
Maestría en Computo Estadístico
Estadística Multivariada
Tarea 4

14 de mayo de 2021

Enrique Santibáñez Cortés

Repositorio de Git: Tarea 4, EM.

Ejercicio 1.

Frecuentemente en las aplicaciones nos encontramos con una variable categórica nominal con k estados excluyentes medida sobre una muestra de $n = n_1 + \dots + n_g$ individuos provenientes de g poblaciones. Se desea obtener una medida de disimilaridad entre estas poblaciones. En estas condiciones, el vector de frecuencias de cada población $n_i = (n_{i1}, \dots, n_{ik})$, para $i = 1, \dots, g$, tiene una distribución conjunta multinomial con parámetros (n_i, p_i) , donde $n_i = n_{i1} + \dots + n_{ik}$ y $p_i = (p_{i1}, \dots, p_{ik})$. Una medida de disimilaridad es la distancia de Bhattacharyya, conocida en genética como distancia Cavalli-Sforza, cuya expresión es:

$$d_{ij}^2 = \arccos \left(\sum_{l=1}^k \sqrt{p_{il}p_{jl}} \right) \quad (1)$$

La siguiente tabla contiene las proporciones génicas (observadas) de los grupos sanguíneos correspondientes a 10 poblaciones.

	Población	Grupo A	Grupo AB	Grupo B	Grupo O
1	Francesa	0.21	0.06	0.06	0.67
2	Checa	0.25	0.04	0.14	0.57
3	Germanica	0.22	0.06	0.08	0.64
5	China	0.18	0.00	0.15	0.67
6	Ainu	0.23	0.00	0.28	0.49
7	Esquimal	0.30	0.00	0.06	0.64
8	Afromericana USA	0.10	0.06	0.13	0.71
9	Española	0.27	0.04	0.06	0.63
10	Egipcia	0.21	0.05	0.20	0.54

a) Obtenga las distancias de Bhattacharyya entre estas poblaciones.

RESPUESTA:

Primero introducimos la matriz de proporciones genéticas observadas,

```
nombres <- c("Francesa", "Checa", "Germanica", "Vasca", "China", "Ainu", "Esquimal",  
             "Afromericana USA", "Española", "Egipcia")  
datos_genetica <- c(0.21, 0.06, 0.06, 0.67, 0.25, 0.04, 0.14, 0.57, 0.22, 0.06, 0.08,  
                   0.64, 0.19, 0.04, 0.02, 0.75, 0.18, 0.00, 0.15, 0.67, 0.23, 0.00,  
                   0.28, 0.49, 0.30, 0.00, 0.06, 0.64, 0.10, 0.06, 0.13, 0.71, 0.27,  
                   0.04, 0.06, 0.63, 0.21, 0.05, 0.20, 0.54) # datos  
p_geneticas <- matrix(datos_genetica, nrow=10, byrow=T) # creamos la matriz de datos
```

Ahora creamos una matriz auxiliar con elementos $(i, j) = \sum_{l=1}^k \sqrt{p_{il}p_{jl}}$, esto con el objetivo de facilitar los cálculos

```
p_p_geneticas <- sqrt(p_geneticas)%*% sqrt(t(p_geneticas)) # matriz auxiliar.
```

Por último, calculamos el arcoseno de cada uno de los elementos de la matriz auxiliar.

```
dist_cavalli <- acos(p_p_geneticas)
dist_cavalli[is.na(dist_cavalli)] <- 0 # imputamos con 0
```

Y por lo tanto, la matriz de distancias de Bhattacharyya es

pob.	1	2	3	4	5	6	7	8	9	10
1	0	0.16	0.04	0.12	0.29	0.4	0.26	0.18	0.08	0.22
2	0.16	0	0.12	0.27	0.22	0.26	0.25	0.21	0.14	0.09
3	0.04	0.12	0	0.17	0.27	0.36	0.26	0.18	0.08	0.18
4	0.12	0.27	0.17	0	0.32	0.47	0.26	0.26	0.15	0.34
5	0.29	0.22	0.27	0.32	0	0.19	0.19	0.27	0.27	0.25
6	0.4	0.26	0.36	0.47	0.19	0	0.31	0.37	0.36	0.24
7	0.26	0.25	0.26	0.26	0.19	0.31	0	0.36	0.2	0.32
8	0.18	0.21	0.18	0.26	0.27	0.37	0.36	0	0.24	0.2
9	0.08	0.14	0.08	0.15	0.27	0.36	0.2	0.24	0	0.22
10	0.22	0.09	0.18	0.34	0.25	0.24	0.32	0.2	0.22	0

Cuadro 1: Matriz de distancias de Bhattacharyya

- b) Construye una configuración MDS de las poblaciones mediante la solución clásica (coordenadas principales), utilizando la matriz de distancias Bhattacharyya.

RESPUESTA:

Ocupando la función *cmdscale* de la librería *stats* procedemos a construir una configuración MDS considerando $k=6$, se escogió este valor de forma arbitrario ya que en el inciso posterior se usa un criterio para seleccionar ese valor. Algo importante resaltar, es que la distancia Battacharry (1) están elevadas al cuadrado por lo que tenemos de utilizar la raíz cuadrada de la matriz calculada anteriormente.

```
# calculo de la configuración mds
resul_mds_clas <- cmdscale(sqrt(dist_cavalli), k=6, eig=TRUE, add=FALSE, x.ret=FALSE)
config_pobla <- resul_mds_clas$points # configuracion mds
```

Entonces la configuración MDS es

pobl.	X_1	X_2	X_3	X_4	X_5	X_6
1	0.21	0.01	-0.02	-0.06	-0.07	-0.02
2	-0.02	0.13	-0.14	0.07	0.02	0.05
3	0.17	0.06	-0.06	-0.03	-0.09	-0.03
4	0.24	-0.15	0.1	-0.11	0.14	0.1
5	-0.23	-0.12	0.12	0.06	-0.13	0.14
6	-0.38	0.02	-0.01	-0.2	0.02	-0.07
7	-0.09	-0.3	-0.05	0.14	0.07	-0.1
8	0.06	0.18	0.26	0.07	0.01	-0.1
9	0.15	-0.06	-0.12	-0.02	-0.06	-0.03
10	-0.1	0.23	-0.08	0.07	0.1	0.06

Cuadro 2: Configuración MDS, $k=6$

- (c) ¿Cuál la dimensión adecuada de la representación euclidiana?, ¿cuál es el porcentaje de la variabilidad explicada por las dos primeras coordenadas principales? Grafica las poblaciones con las dos primeras coordenadas.

RESPUESTA:

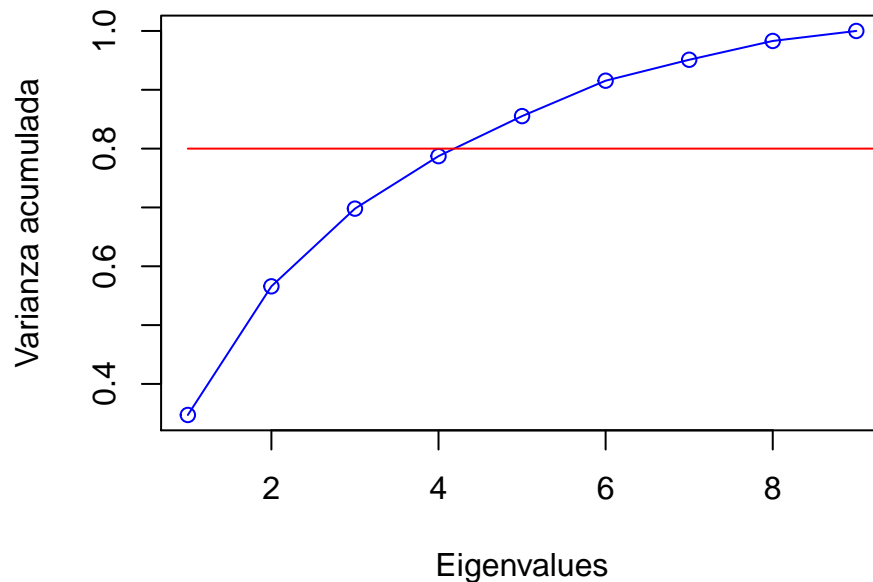
Para saber la dimensión adecuada de la representación utilizaremos el criterio de la proporción de varianza explicada por las primeras m dimensiones, dado por

$$\frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^n |\lambda_i|}.$$

Entonces calculemos la varianza explicada por cada componente,

```
var_acu <- c()
for (i in seq(1:9)){
  var_acu[i] <- cmdscale(sqrt(dist_cavalli), k=i, eig=TRUE, add=FALSE,
    x.ret=FALSE)$GOF[1]
}

plot(seq(1:9), var_acu, type="o", col="blue", xlab="Eigenvalues",
  ylab="Varianza acumulada")
lines(x=seq(1:10), y=rep(0.80,10), type="l", col="red")
```

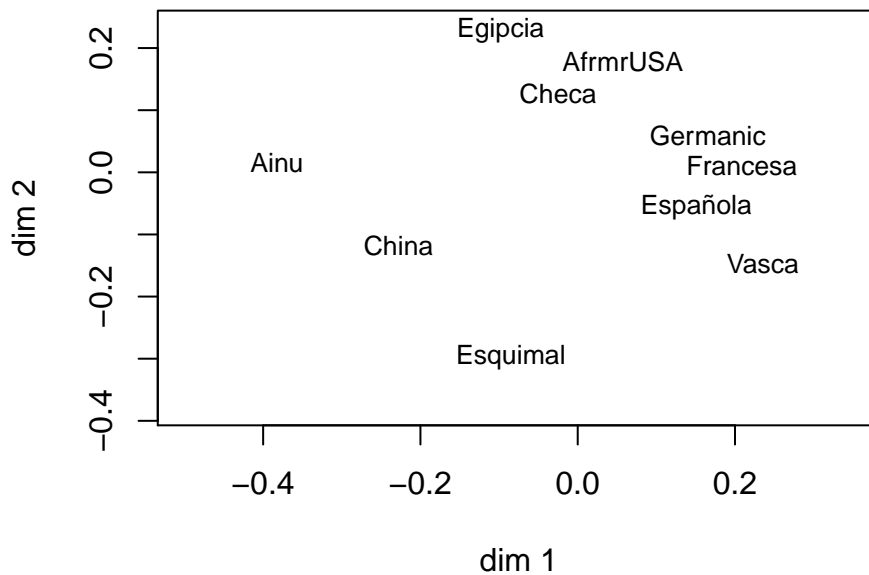


Observando el gráfico anterior, la dimensión adecuada de la representación euclidiana es 4 debido a que la varianza explicada está alrededor del 80 %. Ahora, graficamos las poblaciones con las dos primeras coordenadas, que tiene una varianza explicada igual a 0.5658885.

```
resul_mds_clas<-cmdscale(sqrt(dist_cavalli), k=4, eig=TRUE, add=FALSE, x.ret=FALSE)
config_pobla<-resul_mds_clas$points
dimnames(config_pobla)[[1]]<-nombres

plot(config_pobla[,1], config_pobla[,2], ylim=range(config_pobla[,1]), xlab="dim 1",
  main="Configuracion solucion mediante MDS, poblacion", ylab="dim 2", type="n",
  lwd=2, xlim=range(-0.5,0.35))
text(config_pobla[,1], config_pobla[,2],
  labels=abbreviate(row.names(config_pobla),minlength=8),cex=0.8,lwd=4)
```

Configuración solución mediante MDS, población



En la gráfica anterior se pueden reconocer una separación de los regiones europea: Francesa, Española, Vasca y Germanic. Además de las regiones Africanas: Checa, Egipcia y ArrmrUSA. Y otra región asiática: Ainu y China. Lo cual tiene sentido que se separaran así.

- d) Construye una configuración MDS de las poblaciones utilizando el enfoque de mínimos cuadrados considerando la matriz de distancias Bhattacharyy, tomando como solución inicial la solución clásica y considerando las transformaciones de tipo razón, intervalo y ordinal para las disimilaridades. Compara los resultados obtenidos en cada modelo y justifica la dimensionalidad adecuada de representación y gráfica las dos primeras dimensiones.

RESPUESTA:

Propiedad: 1 Elección de la dimensionalidad adecuada de la representación Un criterio muy utilizado es calcular soluciones MDS para un rango de dimensiones y graficar el STRESS vs la dimensión. Se utiliza el criterio del codo de la curva.

Definición: 1 Stress1 La función STRESS1 evita el problema de la dependencia de escalas de las disimilaridades o de sus transformaciones, es decir, al dividir entre las distancias euclidianas se eliminan las escalas de las disimilaridades o de las disparidades.

$$STRESS1 = \sqrt{\frac{\sum_{i,j} w_{ij}(\hat{d}_{ij} - d_{ij}(\mathbf{X}))^2}{\sum_{i,j} w_{ij}d_{ij}^2(\mathbf{X})}}$$

Ocuparemos la función `smacofSym` de la librería `smacof` para encontrar estas configuraciones utilizando el enfoque de mínimos cuadrados. Empecemos graficando las gráficas de codo,

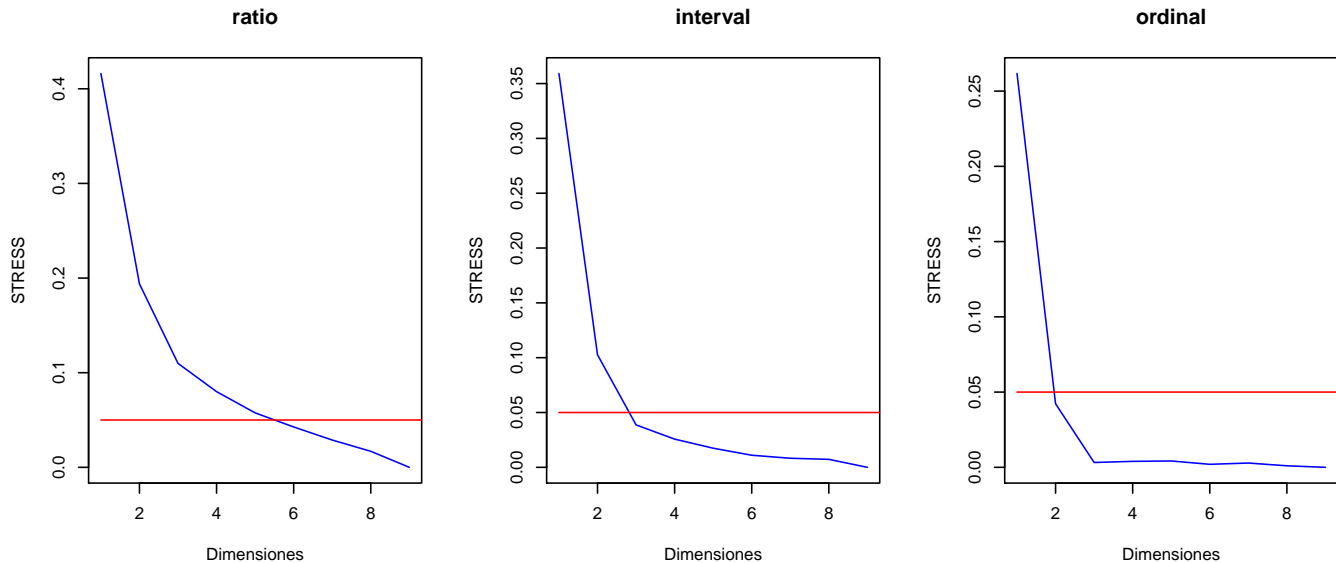
```
library("smacof")
stress_values <- c()
```

```

par(mfrow=c(1,3))
for (j in c("ratio", "interval", "ordinal")){
  for (i in seq(1:9)){
    stress_values[i] <- smacofSym(sqrt(dist_cavalli), ndim=i, type=j, init = "torgerson",
                                itmax = 1000, eps = 1e-06)$stress
  }

  plot(seq(1:9), stress_values, type="l", col="blue", main=j, xlab="Dimensiones", ylab="STRESS")
  lines(x=seq(1:10), y=rep(0.05,10), type="l", col="red")
}

```



Algunos investigadores sugieren que si el valor del STRESS1 para una dimensionalidad dada es menor de 0.05 la representación es muy buena y por tanto se puede elegir esa dimensionalidad M para la configuración solución \mathbf{X} . Entonces con este criterio observamos que cuando se utilizan la transformación de tipo de razón la mejor dimensión serían 6, considerando la transformación intervalo sería 3 y ocupando la tranformación ordinal sería 2. Por lo que, la mejor transformación sería ocupar la ordinal ya que se necesitan menos dimensiones.

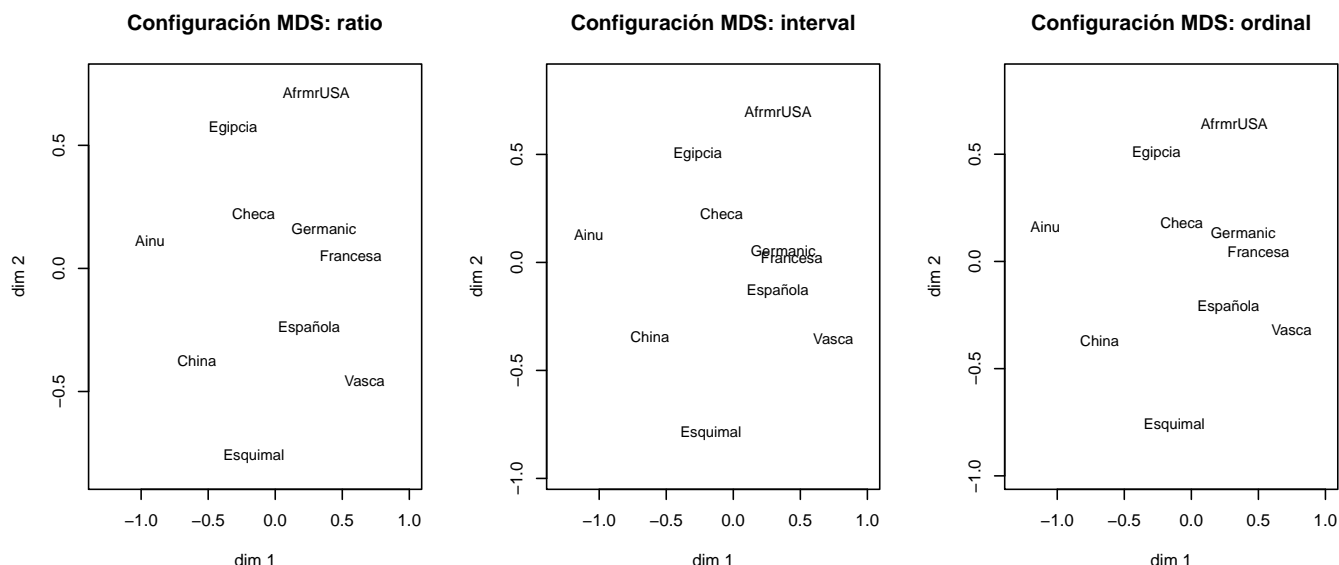
Ahora, proyectamos las dos primeras dimensiones

```

par(mfrow=c(1,3))
for (i in c("ratio", "interval", "ordinal")){
  config_pobla_r <- smacofSym(sqrt(dist_cavalli), ndim=2, type=i, init = "torgerson",
                                itmax = 1000, eps = 1e-06)$conf

  plot(config_pobla_r[,1], config_pobla_r[,2], ylim=range(config_pobla_r[,1]+.1),
        main=paste0("Configuración MDS: ",i), xlab="dim 1",ylab="dim 2",type="n",lwd=2,
        xlim=range(-1.3,1))
  text(config_pobla_r[,1], config_pobla_r[,2],
        labels=abbreviate(nombres,minlength=8),cex=0.9,lwd=2)
}

```



Observamos que no existe mucha diferencias en las proyecciones obtenidas, en general se observa los mismos patrones. Solo que cambia la separacione entre las poblaciones.

- e) Compara las configuraciones MDS obtenidas con el enfoque clásico y de mínimos cuadrados, ¿existen diferencias? ¿Cuáles son las conclusiones?

RESPUESTA:

En la configuración de MDS con el enfoque clásico ocupamos 4 dimensiones para tener una buena representación, en cambio ocupando mínimos cuadrados con la transformación ordinal se necesitan 2 dimensiones para tener una adecuada representación. Entonces comparemos las gráficasa utilizando las dos primeras dimensiones,

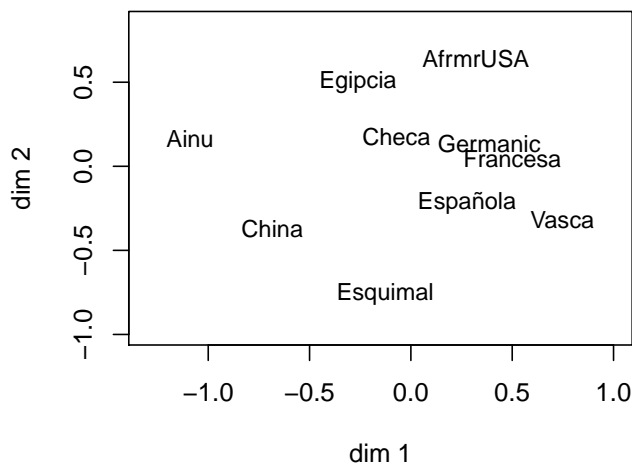
```
par(mfrow=c(1,2))
config_pobla_r <- smacofSym(sqrt(dist_cavalli), ndim=2, type=i, init = "torgerson",
                             itmax = 1000, eps = 1e-06)$conf

plot(config_pobla_r[,1], config_pobla_r[,2], ylim=range(config_pobla_r[,1]+.1),
      main=paste0("Configuración MDS usando MC: ",i), xlab="dim 1",ylab="dim 2",type="n",lwd=2,
      xlim=range(-1.3,1))
text(config_pobla_r[,1], config_pobla_r[,2],
      labels=abbreviate(nombres,minlength=8),cex=0.9,lwd=2)

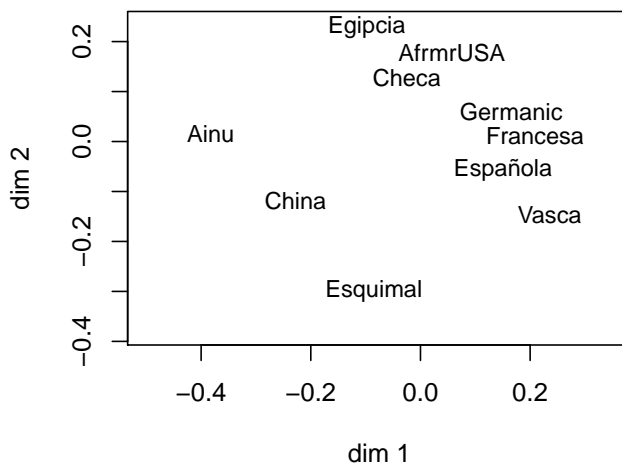
resul_mds_clas<-cmdscale(sqrt(dist_cavalli), k=4, eig=TRUE, add=FALSE, x.ret=FALSE)
config_pobla<-resul_mds_clas$points
dimnames(config_pobla)[[1]]<-nombres

plot(config_pobla[,1], config_pobla[,2], ylim=range(config_pobla[,1]), xlab="dim 1",
      main="Configuracion MDS enfoque clasico", ylab="dim 2", type="n",
      lwd=2, xlim=range(-0.5,0.35))
text(config_pobla[,1], config_pobla[,2],
      labels=abbreviate(row.names(config_pobla),minlength=8),cex=0.9,lwd=4)
```

Configuración MDS usando MC: ordinal



Configuración MDS enfoque clasico



Desde mi perspectiva, la representación obtenida usando el enfoque clásico tiene mejor interpretación. Ya que se observan *tres* grupos claros, un grupo es muy parecido en los dos enfoques. Pero los otros dos restantes se aprecian mejor con los enfoques clasico. Mi conclusión sería usar para este ejercicio usar MDS con el enfoque clásico. ■.

Ejercicio 2.

En muchas situaciones las variables que se observan sobre un conjunto de individuos son de naturaleza binaria. En estos casos para poder disponer de una matriz de distancias entre individuos se utilizan coeficientes de similitud. El coeficiente de similitud entre el individuo i y el individuo j , s_{ij} , se calcula a partir de las frecuencias :

- a ="numero de variables con respuesta 1 en ambos individuos"
- b ="numero de variables con respuesta 0 en el primer individuo y con respuesta 1 en el segundo individuo"
- c ="numero de variables con respuesta 1 en el primer individuo y con respuesta 0 en el segundo individuo"
- d ="numero de variables con respuesta 0 en ambos individuos"

Existen muchos coeficientes de similitud, pero los de Sokal-Michener y de Jacard son especialmente interesantes porque dan lugar a una configuración euclidiana. Se definen como

$$\text{sokal - Michener : } s_{ik} = \frac{a + d}{p}, \quad \text{Jacard : } s_{ik} = \frac{a}{a + b + c}$$

donde p es el número de variables observadas. Aplicando uno de estos coeficientes a un conjunto de n individuos se obtiene una matriz de similitudes $S = \{s_{ij}\}_{n \times n}$. Utilizando la siguiente transformación podemos convertir la matriz de similitudes a una matriz de distancias

$$D^2 = 2(1_n 1_n' - S)$$

Se considera el siguiente conjunto de 6 individuos formado por 5 animales, león, jirafa, vaca, oveja, gato doméstico, junto con el hombre. Se miden 6 variables binarias sobre estos individuos: X_1 = tiene cola, X_2 = es salvaje, X_3 = tiene el cuello largo, X_4 = es animal de granja, X_5 = es carnívoro y X_6 = camina sobre 4 patas.

a) Obtenga la matriz de datos.

RESPUESTA:

Describiremos a los animales según las variables a estudiar. Los leones tienen cola, son salvajes, carnívoros y caminan en 4 patas. Las jirafas tienen cola, son salvajes, tienen cuello largo y caminan en 4 patas. Las vacas y las ovejas tienen cola, son animales de granjas y caminan en cuatro patas. Los gatos domésticos son carnívoros y caminan en cuatro patas. Y por último, el hombre es carnívoro, este último no estoy muy seguro ya que siendo más estricto el hombre sería omnívoro pero se me hizo raro un individuo con todas las variables igual a 0. **Lo anterior se resume en el siguiente Cuadro, que representa la matriz de datos:**

Animal	X_1 (cola)	X_2 (salvaje)	X_3 (cuello largo)	X_4 (granja)	X_5 (carnívoro)	X_6 (4 patas)
León	1	1	0	0	1	1
Jirafa	1	1	1	0	0	1
Vaca	1	0	0	1	0	1
Oveja	1	0	0	1	0	1
Gato domes.	1	0	0	0	1	1
Hombre	0	0	0	0	1	0

Cuadro 3: Matriz de datos de los animales.

Ingresamos esta matriz de datos a R, para futuros cálculos

```
# matriz de datos:
animales <- matrix(c(1,1,1,1,1,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,1,0,0,0,1,1,1,1,1,
1,1,0), nrow=6)
```

b) Calcule los coeficientes de similitud de Sokal-Michener y de Jacard para cada par de individuos y obtenga las matrices de distancias asociadas.

RESPUESTA:

Calculamos las frecuencias a, b, c y d. Debido de como están definidas podemos aprovechar las propiedades de las matrices y calcularlas de la siguiente forma en R:

```
# si P1 es el valor de verdad del primero, y P2 es el valor de valor del segundo
a <- animales %*% t(animales) # P1 & P2
b <- (animales==0)%*%t(animales) # -P1 & P2
c <- animales %*% t(animales==0) # P1 & -P2
d <- (animales==0)%*%t(animales==0) # -P1 & -P2
```

Ahora, procedemos a calcular los coeficientes de similitud de Sokal-Michener y Jacard

```
sokal <- (a+d)/6 # similitud Sokal-Michener
jacard <- a/(a+b+c) # similitud de Jacard
```

Por lo que, las matrices de los dos coeficientes de similitud son:

Animal	León	Jirafa	Vaca	Oveja	Gato	Hombre
León	1	0.67	0.5	0.5	0.83	0.5
Jirafa	0.67	1	0.5	0.5	0.5	0.17
Vaca	0.5	0.5	1	1	0.67	0.33
Oveja	0.5	0.5	1	1	0.67	0.33
Gato	0.83	0.5	0.67	0.67	1	0.67
Hombre	0.5	0.17	0.33	0.33	0.67	1

Cuadro 4: Coeficientes de similitud de Sokal–Michener (redondeado).

Animal	León	Jirafa	Vaca	Oveja	Gato	Hombre
León	1	0.6	0.4	0.4	0.75	0.25
Jirafa	0.6	1	0.4	0.4	0.4	0
Vaca	0.4	0.4	1	1	0.5	0
Oveja	0.4	0.4	1	1	0.5	0
Gato	0.75	0.4	0.5	0.5	1	0.33
Hombre	0.25	0	0	0	0.33	1

Cuadro 5: Coeficientes de similitud de Jacard (redondeado).

Observamos que tienen grandes diferencias los coeficientes. Y ahora, calculamos las matrices de distancias asociadas a estos coeficientes.

```
D2_sokal <- 2*(matrix(c(1),nrow=6, ncol=6)-sokal) # distancias
D2_jacard <- 2*(matrix(c(1),nrow=6, ncol=6)-jacard)
```

Por lo que, la matriz de distancias a estos dos coeficientes son:"

Animal	León	Jirafa	Vaca	Oveja	Gato	Hombre
León	0	0.67	1	1	0.33	1
Jirafa	0.67	0	1	1	1	1.67
Vaca	1	1	0	0	0.67	1.33
Oveja	1	1	0	0	0.67	1.33
Gato	0.33	1	0.67	0.67	0	0.67
Hombre	1	1.67	1.33	1.33	0.67	0

Cuadro 6: Distancias asociadas a los coeficientes de Sokal-Michener (redondeado)

Animal	León	Jirafa	Vaca	Oveja	Gato	Hombre
León	0	0.8	1.2	1.2	0.5	1.5
Jirafa	0.8	0	1.2	1.2	1.2	2
Vaca	1.2	1.2	0	0	1	2
Oveja	1.2	1.2	0	0	1	2
Gato	0.5	1.2	1	1	0	1.33
Hombre	1.5	2	2	2	1.33	0

Cuadro 7: Distancias asociadas a los coeficientes de Jacard (redondeado)

Ejercicio 3.

Sea O un conjunto de n individuos cuya matriz de distancias euclidianas es D y cuya representación en coordenadas principales es \mathbf{X} . Se desean obtener las coordenadas de un nuevo individuo al que llamaremos individuo $n+1$, del cual se conocen los cuadrados de sus distancias a los n individuos del conjunto O . Si $\mathbf{d} = (\delta_{n+1,1}^2, \dots, \delta_{n+1,n}^2)'$ es el vector columna que contiene las distancias al cuadrado del individuo $n+1$ a los restantes n individuos, se puede probar que las coordenadas principales del individuo $n+1$ están dadas por

$$x_{n+1} = \frac{1}{2} \Delta^{-1} \mathbf{X}'(b - d),$$

donde $\mathbf{b} = \text{diag}(\mathbf{B}) = (\mathbf{b}_{11}, \dots, \mathbf{b}_{nn})'$, $\mathbf{B} = \mathbf{X}\mathbf{X}' = \mathbf{U}\mathbf{\Delta}\mathbf{U}'$ y \mathbf{U} es una matriz ortogonal. La ecuación anterior se conoce como fórmula de interpolación de Gower.

Para los datos del ejercicio 2.

- a) Obtenga una representación en coordenadas principales utilizando la matriz de distancias calculada a partir del coeficiente de similitud de Sokal-Michener.

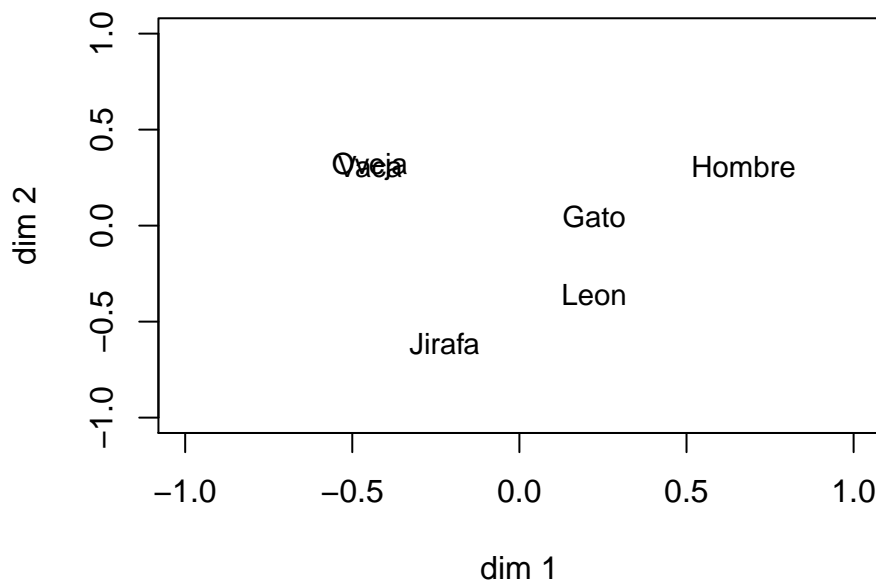
RESPUESTA:

Primero encontremos la representación en coordenadas principales utilizando la matriz de distancias a partir del coeficiente de similitud de Sokal-Michener, para ello consideraremos $k=2$ esto con el objetivo de poder representar visualmente los resultados.

```
nombres <- c("Leon", "Jirafa", "Vaca", "Oveja", "Gato", "Hombre")
resul_mds_clas <- cmdscale(sqrt(D2_sokal), k = 2, eig = TRUE, add = FALSE, x.ret = FALSE)
config_animals <- resul_mds_clas$points
dimnames(config_animals)[[1]] <- nombres

plot(config_animals[,1], config_animals[,2], xlim=range(-1,1),
      main="Configuracion mediante MDS clasico de los Animales",
      ylim=range(-1,1), xlab="dim 1", ylab="dim 2", type="n", lwd=2)
text(config_animals[,1], config_animals[,2],
      labels=abbreviate(row.names(config_animals), minlength=8), cex=0.9, lwd=2)
```

Configuracion mediante MDS clasico de los Animales



Podemos observar que la representación anterior tiene sentido, debido a que los animales de granjas y salvajes se pueden separar muy bien. Además de que el hombre se separa de todos a excepción del león y gato. Además de que la vaca y la oveja están en el mismo lugar.

- b) Sin volver a recalcular las coordenadas principales, añada el elefante al conjunto de animales y obtenga sus coordenadas principales.

RESPUESTA:

Propiedad: 2 Como caso especial, para cada matriz simétrica real $n \times n$, los valores propios son reales y los vectores propios pueden elegirse reales y ortonormales. Por lo tanto, una matriz simétrica real A se puede descomponer como

$$A = U\Lambda U^T$$

donde U es una matriz ortogonal cuyas columnas son los vectores propios de A y Λ es una matriz diagonal cuyas entradas son los valores propios de A . (Horn y Johnson (1985), p. 136, Corolario 2.5.11)

Ocupando la descripción de este problema y la Propiedad 2, primero calculamos la matriz \mathbf{B} , \mathbf{b} y la descomposición espectral de la matriz \mathbf{B}

```
# calculamos la matriz B
B <- config_animals%*%t(config_animals)

# calculamos el vector b
b <- diag(B)

# calculamos la descomposición espectral.
r <- eigen(B)
U <- r$vectors
lam <- diag(r$values)
```

Ahora, calculamos las frecuencias a , b , c y d definidas en el inciso anterior para el elefante:

```
a_elefante <- matrix(c(3,3,2,2,2,0), nrow=1)
b_elefante <- matrix(c(1,1,1,1,1,1), nrow=1)
c_elefante <- matrix(c(0,0,1,1,1,3), nrow=1)
d_elefante <- matrix(c(2,2,2,2,2,2), nrow=1)
```

Calculamos las distancias al cuadrado del elefante usando el coeficiente de Sokal-Michener, es decir, el vector \mathbf{d}

```
sokal_elefante <- (a_elefante+d_elefante)/6
D2_elefante <- 2*(matrix(c(1), ncol=6)-sokal_elefante)
D2_elefante
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 0.3333333 0.3333333 0.6666667 0.6666667 0.6666667 1.333333
```

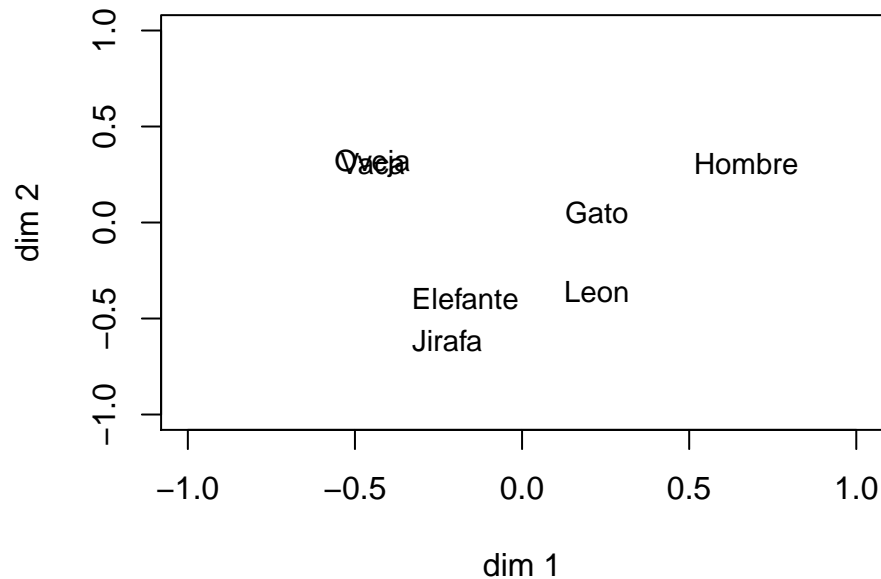
Con todo lo anterior y ocupando la descripción del ejercicio podemos calcular las coordenadas principales del nuevo individuo, el elefante,

```
x_elefante <- (solve(lam[0:2,0:2]))%*%(t(config_animals))%*%t(b-sqrt(D2_elefante)))
```

Por último, graficamos nuevamente las representación incluyendo al elefante

```
config_animals_new <- matrix(c(config_animals[,1], x_elefante[1], config_animals[,2], x_elefante[2]),  
  nrow=nrow(config_animals),  
  dimnames=list(dimnames(config_animals)[[1]], c(nombres, "Elefante"))  
  
plot(config_animals_new[,1],config_animals_new[,2],main="Configuracion mediante MDS clasico",  
  ylim=range(-1, 1), xlim=range(-1,1), xlab="dim 1",ylab="dim 2",type="n",lwd=2)  
text(config_animals_new[,1],config_animals_new[,2],  
  labels=abbreviate(row.names(config_animals_new),minlength=8),cex=0.9,lwd=2)
```

Configuracion mediante MDS clasico



Observando la gráfica anterior, podemos concluir que es una representación muy buena. Considero que el componente uno se puede interpretar como la preferencia en comida: carnívoro o no carnívoro, y el segundo como si es salvaje o no. Por lo que el elefante está muy cercano a la jirafa ya que ambos son salvajes y no son carnívoros. ■.