3  1.3147837  0.5359273
P4  0.2780191 -0.3276719
P5 -0.1471425  0.2098401
P6 -1.3608690 -0.5146096

 Tipo scores:
         [,1]       [,2]
A  0.78939242  1.0658404
B -0.01495695 -0.8690807
C -1.05934601 -0.1634935
D  1.76655743 -2.2716644

#valores de la inercia, son los cuadrados de las correlaciones canónicas.
m1$cor^2 #recordar que este caso da los valores singulares que son la raíz de los eigenvalores

[1] 0.2835876 0.1701068

plot(m1)
```

Ejemplo de aplicación II



Ejemplo de aplicación III

En este caso, no sabemos cuánto representa la inercia respecto al total, pues no tenemos todos los eigenvalores.

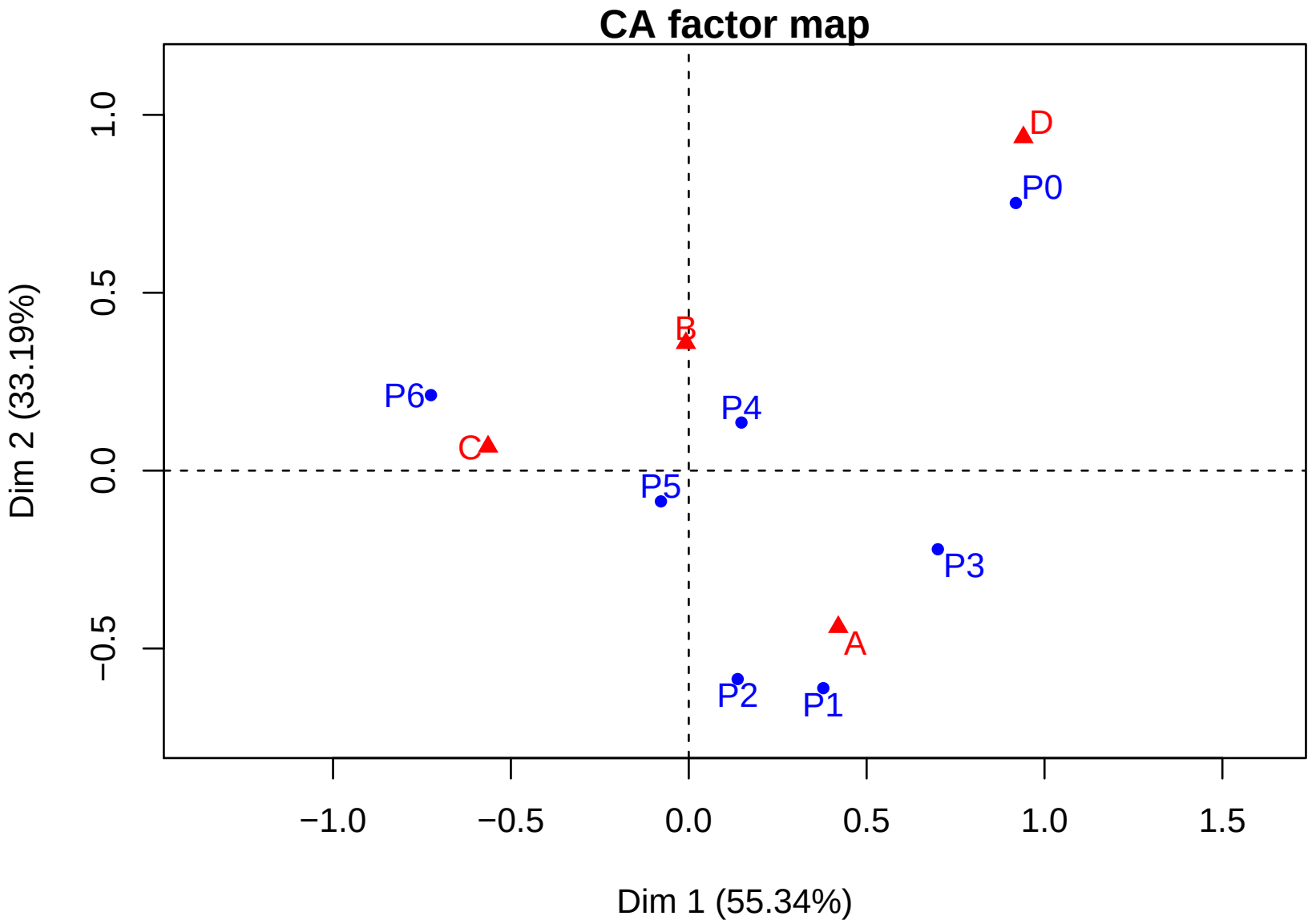
- Utilizando `FactoMineR` y `factoextra`:

```
library(FactoMineR)
library(factoextra)
```

```
Loading required package: ggplot2
Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
m2 <- CA(X, ncp=2, graph=T)
```

Ejemplo de aplicación IV



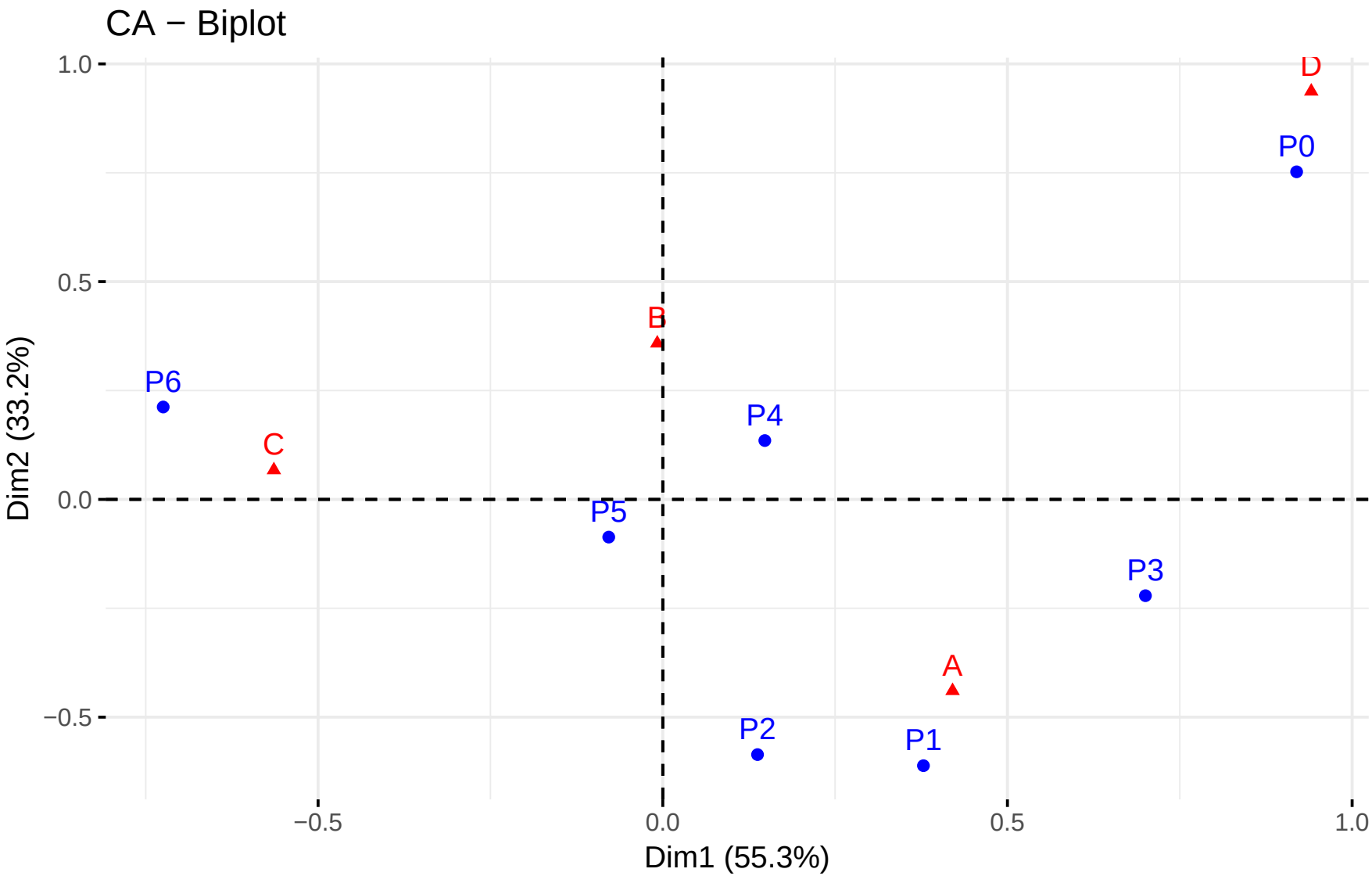
Ejemplo de aplicación V

```
summary(m2)
```

```
Call:
CA(X = X, ncp = 2, graph = T)

The chi square of independence between the two variables is equal to 400.2473 (p-value =  8.126171e-74 ).

Eigenvalues
                       Dim.1   Dim.2   Dim.3
Variance               0.284   0.170   0.059
% of var.             55.336  33.193  11.471
Cumulative % of var.  55.336  88.529 100.000

Rows
     Iner*1000     Dim.1     ctr    cos2     Dim.2     ctr    cos2
P0 |   168.442 |   0.919  33.974   0.572 |   0.752  37.904   0.383 |
P1 |    49.628 |   0.378   4.842   0.277 |  -0.611  21.102   0.723 |
P2 |    57.532 |   0.137   0.989   0.049 |  -0.586  29.971   0.886 |
P3 |    25.735 |   0.700   6.862   0.756 |  -0.221   1.140   0.075 |
P4 |    43.968 |   0.148   1.306   0.084 |   0.135   1.815   0.070 |
P5 |     7.650 |  -0.078   0.333   0.123 |  -0.087   0.677   0.150 |
P6 |   159.525 |  -0.725  51.694   0.919 |   0.212   7.392   0.079 |

Columns
     Iner*1000     Dim.1     ctr    cos2     Dim.2     ctr    cos2
A  |   134.069 |   0.420  22.580   0.478 |  -0.440  41.164   0.522 |
B  |    61.739 |  -0.008   0.003   0.000 |   0.358   8.801   0.242 |
C  |   142.554 |  -0.564  47.849   0.952 |   0.067   1.140   0.014 |
D  |   174.118 |   0.941  29.569   0.482 |   0.937  48.896   0.478 |
```

```
fviz_ca_biplot(m2)
```

Ejemplo de aplicación VI



Escalamiento Multidimensional

Introducción

- El escalamiento multidimensional (MDS) resuelve el problema de representar objetos espacialmente, a través de construir una configuración de puntos en alguna dimensión menor como $\mathbb{R}^k$, para $k = 1, 2, 3$, utilizando la información disponible sobre la *similitud* o *disimilitud* de los objetos, de tal manera que las proximidades entre los items se 'parezcan' lo más posible a las similitudes o disimilitudes originales.
- MDS también se considera una técnica exploratoria de análisis multivariado, así como una técnica de reducción de dimensión.
- Para medir cómo la configuración "ajustada" se apega a la configuración "real", se introduce una medida de cercanía llamada *stress*.
- Fue creado originalmente en 1952 por Warren S. Torgeson y posteriormente desarollado y extendido por Joseph Kruskal.

Tipos de soluciones en MDS I

En multiescalamiento dimensional (MDS) hay dos tipos posibles de soluciones:

- **MDS-no métrico**: cuando sólo se utiliza información ordinal (basada en los rangos) de las similitudes originales.
- **MDS-métrico**: se utilizan las similitudes (o distancias) originales para obtener una representación geométrica en k dimensiones. Esta versión también se conoce como análisis de coordenadas principales.

Características generales de la similitud

- El concepto de similitud es bastante general y puede incluso ser subjetiva. Se puede definir para varios tipos de datos: cuantitativos, binarios, nominales ordinales o mixtos. Por ejemplo:
  - La presencia o ausencia de ciertas características se pueden usar como medida de similitud: los objetos serán más similares si comparten más características
  - 12 marcas de yogurth evaluadas por 10 jueces en nueve variables. Los yogurths son presentados en pares a los panelistas a los que se les pide evaluar que tan similares son las dos muestras en una linea de escala descriptiva de 15cm.
  - La similitud entre códigos Morse puede medirse como el porcentaje de veces que las personas confunden las sucesiones de símbolos después de escucharlos en una sucesión rápida.
- Una función de similitud puede ser simétrica, no negativa y creciente conforme los objetos son más similares. Se considera que una medida de similitud es inversamente proporcional a una medida de distancia. La distancia puede ser considerada como una medida de disimilitud.

Formalizando Similitud I

Definición

- Una matriz de similitud $\mathbf{C}$ es simétrica $(\mathbf{C}' = \mathbf{C})$ y $0 \leqslant c_{ij} \leqslant c_{ii} \quad \forall i, j$
- Una matriz de disimilitud o distancia $\mathbf{D}$ es simétrica $(\mathbf{D}' = \mathbf{D})$ y $d_{ii} = 0, \quad d_{ij} \geqslant 0 \quad i \neq j.$
- Con frecuencia se intercambian los coeficientes de similitud a distancia y viceversa. Posibles transformaciones incluyen:
  - $d_{ij} = c - c_{ij}$ para alguna constante c.
  - $c_{ij} = \frac{1}{1+d_{ij}}$
  - La transformación estándar: $d_{ij} = (c_{ii} - 2c_{ij} + c_{jj})^{1/2}$

A continuación consideraremos varios ejemplos de medidas, tomando en cuenta el tipo de variable (discreta, continua, binaria) y las escalas de medición (nominal, ordinal, de intervalo, de razón). Algunas de estas ya las hemos definido antes:

- Continuas:
  - **Distancia Euclideana:** La distancia usual para variables numéricas: $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})'(\mathbf{x} - \mathbf{y})}$

Formalizando Similitud II

- **Distancia de Mahalanobis:** Los datos se ponderan por su variabilidad: $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})'\mathbf{S}^{-1}(\mathbf{x}-\mathbf{y})}$, aunque no siempre se conocen los grupos de antemano y por lo tanto no se puede estimar $\mathbf{S}$ (como en conglomerados).
- **Norma supremo:** $d(\mathbf{x}, \mathbf{y}) = max|x_i - y_i|$
- **Distancia de Minkowski:** $d_m(\mathbf{x}, \mathbf{y}) = [\sum_{i=1}^{p} |x_i - y_i|^m]^{1/m}$. Cuando $m = 1$ es la 'distancia Manhattan'.

- No negativas:
  - **métrica de Canberra:** $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{x_i + y_i}$.
  - **coeficiente de Czekanowski:** $d(\mathbf{x}, \mathbf{y}) = 1 - \frac{2\sum_{i=1}^{p} mín(x_i, y_i)}{\sum_{i=1}^{p} x_i + y_i}$
- Binarias:
  - **Presencia o ausencia de características:** $\sum_{i=1}^{p}(x_{ij} - x_{kj})^2$, donde:

$$(x_{ij} - x_{kj})^2 = \begin{cases} 0 & x_{ij} = x_{kj} = 1 \text{ o } x_{ij} = x_{kj} = 0 \\ 1 & x_{ij} \neq x_{kj} \end{cases}$$

donde estamos comparando la i—ésima variable de los items j y k.

Fórmula de Gower I

- Siempre es posible construir similaridades a partir de distancias, con las transformaciones mencionadas antes.
- Sin embargo, disimilitudes que son distancias reales no siempre pueden ser construidas a partir de similitudes. Esto sólo se puede hacer si la matriz $\mathbf{C}$ es definida positiva (Gower, 1971).
- Con la condición anterior, y con la similitud máxima escalada de tal forma que $\tilde{c}_{ii} = 1$,

$$d_{ik} = \sqrt{2(1 - \tilde{c}_{ik})}$$

Esta es la fórmula de Gower, que tiene propiedades de distancia.

Funciones en R para distancias I

- En R hay algunas funciones para calcular matrices de distancias a partir de datos:
  - la función `dist` que puede calcular a partir de una matriz numérica o data.frame las distancias: euclidean, max, manhattan, canberra, binary o minkoswi:

```
x <- matrix(rnorm(100),nrow=5)
dist(x) #euclideana por default
```

```
         1        2        3        4
2 4.461962
3 5.334528 6.688427
4 5.697314 5.833231 7.076868
5 6.600100 6.826245 6.968016 5.759782
```

```
dist(x,"canberra")
```

```
         1        2        3        4
2 13.35875
3 10.46372 15.87198
4 15.09534 15.72441 15.97905
5 14.58323 13.88475 15.01504 12.85311
```

```
dist(x,"binary") #revisar definición de binary
```

```
  1 2 3 4
2 0
3 0 0
4 0 0 0
5 0 0 0 0
```

Funciones en R para distancias II

- La función `daisy` que calcula matrices de disimilaridades en donde las variables pueden ser de tipos mezclados. En este caso, aplica una generalización de la transformación de Gower que se mencionó arriba:

Funciones en R para distancias III

```
library(cluster)
data(flower) #características de 18 flores,
str(flower)
```

```
'data.frame': 18 obs. of  8 variables:
 $ V1: Factor w/ 2 levels "0","1": 1 2 1 1 1 1 1 2 2 ...
 $ V2: Factor w/ 2 levels "0","1": 2 1 2 1 2 2 1 1 2 2 ...
 $ V3: Factor w/ 2 levels "0","1": 2 1 1 2 1 1 1 2 1 1 ...
 $ V4: Factor w/ 5 levels "1","2","3","4",..: 4 2 3 4 5 4 4 2 3 5 ...
 $ V5: Ord.factor w/ 3 levels "1"<"2"<"3": 3 1 3 2 2 3 3 2 1 2 ...
 $ V6: Ord.factor w/ 18 levels "1"<"2"<"3"<"4"<..: 15 3 1 16 2 12 13 7 4 14 ...
 $ V7: num  25 150 150 125 20 50 40 100 25 100 ...
 $ V8: num  15 50 50 50 15 40 20 15 15 60 ...
```

```
round(daisy(flower,metric="gower"),2)
```

```
Dissimilarities :
      1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
2  0.89
3  0.53 0.51
4  0.35 0.55 0.57
5  0.41 0.62 0.37 0.64
6  0.23 0.66 0.30 0.42 0.34
7  0.29 0.60 0.49 0.34 0.42 0.19
8  0.42 0.46 0.60 0.30 0.47 0.57 0.41
9  0.58 0.43 0.45 0.81 0.33 0.51 0.59 0.64
10 0.61 0.45 0.47 0.56 0.38 0.41 0.59 0.66 0.43
11 0.33 0.71 0.60 0.65 0.39 0.48 0.57 0.50 0.43 0.39
12 0.43 0.59 0.60 0.51 0.50 0.52 0.64 0.42 0.42 0.38 0.26
13 0.52 0.52 0.54 0.75 0.29 0.45 0.53 0.58 0.22 0.36 0.34 0.23
14 0.29 0.59 0.61 0.37 0.52 0.37 0.50 0.46 0.44 0.36 0.28 0.16 0.38
15 0.62 0.39 0.53 0.55 0.46 0.51 0.33 0.45 0.25 0.42 0.48 0.43 0.32 0.44
16 0.69 0.36 0.62 0.34 0.73 0.51 0.44 0.64 0.65 0.35 0.74 0.61 0.59 0.46 0.39
17 0.78 0.19 0.58 0.42 0.69 0.59 0.52 0.47 0.61 0.31 0.70 0.56 0.55 0.54 0.35 0.17
18 0.46 0.45 0.72 0.44 0.48 0.64 0.47 0.14 0.52 0.81 0.54 0.55 0.57 0.57 0.51 0.78 0.61

Metric :  mixed ;  Types = N, N, N, N, O, O, I, I
Number of objects : 18
```

Medidas de bondad de ajuste I

Dada una configuración de n puntos en $\mathbb{R}^k$, $\mathbf{X}_{n\times k}$ con distancias entre puntos dadas por $\hat{d}_{ij}$ y una matriz de distancia arbitraria $\mathbf{D} = (d_{ij})$, se utilizan las siguientes funciones como medidas de la bondad de ajuste de las configuraciones a las configuraciones originales:

- La función:

$$S_k(\mathbf{X}) = \sqrt{\frac{\sum_{i<j}(d_{ij}^{(k)} - \hat{d}_{ij}^{(k)})^2}{\sum_{i<j}\hat{d}_{ij}^{(k)2}}}$$

  se conoce como la función *stress* y fue propuesta por Joseph Kruskal.

- A la función

$$SS_k(\mathbf{X}) = \sqrt{\frac{\sum_{i<j}(d_{ij}^{(k)2} - \hat{d}_{ij}^{(k)2})^2}{\sum_{i<j}\hat{d}_{ij}^{(k)4}}}$$

  se conoce como la función *sstress*, propuesta por Takane y otros.

Solución métrica I

- Consideren n puntos $P_1 = \mathbf{x}_1, \ldots, P_n = \mathbf{x}_n$ en $\mathbb{R}^p$. Entonces la distancia euclídea entre los puntos P_i y P_j está dado por d_{ij}.
- La matriz de *productos interiores* $\mathbf{B}$ tiene componentes $b_{ij} = \mathbf{x}_i'\mathbf{x}_j$. La solución clásica usa $\mathbf{D}$ para encontrar $\mathbf{B}$ y entonces de $\mathbf{B}$ se obtienen los puntos P_i.
- Una posible configuración in $\mathbb{R}^p$ se obtiene de $\mathbf{VL}$ donde:
  - $\mathbf{V}$ es la matriz con los primeros p eigenvectores of $\mathbf{B}$
  - $\mathbf{L}$ es una matriz diagonal con los primeros p eigenvalues de $\mathbf{B}$.
- Si el problema comienza con una matriz de similaridad, entonces se puede usar la transformación estándar a una matriz de distancia.

Un algoritmo para el método clásico es el siguiente:

1. de $\mathbf{D}$ se construye $\mathbf{A} = (-\frac{1}{2}d_{ij}^2)$
2. Calcula $\mathbf{B} = \mathbf{HAH}$ con $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{11}'$.
3. Elige un número apropiado de dimensiones k, usando el cociente de sumas de eigenvalores, o graficando la función stress o sstress contra k.

Solución métrica II

4. Encuentra los k eigenvalores de $\mathbf{B}$, $\lambda_1 \geqslant, \ldots, \geqslant \lambda_k$ con correspondientes eigenvectores $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_k)$, escalados para tener norma unitaria.
5. Las coordenadas requeridas de los puntos P_i están dados por las columnas de $\mathbf{X} = \mathbf{V}\mathbf{L}^{1/2}$ donde $\mathbf{L}$ es la matriz diagonal con los k primeros eigenvalores más grandes de $\mathbf{B}$.

Bondad de ajuste I

- Usando $\hat{d}_{ij}^{(k)}$ se mueven los puntos para obtener una mejor configuración por un procedimiento de minimización aplicada a S_k. Se espera que una nueva configuración tendrá nuevos valores $d's$ y menor estress. Los criterios propuestos por Kruskal pata evaluar el stress son los siguientes:

S_k	Ajuste
20 %	Pobre
10 %	Débil
5 %	Bueno
2.5 %	Excelente
0 %	Perfecto

En el caso de la función $SS_k \in [0, 1]$, se busca que tenga valores menores a 0.1.

Solución no métrica I

Se tienen N items y hay $M = \binom{N}{2}$ similitudes entre pares de items. En una primera propuesta, se supone que no hay empates entre las similitudes. Este supuesto se puede eliminar haciendo algunas modificaciones al algoritmo. Para cada dimensión k se puede obtener el stress mínimo:

1. Se ordenan las similitudes o las distancias:

$$c_{i_1 j_1} < \quad \dots \quad < c_{i_M j_M} \tag{1}$$

$$d^{(k)}_{i_1 j_1} > \quad \dots \quad > d^{(k)}_{i_M j_M} \tag{2}$$

   Si no es posible calcular similitudes, usar rangos.

2. Con una configuración de prueba en $\mathbb{R}^k$, se calculan los valores d^k_{ij} y $\hat{d}^{(k)}_{ij}$ Usualmente éstos últimos se calculan a través de ciertos modelos de regresión. No son propiamente distancias, sólo números de referencia para evaluar la monotonía de (2)
3. Se grafica el mínimo S_k versus k y se elige el mejor número de dimensiones examinando donde se forme el codo respectivo.