

Problema Multicriterio

Elección del mejor plan de ocio

Claudia Trias de Tena

Índice

Enunciado	2
METODO AHP:	5
Funciones de clase:	5
Estudio de la inconsistencia	8
Librería AHP:	9
METODO PROMETHEE:	12
Promethee I:	12
Promethee II:	13
Resolución con Promethee Windows:	14
METODO ELECTRE:	16
Tablas de Concordancia, Discordancia y Superación:	19
METODO AXIOMATICO DE ARROW Y RAYMOND	22
Conclusión Final	24

Enunciado

Se desea elegir el mejor plan de ocio para realizar en un fin de semana. Se proponen las siguientes **alternativas**:

- (1) Viaje a Madrid: Pasar el fin de semana en la capital con amigos, que incluirá planes culturales y gastronómicos.
- (2) Centro de Sevilla: Pasar el domingo en el centro con amigos, con un paseo por la ciudad y una ruta de tapas por los bares más auténticos.
- (3) Playa de Cádiz: Pasar el sábado en alguna playa de Cádiz.
- (4) Ruta de senderismo por Sierra Morena.

Para valorar las posibles alternativas se han definido los siguientes **criterios**:

- (1) Coste:
 - Transporte: Los viajes a los distintos destinos se harán en coche, luego debemos considerar el gasto de la gasolina.
 - Comida: Para la playa y la escapada a la sierra se plantea una comida tipo picnic, mientras que en Madrid y el centro de Sevilla se comerá en bares/restaurantes.
 - Alojamiento: El gasto de alojamiento solo se considerará en Madrid, pues los otros planes se han planteado para ida y vuelta el mismo día.
- (2) Distancia
- (3) Diversión

Los primeros dos criterios son de **minimizar** y el último es de **maximizar**.

Con el propósito de esta valoración, se llevarán a cabo distintos **métodos de decisión multicriterio**:

- AHP
- Electre
- Promethee
- Axiomático de Arrow y Raymond

En base a los juicios emitidos por el grupo de amigos, las **matrices de comparaciones** a pares son:

Comparación de Criterios:

	Coste	Distancia	Diversión
Coste	1.00	3	0.33
Distancia	0.33	1	0.20
Diversión	3.00	5	1.00

Criterio: Coste

	Trans.	Comida	Alojamiento
Trans.	1	0.33	0.14
Comida	3	1.00	0.33
Alojamiento	7	3.00	1.00

Subcriterio: Transporte

	Madrid	Sevilla	Cádiz	Sierra
Madrid	1	0.14	0.33	0.5
Sevilla	7	1.00	3.00	5.0
Cádiz	3	0.33	1.00	2.0
Sierra	2	0.20	0.50	1.0

Subcriterio: Comida

	Madrid	Sevilla	Cádiz	Sierra
Madrid	1	0.33	0.14	0.14
Sevilla	3	1.00	0.20	0.20
Cádiz	7	5.00	1.00	1.00
Sierra	7	5.00	1.00	1.00

Subcriterio: Alojamiento

	Madrid	Sevilla	Cádiz	Sierra
Madrid	1	0.11	0.11	0.11
Sevilla	9	1.00	1.00	1.00
Cádiz	9	1.00	1.00	1.00
Sierra	9	1.00	1.00	1.00

Criterio: Distancia

	Madrid	Sevilla	Cadiz	Sierra
Madrid	1	0.14	0.33	0.5
Sevilla	7	1.00	3.00	5.0
Cadiz	3	0.33	1.00	2.0
Sierra	2	0.20	0.50	1.0

Criterio: Diversión

	Madrid	Sevilla	Cadiz	Sierra
Madrid	1	0.14	0.2	0.2
Sevilla	7	1.00	3.0	3.0
Cadiz	5	0.33	1.0	1.0
Sierra	5	0.33	1.0	1.0

METODO AHP:

Se han estudiado dos vías para la resolución de un problema multicriterio por el método AHP. La primera es mediante las funciones de clase y la segunda a través de la librería ahp de R. veamos ambas formas:

Funciones de clase:

Cargamos los archivos donde están definidas las funciones que vamos a utilizar para resolver el problema multicriterio. Estos nos servirán para la implementación de todos los métodos que vamos a considerar.

```
source("teoriadecision_funciones_multicriterio.R")
source("teoriadecision_funciones_multicriterio_diagram.R")
source("teoriadecision_funciones_multicriterio_utiles.R")
```

Cargamos las librerías y funciones necesarias:

```
library(formattable)
library(htmltools)
library(webshot)

export_formattable <- function(f, file, width = "100%", height = NULL,
background = "white", delay = 0.2)
{
  w <- formattable::as.htmlwidget(f, width = width, height = height)
  path <- htmltools::html_print(w, background = background, viewer = NULL)
  url <- paste0("file://", gsub("\\\\", "/", normalizePath(path)))
  webshot::webshot(url,
file = file,
selector = ".formattable_widget",
delay = delay)
}
```

Creamos las matrices de comparaciones y guardamos los resultados obtenidos al aplicar el método AHP:

```
# Comparaciones entre los criterios:

tb_C <- multicriterio.crea.matrizvaloraciones_mej(c(3, 1/3, 1/5),
numalternativas = 3,
c("Coste", "Distancia",
"Diversión"))

res_C <- multicriterio.metodoAHP.variante3.basico(tb_C)

# Comparaciones entre los subcriterios:
```

```

tb_SC <- multicriterio.crea.matrizvaloraciones_mej(c(1/3, 1/7, 1/3),
                                                    numalternativas =3,
                                                    c("Trans.", "Comida",
                                                      "Alojamiento"))

res_SC <- multicriterio.metodoAHP.variante3.basico(tb_SC)

# Transporte

tb_trans <- multicriterio.crea.matrizvaloraciones_mej(c(1/7, 1/3, 1/2, 3, 5,
2),
                                                    numalternativas =4,
                                                    c("Madrid", "Sevilla",
                                                      "Cadiz", "Sierra"))

res_trans <- multicriterio.metodoAHP.variante3.basico(tb_trans)

# Comida

tb_comida <- multicriterio.crea.matrizvaloraciones_mej(c(1/3, 1/7, 1/7, 1/5,
1/5, 1),
                                                    numalternativas =4,
                                                    c("Madrid", "Sevilla",
                                                      "Cadiz", "Sierra"))

res_comida <- multicriterio.metodoAHP.variante3.basico(tb_comida)

# Alojamiento

tb_alojamiento <- multicriterio.crea.matrizvaloraciones_mej(c(1/9, 1/9, 1/9,
1, 1, 1),
                                                    numalternativas =4,
                                                    c("Madrid", "Sevilla",
                                                      "Cadiz", "Sierra"))

res_alojamiento <- multicriterio.metodoAHP.variante3.basico(tb_alojamiento)

# Distancia

tb_dist <- multicriterio.crea.matrizvaloraciones_mej(c(1/7,1/3,1/2,3,5,2),
                                                    numalternativas =4,
                                                    c("Madrid", "Sevilla",
                                                      "Cadiz", "Sierra"))

res_dist <- multicriterio.metodoAHP.variante3.basico(tb_dist)

# Diversion

```

```
tb_diversion <-
multicriterio.crea.matrizvaloraciones_mej(c(1/7,1/5,1/5,3,3,1),
                                           numalternativas =4,
                                           c("Madrid", "Sevilla",
                                              "Cadiz", "Sierra"))

res_diversion <- multicriterio.metodoAHP.variante3.basico(tb_diversion)
```

Calculamos ahora los pesos de los criterios/subcriterios con respecto a elegir el mejor plan de ocio:

```
c11 <- (as.numeric(res_C$valoraciones.ahp[1]*res_SC$valoraciones.ahp[1]))
c12 <- (as.numeric(res_C$valoraciones.ahp[1]*res_SC$valoraciones.ahp[2]))
c13 <- (as.numeric(res_C$valoraciones.ahp[1]*res_SC$valoraciones.ahp[3]))
c2 <- (as.numeric(res_C$valoraciones.ahp[2]))
c3 <- (as.numeric(res_C$valoraciones.ahp[3]))

crisub = c(c11,c12,c13,c2,c3)
```

Agrupamos en una matriz los pesos locales de las alternativas:

```
mat <- matrix(c(as.vector(res_trans$valoraciones.ahp),
                 as.vector(res_comida$valoraciones.ahp),
                 as.vector(res_alojamiento$valoraciones.ahp),
                 as.vector(res_dist$valoraciones.ahp),
                 as.vector(res_diversion$valoraciones.ahp)),
              ncol = 4,
              nrow = 5,
              byrow = TRUE)
```

Por último, calculamos los pesos globales de cada una de las alternativas:

```
pesos.globales = crisub %*% mat
colnames(pesos.globales) = c("Madrid", "Sevilla", "Cadiz", "Sierra.M")
knitr::kable(pesos.globales)
```

Madrid	Sevilla	Cadiz	Sierra.M
0.0523337	0.4701735	0.2448409	0.2326519

Por lo tanto, concluimos que la alternativa pasar el domingo en el centro de Sevilla es la óptima (con un peso global de 47% aprox.), seguida de pasar el sábado en alguna playa de Cádiz (con un peso global de 24% aprox.), una ruta de senderismo por Sierra Morena (con un peso global de 23% aprox.) y por último, la peor alternativa es pasar el fin de semana en Madrid (con un peso global de 5% aprox.).

Estudio de la inconsistencia

Al tratarse de matrices 3x3 y 4x4 hay que estudiar la inconsistencia.

```
incons_1 = multicriterio.metodoAHP.coef.inconsistencia(tb_C)
incons_2 = multicriterio.metodoAHP.coef.inconsistencia(tb_SC)
incons_3 = multicriterio.metodoAHP.coef.inconsistencia(tb_trans)
incons_4 = multicriterio.metodoAHP.coef.inconsistencia(tb_comida)
incons_5 = multicriterio.metodoAHP.coef.inconsistencia(tb_alojamiento)
incons_6 = multicriterio.metodoAHP.coef.inconsistencia(tb_dist)
incons_7 = multicriterio.metodoAHP.coef.inconsistencia(tb_diversion)

c(incons_1$mensaje, round(incons_1$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0332"
```

```
c(incons_2$mensaje, round(incons_2$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0061"
```

```
c(incons_3$mensaje, round(incons_3$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0071"
```

```
c(incons_4$mensaje, round(incons_4$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0272"
```

```
c(incons_5$mensaje, round(incons_5$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0"
```

```
c(incons_6$mensaje, round(incons_6$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0071"
```


Librería AHP:

Escribimos el modelo en un fichero ahp, "ejmultinivel.ahp".

Cargamos el modelo

```
library(ahp)
datos = Load("ejmultinivel.ahp")
```

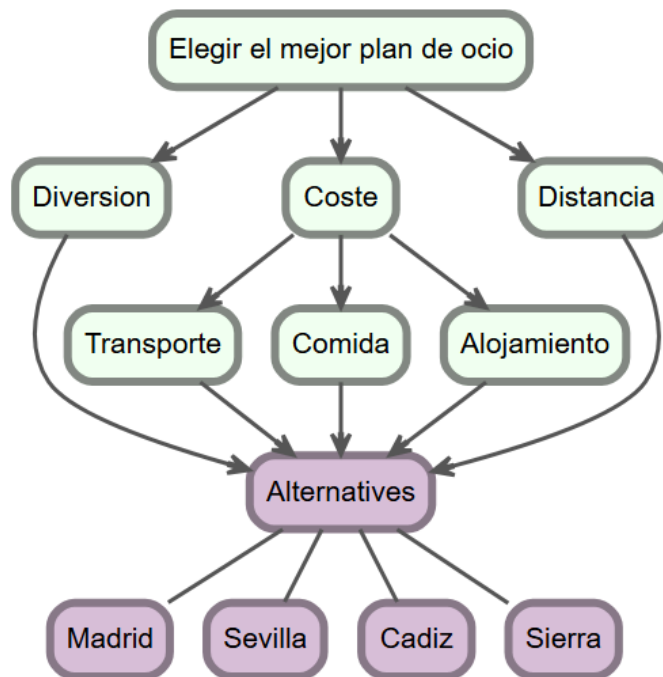
Calculamos las prioridades

```
Calculate(datos)
print(datos, priority = function(x) x$parent$priority["Total", x$name])
```

	levelName	priority
1	Elegir el mejor plan de ocio	NA
2	--Diversión	0.63698557
3	--Madrid	0.05187826
4	--Sevilla	0.52811896
5	--Cadiz	0.21000139
6	°--Sierra	0.21000139
7	--Coste	0.25828499
8	--Transporte	0.08794621
9	--Madrid	0.07217160
10	--Sevilla	0.58716568
11	--Cadiz	0.21787643
12	°--Sierra	0.12278629
13	--Comida	0.24263692
14	--Madrid	0.05059764
15	--Sevilla	0.10437661
16	--Cadiz	0.42251288
17	°--Sierra	0.42251288
18	°--Alojamiento	0.66941687
19	--Madrid	0.03571429
20	--Sevilla	0.32142857
21	--Cadiz	0.32142857
22	°--Sierra	0.32142857
23	°--Distancia	0.10472943
24	--Madrid	0.07217160
25	--Sevilla	0.58716568
26	--Cadiz	0.21787643
27	°--Sierra	0.12278629

Visualizamos la jerarquía

```
Visualize(datos)
```



Analizamos

```
aa = AnalyzeTable(datos, variable = "priority", sort="orig")
formattable::as.htmlwidget(aa)
```

	Priority	Madrid	Sevilla	Cadiz	Sierra	Inconsistency
Elegir el mejor plan de ocio	100.0%					3.7%
Diversión	63.7%	5.2%	52.8%	21.0%	21.0%	2.8%
Coste	25.8%					0.7%
Transporte	8.8%	7.2%	58.7%	21.8%	12.3%	0.7%
Comida	24.3%	5.1%	10.4%	42.3%	42.3%	2.8%
Alojamiento	66.9%	3.6%	32.1%	32.1%	32.1%	0.0%
Distancia	10.5%	7.2%	58.7%	21.8%	12.3%	0.7%

```
ab = AnalyzeTable(datos, sort="orig")
formattable::as.htmlwidget(ab)
```

	Weight	Madrid	Sevilla	Cádiz	Sierra	Inconsistency
Elegir el mejor plan de ocio	100.0%	5.2%	47.3%	24.4%	23.1%	3.7%
Diversion	63.7%	3.3%	33.6%	13.4%	13.4%	2.8%
Coste	25.8%	1.1%	7.5%	8.7%	8.5%	0.7%
Transporte	2.3%	0.2%	1.3%	0.5%	0.3%	0.7%
Comida	6.3%	0.3%	0.7%	2.6%	2.6%	2.8%
Alojamiento	17.3%	0.6%	5.6%	5.6%	5.6%	0.0%
Distancia	10.5%	0.8%	6.1%	2.3%	1.3%	0.7%

Luego, concluimos que la alternativa pasar el domingo en el centro de Sevilla es la óptima (con un peso global de 47.3%), seguida de pasar el sábado en alguna playa de Cádiz (con un peso global de 24.4%), una ruta de senderismo por Sierra Morena (con un peso global de 23.1%) y por último, la peor alternativa es pasar el fin de semana en Madrid (con un peso global de 5.2%).

En la primera tabla (Tabla de PESOS LOCALES) observamos que el subcriterio dentro del criterio Coste con mayor peso es Alojamiento. Dentro de este subcriterio, todas las alternativas menos Madrid tienen el mismo peso, como era de esperar, por las valoraciones otorgadas.

En la primera tabla, en el subcriterio comida, las alternativas Cádiz y Sierra tienen un mayor peso, igual para ambas. El peso dentro de los criterios o subcriterios de estas dos alternativas solo difiere en el criterio distancia y transporte, teniendo un mayor peso local la alternativa Cádiz.

La columna de inconsistencia nos proporciona valores menores a 10%, concluyendo así, que la consistencia de las matrices de comparaciones es aceptable.

Los pesos usados para los siguientes métodos son los pesos globales obtenidos por el método AHP.

METODO PROMETHEE:

Para la resolución del problema multicriterio mediante el método promethee, consideramos la siguiente matriz de decisión. Para ello, hemos tenido en cuenta que todos los criterios, menos el criterio “Diversión” son de minimizar, luego los valores correspondientes a estos están multiplicados por menos 1.

	C_trans	C_com	C_aloj	Dist	Div
Madrid	-200	-100	-200	-550	3
Sevilla	-5	-50	0	-30	7
Cadiz	-50	-5	0	-100	5
Sierra M	-70	-5	0	-200	5

Como hemos mencionado anteriormente, los pesos utilizados en este método son aquellos obtenidos mediante el método AHP.

Además las funciones de preferencia de cada criterio son las siguientes:

- Criterio Coste_Transporte -> Cuasi Criterio (q = 25)
- Criterio Coste_Comida -> Cuasi Criterio (q = 15)
- Criterio Coste_Alojamiento -> Cuasi Criterio (q = 20)
- Criterio Distancia -> Criterio Nivel (q = 30, p = 70)
- Criterio Diversión -> Criterio Nivel (q = 1, p = 3)

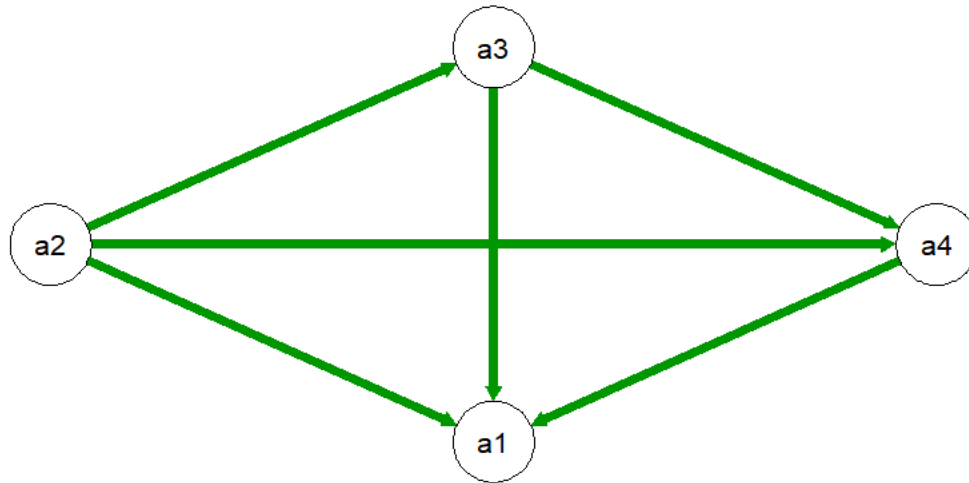
Promethee I:

Creamos la matriz de decisión, introducimos los pesos y las funciones de preferencia.

```
t = multicriterio.crea.matrizdecision(  
    c(-200,-100,-200,-550,3,  
      -5, -50, 0, -30, 7,  
      -50, -5, 0, -100, 5,  
      -70, -5, 0, -200, 5),  
    numalternativas = 4, numcriterios = 5)  
  
pesos = crisub  
tab.fpref = matrix(c( 2, 25, 0,0,  
                     2, 15, 0,0,  
                     2, 20, 0,0,  
                     4, 30,70,0,  
                     4, 1, 3,0), ncol = 4, byrow=T)
```

Resolvemos por el Metodo Promethee I:

```
tab = multicriterio.metodo.promethee_i(t, pesos, tab.fpref)
require("qgraph")
qgraph(tab$tablarelacionsupera)
```

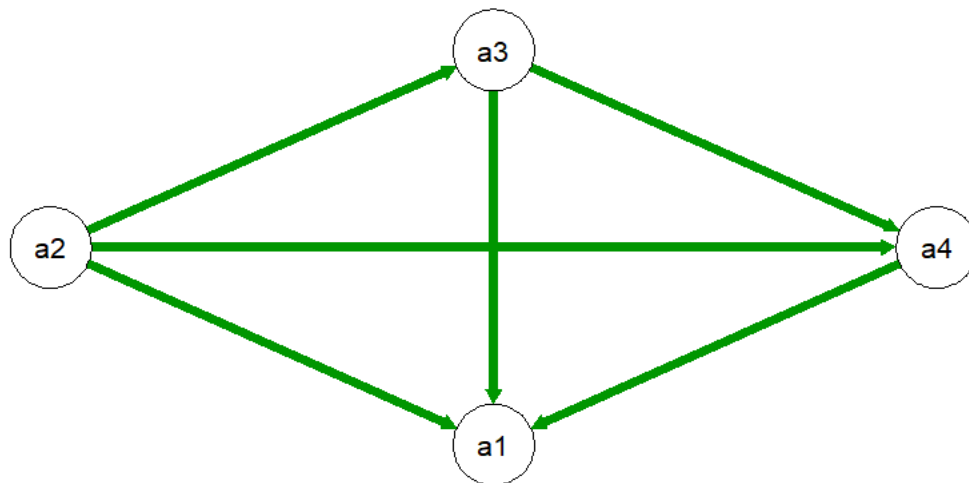


Vemos como la alternativa 2 domina al resto de alternativas y la alternativa 1 es dominada por todas las demás. Las alternativas 3 y 4 dominan y son dominadas.

Promethee II:

Resolvemos por el Metodo Promethee II:

```
tab2 = multicriterio.metodo.promethee_ii(t, pesos, tab.fpref)
require("qgraph")
qgraph(tab2$tablarelacionsupera)
```



La ordenación final de las alternativas es:

```
order(tab2$vflujos.netos, decreasing= T)
```

```
[1] 2 3 4 1
```

Se observa que, mediante el método Promethee II (de ordenación total), la alternativa óptima es la 2 (Pasar el domingo en el centro de Sevilla), seguido de la alternativa 3 (Pasar el día en alguna playa de Cádiz), la alternativa 4 (Ruta de senderismo por Sierra Morena) y, por último, se concluye, que la peor alternativa es la 1 (Pasar el finde en Madrid).

Resolución con Promethee Windows:

```
res = multicriterio.metodo.promethee_windows(t, tab.fpref, round(pesos,4),
                                             fminmax = c("min", "min","min",
"min", "max"))
knitr::kable(res$Escenario)
```

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5
Min/Max	min	min	min	min	max
Pesos	0.023	0.0633	0.1742	0.1062	0.6333
Funciones Preferencias	U-shape (2)	U-shape (2)	U-shape (2)	Level (4)	Level (4)
Q: Indiferencia	25	15	20	30	1
P: Preferencia	0	0	0	70	3
S: Gausiano	0	0	0	0	0
Mínimo	5	5	0	30	3
Máximo	200	100	200	550	7
Media	81.25	40	50	220	5
Desviación Típica	72.49	39.21	86.6	199.87	1.41
a1	200	100	200	550	3
a2	5	50	0	30	7
a3	50	5	0	100	5
a4	70	5	0	200	5

Vemos los flujos entrantes, salientes y netos ordenados según las alternativas óptimas.

```
knitr::kable(res$Acciones, align = c('c','c','c','c'))
```

	Rango	Phi	Phi.mas	Phi.menos
a2	1	0.5884	0.6306	0.0422
a3	2	0.1357	0.2843	0.1486
a4	3	0.0649	0.2489	0.1840
a1	4	-0.7889	0.0000	0.7889

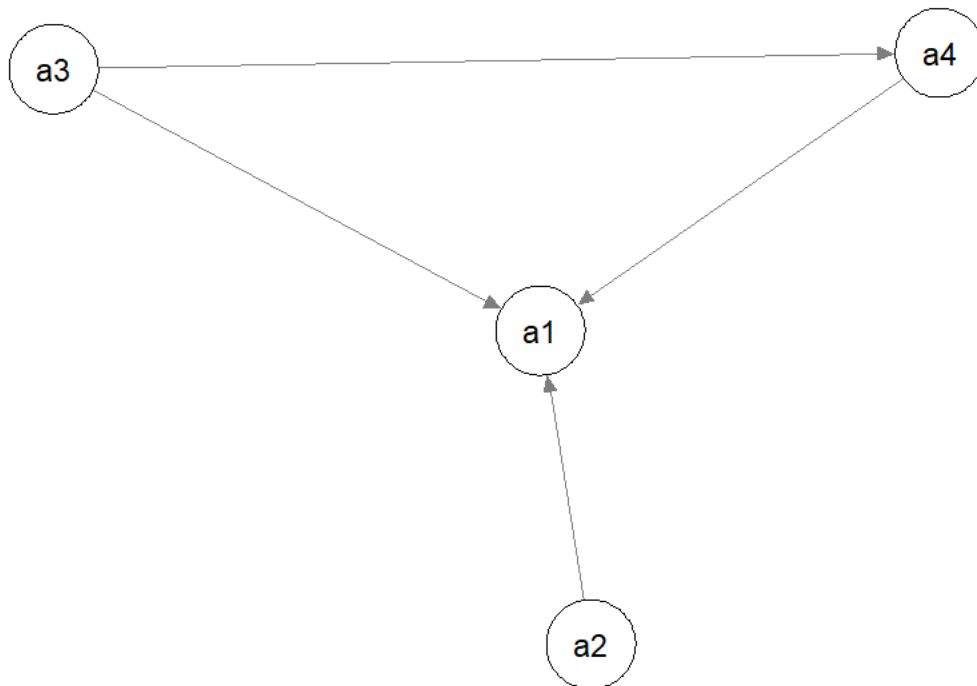
METODO ELECTRE:

Para la resolución del problema multicriterio mediante el método Electre, vamos a hacer uso de la matriz de decisión considerada para el método Promethee, así como de los pesos.

```
t = multicriterio.crea.matrizdecision(  
  c(-200, -100, -200, -550, 3,  
    -5, -50, 0, -30, 7,  
    -50, -5, 0, -100, 5,  
    -70, -5, 0, -200, 5),  
  numalternativas = 4, numcriterios = 5)  
  
pesos = crisub  
d = c(100, 20, Inf, Inf, 2)
```

Resolvemos mediante el Método Electre:

```
soll = multicriterio.metodoELECTRE_I(t, pesos.criterios = pesos,  
  nivel.concordancia.minimo.alpha = 0.7,  
  no.se.compensan = d,  
  que.alternativas = TRUE)  
  
qgraph::qgraph(soll$relacion.dominante)
```



La alternativa 1 es dominada por todas. La alternativa 4 domina a la alternativa 1 y es dominada por la alternativa 3.

En la primera iteración, se tiene que el núcleo (conjunto de alternativas que dominan al resto y no son dominadas por nadie) es:

```
sol1$nucleo_aprox
```

```
a2 a3  
2  3
```

Aplicamos el método Electre a las alternativas del núcleo:

```
sol2 = multicriterio.metodoELECTRE_I(t, pesos.criterios = pesos,  
                                     nivel.concordancia.minimo.alpha = 0.7,  
                                     no.se.compensan = d,  
                                     que.alternativas = c(2,3))  
  
qgraph::qgraph(sol2$relacion.dominante)
```



a2



a3

Reducimos el valor de alpha a 0.55:

```
sol3 = multicriterio.metodoELECTRE_I(t, pesos.criterios = pesos,  
                                     nivel.concordancia.minimo.alpha = 0.55,  
                                     no.se.compensan = d,  
                                     que.alternativas = c(2,3))  
  
qgraph::qgraph(sol3$relacion.dominante)
```



a2



a3

Por último, modificamos d:

```
sol4 = multicriterio.metodoELECTRE_I(t, pesos.criterios = pesos,  
                                     nivel.concordancia.minimo.alpha = 0.55,  
                                     no.se.compensan = c(50, 10, Inf, Inf, 4),  
                                     que.alternativas = c(2,3))
```

```
qgraph::qgraph(sol4$relacion.dominante)
```



a2



a3

Tras haber probado para distintos valores de d y no llegar a una única alternativa, concluimos con este método que las dos alternativas óptimas son “Pasar el domingo en el Centro de Sevilla” y “Pasar el sábado en alguna playa de Cádiz”. No podemos concluir el dominio de una de estas dos alternativas sobre la otra.

Tablas de Concordancia, Discordancia y Superación:

Veamos las tablas de concordancia y discordancia para la primera iteración.

```
s1 = func_ELECTRE_resTConcordancia(sol1)
kableExtra::save_kable(s1$KE, file = "electreTConc.html")
webshot2::webshot("electreTConc.html", file = "electreTConc.png")
```

Alts	Inds	A1	A2	A3	A4
A1	ljk	1	0	0	0
	IGjk	NaN	0	0	0
	TC	F	F	F	F
A2	ljk	1	1	0.9367	0.9367
	IGjk	Inf	NaN	12.0403	12.0403
	TC	T	F	T	T
A3	ljk	1	0.2375	1	1
	IGjk	Inf	0.0831	NaN	Inf
	TC	T	F	F	T
A4	ljk	1	0.2375	0.8709	1
	IGjk	Inf	0.0831	0	NaN
	TC	T	F	F	F

ELECTRE: Test Concordancia

Vemos por ejemplo que (a2,a3) y (a2,a4) han superado el test de concordancia.

```
s2 = func_ELECTRE_resTDiscordancia(sol1)
kableExtra::save_kable(s2$KE, file = "electreTDisc.html")
webshot2::webshot("electreTDisc.html", file = "electreTDisc.png")
```

Alts	Inds	A1	A2	A3	A4
A1	I-		1,2,3,4,5	1,2,3,4,5	1,2,3,4,5
	Djk		195(100),50(20),200(Inf),520(Inf),4(2)	150(100),95(20),200(Inf),450(Inf),2(2)	130(100),95(20),200(Inf),350(Inf),
	TD	T	F	F	F
A2	I-			2	2
	Djk	NA		45(20)	45(20)
	TD	T	T	F	F
A3	I-		1,4,5		
	Djk	NA	45(100),70(Inf),2(2)		NA
	TD	T	F	T	T
A4	I-		1,4,5	1,4	
	Djk	NA	65(100),170(Inf),2(2)	20(100),100(Inf)	
	TD	T	F	T	T

ELECTRE: Test Discordancia: vd=(100,20,Inf,Inf,2)

Vemos como los pares que hemos mencionado para la concordancia, no han superado el test de discordancia.

Para estudiar los pares de alternativas que han superado ambos tests, nos fijamos en la siguiente tabla:

```
s3 = func_ELECTRE_resTSuperacion(sol1)
kableExtra::save_kable(s3$KE, file = "electreTSup.html")
webshot2::webshot("electreTSup.html", file = "electreTSup.png")
```

Alts	Inds	A1	A2	A3	A4
A1	TC	F	F	F	F
	TD	T	F	F	F
	RSup	F	F	F	F
A2	TC	T	F	T	T
	TD	T	T	F	F
	RSup	T	F	F	F
A3	TC	T	F	F	T
	TD	T	F	T	T
	RSup	T	F	F	T
A4	TC	T	F	F	F
	TD	T	F	T	T
	RSup	T	F	F	F

ELECTRE: Relación Superación: $\alpha = 0.7$, $vd=(100,20,Inf,Inf,2)$

Observamos que los pares de alternativas (con la primera dominando la segunda) que han superado ambos tests son: (a2,a1), (a3, a1), (a4, a1), (a3,a4). Esto también lo hemos visto en el primer grafo, donde el arco de un nodo a otro, indica el dominio del nodo saliente sobre el entrante.

METODO AXIOMATICO DE ARROW Y RAYMOND

Para este método consideramos la misma matriz de decisión que la de los métodos Promethee y Electre.

Implementamos el método:

```
sol = multicriterio.metodoaxiomatico.ArrowRaymond(t)
```

Las iteraciones/pasos del método son:

```
sol$pasos
```

```
[[1]]
[[1]]$Mclasificacion
  a1 a2 a3 a4
a1 NA 0.0 0.0 0.0
a2  5  NA 3.5 3.5
a3  5 1.5  NA 3.5
a4  5 1.5 1.5  NA

[[1]]$max.filas
a1 a2 a3 a4
0  5  5  5

[[1]]$indices.ordenados
[1] 1 2 3 4

[[1]]$alternativa.sale
[1] "a1"

[[2]]
[[2]]$Mclasificacion
  a2 a3 a4
a2  NA 3.5 3.5
a3 1.5  NA 3.5
a4 1.5 1.5  NA

[[2]]$max.filas
a2 a3 a4
3.5 3.5 1.5

[[2]]$indices.ordenados
[1] 3 1 2

[[2]]$alternativa.sale
[1] "a4"
```

```

[[3]]
[[3]]$Mclasificacion
      a2  a3
a2  NA 3.5
a3 1.5  NA

[[3]]$max.filas
      a2  a3
3.5 1.5

[[3]]$indices.ordenados
[1] 2 1

[[3]]$alternativa.sale
[1] "a3"

```

Una vez hemos visto todas las iteraciones, vemos las alternativas ordenadas:

```
sol$alternativasordenadas
```

```
[1] "a2" "a3" "a4" "a1"
```

De manera que la alternativa óptima segundo este método es la 2, “Pasar el domingo en el centro de Sevilla”, seguida de la alternativa 3 y 4 y, concluyendo que la peor alternativa es “Pasar el fin de semana en Madrid”.

Conclusión Final

Tras haber aplicado los distintos métodos estudiados en clase para nuestro problema de decisión multicriterio, en el que teníamos que elegir el mejor plan para un fin de semana, podemos concluir que “Pasar el domingo en el Centro de Sevilla” es elegida como la alternativa óptima según la mayoría de los métodos.

A pesar de esto, en el método Electre no hemos podido obtener una relación de dominio entre las alternativas “Pasar el domingo en el Centro de Sevilla” y “Pasar el sábado en alguna playa de Cádiz”.

Por último, recalcar que la alternativa “Pasar el fin de semana en Madrid” ha sido elegida como la peor alternativa según todos los métodos utilizados.