

Análisis multivariante de la comunidad

Carlos Iván Espinosa

Octubre 2016

Contents

Prefacio	5
Objetivos	7
1 Medidas de similitud	9
1.1 Medidas de abundancia	9
1.2 Distancias entre sitios	11
1.3 Similitud	12
1.4 Ejercicio: Análisis de similitud	14
2 Análisis multivariado de la composición de la comunidad	15
2.1 Agrupamiento Jerárquico (Hierarchic Cluster)	15
2.2 Interpretando el cluster	18
2.3 Ejercicio 2: Análisis de clasificación	22
3 Ordenaciones	25
3.1 Pasos previos: Transformación y Estandarización de datos	25
3.2 Ordenación indirecta o no constreñida	27
3.3 Pasos para el análisis de Ordenación	28
4 Ordenaciones Directas o Constreñidas	33
4.1 Realizando una ordenación constreñida	33
4.2 Ejercicio 3: Análisis de Ordenación	37

Prefacio

La comunidad biológica se refiere a una agrupación de poblaciones de especies que se presentan juntas en el espacio y el tiempo (Begon et al. 1999). Este concepto plantea que las comunidades tienen unos límites en el espacio y el tiempo, y que estos límites están dados por la distribución de las poblaciones. Sin embargo, la distribución de las poblaciones no es homogénea y cada población responde diferente en el espacio y el tiempo.

De esta forma la caracterización de una comunidad biológica se constituye en un reto ya que implica poder rescatar los efectos que se dan a varios niveles en la comunidad. El definir por ejemplo ¿Dónde inicia y termina una comunidad? o ¿Cómo difieren las comunidades entre localidades? o ¿Cómo la comunidad responde a las condiciones ambientales o disturbios? representan algunas de las principales preguntas que necesitamos responder. Una de las formas de responder estas preguntas puede ser intentar cuantificar las similitudes entre localidades.

Objetivos

- Comprender las bases teóricas para el cálculo de similitudes de la estructura de la comunidad entre localidades.
- Utilizar herramientas de análisis para calcular índices de similitud y distancias entre comunidades.



Figure 1: *Stenocercus iridicens*

Chapter 1

Medidas de similitud

La caracterización de una comunidad biológica presenta varios retos a los ecólogos. ¿Dónde inicia y termina una comunidad? ¿Cómo difieren las comunidades entre localidades? ¿Cómo la comunidad responde a las condiciones ambientales o disturbios? ¿Cómo se mantiene la diversidad en un área determinada? son algunas de las temáticas con mayor desarrollo científico dentro de la ecología de comunidades. En el presente capítulo se introduce a los estudiantes en los conceptos que los ecólogos utilizan para comparar comunidades y se presenta una guía de algunos de los análisis básicos de análisis de la estructura de las comunidades.

1.1 Medidas de abundancia

“La abundancia se refiere al número de individuos de una especie en una determinada área”

— (Smith and Smith 2010)

Cuando hablamos de la composición de especies de una comunidad nos referimos al conjunto de especies que habitan una determinada localidad. Típicamente, esto incluye cierto grado de abundancia de cada especie, pero puede también ser simplemente un listado de especies en esa localidad, donde se registra la presencia o ausencia de cada especie. Ahora, imaginemos que tenemos cuatro localidades (A, B, C, D) donde recogemos los datos de densidad de dos especies; *Tabebuia billbergii* y *Geofroea spinosa*, especies características de bosques secos tropicales. Podemos introducir datos hipotéticos de abundancia para cada especie en cada una de las localidades.

```
dens <- data.frame(T.bil = c(1, 1, 2, 3), G.spi = c(21, 8, 13, 5))
row.names(dens) <- LETTERS[1:4]
dens

##   T.bil G.spi
## A     1    21
## B     1     8
## C     2    13
## D     3     5
```

Generamos un gráfico para ver cuánto se parece cada sitio (Figura 1.1)

```
par(mar=c(4,4,1,1), mgp=c(1,0.3,0), tcl= -0.2)
plot(dens, type = "n", cex.axis=0.8)
text(dens, row.names(dens), col = "blue")
```

En la Figura 1.1 vemos que la composición de especies en el sitio A es diferente de la composición del sitio D. Es decir, la distancia entre el sitio A y D es mayor que entre los otros sitios. Lo siguiente que nos deberíamos

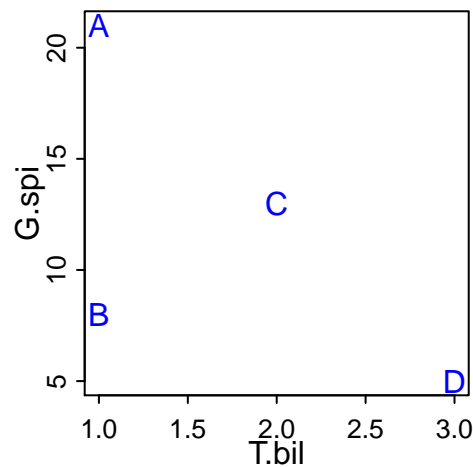


Figure 1.1: Distancias de cuatro localidades hipotéticas

preguntar es; ¿qué tan distantes están los dos sitios? Claramente, esto depende de la escala de medición (los valores de los ejes), y sobre cómo medimos la distancia a través del espacio multivariado (Stevens, 2009).

Estas diferencias entre sitios son dependientes de la abundancia de cada especie. En el caso de *G. spinosa* su eje varía entre 5 y 21, mientras que para *T. billbergii* varía entre 1 y 3. Una forma de corregir esta distorsión es calcular la densidad relativa de cada especie, de esta forma cada especie variará entre 0 y 1 (Stevens, 2009). Para ello dividimos la abundancia de cada especie para la suma total de los individuos de las especies en esa muestra.

“La distancia se refiere a la diferencia en un espacio multidimensional (dado por las especies) entre dos comunidades. Esta distancia puede ser medida por múltiples vías”

```
dens[1, ]/sum(dens[1, ])
```

```
##          T.bil      G spi
## A 0.04545455 0.9545455
```

Este resultado nos muestra que el sitio A está constituido en un 95% por *G. spinosa*, mientras que *T. billbergii* aporta únicamente el 5%. Cuando nos referimos a densidad relativa hablamos de la densidad de una especie con referencia a algo, en el caso anterior con referencia a otras especies en el mismo sitio, pero también podríamos calcular en relación a otros sitios la misma especie.

```
dens[, 1]/sum(dens[, 1])
```

```
## [1] 0.1428571 0.1428571 0.2857143 0.4285714
```

Ahora podemos ver cómo *T. billbergii* varía en su abundancia en los cuatro sitios. El sitio A y B tienen el 14% de individuos mientras que el D tiene el 42% de los individuos de esta especie.

Ya sea que nuestras medidas de abundancia son absoluta o relativa, nos interesa conocer cuán diferente es la comunidad de una muestra (o sitio) con relación a la otra. En el ejemplo ha sido fácil entender la diferencia entre las dos comunidades debido a que teníamos únicamente dos especies, pero con más de tres especies es complicado observar estas diferencias gráficamente. Tal vez la forma más sencilla de describir la diferencia entre los sitios es calcular las *distancias* entre cada par de sitios.

1.2 Distancias entre sitios

La *distancia* entre dos muestras está dada por la diferencia entre la abundancia y la composición de especies, como lo hemos visto esto genera una distancia, en el caso del ejemplo la comunidad A esta más alejada de la comunidad D que de las otras dos.

Existen muchas formas de poder calcular las distancias entre estos puntos una de las más sencillas es la distancia *Euclidiana*. La distancia euclidiana entre dos sitios es simplemente la longitud del vector que conecta los sitios y la podemos obtener como $\sqrt{x^2 + y^2}$, donde “x” y “y” son las coordenadas (x, y) de distancia entre un par de sitios.

En nuestro caso si queremos comparar B y C tenemos que la distancia en el eje *x* es la diferencia de la abundancia de *T. bilbergii* entre el sitio B y C.

```
x <- dens[2, 1] - dens[3, 1]
```

Mientras que la distancia en el eje *y* es la diferencia en la abundancia de *G. spinosa* entre el sitio B y C.

```
y <- dens[2, 2] - dens[3, 2]
```

Ahora obtenemos las distancias entre los dos sitios

```
sqrt(x^2 + y^2)
```

```
## [1] 5.09902
```

Pero como en *R* todo es sencillo podemos utilizar la función *dist*

```
dist(dens)
```

```
##           A           B           C
## B 13.000000
## C  8.062258  5.099020
## D 16.124515  3.605551  8.062258
```

Si bien este cálculo es sencillo con dos especies, si tenemos que calcular la distancia para una comunidad con más de tres especies los cálculos son tediosos y largos. Para calcular la distancia *Euclidiana* entre pares de sitios con *R* especies utilizamos la siguiente ecuación:

$$D_E = \sqrt{\sum_{i=1}^R (x_{ai} - x_{bi})^2}$$

Distancia Euclidiana

Existen otras formas de medir distancias entre dos localidades. En ecología una de las distancias más utilizada es la distancia de *Bray-Curtis*, conocida también como *Sorensen*. Esta distancia es calculada como:

$$D_{BC} = \sum_{i=1}^R \frac{(x_{ai} - x_{bi})}{(x_{ai} + x_{bi})}$$

Distancia de Bray-Curtis

La distancia *Bray-Curtis* no es más que la diferencia total en la abundancia de especies entre dos sitios, dividido para la abundancia total en cada sitio. La distancia Bray-Curtis tiende a resultar más intuitiva debido a que las especies comunes y raras tienen pesos relativamente similares, mientras que la distancia euclidiana depende en mayor medida de las especies más abundantes. Esto sucede porque las distancias euclidianas se basan en diferencias al cuadrado, mientras que Bray-Curtis utiliza diferencias absolutas. El elevar un número al cuadrado siempre amplifica la importancia de los valores más grandes. En la figura 1.2 se compara gráficos basados en distancias euclidianas y Bray-Curtis de los mismos datos.

Como se había comentado es virtualmente imposible representar una distancia en más de tres dimensiones (cada especie es una dimensión). Una forma sencilla de mostrar distancias para tres o más especies es crear un gráfico de dos dimensiones, intentando organizar todos los sitios para que las distancias sean aproximadamente las correctas. Está claro que esto es una aproximación nunca estas serán exactas. Una técnica que intenta crear un arreglo aproximado es escalamiento multidimensional no métrico (NMDS). Vamos a calcular las distancias para nuestra comunidad, primero vamos a añadir dos especies más a nuestra comunidad, *Ceiba trichistandra* y *Colicodendron scabridum*.

```
dens$C.tri<- c(11, 3, 7, 5)
dens$C.sca<- c(16, 0, 9, 4)
```

La función de escalamiento multidimensional no-métrico está en el paquete **vegan**. Aquí mostramos las distancias euclidianas entre sitios (Figura 1.2a) y las distancias de Bray-Curtis (Figura 1.2b).

```
library(vegan)

#Distancia Euclidiana
mdsE <- metaMDS(dens, distance = "euc", autotransform = FALSE, trace = 0)
#Distancia de Bray-Curtis
mdsB <- metaMDS(dens, distance = "bray", autotransform = FALSE, trace = 0)

par(mfcol=c(1,2), oma=c(1,1,1,1), mar=c(4,4,1,1),
    mgp=c(1,0.3,0), tcl= -0.2)

plot(mdsE, display = "sites",
     type = "text",main="a)Euclidiana",
     cex.axis= 0.7, cex.main=0.75, cex.lab=0.7)

plot(mdsB, display = "sites", type = "text",
     main="b)Bray-Curtis",
     cex.axis= 0.7, cex.main=0.75, cex.lab=0.7)
```

1.3 Similitud

Ahora que sabemos cuan distantes son los diferentes sitios, muchas veces nos podría interesar cuan similares son cada uno de los sitios a continuación se describen dos medidas de similitud; *Porcentaje de Similitud* e *Índice de Sorensen*.

El *porcentaje de similitud* puede ser simplemente la suma de los porcentajes mínimos de cada especie en la comunidad. Lo primero que debemos hacer es convertir la abundancia de cada especie a su abundancia relativa dentro de cada sitio. Para ello dividimos la abundancia de cada especie por la suma de las abundancias en cada sitio.

```
dens.RA <- t(apply(dens, 1, function(sp.abun) sp.abun/sum(sp.abun)))
dens.RA
```

```
##          T.bil      G.spi      C.tri      C.sca
## A 0.02040816 0.4285714 0.2244898 0.3265306
## B 0.08333333 0.6666667 0.2500000 0.0000000
## C 0.06451613 0.4193548 0.2258065 0.2903226
## D 0.17647059 0.2941176 0.2941176 0.2352941
```

El siguiente paso para comparar entre sitios, es encontrar el valor mínimo para cada especie entre los sitios que debemos comparar. Vamos a comparar los sitios A y B, para esto utilizamos la función **aply**, la cual nos permite encontrar el valor mínimo entre las filas 1 y 2 (sitio A y B respectivamente). Para *T. billbergi* en el sitio A la abundancia relativa es 0.02 que es menor a la abundancia en el sitio B que es de 0.08.

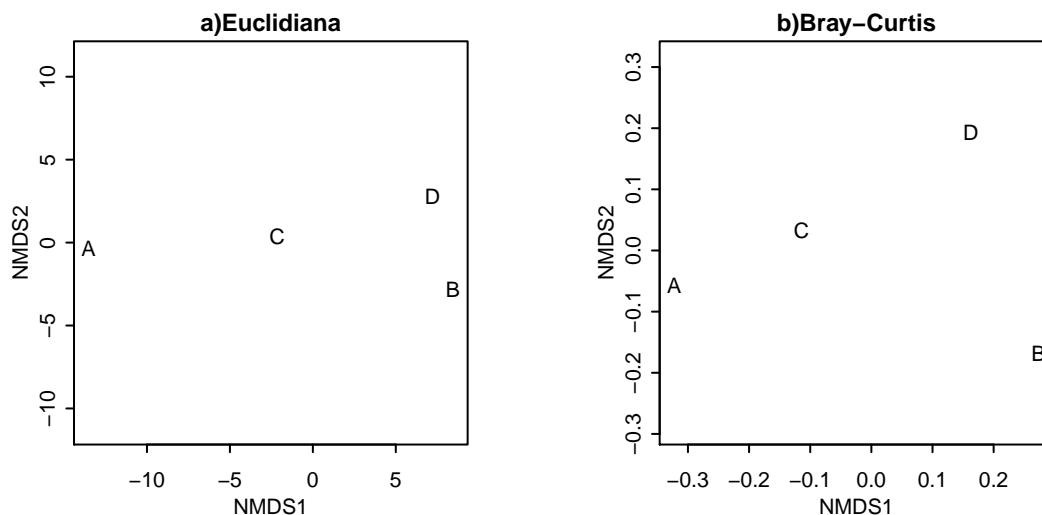


Figure 1.2: Arreglo de las parcelas en distancias multidimensionales no métricas (NMDS). Estas dos figuras muestran los mismos datos en bruto, pero las distancias euclidianas tienden a enfatizar las diferencias debidas a las especies más abundantes, mientras que Bray-Curtis no lo hace.

```
mins <- apply(dens.RA[1:2, ], 2, min)
mins
```

```
##      T.bil      G.spi      C.tri      C.sca
## 0.02040816 0.42857143 0.22448980 0.00000000
```

Finalmente para conocer el porcentaje de similitud entre los dos sitios sumamos estos valores y multiplicamos por 100.

```
sum(mins)*100
```

```
## [1] 67.34694
```

Esto significa que la comunidad A y B tienen un porcentaje de similitud del 67%.

El índice de Sorensen es la segunda medida de similitud que vamos a estudiar, este índice es medido como:

$$S_s = \frac{(2C)}{(A + B)}$$

Índice de Sorensen

Donde C es el número de especies en común entre los dos sitios, y A y B son el número de especies en cada sitio. Esto es equivalente a dividir las especies compartidas por la riqueza media.

Para calcular el índice de Sorensen entre los sitios A y B necesitamos definir el número de especies compartidas y luego la riqueza de cada uno de los dos sitios.

Definimos si alguna de las especies en uno de los sitios la abundancia no es igual a cero, eso nos dirá en qué casos se comparten especies. Finalmente, sumamos todas las especies que su abundancia es mayor a cero.

Table 1.1: Comunidades hipotéticas

	sp1	sp2	sp3	sp4	sp5	sp6	sp7	sp8
A	26	17	16	1995	159	0	362	0
B	0	35	14	236	54	0	496	57
C	24	0	26	17	88	18	907	20
D	35	18	24	2033	175	15	376	16
E	105	129	40	18	191	53	964	134

```
comp<- apply(dens[1:2, ], 2, function(abuns) all(abuns != 0))
comp
```

```
## T.bil G.spi C.tri C.sca
## TRUE TRUE TRUE FALSE
```

```
Rs <- apply(dens[1:2, ], 1, function(x) sum(x > 0))
Rs
```

```
## A B
## 4 3
```

Como vemos, la abundancia de *C. scabridum* en uno de los dos sitios es igual a Cero, lo confirmamos al tener la riqueza por sitio. El sitio B tenemos únicamente 3 especies.

Ahora aplicamos la formula, dividimos las especies compartidas (*comp*) para la riqueza total de los dos sitios y lo multiplicamos por 2.

```
(2*sum(comp))/sum(Rs)
```

```
## [1] 0.8571429
```

Según el índice de Sorensen estos dos sitios son parecidos en un 86%. Los datos de los dos índices utilizados difieren entre sí, el porcentaje de similitud utiliza no solamente la presencia ausencia sino también la abundancia lo que podría estar reduciendo la similitud entre sitios.

1.4 Ejercicio: Análisis de similitud

Una de las preguntas básicas de un ecólogo es saber ¿Cómo de diferentes son dos comunidades?, en el presente ejercicio nos interesa entender comprender la similitud y distancias entre estas cinco comunidades hipotéticas (tabla 1.1)

Con los datos anteriores:

- Convierta los datos en abundancia relativa por sitio (la suma en cada sitio debe ser igual a 1). Dibuje dos gráficas para representar; i) la abundancia total y ii) abundancia relativa de cada localidad. ¿Qué diferencias puede ver en la gráfica i y en la ii?
- Calcule la distancia Euclideana y de Bray Curtis para cada sitio con las dos medidas de abundancia y grafíquelas utilizando el NMDS. ¿Cómo cambia entre distancias y abundancias? Explique las diferencias.
- Evalúe la similitud (Sorensen) y el porcentaje de similitud entre pares de sitios. ¿Cuáles son los sitios más similares? ¿Cuál es la razón de las diferencias entre los índices utilizados?

Chapter 2

Análisis multivariado de la composición de la comunidad

Los índices de similitud nos permiten comparar las comunidades entre dos sitios, pero claramente cuando estudiamos las comunidades nuestros datos no son tan sencillos como lo que hemos utilizado hasta el momento. El organizar los datos de composición de la comunidad y poder interpretarlos en relación a otras comunidades, entender que comunidades son más similares entre sí, y saber si esta similitud o distancia es el resultado de unas respuestas al entorno pueden ser algunas de las cosas que podremos responder utilizando las técnicas de análisis multivariado de la comunidad. A continuación vamos a describir algunas técnicas de clasificación y ordenación que nos permitirán abordar estas temáticas.

Las técnicas de ordenación y clasificación son estrategias alternativas para simplificar los datos. La ordenación intenta simplificar los datos en un mapa que muestra las similitudes entre los puntos. La clasificación simplifica datos colocando los puntos similares en una misma clase o grupo Oksanen 2014¹.

Utilizaremos el paquete *Vegan* para los análisis de ordenación y clasificación, para mayor información puede referirse a Oksanen 2013².

2.1 Agrupamiento Jerárquico (Hierarchic Cluster)

A continuación vamos a realizar un análisis Cluster (análisis de conglomerados) utilizando la función *hclust* del paquete *vegan*. La función *hclust* necesita una matriz de disimilitudes como entrada. El Análisis de conglomerados intenta generar conglomerados que tengan la máxima homogeneidad en cada grupo y la mayor diferencia entre los grupos.

Aunque la función *dist* nos permite calcular disimilitudes, para el análisis de comunidades biológicas utilizaremos la función *vegdist* del paquete *vegan*. Esta función nos permite calcular varios índices de disimilitud. El método de cálculo de la disimilitud por defecto es Bray-Curtis (*“bray”*).

Una de las características importantes del método Bray-Curtis es que varía entre 0 y 1, dos comunidades que no comparten ninguna especie tendrían 1 como resultado.

Calculemos una matriz de disimilitudes usando el método Bray-Curtis, utilizaremos los datos de Barro Colorado Island (BCI) cargados en el paquete *vegan*. Para eso necesitamos cargar el paquete y los datos de BCI, únicamente utilizaremos los datos de los primeros 10 sitios.

¹<http://cc.oulu.fi/~jarioksa/opetus/metodi/session3.pdf>

²<http://cc.oulu.fi/~jarioksa/opetus/metodi/vegantutor.pdf>

```
library(vegan)
data(BCI)

dist<- vegdist(BCI[1:10,], method="bray")
dist[1:10]
```

```
## [1] 0.2706682 0.3501647 0.3682008 0.3725079 0.3744186 0.3518519 0.3424346
## [8] 0.4235706 0.3770140 0.2873051
```

Podemos ver que el sitio 1 es 27% diferente al sitio 2, 35% al sitio 3, 36% al sitio 4 y así sucesivamente con los 10 sitios.

Con la matriz de disimilitudes calculada se puede analizar los puntos que conforman una agrupación. Utilizaremos los métodos de agrupación de la función *hclust* que nos propone tres métodos de agrupamiento: agrupación simple, agrupación completa y agrupación promedio.

Todos los métodos inician con el agrupamiento de las dos comunidades (dos sitios) más similares y a partir de esta primera comparación se continúa con el resto de puntos.

A continuación ejemplificaremos el cálculo de las distancias usando los tres métodos. Extraemos los cinco primeros sitios de la matriz de BCI y generamos un nuevo objeto (S_BCI). Con este nuevo objeto calculamos la distancia entre los cinco sitios.

```
S_BCI<- BCI[1:5,]
dist1<- vegdist(S_BCI, method="bray")
dist1
```

```
##           1           2           3           4
## 2 0.2706682
## 3 0.3501647 0.2873051
## 4 0.3682008 0.3149523 0.3244078
## 5 0.3725079 0.3851064 0.3595041 0.3721619
```

1. En base de la matriz de disimilitudes se busca el par de puntos que se encuentren más cercanos (menos disimiles). En nuestro caso el punto 1 y 2 tienen la distancia más baja 0.27. Una vez identificado, inicia el proceso de agrupación y es donde se diferencian los tres métodos.
2. Con el primer grupo generado debemos comenzar la construcción del resto de grupos, para esto construimos una nueva matriz de disimilitud calculando las distancias desde este primer grupo (1-2) al resto de sitios. El cálculo de esta distancia es dependiente del método.

Recuerde, para los sitios del 3 al 5 tendremos dos distancias, la distancia desde el sitio 1 y del sitio 2 a cada uno de estos sitios. Por tanto utilizaremos estas dos distancias para calcular la distancia desde el grupo.

- En el método de agrupación simple la distancia entre el grupo y el sitio 3 será igual a la distancia más baja comparando entre la distancia del sitio 1 y el sitio 2. En el caso de la distancia al sitio 3 el valor mínimo es 0.287.
- En el método completo el nuevo valor de distancia será el valor más alto, en este caso 0.350, y
- En el método de agrupación promedio, obtenemos el valor promedio entre las distancias primer grupo y el sitio 3 en este caso 0.318 (Tabla 1).

Tabla 1. Cálculo de nuevas distancias entre el grupo 1 (sitio 1 y 2) y los sitios restantes. **A. simple:** cálculo de distancia mediante el método de agrupación simple. **A. completa:** cálculo de distancia mediante el método de agrupación completa. **A. promedio:** cálculo de distancia mediante el método de agrupación promedio.

Sitios	Sitio 1	Sitio 2	A. Simple	A. Completa	A. Media
Sitio 3	0.3501647	0.2873051	0.2873051	0.3501647	0.3187349

Sitios	Sitio 1	Sitio 2	A. Simple	A. Completa	A. Media
Sitio 4	0.3682008	0.3149523	0.3149523	0.3682008	0.3415765
Sitio 5	0.3725079	0.3851064	0.3725079	0.3851064	0.3788071

3. A partir de estos cálculos se construye nuevamente la matriz de distancia. Mostramos las nuevas matrices de distancias según el método de agrupación utilizado.

Para el método de agrupación **simple**

Sitio	Grupo1-2	Sitio 3	Sitio 4
3	0.2873051		
4	0.3149523	0.3244078	
5	0.3725079	0.3595041	0.3721619

Para el método de agrupación **completo**

Sitio	Grupo1-2	Sitio 3	Sitio 4
3	0.3501647		
4	0.3682008	0.3244078	
5	0.3851064	0.3595041	0.3721619

Para el método de agrupación **promedio**

Sitio	Grupo1-2	Sitio 3	Sitio 4
3	0.3187349		
4	0.3415765	0.3244078	
5	0.3788071	0.3595041	0.3721619

4. Se repite el procedimiento, se busca los puntos que tienen la menor disimilitud en la nueva matriz y se vuelve a calcular las distancias desde este nuevo grupo al resto de grupos, esto se repite tantas veces hasta que todos los sitios están asociados.

Podemos calcular directamente la agrupación utilizando la función *hclust*, y graficarlo con la función *plot*.

```
par(mfcol=c(1,3))
```

```
csim <- hclust(dist1, method="single")
ccom <- hclust(dist1, method="complete")
cpro <- hclust(dist1, method="average")

plot(csim, cex.axis=0.7)
plot(ccom, cex.axis=0.7)
plot(cpro, cex.axis=0.7)
```

Como podemos ver en la figura 2.1 en todos los casos el primer grupo es el mismo, el grupo entre los sitios 1 y 2 con una disimilitud de 0.27, a partir de este punto los dendrogramas varían según el método utilizado. En el caso del método simple la disimilitud más baja es entre el grupo 1-2 y el sitio 3, con una disimilitud del 0.287. En el caso del método completo la disimilitud más baja se da entre el sitio 3 y 4 que conforman un segundo grupo con una disimilitud de 0.32. Finalmente, en el caso del método de promedio la menor disimilitud se da entre el grupo 1-2 y el sitio 3 con una disimilitud de 0.31 (Figura 2.1)

Los métodos de agrupamiento jerárquico (cluster) producen clasificaciones donde todas las observaciones se

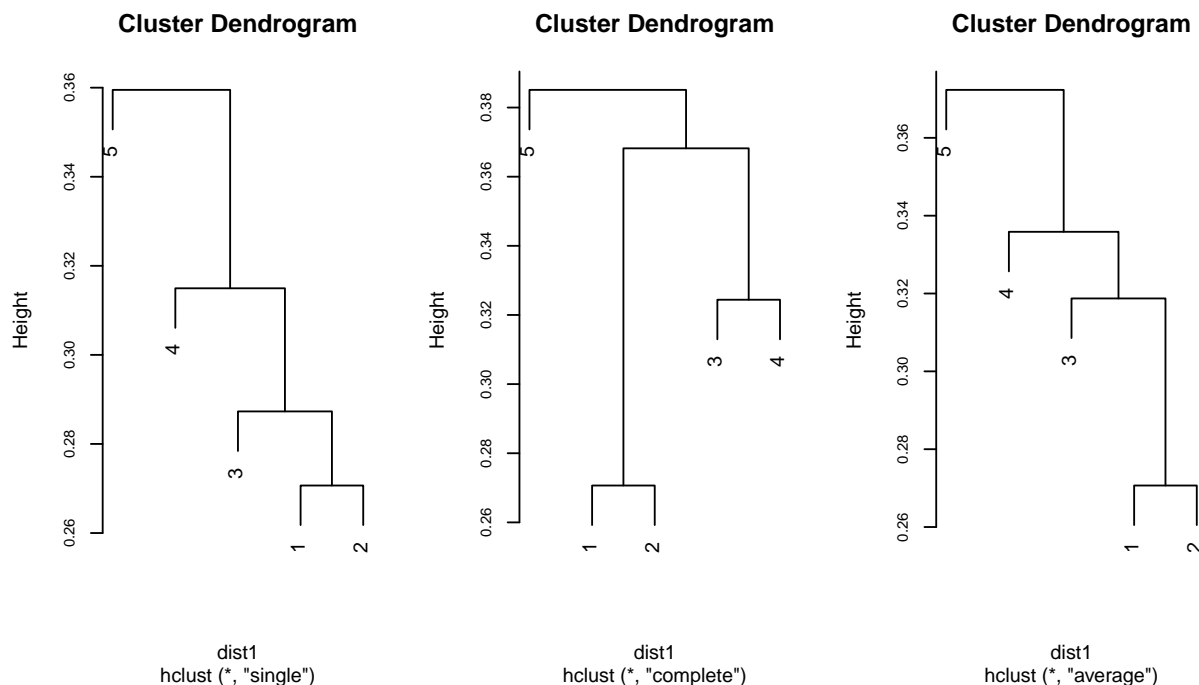


Figure 2.1: Dendrograma construido a partir de los 3 métodos de agrupación

encuentran agrupadas de diferente forma. En los extremos todas las observaciones se encuentran agrupadas en una sola clase o cada observación conforma su clase privada, entre estos extremos las observaciones forman diferentes agrupamientos con niveles de disimilitud variables. Normalmente nos interesa tener un cierto número de clases con niveles de disimilitud establecido. La conformación de estos grupos se puede mostrar visualmente con función **rect.hclust** (Figura 2.2)

```
par(mar=c(2,3,4,2))
plot(ccom, hang=-0.1, cex.axis=0.7, cex.lab=0.8, cex.main=0.8)
rect.hclust(ccom, 3)
```

Ahora podríamos obtener la pertenencia a un grupo y relacionarlo con otra variable explicativa, y analizar si la generación del grupo responde a algún factor.

```
grupo <- cutree(ccom, 3)
grupo
```

```
## 1 2 3 4 5
```

```
## 1 1 2 2 3
```

2.2 Interpretando el cluster

El análisis de conglomerados (cluster) no es un test estadístico, y como vimos hay varios factores que pueden afectar la generación de los grupos (Borcard et al., 2011), por lo que debemos ser conscientes de lo que obtenemos como resultado. Podemos usar la función **summary()** para ver la información que tenemos luego de haber utilizado el **hclust**, estos datos pueden ser utilizados para interpretar el agrupamiento (Borcard et al., 2011).

Como vimos anteriormente el investigador puede decidir, en función de su experiencia y de los árboles generados, cuantos grupos se generan dentro del árbol y que metodo de agrupamiento utilizar, sin embargo,

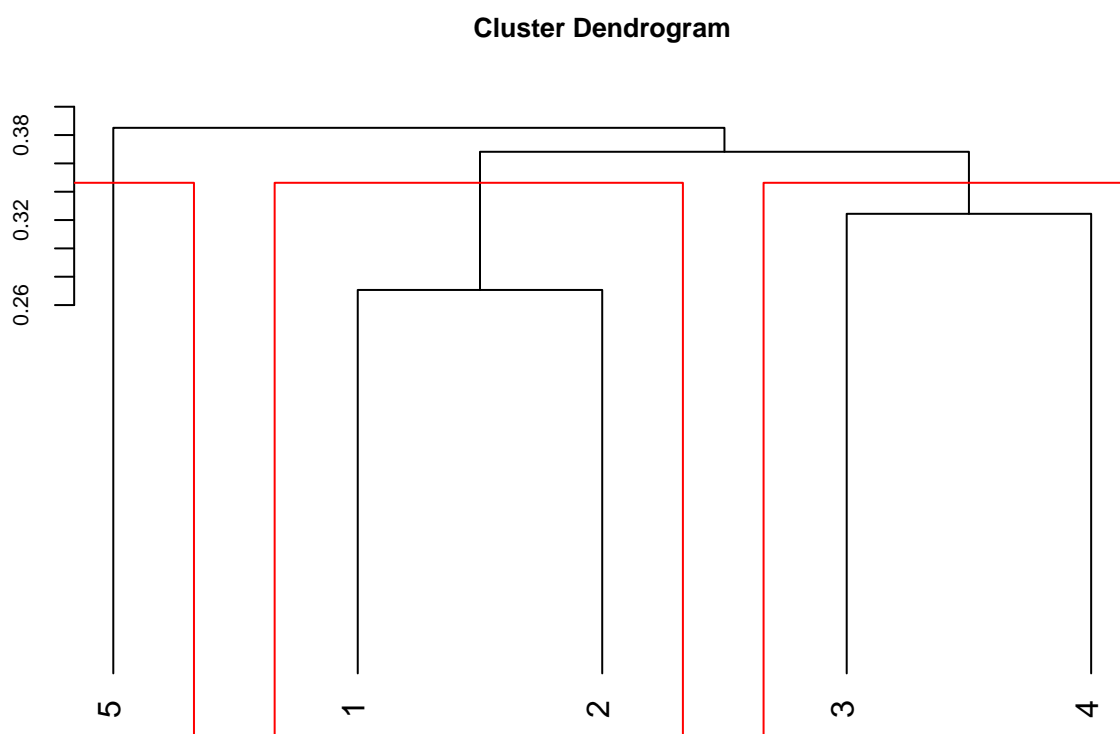


Figure 2.2: Dendrograma con número de grupos

podemos utilizar algunas funciones que nos permitan determinar grupos consistentes.

2.2.1 Elegir la función de enlace

Una forma que podemos utilizar para definir los grupos es la distancia Cofenética. Esta distancia es calculada como la distancia entre dos objetos de un mismo grupo en el dendrograma, la distancia desde el primer objeto al segundo objeto pasando por el nodo de unión de los dos objetos es la distancia Cofenética. Una matriz cofenética es una matriz que representa las distancias cofenéticas entre todos los pares de objetos. Con esta matriz podemos correlacionar con la matriz de disimilitud original. El método con la correlación cofenética más alta puede ser vista como la que produjo el mejor modelo de agrupación para la matriz de distancia.

```
#Calculamos la matriz cofenética para cada método de
#agrupamiento

csim_coph <- cophenetic(csim)
cpro_coph <- cophenetic(cpro)
ccom_coph <- cophenetic(ccom)

#Calculamos la correlación
cor(csim_coph, dist1); cor(cpro_coph, dist1); cor(ccom_coph, dist1)

## [1] 0.8143114
## [1] 0.846916
## [1] 0.7487461
```

Según estos datos el método promedio es el método que produce un mejor agrupamiento.

Otra forma de evaluar el mejor método es calcular la distancia de Gower, calculado como la suma de los cuadrados de la diferencia entre la matriz de distancia y la distancia Cofenética, el menor valor significa que es el mejor método de agrupamiento.

```
sim_gow <- sum((dist1-csim_coph)^2)
pro_gow <- sum((dist1-cpro_coph)^2)
com_gow <- sum((dist1-ccom_coph)^2)

sim_gow; pro_gow; com_gow

## [1] 0.007860928
## [1] 0.003917673
## [1] 0.01068659
```

En este caso vemos que la decisión usando la distancia de Gower y la Cofenética es la misma, el método promedio produce el mejor agrupamiento. Sin embargo, no siempre el resultado es consistente entre los dos métodos.

Este proceso nos ha permitido obtener la mejor función de enlace, sin embargo, para definir cuales son los subconjuntos de datos (tener un punto de corte) se puede utilizar algunas otras herramientas.

2.2.2 Elegir el punto de corte

Como vimos anteriormente yo puedo definir un punto de corte para generar los grupos o puedo decidir cuantos grupos, sin embargo, este procedimiento es subjetivo. Podemos utilizar alguna información que nos permita tomar decisiones fundamentadas.

Podemos utilizar la **silhouette width (anchura de la silueta)** para medir el grado de pertenencia de un objeto a su agrupación, basado en la distancia media entre este objeto y todos los objetos de la agrupación a la que se pertenece, en comparación con la misma medida calculada para el siguiente grupo más cercano (Borcard et al., 2011). Utilizaremos la función **silhouette** del paquete **cluster**. La salida de esta función varía entre 1 y -1. Los valores negativos significan que los objetos correspondientes probablemente se han colocado en un grupo erróneo.

A continuación el proceso utilizado:

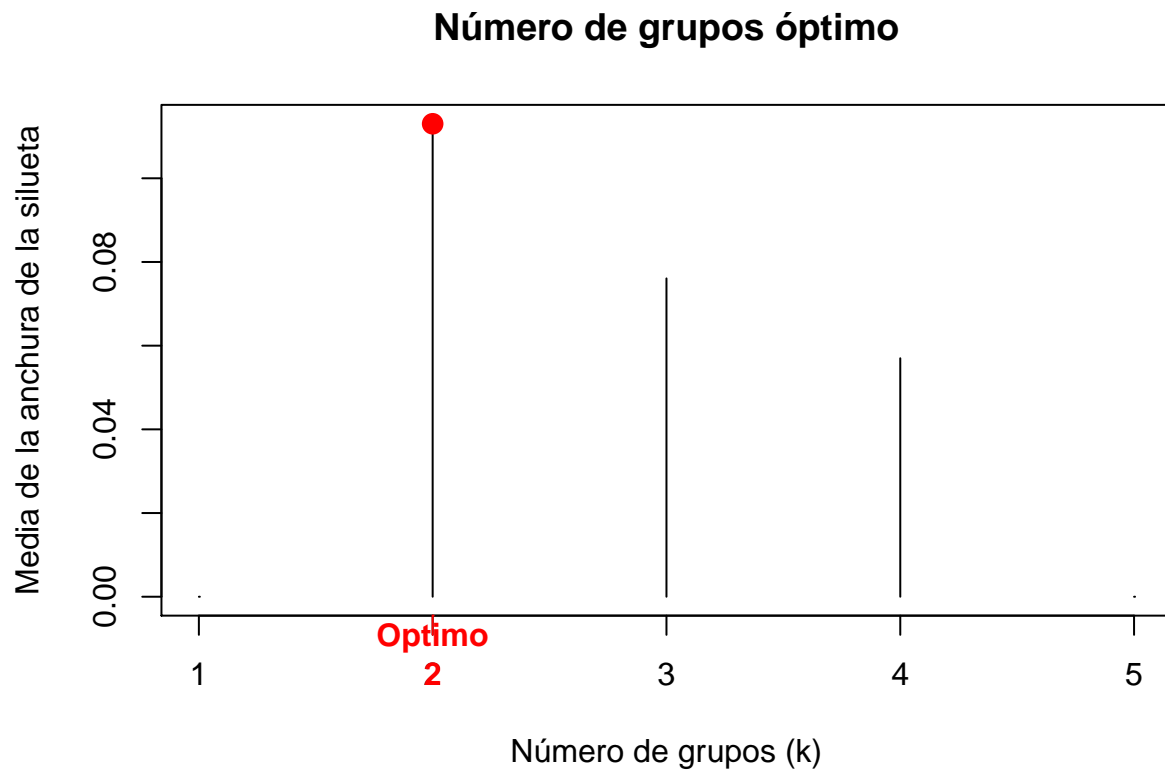
```
library(cluster)

#Generamos un vector vacío para colocar los valores
# medios de la anchura de la silueta (mas)
mas <- numeric(nrow(S_BCI))

#Calculamos y ponemos el <mas> en el vector generado
for( k in 2: (nrow(S_BCI)-1)){
  sil <- silhouette(cutree(ccom, k=k), dist1)
  mas[k] <- summary(sil)$avg.width
}

# Analizamos cual es el mejor punto de corte
k.best <- which.max(mas)

# Graficamos
plot(1:nrow(S_BCI), mas, type = "h", main="Número de grupos óptimo",
     xlab = "Número de grupos (k)", ylab="Media de la anchura de la silueta")
axis(1, k.best, paste("Óptimo", k.best, sep="\n" ),
     col="red", font=2, col.axis="red")
points(k.best, max(mas), pch=16, col="red", cex=1.5)
```



```
cat("", "Número óptimo de grupos k=", k.best, "\n",
      "Con un valor medio de anchura de la silueta de", max(mas), "\n")
```

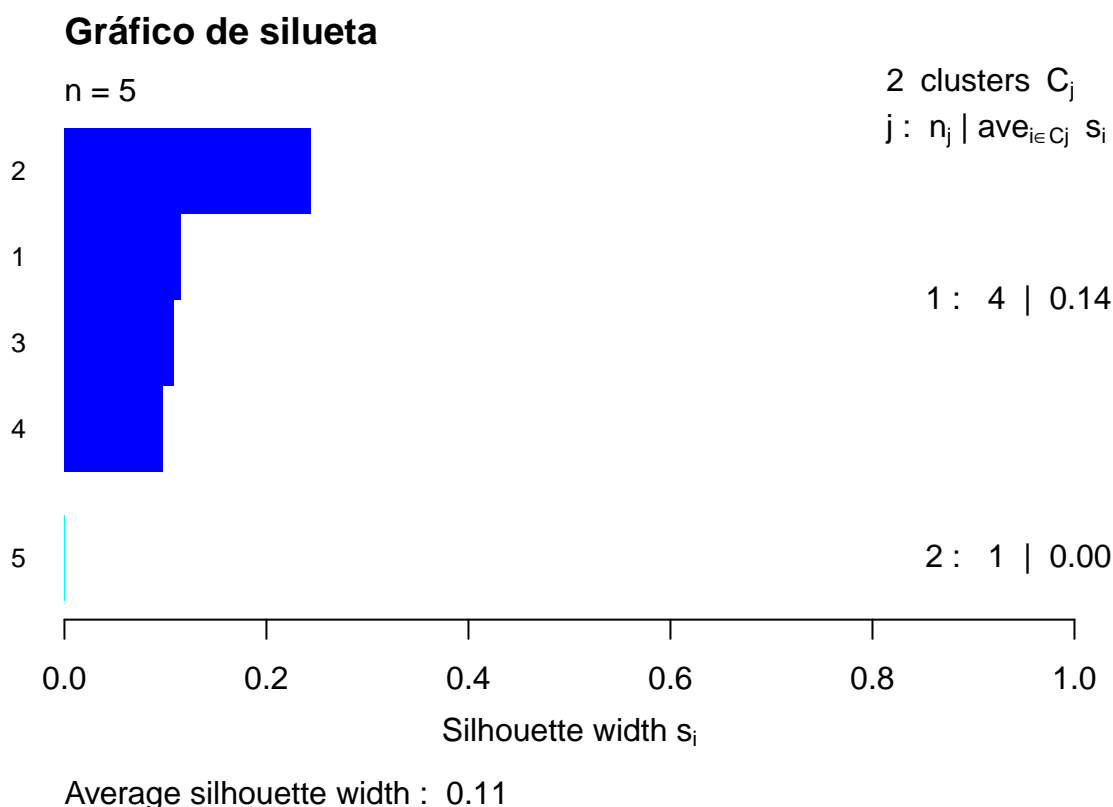
```
## Número óptimo de grupos k= 2
## Con un valor medio de anchura de la silueta de 0.1130222
```

A partir de este punto podría utilizar otras herramientas para definir el número de grupos. Ahora nos interesa saber si los grupos están balanceados y bien delimitados. Podemos utilizar el gráfico de la silueta

```
k<- 2
cutg <- cutree(ccom, k=k)
sil <- silhouette(cutg, dist1)
sil.o <- sortSilhouette(sil)

rownames(sil.o) <- row.names(S_BCI)[attr(sil.o, "i0rd")]

plot(sil.o, main= "Gráfico de silueta", cex.names = 0.8,
      col = cutg+3, nmax.lab=100)
```



Al parecer no ha sido el mejor ejemplo, sin embargo, podemos ver que los 2 grupos han sido consistentes. Vamos a probar con nuevos datos.

2.3 Ejercicio 2: Análisis de clasificación

Con el fin de determinar si existen agrupamientos de herbáceas dentro de una parcela permanente de 9ha en la Reserva Ecológica Arenillas realizaremos un análisis de Agrupamiento (Cluster).

Para esto disponemos de una matriz con datos de la composición de la comunidad que puede ser descargado aquí.

Los datos corresponden a un levantamiento de la vegetación de herbáceas en 4 tiempos distintos; final de invierno (abril 2012), estación seca (noviembre 2012), inicio del invierno (diciembre 2012), invierno (enero 2013). Se levantaron 4 cuadrantes de 0.5x0.5 m en cada vértice y centro de la parcela permanente de 9 hectáreas (113 muestras).

Con estos datos:

1. Calcular una matriz de disimilitud utilizando la distancia de Bray-Curtis.
2. Definir la mejor función de enlace para los tres métodos.
3. Definir usando la función silhouette cuantos grupos deberían generarse.
4. Realizar un gráfico del cluster y mostrar los grupos con la función rect.hclust
5. Evaluar si los grupos obtenidos responden a alguna de las variables de especies leñosas
6. Graficar las coordenadas “x” y “y” de las parcelas y colorear cada punto de acuerdo al grupo al que pertenece. Esto nos permitirá identificar si existe un patrón espacial en la generación de los grupos.

Chapter 3

Ordenaciones

En ecología es bastante normal que dispongamos de un conjunto de datos que estan conformados por una serie de sitios o localidades, para los cuales tenemos una serie de variables. Estas variables puede ser cada especie o cada condición que levantemos de este sitio, de esta forma un sitio va a tener tantas variables como especies o factores levantados.

Aunque como vimos un diagrama de dispersión es una buena herramienta para representar las tendencias del conjunto de grupos, existe la limitante de que esto es factible con dos o máximo tres variables (especies), con más de tres variables es imposible poder graficarlo.

De esta forma, el objetivo de los métodos de ordenación es representar los datos a lo largo de un número reducido de ejes ortogonales, contruidos de tal manera que representan, en orden, las principales tendencias de los datos (Borcard et al., 2011).

Las ordenaciones pueden ser indirectas y directas (constreñidas). Las ordenaciones indirectas pueden ser utilizadas para interpretarse visualmente o asociadas a otros métodos como regresión. Mientras que las ordenaciones directas permiten hacer asociaciones con variables explicativas, generar un orden constreñido por otras variables.

En este capítulo nos centraremos en las Ordenaciones indirectas.

3.1 Pasos previos: Transformación y Estandarización de datos

Cuando trabajamos con datos multivariantes cabe la posibilidad de que los datos dentro de esta matriz tengan diferencias de medidas importantes.

Imaginemos que estamos trabajando con la comunidad de herbáceas y existan especies con abundancias máximas de 500, mientras otras especies no alcanzan 10 individuos, o en comunidades de insectos donde ciertas especies puedan tener valores de miles y otros de decenas. Si no transformamos la ordenación estará determinada básicamente por esta especie.

Al transformar los datos evitamos que las especies más comunes dominen en el resultado final de la ordenación y aumentamos la influencia de las especies subordinadas en el modelo resultante.

Existen varias posibilidades para transformar los datos, por lo que definir que función utilizar para transformar los datos es importante. Cada tipo de transformación produce resultados distintos por lo que debemos utilizarlas con precaución.

Las transformaciones más sencilla o menos intensa es la raíz cuadrada, mientras que el logaritmo es la transformación más intensa, podríamos utilizar la raíz cuarta como una función intermedia. La raíz cuadrada la utilizaríamos cuando tenemos diferencias como en el caso de las aves con variaciones de una magnitud

de diferencia (entre decenas y centenas), mientras que la transformación logarítmica la haríamos con la comunidad de insectos donde hay mas de una magnitud de diferencia (entre decenas y miles).

Aunque hay muchos autores que aconsejan realizar transformaciones hay que ser consientes de lo que estamos haciendo, transformaciones muy fuertes en una matriz con pocas diferencias pueden hacer que, por ejemplo, las especies raras tengan igual peso que las dominantes, esto es lo que queremos?.

Veamos un ejemplo:

```
set.seed(4)
aves<- data.frame(sp1= sample(1:90, 10), sp2= sample(100:250, 10))

insectos<- data.frame(sp1= sample(5:99, 10), sp2= sample(1000:2500, 10))

##¿Que pasa cuando transformamos?
cbind(aves, sqrt(aves),log(aves))
```

```
##      sp1 sp2      sp1      sp2      sp1      sp2
## 1    53 213 7.280110 14.59452 3.970292 5.361292
## 2      1 142 1.000000 11.91638 0.000000 4.955827
## 3    26 114 5.099020 10.67708 3.258097 4.736198
## 4    25 241 5.000000 15.52417 3.218876 5.484797
## 5    70 161 8.366600 12.68858 4.248495 5.081404
## 6    23 166 4.795832 12.88410 3.135494 5.111988
## 7    61 240 7.810250 15.49193 4.110874 5.480639
## 8    76 184 8.717798 13.56466 4.330733 5.214936
## 9    78 237 8.831761 15.39480 4.356709 5.468060
## 10   6 208 2.449490 14.42221 1.791759 5.337538
```

```
cbind(insectos, sqrt(insectos),log(insectos))
```

```
##      sp1 sp2      sp1      sp2      sp1      sp2
## 1    72 1851 8.485281 43.02325 4.276666 7.523481
## 2    98 1358 9.899495 36.85105 4.584967 7.213768
## 3    52 2316 7.211103 48.12484 3.951244 7.747597
## 4    50 1980 7.071068 44.49719 3.912023 7.590852
## 5    64 1722 8.000000 41.49699 4.158883 7.451242
## 6    79 2452 8.888194 49.51767 4.369448 7.804659
## 7    47 1687 6.855655 41.07311 3.850148 7.430707
## 8    94 1929 9.695360 43.92038 4.543295 7.564757
## 9    49 1579 7.000000 39.73663 3.891820 7.364547
## 10   96 1009 9.797959 31.76476 4.564348 6.916715
```

La estandarización de los datos permite modificar las variables transformandolas en unidades de desviación típica, lo que nos permite comparar entre valores de distribuciones normales diferentes, o de valores diferentes.

La estandarización o tipificación se lo realiza restando a cada valor el valor medio de la variable y dividiendo para la desviación estandar.

```
avesE <- (aves[,1]-mean(aves[,1]))/sd(aves[,1])
avesE

##      [1]  0.3819546 -1.4073822 -0.5471241 -0.5815345  0.9669301 -0.6503551
##      [7]  0.6572372  1.1733920  1.2422126 -1.2353306

round(mean(avesE),1);sd(avesE)

##      [1] 0
##      [1] 1
```

Table 3.1: Relación entre el tipo de variable y el método de ordenación a utilizar

Medidas.de.Similitud	Tipo.de.Ordenación
Respuesta lineal	PCA
Respuesta Unimodal	CA/DCA
Bray-Curtis	PcoA/mMDS/nmMDS

Como vemos las variables estandarizadas tienen como propiedad que la desviación estandar es 1 y la media es 0.

3.2 Ordenación indirecta o no constreñida

Existen muchos tipos de ordenaciones indirectas como saber que ordenación debo utilizar. Una posibilidad es ver el tipo de respuesta que tenemos, si es una respuesta lineal o monotónica o una respuesta unimodal.

Para determinar que tipo de respuesta usamos la desviación estandar. Si asumimos una distribución normal, datos con menos de tres desviaciones estandar tendrán una respuesta lineal, mientras que si tiene más de tres desviaciones se asumirá una respuesta unimodal. Para cada una de estas respuestas existe un análisis que se puede ejecutar 3.1.

Algunos comentarios sobre los métodos de ordenación.

Principal Component Analysis (PCA)

Fortalezas

Es sencillo de interpretar ya que las distancias entre las muestras son interpretadas directamente como distancias euclidianas.

Apropiado para análisis multivariantes de variables ambientales. El PCA es fuertemente afectado por ceros algo que es bastante común en datos biológicos, al contrario de datos ambientales donde la cantidad de ceros es menor.

Debilidades

Si dos muestras coinciden en celdas con ceros (dobles ceros) incrementa la similitud entre muestras.

La proyección de las distancias euclídeas en un plano puede distorsionar algunas distancias en otros planos.

Principal Coordinates Analysis (PCoA)

Fortalezas

Es una extensión del PCA, permite una definición más amplia de las disimilitudes entre muestras que la distancia Euclídea.

Es capaz de convertir otras medidas de disimilitud (además de la euclídea) en distancias entre muestras.

Debilidades

La proyección de las distancias en un plano también sufre importantes distorsiones como en el caso del PCA.

Correspondence Analysis

Fortalezas

Implícitamente se generan distancias de Chi-cuadrado entre las muestras.

Se basa en un modelo de respuesta unimodal de las especies a los gradientes ambientales subyacentes.

Debilidades

“Efecto arco” Causado por la respuesta unimodal de la abundancia de especies a un gradiente ambiental.

Detrended Correspondence Analysis (DCA, DECORANA)

Es una extensión del CA

Fortalezas Corrige el “Efecto arco” del CA, realizando una ordenación por segmentos y se realinean los segmentos para evitar la curvatura

Debilidades Excesiva manipulación de los datos (Pielou 1984) Hay quien opina que es demasiado parecido a cortar y pegar los datos con tijeras y pegamento.

Multidimensional Scaling (MDS)

Se puede construir una matriz de semejanzas con el coeficiente de semejanza que se considere biológicamente relevante para los datos.

La ordenación de las muestras utiliza el orden relativo de las distancias entre muestras y no los valores absolutos de los coeficientes de similitud. Por eso la representación gráfica del MDS sufre menos distorsiones respecto a las distancias reales.

pretende satisfacer las condiciones impuestas por el “rango” de similitudes. p.ej.: Si la muestra 1 es más parecida a la 2 que a la 3 entonces se localizará más cercana a la muestra 2 que a la 3.

3.3 Pasos para el análisis de Ordenación

1. Decidir que ordenación realizar

Este es el primer paso para poder realizar el análisis de ordenación. Como lo hemos visto podemos usar información para poder determinar que análisis realizar. El tipo de datos que tenemos y por tanto la distancia a utilizar, si tenemos datos con muchos ceros no podremos utilizar una medida de distancia Euclideana y deberíamos trabajar con distancia de Bray-Curtis, por tanto es posible que esto nos defina el trabajar con Escalado Multidimensional (Multidimensional Scaling). Si estamos utilizando otras distancias podríamos definir la cantidad de desviaciones estándar para decidir el tipo de análisis a realizar (3.1).

Para ver la longitud de la gradiente (cantidad de desviaciones estándar) podríamos utilizar la función decorana.

```
library(vegan)
data(dune)

#str(dune) revisa los datos

ord.dca <- decorana(dune)
```

En el primer eje (DCA1) podemos ver la longitud de este eje, el cual está expresado en Unidades de desviación, en este caso 3.7. Esta información ya nos dice que el análisis que debemos hacer será un CA o un DCA.

2. Transformación de los datos

Lo siguiente que debemos decidir es si es necesario transformar. Siempre lo mejor es no transformar los datos, sin embargo, como vimos antes si los datos son muy distintos es necesario realizar esta transformación. Revisa los datos de Dune, como ven no existen diferencias importantes entre los datos por tanto no transformaremos.

Recuerde para transformar los datos definimos la variación entre variables. Variables con más de tres magnitudes de diferencia logaritmo y con dos raíz cuadrada.

Aunque no es necesario transformar, la función `decostand` del paquete `vegan` se puede utilizar para la estandarización y transformar.

```
dsRaiz <- decostand(dune, "standardize", "hellinger")
#Estandarizado y transformado raíz cuadrada (hellinger)
```

```
dslog <- decostand(dune, "standardize", "log")
#Estandarizado y transformado logaritmo (log)

dsSta <- decostand(dune, "standardize")
#Estandarizado
```

3. Desarrollar el análisis

Realizamos el análisis Decorana y Análisis Canónico (CA) y graficamos los datos.

```
ord.dca <- decorana(dune)
ord.dca

##
## Call:
## decorana(veg = dune)
##
## Detrended correspondence analysis with 26 segments.
## Rescaling of axes with 4 iterations.
##
##              DCA1   DCA2   DCA3   DCA4
## Eigenvalues    0.5117 0.3036 0.12125 0.14267
## Decorana values 0.5360 0.2869 0.08136 0.04814
## Axis lengths   3.7004 3.1166 1.30055 1.47888

ord.ca <- cca(dune)
ord.ca

## Call: cca(X = dune)
##
##              Inertia Rank
## Total              2.115
## Unconstrained      2.115   19
## Inertia is mean squared contingency coefficient
##
## Eigenvalues for unconstrained axes:
##   CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
## 0.5360 0.4001 0.2598 0.1760 0.1448 0.1079 0.0925 0.0809
## (Showed only 8 of all 19 unconstrained eigenvalues)
```

Realizamos el análisis de escalado multidimensional (MDS) usando la función `metaMDS`

```
ord.nmnds <- metaMDS(dune, distance="bray", k=2, trymax=50)
ord.nmnds

##
## Call:
## metaMDS(comm = dune, distance = "bray", k = 2, trymax = 50)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      dune
## Distance: bray
##
## Dimensions: 2
## Stress:     0.1183186
## Stress type 1, weak ties
```

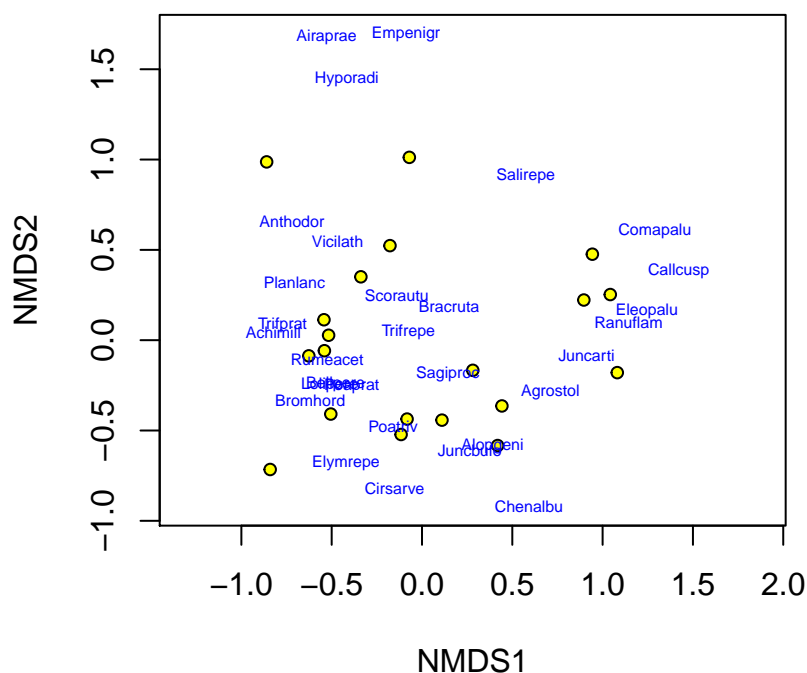


Figure 3.1: Representación gráfica del NMDS

```
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'dune'
```

4. Interpretar los resultados

Para el DCA y CA

Eigenvalue: es la medida de la importancia de un eje, la cantidad de varianza que explica ese eje. En este caso, el primer eje tanto el DCA como el CA explican el 53% de la variación

Axis length: es la longitud del eje principal. Nos da una estima de qué tipo de respuesta tienen nuestros datos.

Para el MDS

El valor de estrés (stress) nos indica la cantidad de varianza que **NO** se ajustó, mientras más bajo es el estrés mayor varianza explica. Algunos autores plantean que bajo 0.2 es aceptable, mientras que valores superiores hay un ajuste pobre. En el caso del ejemplo el ajuste es bueno.

5. Graficar los resultados

Lo siguiente que debemos hacer es graficar los resultados con el fin de ver si existen algunos patrones.

```
plot(ord.nmfs, type="n") # para dibujar un plot vacío
points(ord.nmfs, display="sites", cex=0.8, pch=21, col="black", bg="yellow")
text(ord.nmfs, display="spec", cex=0.5, col="blue")
```

Podemos ver en la figura 3.1 que ciertas especies están asociadas con ciertas localidades esto nos podría servir para explorar los datos.

Chapter 4

Ordenaciones Directas o Constreñidas

Si bien las técnicas de ordenación indirecta nos permite descubrir ciertos patrones, no nos permite testar hipótesis y ver las relaciones de esta matriz con otras variables.

Si disponemos de una matriz de variables explicativas es posible utilizar análisis de ordenación constreñidos. De esta forma, esta matriz representa la información que tenemos sobre cada una de las muestras y podemos usarla para predecir los valores de las variables respuesta (la composición de especies).

Al igual que para los análisis de ordenaciones no constreñidas el tipo de ordenación depende de la respuesta que tenemos en las variables (Tabla 4.1)

Una interesante propiedad de los análisis de ordenación constreñidos es que puedo hacer una ordenación parcial. Esta propiedad me permite evaluar como un grupo de variables pueden influir en mi matriz de respuesta. Podría dividir la información, por ejemplo, en variables ambientales y variables bióticas y ver cuanto explica cada una y cuanto explican en conjunto.

4.1 Realizando una ordenación constreñida

Al igual que en el caso de la ordenación no constreñida debemos decidir el tipo de ordenación constreñida que vamos hacer. En el caso de los datos de *Dune* sabemos que la respuesta es unimodal por lo que escogeremos un análisis canónico de correspondencias (CCA) para nuestra ordenación constreñida.

Para hacer la ordenación constreñida necesitamos una matriz con variables explicativas, utilizaremos las variables provistas en el paquete *vegan* denominadas *env.env*.

```
data("dune.env") #Llamamos a los datos

ord.cca <- cca(dune~ A1 + Use, data=dune.env)
ord.cca

## Call: cca(formula = dune ~ A1 + Use, data = dune.env)
##
```

Table 4.1: Relación entre el tipo de variable y el método de ordenación a utilizar

Medidas.de.Similitud	Tipo.de.Ordenación	Tipo.Ordenación.Constreñida
Respuesta lineal	PCA	RDA (Redundancy Analysis)
Respuesta Unimodal	CA/DCA	CCA (Canonical Correspondence Analysis)
Bray-Curtis	PcoA/mMDS/nmMDS	PERMANOVA

```
##              Inertia Proportion Rank
## Total          2.1153      1.0000
## Constrained    0.4724      0.2233   3
## Unconstrained  1.6429      0.7767  16
## Inertia is mean squared contingency coefficient
##
## Eigenvalues for constrained axes:
##   CCA1   CCA2   CCA3
## 0.27630 0.14929 0.04683
##
## Eigenvalues for unconstrained axes:
##   CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8   CA9   CA10
## 0.3792 0.3091 0.2093 0.1629 0.1308 0.0965 0.0758 0.0731 0.0483 0.0456
##   CA11  CA12  CA13  CA14  CA15  CA16
## 0.0431 0.0237 0.0163 0.0141 0.0108 0.0042
```

Lo que podemos ver es que la variable A1 más Use explican el 22.33% de la variación en los datos.

La decisión de que modelos deberían generar debe responder a una lógica ecológica, así podemos probar como algunas variables juegan o no un rol en la estructura de la comunidad.

Una herramienta que podríamos utilizar para analizar la importancia de cada variable es utilizar la función `envfit`, esta función permite relacionar la ordenación no constreñida con las variables explicativas y mediante un test de permutación mostrarnos que variables se asocian significativamente con la ordenación.

```
fitVar <- envfit(ord.ca, dune.env)
fitVar

##
## ***VECTORS
##
##           CA1           CA2      r2 Pr(>r)
## A1 0.998160 0.060614 0.3104  0.055 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
##
## ***FACTORS:
##
## Centroids:
##           CA1           CA2
## Moisture1  -0.7484 -0.1423
## Moisture2  -0.4652 -0.2156
## Moisture4   0.1827 -0.7315
## Moisture5   1.1143  0.5708
## ManagementBF -0.7258 -0.1413
## ManagementHF -0.3867 -0.2960
## ManagementNM  0.6517  1.4405
## ManagementSF  0.3376 -0.6761
## UseHayfield  -0.2861  0.6488
## UseHaypastu  -0.0735 -0.5602
## UsePasture   0.5163  0.0508
## Manure0      0.6517  1.4405
## Manure1     -0.4639 -0.1738
## Manure2     -0.5872 -0.3600
## Manure3      0.5187 -0.3172
```

```
## Manure4      -0.2059 -0.8775
##
## Goodness of fit:
##              r2 Pr(>r)
## Moisture    0.4113 0.005 **
## Management  0.4441 0.001 ***
## Use         0.1845 0.087 .
## Manure      0.4552 0.010 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

Podemos utilizar el *Goodness of fit* para ver cuales son las variables que ajustan la ordenación y utilizar estas para hacer el modelo constreñido. En este caso utilizaremos Manure y Management.

```
ord.ccafit <- cca(dune~Manure+Management, data=dune.env)
ord.ccafit
```

```
## Call: cca(formula = dune ~ Manure + Management, data = dune.env)
##
##              Inertia Proportion Rank
## Total          2.1153      1.0000
## Constrained    0.8766      0.4144    6
## Unconstrained  1.2386      0.5856   13
## Inertia is mean squared contingency coefficient
## Some constraints were aliased because they were collinear (redundant)
##
## Eigenvalues for constrained axes:
##   CCA1  CCA2  CCA3  CCA4  CCA5  CCA6
## 0.3617 0.2271 0.1454 0.0655 0.0418 0.0353
##
## Eigenvalues for unconstrained axes:
##   CA1  CA2  CA3  CA4  CA5  CA6  CA7  CA8  CA9  CA10
## 0.4082 0.1592 0.1493 0.1252 0.0962 0.0774 0.0649 0.0424 0.0382 0.0312
##   CA11 CA12 CA13
## 0.0251 0.0121 0.0090
```

Como vemos con este procedimiento subimos al 41% de la varianza explicada.

Podemos utilizar la función `anova` para evaluar la significancia de cada variable dentro del modelo, de forma separada.

```
ord.ccaT <- cca(dune~ ., data=dune.env)
anova(ord.ccaT, by="term", permu=1000)
```

```
## Permutation test for cca under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## Model: cca(formula = dune ~ A1 + Moisture + Management + Use + Manure, data = dune.env)
##              Df ChiSquare      F Pr(>F)
## A1              1   0.22476 2.5704 0.012 *
## Moisture        3   0.51898 1.9783 0.012 *
## Management      3   0.39543 1.5074 0.047 *
## Use             2   0.10910 0.6238 0.906
```



```
short <- make.cepnames(names(BCI[1:5]))  
short
```

```
## [1] "Abarmacr" "Vachmela" "Acaldive" "Acalmacr" "Adeltril"
```

Existen procedimientos para construir modelos que ahora no tocaremos, puede encontrar más información en Oksanen 2015

Nota: para realizar un permanova debemos utilizar la función `adonis`. El procedimiento es similar al desarrollo del `cca`.

4.2 Ejercicio 3: Análisis de Ordenación

Con los datos utilizados para realizar el análisis de aglomerados, vamos a realizar un análisis de ordenación constreñida.

- Defina que tipo de ordenación constreñida debe realizar para explicar la variación de los datos de herbáceas.
- Realice un análisis para definir las variables que se debería utilizar en el análisis
- Ajuste un modelo y defina el porcentaje de variación explicado.
- Compara los resultados de la ordenación directa si en vez de transformar los datos corremos los modelos con los datos brutos.
- Realice un gráfico del modelo desarrollado.

Bibliography

Borcard, D., Gillet, F., and Legendre, P. (2011). *Numerical Ecology with R*.

Stevens, M. H. H. (2009). *A Primer of Ecology with R*.