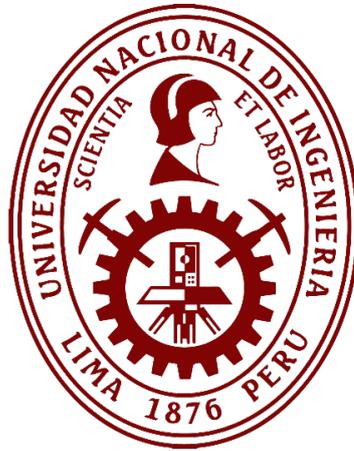


UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE CIENCIAS



TESIS

**“USO DE COEFICIENTES DE ASOCIACIÓN PARA LA
REPRESENTACIÓN GRÁFICA DE DATOS
PRESENCIA/AUSENCIA”**

**PARA OBTENER EL GRADO ACADÉMICO DE MAESTRO EN
CIENCIAS EN MATEMÁTICA APLICADA**

ELABORADA POR:

JOSÉ ANDRÉS ALCALDE SOSA

ASESOR:

DR. OSWALDO JOSÉ VELÁSQUEZ CASTAÑÓN

ASESOR EXTERNO:

DR. SERGIO CAMIZ

LIMA–PERÚ

2019

Dedicatoria

A mis padres Victor y Nelsa,
a mis hermanos Manuel y Teresa,
a mis sobrinos Piero, Harumi y Emily.

AGRADECIMIENTOS

Manifiesto mi gratitud al Director del Instituto de Matemática y Ciencias Afines - IMCA - de la Universidad Nacional de Ingeniería, Dr. Félix Escalante del Águila, y al conductor del Programa “ FORTALECIMIENTO DEL PROGRAMA DE MAESTRÍA EN UNIVERSIDADES PERUANAS”, Dr. Eladio Ocaña Anaya, por haberme hecho parte de este Programa financiado por el Estado Peruano a través del Fondecyt - Ciencia Activa.

A mis padres, Victor y Nelsa, por sus sabios consejos y por el amor tan grande que me han dado.

A mis hermanos por su paciencia y comprensión.

A mis asesores Sergio Camiz y Oswaldo Velásquez, que sin su valiosa ayuda no se hubiera podido realizar este trabajo.

Además quiero agradecer a todos los que hicieron posible que este trabajo se concluyera.

Resumen

El objetivo de este trabajo es mostrar la calidad algunos coeficientes de asociación para datos de presencia/ausencia, para esto se usa una tabla de vegetación en los cuales se van comparando los resultados proporcionados por los diferentes coeficientes utilizando uno o mas tratamientos de análisis de datos diferentes.

Índice de figuras

3.1. Representación en dos dimensiones usando el comando FactoMineR	33
4.1. Representación en dos dimensiones usando el comando cmdscale	47
4.2. Representación en dos dimensiones usando el comando wcmdscale considerando los valores propios positivos	50
4.3. Representación en dos dimensiones usando el comando wcmdscale considerando los valores propios negativos	51
4.4. Representación en dos dimensiones usando el comando pco del paquete ecodist	53
4.5. Representación en dos dimensiones usando el comando pco del paquete labdsv	54
4.6. Representación en dos dimensiones usando el comando pcoa del paquete ape	58
4.7. Representación en dos dimensiones usando el comando dudi.pco del paquete ade4	59
4.8. Comparando las representaciones en dos dimensiones usando las coordenadas principales	61
5.1. Regresión isotónica para los datos del ejemplo	64
5.2. Representación en dos dimensiones usando el comando smacofSym del paquete smacof	72
5.3. Representación en dos dimensiones usando el comando isoMDS del paquete MASS	75
5.4. Representación en dos dimensiones usando el comando sammon del paquete MASS	77
5.5. Representación en dos dimensiones usando el comando metaMDS del paquete vegan	82

6.1. ACoP vs Kruskal (sammon)	95
6.2. ACoP vs Kruskal (metaMDS)	96
6.3. ACoP vs ACP	97
6.4. Braun-Blanquet vs Simpson	99
6.5. Braun-Blanquet vs Russell & Rao	100
B.1. Representación en dos dimensiones usando las componentes principales del comando <code>dudi.pco</code> del paquete <code>ade4</code>	119

Índice de cuadros

2.1. Tabla de frecuencias	6
2.2. Tabla 2	9

Índice general

Resumen	III
1. Introducción	1
2. El tratamiento de datos presencia/ausencia	5
2.1. Coeficientes de asociación para datos presencia/ausencia	6
2.2. La tabla de datos de Ellenberg	9
3. Representación gráfica de datos multidimensionales	10
3.1. Entorno Geométrico	10
3.2. Datos, nubes y métricas	15
3.2.1. Datos	15
3.2.2. Nubes	16
3.2.3. Métricas	17
3.3. Análisis de componentes principales (ACP)	20
3.3.1. Búsqueda del punto en \mathbb{R}^p más cercano a \mathcal{N}	21
3.3.2. Cálculo de I_g , la inercia de \mathcal{N} relativa a su centroide	22
3.3.3. Búsqueda del subespacio q -dimensional más cercano a la nube \mathcal{N}	23
3.3.4. Búsqueda de los ejes principales	25
3.3.5. Búsqueda del plano P del subespacio W_q más cercano a \mathcal{N} . .	29
3.3.6. Calidad de la representación de las unidades y contribución de una unidad a la construcción de un eje	31
3.4. ACP en el caso presencia/ausencia	32
3.5. ACP con técnicas R y Q	32

4. Análisis de coordenadas principales (ACoP)	40
4.1. Cmdscale (stats)	46
4.2. wcmdscale (vegan)	48
4.3. pco (ecodist)	51
4.4. pco (labdsv)	53
4.5. pcoa (ape)	54
4.6. dudi.pco (ade4)	58
5. Representación óptima de distancias	62
5.1. Kruskal	62
5.1.1. Minimizando S con respecto a las disparidades	63
5.1.2. Una configuración con estrés mínimo	66
5.1.3. Técnica iterativa de Kruskal	67
5.2. smacofSym (smacof)	70
5.3. isoMDS (MASS)	73
5.4. sammon (MASS)	75
5.5. metaMDS (vegan)	77
6. Análisis de Procrustes (AP)	83
6.1. Comparando formas	83
6.2. La transformación de Procrustes	87
6.3. Búsqueda de la traslación a	87
6.4. Búsqueda del escalamiento s	88
6.5. Búsqueda de la rotación T	89
6.6. Análisis de Procrustes	91
6.7. Problema ortogonal de Procrustes	92
6.8. Prueba de Procrustes	93
6.9. Aplicación	94
6.9.1. Considerando el coeficiente de asociación de Braun-Blanquet	94
6.9.2. Considerando diferentes coeficientes de asociación	97
7. Conclusiones	101
A. Método de ACP en R	103
A.1. PCA (FactoMineR)	103

B. Métodos de ACoP en R	105
B.1. Cmdscale (stats)	105
B.2. wcmdscale (vegan)	106
B.3. pco (ecodist)	108
B.4. pco (labdsv)	109
B.5. pcoa (ape)	111
B.6. dudi.pco (ade4)	113
C. Métodos de Kruskal en R	120
C.1. smacofSym (smacof)	120
C.2. isoMDS (MASS)	122
C.3. sammon (MASS)	123
C.4. metaMDS (vegan)	124
D. Método de Procrustes en R	126
D.1. procrustes (vegan)	126
Referencias	128

Notaciones

\mathbf{v}' transpuesta del vector \mathbf{v} .

Δu línea recta generada por el vector u .

\mathcal{N} nube de puntos.

g centroide de una colección de puntos

Capítulo 1

Introducción

Los datos son palabras o números que sirven como referencia de algo (DNI, género, talla, peso, ingresos mensuales, etc), estamos rodeados de datos y están presentes desde que uno nace (a un recién nacido lo pesan y miden).

Hay datos tan simples que se pueden recolectar como las ventas de la bodega de la esquina, pasando por las preferencias de clientes, el número promedio de clientes atendidos por hora en una entidad financiera, la producción mensual de determinado producto de una empresa, hasta los datos mas complejos, como los necesarios para obtener la foto de un agujero negro, de ahí su importancia porque los vemos en nuestro día a día hasta eventos de gran significado científico.

Pero no solo basta con recolectar los datos, por que a veces nos podemos saturar de datos innecesarios que no sirven para nuestro propósito, para esto es necesaria la información. La información se puede definir como un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quién debe tomar decisiones, al disminuir su incertidumbre.

El análisis de datos es una herramienta muy importante en nuestros días, no fue hasta la aparición de la computadora, que ayudo al almacenamiento de datos y procesamiento de la información que el análisis de datos se ha desarrollado de manera vertiginosa, surgiendo diferentes técnicas, llegando hasta la ciencia de datos (data science) y el big data.

Una forma amigable de representar estos datos para poder explicar las conclusiones es de manera visual, mostrar los resultados en un gráfico facilita el entendimiento de manera significativa. Morfológicamente podemos visualizar objetos bidimensionales y tridimensionales.

Pero que pasa cuando se tienen demasiadas características para un conjunto de objetos, en este caso sería imposible plasmar la información en una gráfica. ¿Cómo representar los resultados de tal forma que no se pierda la información obtenida?

Así surgió el escalamiento multidimensional, el cual es un conjunto de técnicas que permite una representación de los datos en menor dimensión (de preferencia bidimensional) de tal forma que no se pierda la información original.

El análisis de componentes principales (ACP) es probablemente una de las técnicas más antigua y mejor conocida del análisis multivariado. Fue introducido por primera vez por Pearson (1901) y desarrollado independientemente por Hotelling (1933). Como muchos métodos multivariados, no fue ampliamente utilizado hasta la llegada de la computadora, pero ahora se encuentra en cada software estadístico Jolliffe (2002).

En el ACP se buscan subespacios de menor dimensión que represente la nube de puntos original, usando la inercia como medida de proximidad. El centroide es el punto más cercano a la nube \mathcal{N} (pues hace que la inercia sea mínima). Para buscar el primer eje principal se desea hallar una línea recta (generada por un vector u_1) que contiene al centroide g sobre la cual la inercia de la nube de puntos sea máxima, para esto se emplea los multiplicadores de Lagrange sujeto a la condición de que el vector debe ser unitario. Para determinar el segundo eje principal se resuelve el mismo problema agregando la condición de que el segundo vector debe ser ortogonal a u_1 .

Se entiende por similitud a que tan parecidos son dos objetos, el coeficiente de asociación expresa la similitud entre objetos, este coeficiente tiene un rango que depende de como se haya definido, en el presente trabajo se da una lista de distintos coeficientes de similitud y sus respectivos rangos. Se puede entender por disimilitud entre dos objetos como qué tan «diferentes» son dichos objetos.

El análisis de coordenadas principales (ACoP) es un método de escalamiento u ordenamiento, fue popularizado por Gower (1966). En este caso el punto de partida es

una matriz \mathbf{T} $n \times n$ de (dis)similitudes, supongamos que dicha matriz es semidefinida positiva simétrica, el ACoP intenta determinar el mejor ajuste q -dimensional ($q < n$) como una aproximación de la representación n -dimensional, por lo tanto consiste de dos etapas, ambas implican calcular los autovalores y autovectores de matrices $n \times n$:

- (i) Calcular los autovectores $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n$ de \mathbf{T} , normalizados para tener longitudes iguales a sus respectivos valores propios, y representar las n observaciones como puntos en un espacio n -dimensional con coordenadas c_{ij} para la i -ésima observación en la j -ésima dimensión.
- (ii) Determinar los componentes principales para el conjunto de datos en n dimensiones definidos en (i), y calcular las coordenadas de las n observaciones con respecto a las componentes principales.

Una de las áreas que ayudó al desarrollo de estas técnicas fue la psiquiatría, pues se buscaba estudiar los datos del comportamiento de los pacientes. En 1964 dos artículos de Kruskal aparecieron en la revista *Psychometrika*. Kruskal trabajó con (dis)similitudes y distancias entre unidades y buscó una relación monótona entre ellas, introdujo la bondad de ajuste y el estrés. En el método de Kruskal el objetivo es que el estrés sea mínimo; estos valores se pueden encontrar en Kruskal (1964a) y el método numérico se puede apreciar en Kruskal (1964b). Según Casas (2012) en el método de Kruskal el estrés no es propiamente una medida de la bondad del ajuste, sino una medida de la no bondad del ajuste, por lo que se desea que sea mínimo.

En análisis de datos, el análisis de Procrustes es un método que proporciona el mejor ajuste de un conjunto de puntos, llamado *nube de prueba*, a un conjunto dado, llamado *nube objetivo*, de acuerdo a transformaciones rígidas.

En este trabajo inicialmente se da una idea de similitud y disimilitud, luego una lista de coeficientes de similitud.

Después sigue un marco teórico en el cual inicialmente se describe el entorno en el que se trabaja, los espacios vectoriales reales, dentro de los cuales se consideran conceptos como independencia lineal, base, métrica, norma, producto interno, proyección ortogonal, espacio afín. Luego se ve algunas definiciones y propiedades respecto a lo que son los datos (unidades, tipos de características, nubes). Posteriormente se

desarrolla la parte matemática del ACP y se aplica a la muestra considerada.

La muestra considerada es la información obtenida de 25 parcelas de una pradera situada en el valle del Danubio al sur de Ulm, Mueller-Dombois y Ellenberg (1974), es nuestro caso vamos a analizar la presencia/ausencia de 72 especímenes en dichas parcelas, y se ha considerando el coeficiente de asociación de Braun-Blanquet.

Se desarrolla la parte matemática del ACoP y se aplican los distintos programas a la muestra. Se da una breve explicación del método de Kruskal, así como el algoritmo y se aplican los programas a la muestra. Después se detalla la parte teórica del análisis de Procrustes y se halla la matriz de Procrustes, inicialmente entre las distintas salidas para el coeficiente de Braun-Blanquet, tomando como matriz objetivo la salida del ACoP y luego para diferentes coeficientes de asociación (Simpson y Russell & Rao).

La programación se ha realizado con el software R.

Capítulo 2

El tratamiento de datos presencia/ausencia

Por similitud se entiende que tan «parecidos» son dos objetos; a mayor similitud entre dos objetos estos serán mas parecidos. Con esta idea podemos tratar de entender la disimilitud entre dos objetos como qué tan «diferentes» son dichos objetos. Si tenemos dos objetos con disimilitud pequeña ellos serán parecidos, mientras que si la disimilitud es grande ellos serán distintos.

Generalmente, las similitudes están acotadas en el rango de cero a uno; un aumento de la similitud implica un aumento de la semejanza entre objetos, y toda similitud de un objeto consigo mismo debería ser igual al máximo valor posible, es decir, uno. Las disimilitudes en cambio disminuyen con un aumento del parecido, son no negativas y la disimilitud de un elemento consigo mismo es cero. Tanto las matrices de similitudes como las de disimilitudes son simétricas.

Las similitudes se pueden transformar en disimilitudes: por ejemplo, sea s la similitud y d la disimilitud, tenemos:

$$d = s_{max} - s, \quad d = \sqrt{s_{max} - s},$$

donde s_{max} es el mayor valor de la similitud del conjunto considerado.

2.1. Coeficientes de asociación para datos presencia/ausencia

Los coeficientes de asociación expresan similitud entre los objetos a pesar de presencias/ausencias conjuntas o no.

La información necesaria para calcular los coeficientes de asociación, basados en datos de tipo presencia/ausencia, entre cualquier par de objetos, se puede representar en la siguiente tabla de contingencia 2×2 :

		Objeto 2		
		1	0	
Objeto 1	1	a	b	$a + b$
	0	c	d	$c + d$
		$a + c$	$b + d$	$p = a + b + c + d$

Cuadro 2.1: Tabla de frecuencias

En esta tabla:

a es la cantidad de características presentes en ambos objetos.

b es la cantidad de características presentes en el objeto 1 y ausentes en el objeto 2.

c es la cantidad de características presentes en el objeto 2 y ausentes en el objeto 1.

d representa la cantidad de características ausentes en ambos objetos, pero presente en al menos un objeto de la muestra.

p es la suma de las cantidades anteriores.

A continuación se muestra una lista de algunos coeficientes de similitud con el rango de sus valores:

Coeficiente	Origen	Rango
$S_1 = \frac{a}{\max\{(a+b), (a+c)\}}$	Braun-Blanquet (1932)	$\langle 0, 1 \rangle$
$S_2 = \frac{a}{\min\{(a+b), (a+c)\}}$	Simpson (1943)	$\langle 0, 1 \rangle$

Coeficiente	Origen	Rango
$S_3 = \frac{a}{b+c}$	Kulczynski (1927)	$\langle 0, \infty \rangle$
$S_4 = \frac{a}{a+b+c}$	Jaccard (1901)	$\langle 0, 1 \rangle$
$S_5 = \frac{a}{a + \frac{1}{2}(b+c)}$	Dice (1945) Sorensen (1948)	$\langle 0, 1 \rangle$
$S_6 = \frac{a}{a + 2(b+c)}$	Sokal & Sneath (1963)	$\langle 0, 1 \rangle$
$S_7 = \frac{1}{2} \left(\frac{a}{a+b} + \frac{a}{a+c} \right)$	Kulczynski (1927)	$\langle 0, 1 \rangle$
$S_8 = \frac{a}{2} \left(\frac{1}{a+b} + \frac{1}{a+c} \right)$	Driver & Kroeber (1932)	$\langle 0, 1 \rangle$
$S_9 = \frac{a}{a+b} + \frac{a}{a+c}$	Johnson (1967)	$\langle 0, 2 \rangle$
$S_{10} = \frac{a^2 - bc}{(a+b)(a+c)}$	McConnaughey (1964)	$\langle -1, 1 \rangle$
$S_{11} = \frac{a}{\sqrt{(a+b)(a+c)}}$	Driver & Kroeber (1932) Ochiai (1957)	$\langle 0, 1 \rangle$
$S_{12} = \frac{a^2}{(a+b)(a+c)}$	Sorgenfrei (1959)	$\langle 0, 1 \rangle$
$S_{13} = \frac{a}{\sqrt{(a+b)(a+c)}} - \frac{1}{2} \max\{(a+b), (a+c)\}$	Fager & McGowan (1963)	$\langle -\infty, 0,5 \rangle$
$S_{14} = \frac{a}{a+b+c+d}$	Russell & Rao (1940)	$\langle 0, 1 \rangle$
$S_{15} = \frac{a}{\frac{1}{2}a(b+c) + bc}$	Mountford (1962)	$\langle 0, 1 \rangle$
$S_{16} = \frac{ab+bc}{ab+2bc+cd}$	Peirce (1884)	$\langle 0, 1 \rangle$
$S_{17} = \frac{a - (a+b)(a+c)}{(a+b)(c+d)(a+c)(b+d)}$	Eyraud (1936)	$\langle -1, 0 \rangle$
$S_{18} = \frac{1}{2} \left(\frac{a}{a+b} + \frac{a}{a+c} + \frac{d}{c+d} + \frac{b}{b+d} \right)$	Sokal & Sneath (1963)	$\langle 0, 1 \rangle$
$S_{19} = \frac{a+d}{b+c}$	Sokal & Sneath (1963)	$\langle 0, \infty \rangle$
$S_{20} = \frac{a+d}{a+b+c+d}$	Sokal & Michener (1958)	$\langle 0, 1 \rangle$

Coeficiente	Origen	Rango
$S_{21} = \frac{2}{\pi} \arcsin \sqrt{S_{20}}$	Goodall (1967) Austin & Colwell (1977)	$\langle 0, 0,573 \rangle$
$S_{22} = \frac{a+d}{a + \frac{1}{2}(b+c) + d}$	Sokal & Sneath (1963)	$\langle 0, 1 \rangle$
$S_{23} = \frac{a+d}{a + 2(b+c) + d}$	Rogers & Tanimoto (1960)	$\langle 0, 1 \rangle$
$S_{24} = \frac{a+d-b-c}{p}$	Hamann (1961)	$\langle -1, 1 \rangle$
$S_{25} = \frac{ad}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$	Sokal & Sneath (1963)	$\langle 0, 1 \rangle$
$S_{26} = \frac{ad-bc}{(a+c)(b+d)}$	Peirce (1884)	$\langle -1, 1 \rangle$
$S_{27} = \frac{n(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)} = \chi^2$	Pearson (1905)	$\langle 0, \infty \rangle$
$S_{28} = \sqrt{\frac{\chi^2}{p + \chi^2}}$	Pearson (1905)	$\langle 0, 1 \rangle$
$S_{29} = \frac{\sqrt{2}(ad-bc)}{\sqrt{(ad-bc)^2 - (a+b)(c+d)(a+c)(b+d)}}$	Cole (1949)	$\langle -1, 1 \rangle$
$S_{30} = \frac{ad-bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$	Yule (1912) Pearson & Heron (1913)	$\langle -1, 1 \rangle$
$S_{31} = \frac{(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)} = S_{30}^2$	Doolittle (1885) Pearson (1926)	$\langle 0, 1 \rangle$
$S_{32} = \frac{\sqrt{ad} + a}{\sqrt{ad} + a + b + c}$	Baroni-Urbani & Buser (1976)	$\langle 0, 1 \rangle$
$S_{33} = \frac{\sqrt{ad} + a - b - c}{\sqrt{ad} + a + b + c}$	Baroni-Urbani & Buser (1976)	$\langle -1, 1 \rangle$
$S_{34} = \pm \sqrt{\frac{\chi^2}{\chi_{\text{máx}}^2}}$ con el signo de $(ad-bc)$	Cole (1949) Hurlbert (1969)	$\langle -1, 1 \rangle$
$S_{35} = \pm \sqrt{\frac{\chi^2 - \chi_{\text{mín}}^2}{\chi_{\text{máx}}^2 - \chi_{\text{mín}}^2}}$ con el signo de $(ad-bc)$	Hurlbert (1969)	$\langle -1, 1 \rangle$

Coeficiente	Origen	Rango
$S_{36} = \frac{ad - bc}{ad + bc}$	Yule (1900)	$\langle -1, 1 \rangle$
$S_{37} = \frac{\sqrt{ad} - \sqrt{bc}}{\sqrt{ad} + \sqrt{bc}}$	Yule (1912)	$\langle -1, 1 \rangle$
$S_{38} = \cos \frac{180\sqrt{bc}}{\sqrt{ad} + \sqrt{bc}}$	Pearson & Heron (1913)	$\langle -1, 1 \rangle$
$S_{39} = \frac{4(ad - bc)}{(a + d)^2 + (b + c)^2}$	Michael (1920)	$\langle -1, 1 \rangle$
$S_{40} = \frac{a}{(a + b)(a + c)/p}$	Forbes (1907)	$\langle 0, \infty \rangle$
$S_{41} = \log a - \log p - \log \left(\frac{a + b}{p} \right) - \log \left(\frac{a + c}{p} \right)$	Gilbert & Wells (1966)	$\langle -\infty, \infty \rangle$
$S_{42} = \frac{pa - (a + b)(a + c)}{p \min\{(a + b), (a + c)\} - (a + b)(a + c)}$	Forbes (1925)	$\langle -1, 1 \rangle$
$S_{43} = \frac{pa - (a + b)(a + c)}{pa + (a + b)(a + c)}$	Tarwid (1960)	$\langle -1, 1 \rangle$
$S_{44} = \frac{3a}{3a + b + c}$	Legendre & Legendre (1998)	$\langle 0, 1 \rangle$
$S_{45} = \frac{a + \frac{d}{2}}{p}$	Faith (1983)	$\langle 0, 1 \rangle$

Cuadro 2.2: Tabla 2

2.2. La tabla de datos de Ellenberg

Para el estudio de los coeficientes de asociación se va a considerar la información obtenida de 25 parcelas de una pradera situada en el valle del Danubio al sur de Ulm, Mueller-Dombois y Ellenberg (1974), es nuestro caso vamos a analizar la presencia/ausencia de 72 especímenes en dichas parcelas, no se consideran las parcelas en las que no hay ningún espécimen ni aquellas en la que están presentes todos los especímenes.

Capítulo 3

Representación gráfica de datos multidimensionales

3.1. Entorno Geométrico

En esta sección se ha considerado el punto de vista de Camiz (2016).

Vamos a trabajar en un espacio vectorial real V , por lo tanto se asumen conocidas sus propiedades, luego:

Vamos a considerar $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_j, \dots, \mathbf{e}_p)$ y $\mathbf{f} = (\mathbf{f}_1, \dots, \mathbf{f}_i, \dots, \mathbf{f}_n)$ las bases canónicas de \mathbb{R}^p y \mathbb{R}^n respectivamente, donde:

$$\mathbf{e}_j = (0, \dots, 1, \dots, 0)' \in \mathbb{R}^p, \text{ (donde la } j\text{-ésima componente es 1, el resto cero)}$$

y

$$\mathbf{f}_i = (0, \dots, 1, \dots, 0)' \in \mathbb{R}^n, \text{ (donde la } i\text{-ésima componente es 1, el resto cero).}$$

Podemos representar un vector $\mathbf{x}_i \in \mathbb{R}^p$ en la forma:

$$\mathbf{x}_i = \sum_{j=1}^p x_i^j \mathbf{e}_j$$

y su transpuesta:

$$\mathbf{x}'_i = (x_i^1, \dots, x_i^j, \dots, x_i^p)$$

De manera similar un vector $\mathbf{x}^j \in \mathbb{R}^n$ se expresa como:

$$\mathbf{x}^j = \sum_{i=1}^n x_i^j \mathbf{f}_i$$

y su transpuesta:

$$\mathbf{x}^{j'} = (x_1^j, \dots, x_i^j, \dots, x_n^j).$$

Por *métrica* nos referimos a una *distancia*, una función $d : E \times E \rightarrow \mathbb{R}_0^+$ con las siguientes propiedades:

1. $d(P, Q) \geq 0, \forall P, Q \in E$
2. $d(P, Q) = 0 \iff P = Q, \forall P, Q \in E$
3. $d(P, Q) = d(Q, P), \forall P, Q \in E$
4. $d(P, Q) \leq d(P, R) + d(R, Q), \forall P, Q, R \in E$

Un conjunto E en el cual se define la distancia entre sus puntos es llamado un *espacio métrico*. La distancia es un tipo particular de disimilitud (que solo cumple 1., 2. y 3.) caracterizado por satisfacer la desigualdad triangular. Por *norma* nos referimos a una función de valores reales $\|\cdot\| : V \rightarrow \mathbb{R}_0^+$ que a cada vector \mathbf{v} le asigna un valor no negativo real $\|\mathbf{v}\|$, tal que:

1. $\|\mathbf{v}\| \geq 0$, con $\|\mathbf{v}\| = 0 \iff \mathbf{v} = \mathbf{0}$;
2. $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|, \forall a \in \mathbb{R}, \forall \mathbf{v} \in V$;
3. $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|, \forall \mathbf{v}, \mathbf{w} \in V$.

A un espacio vectorial con norma se le denomina *espacio normado*. De manera intuitiva, la norma de un vector se puede entender como la forma de definir su longitud.

Hay varias formas de definir normas: las más conocidas son las normas L_p , basadas en las componentes de los vectores. Para $p \geq 1$, la p -norma de un vector $\mathbf{v} = (x_1, \dots, x_n)$, es dada por:

$$\|\mathbf{v}\|_p = \left(\sum_{i=1}^n \|x_i\|^p \right)^{1/p}$$

Los casos especiales son la norma L_1 también llamada de *Manhattan* o del *taxista*, la norma *euclidiana* L_2 , y la norma del máximo L_∞ , dada por:

$$\|\mathbf{v}\|_\infty = \lim_{p \rightarrow \infty} \|\mathbf{v}\|_p = \max_i |x_i|.$$

Dado un espacio vectorial V , un *producto escalar* (también llamado *producto interno* o *punto*) es una forma bilineal $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ con las siguientes propiedades:

1. *simetría*: $\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{w}, \mathbf{v} \rangle$, $\forall \mathbf{v}, \mathbf{w} \in V$;
2. *linealidad* (en ambos factores, debido a 1.):
 $\langle a\mathbf{v} + b\mathbf{w}, \mathbf{z} \rangle = a\langle \mathbf{v}, \mathbf{z} \rangle + b\langle \mathbf{w}, \mathbf{z} \rangle$, $\forall \mathbf{v}, \mathbf{w}, \mathbf{z} \in V$, $\forall a, b \in \mathbb{R}$;
3. *definida*: $\langle \mathbf{0}, \mathbf{0} \rangle = 0$ (esto viene de 2.);
4. *positiva*: $\langle \mathbf{v}, \mathbf{v} \rangle > 0$, $\forall \mathbf{v} \in V, \mathbf{v} \neq \mathbf{0}$.

Un espacio vectorial provisto de un producto escalar es llamado un *espacio con producto interno*. Tal espacio tiene las siguientes características:

1. La función $\|\cdot\|_2$ definida por:

$$\|\mathbf{v}\|_2 = \langle \mathbf{v}, \mathbf{v} \rangle^{1/2} = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}.$$

Es una norma, llamada *norma asociada al producto interno* $\langle \cdot, \cdot \rangle$.

En el caso de \mathbb{R}^n , para $\mathbf{v} = (x_1, \dots, x_n)$ y $\mathbf{u} = (y_1, \dots, y_n)$ se tiene:

$$\langle \mathbf{v}, \mathbf{u} \rangle = \sum_{i=1}^n x_i y_i$$

el producto interno de \mathbb{R}^n .

2. Un conjunto $B \subset V$ es una *base* si

- i) B genera V .
- ii) B es linealmente independiente.

En caso en que B sea finito, por ejemplo $B = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$, la condición de que B sea una base de V se expresa así:

- i) Todo elemento $\mathbf{v} \in V$ se puede expresar en la forma

$$\mathbf{v} = \sum_{j=1}^p x_j \mathbf{v}_j, \text{ donde } x_j \in \mathbb{R}.$$

- ii) Los elementos $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p$ son linealmente independientes (cf. Chávez (2004; Sección 1.3))

3. Para poder definir un producto escalar, solo es necesario saber como está definido sobre los vectores de una base. Esto, debido a la bilinealidad:

$$\langle \mathbf{v}, \mathbf{w} \rangle = \left\langle \sum_i a_i \mathbf{v}_i, \sum_j b_j \mathbf{v}_j \right\rangle = \sum_i \sum_j a_i b_j \langle \mathbf{v}_i, \mathbf{v}_j \rangle.$$

Esto se puede escribir de forma matricial como sigue:

$$\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{a}'_i \mathbf{v}_i, \mathbf{b}'_j \mathbf{v}_j \rangle = \mathbf{a}'_i \langle \mathbf{v}_i, \mathbf{v}_j \rangle \mathbf{b}_j = \mathbf{a}' M \mathbf{b}, \quad (3.1)$$

donde \mathbf{a} y \mathbf{b} son los vectores *columna* de las coordenadas de \mathbf{v} y \mathbf{w} con respecto a la base B , respectivamente, y M es la matriz (simétrica definida positiva) de los productos escalares entre los vectores en B , esto se conoce como el *Gramiano* Chávez (2004; pág 189). A la matriz M usualmente se le denomina la *métrica* del espacio, por razones que se verán mas adelante.

4. Se dice que dos vectores \mathbf{v} y \mathbf{w} son *ortogonales* si $\langle \mathbf{v}, \mathbf{w} \rangle = 0$. De hecho, la noción de producto escalar permite definir ángulos entre vectores.

El coseno del ángulo θ formado entre $\mathbf{v} \neq 0$ y $\mathbf{w} \neq 0$ se define como:

$$\theta = \arccos \left(\frac{\langle \mathbf{v}, \mathbf{w} \rangle}{\|\mathbf{v}\| \|\mathbf{w}\|} \right).$$

5. Dados dos vectores \mathbf{v} y \mathbf{w} , llamamos *proyección ortogonal* de \mathbf{v} sobre \mathbf{w} al vector:

$$\mathcal{P}_{\mathbf{w}}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{w} \rangle}{\|\mathbf{w}\|^2} \mathbf{w}.$$

Es fácil de mostrar que:

$$\langle \mathcal{P}_{\mathbf{w}}(\mathbf{v}), \mathbf{v} - \mathcal{P}_{\mathbf{w}}(\mathbf{v}) \rangle = 0. \quad (3.2)$$

6. La proyección de un vector sobre otro y el hecho de que, por (3.2), cualquier vector se puede descomponer en dos componentes ortogonales, nos lleva a pensar en una *base ortogonal*, esto es, una base conformada por vectores ortogonales dos a dos, sea la base $B = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$ tal que:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = 0, \quad \text{para } i \neq j.$$

Sobre tal base todas las componentes de los vectores con respecto a los vectores de la base, son sus proyecciones ortogonales sobre ellos.

7. (Descomposición de Gram-Schmidt). Dada una base $B = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$ de V , existe una base ortogonal $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$ de V tal que $\mathbf{u}_j \in \mathcal{L}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$, para cada $j = 1, 2, \dots, p$ Chávez (2004; pág 46).

Un *espacio afín* E_V asociado a un espacio vectorial real V es un conjunto de puntos junto con una función biyectiva $t : V \times E \rightarrow E$, definida como:

$$t(\mathbf{v}, P) = \mathbf{v} + P$$

cumple:

1. $\mathbf{0} + P = P, \forall P \in E$;
2. $\mathbf{v} + (\mathbf{w} + P) = (\mathbf{v} + \mathbf{w}) + P, \forall P \in E, \forall \mathbf{v}, \mathbf{w} \in V$.

Se puede decir que aplicando un vector \mathbf{v} a un punto P resulta una *traslación* de P al punto $Q = \mathbf{v} + P$. Una *traslación* de un subconjunto de E significa aplicar un vector a todos los puntos del subconjunto. En efecto, se puede tomar ventaja de algunas propiedades de los espacios afines, heredadas del espacio vectorial asociado, en particular el concepto de línea recta, que proviene de los subespacios vectoriales 1-dimensionales.

Dado un espacio afín E , una *variedad lineal*, o *subespacio afín*, es un subconjunto de E de la forma $S + P$, donde P es cualquier punto de E y S es un subespacio de V . En otras palabras, una variedad lineal es el conjunto de puntos que se pueden obtener a partir de P vía una translación contenida en S . Se puede decir que dos variedades $S + P$ y $T + Q$ son *paralelas* si S es subespacio de T o viceversa y el vector $\mathbf{Q} - \mathbf{P}$ no pertenece ni a S ni a T . Por ejemplo los puntos, líneas, planos, hiperplanos son variedades lineales.

Si el espacio vectorial es normado, es natural introducir una métrica en el espacio afín como la norma del vector de translación que envía un punto a otro:

$$d(P, Q) = \|\mathbf{Q} - \mathbf{P}\| \text{ (métrica inducida por la norma).}$$

De esta forma E se convierte en un espacio métrico. En particular, la desigualdad triangular se conserva debido al tercer axioma de la norma. La norma L_p induce una distancia L_p :

$$d_p(P, Q) = \|\mathbf{Q} - \mathbf{P}\|_p.$$

Es de particular relevancia el caso en el cual la norma L_2 es inducida por un producto escalar definido sobre V : en efecto, si este fuera el caso, si M es la matriz del producto escalar respecto a la base B y las coordenadas de los puntos de acuerdo a la misma base son $P = (x_1, \dots, x_p)$ y $Q = (y_1, \dots, y_p)$ respectivamente, entonces se tiene:

$$d(P, Q) = \sqrt{(Q - P)'M(Q - P)} = \sqrt{(\mathbf{y} - \mathbf{x})'M(\mathbf{y} - \mathbf{x})}. \quad (3.3)$$

Llamamos *espacio euclidiano* a un espacio afín cuya métrica es la L_2 inducida por M el producto escalar de V dado por (3.3). En el caso cartesiano se tiene:

$$d(P, Q) = \sqrt{\sum_{i=1}^p (y_i - x_i)^2}. \quad (3.4)$$

es una simplificación de (3.3) cuando $M = I$.

3.2. Datos, nubes y métricas

En esta sección se ha considerado el punto de vista de Camiz (2017a).

3.2.1. Datos

Supongamos que tenemos un conjunto de n objetos y los describimos de acuerdo a un conjunto de atributos o su relación con los otros objetos.

Los objetos a los que llamamos *unidades* son descritos por algunas características observadas en ellos, lo que podemos plasmar en una matriz \mathbf{X} de datos $n \times p$, cuyas columnas \mathbf{x}_j , $j = 1, \dots, p$ son los niveles de las p características observadas en las unidades y cuyas filas \mathbf{x}'_i , $i = 1, \dots, n$ son las correspondientes observaciones, esto es, el conjunto de los niveles de las p características detectadas en la i -ésima unidad. Los niveles de las características pueden ser de diferente tipo.

Los siguientes tipos de características son relevantes:

Presencia/ausencia: características que pueden estar presentes o no en una unidad, toman el valor 0 si la característica está ausente y el valor 1 si está presente.

Nominal: los niveles de las características son atributos, por ejemplo: color, forma, tipo de documento, etc.

Escala: características cuyos niveles se pueden ordenar totalmente, por ejemplo: una nota escolar, el grado de conformidad con una declaración, etc.

Rangos: los niveles representan la posición de la unidad en la fila compuesta por todos los objetos disponibles, ordenados de acuerdo a cierto criterio. Por ejemplo: orden de llegada.

Frecuencias: los niveles son conteos de alguna entidad, por lo tanto enteros positivos, aunque se pueden usar decimales positivos. Por ejemplo: cantidad de especies presentes en una muestra, cantidad de personas de cierto nivel socio-económico, riqueza que se expresa en moneda y decimales, etc.

Medidas: los niveles representan medidas tomadas sobre los objetos, por lo tanto son números reales.

En este trabajo nos enfocaremos en datos de tipo presencia/ausencia. Para trabajar necesitamos definir relaciones entre objetos (unidades o caracteres) que nos permitan describir como estos se van relacionando entre ellos.

3.2.2. Nubes

Llamamos nube de puntos \mathcal{N} a la representación geométrica que se espera que represente los atributos de las unidades y la relación entre ellos. Se pueden construir dos nubes de datos a partir de la tabla \mathbf{X} :

- La nube de puntos \mathcal{N}_n que representa las n unidades en el espacio \mathbb{R}^p , donde se puede definir un producto escalar M .
- La nube de puntos \mathcal{N}_p que representa los p caracteres en el espacio \mathbb{R}^n , donde se puede definir un producto escalar D_n .

Ejemplos:

- Caso de estudio.

En nuestro caso vamos a analizar la presencia/ausencia de 72 especímenes en 25 parcelas, en este caso las unidades serían las parcelas y los caracteres serían los especímenes. Lo cual se representa de la siguiente manera

$$\mathbf{X} = \begin{array}{c} \text{Parcelas} \\ \left(\begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1,72} \\ x_{21} & x_{22} & \cdots & x_{2,72} \\ \vdots & \vdots & \ddots & \vdots \\ x_{25,1} & x_{25,2} & \cdots & x_{25,72} \end{array} \right) \end{array} \begin{array}{c} \text{Especímenes} \end{array}$$

donde

$$x_{i,j} = \begin{cases} 1; & \text{indica la presencia del espécimen } j \text{ en la parcela } i \\ 0; & \text{indica la ausencia del espécimen } j \text{ en la parcela } i \end{cases}.$$

- Tolerancia a la sequia de diferentes variedades de papa.

Se siembran variedades de papa (canchán, amarilla, yungay, huayro, peruanita) donde bajo las mismas condiciones climáticas, mismo tipo de suelo, nutrientes y volumen hídrico; se estudia que variedades tienen mayor tolerancia al déficit hídrico (falta de riego). En este contexto, se miden diferentes parámetros fisiológicos como: el rendimiento, número de hojas, rendimiento (producción), clorofila de las hojas, nitrógeno de las hojas. En este caso las unidades serían las variedades de papa y los caracteres serían los parámetros fisiológicos.

3.2.3. Métricas

Las métricas (determinadas como matrices por (3.3)) definidas sobre los espacios \mathbb{R}^p y \mathbb{R}^n nos permiten medir la proximidad entre dos elementos que pertenecen a la misma nube, lo que permite ver la geometría de la nube y analizar la estructura de los datos.

En \mathbb{R}^p usualmente se considera $M = \mathbf{I}_p$. La distancia al cuadrado entre las unidades i y k es:

$$\|\mathbf{x}_i - \mathbf{x}_k\|_{\mathbf{I}_p}^2 = (\mathbf{x}_i - \mathbf{x}_k)' \mathbf{I}_p (\mathbf{x}_i - \mathbf{x}_k) = (\mathbf{x}_i - \mathbf{x}_k)' (\mathbf{x}_i - \mathbf{x}_k)$$

Luego:

$$\|\mathbf{x}_i - \mathbf{x}_k\|_{\mathbf{I}_p}^2 = \sum_{j=1}^p (x_i^j - x_k^j)^2, \quad (3.5)$$

donde \mathbf{I}_p es la matriz identidad.

Siendo D_σ la matriz diagonal asociada a $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_p)$ y dada por:

$$\mathbf{D}_\sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_p \end{pmatrix}.$$

También se puede elegir $\mathbf{M} = \mathbf{D}_\sigma^{-2}$. La distancia al cuadrado entre las unidades i e j viene dada por:

$$\|\mathbf{x}_i - \mathbf{x}_k\|_{\mathbf{D}_\sigma^{-2}}^2 = (\mathbf{x}_i - \mathbf{x}_k)' \mathbf{D}_\sigma^{-2} (\mathbf{x}_i - \mathbf{x}_k) = \sum_{j=1}^p \left(\frac{x_i^j - x_k^j}{\sigma_j} \right)^2.$$

donde σ_j^2 es la varianza del carácter j y \mathbf{D}_σ^{-2} es la matriz diagonal definida por:

$$\mathbf{D}_\sigma^{-2} = \begin{pmatrix} 1/\sigma_1^2 & & & \\ & \ddots & & \\ & & 1/\sigma_j^2 & \\ & & & \ddots & \\ & & & & 1/\sigma_p^2 \end{pmatrix}.$$

En \mathbb{R}^n usualmente se considera la ponderación $\mathbf{M} = \mathbf{D}_w$, en la cual se asocia un peso $w_i > 0$ a la unidad i , tal que $\sum_{i=1}^n w_i = 1$, \mathbf{D}_w es la matriz diagonal definida por:

$$\mathbf{D}_w = \begin{pmatrix} w_1 & & & \\ & \ddots & & \\ & & w_i & \\ & & & \ddots & \\ & & & & w_n \end{pmatrix}.$$

Considerando $\mathbf{1}_n$ el vector columna de \mathbb{R}^n que tiene todas sus componentes iguales 1, se tiene

$$\|\mathbf{1}_n\|_{\mathbf{D}_w}^2 = \mathbf{1}_n' \mathbf{D}_w \mathbf{1}_n = 1 \quad (\text{vector unitario}).$$

Por lo tanto:

$$\mathbf{1}'_n \mathbf{D}_w \mathbf{x}^j = \sum_{i=1}^n w_i x_i^j = \bar{x}^j.$$

Esto es, el producto escalar de los vectores $\mathbf{1}_n$ y \mathbf{x}^j es igual a la media del caracter j . Geométricamente, esto significa que la coordenada de la proyección ortogonal de \mathbf{x}^j sobre el eje generado por $\mathbf{1}_n$ ($\Delta_{\mathbf{1}_n}$) es \bar{x}^j .

Un caracter \mathbf{x}^j está centrado cuando $\bar{x}^j = 0$; esto se puede lograr reemplazando cada una de sus componentes x_i^j por $x_i^j - \bar{x}^j$.

Si denotamos por \mathbf{z}^j al caracter \mathbf{x}^j centrado, su representación en \mathbb{R}^n es la proyección de \mathbf{x}^j sobre el hiperplano $\Delta_{\mathbf{1}_n}^\perp$ ortogonal a $\Delta_{\mathbf{1}_n}$. Por lo tanto se tiene:

$$\|\mathbf{z}^j\|_{\mathbf{D}_w}^2 = \sum_{i=1}^n w_i (x_i^j - \bar{x}^j)^2 = \text{var } \mathbf{x}^j = \sigma_j^2.$$

Y el producto escalar entre los caracteres \mathbf{z}^j y \mathbf{z}^k viene dado por:

$$\langle \mathbf{z}^j, \mathbf{z}^k \rangle_{\mathbf{D}_w} = (\mathbf{z}^j)' \mathbf{D}_w \mathbf{z}^k = \sum_{i=1}^n w_i (x_i^j - \bar{x}^j)(x_i^k - \bar{x}^k) = \text{cov}(\mathbf{x}^j, \mathbf{x}^k).$$

De donde

$$\rho = \cos \alpha = \frac{\langle \mathbf{z}^j, \mathbf{z}^k \rangle_{\mathbf{D}_w}}{\|\mathbf{z}^j\|_{\mathbf{D}_w} \|\mathbf{z}^k\|_{\mathbf{D}_w}} = \frac{\text{cov}(\mathbf{x}^j, \mathbf{x}^k)}{\sigma_j \sigma_k}$$

es el coeficiente de correlación ρ entre los caracteres j y k .

Observaciones

- Si los caracteres son centrados, su representación está sobre el hiperplano $\Delta_{\mathbf{1}_n}^\perp$.
- Si los caracteres son estandarizados, ellos están sobre la hiperesfera con centro 0 y radio 1.
- Considerando \mathbf{Z} la tabla de n filas y p columnas, cuya j -ésima columna ($j = 1, \dots, p$) corresponde a los componentes del caracter centrado \mathbf{z}^j , la matriz $\mathbf{V} = \mathbf{Z}' \mathbf{D}_w \mathbf{Z}$ es denominada la *matriz de varianza-covarianza*, la cual es simétrica, en su diagonal se encuentran las varianzas de los caracteres $\mathbf{x}^1, \dots, \mathbf{x}^p$ y en la fila j y columna k tenemos $\text{cov}(\mathbf{x}^j, \mathbf{x}^k)$, esto es:

$$\mathbf{V} = \begin{pmatrix} \text{var}(\mathbf{x}^1) & \text{cov}(\mathbf{x}^1, \mathbf{x}^j) & \text{cov}(\mathbf{x}^1, \mathbf{x}^p) \\ & \ddots & \\ \text{cov}(\mathbf{x}^j, \mathbf{x}^1) & \text{var}(\mathbf{x}^j) & \\ & & \ddots & \\ \text{cov}(\mathbf{x}^p, \mathbf{x}^1) & & & \text{var}(\mathbf{x}^p) \end{pmatrix}. \quad (3.6)$$

En particular:

$$\text{traza } \mathbf{V} = \sum_{j=1}^p \text{var}(\mathbf{x}^j).$$

3.3. Análisis de componentes principales (ACP)

En esta sección se considera el punto de vista de Camiz (2017a).

- ¿Cual es el punto de \mathbb{R}^p más cercano a la nube original \mathcal{N} de los puntos \mathbf{x}_i , que es el punto que mejor los representa?
- ¿Cual es la línea recta en \mathbb{R}^p más cercana a la nube \mathcal{N} ? ¿Está la nube \mathcal{N} incluida en esta línea?

El análisis de componentes principales (ACP) es probablemente una de las técnicas más antiguas y mejor conocidas del análisis multivariado. Fue introducido por primera vez por Pearson (1901) y desarrollado independientemente por Hotelling (1933). Como muchos métodos multivariados, no fue ampliamente utilizado hasta la llegada de la computadora, pero ahora se encuentra en cada software estadístico Jolliffe (2002).

El ACP tiene por objeto transformar un conjunto de variables originales, en un nuevo conjunto de variables denominadas componentes principales. Estas últimas se caracterizan porque son incorreladas entre sí y, además, pueden ordenarse de acuerdo con la información que llevan incorporada, como medida de esta información se considera su varianza, es decir, cuanto mayor sea su varianza mayor es la cantidad de información que lleva incorporada dicha componente. Por esta razón se selecciona como primera componente aquella que tenga mayor varianza, mientras que la última componente es la de menor varianza. Las componentes principales se expresan como una combinación lineal de las variables originales. La cantidad de componentes principales coincide en número con el de las variables originales.

El ACP nos permite considerar una cantidad de componentes (menor que la original) que pueden explicar una proporción aceptable de la varianza global o inercia de la nube de puntos, es decir, permite reducir la dimensión del número de variables originales. En otras palabras nos permite hallar subespacios de menor dimensión que represente la Nube de puntos original.

Como medida de proximidad definimos la inercia.

Definición 3.3.1 *La inercia de una nube de puntos \mathcal{N} con respecto a un punto $a \in \mathbb{R}^p$ se define como:*

$$I_a = \sum_{i=1}^n w_i \|a - \mathbf{x}_i\|_M^2.$$

Se puede interpretar la inercia como una medida de la dispersión de los datos respecto a un determinado punto, cuando los datos se encuentran muy dispersos entre sí su inercia respecto a dicho punto aumenta y cuando se encuentran poco dispersos su inercia disminuye. La inercia es, por lo tanto, un concepto cercano al de varianza.

3.3.1. Búsqueda del punto en \mathbb{R}^p más cercano a \mathcal{N}

Como hemos considerado la inercia como medida de proximidad, el siguiente teorema nos ayuda a determinar cual es el punto más cercano a la nube \mathcal{N} .

Teorema 3.1 *El punto en \mathbb{R}^p más cercano a la nube \mathcal{N} es su centroide:*

$$g = \sum_{i=1}^n w_i \mathbf{x}_i = \bar{\mathbf{x}}.$$

En el sentido que g es el punto que minimiza la inercia I_a , que es el cuadrado de la distancia ponderada de la nube \mathcal{N} al punto a .

Prueba

Para cualquier punto $a \in \mathbb{R}^p$ tenemos:

$$\begin{aligned} I_a &= \sum_{i=1}^n w_i \|a - g + g - \mathbf{x}_i\|_M^2 \\ &= \sum_{i=1}^n w_i \|a - g\|_M^2 + \sum_{i=1}^n w_i \|g - \mathbf{x}_i\|_M^2 + 2 \sum_{i=1}^n w_i (a - g, g - \mathbf{x}_i)_M \\ &= \|a - g\|_M^2 + I_g + 2 \langle a - g, \sum_{i=1}^n w_i (g - \mathbf{x}_i) \rangle_M \end{aligned}$$

Como

$$\sum_{i=1}^n w_i (g - \mathbf{x}_i) = 0,$$

entonces

$$I_a = I_g + \|a - g\|_M^2, \quad (3.7)$$

y el punto $a \in \mathbb{R}^p$ para el cual I_a es mínimo es g (independientemente de la métrica elegida en \mathbb{R}^p). ■

En adelante consideraremos que el sistema de referencia donde se representan las unidades tiene su origen en el centroide g . Esto corresponde a asociar a cada unidad $i = 1, \dots, n$, representada por \mathbf{x}_i en el sistema cuyo origen es O y coordenadas $(x_i^j, j = 1, \dots, p)$ los puntos de \mathbb{R}^p cuyas coordenadas son $(z_i^j, j = 1, \dots, p)$ en el sistema cuyo origen es g .

Por lo tanto para $j = 1, \dots, p$ se tiene:

- $\sum_{i=1}^n w_i z_i^j = \bar{z}^j = 0.$
- $\text{Var } z^j = \sum_{i=1}^n w_i (z_i^j)^2 = \sigma_j^2.$
- El caracter z^j se ubica en el hiperplano $\Delta_{\mathbf{1}_n}^\perp$.

3.3.2. Cálculo de I_g , la inercia de \mathcal{N} relativa a su centroide

Se va a calcular I_g , la inercia de \mathcal{N} relativa a su centroide g , $g = O$:

$$I_g = \sum_{i=1}^n w_i \|\mathbf{x}_i\|_M^2. \quad (3.8)$$

En general se tiene:

$$I_g = \text{traza}(\mathbf{V}\mathbf{M}), \quad (3.9)$$

donde

$$\mathbf{V} = \mathbf{X}' \mathbf{D}_n \mathbf{X} = \sum_{i=1}^n w_i \mathbf{x}_i \mathbf{x}_i'.$$

En efecto:

$$\begin{aligned} I_g &= \sum_{i=1}^n w_i \|\mathbf{x}_i\|_M^2 = \sum_{i=1}^n w_i \mathbf{x}_i' \mathbf{M} \mathbf{x}_i \\ &= \sum_{i=1}^n w_i \text{traza} (\mathbf{x}_i' \mathbf{M} \mathbf{x}_i) = \sum_{i=1}^n w_i \text{traza} (\mathbf{x}_i \mathbf{x}_i' \mathbf{M}), \end{aligned}$$

esto último debido a que $\text{traza} (\mathbf{A}\mathbf{B}) = \text{traza} (\mathbf{B}\mathbf{A})$, y como el operador traza es lineal, se tiene que

$$I_g = \text{traza} \left(\sum_{i=1}^n w_i \mathbf{x}_i \mathbf{x}_i' \mathbf{M} \right) = \text{traza} (\mathbf{V}\mathbf{M}).$$

- Si $\mathbf{M} = \mathbf{I}_p$, por (3.9) se tiene:

$$I_g = \text{traza} (\mathbf{V}\mathbf{I}_p) = \text{traza} (\mathbf{V}).$$

Si los caracteres son estandarizados se tiene que $I_g = p$.

- Si $\mathbf{M} = \mathbf{D}_\sigma^{-2}$ es la matriz diagonal cuyos elementos son las inversas de las varianzas de los caracteres, se tiene que

$$I_g = \text{traza} (\mathbf{V}\mathbf{D}_\sigma^{-2}).$$

3.3.3. Búsqueda del subespacio q -dimensional más cercano a la nube \mathcal{N}

Ahora se va a buscar la línea recta, el plano, en general el subespacio afín de dimensión q más cercano a la nube \mathcal{N} .

Sean W un subespacio de dimensión q que contiene a $g = O$, $a \in W^\perp$ y W_a el subespacio paralelo a W que contiene a a , denotando α_i la proyección de x_i sobre W y β_i la proyección sobre W^\perp , entonces:

$$x_i = \alpha_i + \beta_i.$$

De esto se tiene:

$$g = O = \sum_{i=1}^n w_i x_i = \sum_{i=1}^n w_i \alpha_i + \sum_{i=1}^n w_i \beta_i.$$

La proyección de x_i sobre W_a es $\alpha_i + a$.

Se tiene que $\sum_{i=1}^n w_i \alpha_i = O$ y $\sum_{i=1}^n w_i \beta_i = O$ porque los datos son centrados y pertenecen a subespacios ortogonales.

Luego

$$I_{W_a} = \sum_{i=1}^n w_i \|\beta_i - a\|_M^2 = \sum_{i=1}^n w_i \|\beta_i\|_M^2 + \|a\|_M^2 - 2 \left\langle \sum_{i=1}^n w_i \beta_i, a \right\rangle_M,$$

de donde se tiene que

$$I_{W_a} = I_W + \|a\|_M^2.$$

Luego, el subespacio afín más cercano a la nube \mathcal{N} de las unidades contiene al centroide de \mathcal{N} .

Ahora buscaremos la línea recta Δu que contiene a g más cercana a los puntos \mathbf{x}_i , que es tal que $I_{\Delta u}$ sea mínima.

Como $x_i = \alpha_i + \beta_i$, donde $\alpha_i \in \Delta u$ y $\beta_i \in \Delta u^\perp$, entonces

$$\|x_i\|_M^2 = \|\alpha_i\|_M^2 + \|\beta_i\|_M^2$$

Además,

$$I_{\Delta u} = \sum_{i=1}^n w_i \|\beta_i\|_M^2$$

e

$$I_{\Delta u^\perp} = \sum_{i=1}^n w_i \|\alpha_i\|_M^2.$$

De estos resultados y de (3.8) se obtiene que

$$I_g = I_{\Delta u} + I_{\Delta u^\perp},$$

lo cual es equivalente a buscar Δu :

- Minimizando $I_{\Delta u}$, donde Δu es la línea recta más cercana a los puntos \mathbf{x}_i de \mathcal{N} .
- Maximizando $I_{\Delta u^\perp}$, donde Δu es tal que la inercia relativa al centroide $g = O$ de las proyecciones de los puntos de \mathcal{N} sobre Δu sea máxima.

Se aplica el mismo argumento cuando se busca un subespacio de dimensión q más cercano a los \mathbf{x}_i , mostrando que es equivalente a buscar W tal que la inercia I_W de la nube proyectada sobre W relativa al centroide $g = O$ es máxima.

Observación

Buscar el subespacio W de dimensión q sobre el cual la inercia I_W de la nube \mathcal{N}_W proyectada sobre W es máxima, corresponde a la idea de buscar W tal que la nube \mathcal{N} sea menos sesgada por la proyección sobre un espacio de dimensión q . En efecto, la inercia relativa a g de la nube $\mathcal{N}_W = \{\alpha_i : i = 1, \dots, n\}$, la proyección de \mathcal{N} sobre W , es igual a:

$$I_W = \sum_{i=1}^n w_i \|\alpha_i\|_M^2 = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_i w_j \|\alpha_i - \alpha_j\|_M^2.$$

Porque:

$$\sum_{i=1}^n \sum_{j=1}^n w_i w_j \langle \alpha_i, \alpha_j \rangle_M = \left\langle \sum_{i=1}^n w_i \alpha_i, \sum_{j=1}^n w_j \alpha_j \right\rangle_M = \langle O, O \rangle_M = 0.$$

debido a que el centroide $\sum_{i=1}^n w_i \alpha_i$ de \mathcal{N}_W es la proyección del centroide g de \mathcal{N} sobre W , esto es el mismo g . Luego por proyección las distancias se reducen, y así el subespacio W de dimensión q es tal que la media de las distancias al cuadrado entre los puntos de la nube proyectados sobre W es tan grande como sea posible.

3.3.4. Búsqueda de los ejes principales

Para buscar el primer eje principal, lo que se desea es hallar una línea recta que contiene al centroide $g = O$, sobre la cual la inercia de la nube de puntos proyectada sea máxima. Sea $u \in \mathbb{R}^p$ tal que $\|u\|_M = 1$ y Δu la línea recta generada por u .

Si α_i es la proyección de x_i sobre Δu , tenemos $\alpha_i = c_i u$ y $c_i = \langle u, x_i \rangle_M = x_i' M u$ es la abscisa de la proyección de x_i sobre Δu .

La inercia de la nube proyectada sobre Δu es igual a:

$$I_{\Delta u} = \sum_{i=1}^n w_i \|c_i u\|_M^2 = \sum_{i=1}^n w_i (c_i)^2.$$

Definimos el vector $c \in \mathbb{R}^n$ cuyas componentes son los c_i relativos a la base canónica $(e'_i)_{i=1}^n$

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_i \\ \vdots \\ c_n \end{pmatrix} = \sum_{i=1}^n c_i e'_i.$$

De esto se tiene

$$\bar{c} = \sum_{i=1}^n w_i c_i = \sum_{i=1}^n w_i x'_i M u = g' M u = 0,$$

y por lo tanto c está centrado como los originales.

La varianza de c es:

$$\text{Var}(c) = \sum_{i=1}^n w_i (c_i)^2 = c' D_n c = I_{\Delta u^\perp}$$

y si Δu es la línea recta para la cual $I_{\Delta u^\perp}$ es máxima, la varianza de c es máxima.

Por lo tanto:

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_i \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} x'_1 M u \\ \vdots \\ x'_i M u \\ \vdots \\ x'_n M u \end{pmatrix} = X M u \in \mathbb{R}^n,$$

de donde

$$I_{\Delta u^\perp} = c' D_n c = u' M X' D_n X M u.$$

Dado que se supone que los caracteres x^j son centrados, tenemos que

$$X' D_n X = V,$$

de donde

$$I_{\Delta u^\perp} = u' M V M u.$$

El problema es hallar un vector $u \in \mathbb{R}^p$ tal que $I_{\Delta u^\perp}$ sea máximo, por lo tanto tenemos que resolver el siguiente problema de optimización:

$$\begin{cases} \text{Maximizar} & u' \mathbf{M} \mathbf{V} \mathbf{M} u \\ \text{sujeto a} & u' \mathbf{M} u = 1. \end{cases} \quad (3.10)$$

Aplicando multiplicadores de Lagrange, definimos la función:

$$L(u, \lambda) = u' \mathbf{M} \mathbf{V} \mathbf{M} u - \lambda(u' \mathbf{M} u - 1).$$

Derivando con respecto a u se tiene:

$$\frac{\partial}{\partial u} L(u, \lambda) = \frac{\partial(u' \mathbf{M} \mathbf{V} \mathbf{M} u)}{\partial u} - \frac{\partial(\lambda(u' \mathbf{M} u - 1))}{\partial u} = 2\mathbf{M} \mathbf{V} \mathbf{M} u - 2\lambda \mathbf{M} u.$$

Igualando a cero esta expresión:

$$2\mathbf{M} \mathbf{V} \mathbf{M} u - 2\lambda \mathbf{M} u = 0 \text{ (siendo } \mathbf{M} \text{ no singular)}$$

y se obtiene

$$\mathbf{V} \mathbf{M} u = \lambda u. \quad (3.11)$$

Cabe resaltar que u es un autovector de la matriz $\mathbf{V} \mathbf{M}$ asociado al autovalor λ .

Luego:

$$u' \mathbf{M} \mathbf{V} \mathbf{M} u = u' \mathbf{M} \lambda u = \lambda u' \mathbf{M} u = \lambda \|u\|_{\mathbf{M}}^2.$$

Por lo tanto el vector $u_1 \in \mathbb{R}^p$ que maximiza $I_{\Delta u^\perp}$ es un autovector unitario asociado al mayor autovalor λ_1 de $\mathbf{V} \mathbf{M}$.

Por lo tanto se obtiene:

$$I_{\Delta u_1^\perp} = \lambda_1.$$

y

$$\text{Var } c^1 = (c^1)' \mathbf{D}_n c^1 = \|c^1\|_{\mathbf{D}_n}^2 = \lambda_1$$

El eje Δu_1 es denominado *primer eje principal* y c^1 el *primer componente principal*, c^1 está centrado y tiene varianza λ_1 . Si $I_{\Delta u_1^\perp} = I_g$, entonces la nube \mathcal{N} está sobre la línea recta Δu_1 , luego obtenemos una representación exacta de \mathcal{N} sobre esta línea y una simplificación del conjunto de datos. En efecto, en lugar de np números, ahora \mathcal{N} está completamente identificada solamente por $n + p$ números, las p componentes

del vector u_1 y las n abscisas de los puntos sobre u_1 .

Vamos a determinar ahora el segundo eje principal. El procedimiento es similar al anterior, resolviendo el siguiente problema de optimización:

$$\left\{ \begin{array}{l} \text{Maximizar } u' \mathbf{M} \mathbf{V} \mathbf{M} u \\ \text{sujeto a } u' \mathbf{M} u = 1 \\ u' \mathbf{M} u_1 = 0. \end{array} \right. \quad (3.12)$$

Aplicando multiplicadores de Lagrange, definimos la función:

$$L(u, \lambda, \mu) = u' \mathbf{M} \mathbf{V} \mathbf{M} u - \lambda(u' \mathbf{M} u - 1) - \mu u' \mathbf{M} u_1.$$

Derivando con respecto a u se tiene:

$$\begin{aligned} \frac{\partial}{\partial u} L(u, \lambda, \mu) &= \frac{\partial(u' \mathbf{M} \mathbf{V} \mathbf{M} u)}{\partial u} - \frac{\partial(\lambda(u' \mathbf{M} u - 1))}{\partial u} - \frac{\partial(\mu u' \mathbf{M} u_1)}{\partial u} \\ &= 2 \mathbf{M} \mathbf{V} \mathbf{M} u - 2 \lambda \mathbf{M} u - \mu \mathbf{M} u_1. \end{aligned}$$

Igualando a cero esta expresión:

$$2 \mathbf{M} \mathbf{V} \mathbf{M} u - 2 \lambda \mathbf{M} u - \mu \mathbf{M} u_1 = 0. \quad (3.13)$$

Multiplicando a la izquierda por u_1' tenemos:

$$2 u_1' \mathbf{M} \mathbf{V} \mathbf{M} u - 2 \lambda u_1' \mathbf{M} u - \mu u_1' \mathbf{M} u_1 = 0. \quad (3.14)$$

De (3.12) se tiene que

$$u_1' \mathbf{M} u = 0,$$

y como u_1 es solución de (3.10), se cumple lo siguiente:

$$u_1' \mathbf{M} u_1 = 1.$$

Reemplazando estos resultados en (3.14) tenemos:

$$2 u_1' \mathbf{M} \mathbf{V} \mathbf{M} u = \mu.$$

Pero

$$\begin{aligned} u_1' \mathbf{M} \mathbf{V} \mathbf{M} u &= (u_1' \mathbf{M} \mathbf{V} \mathbf{M} u)' = u' \mathbf{M} \mathbf{V} \mathbf{M} u_1 \\ &= u' \mathbf{M} \lambda_1 u_1 = \lambda_1 u' \mathbf{M} u_1 = 0. \end{aligned}$$

Por lo tanto $\mu = 0$ y reemplazando el valor de μ en (3.13) se tiene:

$$2\mathbf{M}\mathbf{V}\mathbf{M}u - 2\lambda\mathbf{M}u = 0,$$

de donde:

$$\mathbf{V}\mathbf{M}u = \lambda u$$

Por lo tanto el vector $u_2 \in \mathbb{R}^p$ que maximiza $I_{\Delta u^\perp}$ es un autovector unitario asociado al autovalor λ_2 de $\mathbf{V}\mathbf{M}$, el cual es menor que λ_1 pero mayor que los otros autovalores de dicha matriz.

3.3.5. Búsqueda del plano P del subespacio W_q más cercano a \mathcal{N}

Ahora vamos a buscar cual es el plano P y, en general, cual es el subespacio q -dimensional W_q más cercano a todos los x_i , el único sobre el cual la inercia relativa a g de la nube proyectada sobre W_q , es decir $I_{W_q^\perp}$ es máximo.

- Se debe mostrar que los subespacios buscados son encapsulados: sea W_k un subespacio k -dimensional tal que $I_{W_k^\perp}$ es máxima, definimos un subespacio $k+1$ -dimensional W_{k+1} tal que $I_{W_{k+1}^\perp}$ sea máxima por la suma directa $W_{k+1} = W_k \oplus \Delta\nu$, donde $\Delta\nu \subset W_k^\perp$ tal que $I_{\Delta\nu^\perp}$ sea máxima.
- Si $(u_l)_{l=1}^p$ es una base M -ortonormal de \mathbb{R}^p formada por los autovectores de la matriz $\mathbf{V}\mathbf{M}$, se muestra que el subespacio buscado W_q es generado por los autovectores u_1, \dots, u_q asociados a los mayores q autovalores de $\mathbf{V}\mathbf{M}$.

Como un ejemplo, se demostró que el plano P cercano a todos los x_i es generado por los autovectores de M -norma u_1 y u_2 de $\mathbf{V}\mathbf{M}$ asociados a los dos mayores autovalores λ_1 y λ_2 . Sabemos que P contiene g y que es generado por Δu_1 y un eje $\Delta\nu$, generado por un vector ν de M -norma ortogonal a u_1 tal que $I_{\Delta\nu^\perp}$ es máxima.

En la base ortonormal $(u_l)_{l=1}^p$, resulta $\nu = \sum_{l=1}^p \alpha_l u_l$, el problema ahora es buscar α_l tal que:

$$I_{\Delta\nu^\perp} = \nu' \mathbf{M}\mathbf{V}\mathbf{M}\nu = \sum_{l=1}^p (\alpha_l)^2 \lambda_l$$

sea máxima, con las condiciones:

$$\begin{cases} \|\nu\|_M^2 = 1 = \sum_{l=1}^p (\alpha_l)^2 \\ M(u_1, \nu) = \alpha_1 = 0. \end{cases}$$

Como todos los autovalores son tales que

$$\lambda_1 \geq \dots \geq \lambda_p \geq 0,$$

se tiene:

$$I_{\Delta\nu^\perp} = \sum_{l=2}^p (\alpha_l)^2 \lambda_l \leq \lambda_2$$

donde $I_{\Delta\nu^\perp}$ es máximo e igual a λ_2 si $\alpha_2^2 = 1$ y $\alpha_1 = \alpha_3 = \dots = \alpha_p = 0$, es decir ν coincide con cualquiera de los dos autovectores, u_2 o $-u_2$, de \mathbf{VM} asociados al autovalor λ_2 . Al eje Δu_2 se le asocia el caracter $c^2 \in \mathbb{R}^n$ cuyas coordenadas son las abscisas de las proyecciones de los x_i sobre el eje Δu_2 , esto es:

$$c^2 = \begin{pmatrix} c_1^2 \\ \vdots \\ c_i^2 \\ \vdots \\ c_n^2 \end{pmatrix} = \mathbf{X} \mathbf{M} u_2.$$

Calculamos:

$$\overline{c^2} = \sum_{i=1}^n p_i c_i^2 = \sum_{i=1}^n p_i x_i' \mathbf{M} u_2 = g' \mathbf{M} u_2 = 0.$$

Por lo tanto el caracter c^2 está centrado y

$$\text{Var } c^2 = \sum_{i=1}^n p_i (c_i^2)^2 = (c^2)' \mathbf{D}_n c^2 = u_2' \mathbf{M} \mathbf{X}' \mathbf{D}_n \mathbf{X} \mathbf{M} u_2 = I_{\Delta u_2^\perp} = \lambda_2.$$

Por lo tanto:

$$(c^2)' \mathbf{D}_n c^1 = u_2' \mathbf{M} \mathbf{X}' \mathbf{D}_n \mathbf{X} \mathbf{M} u_1 = \lambda_1 u_2' \mathbf{M} u_1 = 0$$

tal que c^2 y c^1 tienen covarianza nula, por lo tanto no están correlacionados.

Vamos a calcular I_{P^\perp} , la proyección de la inercia de la nube \mathcal{N} sobre P :

$$I_{P^\perp} = \sum_{i=1}^n p_i \|\alpha_i\|_M^2 = \sum_{i=1}^n p_i (\|\delta_i\|_M^2 + \|\gamma_i\|_M^2) = I_{\Delta u_1^\perp} + I_{\Delta u_2^\perp} = \lambda_1 + \lambda_2.$$

Luego si $I_{P^\perp} = \text{Traza} \mathbf{V} \mathbf{M} = I_g$, la nube \mathcal{N} está sobre el plano P y tenemos una representación fácil de mostrar. De otra forma construimos W_3 , subespacio 3-dimensional generado por u_1, u_2 y u_3 y así consecutivamente.

Los ejes principales son generados por los autovectores u_j de M -norma, asociados a los autovalores λ_j de la matriz $\mathbf{V} \mathbf{M}$. El número k de autovalores diferentes de cero es igual al rango de \mathbf{X} , en general $p < n$, por lo tanto k es igual al número de caracteres linealmente independientes entre x^1, \dots, x^p , casi siempre $k = p$. Supongamos que $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > 0$ y $\lambda_{k+1} = \dots = \lambda_p = 0$, recordando que $I_g = \text{Traza}(\mathbf{V} \mathbf{M}) = \sum_{j=1}^p \lambda_j$.

Para $q = 1, \dots, k$, el subespacio W_q mas cercano a la nube \mathcal{N} , es aquel cuya inercia de la proyección de \mathcal{N} sobre W_q relativa a g es máxima, contiene a g y es definido por los autovectores u_1, \dots, u_q de $\mathbf{V} \mathbf{M}$ asociados a los q mayores autovalores $\lambda_1, \dots, \lambda_q$. Tenemos:

$$\begin{aligned} \mathbf{V} \mathbf{M} u_j &= \lambda_j u_j \quad j = 1, \dots, q \\ I_{\Delta u_j} &= \lambda_j \quad j = 1, \dots, q \end{aligned}.$$

Y

$$I_{W_q} = \sum_{j=1}^q I_{\Delta u_j} = \sum_{j=1}^q \lambda_j.$$

Además:

$$\mathcal{N} \subset W_q \Leftrightarrow I_{W_q} = \text{Traza}(\mathbf{V} \mathbf{M}) \Leftrightarrow q = k.$$

3.3.6. Calidad de la representación de las unidades y contribución de una unidad a la construcción de un eje

Se denomina *calidad global* de la representación de \mathcal{N} por su proyección sobre W_q al cociente:

$$\frac{\sum_{j=1}^q \lambda_j}{\text{Traza}(\mathbf{V} \mathbf{M})}.$$

Este cociente mide la porción de la inercia explicada por W_q , que es la porción de la inercia total de \mathcal{N} relativa a g , considerada cuando cada unidad x_i es sustituida por su proyección α_i sobre W_q .

Se denomina *calidad de representación de la unidad x_i por α_i* , al valor:

$$\frac{\|\alpha_i\|^2}{\|x_i\|^2} = \cos^2 \theta.$$

Como los ejes $\Delta u_1, \dots, \Delta u_q$ forman una base M -ortonormal para W_q , la calidad de la representación de x_i sobre W_q es la suma de las calidades de representación de x_i sobre cada eje $\Delta u_1, \dots, \Delta u_q$.

Como ejemplo, sea $W_2 = P$ generado por u_1 y u_2 , se tiene:

$$\frac{\|\alpha_i\|^2}{\|x_i\|^2} = \frac{(c_i^1)^2}{\|x_i\|^2} + \frac{(c_i^2)^2}{\|x_i\|^2}.$$

La contribución de la unidad x_i a la formación del eje Δu_j es medida por su aporte en el cálculo de $I_{\Delta u_j} = \sum_{\min}^{\max} p_k (c_k^j)^2 = \lambda_j$, esto es

$$\frac{p_i (c_i^j)^2}{\lambda_j}.$$

La posición del eje Δu_j es determinada por las unidades cuya contribución es alta, una vez que las condiciones de ortogonalidad con los ejes con menor índice son tomadas en cuenta.

3.4. ACP en el caso presencia/ausencia

Utilizando la función FactoMineR¹ del R a la tabla de datos de Ellenberg se tiene la siguiente gráfica:

En la figura 3.1 se observa $0,2181 \leq \text{Dim } 1 \leq 0,8147$ y $-0,49279 \leq \text{Dim } 2 \leq 0,64689$.

3.5. ACP con técnicas R y Q

En esta sección se considera el punto de vista de Orloci (1966).

¹Para mas detalles revisar Apéndice A.

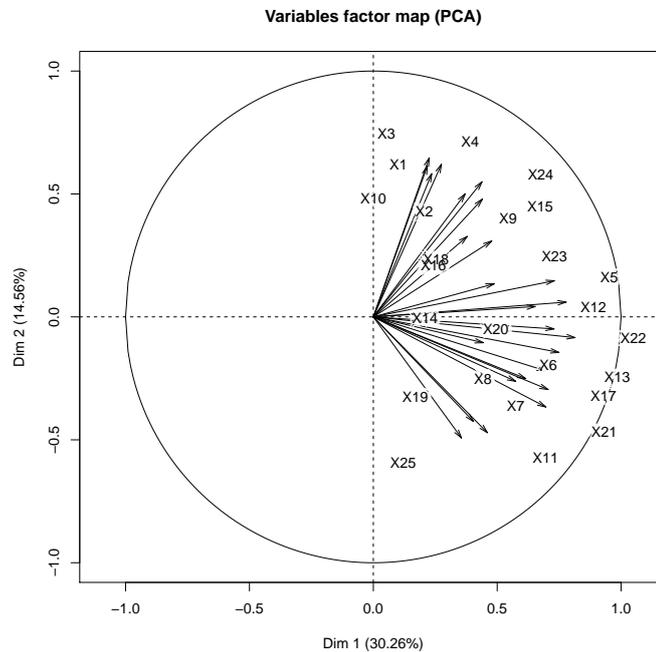


Figura 3.1: Representación en dos dimensiones usando el comando FactoMineR

El análisis de componentes principales se estudia como una técnica de construcción de ejes para lograr un ordenamiento eficiente de unidades. Tal ordenamiento se puede lograr sobre la base de una matriz de coeficientes la cual define las correlaciones entre las características (análisis tipo R) o las similitudes entre las unidades (análisis tipo Q).

Cuando hay mas unidades que características se recomienda usar el análisis tipo R, también se puede usar el análisis tipo Q. Ambos tipos de análisis llegan al mismo conjunto de resultados, dado que se analizan los coeficientes apropiados. Cuando se van a agregar nuevas muestras a un ordenamiento ya realizado el análisis tipo R es el apropiado.

El análisis tipo R se considera en este caso en conexión con una matriz simétrica:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdot & \cdot & \cdots & \cdot \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

cuyos elementos definen las correlaciones o las varianza-covarianza entre las N especies. Una generalización de estudios tipo Q donde los elementos de \mathbf{A} definen las similitudes entre las muestras se hara a continuación. En adelante las unidades a ser ordenadas son muestras de vegetación. Los elementos de \mathbf{A} son los coeficientes de las variables en la ecuación (G) de un hiperelipsoide cuyo primer diámetro principal (o eje) corresponde a la variación máxima con el grupo de puntos. La forma corta de esta ecuación se puede escribir sobre la base de los puntajes de especies originales como:

$$G = \sum_{j,h=1}^N a_{jh} X_j X_h.$$

1 Análisis tipo R

El problema del análisis consiste en hallar los coeficientes de transformación (\mathbf{B}) los cuales relacionan los ejes de las características originales (X_1, \dots, X_N) y los ejes principales (F_1, \dots, F_N) del hiperelipsoide, de la forma:

$$\begin{array}{ccccccc} & F_1 & F_2 & \cdots & F_N & & \\ X_1 & b_{11} & b_{12} & \cdots & b_{1N} & & \\ X_2 & b_{21} & b_{22} & \cdots & b_{2N} & \cdot & \\ \cdot & \cdot & \cdot & \cdots & \cdot & & \\ X_N & b_{N1} & b_{N2} & \cdots & b_{NN} & & \end{array}$$

Las nuevas coordenadas de las muestras (F_{ij}) a lo largo de los ejes principales se obtienen del sistema:

$$\begin{aligned} F_{1j} &= b_{11}X_{1j} + b_{21}X_{2j} + \cdots + b_{N1}X_{Nj} \\ \cdot &= \cdots \\ F_{Nj} &= b_{1N}X_{1j} + b_{2N}X_{2j} + \cdots + b_{NN}X_{Nj}. \end{aligned}$$

La ecuación general del hiperelipsoide después de transformaciones se reduce a la siguiente forma simple:

$$G = \sum_{i=1}^N \lambda_i F_i^2.$$

Donde los λ son los autovalores de \mathbf{A} . Los F son no correlacionados. Los λ son todos positivos si la superficie es un elipsoide.

Una transformación de este tipo deja las distancias entre las muestras sin cambios. Por lo tanto se tiene:

$$D_{jh}^2 = \sum_{i=1}^N (X_{ij} - X_{ih})^2 = \sum_{i=1}^N (F_{ij} - F_{ih})^2.$$

2 Derivación de los valores numéricos por los coeficientes de transformación.

Esto requiere la solución de un sistema de ecuaciones en las N variables en cada columna de \mathbf{B} . Las ecuaciones adecuadas se obtienen sobre la base de un teorema algebraico, el cual dice que para cada matriz \mathbf{A} simétrica y real existe una matriz ortogonal \mathbf{B} tal que $\mathbf{B}^T \mathbf{A} \mathbf{B}$ es una matriz diagonal cuyos elementos son los valores propios de \mathbf{A} , esto es:

$$\mathbf{B}^T \mathbf{A} \mathbf{B} = \begin{bmatrix} \lambda_1 & \cdot & \cdot & \cdot \\ \cdot & \lambda_2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \lambda_N \end{bmatrix} = \mathbf{C}.$$

Desde que $\mathbf{B}^T \mathbf{B}$ es una matriz identidad, se tiene $\mathbf{A} \mathbf{B} = \mathbf{B} \mathbf{C}$. Esta ecuación se puede escribir como:

$$\sum_{s=1}^N a_{js} b_{sh} = \lambda_h b_{jh}$$

de la cual desarrollando para $j = 1, 2, \dots, N$, se obtiene el sistema:

$$\begin{aligned} (a_{11} - \lambda_h) b_{1h} + a_{12} b_{2h} + \dots + a_{1N} b_{Nh} &= 0 \\ \dots &= \cdot \\ a_{N1} b_{1h} + a_{N2} b_{2h} + \dots + (a_{NN} - \lambda_h) b_{Nh} &= 0. \end{aligned}$$

De acuerdo a un teorema del álgebra, un sistema de ecuaciones homogéneo como el anterior, tiene solución diferente de cero si el determinante de los coeficientes es cero. La expansión del determinante:

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} - \lambda & \dots & a_{2N} \\ \cdot & \cdot & \dots & \cdot \\ a_{N1} & a_{N2} & \dots & a_{NN} - \lambda \end{vmatrix} = 0$$

nos dá un polinomio de grado N en λ . Las raíces $(\lambda_1, \dots, \lambda_N)$ se hallan de manera decreciente. Los valores de b de la primera columna de \mathbf{B} se obtienen sustituyendo λ_h por el mayor valor λ_1 y resolviendo el sistema homogéneo, la solución es el vector propio de \mathbf{A} correspondiente a λ_1 . Si estos b se escogen tal que se cumpla:

$$b_{11}^2 + b_{21}^2 + \dots + b_{N1}^2 = 1$$

entonces son los cosenos directores² relativos al primer eje principal (F_1) y los ejes de características originales (X_1, \dots, X_N) . Los valores de b en cualquier columna subsiguiente de \mathbf{B} , como cosenos directores, se obtienen del vector propio correspondiente a su valor propio. La transformación a las nuevas coordenadas de las muestras a lo largo de los ejes principales se obtiene utilizando los valores de b de la manera ya discutida.

3 Análisis tipo Q

Los ejes principales del sistema son los diámetros de las elipsoides dado que los coeficientes de similitud usados en el análisis logran una traslación del origen al centro de gravedad del grupo de puntos. Cuando los grupos representan muestras, el centro de gravedad es la muestra promedio con coordenadas:

$$\bar{X}_i = \left(\sum_{j=1}^n X_{ij} \right) / n, \quad i = 1, 2, \dots, N.$$

lo cual representa el promedio aritmético de los puntajes de especies en la j -ésima muestra.

El eje principal que resulta del análisis de los coeficientes de correlación o de la varianza-covarianza en estudios tipo Q no tiene una relación definida a la variación dentro del grupo. El primer eje no coincide con la variación máxima dentro del grupo y los ejes subsiguientes también se desvían de sus posiciones deseadas. Por lo tanto estas medidas no se pueden aplicar en análisis tipo Q sin una pérdida de eficiencia en el ordenamiento. Para obtener un coeficiente de similitud el cual califica en estudios tipo Q, vamos a considerar la expresión:

$$V_j = \left[\sum_{i=1}^N (X_{ij} - \bar{X}_i)^2 \right]^{1/2}.$$

²Los cosenos directores de un vector \vec{b} son los cosenos de los ángulos formados \vec{b} y los ejes coordenados.

Esta expresión define el desplazamiento de la muestra j con respecto al centro de gravedad del grupo, esto es, la longitud del j -ésimo vector posición. El término \bar{X}_i es el promedio aritmético de las i -ésimas especies para todas las muestras.

El producto escalar de cualquier par de vectores posición será:

$$V_j V_h \cos \alpha_{jh} = \frac{1}{2}(V_j^2 + V_h^2 - D_{jh}^2),$$

o directamente de los puntajes de especies:

$$V_j V_h \cos \alpha_{jh} = \sum_{i=1}^N (X_{ij} - \bar{X}_i)(X_{ih} - \bar{X}_i)$$

esta última expresión es el coeficiente de similitud ponderado. Este coeficiente define las relaciones entre las muestras sin alterar su posición espacial original. En el caso de auto comparaciones, el coeficiente de similitud ponderado se reduce a:

$$\sum_{i=1}^N (X_{ij} - \bar{X}_i)^2.$$

El producto escalar calculado como:

$$\sum_{i=1}^N (X_{ij} - \bar{X}_j)(X_{ih} - \bar{X}_h),$$

donde \bar{X}_j es el promedio de todos los puntajes de la muestra j , cuando es dividido por $N - 1$ es la varianza-covarianza entre muestras.

Cuando \mathbf{A} es una matriz cuyos elementos son valores de los coeficientes de similitud ponderado, los ejes principales son orientados de forma que cada uno representa una posible porción máxima de la variación dentro del grupo de puntos. La estandarización de las soluciones (vector propio) correspondiente a λ_k de \mathbf{A} tal que se cumpla:

$$b_{1k}^2 + b_{2k}^2 + \dots + b_{nk}^2 = \lambda_k$$

y luego los b se pueden usar directamente como las coordenadas de las muestras en el k -ésimo eje principal, por ejemplo $b_{1k} = F_{k1}$ es la coordenada de la primera muestra sobre el k -ésimo eje principal.

4 La eficiencia de los ejes principales.

La eficiencia del k -ésimo eje principal con respecto a la distancia entre muestras es expresada por la relación:

$$\frac{\sum_{j,h=1}^n d_{kjh}^2}{\sum_{j,h=1}^n D_{jh}^2}$$

donde $d_{kjh} = F_{kj} - F_{kh}$ es la diferencia de las coordenadas de las muestras j y h en el k -ésimo eje principal. La relación de las distancias es igual a la relación de las raíces:

$$\frac{\lambda_k}{\sum_{j=1}^n \lambda_j},$$

si el origen de coordenadas coincide con el centro de gravedad del grupo de puntos.

Dado que:

$$\sum_{j=1}^n F_{kj}^2 = \lambda_k \text{ y } \sum_{j=1}^n V_j^2 = \sum_{j=1}^n \lambda_j = \sum_{j=1}^n a_{jj},$$

La relación de las raíces es siempre igual a la relación de los vectores posición:

$$\frac{\sum_{j=1}^n F_{kj}^2}{\sum_{j=1}^n V_j^2} = \frac{\sum_{j=1}^n F_{kj}^2}{\sum_{j=1}^n a_{jj}}.$$

Donde $\sum_{j=1}^n a_{jj}$ es la suma de los elementos de la diagonal de \mathbf{A} .

Cuando se usa un coeficiente de similitud inadecuado, como por ejemplo los coeficientes de correlación en estudios tipo Q, la traslación del origen al centro de gravedad no es lograda, la relación de raíces es muy alta sobre el primer eje principal, pero la relación de las distancias es muy bajo. Tal eje principal no es una buena representación de la distancia entre muestras, por lo tanto es de poca utilidad en ordenamiento ecológico.

Orloci hizo un análisis de la vegetación de la duna de Newborough, Warren, Anglesey. Una descripción detallada de la región se puede hallar en Ranwell (1958, 1959, 1960), los datos de observación fueron las frecuencias de las diferentes especies.

El muestreo se realizó mediante 131 parcelas de muestra de 2 m x 2 m cada una, distribuidas de manera aleatoria dentro de la reserva natural nacional. Se determinaron las frecuencias de disparo para todas las especies encontradas en cincuenta cuadrantes aleatorios de 20 cm x 20 cm dentro de cada parcela de muestra. Un total de 100 parcelas se usa en la ordenación; se excluyen zonas de mareas, pantanos, y zonas desocupadas. Se omiten las treinta y siete especies menos frecuentes, lo que reduce la cantidad utilizada en los cálculos a 101 especies.

Capítulo 4

Análisis de coordenadas principales (ACoP)

En este capítulo se considera el punto de vista de Jolliffe (2002).

El análisis de coordenadas principales es un método de escalamiento u ordenamiento, algunas veces conocido como escalamiento clásico. Fue popularizado por Gower (1966). Torgerson (1952, 1958) discutió ideas similares, pero no notó las relaciones entre el análisis de coordenadas principales y el análisis de componentes principales (ACP) que fueron notadas por Gower. Como el más conocido escalamiento dimensional no métrico Kruskal (1964a) y Kruskal (1964b), la técnica empieza con una matriz de similitudes o disimilitudes entre un conjunto de observaciones, y el objetivo es generar una representación gráfica de dimensión pequeña de los datos de tal forma que las distancias entre los puntos en la gráfica sean aproximadas a las disimilitudes originales. Existen varias técnicas de escalamiento, una gran cantidad de ellas están detalladas en Cox y Cox (2000).

El punto de partida (una matriz $n \times n$ de (dis)similitudes) del ACoP es diferente que el del ACP el cual usualmente empieza con una matriz de datos $n \times p$. Sin embargo en casos especiales discutidos mas adelante las dos técnicas generan precisamente la misma representación de dimensión baja. Además el ACP se puede usar para hallar la configuración inicial para los algoritmos iterativos asociados al escalamiento multidimensional no métrico. Antes de mostrar la equivalencia entre ACP y ACoP, primero necesitamos describir el ACoP en detalle.

Supongamos que \mathbf{T} es una matriz $n \times n$ semidefinida positiva simétrica de similitudes entre un conjunto de n observaciones. De la descomposición espectral de \mathbf{T}

tenemos:

$$\mathbf{T} = \tau_1 \mathbf{b}_1 \mathbf{b}'_1 + \tau_2 \mathbf{b}_2 \mathbf{b}'_2 + \cdots + \tau_n \mathbf{b}_n \mathbf{b}'_n \quad (4.1)$$

donde $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_n$ son los valores propios de \mathbf{T} y $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$ son sus respectivos autovectores. También se puede escribir de la forma:

$$\mathbf{T} = \mathbf{c}_1 \mathbf{c}'_1 + \mathbf{c}_2 \mathbf{c}'_2 + \cdots + \mathbf{c}_n \mathbf{c}'_n, \quad (4.2)$$

donde

$$\mathbf{c}_j = \tau_j^{1/2} \mathbf{b}_j, \quad j = 1, 2, \dots, n.$$

Ahora se consideran las n observaciones como puntos en un espacio n -dimensional con la j -ésima coordenada para la i -ésima observación igual a c_{ij} , el i -ésimo elemento de \mathbf{c}_j . Con esta interpretación geométrica de las n observaciones, la distancia euclidiana entre la i -ésima y h -ésima observación es:

$$\begin{aligned} \Delta_{hi}^2 &= \sum_{j=1}^n (c_{hj} - c_{ij})^2 \\ &= \sum_{j=1}^n c_{hj}^2 + \sum_{j=1}^n c_{ij}^2 - 2 \sum_{j=1}^n c_{hj} c_{ij}. \end{aligned}$$

Pero de (4.2), el (h, i) -ésimo elemento de \mathbf{T} se puede escribir como:

$$t_{hi} = \sum_{j=1}^n c_{hj} c_{ij}, \quad h, i = 1, 2, \dots, n.$$

Luego

$$\Delta_{hi}^2 = t_{hh} + t_{ii} - 2t_{hi}.$$

El ACoP intenta determinar el mejor ajuste q -dimensional ($q < n$) como una aproximación de la representación n -dimensional definida anteriormente. Una representación de coordenadas principales q -dimensional es dada por las coordenadas de las observaciones con respecto a las primeras q componentes principales. El ACoP por lo tanto consiste de dos etapas, ambas implican calcular los autovalores y autovectores de matrices $n \times n$:

- (i) Calcular los autovectores $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n$ de \mathbf{T} , normalizados para tener longitudes iguales a sus respectivos valores propios, y representar las n observaciones como puntos en un espacio n -dimensional con coordenadas c_{ij} para la i -ésima observación en la j -ésima dimensión.

- (ii) Determinar los componentes principales para el conjunto de datos en n dimensiones definidos en (i), y calcular las coordenadas de las n observaciones con respecto a las componentes principales.

Si los vectores \mathbf{c}_j definidos en la primera etapa cumplen $\sum_{i=1}^n c_{ij} = 0$ entonces la matriz de covarianza que se calcula en la segunda etapa será proporcional a $\mathbf{C}'\mathbf{C}$, donde \mathbf{C} es una matriz $n \times n$ con columnas \mathbf{c}_j , $j = 1, 2, \dots, n$. Pero:

$$\mathbf{c}'_j \mathbf{c}_k = \begin{cases} \tau_j, & j = k \\ 0, & j \neq k \end{cases}$$

así como los autovectores en la descomposición espectral (4.1) tienen la propiedad:

$$\mathbf{b}'_j \mathbf{b}_k = \begin{cases} 1_j, & j = k \\ 0, & j \neq k \end{cases}$$

y

$$\mathbf{c}_j = \tau_j^{1/2} \mathbf{b}_j, \quad j = 1, 2, \dots, n.$$

Por lo tanto la matriz $\mathbf{C}'\mathbf{C}$ es diagonal, con elementos τ_j , $j = 1, 2, \dots, n$, y las primeras q coordenadas principales de las n observaciones son simplemente los valores de c_{ij} para $i = 1, 2, \dots, n$, $j = 1, 2, \dots, q$. Cuando $\sum_{i=1}^n c_{ij} = 0$, la etapa (ii) es innecesaria.

En general, a pesar de que una matriz de similitud \mathbf{T} no siempre nos lleva a que $\sum_{i=1}^n c_{ij} = 0$, esta propiedad se puede conseguir fácilmente reemplazando \mathbf{T} por una matriz de similitud modificada. En esta modificación, t_{hi} es reemplazado por

$$t_{hi} - \bar{t}_h - \bar{t}_i + \bar{t},$$

donde:

- \bar{t}_h denota la media de los elementos de la h -ésima fila (o columna, dado que \mathbf{T} es simétrica) de \mathbf{T} .
- \bar{t} es la media de todos los elementos de \mathbf{T} .

Esta matriz modificada cumple que:

$$\sum_{i=1}^n c_{ij} = 0$$

y da el mismo valor de Δ_{hi}^2 para cada par de observaciones como lo hace \mathbf{T} (Gower, 1966). Por lo tanto se puede reemplazar la segunda etapa del ACoP por una modificación inicial de \mathbf{T} , para cualquier matriz de similitudes \mathbf{T} .

El ACoP es equivalente a graficar con respecto a los primeros q componentes principales cuando la medida de similitud entre dos puntos es proporcional a $-d_{hi}^2$, donde d_{hi}^2 es la distancia euclidiana al cuadrado entre la h -ésima e i -ésima observación, calculadas de la matriz de datos $n \times p$. Asumiendo que $t_{hi} = -\gamma d_{hi}^2$, donde γ es una constante positiva, entonces si la etapa (i) del ACoP se lleva a cabo, la “distancia” entre un par de puntos en el espacio n -dimensional construido es:

$$\begin{aligned} \Delta_{hi}^2 &= t_{hh} + t_{ii} - 2t_{hi} \\ &= \gamma(-d_{hh}^2 - d_{ii}^2 + 2d_{hi}^2) \\ &= 2\gamma d_{hi}^2. \end{aligned}$$

Los dos primeros términos se anulan debido a que la distancia euclidiana de un punto a si mismo es cero. Por lo tanto, aparte de un posible re-escalamiento si se toma $\gamma = 1/2$, la primera etapa del ACoP reproduce correctamente las posiciones relativas de las n observaciones, las cuales están en un subespacio p dimensional de un espacio n -dimensional, así que el siguiente ACP en la etapa (ii) da el mismo resultado que el ACP en los datos originales.

Casos de interés

Hay dos casos relacionados de interés. El primero considera la situación donde todas las variables son binarias. Una medida de similitud común usada entre los objetos h e i es la proporción de las p variables para las cuales h e i tienen el mismo valor, y se puede demostrar fácilmente Gower (1966) que esta medida es equivalente a la distancia euclidiana. Por lo tanto, aunque el ACP de datos discretos (y en particular binarios) tiene sus críticas, es equivalente al ACoP con una medida muy razonable de similitud.

El segundo caso especial ocurre cuando los elementos de la matriz de similitud \mathbf{T} son definidos como «covarianzas» entre las observaciones, de modo que \mathbf{T} es proporcional a $\mathbf{X}\mathbf{X}'$, donde \mathbf{X} es una matriz $n \times p$ centrada en columnas, cuyo (i, j) -ésimo elemento es el valor de la j -ésima variable, medida sobre su medida \bar{x}_j , para la i -ésima observación. En este caso la (h, i) -ésima similitud es, salvo una constante

$$t_{hi} = \sum_{j=1}^p x_{hj}x_{ij}$$

y la distancia entre los puntos en el espacio n -dimensional construido en la primera etapa del ACoP son:

$$\begin{aligned} \Delta_{hi}^2 &= t_{hh} + t_{ii} - 2t_{hi} \\ &= \sum_{j=1}^p x_{hj}^2 + \sum_{j=1}^p x_{ij}^2 - 2 \sum_{j=1}^p x_{hj}x_{ij} \\ &= \sum_{j=1}^p (x_{hj} - x_{ij})^2 = d_{hi}^2, \end{aligned}$$

esto es, las distancias euclidianas entre las observaciones usando las p variables originales. Como antes, el ACP en la segunda etapa del ACoP da el mismo resultado que el ACP sobre los datos originales. Note, sin embargo, que $\mathbf{X}\mathbf{X}'$ no es una matriz de similitud muy obvia. Para una matriz de covarianzas entre observaciones es más natural usar una versión de la matriz \mathbf{X} centrada en filas.

Aún en casos donde el ACP y el ACoP generan gráficas en dos dimensiones equivalentes, existe una diferencia, es decir que en el ACoP no hay vectores de coeficientes definiendo los ejes en función de las variables originales. Esto significa que, a diferencia del ACP, los ejes en el ACoP no se pueden interpretar, a menos que también se haya hecho el correspondiente ACP.

A la equivalencia entre el ACP y el ACoP en las circunstancias descritas se le denomina *dualidad* entre las dos técnicas de Gower (1966). Las técnicas son duales en el sentido que el ACP opera sobre una matriz de similitudes entre variables, mientras que el ACoP opera sobre una matriz de similitudes entre las observaciones (individuales), pero ambos llevan a resultados equivalentes.

En resumen, el ACoP da una representación de los datos en una dimensión baja cuando los datos son dados en la forma de una matriz de similitud o disimilitud. Como

se puede usar con cualquier forma de una matriz de similitud o disimilitud, esto es, en cierto sentido, mas poderoso que el ACP, a la vez que extiende su aplicabilidad el ACP Gower (1967). Sin embargo el ACP tiene muchos usos además de representar datos gráficamente, el cual es el principal propósito del ACoP.

Excepto en los casos especiales discutidos anteriormente, el ACoP no tiene relación directa con el ACP.

En la mayoría de los casos, si se dispone de los datos de la forma de una matriz $n \times p$ con p variables medidas para las n observaciones, no hay ventaja en hacer el ACoP en lugar del ACP, a menos que por alguna razón se considere una medida de similitud distinta que la distancia euclidiana. Sin embargo, ocurre una excepción cuando $n < p$, especialmente si n es mucho más pequeño que p lo que se presenta para algunos tipos de datos químicos, meteorológicos y biológicos. Como el ACoP y el ACP hallan valores propios de una matriz $n \times n$ y $n \times p$ respectivamente, el análisis dual basado en componentes principales tendrá ventajas computacionales en tales casos.

A continuación se muestran las diferentes salidas en el programa R , considerando el coeficiente de asociación de Braun-Blanquet

$$S_1 < 0, 1 > = \frac{a}{\max \{(a + b), (a + c)\}}.$$

Las descripciones de las rutinas se muestran en el apéndice B.

4.1. Cmdscale (stats)

En este caso cuando ejecutamos el comando para $k = 24$ sale una advertencia:

Salida

```
> mds1 = cmdscale(misdatos,k=24,eig = TRUE);
Warning message:
In cmdscale(misdatos, k = 24, eig = TRUE) :
only 17 of the first 24 eigenvalues are > 0
```

Al revisar la estructura tenemos:

Estructura

```
> str(mds1)
List of 5
 $ points: num [1:25, 1:17] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : NULL
 .. ..$ : NULL
 $ eig   : num [1:25] 1.164 0.461 0.322 0.262 0.252 ...
 $ x     : NULL
 $ ac    : num 0
 $ GOF   : num [1:2] 0.947 1
```

De donde observamos las primeras 17 componentes (correspondientes a los valores propios positivos).

Los 25 valores propios.

Vamos a visualizar las dos primeras coordenadas y los valores propios

Coordenadas y valores propios

```
> mds1$points[,1:2]
[,1]      [,2]
[1,] -0.33438754 -0.20248243
[2,] -0.19157178  0.06152301
...
[25,]  0.31266121 -0.07321306
```

```

> mds1$eig
[1] 1.163612e+00 4.605353e-01 3.216755e-01 2.624594e-01 2.516570e-01
[6] 2.131520e-01 1.907049e-01 1.429427e-01 1.193087e-01 1.167721e-01
[11] 9.443902e-02 8.416027e-02 6.238981e-02 4.621377e-02 4.063409e-02
[16] 3.261181e-02 1.928208e-02 -1.942890e-16 -3.161565e-03 -4.179202e-03
[21] -1.537939e-02 -1.616276e-02 -4.165521e-02 -5.273388e-02 -6.805512e-02

```

se observa que los primeros 17 son positivos.

Para resolver esto, debería que escribirse “add=TRUE”

La representación en dos dimensiones es:

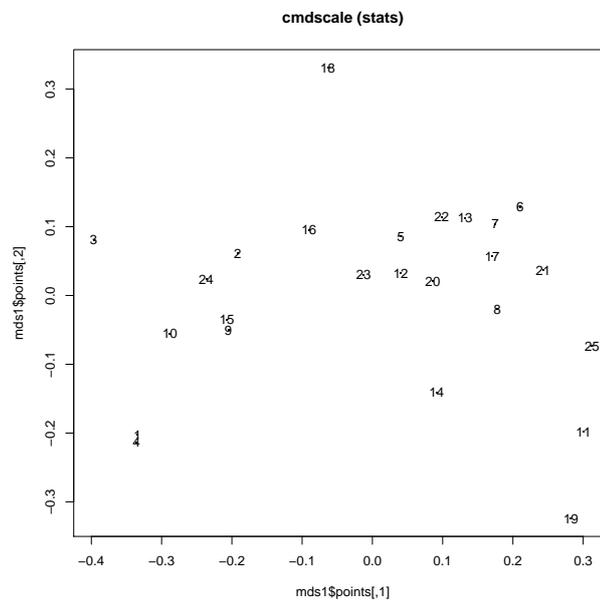


Figura 4.1: Representación en dos dimensiones usando el comando cmdscale

En la figura 4.1 se observa $-0,3967053 \leq \text{Coordenada 1} \leq 0,3126612$ y $-0,3240137 \leq \text{Coordenada 2} \leq 0,3311205$.

4.2. wcmdscale (vegan)

En este caso cuando ejecutamos el comando para $k = 24$ y vemos su estructura

Estructura

```
> mds2 = wcmdscale(misdatos, k=24, eig=TRUE, w=rep(1,25));
> str(mds2)
List of 9
 $ points : num [1:25, 1:17] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:25] "1" "2" "3" "4" ...
 .. ..$ : chr [1:17] "Dim1" "Dim2" "Dim3" "Dim4" ...
 $ eig    : num [1:24] 1.164 0.461 0.322 0.262 0.252 ...
 $ x      : NULL
 $ ac     : logi NA
 $ add    : logi FALSE
 $ GOF    : num [1:2] 0.947 1
 $ weights: num [1:25] 1 1 1 1 1 1 1 1 1 1 1 ...
 $ negaxes: num [1:25, 1:7] -0.0029 0.01314 -0.00282 0.00525 -0.02726 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:25] "1" "2" "3" "4" ...
 .. ..$ : NULL
 $ call   : language wcmdscale(d = misdatos, k = 24, eig = TRUE, w = rep(1,
- attr(*, "class")= chr "wcmdscale"
```

\$ points muestra las primeras 17 componentes.

\$ eig muestra los 25 valores propios.

\$ weights los pesos (en este caso hemos considerado 1)

\$ negaxes muestra las 7 componentes (correspondientes a los valores propios negativos)

Vamos a visualizar las dos primeras coordenadas (correspondiente a los valores propios positivos)

Coordenadas

```
> mds2$points[,1:2]
Dim1      Dim2
```

```
1 -0.33438754 -0.20248243
2 -0.19157178  0.06152301
...
25  0.31266121 -0.07321306
```

Vamos a visualizar las dos primeras coordenadas (correspondiente a los valores propios negativos) (aca se usa el *negaxes*)

Coordenadas correspondientes a los valores propios negativos

```
> mds2$negaxes[,1:2]
[,1]          [,2]
1 -0.0028986494  4.881946e-03
2  0.0131425182 -1.418539e-02
...
25 -0.0042714933 -6.362757e-03
```

Vamos a mostrar la tabla de valores propios, valores propios entre la Inercia y el acumulado de esta última.

Tabla de valores propios

```
> resvalprop
col1          col2          col3
[1,]  1.163611539  0.3401156683  0.3401157
[2,]  0.460535329  0.1346113164  0.4747270
...
[17,]  0.019282085  0.0056360210  1.0588465
[18,] -0.003161565 -0.0009241039  1.0579224
...
[24,] -0.068055121 -0.0198920450  1.0000000
```

La representación en dos dimensiones para los valores propios positivos es:

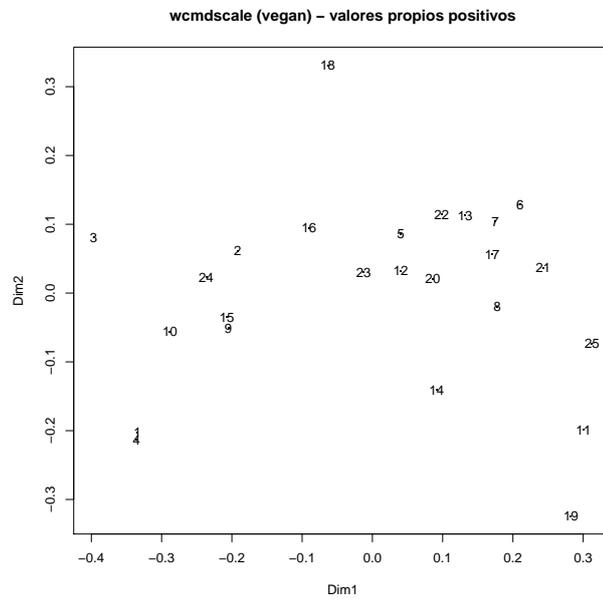


Figura 4.2: Representación en dos dimensiones usando el comando wcmdscale considerando los valores propios positivos

En la figura 4.2 se observa $-0,3967053 \leq \text{Coordenada 1} \leq 0,3126612$ y $-0,3240137 \leq \text{Coordenada 2} \leq 0,3311205$.

La representación en dos dimensiones para los valores propios negativos es:

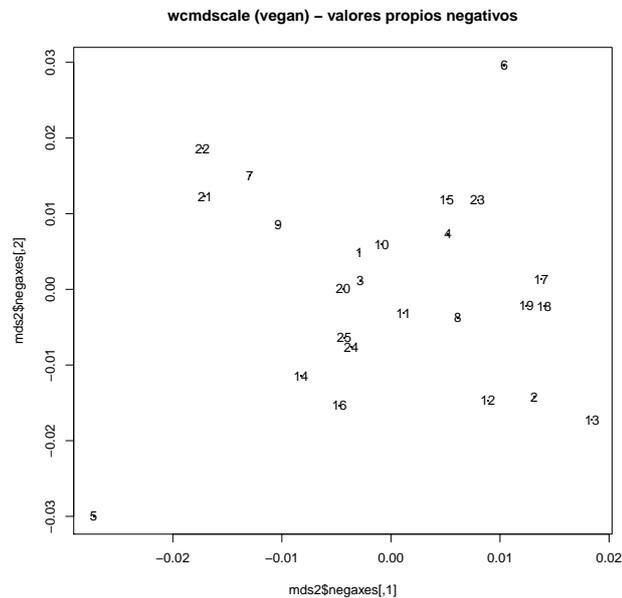


Figura 4.3: Representación en dos dimensiones usando el comando `wcmdscale` considerando los valores propios negativos

En la figura 4.3 se observa $-0,02725678 \leq \text{Coordenada 1} \leq 0,01844134$ y $-0,02996789 \leq \text{Coordenada 2} \leq 0,02959236$.

4.3. pco (ecodist)

En este caso cuando ejecutamos el comando para $k = 24$ y vemos su estructura

Estructura

```
> mds3 = ecodist::pco(as.dist(misdatos),negvals ="rm",dround=0)
> str(mds3)
List of 2
 $ values : num [1:25] 1.232 0.529 0.39 0.331 0.32 ...
 $ vectors: num [1:25, 1:25] -0.279 -0.16 -0.331 -0.281 0.034 ...
```

hemos usado `negvals = "rm"` sino mostraba 8 ceros (a los valores propios negativos los hace cero)

`$ values` muestra los 25 valores propios.

`$vectors` muestra las 25 coordenadas.

Vamos a visualizar las dos primeras coordenadas y los valores propios

Coordenadas y valores propios

```
> mds3$vectors[,1:2]
[,1]      [,2]
[1,] -0.27931836 -0.41038946
[2,] -0.16002246  0.12469424
...
[25,]  0.26117006 -0.14838755
> mds3$values
[1]  1.231667e+00  5.285905e-01  3.897306e-01  3.305145e-01
[5]  3.197121e-01  2.812072e-01  2.587600e-01  2.109979e-01
[9]  1.873638e-01  1.848272e-01  1.624941e-01  1.522154e-01
[13] 1.304449e-01  1.142689e-01  1.086892e-01  1.006669e-01
[17] 8.733721e-02  6.489356e-02  6.387592e-02  5.267573e-02
[21] 5.189237e-02  2.639991e-02  1.532124e-02  6.623546e-17
[25] -2.133590e-16
```

La representación en dos dimensiones es:

En la figura 4.4 se observa $-0,3313732 \leq \text{Coordenada 1} \leq 0,2611701$ y $-0,6567079 \leq \text{Coordenada 2} \leq 0,6711119$.

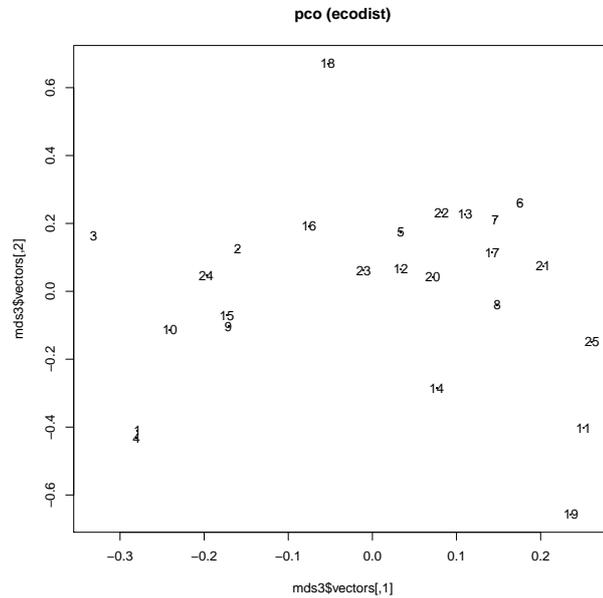


Figura 4.4: Representación en dos dimensiones usando el comando pco del paquete ecodist

4.4. pco (labdsv)

Vamos a visualizar las dos primeras coordenadas y sus valores propios

Coordenadas y valores propios

```
> mds4$points[,1:2]
[,1]      [,2]
[1,] -0.33438754 -0.20248243
[2,] -0.19157178  0.06152301
...
[25,]  0.31266121 -0.07321306
> mds4$eig
[1]  1.163612e+00  4.605353e-01  3.216755e-01  2.624594e-01
[5]  2.516570e-01  2.131520e-01  1.907049e-01  1.429427e-01
[9]  1.193087e-01  1.167721e-01  9.443902e-02  8.416027e-02
[13]  6.238981e-02  4.621377e-02  4.063409e-02  3.261181e-02
[17]  1.928208e-02 -1.942890e-16 -3.161565e-03 -4.179202e-03
```



```
[21] -1.537939e-02 -1.616276e-02 -4.165521e-02 -5.273388e-02
[25] -6.805512e-02
```

La representación en dos dimensiones es: En la figura 4.5 se observa $-0,3967053 \leq$

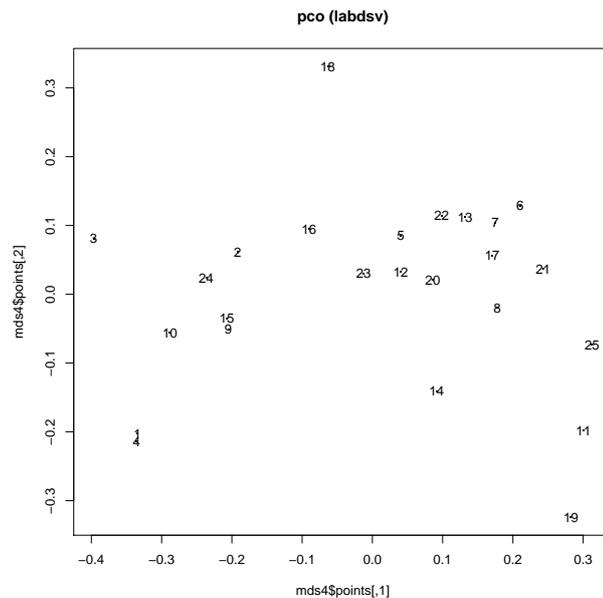


Figura 4.5: Representación en dos dimensiones usando el comando pco del paquete labdsv

Coordenada 1 $\leq 0,3126612$ y $-0,3240137 \leq$ Coordenada 2 $\leq 0,3311205$.

4.5. pcoa (ape)

En este caso cuando ejecutamos el comando y vemos su estructura

Estructura

```
> mds5 = pcoa(misdatos)
> str(mds5)
List of 5
 $ correction: chr [1:2] "none" "1"
 $ note      : chr "No correction was applied to the negative eigenvalues"
```

```

$ values      :'data.frame': 25 obs. of  6 variables:
..$ Eigenvalues   : num [1:25] 1.164 0.461 0.322 0.262 0.252 ...
..$ Relative_eig  : num [1:25] 0.3401 0.1346 0.094 0.0767 0.0736 ...
..$ Rel_corr_eig  : num [1:25] 0.2437 0.1046 0.0771 0.0654 0.0633 ...
..$ Broken_stick  : num [1:25] 0.1624 0.1189 0.0971 0.0827 0.0718 ...
..$ Cum_corr_eig  : num [1:25] 0.244 0.348 0.425 0.491 0.554 ...
..$ Cumul_br_stick: num [1:25] 0.162 0.281 0.378 0.461 0.533 ...
$ vectors      : num [1:25, 1:17] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : NULL
.. ..$ : chr [1:17] "Axis.1" "Axis.2" "Axis.3" "Axis.4" ...
$ trace       : num 3.42
- attr(*, "class")= chr "pcoa"

```

Vamos a visualizar las dos primeras coordenadas y sus valores propios

Coordenadas y tabla de valores propios

```

> mds5$vectors[,1:2]
Axis.1      Axis.2
[1,] -0.33438754 -0.20248243
[2,] -0.19157178  0.06152301
[3,] -0.39670526  0.08083080
[4,] -0.33595817 -0.21357860
[5,]  0.04069364  0.08565242
[6,]  0.21039294  0.12899411
[7,]  0.17476366  0.10404712
[8,]  0.17757421 -0.01985020
[9,] -0.20508787 -0.05066555
[10,] -0.28813406 -0.05574471
...
[16,] -0.09036761  0.09504267
[17,]  0.17072130  0.05662097
[18,] -0.06348408  0.33112052
[19,]  0.28248849 -0.32401370
[20,]  0.08597451  0.02100940

```

```

[21,] 0.24308172 0.03674833
[22,] 0.09879386 0.11437174
[23,] -0.01251359 0.03012790
[24,] -0.23640118 0.02341090
[25,] 0.31266121 -0.07321306

```

```
> mds5$values
```

	Eigenvalues	Relative_eig	Rel_corr_eig	Broken_stick
1	1.163611539	0.3401156683	0.243675038	0.162360500
2	0.460535329	0.1346113164	0.104577238	0.118882240
3	0.321675503	0.0940235421	0.077104973	0.097143109
4	0.262459422	0.0767150881	0.065389562	0.082650356
5	0.251656993	0.0735576120	0.063252391	0.071780790
6	0.213152050	0.0623028815	0.055634507	0.063085138
7	0.190704889	0.0557417304	0.051193523	0.055838761
8	0.142942744	0.0417811833	0.041744178	0.049627581
9	0.119308696	0.0348731131	0.037068378	0.044192799
10	0.116772097	0.0341316827	0.036566533	0.039361881
11	0.094439019	0.0276038772	0.032148118	0.035014055
12	0.084160270	0.0245994695	0.030114553	0.031061485
13	0.062389806	0.0182361123	0.025807448	0.027438297
14	0.046213773	0.0135079688	0.022607153	0.024093815
15	0.040634089	0.0118770655	0.021503259	0.020988225
16	0.032611807	0.0095322074	0.019916117	0.018089675
17	0.019282085	0.0056360210	0.017278942	0.015372283
18	0.000000000	0.0000000000	0.012838652	0.012814739
19	-0.003161565	-0.0009241039	0.012637321	0.010399280
20	-0.004179202	-0.0012215520	0.010421456	0.008110950
21	-0.015379394	-0.0044952914	0.010266474	0.005937037
22	-0.016162756	-0.0047242626	0.005223003	0.003866644
23	-0.041655211	-0.0121755324	0.003031180	0.001890359
24	-0.052733884	-0.0154137525	0.000000000	0.000000000
25	-0.068055121	-0.0198920450	0.000000000	0.000000000
	Cum_corr_eig	Cumul_br_stick		
1	0.2436750	0.1623605		

2	0.3482523	0.2812427
3	0.4253572	0.3783858
4	0.4907468	0.4610362
5	0.5539992	0.5328170
6	0.6096337	0.5959021
7	0.6608272	0.6517409
8	0.7025714	0.7013685
9	0.7396398	0.7455613
10	0.7762063	0.7849232
11	0.8083544	0.8199372
12	0.8384690	0.8509987
13	0.8642764	0.8784370
14	0.8868836	0.9025308
15	0.9083869	0.9235190
16	0.9283030	0.9416087
17	0.9455819	0.9569810
18	0.9584206	0.9697957
19	0.9710579	0.9801950
20	0.9814793	0.9883060
21	0.9917458	0.9942430
22	0.9969688	0.9981096
23	1.0000000	1.0000000
24	1.0000000	1.0000000
25	1.0000000	1.0000000

La representación en dos dimensiones es:

En la figura 4.6 se observa $-0,3967053 \leq \text{Coordenada 1} \leq 0,3126612$ y $-0,3240137 \leq \text{Coordenada 2} \leq 0,3311205$.

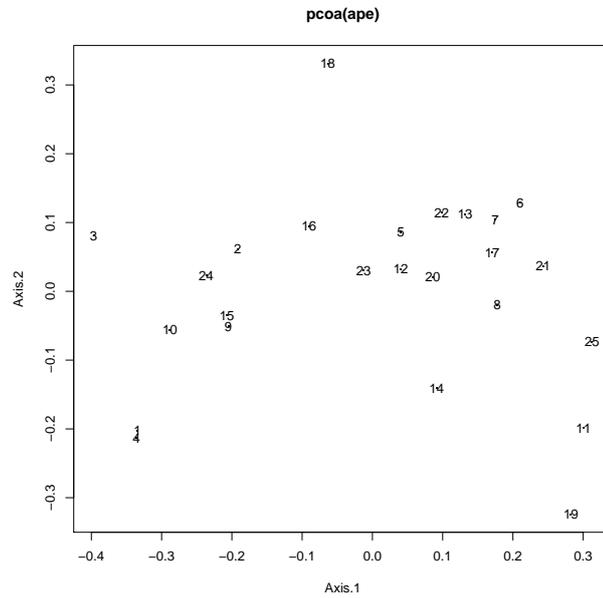


Figura 4.6: Representación en dos dimensiones usando el comando pcoa del paquete ape

4.6. dudi.pco (ade4)

Vamos a visualizar las dos primeras coordenadas y sus valores propios

Coordenadas y valores propios

```
> mds6$li[,1:2]
A1      A2
1 -0.33438754 -0.20248243
2 -0.19157178  0.06152301
3 -0.39670526  0.08083080
4 -0.33595817 -0.21357860
5  0.04069364  0.08565242
6  0.21039294  0.12899411
7  0.17476366  0.10404712
8  0.17757421 -0.01985020
9 -0.20508787 -0.05066555
10 -0.28813406 -0.05574471
```

```

...
16 -0.09036761  0.09504267
17  0.17072130  0.05662097
18 -0.06348408  0.33112052
19  0.28248849 -0.32401370
20  0.08597451  0.02100940
21  0.24308172  0.03674833
22  0.09879386  0.11437174
23 -0.01251359  0.03012790
24 -0.23640118  0.02341090
25  0.31266121 -0.07321306
> mds6$eig
[1] 0.0465444616 0.0184214132 0.0128670201 0.0104983769
[5] 0.0100662797 0.0085260820 0.0076281956 0.0057177098
[9] 0.0047723478 0.0046708839 0.0037775608 0.0033664108
[13] 0.0024955923 0.0018485509 0.0016253636 0.0013044723
[17] 0.0007712834

```

La representación en dos dimensiones es:

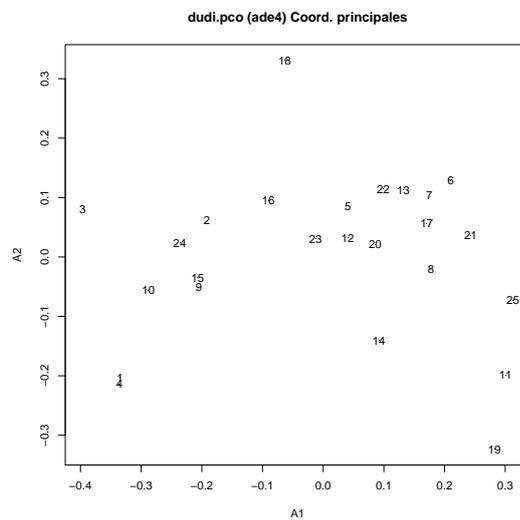


Figura 4.7: Representación en dos dimensiones usando el comando `dudi.pco` del paquete `ade4`

En la figura 4.7 se observa $-0,3967053 \leq \text{Coordenada 1} \leq 0,3126612$ y $-0,3240137 \leq \text{Coordenada 2} \leq 0,3311205$.

Por lo tanto comparando los autovalores:

	cmdscale	wcmdscale	ecodist	labdsv	pcoa	dudi.pco
1	1.163612e+00	1.16361153944945	1.231667e+00	1.163612e+00	1.163611539	0.046544461577978
2	4.605353e-01	0.460535328833342	5.285905e-01	4.605353e-01	0.460535329	0.0184214131533337
3	3.216755e-01	0.321675502832288	3.897306e-01	3.216755e-01	0.321675503	0.0128670201132915
4	2.624594e-01	0.262459422329926	3.305145e-01	2.624594e-01	0.262459422	0.010498376893197
5	2.516570e-01	0.251656992530647	3.197121e-01	2.516570e-01	0.251656993	0.0100662797012259
6	2.131520e-01	0.213152049750738	2.812072e-01	2.131520e-01	0.213152050	0.00852608199002952
7	1.907049e-01	0.190704889004921	2.587600e-01	1.907049e-01	0.190704889	0.00762819556019684
8	1.429427e-01	0.142942744435994	2.109979e-01	1.429427e-01	0.142942744	0.00571770977743975
9	1.193087e-01	0.11930869573068	1.873638e-01	1.193087e-01	0.119308696	0.00477234782922721
10	1.167721e-01	0.116772097089345	1.848272e-01	1.167721e-01	0.116772097	0.0046708838835738
11	9.443902e-02	0.0944390188966929	1.624941e-01	9.443902e-02	0.094439019	0.00377756075586772
12	8.416027e-02	0.0841602701603766	1.522154e-01	8.416027e-02	0.084160270	0.00336641080641506
13	6.238981e-02	0.0623898064236689	1.304449e-01	6.238981e-02	0.062389806	0.00249559225694676
14	4.621377e-02	0.0462137733588927	1.142689e-01	4.621377e-02	0.046213773	0.00184855093435571
15	4.063409e-02	0.0406340894951116	1.086892e-01	4.063409e-02	0.040634089	0.00162536357980447
16	3.261181e-02	0.0326118069740362	1.006669e-01	3.261181e-02	0.032611807	0.00130447227896145
17	1.928208e-02	0.0192820845592492	8.733721e-02	1.928208e-02	0.019282085	0.00077128338236997
18	-1.942890e-16	-0.0031615653688984	6.489356e-02	-1.942890e-16	0.000000000	-
19	-3.161565e-03	-0.00417920167573547	6.387592e-02	-3.161565e-03	-0.003161565	-
20	-4.179202e-03	-0.0153793942674019	5.267573e-02	-4.179202e-03	-0.004179202	-
21	-1.537939e-02	-0.0161627558615155	5.189237e-02	-1.537939e-02	-0.015379394	-
22	-1.616276e-02	-0.0416552110547965	2.639991e-02	-1.616276e-02	-0.016162756	-
23	-4.165521e-02	-0.0527338843329157	1.532124e-02	-4.165521e-02	-0.041655211	-
24	-5.273388e-02	-0.068055121314404	6.623546e-17	-5.273388e-02	-0.052733884	-
25	-6.805512e-02	-	-2.133590e-16	-6.805512e-02	-0.068055121	-

Se observa que las salidas tienen diferentes autovalores pero a pesar de esto las representaciones gráficas bidimensionales tienen la misma forma (figura 4.8). Por lo tanto se elige el `wcmdscale` por que permite conocer las coordenadas principales para los valores propios negativos.

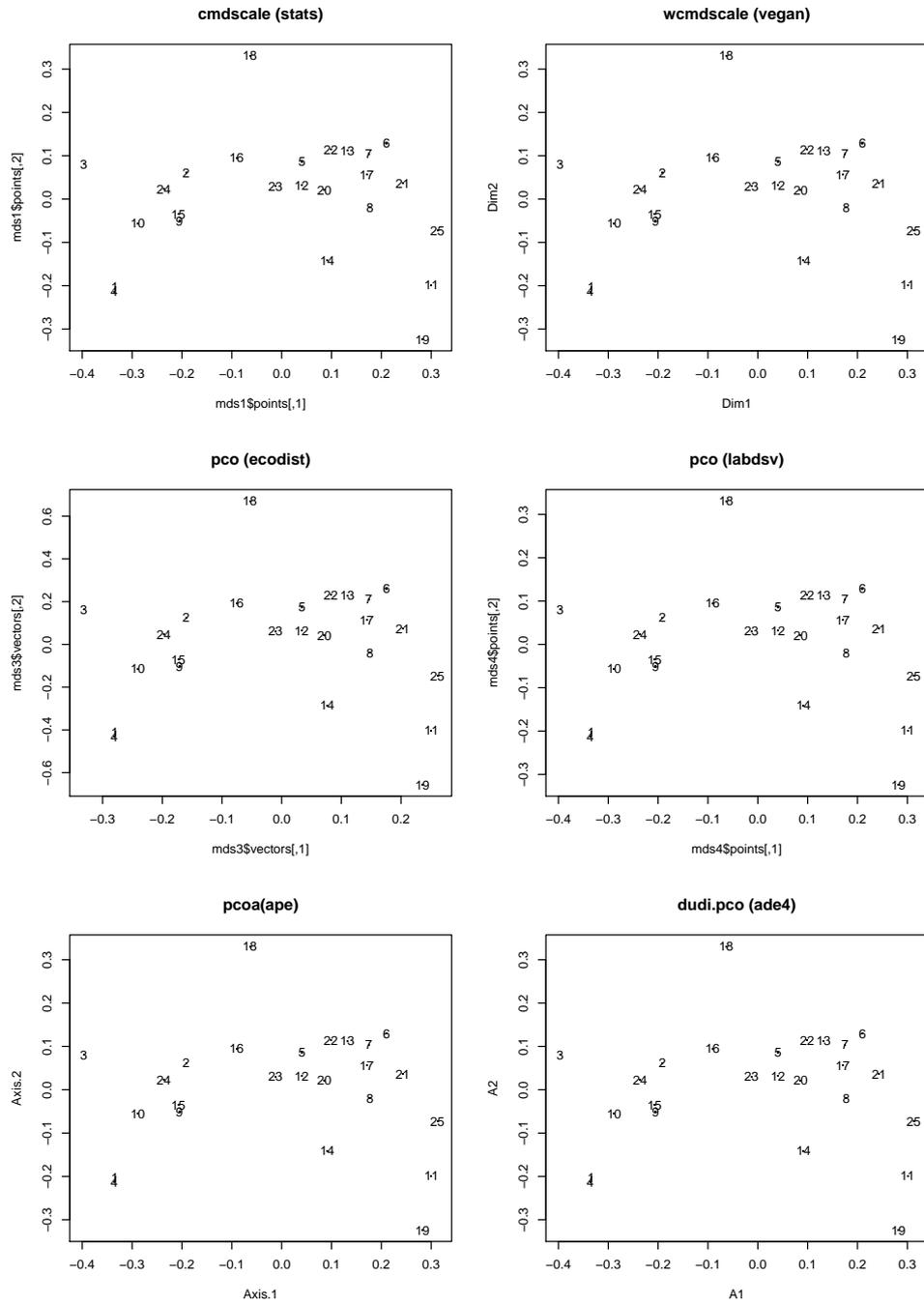


Figura 4.8: Comparando las representaciones en dos dimensiones usando las coordenadas principales

Capítulo 5

Representación óptima de distancias

5.1. Kruskal

En esta sección se considera el punto de vista de Cox y Cox (2000).

Sean \mathbf{x}_r y \mathbf{x}_s dos puntos de \mathbb{R}^p , la distancia entre dichos puntos, considerando la métrica de Minkowski, viene dada por

$$d_{rs} = \left[\sum_{i=1}^p |x_{ri} - x_{si}|^\lambda \right]^{1/\lambda} \quad (\lambda > 0).$$

Definiendo las disparidades $\{\hat{d}_{rs}\}$, como una función que depende de las distancias $\{d_{rs}\}$, esto es

$$\hat{d}_{rs} = f(d_{rs}),$$

donde f es una función monótona tal que

$$\hat{d}_{rs} \leq \hat{d}_{tu} \text{ cuando } \delta_{rs} < \delta_{tu} \quad (\text{Condición } C_1)$$

y $\{\delta_{rs}\}$ son las disimilitudes.

Definimos la función de pérdida:

$$S = \sqrt{\frac{S^*}{T^*}} \tag{5.1}$$

donde $S^* = \sum_{r,s} (d_{rs} - \hat{d}_{rs})^2$ y $T^* = \sum_{r,s} d_{rs}^2$. Note que las sumas en la función de pérdida son tomadas sobre $1 = r < s = n$, tal que $\delta_{rs} = \delta_{sr}$ para todo r, s . La función de pérdida es minimizada con respecto a $\{d_{rs}\}$, esto es con respecto a $\{x_{ri}\}$, las coordenadas de la configuración, y también con respecto a $\{\hat{d}_{rs}\}$ usando regresión isotónica.

5.1.1. Minimizando S con respecto a las disparidades

Por conveniencia, las disimilitudes $\{\delta_{rs}\}$ serán reetiquetadas $\{\delta_i, i = 1, \dots, N\}$ asumiendo que han sido colocadas en orden numérico y que no hay lazos. También, reetiquetando las distancias $\{d_{rs}\}$ como $\{d_i, i = 1, \dots, N\}$ donde d_i corresponde a la disimilitud δ_i .

Veamos el siguiente ejemplo:

Supongamos que se tienen cuatro objetos con disimilitudes:

$$\delta_{12} = 2,1, \delta_{13} = 3,0, \delta_{14} = 2,4, \delta_{23} = 1,7, \delta_{24} = 3,9, \delta_{34} = 3,2,$$

siendo las distancias entre los objetos

$$d_{12} = 3,3, d_{13} = 4,5, d_{14} = 5,7, d_{23} = 3,3, d_{24} = 4,3, d_{34} = 1,3.$$

Luego las disimilitudes ordenadas con la nueva notación, juntos con sus distancias asociadas será:

$$\begin{aligned} \delta_1 &= 1,7, \delta_2 = 2,1, \delta_3 = 2,4, \delta_4 = 3,0, \delta_5 = 3,2, \delta_6 = 3,9 \\ d_1 &= 3,3, d_2 = 3,3, d_3 = 5,7, d_4 = 4,5, d_5 = 1,3, d_6 = 4,3. \end{aligned}$$

Minimizar S es equivalente a minimizar $S' = \sum_i (d_i - \hat{d}_i)^2$, usando la nueva notación de los subíndices. Sean las sumas acumuladas de $\{d_i\}$, esto es:

$$D_i = \sum_{j=1}^i d_j \quad (i = 1, \dots, N).$$

Considerando un gráfico de D_i versus i , con los puntos P_0, P_1, \dots, P_N , donde P_0 representa al origen, la figura muestra la gráfica del ejemplo. Note que la pendiente de la línea que une P_{i-1} y P_i es solo d_i . El mayor convexo minorante de las sumas acumuladas es el gráfico del supremo de todas las funciones convexas cuyas gráficas

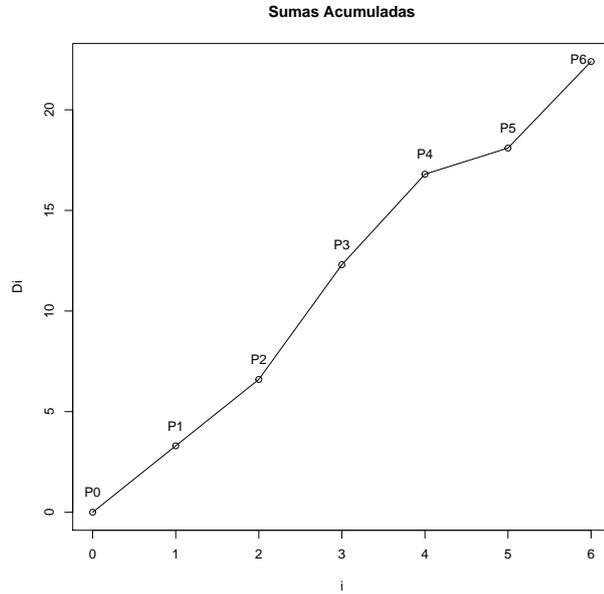


Figura 5.1: Regresión isotónica para los datos del ejemplo

están sobre el gráfico de las sumas acumuladas. (Sosteniendo una porción de cuerda tensa en P_0 y P_N daría el mayor convexo minorante). En la figura también se muestra el mayor convexo minorante. El $\{\hat{d}_i\}$ que minimiza S' es dado por el mayor convexo minorante, donde \hat{d}_i es el valor del minorante en la abscisa i . De la figura se observa que algunos de los valores de \hat{d}_i son de hecho iguales a d_i , y obviamente $S' = 0$ si $\hat{d}_i = d_i$, para todos los i . Note que $\hat{d}_i = \hat{D}_i - \hat{D}_{i-1}$ y es la pendiente de la línea. Por lo tanto si $\hat{D}_i < D_i$ entonces $\hat{d}_i = \hat{d}_{i-1}$.

Mostremos que este $\{\hat{d}_i\}$ minimiza S' .

En efecto sea $\{d_i^*\}$ un conjunto arbitrario de valores reales que satisface la condición:

$$d_i^* \leq d_j^* \text{ cuando } i \leq j$$

simplemente se tiene que demostrar que:

$$\sum_{i=1}^N (d_i - d_i^*)^2 \geq \sum_{i=1}^N (d_i - \hat{d}_i)^2.$$

Sean:

$$D_i^* = \sum_{j=1}^i d_j^*, \quad \hat{D}_i = \sum_{j=1}^i \hat{d}_j.$$

Considerando la fórmula de Abel:

$$\sum_{i=1}^N a_i b_i = \sum_{i=1}^{N-1} A_i (b_i - b_{i+1}) + A_N b_N,$$

donde $A_i = \sum_{j=1}^i a_j$ son las sumas parciales.

Entonces

$$\begin{aligned} \sum_{i=1}^N (d_i - d_i^*)^2 &= \sum_{i=1}^N [(d_i - \hat{d}_i) + (\hat{d}_i - d_i^*)]^2 \\ &= \sum_{i=1}^N (d_i - \hat{d}_i)^2 + \sum_{i=1}^N (\hat{d}_i - d_i^*)^2 + 2 \sum_{i=1}^N (d_i - \hat{d}_i)(\hat{d}_i - d_i^*). \end{aligned}$$

De donde

$$\begin{aligned} \sum_{i=1}^N (d_i - \hat{d}_i)(\hat{d}_i - d_i^*) &= \sum_{i=1}^{N-1} (D_i - \hat{D}_i)(\hat{d}_i - \hat{d}_{i+1}) - \sum_{i=1}^{N-1} (D_i - \hat{D}_i)(d_i^* - d_{i+1}^*) \\ &\quad + (D_N - \hat{D}_N)(\hat{d}_N - d_N^*). \end{aligned} \quad (5.2)$$

Luego $D_N - \hat{D}_N = 0$ debido a que el último punto del mayor minorante convexo y P_N coinciden.

Analizando:

$$(D_i - \hat{D}_i)(\hat{d}_i - \hat{d}_{i+1})$$

Si el i -ésimo punto del mayor minorante convexo coincide con P_i entonces $D_i = \hat{D}_i$ y el término vale cero. Por otro lado, si $\hat{D}_i < D_i$ entonces $\hat{d}_i = \hat{d}_{i+1}$ y el término también vale cero. Luego dado que $D_i - \hat{D}_i \geq 0$ y por la condición C_1 , $d_i^* < d_{i+1}^*$ en (5.2) se tiene que:

$$-\sum_{i=1}^{N-1} (D_i - \hat{D}_i)(d_i^* - d_{i+1}^*) \geq 0.$$

Luego:

$$\sum_{i=1}^N (d_i - d_i^*)^2 \geq \sum_{i=1}^N (d_i - \hat{d}_i)^2 + \sum_{i=1}^N (\hat{d}_i - d_i^*)^2$$

y finalmente:

$$\sum_{i=1}^N (d_i - d_i^*)^2 \geq \sum_{i=1}^N (d_i - \hat{d}_i)^2$$

Estos $\{\hat{d}_{rs}\}$ dan S' , y por lo tanto S tiene un mínimo, es la regresión isotónica de $\{d_{rs}\}$ (usando pesos iguales) con respecto al ordenamiento simple de $\{\delta_{rs}\}$. En el MDS, se refiere a la regresión isotónica como regresión primaria monótona de mínimos cuadrados de $\{d_{rs}\}$ sobre $\{\delta_{rs}\}$.

Luego para el ejemplo se tiene:

$$d_1 = d_2 = 2,45, d_3 = d_4 = d_5 = 3,83, d_r = 4,3,$$

notando que \hat{d}_1 y \hat{d}_2 son las medias de d_1 y d_2 , \hat{d}_3 , \hat{d}_4 y \hat{d}_5 son las medias de d_3 , d_4 y d_5 , \hat{d}_6 es igual a d_6 . El valor de S es 0.14.

5.1.2. Una configuración con estrés mínimo

Con $\{\hat{d}_{rs}\}$ definido como la regresión monótona de mínimos cuadrados de $\{d_{rs}\}$ sobre $\{\delta_{rs}\}$, llamamos a S el estrés de la configuración, y S^* es denominado el estrés bruto. El denominador T^* en la fórmula para el estrés es usado como un factor normalizante que permite que el estrés sea adimensional.

Ahora se busca una configuración que minimice el estrés S . Particularmente la minimización de S no es una tarea fácil de realizar.

El primer paso es ubicar todas las coordenadas de los puntos de \mathbf{X} en un vector

$$\mathbf{x} = (x_{11}, \dots, x_{1p}, \dots, x_{np})'$$

con np elementos.

Entonces se considera al estrés S como una función de \mathbf{x} , y es minimizado con respecto a \mathbf{x} de manera iterativa. Se usa el método del máximo descenso: si \mathbf{x}_m es el vector de coordenadas después de la m -ésima iteración,

$$\mathbf{x}_{m+1} = \mathbf{x}_m - \frac{\frac{\partial S}{\partial \mathbf{x}}(\mathbf{x}_m) \times sl}{\left| \frac{\partial S}{\partial \mathbf{x}}(\mathbf{x}_m) \right|},$$

donde sl es la longitud del paso.

Luego se tiene:

$$\begin{aligned}
\frac{\partial S}{\partial x_{ui}} &= \frac{1}{2} \sqrt{\frac{T^*}{S^*}} \left(\frac{T^* \frac{\partial S^*}{\partial x_{ui}} - S^* \frac{\partial T^*}{\partial x_{ui}}}{T^{*2}} \right) \\
&= \frac{1}{2} S \left(\frac{1}{S^*} \frac{\partial S^*}{\partial x_{ui}} - \frac{1}{T^*} \frac{\partial T^*}{\partial x_{ui}} \right), \\
\frac{\partial S^*}{\partial x_{ui}} &= 2 \sum_{r,s} (d_{rs} - \hat{d}_{rs}) \frac{\partial d_{rs}}{\partial x_{ui}}, \\
\frac{\partial T^*}{\partial x_{ui}} &= 2 \sum_{r,s} d_{rs} \frac{\partial d_{rs}}{\partial x_{ui}}.
\end{aligned}$$

Por la métrica de Minkowski se tiene:

$$\frac{\partial d_{rs}}{\partial x_{ui}} = d_{rs}^{1-\lambda} \sum_{r,s} (x_{ri} - x_{si})^{\lambda-1} (\delta^{ru} - \delta^{su}) \text{sign}(x_{ri} - x_{si})$$

y luego:

$$\begin{aligned}
\frac{\partial S}{\partial x_{ui}} &= S \sum_{r,s} (\delta^{ru} - \delta^{su}) \left[\frac{d_{rs} - \hat{d}_{rs}}{S^*} - \frac{d_{rs}}{T^*} \right] \\
&\quad \cdot \frac{|x_{ri} - x_{si}|^{\lambda-1}}{d_{rs}^{\lambda-1}} \text{sign}(x_{ri} - x_{si})
\end{aligned}$$

dado por Kruskal (1964b).

Se necesita elegir una configuración inicial \mathbf{x}_0 . Una posibilidad es generar n puntos de acuerdo a un proceso de Poisson en una región de \mathbb{R}^p . En su forma más simple, esto significa simular cada coordenada de manera individual para cada punto, independientemente de una distribución uniforme sobre $[0, 1]$.

Una vez que se ha elegido \mathbf{x}_0 , se puede emplear el método del máximo descenso para hallar una configuración con estrés mínimo usando el siguiente algoritmo, el cual es un resumen de Kruskal (1964b).

5.1.3. Técnica iterativa de Kruskal

Lo siguiente resume la técnica iterativa usada para hallar una configuración con estrés mínimo.

1. Elegir una configuración inicial cualquiera (\mathbf{x}_0), hacer $n = 0$.
2. Normalizar la configuración para que tenga su centroide en el origen y distancia media cuadrática unitaria del origen. Esto se puede hacer debido a que el estrés es invariante por traslaciones, dilatación uniforme; de otra manera las iteraciones sucesivas del procedimiento podrían hacer que la configuración se expanda continuamente o deambule por el plano.
3. Calcular $\{d_{rs}\}$ de la configuración normalizada.
4. Ajustar $\{\hat{d}_{rs}\}$. Se vio que la regresión monótona de mínimos cuadrados de $\{d_{rs}\}$ sobre $\{\delta_{rs}\}$, divide $\{\delta_{rs}\}$ en bloques en los cuales los valores de \hat{d}_{rs} son constantes, e igual a la media de los correspondientes valores de d_{rs} . Para encontrar la partición adecuada de δ_{rs} , primero se halla la partición más fina, la cual tiene N bloques cada uno conteniendo un solo δ_i , usando la notación alterna. Si esta partición inicial cumple:

$$d_1 \leq d_2 \leq \dots \leq d_N,$$

entonces $\hat{d}_i = d_i$ y esta partición es la final.

En caso contrario se unen dos bloques consecutivos donde $\delta_i > \delta_{i+1}$ y entonces:

$$\hat{d}_i = \hat{d}_{i+1} = \frac{d_i + d_{i+1}}{2}.$$

Los bloques se unen continuamente y se hallan nuevos \hat{d}_i 's hasta que se alcanza la partición requerida. Una explicación detallada se encuentra en Kruskal (1964a). La partición requerida también se puede hallar a partir del gráfico de las sumas acumuladas, D_i y hallando el mayor minorante convexo. La pendiente, s_i , de D_i desde el origen es D_i/i . El punto con la menor pendiente debe pertenecer al mayor minorante convexo. Todos los puntos que preceden a este punto no están en el minorante y sus pendientes se pueden remover. El punto con la siguiente menor pendiente se halla de estas pendientes restantes. Este punto está sobre el minorante, pero los puntos entre el punto minorante precedente y este, no lo están. Sus pendientes son descartadas. Este procedimiento continúa hasta que se alcanza el N -ésimo punto. Una vez que se ha establecido el mayor minorante convexo es fácil hallar $\{\hat{d}_i\}$.

5. Calcular el gradiente $\frac{\partial S}{\partial \mathbf{x}}$.

Si $\left| \frac{\partial S}{\partial \mathbf{x}}(\mathbf{x}_n) \right| < \epsilon$, donde ϵ es un valor muy pequeño predefinido, entonces se ha hallado una configuración con mínimo estrés y el proceso iterativo se puede detener.

Hay que notar que esta configuración da un mínimo local para el estrés y no el mínimo global.

6. Calcular la nueva longitud del paso sl . Kruskal recomienda el siguiente cambio de paso adecuado:

$$sl_{\text{presente}} = sl_{\text{previo}} \alpha \beta \gamma,$$

donde:

- α : factor angular = $4,0^{\cos^3 \theta}$,
- θ es el ángulo entre el gradiente previo y el actual,
- β : factor de relajación = $\frac{1,3}{1 + (5 \text{ razón de paso})^5}$,
- 5 razón de paso = $\min \left\{ 1, \frac{\text{estrés presente}}{\text{estrés de hace 5 iteraciones}} \right\}$,
- γ : factor de buena suerte = $\min \left\{ 1, \frac{\text{estrés presente}}{\text{estrés previo}} \right\}$.

7. Calcular la nueva configuración:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - sl \frac{\frac{\partial S}{\partial \mathbf{x}}(\mathbf{x}_n)}{\left| \frac{\partial S}{\partial \mathbf{x}}(\mathbf{x}_n) \right|}.$$

8. Volver al paso 2.

A continuación se muestran las diferentes salidas en el programa R , considerando el Coeficiente de Asociación de Braun-Blanquet

$$S_1 < 0, 1 > = \frac{a}{\max \{(a + b), (a + c)\}}.$$

Las descripciones de las rutinas se muestran en el apéndice C.

5.2. smacofSym (smacof)

En este caso cuando ejecutamos el comando para $n = 24$ y vemos su estructura

Estructura

```
> mds7 = smacofSym(misdatos, ndim = 24,init="torgerson")
> str(mds7)
List of 17
 $ delta      :Class 'dist'  atomic [1:300] 0.522 0.435 0.25 0.536 0.679 ...
 .. ..- attr(*, "Size")= int 25
 .. ..- attr(*, "call")= language as.dist.default(m = b)
 .. ..- attr(*, "Diag")= logi FALSE
 .. ..- attr(*, "Upper")= logi FALSE
 .. ..- attr(*, "Labels")= chr [1:25] "1" "2" "3" "4" ...
 $ dhat       :Class 'dist'  atomic [1:300] 0.977 0.814 0.468 1.003 1.271 ...
 .. ..- attr(*, "Size")= int 25
 .. ..- attr(*, "call")= language as.dist.default(m = b)
 .. ..- attr(*, "Diag")= logi FALSE
 .. ..- attr(*, "Upper")= logi FALSE
 .. ..- attr(*, "Labels")= chr [1:25] "1" "2" "3" "4" ...
 $ confdiss   :Class 'dist'  atomic [1:300] 1.008 0.83 0.499 0.998 1.263 ...
 .. ..- attr(*, "Size")= int 25
 .. ..- attr(*, "Labels")= chr [1:25] "1" "2" "3" "4" ...
 .. ..- attr(*, "Diag")= logi FALSE
 .. ..- attr(*, "Upper")= logi FALSE
 .. ..- attr(*, "method")= chr "euclidean"
 .. ..- attr(*, "call")= language dist(x = y)
 $ iord       : int [1:300] 98 3 268 117 91 108 135 112 125 160 ...
 $ conf       : num [1:25, 1:24] -0.6245 -0.3562 -0.7392 -0.6286 0.0767 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : chr [1:25] "1" "2" "3" "4" ...
 .. ..$ : chr [1:24] "D1" "D2" "D3" "D4" ...
 $ stress     : num 0.0164
 $ spp       : Named num [1:25] 5.62 3.31 3.34 3.64 1.89 ...
 ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
```

```

$ ndim      : num 24
$ weightmat:Class 'dist'  atomic [1:300] 1 1 1 1 1 1 1 1 1 1 1 ...
.. ..- attr(*, "Size")= int 25
.. ..- attr(*, "call")= language as.dist.default(m = ww)
.. ..- attr(*, "Diag")= logi FALSE
.. ..- attr(*, "Upper")= logi FALSE
$ resmat    : num [1:25, 1:25] NA 9.80e-04 2.42e-04 9.33e-04 2.41e-05 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:25] "1" "2" "3" "4" ...
.. ..$ : chr [1:25] "1" "2" "3" "4" ...
$ rss       : num 0.0805
$ init      : num [1:25, 1:24] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:25] "1" "2" "3" "4" ...
.. ..$ : NULL
$ model     : chr "Symmetric SMACOF"
$ niter     : num 20
$ nobj      : int 25
$ type      : chr "ratio"
$ call      : language smacofSym(delta = misdatos, ndim = 24, init = "torgerson")
- attr(*, "class")= chr [1:2] "smacofB" "smacof"

```

Vamos a visualizar las dos primeras coordenadas y el estrés

Coordenadas y estrés

```

> mds7$conf[,1:2]
D1      D2
1 -0.62454441 -0.36708402
2 -0.35619394  0.11389064
3 -0.73920352  0.14883852
4 -0.62856434 -0.39745764
5  0.07674496  0.15843630
6  0.38700870  0.22881441
7  0.33054238  0.19443686
8  0.33763761 -0.03270078

```

```

9 -0.38094257 -0.09208690
10 -0.53718935 -0.10619813
...
16 -0.16728756 0.17564481
17 0.31928475 0.10503640
18 -0.11503101 0.59798780
19 0.53172971 -0.59434600
20 0.16373859 0.04226861
21 0.43748717 0.05784527
22 0.18258607 0.22008277
23 -0.02363337 0.05673044
24 -0.43948801 0.04514577
25 0.58096673 -0.13734998
> mds7$stress
[1] 0.01638499

```

La representación en dos dimensiones es:

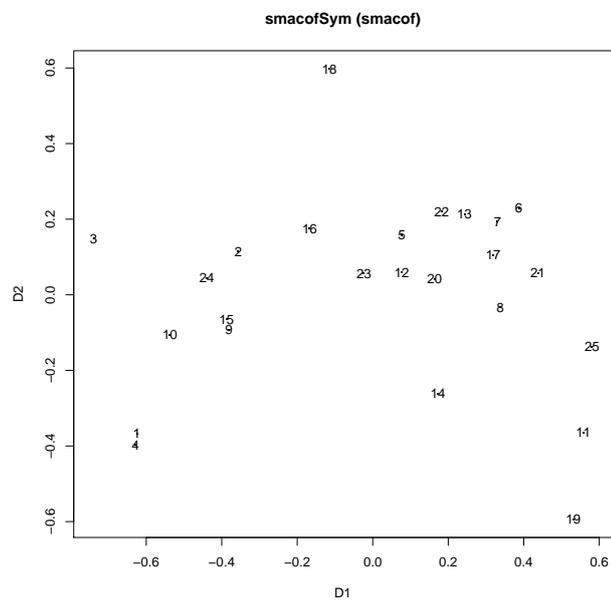


Figura 5.2: Representación en dos dimensiones usando el comando smacofSym del paquete smacof

En la figura 5.2 se observa $-0,7392035 \leq \text{Coordenada 1} \leq 0,5809667$ y $-0,594346 \leq \text{Coordenada 2} \leq 0,5979878$.

5.3. isoMDS (MASS)

En este caso cuando ejecutamos el comando para $k = 17$ (para $k=24$ sale error) y vemos su estructura:

Salida

```
> mds8=isoMDS(as.matrix(misdatos),k=17);
initial value 1.222069
iter 5 value 0.655821
iter 10 value 0.491456
iter 15 value 0.426766
iter 20 value 0.392826
iter 25 value 0.365288
iter 30 value 0.315817
iter 35 value 0.261747
iter 40 value 0.233376
iter 45 value 0.213870
iter 50 value 0.202358
final value 0.202358
stopped after 50 iterations
```

Estructura

```
> str(mds8)
List of 2
 $ points: num [1:25, 1:17] -0.3446 -0.1854 -0.3737 -0.3424 0.0434 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : NULL
 .. ..$ : NULL
 $ stress: num 0.202
```

Si se cambia el valor de p la gráfica varía y conforme aumenta los puntos se van a las esquinas. Vamos a visualizar las dos primeras coordenadas y el estrés:

Coordenadas y estrés

```
> mds8$points[,1:2]
[,1]      [,2]
[1,] -0.34462260 -0.19713935
[2,] -0.18539506  0.05642376
[3,] -0.37365727  0.04527690
[4,] -0.34240491 -0.22432487
[5,]  0.04341244  0.10378806
[6,]  0.21560925  0.13699930
[7,]  0.18152228  0.10457001
[8,]  0.17921780 -0.01893353
[9,] -0.21220736 -0.03991999
[10,] -0.29185189 -0.07170624
...
[16,] -0.09284650  0.10698606
[17,]  0.17352525  0.05784662
[18,] -0.06323612  0.32350648
[19,]  0.26582367 -0.31165020
[20,]  0.09338616  0.01781237
[21,]  0.23935211  0.03460080
[22,]  0.10940767  0.12270571
[23,] -0.00843669  0.04130117
[24,] -0.25015385  0.03551053
[25,]  0.29669992 -0.09026267
> mds8$stress
[1] 0.2023582
```

La representación en dos dimensiones es:

En la figura 5.3 se observa $-0,3736573 \leq$ Coordenada 1 $\leq 0,2966999$ y $-0,3116502 \leq$ Coordenada 2 $\leq 0,3235065$.

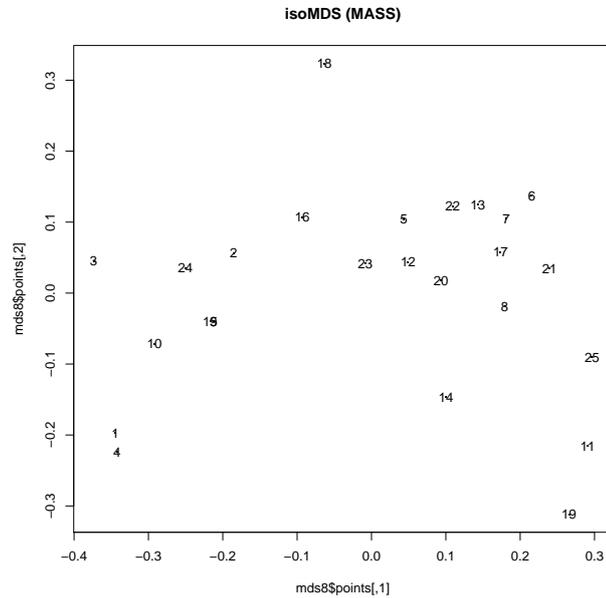


Figura 5.3: Representación en dos dimensiones usando el comando isoMDS del paquete MASS

5.4. sammon (MASS)

En este caso cuando ejecutamos el comando para $k = 17$ (para $k=24$ sale error) y vemos su estructura:

Salida

```
> mds9=sammon((misdatos),k=17);
Initial stress      : 0.00185
stress after 10 iters: 0.00040, magic = 0.018
stress after 20 iters: 0.00025, magic = 0.500
stress after 30 iters: 0.00025, magic = 0.150
```

Estructura

```
> str(mds9)
List of 3
 $ points: num [1:25, 1:17] -0.3341 -0.1893 -0.3957 -0.3362 0.0412 ...
 ..- attr(*, "dimnames")=List of 2
```

```
.. ..$ : NULL
.. ..$ : NULL
$ stress: num 0.000248
$ call  : language sammon(d = (misdatos), k = 17)
```

Vamos a visualizar las dos primeras coordenadas y el estrés:

Coordenadas y estrés

```
> mds9$points[,1:2]
[,1]      [,2]
[1,] -0.33405431 -0.19603886
[2,] -0.18930031  0.05942324
[3,] -0.39573128  0.08135425
[4,] -0.33620433 -0.21223143
[5,]  0.04124578  0.08203730
[6,]  0.20676061  0.12032823
[7,]  0.17723015  0.10196653
[8,]  0.18135280 -0.01573300
[9,] -0.20179546 -0.04745085
[10,] -0.28662276 -0.05814156
...
[16,] -0.08901006  0.09126665
[17,]  0.17031487  0.05488283
[18,] -0.06127214  0.32257400
[19,]  0.28426890 -0.32050738
[20,]  0.08656120  0.02510463
[21,]  0.23280775  0.03125910
[22,]  0.09705690  0.11769276
[23,] -0.01293713  0.02948389
[24,] -0.23449046  0.02364053
[25,]  0.30945242 -0.07036368
> mds9$stress
[1] 0.000248425
```

La representación en dos dimensiones es:

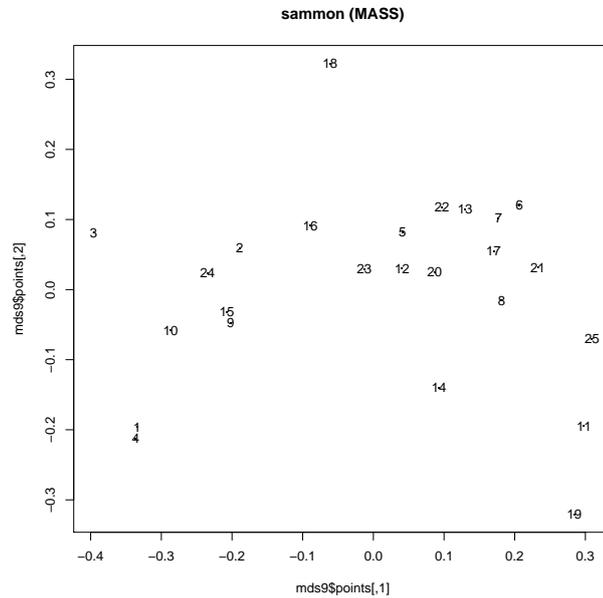


Figura 5.4: Representación en dos dimensiones usando el comando sammon del paquete MASS

En la figura 5.4 se observa $-0,3957313 \leq \text{Coordenada 1} \leq 0,3094524$ y $-0,3205074 \leq \text{Coordenada 2} \leq 0,322574$.

5.5. metaMDS (vegan)

En este caso cuando ejecutamos el comando para $k = 24$ y vemos su estructura

Salida

```
> mds10=metaMDS(misdatos,k=24);
Run 0 stress 0.0006731704
Run 1 stress 9.733491e-05
... New best solution
... Procrustes: rmse 0.1022087 max resid 0.1274361
Run 2 stress 9.415462e-05
... New best solution
... Procrustes: rmse 0.03679151 max resid 0.09126561
```


Run 3 stress 8.749315e-05
... New best solution
... Procrustes: rmse 0.04914222 max resid 0.09769743
Run 4 stress 9.557568e-05
... Procrustes: rmse 0.04045253 max resid 0.07935845
Run 5 stress 9.256363e-05
... Procrustes: rmse 0.02819197 max resid 0.060882
Run 6 stress 9.524068e-05
... Procrustes: rmse 0.04112895 max resid 0.06355625
Run 7 stress 9.57352e-05
... Procrustes: rmse 0.04661298 max resid 0.07435957
Run 8 stress 9.574775e-05
... Procrustes: rmse 0.04119429 max resid 0.06536566
Run 9 stress 9.944143e-05
... Procrustes: rmse 0.03565912 max resid 0.06500368
Run 10 stress 9.881421e-05
... Procrustes: rmse 0.03313682 max resid 0.05582453
Run 11 stress 9.488066e-05
... Procrustes: rmse 0.03421523 max resid 0.05004637
Run 12 stress 8.826315e-05
... Procrustes: rmse 0.02488946 max resid 0.05350741
Run 13 stress 9.721059e-05
... Procrustes: rmse 0.04835377 max resid 0.08816846
Run 14 stress 9.572315e-05
... Procrustes: rmse 0.04270058 max resid 0.05644513
Run 15 stress 9.759352e-05
... Procrustes: rmse 0.04462417 max resid 0.06865914
Run 16 stress 9.66021e-05
... Procrustes: rmse 0.04082102 max resid 0.1100038
Run 17 stress 9.732765e-05
... Procrustes: rmse 0.04530381 max resid 0.06908852
Run 18 stress 8.549272e-05
... New best solution
... Procrustes: rmse 0.04078735 max resid 0.07883192

```

Run 19 stress 8.405443e-05
... New best solution
... Procrustes: rmse 0.03097178  max resid 0.05558776
Run 20 stress 9.955115e-05
... Procrustes: rmse 0.04862426  max resid 0.1013483
*** No convergence -- monoMDS stopping criteria:
20: stress < smin
Warning messages:
1: In cmdscale(dist, k = k) : only 17 of the first 24 eigenvalues are > 0
2: In metaMDS(misdatos, k = 24) :
Stress is (nearly) zero - you may have insufficient data

```

Estructura

```

> str(mds10)
List of 35
 $ nobj      : int 25
 $ nfix      : int 0
 $ ndim      : num 24
 $ ndis      : int 300
 $ ngrp      : int 1
 $ diss      : num [1:300] 0.242 0.25 0.267 0.273 0.286 ...
 $ iidx      : int [1:300] 13 4 21 13 23 6 13 21 21 8 ...
 $ jidx      : int [1:300] 5 1 17 6 5 5 7 6 20 6 ...
 $ xinit     : num [1:600] 0.0914 0.6266 0.5959 0.5317 0.7426 ...
 $ istart    : int 1
 $ isform    : int 1
 $ ities     : int 1
 $ iregn     : int 1
 $ iscal     : int 1
 $ maxits    : int 200
 $ sratmx    : num 1
 $ strmin    : num 1e-04
 $ sfgrmn    : num 1e-07
 $ dist      : num [1:300] 0.724 0.724 0.724 0.724 0.724 ...

```

```

$ dhat      : num [1:300] 0.724 0.724 0.724 0.724 0.724 ...
$ points    : num [1:25, 1:24] -0.2578 -0.1264 -0.2586 -0.2639 0.0342 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : chr [1:25] "1" "2" "3" "4" ...
.. ..$ : chr [1:24] "MDS1" "MDS2" "MDS3" "MDS4" ...
..- attr(*, "centre")= logi TRUE
..- attr(*, "pc")= logi TRUE
..- attr(*, "halfchange")= logi FALSE
..- attr(*, "internalscaling")= num 1.88
$ stress    : num 8.41e-05
$ grstress  : num 8.41e-05
$ iters     : int 66
$ icause    : int 2
$ call      : language metaMDS(comm = misdatos, k = 24)
$ model     : chr "global"
$ distmethod: chr "user supplied"
$ distcall  : chr "as.dist.default(m = comm)"
$ distance  : chr "user supplied"
$ converged : logi FALSE
$ tries     : num 20
$ engine    : chr "monoMDS"
$ species   : logi NA
$ data      : chr "misdatos"
- attr(*, "class")= chr [1:2] "metaMDS" "monoMDS"

```

Vamos a visualizar las dos primeras coordenadas:

Coordenadas

```

> mds10$points[,1:2]
MDS1      MDS2
1 -0.257835182  0.21414054
2 -0.126442859 -0.10732479
3 -0.258619549 -0.06897146
4 -0.263860064  0.16838227
5  0.034217267 -0.07851407

```

```
6  0.164343666 -0.12045415
7  0.122547880 -0.11793351
8  0.145212044  0.09579750
9  -0.175233889  0.03536005
10 -0.225174126  0.04816230
...
16 -0.065859199 -0.14360178
17  0.106689620 -0.03137504
18 -0.007960558 -0.28034144
19  0.217051355  0.25727784
20  0.052150131 -0.03901375
21  0.212594502 -0.02457935
22  0.072616225 -0.13211440
23  0.009808798  0.00155283
24 -0.182002791 -0.05816077
25  0.232998962  0.04882354
```

La representación en dos dimensiones es:

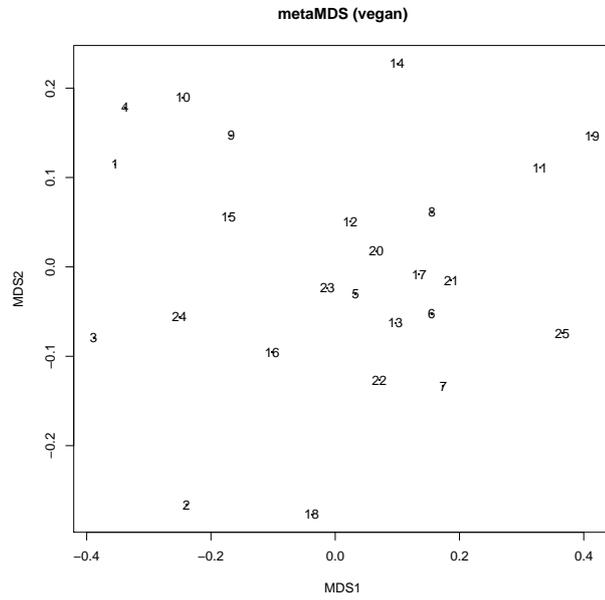


Figura 5.5: Representación en dos dimensiones usando el comando metaMDS del paquete vegan

En la figura 5.5 se observa $-0,3887738 \leq \text{Coordenada 1} \leq 0,4133888$ y $-0,2767129 \leq \text{Coordenada 2} \leq 0,227517$.

Capítulo 6

Análisis de Procrustes (AP)

En este capítulo se considera el punto de vista de Camiz (2017b).

En análisis de datos, el análisis de Procrustes es un método que proporciona el mejor ajuste de un conjunto de puntos, llamado nube de prueba, a un conjunto dado, llamado nube objetivo, de acuerdo a transformaciones que no cambian, excepto por un factor de escala, las distancias recíprocas entre los puntos de la nube de prueba. Este método se usa ampliamente en reconocimiento de patrones y en el llamado análisis de formas, como un primer ajuste de transformaciones más complejas, pero se puede aplicar a todas las situaciones en las cuales se requieren las comparaciones directas entre configuraciones de los mismos objetos bajo diferentes representaciones.

6.1. Comparando formas

Para comparar la forma de dos (o más) objetos, los objetos deben estar superpuestos de manera óptima. Consideremos un conjunto de n puntos de referencias $\mathcal{L} = \{L_1, L_2, \dots, L_n\}$, esto es, puntos que correspondientes a cada otro en los dos objetos, digamos $\mathcal{L}^1 = \{L_1^1, L_2^1, \dots, L_n^1\}$ y $\mathcal{L}^2 = \{L_1^2, L_2^2, \dots, L_n^2\}$. El objetivo es hallar la mejor manera de superponerlos si es que la hubiera. Así si consideramos los dos conjuntos de puntos de referencia como nubes de puntos en un espacio topológico y buscamos una transformación continua del espacio que lleve cada punto a su correspondiente, la expresión analítica de la transformación, si la hubiera, podría ser muy compleja. Así que podemos imaginar alguna transformación más simple que lleve los puntos de la primera nube tan cerca como sea posible a la segunda correspondiente.

Para definir cercanía y estimar que tan cerca están estos dos puntos, tenemos que considerar los conjuntos en un espacio métrico y definir la cercanía en términos de la distancia. Para simplificar se elige la distancia euclidiana por algunas de sus características especiales. También un concepto relevante en espacios euclidianos es la inercia de una nube de puntos con respecto a un punto dado.

La Superposición de Procrustes (SP) se realiza por traslación óptima, rotación y escalamiento uniforme de los objetos. En otras palabras se ajustan libremente las ubicaciones en el espacio y el tamaño de los objetos. El objetivo es obtener una ubicación y tamaño similar, minimizando una medida de diferencia de forma llamada distancia de Procrustes entre los objetos. Esta distancia algunas veces se llama completa, como opuesto a la SP parcial, en el cual no se realiza el escalamiento (esto es, se preserva el tamaño de los objetos). Hay que notar que después de una SP completa, los objetos coincidirán exactamente si su forma es idéntica. Por ejemplo con la SP completa dos esferas con radios diferentes siempre coincidirán, porque ellas tienen exactamente la misma forma. Con la SP parcial ellas nunca coinciden. Esto implica que, por la definición estricta de forma en geometría, el análisis de forma se debe realizar usando la SP completa. Un análisis estadístico basado en la SP parcial no es un análisis de forma pura ya que no solo es sensible a las diferencias de forma, sino también a las diferencias de tamaño. La SP completa y parcial nunca hará coincidir perfectamente dos objetos con formas diferentes, como un cubo y una esfera, o una mano derecha y una mano izquierda.

En algunos casos, la SP completa y parcial pueden incluir reflexión. La reflexión permite, una superposición exitosa (posiblemente perfecta) de una mano izquierda a una mano derecha. Así, la SP parcial con reflexión preserva el tamaño pero permite traslación, rotación y reflexión, mientras que la SP completa con reflexión permite traslación, rotación, escalamiento y reflexión.

- Un problema de Procrustes ortogonal es un método el cual se puede usar para hallar la rotación óptima y/o reflexión (esto es, la transformación lineal ortogonal óptima) para la SP de un objeto con respecto a otro.
- Un problema de Procrustes ortogonal con restricciones, sujeto a $\det(R) = 1$ (donde R es una matriz de rotación), es un método el cual puede ser usado para determinar la rotación óptima para la SP de un objeto con respecto a otro (sin reflexión). En algunos contextos, este método es llamado el algoritmo de Kabsch.

Dada una matriz de covarianzas A entre dos conjuntos de caracteres. Es posible calcular la rotación óptima U basada en la fórmula matricial:

$$U = (A'A)^{1/2}A^{-1}$$

pero implementar una solución numérica a esta fórmula es complicada en el caso de que A no tenga inversa. En efecto, esto se puede desarrollar mejor a través de la descomposición de valores singulares de A : Si $A = V\Lambda W'$, luego la rotación es dada por $U = WV'$. En efecto, se debe prestar atención al caso de rotaciones. Basados en el signo de $d = \det(WV')$, en el caso bidimensional es suficiente cambiar el signo de cada vector. En el caso tridimensional se tiene:

$$U = W \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{pmatrix} V'$$

Así, los diferentes pasos se pueden describir como sigue:

- **Traslación**

Los componentes traslacionales de un objeto pueden ser removidos trasladando este de tal forma que la media de todos los objetos (esto es el centroide) este en el origen.

Tomando k puntos en dos dimensiones, esto es $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$, la media de estos puntos es (\bar{x}, \bar{y}) , donde:

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_k}{k}, \quad \bar{y} = \frac{y_1 + y_2 + \dots + y_k}{k}.$$

Ahora trasladando estos puntos de tal forma de que su media se traslade al origen:

$$(x, y) \rightarrow (x - \bar{x}, y - \bar{y}),$$

se obtienen los puntos $\{(x_1 - \bar{x}, y_1 - \bar{y}), \dots, (x_k - \bar{x}, y_k - \bar{y})\}$.

- **Escalamiento uniforme**

De manera similar la componente escalar se puede remover escalando los objetos tal que la raíz de la distancia media cuadrática (RDMC) de los puntos al origen

trasladado sea 1. Esta RDMC es una medida estadística de la escala de los objetos o tamaño:

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (y_1 - \bar{y})^2 + \dots + (x_k - \bar{x})^2 + (y_k - \bar{y})^2}{k}}$$

La escala es 1 cuando las coordenadas de los puntos son divididas por la escala inicial del objeto:

$$\left\{ \left(\frac{x_1 - \bar{x}}{s}, \frac{y_1 - \bar{y}}{s} \right), \dots, \left(\frac{x_k - \bar{x}}{s}, \frac{y_k - \bar{y}}{s} \right) \right\}.$$

■ Rotación

Remover el componente rotacional es más complejo, puesto que no siempre hay un referente de orientación. Considere dos objetos compuestos de la misma cantidad de puntos con escala y traslación removidos. Digamos $\{(x_1, y_1), \dots, (x_k, y_k)\}$, $\{(w_1, z_1), \dots, (w_k, z_k)\}$. Uno de estos objetos se puede usar para proporcionar una orientación de referencia. Fijando el objeto de referencia y rotando el otro alrededor del origen, hasta que se halle un ángulo θ óptimo de rotación tal que la suma del cuadrado de las distancias entre los puntos correspondientes sea minimizada (un ejemplo de técnica de mínimos cuadrados).

Una rotación por un ángulo θ da:

$$(u, v) = (w_1 \cos \theta - z_1 \sin \theta, w_1 \sin \theta + z_1 \cos \theta),$$

donde (u, v) son las coordenadas del punto girado. Luego para k puntos se tiene:

$$\theta = \tan^{-1} \left(\frac{\sum_{i=1}^k (w_i y_i - z_i x_i)}{\sum_{i=1}^k (w_i x_i + z_i y_i)} \right)$$

Cuando el objeto es tridimensional, la rotación óptima se representa por una matriz R de rotación 3×3 , en lugar de un ángulo simple y en este caso se puede usar la descomposición por valores singulares para hallar el valor óptimo de R (ver la solución para el problema de Procrustes ortogonal con restricciones, sujeto a $\det(R) = 1$).

6.2. La transformación de Procrustes

El objetivo del Análisis de Procrustes (AP) es de ajustar lo mejor posible dos nubes de puntos en un espacio geométrico, esto es ajustar la nube de prueba a la nube objetivo tan bien como sea posible a través de una transformación rígida. Se consideran transformaciones rígidas a las traslaciones, rotaciones y escalamientos, o cualquier composición de estas.

Sean dos nubes de puntos en \mathbb{R}^p , la nube objetivo \mathcal{N}_x de n puntos \mathbf{x}'_i cuya matriz de coordenadas es X y la nube de prueba \mathcal{N}_z de n puntos \mathbf{z}'_i , cuya matriz de coordenadas es Z , indexadas por el mismo conjunto I , con $\text{card}(I) = n$ y pesos p_i positivos que suman 1, $\sum_{i=1}^n p_i = 1$. Además, se asume que se tiene una métrica representada por la matriz definida positiva M .

El AP consiste en hallar la llamada Transformación de Procrustes (TP): $P(a, T, s)$, compuesta por una traslación a , $a \in \mathbb{R}^p$, una rotación T , $T \in \mathbb{R}^{p \times p}$, y un escalamiento s , $s \in \mathbb{R}$, tal que las imágenes de los puntos de la nube de prueba \mathcal{N}_z transformada por P , esto es

$$\mathbf{u}_i = P(\mathbf{z}_i) = sT\mathbf{z}_i + a, \quad i \in I$$

estén tan cerca como sea posible a los puntos de la nube objetivo \mathcal{N}_x en el sentido de los mínimos cuadrados. Por lo tanto la función objetivo a minimizar es:

$$PA = \sum_{i \in I} p_i \|\mathbf{x}_i - \mathbf{u}_i\|_M^2,$$

con la norma $\|\cdot\|_M$ que depende de la métrica M .

6.3. Búsqueda de la traslación a

Si introducimos los centroides de ambas nubes $G = \sum_{i \in I} p_i \mathbf{x}_i$ y $H = \sum_{i \in I} p_i \mathbf{z}_i$, es fácil observar que la transformación de H , dada por:

$$U = P(H) = sTH + a,$$

que minimiza PA se tiene cuando $G = U$, esto implica:

$$a = G - sTH.$$

En efecto, de la función objetivo se tiene:

$$\begin{aligned}
PA &= \sum_{i \in I} p_i \|\mathbf{x}_i - \mathbf{u}_i\|^2 \\
&= \sum_{i \in I} p_i \|\mathbf{x}_i - G\|^2 + \sum_{i \in I} p_i \|\mathbf{u}_i - U\|^2 \\
&\quad - 2 \sum_{i \in I} p_i \langle \mathbf{x}_i - G, \mathbf{u}_i - U \rangle + \|G - U\|^2
\end{aligned}$$

donde el único término que depende de T es $\|G - U\|^2$, el cual es mínimo (esto es 0) cuando $G = U$.

6.4. Búsqueda del escalamiento s

Si se introduce la traslación hallada a , en la función objetivo PA , se tiene:

$$PA = s^2 \sum_{i \in I} p_i \|T(\mathbf{z}_i - H)\|^2 - 2s \sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle$$

que alcanza su mínimo cuando la derivada parcial con respecto a s es cero, esto es:

$$\frac{\partial PA}{\partial s} = 2s \sum_{i \in I} p_i \|T(\mathbf{z}_i - H)\|^2 - 2 \sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle = 0,$$

de donde:

$$s = \frac{\sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle}{\sum_{i \in I} p_i \|T(\mathbf{z}_i - H)\|^2}.$$

Como cualquier rotación es una isometría que preserva la norma, T no afecta el denominador y se puede simplificar, luego se tiene:

$$s = \frac{\sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle}{\sum_{i \in I} p_i \|\mathbf{z}_i - H\|^2}.$$

6.5. Búsqueda de la rotación T

Una vez introducido el escalamiento s en la función objetivo PA , se tiene:

$$\begin{aligned} PA &= \sum_{i \in I} p_i \|\mathbf{x}_i - \mathbf{u}_i\|^2 \\ &= \sum_{i \in I} p_i \|\mathbf{x}_i - G\|^2 + \frac{\left(\sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle \right)^2}{\sum_{i \in I} p_i \|\mathbf{z}_i - H\|^2} - 2 \frac{\left(\sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle \right)^2}{\sum_{i \in I} p_i \|\mathbf{z}_i - H\|^2} \end{aligned}$$

De donde se tiene:

$$PA = \sum_{i \in I} p_i \|\mathbf{x}_i - \mathbf{u}_i\|^2 = \sum_{i \in I} p_i \|\mathbf{x}_i - G\|^2 - \frac{\left(\sum_{i \in I} p_i \langle \mathbf{x}_i - G, T(\mathbf{z}_i - H) \rangle \right)^2}{\sum_{i \in I} p_i \|\mathbf{z}_i - H\|^2}.$$

Si se centran los vectores \mathbf{x}_i y \mathbf{z}_i a sus respectivos centroides, esto es $\tilde{\mathbf{x}}_i = \mathbf{x}_i - G$ y $\tilde{\mathbf{z}}_i = \mathbf{z}_i - H$, la búsqueda de la rotación T se reduce a maximizar $\sum_{i \in I} p_i \langle \tilde{\mathbf{x}}_i, T\tilde{\mathbf{z}}_i \rangle$.

Se definen ahora las matrices:

$$V = \sum_{i \in I} p_i \tilde{\mathbf{x}}_i \tilde{\mathbf{z}}_i', \quad W = VMV'M$$

donde M es la métrica en \mathbb{R}^p , ciertamente MW es simétrica (a W se le denomina M -simétrica). Así M admite una base M -ortonormal de autovectores, denotados por $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_p\}$ ordenados de acuerdo al orden decreciente de sus correspondientes autovectores $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ Golub y Loan (1996). Si los primeros r autovalores son diferentes de cero, se pueden definir r vectores M -ortonormales:

$$\mathbf{f}_\alpha = \frac{V'M\mathbf{e}_\alpha}{\sqrt{\lambda_\alpha}} \in \mathbb{R}^p, \quad \alpha = 1, 2, \dots, r$$

que se pueden completar a una base M -ortonormal de \mathbb{R}^p . En efecto:

$$\langle \mathbf{f}_\alpha, \mathbf{f}_\beta \rangle_M = \mathbf{f}_\alpha' M \mathbf{f}_\beta = \frac{\mathbf{e}_\alpha' M V M V' M \mathbf{e}_\beta}{\sqrt{\lambda_\alpha \lambda_\beta}} = \begin{cases} 0, & \text{si } \alpha \neq \beta \\ 1, & \text{si } \alpha = \beta \end{cases}$$

Bourgeois (1978) demostró que un posible T que da el óptimo buscado es definido por:

$$T\mathbf{f}_\alpha = \mathbf{e}_\alpha, \forall \alpha \in \{1, \dots, r\}.$$

Observaciones

1) Sean X y Z son dos tablas de dimensiones $n \times p$, cuyas filas corresponden a las nubes \mathcal{N}_x y \mathcal{N}_z respectivamente. Cuando la métrica M es la euclidiana y todos los pesos son iguales, entonces los tres pasos de la transformación, esto es la búsqueda de la rotación T , la traslación a y el escalamiento s se puede describir como (ver Borg y Groenen (2005)):

1. Centrar las tablas \tilde{X} y \tilde{Z} .
2. Calcular el producto de las tablas $C = \tilde{X}'\tilde{Z}$.
3. Calcular la descomposición de valores singulares de $C = P\Lambda Q'$.

Obteniéndose los siguientes resultados:

- La rotación se reduce a $T = PQ'$.
- El escalamiento se reduce a $s = \frac{\text{traza}(\tilde{X}'\tilde{Z}T')}{\text{traza}(\tilde{Z}'\tilde{Z})}$.
- La traslación viene a ser $a = \frac{1}{n}(X - sYT')\mathbf{1}_n$, donde $\mathbf{1}_n$ es un vector cuyos componentes son todos 1.

2) La función objetivo:

$$PA = \sum_{i \in I} p_i \|\mathbf{x}_i - \mathbf{u}_i\|,$$

con $\mathbf{u}_i = P(\mathbf{z}_i) = sT\mathbf{z}_i + a, \forall i \in I$, se puede reformular como:

$$PA = \text{traza}((X - (sZT' + \mathbf{1}_n\mathbf{a}'))'(X - (sZT' + \mathbf{1}_n\mathbf{a}'))).$$

Esta formulación permite generalizar el criterio a mas de dos nubes.

6.6. Análisis de Procrustes

El Análisis de Procrustes, es un método para comparar dos conjuntos de coordenadas X e Y cuyas filas se refieren a las mismas observaciones, considerando a δ_{ij} y ε_{ij} las distancias originadas por los dos conjuntos. La idea es rotar y/o reflejar el patrón de un conjunto de puntos que representan las observaciones de acuerdo a su matriz en orden a maximizar su similitud con el patrón generado por el otro.

Así, se requieren una matriz ortogonal H y un vector \mathbf{m} tal que $XH + \mathbf{1m}$ se aproxima mejor a Y , el beneficio del ajuste se puede evaluar como sigue:

$$f(\delta, \varepsilon) = \text{traza} ((Y - XH - \mathbf{1m})'(Y - XH - \mathbf{1m})), \quad (6.1)$$

esta debe ser la cantidad a ser minimizada para hallar H y \mathbf{m} .

Suponiendo sin pérdida de generalidad, que ambas coordenadas son centradas con respecto a los centroides, el problema se reduce haciendo $m_i = 0$ y el problema es equivalente a maximizar la traza de $Y'XH$. Luego, dada la descomposición por valores singulares de $Y'X = UTV'$, se tiene:

$$\text{traza} (Y'XH) = \text{traza} (UTV'H) = \text{traza} (\Gamma V'HU) \quad (6.2)$$

Como $V'HU = Q$ es diagonal, luego (6.2) se puede escribir de la forma:

$$\text{traza} (Y'XH) = \sum_i \gamma_i q_{ii}, \quad (6.3)$$

donde los γ_i diferentes de cero son positivos y $-1 \leq q_{ii} \leq 1$. Entonces el valor máximo se obtiene haciendo $q_{ii} = 1$, esto es $Q = I$, correspondiente a $H = VU'$.

Entonces la suma residual de cuadrados resulta ser:

$$f(\delta, \varepsilon) = \text{traza} (X'X) \text{traza} (Y'Y) - 2 \text{traza} \Gamma \quad (6.4)$$

Se asumió, que X e Y tienen la misma cantidad de columnas, pero si no fuera el caso, la menor matriz se puede completar con columnas de ceros, dando solo algunos autovalores cero (pero el correspondiente q_{ii} se puede elegir libremente).

Como resultado de ajustar X a Y , la configuración XH se puede graficar junto con Y . Si uno de los conjuntos es mas grande que el otro, es suficiente normalizar X e Y para tener:

$$\text{traza} (X'X) = \text{traza} (Y'Y)$$

y resolver el problema.

Gower (1975) y Sibson (1978) consideran el análisis de Procrustes con escalamiento isométrico (dilatación), en este caso se ha probado que el escalamiento puede seguir a una rotación/reflexión previa, teniendo en este caso:

$$f(\delta, \varepsilon) = \text{traza}(XX') - \frac{(\text{traza}(YX'XY')^{1/2})^2}{\text{traza}(Y')} \quad (6.5)$$

Hay que notar que este coeficiente no es simétrico en X e Y .

6.7. Problema ortogonal de Procrustes

Schönemann (1966) define el problema ortogonal de Procrustes la transformación por mínimos cuadrados de una matriz dada A en una matriz B por medio de una matriz ortonormal de transformación T , tal que la suma de los cuadrados de la matriz residual $E = AT - B$ es mínima. Esto es:

$$\begin{cases} AT = B + E \\ TT' = T'T = I \\ \text{mín}\{\text{traza}(E'E)\} \end{cases}$$

Luego:

$$\text{traza}(E'E) = \text{traza}((AT - B)'(AT - B)) = \text{traza}(T'A'AT - 2T'A'B + B'B)$$

con la condición $T'T - I = 0$, tenemos la función de Lagrange:

$$\mathcal{L} = \text{traza}(T'A'AT - 2T'A'B + B'B) - L(T'T - I)$$

con los multiplicadores de Lagrange L , luego el sistema de derivadas es:

$$\frac{\partial \mathcal{L}}{\partial T} = 2(A'A)T - 2A'B + 2LT = 0$$

Lo que nos lleva a resolver:

$$S = PT + TQ$$

con $P = A'A$, $S = A'B$ y $Q = (L + L')/2$.

Como P y Q son simétricos, la condición $T'T = I$ da:

$$Q = T'S - T'PT = Q',$$

por lo tanto $T'S$ también es simétrico y se tiene: $T'S = S'T$, luego:

$$S = TS'T$$

y

$$SS' = TS'ST'$$

Luego SS' y $S'S$ tienen los mismos autovalores, de tal forma que de las descomposiciones $SS' = WD_SW'$ y $S'S = VD_SV'$ y con las condiciones de ortogonalidad de W y V se tiene:

$$WD_SW' = TVD'_VT'$$

y por lo tanto:

$$W = TV \text{ o } T = WV'$$

6.8. Prueba de Procrustes

En la rotación simétrica de Procrustes, descrita aquí, cada uno de los dos conjuntos de datos Y_1 ($n \times p_1$) e Y_2 ($n \times p_2$), están estandarizados para que tengan varianza total igual a 1 previa a la rotación. Esto se obtiene por la estandarización de Gower (1971), la cual consiste en dividir todos los elementos de una matriz centrada por columnas entre la raíz cuadrada de la varianza total de la matriz, la cual también es la raíz cuadrada de la suma de sus autovalores: si Λ_1 y Λ_2 son las matrices diagonales de los valores propios de Y_1 e Y_2 respectivamente, y sean:

$$Z_i = \frac{Y_i}{\sqrt{\text{traza}(\Lambda_i)}}, i = 1, 2.$$

Por lo tanto la covarianza de Y_1 e Y_2 es:

$$\text{cov}(Y_1, Y_2) = \frac{Z_1'Z_2}{n-1}.$$

Hay que notar que el análisis de co inercia de Z_1 y Z_2 producen los mismo autovalores relativos, el coeficiente RV , y grafica como el de Y_1 e Y_2 . El análisis de Procrustes

difiere del análisis de co-inercia en que emplea diferentes matrices de salida para la representación conjunta de los dos conjuntos de objetos.

Luego aplicando la descomposición por valores singulares de esta matriz de covarianza se tiene:

$$\text{cov}(Y_1, Y_2) = W\Delta U'$$

con traza $\Delta > 0$.

Luego la estadística de Procrustes residual de la suma de cuadrados Gower (1971, 1975), Davis (1978) es:

$$m_{12}^2 = 1 - \text{traza}(\Delta^2).$$

Entonces, la matriz de rotación que proporciona el mejor ajuste de los objetos de Y_2 a los objetos de Y_1 se calcula como:

$$H = UV'.$$

La matriz rotada Z_{2r} se calcula como sigue:

$$Z_{2r} = \text{traza}(\Delta)Z_2H$$

donde $\text{traza}(\Delta)$ actúa como un factor de escalamiento.

6.9. Aplicación

A continuación se va a hallar la matriz de Procrustes en diferentes casos.

6.9.1. Considerando el coeficiente de asociación de Braun-Blanquet

Vamos a hallar la matriz de Procrustes entre las diferentes salidas obtenidas por el ACP, ACoP y Kruskal (se van a considerar las dos primeras componentes), vamos a elegir la representación obtenida en el ACoP (usaremos la salida del `wcmdscale` como matriz objetivo)

ACoP vs Kruskal (sammon)

Rotation matrix:

[,1] [,2]

```
[1,] 0.9999996989 0.0007760331
[2,] -0.0007760331 0.9999996989
```

Translation of averages:

```
[,1]      [,2]
[1,] -4.385804e-17 -4.194196e-17
```

Scaling of target:

```
[1] 1.009032
```

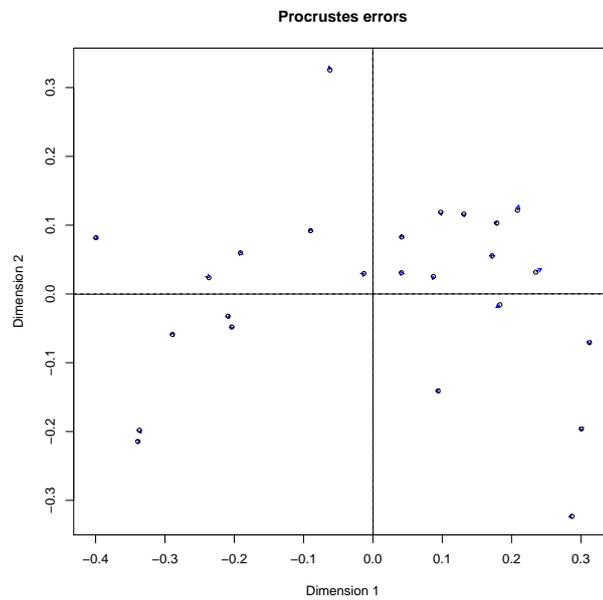


Figura 6.1: ACoP vs Kruskal (sammon)

ACoP vs Kruskal (metaMDS)

Rotation matrix:

```
[,1]      [,2]
[1,]  0.99705103 -0.07674136
[2,] -0.07674136 -0.99705103
```

Translation of averages:

```
[,1]      [,2]
[1,] -3.918247e-17 -4.090025e-17
```

Scaling of target:

```
[1] 0.9353269
```

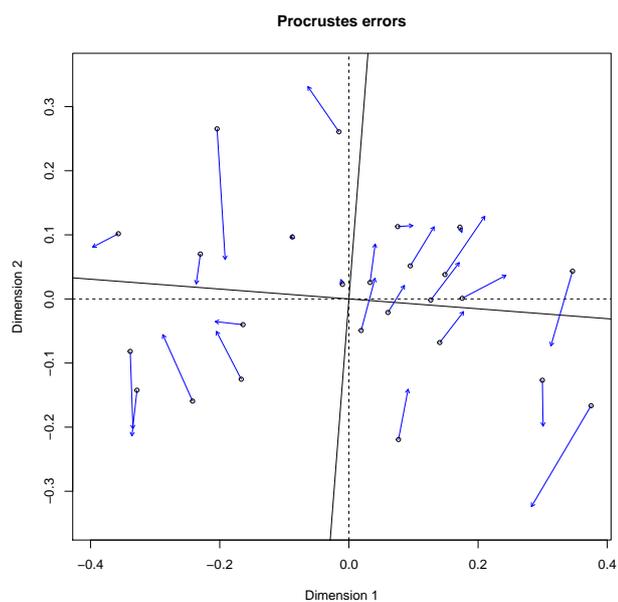


Figura 6.2: ACoP vs Kruskal (metaMDS)

ACoP vs ACP

Rotation matrix:

```
[,1]      [,2]
[1,]  0.2538110  0.9672538
[2,] -0.9672538  0.2538110
```

Translation of averages:

```
[,1]      [,2]
[1,] -0.03293784 -0.2850439
```

Scaling of target:

```
[1] 0.5479666
```

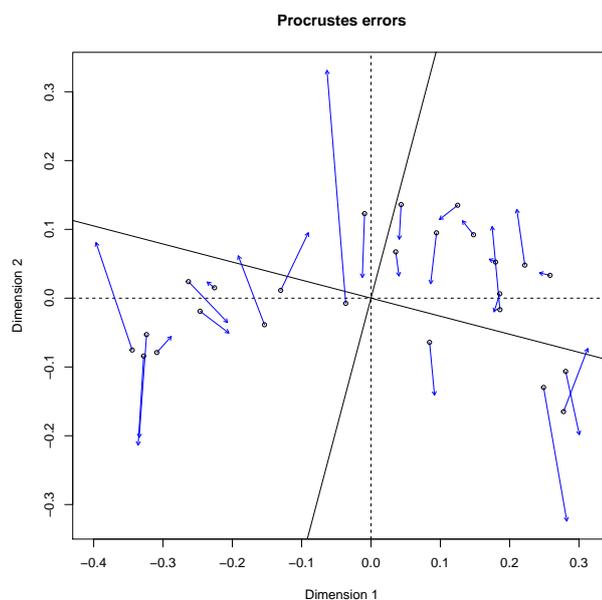


Figura 6.3: ACoP vs ACP

6.9.2. Considerando diferentes coeficientes de asociación

En este caso vamos a hallar la matriz de Procrustes para coeficientes de asociación distintos, vamos a considerar la salida `wcmdscale` del ACoP.

Braun-Blanquet y Simpson

En este caso vamos a hallar la matriz de Proustes para los coeficientes de asociación de Braun-Blanquet y de Simpson.

El coeficiente de asociación de Simpson, es de la siguiente forma

$$S_2 = \frac{a}{\min\{(a+b), (a+c)\}}.$$

De donde se obtiene

Rotation matrix:

```
[,1]      [,2]
[1,] 0.99952682 -0.03075917
[2,] 0.03075917  0.99952682
```

Translation of averages:

```
[,1]      [,2]
[1,] -6.638186e-17 3.427351e-17
```

Scaling of target:

```
[1] 1.004019
```

Braun-Blanquet y Russell & Rao

En este caso vamos a hallar la matriz de Proustes para los coeficientes de asociación de Braun-Blanquet y de Russell & Rao.

El coeficiente de asociación de Russell & Rao, es de la siguiente forma

$$S_{14} = \frac{a}{a+b+c+d}.$$

De donde se obtiene

Rotation matrix:

```
[,1]      [,2]
[1,] 0.99760797  0.06912554
[2,] 0.06912554 -0.99760797
```

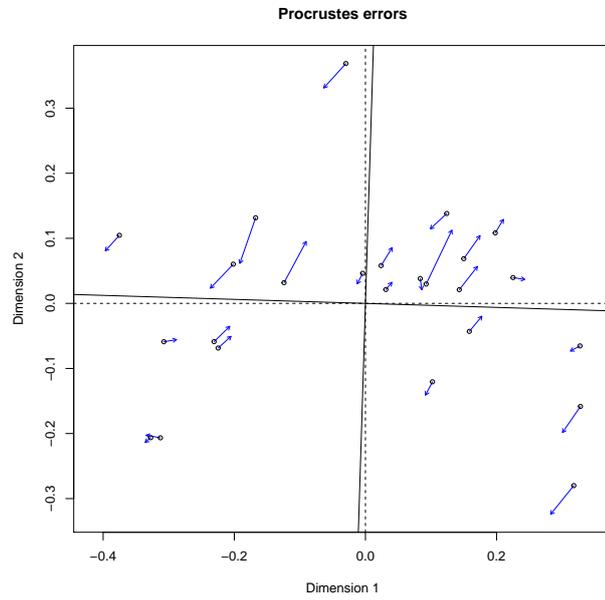


Figura 6.4: Braun-Blanquet vs Simpson

Translation of averages:

```
[,1]      [,2]
[1,] -7.598087e-18 4.906189e-17
```

Scaling of target:

```
[1] 1.050516
```

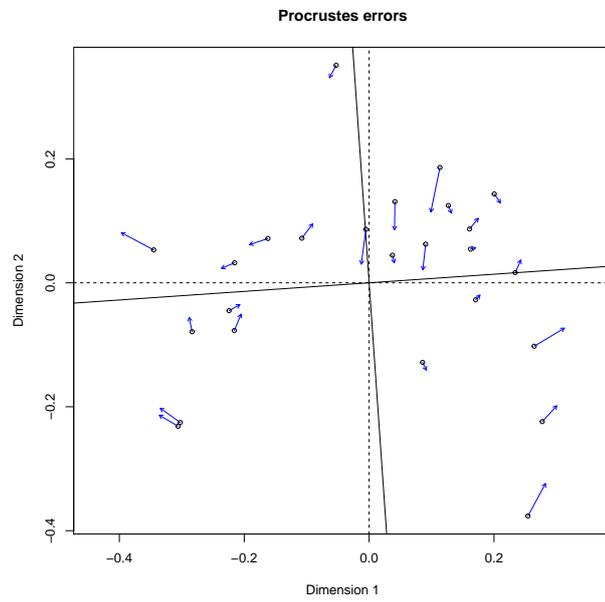


Figura 6.5: Braun-Blanquet vs Russell & Rao

Capítulo 7

Conclusiones

En el ACoP se observa a pesar de que las salidas tienen diferentes autovalores las representaciones gráficas bidimensionales tienen la misma forma (figura 4.8). Por lo tanto se elige el `wcmdscale` pues permite conocer las coordenadas principales para los valores propios negativos.

Usando el método de Kruskal, las gráficas de las salidas tienen la misma forma que las obtenidas en el ACoP, excepto con la salida del `metaMDS`. La función que se recomienda usar es la de `sammon`, pues se obtiene el menor estrés (0.000248).

Cuando aplicamos el método de Procrustes para las distintas salidas para el coeficiente de Braun-Blanquet tenemos

- En el caso ACoP vs Kruskal (`sammon`) en la gráfica se ve que la variación es casi nula, la matriz de rotación es casi la identidad y el escalamiento es casi 1.
- En el caso ACoP vs Kruskal (`metaMDS`) en la gráfica se ve que hay una variación, la matriz de rotación es diferente de la identidad y hay un escalamiento.
- En el caso ACoP vs ACP en la gráfica se ve que hay una variación, la matriz de rotación varía y hay un escalamiento.

Cuando aplicamos el método de Procrustes para diferentes coeficientes de asociación (en este caso Simpson y Russell & Rao) tomando como objetivo la salida obtenida para el coeficiente de Braun-Blanquet ha habido variación en la matriz de rotación y en el escalamiento

Hay que resaltar que el rango de los coeficiente de similitud (S) de Braun-Blanquet, Simpson y Russell & Rao es $\langle 0, 1 \rangle$ y se ha considerado la Disimilitud (D) como $D = 1 - S$, quedaría pendiente ver que pasa con disimilitudes distintas. También quedaría pendiente comparar las matrices de Procrustes para una mayor cantidad de coeficientes de similitud.

Apéndice A

Método de ACP en R

A.1. PCA (FactoMineR)

Realiza un ACP con individuos suplementarios, variables cuantitativas suplementarias y variables categóricas suplementarias, los valores faltantes son reemplazados por la media de la columna.

Sintaxis:

```
PCA(X, scale.unit = TRUE, ncp = 5, ind.sup = NULL,  
    quanti.sup = NULL, quali.sup = NULL, row.w = NULL,  
    col.w = NULL, graph = TRUE, axes = c(1,2))
```

X es una tabla de datos con n filas (individuos) y p columnas (variables numéricas).

ncp número de dimensiones consideradas en los resultados (5 por defecto) .

scale.unit un booleano, si es *TRUE* (valor asignado por defecto) entonces los datos son escalados a una varianza unitaria.

ind.sup un vector que muestra los índices de los individuos suplementarios.

quanti.sup un vector que muestra los índices de las variables cuantitativas suplementarias.

quali.sup un vector que muestra los índices de las variables categóricas suplementarias.

row.w pesos opcionales de las filas (por defecto un vector de 1 para filas de peso uniforme), los pesos solo son dados por los individuos activos.

col.w pesos opcionales de las columnas (por defecto un vector de 1 para columnas de peso uniforme), los pesos solo son dados por las variables activas.

graph un booleano, si es *TRUE* se muestra el gráfico.

axes un vector de longitud 2 que especifica los componentes a graficar.

Salida

Regresa una lista que incluye:

eig una matriz que contiene los valores propios, el porcentaje de varianza y el porcentaje de varianza acumulado.

var una lista de matrices que contiene todos los resultados para las variables activas (coordenadas, correlación entre las variables y los ejes, coseno cuadrado, contribuciones).

ind una lista de matrices que contiene todos los resultados para los individuos activos (coordenadas, coseno cuadrado, contribuciones).

ind.sup una lista de matrices que contiene todos los resultados para los individuos suplementarios (coordenadas, coseno cuadrado).

quanti.sup una lista de matrices que contiene todos los resultados para las variables cuantitativas suplementarias (coordenadas, correlación entre las variables y ejes).

quali.sup una lista de matrices que contiene todos los resultados para las variables categóricas suplementarias (coordenadas de las categorías de cada variable, *v.test* el cual es un criterio con una distribución normal y *eta2* el cual es el coeficiente de correlación cuadrático entre una variable cuantitativa y una dimensión).

Regresa el mapa de factores individuales y el mapa de factores variables.

Apéndice B

Métodos de ACoP en R

B.1. Cmdscale (stats)

Escalamiento Multidimensional Clásico de una matriz de datos, también conocido como ACoP Team (2016).

Sintaxis:

```
cmdscale(d, k = 2, eig = FALSE, add = FALSE, x.ret = FALSE,  
list. = eig || add || x.ret)
```

d tiene la estructura de distancia (salida del comando *dist*) o una matriz simétrica conteniendo las disimilitudes.

k la dimensión máxima del espacio en el cual se van a representar los datos, varía de $\{1, 2, \dots, n - 1\}$.

eig si es *TRUE* se muestran los valores propios.

add valor lógico que indica si se va a calcular una constante *c* la cual se suma a las disimilitudes no diagonales tal que las disimilitudes modificadas sean euclidianas. Única alternativa “cailliez”.

x.ret indica cuando se debe retornar una matriz simétrica de distancias doblemente centradas.

list. valor lógico, si es TRUE se muestra una lista con todos los valores de la salida, en caso contrario solo una matriz $n \times k$.

Salida

Si `.list` es falso (por defecto), se visualiza una matriz con k columnas cuyas filas representan las coordenadas de los puntos elegidos para representar las disimilitudes.

De otra forma, se visualiza una lista con los siguientes componentes:

points una matriz con k columnas cuyas filas representan las coordenadas de los puntos elegidos para representar las disimilitudes.

eig los n valores propios hallados durante el proceso de escalamiento si `eig` es TRUE.

x la matriz de distancias doblemente centrada si `x.ret` es TRUE.

ac la constante aditiva c , es 0 si `add=FALSE`.

GOF un vector numérico de longitud 2, $(g1, g2)$ tal que:

$$g1 = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^n |\lambda_j|}, \quad g2 = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^n \max\{\lambda_j, 0\}}$$

donde λ_j son los valores propios ordenados de forma decreciente

B.2. wcmdscale (vegan)

Escalamiento Multidimensional Clásico ponderado también conocido como ACoP ponderado Oksanen *et al.* (2018).

Sintaxis:

```
wcmdscale(d, k, eig = FALSE, add = FALSE, x.ret = FALSE, w)
## S3 method for class 'wcmdscale'
plot(x, choices = c(1, 2), type = "t", ...)
## S3 method for class 'wcmdscale'
scores(x, choices = NA, ...)
```

d tiene la estructura de distancia (salida del comando *dist*) o una matriz simétrica conteniendo las disimilitudes.

k la dimensión máxima del espacio en el cual se van a representar los datos, varía de $\{1, 2, \dots, n - 1\}$. Si no se escribe, todas las dimensiones con autovalores por encima de cero.

eig si es *TRUE* se muestran los valores propios.

add valor lógico que indica si se va a calcular una constante *c* la cual se suma a las disimilitudes no diagonales tal que las disimilitudes modificadas sean euclidianas (todos los valores propios son no negativos). Los valores pueden ser “lingoes” (valor por defecto si se escribe *TRUE*) y “cailliez” (única alternativa para el “cmdscale”)

x.ret indica cuando se debe retornar una matriz simétrica de distancias doblemente centradas.

w pesos de los puntos.

x El objeto resultante de “wcmdscale” cuando la función se llama con las opciones *eig = TRUE* o *x.ret = TRUE*.

choices Ejes a ser retornados, *NA* retorna todos los ejes reales.

type tipo de gráfico el cual puede ser “text”, “points” o “none”.

... Otros argumentos pasados a las funciones gráficas.

La función *wcmdscale* está basada en la función *cmdscale* pero usa puntos con pesos. Los puntos con pesos altos tendrán una influencia mayor que los que tienen pesos bajos, haciendo $w = 1$ se obtendrá el escalamiento multidimensional ordinario.

Con las opciones por defecto, la función regresa solo una matriz.

El método de *Lingoes* (*add = lingoes*) suma la constante *c* al cuadrado de las disimilitudes usando $\sqrt{d^2 + 2c}$ y el método de *Cailliez* (*add = cailliez*) a las disimilitudes $d + c$.

Salida

Si *eig = FALSE* y *x.ret = FALSE* (por defecto), se visualiza una matriz con

k columnas, cuyas filas representan las coordenadas de los puntos correspondientes valores propios positivos.

De otra forma, se visualiza una objeto de clase *wcmdscale* con los componentes similares al *cmdscale*:

points una matriz con k columnas cuyas filas representan las coordenadas de los puntos elegidos para representar las disimilitudes.

eig los $n - 1$ valores propios hallados durante el proceso de escalamiento si *eig* es TRUE.

x la matriz de distancias doblemente centrada si *x.ret* es TRUE.

ac la constante aditiva c , es 0 si *add*=FALSE.

GOF un vector numérico de longitud 2, $(g1, g2)$ tal que:

$$g1 = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^k |\lambda_j|}, \quad g2 = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^k \max\{\lambda_j, 0\}}$$

donde λ_j son los valores propios ordenados de forma decreciente.

weights los pesos.

negaxes Una matriz de valores para ejes con valores propios negativos escalados por los valores propios absolutos como *points*. Este es *NULL* si hay valores propios no negativos.

call Función call.

B.3. pco (ecodist)

Realiza un ACoP Goslee y Urban (2007).

Sintaxis:

```
ecodist::pco(x, negvals = "zero", dround = 0)
```

x es una matriz de disimilitudes triangular inferior (debe ser tipo *dist* y si no lo es usar *as.dist*).

negvals si es = "*zero*" todos los valores propios negativos los hace cero, si es = "*rm*" los corrige para valores propios negativos usando el método 1 de Legendre y Anderson 1999.

dround si es mayor que 0, intenta corregir el error de redondeo redondeando a esa cantidad de lugares.

PCO hace un escalamiento multidimensional métrico (escalamiento clásico) es muy similar al ACP, pero permite el uso de cualquier disimilitud métrica.

Salida

values muestra los valores propios para cada componente. Esto es una medida de la varianza explicada por cada dimensión.

vectors muestra los autovectores cada columna contiene los valores para cada dimensión.

B.4. pco (labdsv)

ACoP es un análisis matrices de distancia o de disimilitudes métricas Roberts (2016).

Sintaxis:

```
pco(dis, k=2)
```

dis la matriz de distancia o disimilitud.

k la cantidad de dimensiones de salida.

Salida

values

call

En este caso cuando ejecutamos el comando para $k = 24$ y vemos su estructura

Estructura

```
> mds4 = labdsv::pco(misdatos,k=24);
Warning message:
In cmdscale(dis, k = k, eig = TRUE) :
only 17 of the first 24 eigenvalues are > 0
> str(mds4)
List of 5
 $ points: num [1:25, 1:17] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
 ..- attr(*, "dimnames")=List of 2
 .. ..$ : NULL
 .. ...$ : NULL
 $ eig   : num [1:25] 1.164 0.461 0.322 0.262 0.252 ...
 $ x     : NULL
 $ ac    : num 0
 $ GOF   : num [1:2] 0.947 1
 - attr(*, "class")= chr "pco"
```

Sale un mensaje de error por los autovalores negativos

$\$points$ muestras las 17 coordenadas (17 valores propios positivos)

$\$eig$ los 25 valores propios hallados durante el proceso de escalamiento si eig es TRUE.

$\$x$ la matriz de distancias doblemente centrada, en este caso NULL.

$\$ac$ la constante aditiva c , es 0.

$\$GOF$ un vector numérico de longitud 2, $(g1, g2)$ tal que:

$$g1 = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^n |\lambda_j|}, \quad g2 = \frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^n \max\{\lambda_j, 0\}}$$

donde λ_j son los valores propios ordenados de forma decreciente.

B.5. pcoa (ape)

Calcula la descomposición de coordenadas principales (también denominado escalamiento clásico) de una matriz de distancias D (Gower 1966). Se implementan dos métodos de corrección para valores propios negativos Paradis, Claude, y Strimmer (2004).

Sintaxis:

```
pcoa(D, correction="none", rn=NULL)
```

```
## S3 method for class 'pcoa'  
biplot(x, Y=NULL, plot.axes = c(1,2), dir.axis1=1,  
dir.axis2=1, rn=NULL, main=NULL, ...)
```

D una matriz de distancias de clase *dist* o *matrix*.

correction métodos de corrección para valores propios negativos “*lingoes*” o “*cailliez*”, el valor por defecto es “*none*”.

rn un vector opcional de nombres de fila, de longitud n , para los n objetos.

entradas para el *biplot*

x objeto salida del *pcoa*.

Esta función implementa dos métodos para valores propios negativos en el ACoP. Los valores propios negativos se presentan en el ACoP al descomponer matrices de distancias producidas por coeficientes que no son euclidianos.

En el ACoP, cuando se presentan valores propios negativos en la descomposición, la matriz de distancias D se puede modificar usando el procedimiento de Lingoés o Cailliez para producir resultados sin valores propios negativos.

En el procedimiento de Lingoés se suma una constante c_1 igual al doble del valor absoluto del mayor valor negativo del ACoP original, al cuadrado de la distancia original en la matriz distancia, excepto a los valores de la diagonal. El nuevo ACoP, desarrollado sobre las distancias modificadas, tiene a los más $n - 2$ valores propios positivos, al menos dos valores propios nulos y ningún autovalor negativo.

En el procedimiento de Cailliez se suma una constante c_2 a las distancias originales en la matriz distancia, excepto en la diagonal. El nuevo ACoP, desarrollado sobre las distancias modificadas, tiene a los mas $n - 2$ valores propios positivos, al menos dos valores propios nulos y ningún autovalor negativo.

En todos los casos, en la salida solo se muestran los autovectores correspondientes a los valores propios positivos. Los autovectores son escalados a la raíz cuadrada de los correspondientes valores propios. Gower (1966) demostró que los autovectores escalados de tal forma preservan la distancia original (en la matriz D) entre los objetos. Estos autovectores se pueden usar para representar un ordenamiento de los objetos.

Se recomienda no usar ACoP para producir ordenamientos en las distancias chi cuadrado, perfil de abundancia o Hellinger. Es más fácil transformar primero los datos de la comunidad usando las siguientes disponibles en la función *decostand* del paquete *vegan* y luego llevar a cabo el ACP sobre la data transformada:

transformación Chord: `decostand(spiders, "normalize")`

Transformación relativa a perfiles de abundancia profiles: `decostand(spiders, "total")`

Transformación de Hellinger: `decostand(spiders, "hellinger")`

transformación Chi-cuadrado: `decostand(spiders, "chi.square")`

El ordenamiento resultante será idéntico y los cálculos mas cortos.

A este método de ordenamiento de dos pases se le llama ACP basado en transformación.

La función *biplot.pcoa* produce gráficos para cualquier par de coordenadas principales. Las variables originales se pueden proyectar sobre la gráfica de ordenamiento.

Salida

correction los valores del parámetro *correction* y la variable *correct* en la función.

done Una nota describiendo el tipo de corrección hecha.

values Los valores propios y la información relacionada.

Eigenvalues Todos los valores propios (positivos, nulos y negativos).

Relative_eig valores propios relativos.

Corr_eig Valores propios corregidos.

Rel_corr_eig Valores propios relativos despues de la corrección de Lingoes o Cailliez.

Broken_stick Fracciones de varianza esperada bajo el modelo de broken stick.

Cum_corr_eig valores propios corregidos relativos acumulado.

Cumul_br_stick fracciones acumuladas del broken stick.

vectors las coordenadas principales con valores propios positivos.

trace la traza de la matriz distancias. Esto también es la suma de todos los valores propios, positivos y negativos.

B.6. dudi.pco (ade4)

Análisis de Coordenadas principales

dudi.pco desarrolla un ACoP de una matriz de distancias euclidianas y devuelve los resultados como objetos de clase *pco* y *dudi* Dray y Dufour (2007).

Sintaxis:

```
dudi.pco(d, row.w = "uniform", scannf = TRUE, nf = 2,  
full = FALSE, tol = 1e-07)  
## S3 method for class 'pco'  
scatter(x, xax = 1, yax = 2, clab.row = 1, posieig = "top",  
sub = NULL, csub = 2, ...)
```

d un objeto de clase *dist* que contiene una matriz de distancias euclidianas.

row.w pesos opcionales para las filas de la matriz distancia. Si no es `NULL`, debe ser un vector de números positivos con longitud igual al tamaño de la matriz distancia.

scannf un valor lógico que indica si se muestra un gráfico de barras de los valores propios.

nf si *scannf* = *FALSE*, un entero indicando la cantidad de ejes a mostrar.

full un valor lógico que indica cuando se muestran todos los ejes.

tol un valor de tolerancia para probar cuando la matriz de distancias es euclidiana: un autovalor es considerado positivo si es mayor que $-tol \cdot \lambda_1$, donde λ_1 es el mayor autovalor.

Salida

Devuelve una lista de clase *pco* y *dudi*.

En este caso cuando ejecutamos el comando y vemos su estructura

Salida

```
> mds6 =dudi.pco( as.dist(misdatos), scannf = FALSE, nf = 24)
Warning message:
In dudi.pco(as.dist(misdatos), scannf = FALSE, nf = 24) :
Non euclidean distance
```

Estructura

```
> str(mds6)
List of 11
 $ eig : num [1:17] 0.0465 0.0184 0.0129 0.0105 0.0101 ...
 $ rank: int 17
 $ nf  : int 17
 $ cw  : num [1:17] 1 1 1 1 1 1 1 1 1 1 ...
 $ tab : 'data.frame': 25 obs. of 17 variables:
 ..$ A1 : num [1:25] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
 ..$ A2 : num [1:25] -0.2025 0.0615 0.0808 -0.2136 0.0857 ...
 ..$ A3 : num [1:25] 0.009 -0.3626 -0.0649 0.0836 0.0576 ...
 ..$ A4 : num [1:25] 0.0085 0.02501 -0.06409 -0.04685 -0.00582 ...
 ..$ A5 : num [1:25] -0.0893 -0.0431 -0.1011 -0.0194 0.0463 ...
 ..$ A6 : num [1:25] -0.1902 -0.0125 0.0114 -0.1091 -0.1029 ...
 ..$ A7 : num [1:25] -0.0503 -0.1363 0.098 -0.0646 -0.0807 ...
 ..$ A8 : num [1:25] -0.000657 0.121632 -0.114263 -0.104068 0.031317 ...
 ..$ A9 : num [1:25] 0.05316 -0.06314 -0.00575 -0.07176 -0.01818 ...
 ..$ A10: num [1:25] 0.0471 0.0352 0.0214 -0.0571 -0.1071 ...
 ..$ A11: num [1:25] 0.0119 -0.0316 -0.0892 0.0439 0.0211 ...
 ..$ A12: num [1:25] 0.017 0.0154 0.0797 -0.0127 -0.0321 ...
 ..$ A13: num [1:25] 0.05525 0.0144 0.00279 -0.06236 0.01393 ...
```

```

..$ A14: num [1:25] 0.04355 -0.00478 -0.00627 -0.07349 0.03515 ...
..$ A15: num [1:25] 0.0396 0.0245 -0.1027 0.0289 -0.0742 ...
..$ A16: num [1:25] 0.03704 0.02642 -0.00244 -0.03852 0.01511 ...
..$ A17: num [1:25] 0.00279 0.01097 -0.01719 -0.00504 0.01018 ...
$ li : 'data.frame': 25 obs. of 17 variables:
..$ A1 : num [1:25] -0.3344 -0.1916 -0.3967 -0.336 0.0407 ...
..$ A2 : num [1:25] -0.2025 0.0615 0.0808 -0.2136 0.0857 ...
..$ A3 : num [1:25] 0.009 -0.3626 -0.0649 0.0836 0.0576 ...
..$ A4 : num [1:25] 0.0085 0.02501 -0.06409 -0.04685 -0.00582 ...
..$ A5 : num [1:25] -0.0893 -0.0431 -0.1011 -0.0194 0.0463 ...
..$ A6 : num [1:25] -0.1902 -0.0125 0.0114 -0.1091 -0.1029 ...
..$ A7 : num [1:25] -0.0503 -0.1363 0.098 -0.0646 -0.0807 ...
..$ A8 : num [1:25] -0.000657 0.121632 -0.114263 -0.104068 0.031317 ...
..$ A9 : num [1:25] 0.05316 -0.06314 -0.00575 -0.07176 -0.01818 ...
..$ A10: num [1:25] 0.0471 0.0352 0.0214 -0.0571 -0.1071 ...
..$ A11: num [1:25] 0.0119 -0.0316 -0.0892 0.0439 0.0211 ...
..$ A12: num [1:25] 0.017 0.0154 0.0797 -0.0127 -0.0321 ...
..$ A13: num [1:25] 0.05525 0.0144 0.00279 -0.06236 0.01393 ...
..$ A14: num [1:25] 0.04355 -0.00478 -0.00627 -0.07349 0.03515 ...
..$ A15: num [1:25] 0.0396 0.0245 -0.1027 0.0289 -0.0742 ...
..$ A16: num [1:25] 0.03704 0.02642 -0.00244 -0.03852 0.01511 ...
..$ A17: num [1:25] 0.00279 0.01097 -0.01719 -0.00504 0.01018 ...
$ l1 : 'data.frame': 25 obs. of 17 variables:
..$ RS1 : num [1:25] -1.55 -0.888 -1.839 -1.557 0.189 ...
..$ RS2 : num [1:25] -1.492 0.453 0.596 -1.574 0.631 ...
..$ RS3 : num [1:25] 0.0793 -3.1962 -0.5722 0.7369 0.5078 ...
..$ RS4 : num [1:25] 0.083 0.244 -0.6255 -0.4572 -0.0568 ...
..$ RS5 : num [1:25] -0.89 -0.43 -1.008 -0.193 0.462 ...
..$ RS6 : num [1:25] -2.06 -0.135 0.124 -1.182 -1.114 ...
..$ RS7 : num [1:25] -0.576 -1.561 1.122 -0.74 -0.924 ...
..$ RS8 : num [1:25] -0.00869 1.60856 -1.51111 -1.37628 0.41416 ...
..$ RS9 : num [1:25] 0.7695 -0.9139 -0.0832 -1.0388 -0.2631 ...
..$ RS10: num [1:25] 0.69 0.515 0.314 -0.835 -1.567 ...
..$ RS11: num [1:25] 0.193 -0.514 -1.451 0.715 0.343 ...

```

```

..$ RS12: num [1:25] 0.292 0.265 1.374 -0.219 -0.554 ...
..$ RS13: num [1:25] 1.1061 0.2883 0.0558 -1.2483 0.2789 ...
..$ RS14: num [1:25] 1.013 -0.111 -0.146 -1.709 0.817 ...
..$ RS15: num [1:25] 0.983 0.609 -2.548 0.716 -1.839 ...
..$ RS16: num [1:25] 1.0255 0.7314 -0.0675 -1.0665 0.4184 ...
..$ RS17: num [1:25] 0.1 0.395 -0.619 -0.182 0.366 ...
$ c1 : 'data.frame': 17 obs. of 17 variables:
..$ CS1 : num [1:17] 1 0 0 0 0 0 0 0 0 0 ...
..$ CS2 : num [1:17] 0 1 0 0 0 0 0 0 0 0 ...
..$ CS3 : num [1:17] 0 0 1 0 0 0 0 0 0 0 ...
..$ CS4 : num [1:17] 0 0 0 1 0 0 0 0 0 0 ...
..$ CS5 : num [1:17] 0 0 0 0 1 0 0 0 0 0 ...
..$ CS6 : num [1:17] 0 0 0 0 0 1 0 0 0 0 ...
..$ CS7 : num [1:17] 0 0 0 0 0 0 1 0 0 0 ...
..$ CS8 : num [1:17] 0 0 0 0 0 0 0 1 0 0 ...
..$ CS9 : num [1:17] 0 0 0 0 0 0 0 0 1 0 ...
..$ CS10: num [1:17] 0 0 0 0 0 0 0 0 0 1 ...
..$ CS11: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
..$ CS12: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
..$ CS13: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
..$ CS14: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
..$ CS15: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
..$ CS16: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
..$ CS17: num [1:17] 0 0 0 0 0 0 0 0 0 0 ...
$ co : 'data.frame': 17 obs. of 17 variables:
..$ Comp1 : num [1:17] 0.216 0 0 0 0 ...
..$ Comp2 : num [1:17] 0 0.136 0 0 0 ...
..$ Comp3 : num [1:17] 0 0 0.113 0 0 ...
..$ Comp4 : num [1:17] 0 0 0 0.102 0 ...
..$ Comp5 : num [1:17] 0 0 0 0 0.1 ...
..$ Comp6 : num [1:17] 0 0 0 0 0 ...
..$ Comp7 : num [1:17] 0 0 0 0 0 ...
..$ Comp8 : num [1:17] 0 0 0 0 0 ...
..$ Comp9 : num [1:17] 0 0 0 0 0 ...

```

```

..$ Comp10: num [1:17] 0 0 0 0 0 0 ...
..$ Comp11: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
..$ Comp12: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
..$ Comp13: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
..$ Comp14: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
..$ Comp15: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
..$ Comp16: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
..$ Comp17: num [1:17] 0 0 0 0 0 0 0 0 0 0 0 ...
$ lw : num [1:25] 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 0.04 ...
$ call: language dudi.pco(d = as.dist(misdatos), scannf = FALSE, nf = 24)
- attr(*, "class")= chr [1:2] "pco" "dudi"

```

Vemos una advertencia que la distancia no es euclidiana.

\$eig los valores propios positivos.

\$rank el rango de la matriz de los puntos con sus coordenadas .

\$nf la cantidad de coordenadas a considerar.

\$cwt el peso de las columnas (17) en este caso.

\$tab las 17 coordenadas para los 25 puntos.

\$li los 25 puntos con 2 (por el $nf = 2$) coordenadas.

\$l1 componentes principales, con n filas y nf columnas.

\$c1 ejes principales, con p filas y nf columnas.

\$co column coordinates, con p filas y nf columnas.

\$lw pesos de las filas (líneas), un vector con p componentes.

\$call la función llamada.

otra forma de llamar a las coordenadas:

```

> mds6$li$A1
[1] -0.33438754 -0.19157178 -0.39670526 -0.33595817
[5]  0.04069364  0.21039294  0.17476366  0.17757421
[9] -0.20508787 -0.28813406  0.30055489  0.04036965
[13]  0.13189790  0.09162010 -0.20697692 -0.09036761
[17]  0.17072130 -0.06348408  0.28248849  0.08597451
[21]  0.24308172  0.09879386 -0.01251359 -0.23640118
[25]  0.31266121
> mds6$li$A2

```



```

[1] -0.20248243  0.06152301  0.08083080 -0.21357860
[5]  0.08565242  0.12899411  0.10404712 -0.01985020
[9] -0.05066555 -0.05574471 -0.19845006  0.03228260
[13]  0.11246048 -0.14091983 -0.03532483  0.09504267
[17]  0.05662097  0.33112052 -0.32401370  0.02100940
[21]  0.03674833  0.11437174  0.03012790  0.02341090
[25] -0.07321306

```

En este caso también se tiene componentes principales:

```

> mds6$11[,1:2]
RS1      RS2
1  -1.54994426 -1.4918524
2  -0.88796845  0.4532899
3  -1.83879772  0.5955461
4  -1.55722439 -1.5736069
5   0.18862209  0.6310710
6   0.97520779  0.9504043
7   0.81005987  0.7665996
8   0.82308727 -0.1462525
9  -0.95061785 -0.3732942
10 -1.33555138 -0.4107166
...
16 -0.41886955  0.7002565
17  0.79132284  0.4171726
18 -0.29425973  2.4396337
19  1.30938316 -2.3872720
20  0.39850675  0.1547933
21  1.12672595  0.2707548
22  0.45792667  0.8426695
23 -0.05800265  0.2219767
24 -1.09576050  0.1724871
25  1.44923893 -0.5394201

```

y su representación es:

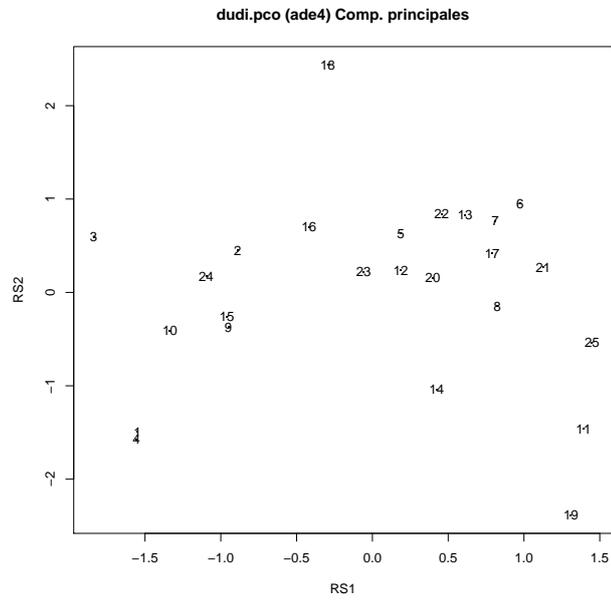


Figura B.1: Representación en dos dimensiones usando las componentes principales del comando `dudi.pco` del paquete `ade4`

En la figura B.1 se observa $-1,838798 \leq \text{Coordenada 1} \leq 1,449239$ y $-2,387272 \leq \text{Coordenada 2} \leq 2,439634$.

Apéndice C

Métodos de Kruskal en R

C.1. smacofSym (smacof)

Escalamiento multidimensional sobre una matriz de disimilitud usando *SMACOF* de Leeuw y Mair (2009).

Sintaxis:

```
smacofSym(delta, ndim = 2, type = c("ratio", "interval", "ordinal", "mspline"),
weightmat = NULL, init = "torgerson", ties = "primary", verbose = FALSE,
relax = FALSE, modulus = 1, itmax = 1000, eps = 1e-06,
spline.degree = 2, spline.intKnots = 2)
```

```
mds(delta, ndim = 2, type = c("ratio", "interval", "ordinal", "mspline"),
weightmat = NULL, init = "torgerson", ties = "primary", verbose = FALSE,
relax = FALSE, modulus = 1, itmax = 1000, eps = 1e-06,
spline.degree = 2, spline.intKnots = 2)
```

delta una matriz simétrica de disimilitud o un objeto de clase *dist*.

ndim Número de dimensiones.

weightmat Una matriz opcional con pesos de las disimilitudes.

init configuración inicial puede ser de *torgerson* (solución inicial de escalamiento clásico), *random* (configuración aleatoria), o una matriz definida por el usuario.

type tipo del MDS: “interval”, “ratio”, “ordinal” (MDS no métrico), o “mspline”.

ties especificación del TIE (solo MDS ordinal): “primary”, “secondary” o “tertiary”.

verbose si es *TRUE*, se muestra el estrés intermedio.

relax Si es *TRUE* se usa la relajación de bloques para mayorización.

modulus Número de iteraciones smacof por llamado de regresión monótona.

itmax número máximo de iteraciones.

eps criterio de convergencia.

spline.degree grado del spline para MDS de tipo “mspline”.

spline.intKnots número de nodos interiores del spline para MDS de tipo “mspline”.

Esta es la versión mas simple de MDS-SMACOF del paquete. Resuelve la función objetivo estrés para disimilitudes simétricas por medio de la aproximación por mayorización (SMACOF) y regresa el valor del Stress-1 (normalizado). La salida principal son las coordenadas en el espacio de dimensión baja (configuraciones, *conf*).

Salida

delta disimilitudes observadas, no normalizadas.

dhat Disparidades (proximidades transformadas, distancias aproximadas, d-hats).

confdiss Configuración de distancias.

conf Matrix de configuraciones fijas.

stress Valor del Stress-1.

spp Stress por puntos (contribución del estrés en porcentajes).

resmat Matriz con residuos al cuadrado.

rss suma de cuadrados residual.

weightmat matriz de pesos.

ndim Número de dimensiones.
init Configuración inicial.
model Nombre del modelo de smacof.
niter Número de iteraciones.
nobj Número de objetos.
type Tipo de modelo MDS.

C.2. isoMDS (MASS)

Escalamiento multidimensional de Kruskal no métrico Venables y Ripley (2002).

Sintaxis:

```
isoMDS(d, y = cmdscale(d, k), k = 2, maxit = 50, trace = TRUE,  
tol = 1e-3, p = 2)
```

```
Shepard(d, x, p = 2)
```

d una matriz simétrica o con estructura de distancia (*as.dist*). Se asume que los datos son disimilitudes o distancias relativas, pero deben ser positivas excepto por la distancia a si misma. Se permiten valores infinitos y ausentes.

y Una configuración inicial, sino se define ninguna se usa *cmdscale* para obtener una configuración inicial, a menos que hayan disimilitudes faltantes o infinitas.

k La dimensión deseada para la solución, pasado a *cmdscale*.

maxit El número máximo de iteraciones.

trace Valor lógico para tracing optimization. valor por defecto *TRUE*.

tol tolerancia para convergencia.

p potencia de la distancia de Minkowski para la configuración espacial.

x La configuración final.

Escoge una configuración k -dimensional para minimizar el estrés, la raíz cuadrada del cociente de la suma de las diferencias al cuadrado entre las distancias de entrada y aquellas de la configuración entre la suma de las distancias de la configuración al cuadrado. Sin embargo se permite una transformación monótona de las distancias de entrada.

Se usa un algoritmo iterativo, el cual usualmente converge en alrededor de 10 iteraciones. Se necesitan un cálculo de $O(n^2)$, por lo cual es lento para conjuntos de datos grandes. Además, para el valor por defecto de p ($p = 2$) la configuración solo se determina para rotaciones y reflexiones (con centroide en el origen por convención), el resultado puede variar considerablemente de máquina a máquina.

Salida

points Un vector de k de la configuración ajustada.

stress el estrés final obtenido (en porcentaje).

Si `trace` es `TRUE` se muestra el estrés inicial y el estrés actual cada 5 iteraciones.

C.3. `sammon` (MASS)

Un tipo de escalamiento multidimensional no métrico Venables y Ripley (2002).

Sintaxis:

```
sammon(d, y = cmdscale(d, k), k = 2, niter = 100, trace = TRUE,  
magic = 0.2, tol = 1e-4)
```

d una matriz simétrica o con estructura de distancia (*as.dist*). Se asume que los datos son disimilitudes o distancias relativas, pero deben ser positivas excepto por la distancia a si misma. Se permiten valores ausentes.

y Una configuración inicial, sino se define ninguna se usa *cmdscale* para obtener una configuración inicial, (si hay disimilitudes ausentes en *d*, se debe proporcionar una configuración inicial). No debe tener valores duplicados.

k La dimensión deseada para la solución.

niter El número máximo de iteraciones.

trace Valor lógico para tracing optimization. valor por defecto *TRUE*.

magic valor inicial del tamaño de paso constante en el método de Newton Diagonal.

tol tolerancia para detenerse, en unidades del estrés.

Escoge una configuración bidimensional para minimizar el estrés, la suma de las diferencias al cuadrado entre las distancias de entrada y aquellas de la configuración ponderadas por la distancia, toda la suma se divide por la suma de las distancias de entrada para hacer el estrés sin escalar.

Se usa un algoritmo iterativo, el cual usualmente converge en alrededor de 50 iteraciones. Se necesita un cálculo de $O(n^2)$, por lo cual es lento para conjuntos de datos grandes. Además, para el valor por defecto de p ($p = 2$) la configuración solo se determina para rotaciones y reflexiones (con centroide en el origen por convención), el resultado puede variar considerablemente de máquina a máquina. En esta versión el algoritmo ha sido modificado, agregando una búsqueda de longitud de paso (*magic*) para asegurar que siempre disminuya.

Salida

points Un vector de k de la configuración ajustada.

stress el estrés final obtenido (en porcentaje).

Si *trace* es *TRUE* es muestra el estrés inicial y el estrés actual cada 10 iteraciones.

C.4. metaMDS (vegan)

Oksanen *et al.* (2018)

Sintaxis:

```
metaMDS(comm, distance = "bray", k = 2, try = 20, trymax = 20,  
engine = c("monoMDS", "isoMDS"), autotransform = TRUE,  
noshare = (engine == "isoMDS"), wascores = TRUE, expand = TRUE,
```

```

trace = 1, plot = FALSE, previous.best, ...)
## S3 method for class 'metaMDS'
plot(x, display = c("sites", "species"), choices = c(1, 2),
type = "p", shrink = FALSE, ...)
## S3 method for class 'metaMDS'
points(x, display = c("sites", "species"),
choices = c(1,2), shrink = FALSE, select, ...)
## S3 method for class 'metaMDS'
text(x, display = c("sites", "species"), labels,
choices = c(1,2), shrink = FALSE, select, ...)
## S3 method for class 'metaMDS'
scores(x, display = c("sites", "species"), shrink = FALSE,
choices, ...)
metaMDSdist(comm, distance = "bray", autotransform = TRUE,
noshare = TRUE, trace = 1, commname, zerodist = "ignore",
distfun = vegdist, ...)
metaMDSiter(dist, k = 2, try = 20, trymax = 20, trace = 1, plot = FALSE,
previous.best, engine = "monoMDS", maxit = 200,
parallel = getOption("mc.cores"), ...)
initMDS(x, k=2)
postMDS(X, dist, pc=TRUE, center=TRUE, halfchange, threshold=0.8,
nthreshold=10, plot=FALSE, ...)
metaMDSredist(object, ...)

```


Apéndice D

Método de Procrustes en R

D.1. `procrustes` (vegan)

La función `procrustes`

Sintaxis:

```
procrustes(X, Y, scale = TRUE, symmetric = FALSE, scores = "sites", ...)
```

X matriz objetivo.

Y matriz a ser rotada.

scale permite el escalamiento de ejes de *Y*.

symmetric usa estadística de Procrustes simétrica.

scores tipos de objetivos usados.

La rotación de Procrustes rota una matriz de similitud máxima con una matriz objetivo minimizando la suma de los cuadrados de las diferencias. La rotación de Procrustes típicamente se usa en la comparación de resultados ordenados. Se usa particularmente en la comparación de soluciones alternas en el escalamiento multidimensional. Si `scale=FALSE`, la función solo rota la matriz *Y*. Si `scale=TRUE`, se escala la configuración *Y* linealmente para una máxima similitud.

Salida

Yrot matriz rotada *Y*.

X matriz objetivo.

ss suma de cuadrado de las diferencias.

rotation matriz de rotación ortogonal.

translation traslación del origen.

scale factor de escalamiento.

xmean el centroide del objetivo

symmetric tipo de estadístico ss.

call función call.

Referencias

- Borg, I., & Groenen, P. (2005). *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media.
- Bourgeois, P. (1978). Recherche du déplacement minimisant la distance entre deux ensembles de points homologues. *Cahiers de l'Analyse des Données*, 3(4), 440-448.
- Camiz, S. (2016). A theoretical background for principal component analysis.
- Camiz, S. (2017a). Principal component analysis.
- Camiz, S. (2017b). Procrustes statistics.
- Casas, F. M. G. (2012). El análisis de escalamiento multidimensional: Una alternativa y un complemento a otras técnicas multivariantes. *La Sociología en sus escenarios*(25).
- Chávez, C. (2004). *Algebra lineal*. Moshera.
- Cox, T. F., & Cox, M. A. (2000). *Multidimensional scaling*. CRC press.
- Davis, A. W. (1978). On the asymptotic distribution of gower's m^2 goodness-of-fit criterion in a particular case. *Annals of the Institute of Statistical Mathematics*, 30(1), 71-79.
- de Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software*, 31(3), 1-30. Descargado de <http://www.jstatsoft.org/v31/i03/>
- Dray, S., & Dufour, A. B. (2007). The ade4 package: implementing the duality diagram for ecologists. *Journal of Statistical Software*, 22(4), 1-20.

- Golub, G. H., & Loan, C. F. V. (1996). *Matrix computations*. Baltimore: John Hopkins University Press.
- Goslee, S. C., & Urban, D. L. (2007). The ecodist package for dissimilarity-based analysis of ecological data. *Journal of Statistical Software*, *22*, 1-19.
- Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, *53*(3-4), 325–338.
- Gower, J. C. (1967). Multivariate analysis and multidimensional geometry. *Journal of the Royal Statistical Society: Series D (The Statistician)*, *17*(1), 13–28.
- Gower, J. C. (1971). Statistical methods of comparing different multivariate analyses of the same data. *Mathematics in the archaeological and historical sciences*, 138–149.
- Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*(1), 33–51.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.
- Jolliffe, I. T. (2002). Principal components in regression analysis. *Principal component analysis*, 167–198.
- Kruskal, J. B. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*(1), 1–27.
- Kruskal, J. B. (1964b). Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, *29*(2), 115–129.
- Mueller-Dombois, D., & Ellenberg, H. (1974). Aims and methods of vegetation ecology.
- Oksanen, J., Blanchet, F. G., Friendly, M., Kindt, R., Legendre, P., McGlinn, D., . . . Wagner, H. (2018). *vegan: Community ecology package* [Manual de software informático]. Descargado de <https://CRAN.R-project.org/package=vegan> (R package version 2.4-6)
- Orloci, L. (1966). Geometric models in ecology: I. the theory and application of some ordination methods. *The Journal of Ecology*, 193–215.

- Paradis, E., Claude, J., & Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, *20*, 289-290.
- Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559-572.
- Ranwell, D. (1958). Movement of vegetated sand dunes at newborough warren, anglesey. *The Journal of Ecology*, 83-100.
- Ranwell, D. (1959). Newborough warren, anglesey: I. the dune system and dune slack habitat. *The Journal of Ecology*, 571-601.
- Ranwell, D. (1960). Newborough warren, anglesey: Ii. plant associes and succession cycles of the sand dune and dune slack vegetation. *The Journal of Ecology*, 117-141.
- Roberts, D. W. (2016). labdsv: Ordination and multivariate analysis for ecology [Manual de software informático]. Descargado de <https://CRAN.R-project.org/package=labdsv> (R package version 1.8-0)
- Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, *31*(1), 1-10.
- Sibson, R. (1978). Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 234-238.
- Team, R. C. (2016). R: A language and environment for statistical computing [Manual de software informático]. Vienna, Austria. Descargado de <https://www.R-project.org/>
- Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, *17*(4), 401-419.
- Torgerson, W. S. (1958). Theory and methods of scaling.
- Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with s* (Fourth ed.). New York: Springer. Descargado de <http://www.stats.ox.ac.uk/pub/MASS4> (ISBN 0-387-95457-0)