

Relacion 2

Curso 2024-2025

Índice

EJERCICIO 1	3
Apartado a)	3
Apartado b)	4
Apartado c)	4
Apartado d)	5
Apartado e)	6
EJERCICIO 2	7
Apartado a)	7
Apartado b)	8
EJERCICIO 3	9
Apartado a)	9
Apartado b)	10
EJERCICIO 4	10
Funciones de clase	10
Diagrama Jerarquias	11
Método ahp	12
EJERCICIO 5	14
Funciones de clase	14
Diagrama Jerarquias	15
Método ahp	16
EJERCICIO 6	18
Funciones de clase	18
Método 1 (mayor autovalor)	18
Método 2 (completo)	19
Diagrama Jerarquias	19
Estudio de la inconsistencia con funciones R de clase.	20
Método ahp	21

EJERCICIO 7	22
Iteración 1. Introducir datos y resolver	22
Iteración 2 y 3. Se reducen alternativas y/o alpha	23
Método ELECTRE I	25
Método PROMETHEE	26
PROMETHEE I	26
PROMETHEE II	28
PROMETHEE I (medias)	29
PROMETHEE II (medias)	30
Ordenación final alternativas Mét. Promethee II (medias)	31
Comparativa Promethee II: sin medias y con medias	31
Resolución con Promethee Windows	31
EJERCICIO 8	32
PROMETHEE I	33
PROMETHEE II	34
PROMETHEE I (medias)	35
PROMETHEE II (medias)	36
Ordenación final alternativas Mét. Promethee II (medias)	37
Comparativa Promethee II: sin medias y con medias	38
Resolución con Promethee Windows	38

EJERCICIO 1

Apartado a)

```
tab01a <- multicriterio.crea.matrizvaloraciones(c(1,0,1,
                                                  1,1,1,
                                                  0,0,1), numalternativas = 3)
```

	a1	a2	a3
a1	1	0	1
a2	1	1	1
a3	0	0	1

- Método de construcción de la función de utilidad “Maximal”

```
(sol01a <- multicriterio.constfuncutilidad.maximales(tab01a)) # suma por filas
```

```
## a1 a2 a3
## 2 3 1
```

Cuánto más alto mejor, por lo tanto nos está diciendo que la mejor alternativa sería a2. La siguiente es la a1 y la peor es a 3. Ahora ordenamos de mejor a peor:

```
sort(sol01a, decreasing = T)
```

```
## a2 a1 a3
## 3 2 1
```

- Método de construcción de la función de utilidad “Borroso”

```
(sol01aBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab01a)) #calcula del flujo neto
```

```
## a1 a2 a3
## 0 2 -2
```

```
sort(sol01aBor, decreasing = T)
```

```
## a2 a1 a3
## 2 0 -2
```

La mejor es la alternativa 2

Apartado b)

```
tab01b <- multicriterio.crea.matrizvaloraciones(c(1,0,1,
                                                  1,1,1,
                                                  1,0,1), numalternativas = 3)
```

	a1	a2	a3
a1	1	0	1
a2	1	1	1
a3	1	0	1

- Método de construcción de la función de utilidad “Maximal”.

```
(sol01b <- multicriterio.constfuncutilidad.maximales(tab01b)) # suma por filas
```

```
## a1 a2 a3
## 2 3 2
```

```
sort(sol01b, decreasing = T)
```

```
## a2 a1 a3
## 3 2 2
```

La mejor es la alternativa 2

- Método de construcción de la función de utilidad “Borroso”

```
(sol01bBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab01b)) #calcula del flujo neto
```

```
## a1 a2 a3
## -1 2 -1
```

```
sort(sol01bBor, decreasing = T)
```

```
## a2 a1 a3
## 2 -1 -1
```

La mejor es la alternativa 2

Apartado c)

```
tab01c <- multicriterio.crea.matrizvaloraciones(rep(1,9), numalternativas = 3)
```

	a1	a2	a3
a1	1	1	1
a2	1	1	1
a3	1	1	1

- Método de construcción de la función de utilidad “Maximal”.

```
(sol01c <- multicriterio.constfuncutilidad.maximales(tab01c)) # suma por filas
```

```
## a1 a2 a3
## 3 3 3
```

```
sort(sol01c, decreasing = T)
```

```
## a1 a2 a3
## 3 3 3
```

Cualquiera de los 3 es la mejor alternativa

- Método de construcción de la función de utilidad “Borroso”

```
(sol01cBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab01c)) #calcula del flujo neto
```

```
## a1 a2 a3
## 0 0 0
```

```
sort(sol01cBor, decreasing = T)
```

```
## a1 a2 a3
## 0 0 0
```

Cualquiera de los 3 es la mejor alternativa

Apartado d)

```
tab01d <- multicriterio.crea.matrizvaloraciones(c(1,1,1,0,1,
                                                    0,1,0,0,1,
                                                    1,0,1,0,1,
                                                    1,1,1,1,0,
                                                    0,0,0,0,1), numalternativas = 5)
```

	a1	a2	a3	a4	a5
a1	1	1	1	0	1
a2	0	1	0	0	1
a3	1	0	1	0	1
a4	1	1	1	1	0
a5	0	0	0	0	1

- Método de construcción de la función de utilidad “Maximal”.

```
(sol01d <- multicriterio.constfuncutilidad.maximales(tab01d)) # suma por filas
```

```
## a1 a2 a3 a4 a5
## 4 2 4 5 1
```

```
sort(sol01d, decreasing = T)
```

```
## a4 a1 a3 a2 a5
## 5 4 4 2 1
```

La mejor es la alternativa 4

- Método de construcción de la función de utilidad “Borroso”

```
(sol01dBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab01d)) #calcula del flujo neto
```

```
## a1 a2 a3 a4 a5
## 1 -1 0 3 -3
```

```
sort(sol01dBor, decreasing = T)
```

```
## a4 a1 a3 a2 a5
## 3 1 0 -1 -3
```

La mejor es la alternativa 4

Apartado e)

```
tab01e <- multicriterio.crea.matrizvaloraciones(c(1,1,1,0,1,
                                                    0,1,0,0,1,
                                                    1,1,1,0,1,
                                                    1,1,1,1,0,
                                                    0,0,0,0,1), numalternativas = 5)
```

	a1	a2	a3	a4	a5
a1	1	1	1	0	1
a2	0	1	0	0	1
a3	1	1	1	0	1
a4	1	1	1	1	0
a5	0	0	0	0	1

- Método de construcción de la función de utilidad “Maximal”.

```
(sol01e <- multicriterio.constfuncutilidad.maximales(tab01e)) # suma por filas
```

```
## a1 a2 a3 a4 a5
## 4 2 4 5 1
```

```
sort(sol01e, decreasing = T)
```

```
## a4 a1 a3 a2 a5
## 5 4 4 2 1
```

La mejor es la alternativa 4

- Método de construcción de la función de utilidad “Borroso”

```
(sol01eBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab01e)) #calcula del flujo neto
```

```
## a1 a2 a3 a4 a5
## 1 -2 1 3 -3
```

```
sort(sol01eBor, decreasing = T)
```

```
## a4 a1 a3 a2 a5
## 3 1 1 -2 -3
```

La mejor es la alternativa 4

EJERCICIO 2

Apartado a)

```
tab02a <- multicriterio.crea.matrizvaloraciones(c(0,0.4,0.7,
                                                  0.2,0,0.5,
                                                  0.3,0.6,0), numalternativas = 3)
```

	a1	a2	a3
a1	0.0	0.4	0.7
a2	0.2	0.0	0.5
a3	0.3	0.6	0.0

- Método de construcción de la función de utilidad “Maximal”.

```
(sol02a <- multicriterio.constfuncutilidad.maximales(tab02a)) # suma por filas
```

```
## a1 a2 a3
## 3 1 2
```

```
sort(sol02a, decreasing = T)
```

```
## a1 a3 a2
## 3 2 1
```

La mejor es la alternativa 1

- Método de construcción de la función de utilidad “Borroso”

```
(sol02aBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab02a)) #calcula del flujo neto
```

```
## a1 a2 a3
## 0.6 -0.3 -0.3
```

```
sort(sol02aBor, decreasing = T)
```

```
## a1 a2 a3
## 0.6 -0.3 -0.3
```

La mejor es la alternativa 1

Apartado b)

```
tab02b <- multicriterio.crea.matrizvaloraciones(c(0,0.2,0.4,
                                                  0.9,0,0.8,
                                                  0.1,0.3,0), numalternativas = 3)
```

	a1	a2	a3
a1	0.0	0.2	0.4
a2	0.9	0.0	0.8
a3	0.1	0.3	0.0

- Método de construcción de la función de utilidad “Maximal”.

```
(sol02b <- multicriterio.constfuncutilidad.maximales(tab02b)) # suma por filas
```

```
## a1 a2 a3
## 2 3 1
```



```
sort(sol02b, decreasing = T)
```

```
## a2 a1 a3
## 3 2 1
```

La mejor es la alternativa 2

- Método de construcción de la función de utilidad “Borroso”

```
(sol02bBor <- multicriterio.constfuncutilidad.estructuraborrosa(tab02b)) #calculo del flujo neto
```

```
## a1 a2 a3
## -0.4 1.2 -0.8
```

```
sort(sol02bBor, decreasing = T)
```

```
## a2 a1 a3
## 1.2 -0.4 -0.8
```

La mejor es la alternativa 2

EJERCICIO 3

```
tab03 <- multicriterio.crea.matrizdecision(c(100,15,7,40,50,
                                             200,25,7,60,200,
                                             100,20,4,25,25,
                                             200,30,20,70,350,
                                             250,25,25,100,500), numalternativas = 5, numcriterios = 5)
```

	C1	C2	C3	C4	C5
a1	100	15	7	40	50
a2	200	25	7	60	200
a3	100	20	4	25	25
a4	200	30	20	70	350
a5	250	25	25	100	500

Apartado a)

Vamos a homogeneizar las columnas de la tabla de decisión por el método Nadir

```
sol03a <- round(multicriterio.homogeneizacion.nadir(tab03), 4)
```

	C1	C2	C3	C4	C5
a1	0.0000	0.0000	0.1429	0.2000	0.0526
a2	0.6667	0.6667	0.1429	0.4667	0.3684
a3	0.0000	0.3333	0.0000	0.0000	0.0000
a4	0.6667	1.0000	0.7619	0.6000	0.6842
a5	1.0000	0.6667	1.0000	1.0000	1.0000

Apartado b)

Vamos a homogeneizar las columnas de la tabla de decisión por el método Ptomethee

```
sol03b <- round(multicriterio.homogeneizacion.promethee(tab03,
                                                         v.delta.min = c(30,3,4,20,100),
                                                         v.delta.max = c(120,12,10,60,400)), 4)
```

	C1	C2	C3	C4	C5
a1	-2.5556	-2.7778	-2.0000	-1.250	-1.8333
a2	1.3333	0.7778	-2.0000	-0.125	-0.4167
a3	-2.5556	-1.0000	-2.0000	-2.000	-2.0000
a4	1.3333	2.2222	2.8333	0.625	1.4167
a5	2.4444	0.7778	3.1667	2.750	2.8333

EJERCICIO 4

Funciones de clase

Introducción datos

```
#matriz criterios
tab1 <- multicriterio.crea.matrizvaloraciones_mej(c(2), numalternativas = 2,
                                                  v.nombres.alternativas = c("Rendimiento",
                                                                              "Riesgo"))

#matriz rendimiento
tab2 <- multicriterio.crea.matrizvaloraciones_mej(c(3), numalternativas = 2,
                                                  v.nombres.alternativas = c("A", "B"))

#matriz riesgo
tab3 <- multicriterio.crea.matrizvaloraciones_mej(c(1/2), numalternativas = 2,
                                                  v.nombres.alternativas = c("A", "B"))
```

Cálculo pesos locales

Método mayor autovalor

```
pes1 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab1)
pes2 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab2)
pes3 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab3)
```

Cálculo pesos globales

```
tab04 <- multicriterio.metodoAHP.pesosglobales_entabla(pes1$valoraciones.ahp,
                                                         rbind(pes2$valoraciones.ahp,
                                                         pes3$valoraciones.ahp))
```

	Rendimiento	Riesgo	Ponderadores Globales
A	0.7500000	0.3333333	0.6111111
B	0.2500000	0.6666667	0.3888889
Ponder.Criterios	0.6666667	0.3333333	NA

La mejor decisión es la alternativa A (peso global del 61,11%)

```
which.max(tab04[,1])
```

```
## A
## 1
```

```
which.max(tab04[,2])
```

```
## B
## 2
```

Para el rendimiento la mejor alternativa es la A

Para el riesgo la mejor alternativa es la B

Además del método de mayor autovalor, tambien tenemos:

Método de media geométrica

```
pes1 <- multicriterio.metodoAHP.variante2.mediageometrica(tab1)
pes2 <- multicriterio.metodoAHP.variante2.mediageometrica(tab2)
pes3 <- multicriterio.metodoAHP.variante2.mediageometrica(tab3)
```

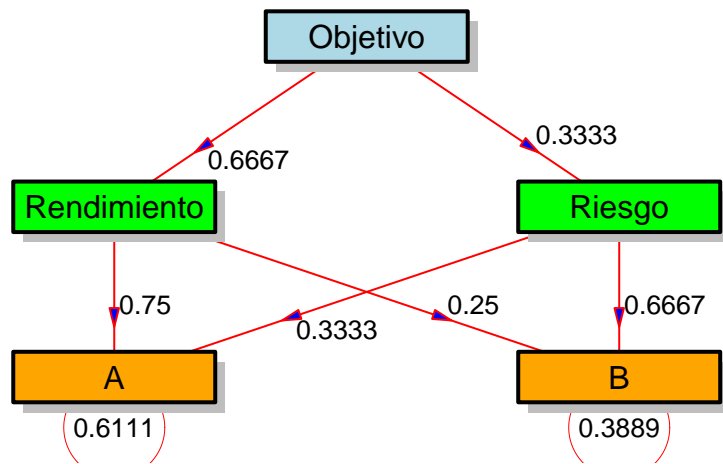
Método básico

```
pes1 <- multicriterio.metodoAHP.variante3.basico(tab1)
pes2 <- multicriterio.metodoAHP.variante3.basico(tab2)
pes3 <- multicriterio.metodoAHP.variante3.basico(tab3)
```

Diagrama Jerarquias

```
num.alt <- 2
num.crt <- 2
Xmatriznivel2_04 <- array(NA, dim = c(num.alt, num.alt, num.crt))
Xmatriznivel2_04[, ,1] <- tab2
Xmatriznivel2_04[, ,2] <- tab3
dimnames(Xmatriznivel2_04)[[1]] <- c("A","B")
multicriterio.metodoahp.diagrama(tab1, Xmatriznivel2_04)
```

Estructura Jerárquica (AHP)



La mejor decisión es la alternativa A (peso global del 61,11%)

Método ahp

```
library(ahp)
datos04 = Load("problema4.ahp")
Calculate(datos04)
```

```
Visualize(datos04)
```

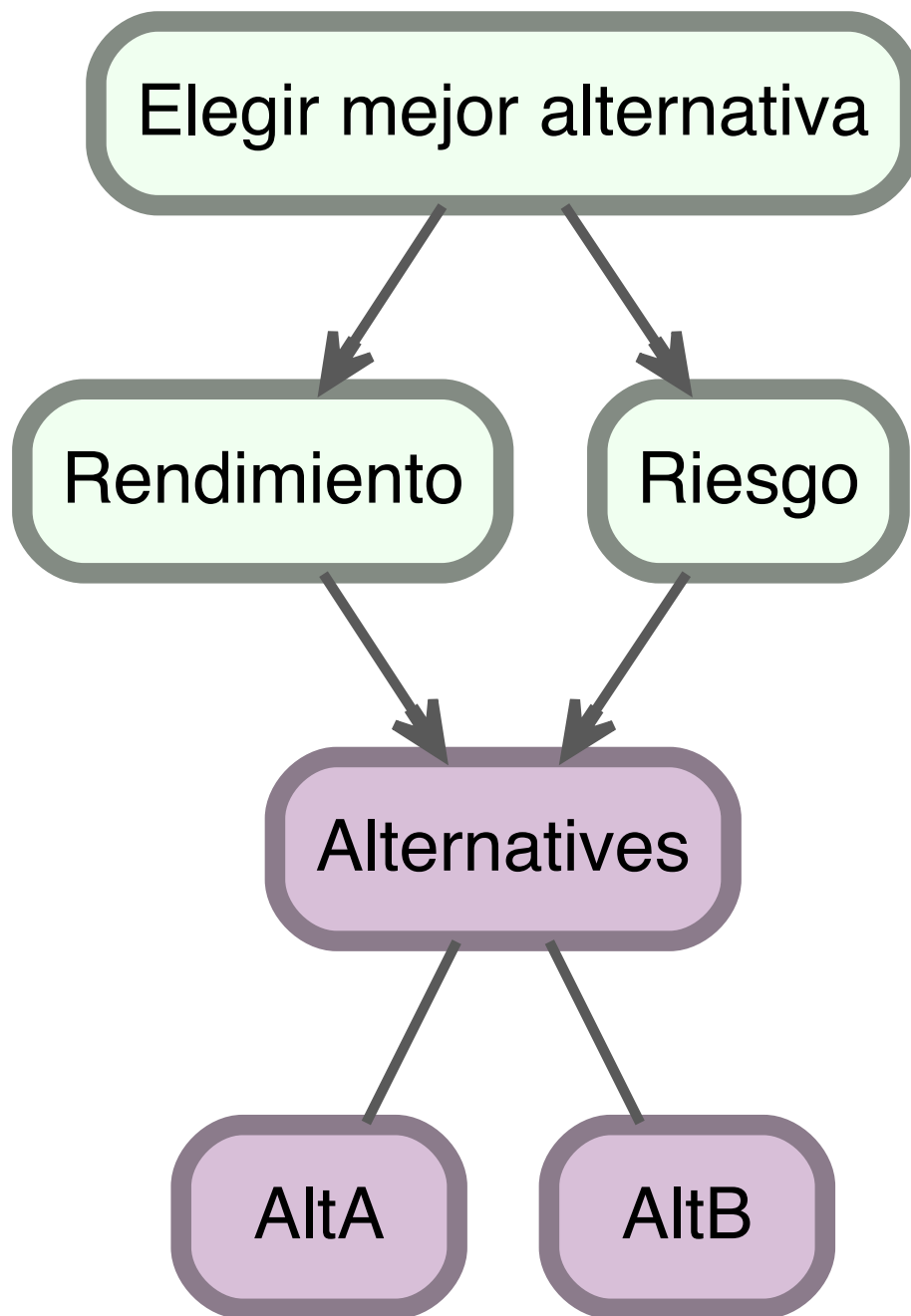


Tabla solución (contribución total)

```
export_formattable(AnalyzeTable(datos04), file = "tablaahp104.png")
```

	Weight	AltA	AltB	Inconsistency
Elegir mejor alternativa	100.0%	61.1%	38.9%	0.0%
Rendimiento	66.7%	50.0%	16.7%	0.0%
Riesgo	33.3%	11.1%	22.2%	0.0%

La mejor decisión es la alternativa A (peso global del 61,1%)

Tabla solución (pesos locales)

```
t2 = AnalyzeTable(datos04, variable = "priority")
export_formattable(t2, file = "tablaahp204.png")
```

	Priority	AltA	AltB	Inconsistency
Elegir mejor alternativa	100.0%			0.0%
Rendimiento	66.7%	75.0%	25.0%	0.0%
Riesgo	33.3%	33.3%	66.7%	0.0%

EJERCICIO 5

Funciones de clase

Introducción datos

```
#matriz criterios
tab1 <- multicriterio.crea.matrizvaloraciones_mej(c(1/3,1/4,2), numalternativas = 3,
v.nombres.alternativas = c("Liderazgo",
"Habilidad personal", "Hab.

#matriz liderazgo
tab2 <- multicriterio.crea.matrizvaloraciones_mej(c(4), numalternativas = 2,
v.nombres.alternativas = c("A","B"))

#matriz habilidad personal
tab3 <- multicriterio.crea.matrizvaloraciones_mej(c(3), numalternativas = 2,
v.nombres.alternativas = c("A","B"))

#matriz habilidad gestión
tab4 <- multicriterio.crea.matrizvaloraciones_mej(c(2), numalternativas = 2,
v.nombres.alternativas = c("A","B"))
```

Cálculo pesos locales

Método mayor autovalor

```
pes1 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab1)
pes2 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab2)
pes3 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab3)
pes4 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab4)
```

Cálculo pesos globales

```
tab05 <- multicriterio.metodoAHP.pesosglobales_entabla(pes1$valoraciones.ahp,
rbind(pes2$valoraciones.ahp,
pes3$valoraciones.ahp,
pes4$valoraciones.ahp))
```

	Liderazgo	Habilidad personal	Habilidad Gestión	Ponderadores Globales
A	0.800000	0.750000	0.666667	0.7263353
B	0.200000	0.250000	0.333333	0.2736647
Ponder.Criterios	0.124306	0.5171336	0.3585604	NA

La mejor decisión es la alternativa A (peso global del 72,63%)

```
which.max(tab05[,1])
```

```
## A
## 1
```

```
which.max(tab05[,2])
```

```
## A
## 1
```

```
which.max(tab05[,3])
```

```
## A
## 1
```

Para el liderazgo la mejor alternativa es la A

Para la habilidad personal la mejor alternativa es la A

Para la habilidad gestión la mejor alternativa es la A

Además del método de mayor autovalor, tambien tenemos:

Método de media geométrica

```
pes1 <- multicriterio.metodoAHP.variante2.mediageometrica(tab1)
pes2 <- multicriterio.metodoAHP.variante2.mediageometrica(tab2)
pes3 <- multicriterio.metodoAHP.variante2.mediageometrica(tab3)
pes4 <- multicriterio.metodoAHP.variante2.mediageometrica(tab4)
```

Método básico

```
pes1 <- multicriterio.metodoAHP.variante3.basico(tab1)
pes2 <- multicriterio.metodoAHP.variante3.basico(tab2)
pes3 <- multicriterio.metodoAHP.variante3.basico(tab3)
pes4 <- multicriterio.metodoAHP.variante3.basico(tab4)
```

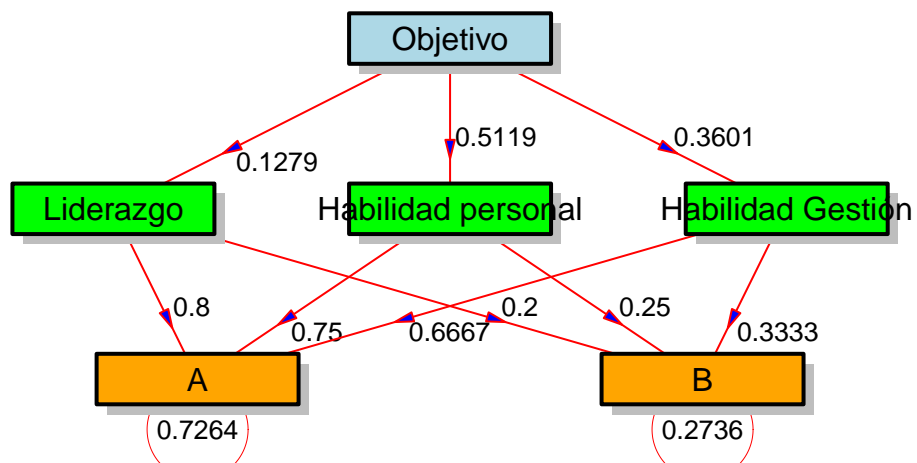
Diagrama Jerarquias

```

num.alt <- 2
num.crt <- 3
Xmatriznivel2_05 <- array(NA, dim = c(num.alt, num.alt, num.crt))
Xmatriznivel2_05[, , 1] <- tab2
Xmatriznivel2_05[, , 2] <- tab3
Xmatriznivel2_05[, , 3] <- tab4
dimnames(Xmatriznivel2_05)[[1]] <- c("A", "B")
dimnames(Xmatriznivel2_05)[[2]] <- c("A", "B")
dimnames(Xmatriznivel2_05)[[3]] <- c("Liderazgo", "Habilidad personal", "Habilidad gestión")
multicriterio.metodoahp.diagrama(tab1, Xmatriznivel2_05)

```

Estructura Jerárquica (AHP)



La mejor decisión es la alternativa A (peso global del 72,64%)

Método ahp

```

library(ahp)
datos05 = Load("problema5.ahp")
Calculate(datos05)

```

```

Visualize(datos05)

```

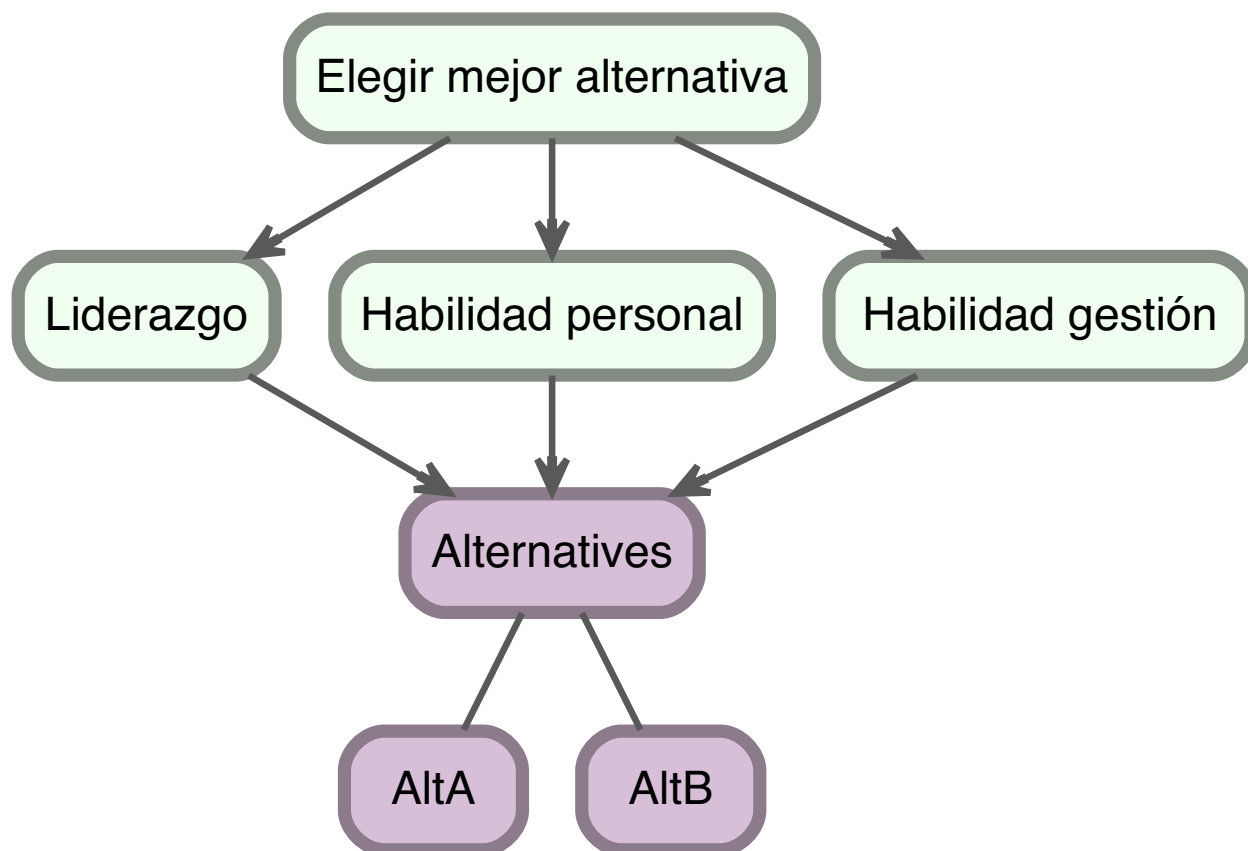



Tabla solución (contribución total)

```
export_formattable(AnalyzeTable(datos05), file = "tablaahp105.png")
```

	Weight	AltA	AltB	Inconsistency
Elegir mejor alternativa	100.0%	72.6%	27.4%	10.3%
Habilidad personal	51.7%	38.8%	12.9%	0.0%
Habilidad gestión	35.9%	23.9%	12.0%	0.0%
Liderazgo	12.4%	9.9%	2.5%	0.0%

La mejor decisión es la alternativa A (peso global del 72,6%)

Tabla solución (pesos locales)

```
t2 = AnalyzeTable(datos05, variable = "priority")
export_formattable(t2, file = "tablaahp205.png")
```

	Priority	AltA	AltB	Inconsistency
Elegir mejor alternativa	100.0%			10.3%
Habilidad personal	51.7%	75.0%	25.0%	0.0%
Habilidad gestión	35.9%	66.7%	33.3%	0.0%
Liderazgo	12.4%	80.0%	20.0%	0.0%

EJERCICIO 6

Funciones de clase

Introducción datos

```
#matriz criterios
n.criterios = c("Costo", "Confiabilidad", "Plazo Entrega")
tab1 <- multicriterio.crea.matrizvaloraciones_mej(c(7,9,3), numalternativas = 3,
v.nombres.alternativas = n.criterios)

#matriz costo
n.alternativas <- c("A","B","C")
tab2 <- multicriterio.crea.matrizvaloraciones_mej(c(1/3,6,8), numalternativas = 3,
v.nombres.alternativas = n.alternativas)

#matriz confiabilidad
tab3 <- multicriterio.crea.matrizvaloraciones_mej(c(6,2,1/3), numalternativas = 3,
v.nombres.alternativas = n.alternativas)

#matriz plazo entrega
tab4 <- multicriterio.crea.matrizvaloraciones_mej(c(8,1,1/8), numalternativas = 3,
v.nombres.alternativas = n.alternativas)
```

Método 1 (mayor autovalor)

Cálculo pesos locales

Método mayor autovalor

```
pes1 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab1)
pes2 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab2)
pes3 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab3)
pes4 <- multicriterio.metodoAHP.variantel1.autovectormayorautovalor(tab4)
```

Cálculo pesos globales

```
tab06 <- multicriterio.metodoAHP.pesosglobales_entabla(pes1$valoraciones.ahp,
rbind(pes2$valoraciones.ahp,
pes3$valoraciones.ahp,
pes4$valoraciones.ahp))
```

	Costo	Confiabilidad	Plazo Entrega	Ponderadores Globales
A	0.2850770	0.6000000	0.4705882	0.3441478
B	0.6526635	0.1000000	0.0588235	0.5313479
C	0.0622594	0.3000000	0.4705882	0.1245043
Ponder.Criterios	0.7853912	0.1488151	0.0657937	NA

La mejor decisión es la alternativa B (peso global del 53,1%)

Además del método de mayor autovalor, también tenemos:

Método de media geométrica

```
pes1 <- multicriterio.metodoAHP.variante2.mediageometrica(tab1)
pes2 <- multicriterio.metodoAHP.variante2.mediageometrica(tab2)
pes3 <- multicriterio.metodoAHP.variante2.mediageometrica(tab3)
pes4 <- multicriterio.metodoAHP.variante2.mediageometrica(tab4)
```

Método básico

```
pes1 <- multicriterio.metodoAHP.variante3.basico(tab1)
pes2 <- multicriterio.metodoAHP.variante3.basico(tab2)
pes3 <- multicriterio.metodoAHP.variante3.basico(tab3)
pes4 <- multicriterio.metodoAHP.variante3.basico(tab4)
```

Método 2 (completo)

```
num.alt = 3
num.crt = 3
Xarray_nivel2 = array(NA, dim=c(num.alt, num.alt, num.crt))
Xarray_nivel2[, ,1] = tab2
Xarray_nivel2[, ,2] = tab3
Xarray_nivel2[, ,3] = tab4
pg06com = multicriterio.metodoAHP.variante3.completo(tab1,Xarray_nivel2)
tab06_com <- pg06com$pesos.globales.entabla
```

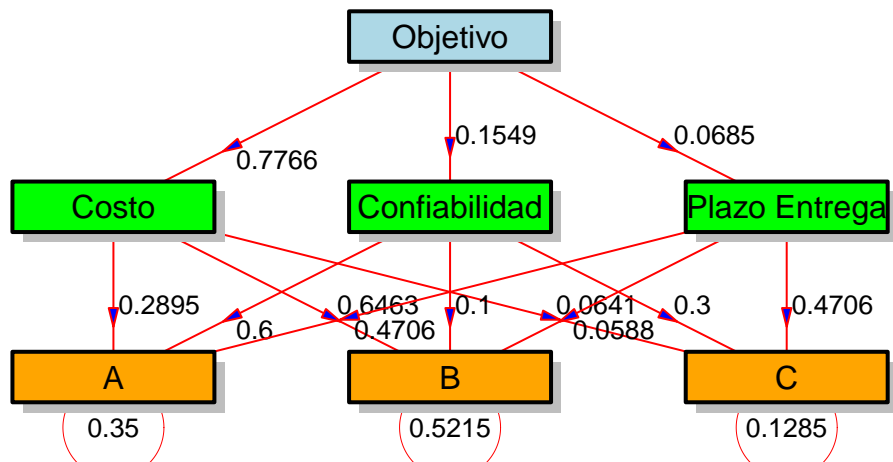
	Costo	Confiabilidad	Plazo Entrega	Ponderadores Globales
	0.2895238	0.6000000	0.4705882	0.3500206
	0.6463492	0.1000000	0.0588235	0.5214694
	0.0641270	0.3000000	0.4705882	0.1285099
Ponder.Criterios	0.7765920	0.1548978	0.0685102	NA

La mejor decisión es la alternativa B (peso global del 52,1%)

Diagrama Jerarquias

```
num.alt <- 3
num.crt <- 3
Xmatriznivel2_06 <- array(NA, dim = c(num.alt, num.alt, num.crt))
Xmatriznivel2_06[, ,1] <- tab2
Xmatriznivel2_06[, ,2] <- tab3
Xmatriznivel2_06[, ,3] <- tab4
dimnames(Xmatriznivel2_06)[[1]] <- n.alternativas
multicriterio.metodoahp.diagrama(tab1, Xmatriznivel2_06)
```

Estructura Jerárquica (AHP)



La mejor decisión es la alternativa B (peso global del 53,15%)

Estudio de la inconsistencia con funciones R de clase.

Al ser matrices "3x3" hay que estudiar la inconsistencia

```
(inconsistencia1 <- multicriterio.metodoAHP.coef.inconsistencia(tab1))
```

```
## $lambda
## [1] 3.0803
##
## $m
## [1] 3
##
## $CI.coef.inconsistencia
## [1] 0.04014992
##
## $CA.aleatorio
## [1] 0.58
##
## $RI.coef.inconsistencia
## [1] 0.069224
##
## $mensaje
## [1] "Consistencia aceptable"
```

```
inconsistencia2 <- multicriterio.metodoAHP.coef.inconsistencia(tab2)
c(inconsistencia2$mensaje,round(inconsistencia2$RI.coef.inconsistencia,4))
```

```
## [1] "Consistencia aceptable" "0.0634"
```

```
inconsistencia3 <- multicriterio.metodoAHP.coef.inconsistencia(tab3)
c(inconsistencia3$mensaje,round(inconsistencia3$RI.coef.inconsistencia,4))
```

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

```
inconsistencia4 <- multicriterio.metodoAHP.coef.inconsistencia(tab4)
c(inconsistencia4$mensaje,round(inconsistencia4$RI.coef.inconsistencia,4))
```

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

Método ahp

```
library(ahp)
datos06 = Load("problema6.ahp")
Calculate(datos06)
```

```
Visualize(datos06)
```

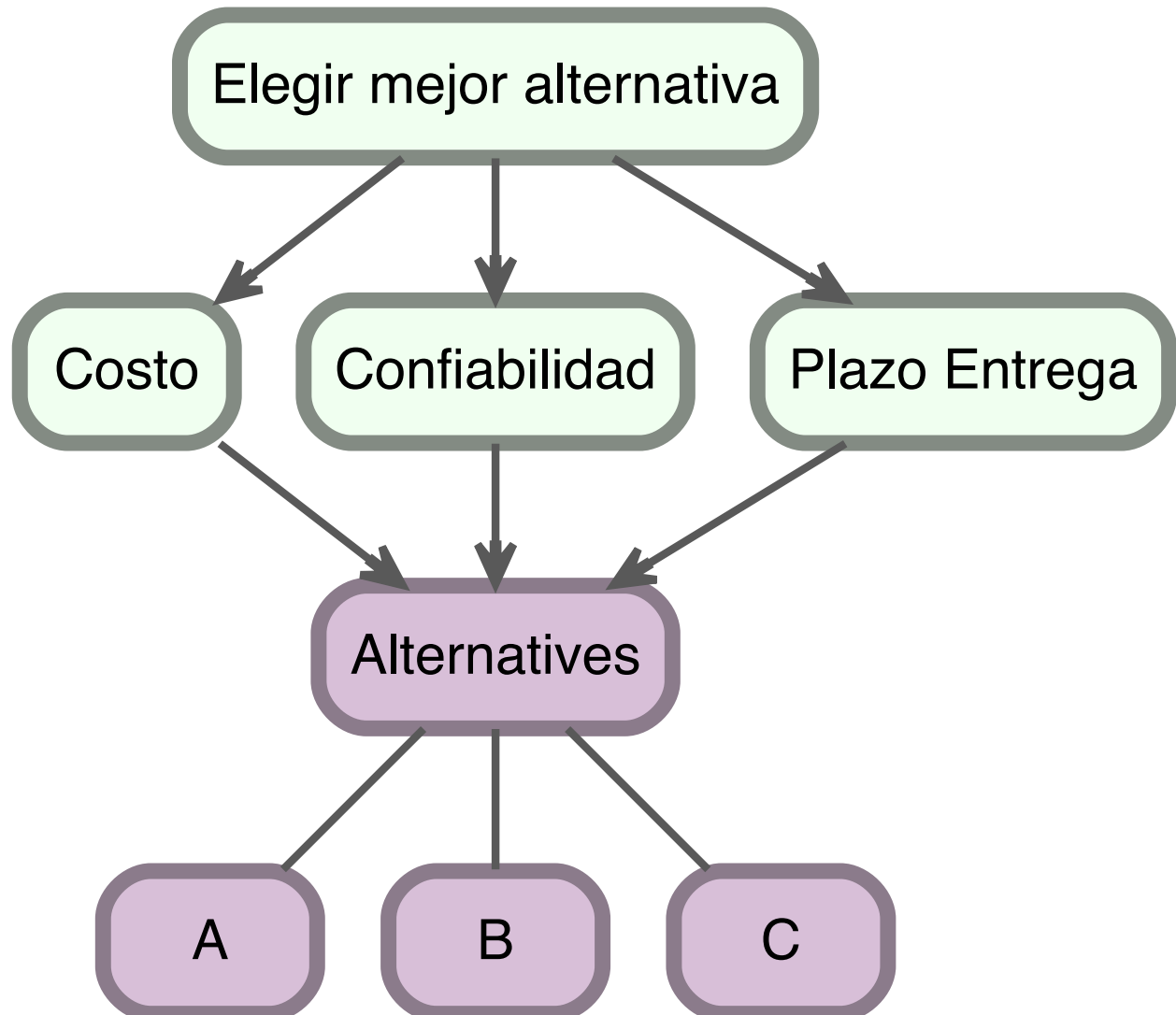


Tabla solución (contribución total)

```
export_formattable(AnalyzeTable(datos06), file = "tablaahp106.png")
```

	Weight	B	A	C	Inconsistency
Elegir mejor alternativa	100.0%	53.1%	34.4%	12.5%	7.7%
Costo	78.5%	51.3%	22.4%	4.9%	7.0%
Confiabilidad	14.9%	1.5%	8.9%	4.5%	0.0%
Plazo Entrega	6.6%	0.4%	3.1%	3.1%	0.0%

La mejor decisión es la alternativa B (peso global del 53.1%)

Tabla solución (pesos locales)

```
t2 = AnalyzeTable(datos06, variable = "priority")
export_formattable(t2, file = "tablaahp206.png")
```

	Priority	B	A	C	Inconsistency
Elegir mejor alternativa	100.0%				7.7%
Costo	78.5%	65.3%	28.5%	6.2%	7.0%
Confiabilidad	14.9%	10.0%	60.0%	30.0%	0.0%
Plazo Entrega	6.6%	5.9%	47.1%	47.1%	0.0%

EJERCICIO 7

Iteración 1. Introducir datos y resolver

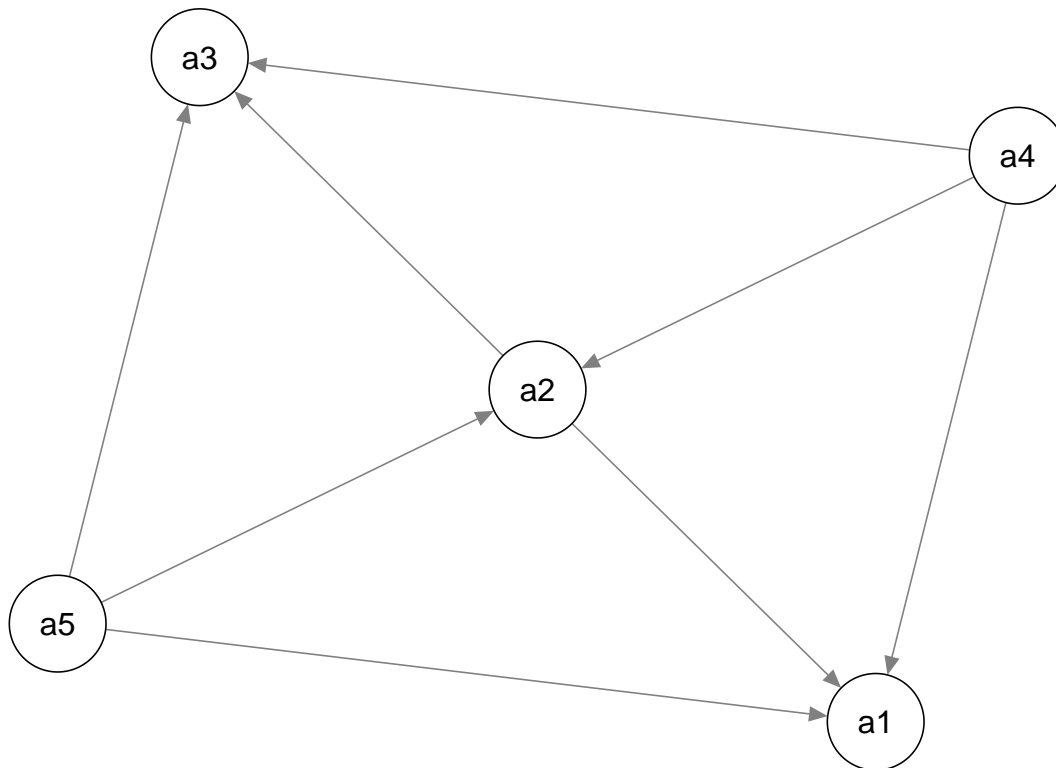
```
tab07 <- multicriterio.crea.matrizdecision(c(100,15,7,40,-50,
      200,25,7,60,-200,
      100,20,4,25,-25,
      200,30,20,70,-350,
      250,25,15,100,-500),
      numalternativas = 5,
      numcriterios = 5)
```

	C1	C2	C3	C4	C5
a1	100	15	7	40	-50
a2	200	25	7	60	-200
a3	100	20	4	25	-25
a4	200	30	20	70	-350
a5	250	25	15	100	-500

```
sal07 <- multicriterio.metodoELECTRE_I(tab07,
      pesos.criterios = c(0.25,0.25,0.2,0.2,0.2),
      nivel.concordancia.minimo.alpha = 0.7,
      no.se.compensan = c(60, Inf, 4, Inf, Inf),
```

```
que.alternativas = T)
```

```
qgraph::qgraph(sal07$relacion.dominante)
```



```
sal07$nucleo_aprox
```

```
## a4 a5  
## 4 5
```

Iteración 2 y 3. Se reducen alternativas y/o alpha

Para intentar quedarse con una única alternativa óptima:

- reducir el grafo a las alternativas en el núcleo y/o
- reducir el valor de alpha [0.5, 1)

```
sal07_2 <- multicriterio.metodoELECTRE_I(tab07,  
  pesos.criterios = c(0.25,0.25,0.2,0.2,0.2),  
  nivel.concordancia.minimo.alpha = 0.7,  
  no.se.compensan = c(60, Inf, 4, Inf, Inf),  
  que.alternativas = c(4,5))
```

```
qgraph::qgraph(sal07_2$relacion.dominante)
```

a4

a5

```
sal07_2$nucleo_aprox
```

```
## a4 a5  
## 1 2
```

```
sal07_3 <- multicriterio.metodoELECTRE_I(tab07,  
                                          pesos.criterios = c(0.25,0.25,0.2,0.2,0.2),  
                                          nivel.concordancia.minimo.alpha = 0.55,  
                                          no.se.compensan = c(60, Inf, 4, Inf, Inf),  
                                          que.alternativas = c(4,5))  
qgraph::qgraph(sal07_3$relacion.dominante)
```




```
sal07_3$nucleo_aprox
```

```
## a4
## 1
```

Obtenemos un único núcleo

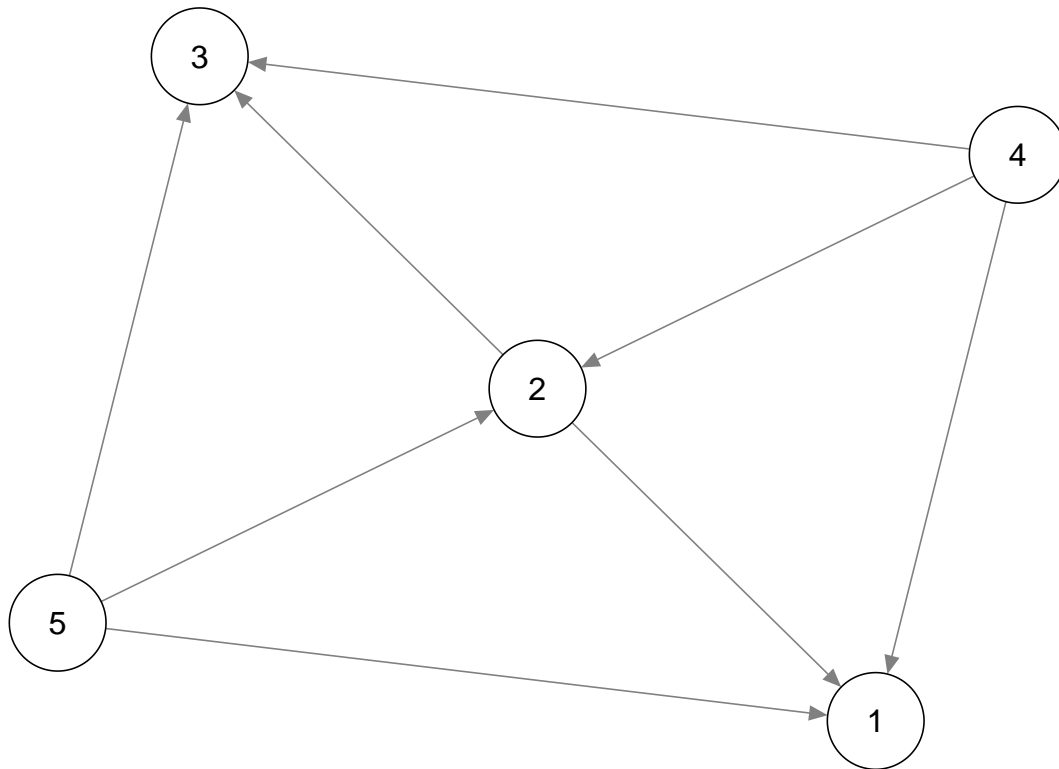
Método ELECTRE I

```
elec_07 <- func_ELECTRE_Completo(sal07)
elec_07$Grafo
```

```
## De A
## 1 2 1
## 2 2 3
## 3 4 1
## 4 4 2
## 5 4 3
## 6 5 1
## 7 5 2
## 8 5 3
```

Tenemos que: a_2Sa_1 , a_2Sa_3 , a_4Sa_1 , a_4Sa_3 , a_5Sa_1 , a_5Sa_3

```
qgraph::qgraph(elec_07$Grafo)
```



```
elec_07$Nucleo
```

```
## a4 a5
## 4 5
```

Método PROMETHEE

```

pesos.criterios = c(0.25/1.1,0.25/1.1,0.2/1.1,0.2/1.1,0.2/1.1) #se divide entre la suma de los pesos to
tab.fpref = matrix(c(1,0,0,0,
                    1,0,0,0,
                    1,0,0,0,
                    1,0,0,0,
                    1,0,0,0),ncol=4,byrow=T)

```

PROMETHEE I

```

tab.Pthee.i = multicriterio.metodo.promethee_i(tab07,pesos.criterios,tab.fpref)
tab.Pthee.i

```

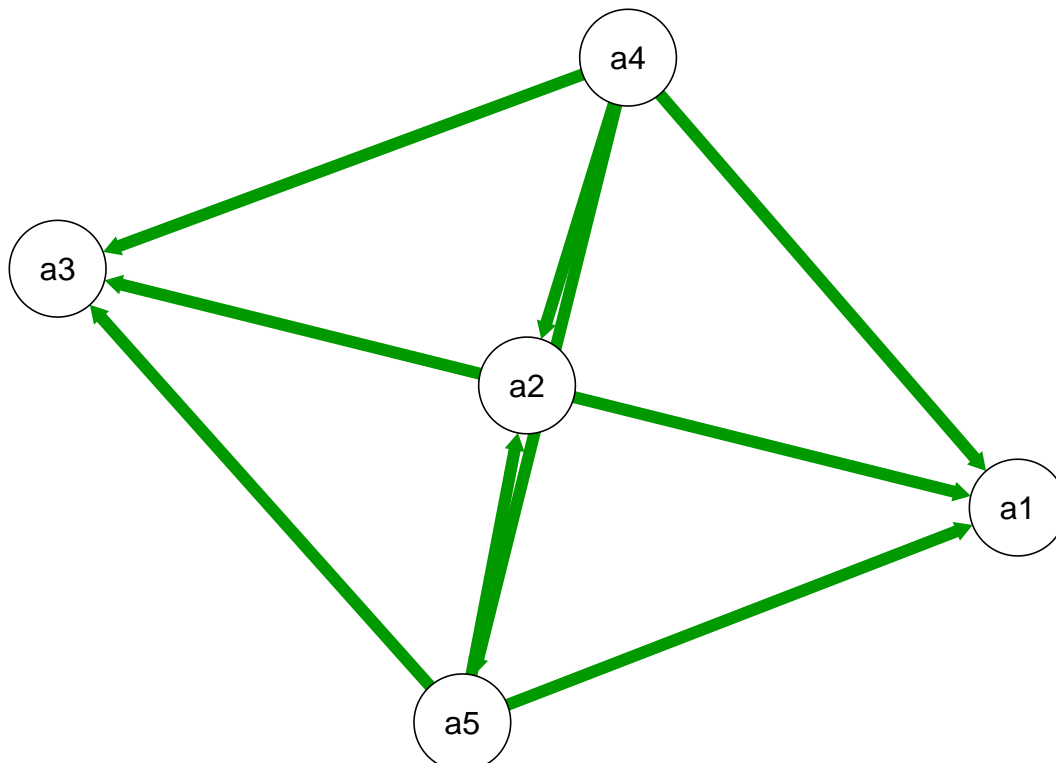
```
## $tabla.indices
```

```
##           a1           a2           a3           a4           a5
## a1 0.0000000 0.1818182 0.3636364 0.1818182 0.1818182
## a2 0.6363636 0.0000000 0.8181818 0.1818182 0.1818182
## a3 0.4090909 0.1818182 0.0000000 0.1818182 0.1818182
## a4 0.8181818 0.5909091 0.8181818 0.0000000 0.5909091
## a5 0.8181818 0.5909091 0.8181818 0.4090909 0.0000000
##
## $vflujos.ent
##           a1           a2           a3           a4           a5
## 0.9090909 1.8181818 0.9545455 2.8181818 2.6363636
##
## $vflujos.sal
##           a1           a2           a3           a4           a5
## 2.6818182 1.5454545 2.8181818 0.9545455 1.1363636
##
## $tablarelacionsupera
##    a1 a2 a3 a4 a5
## a1 0.5 0.0 0.0 0.0 0.0
## a2 1.0 0.5 1.0 0.0 0.0
## a3 0.0 0.0 0.0 0.5 0.0
## a4 1.0 1.0 1.0 0.5 1.0
## a5 1.0 1.0 1.0 0.0 0.5
```

```
require ("qgraph")
```

```
## Loading required package: qgraph
```

```
qgraph(tab.Pthee.i$tablarelacionsupera)
```

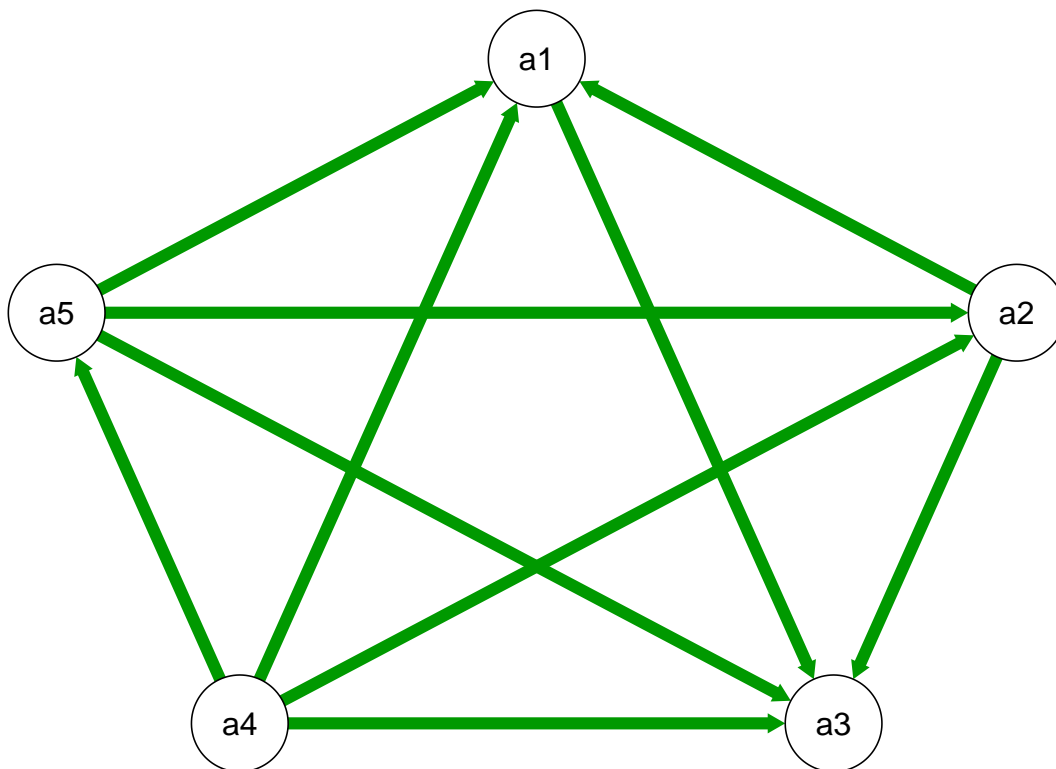


PROMETHEE II

```
tab.Pthee.ii = multicriterio.metodo.promethee_ii(tab07,pesos.criterios,tab.fpref)
tab.Pthee.ii
```

```
## $tabla.indices
##      a1      a2      a3      a4      a5
## a1 0.0000000 0.1818182 0.3636364 0.1818182 0.1818182
## a2 0.6363636 0.0000000 0.8181818 0.1818182 0.1818182
## a3 0.4090909 0.1818182 0.0000000 0.1818182 0.1818182
## a4 0.8181818 0.5909091 0.8181818 0.0000000 0.5909091
## a5 0.8181818 0.5909091 0.8181818 0.4090909 0.0000000
##
## $vflujos.netos
##      a1      a2      a3      a4      a5
## -1.7727273 0.2727273 -1.8636364 1.8636364 1.5000000
##
## $tablarelacionsupera
##      a1 a2 a3 a4 a5
## a1 0.5 0.0 1.0 0.0 0.0
## a2 1.0 0.5 1.0 0.0 0.0
## a3 0.0 0.0 0.5 0.0 0.0
## a4 1.0 1.0 1.0 0.5 1.0
## a5 1.0 1.0 1.0 0.0 0.5
```

```
qgraph(tab.Pthee.ii$tablarelacionsupera)
```

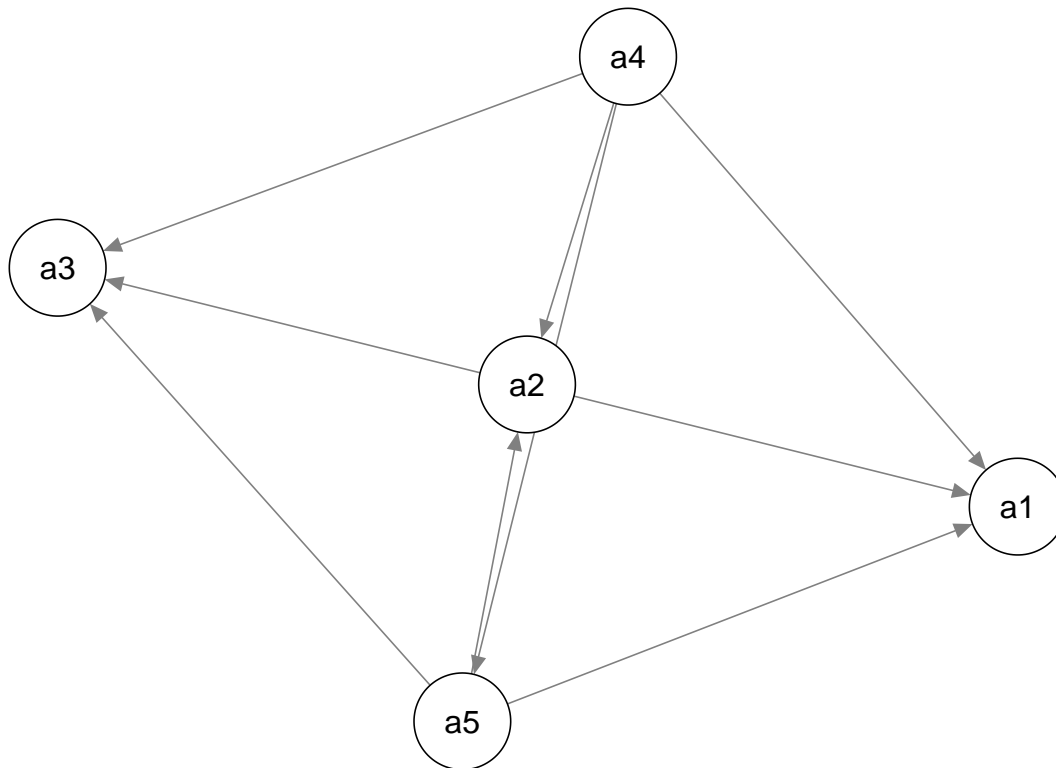


PROMETHEE I (medias)

```
tab.Pthee.i_med = multicriterio.metodo.promethee_i_med(tab07,pesos.criterios,tab.fpref)
tab.Pthee.i_med
```

```
## $tabla.indices
##      a1      a2      a3      a4      a5
## a1 0.0000000 0.1818182 0.3636364 0.1818182 0.1818182
## a2 0.6363636 0.0000000 0.8181818 0.1818182 0.1818182
## a3 0.4090909 0.1818182 0.0000000 0.1818182 0.1818182
## a4 0.8181818 0.5909091 0.8181818 0.0000000 0.5909091
## a5 0.8181818 0.5909091 0.8181818 0.4090909 0.0000000
##
## $vflujos.ent
##      a1      a2      a3      a4      a5
## 0.2272727 0.4545455 0.2386364 0.7045455 0.6590909
##
## $vflujos.sal
##      a1      a2      a3      a4      a5
## 0.6704545 0.3863636 0.7045455 0.2386364 0.2840909
##
## $tablarelacionsupera
##      a1 a2 a3 a4 a5
## a1  0  0  0  0  0
## a2  1  0  1  0  0
## a3  0  0  0  0  0
## a4  1  1  1  0  1
## a5  1  1  1  0  0
```

```
qgraph (tab.Pthee.i_med$tablarelacionsupera)
```



PROMETHEE II (medias)

```

tab.Pthee.ii_med = multicriterio.metodo.promethee_ii_med(tab07,pesos.criterios,tab.fpref)
tab.Pthee.ii_med

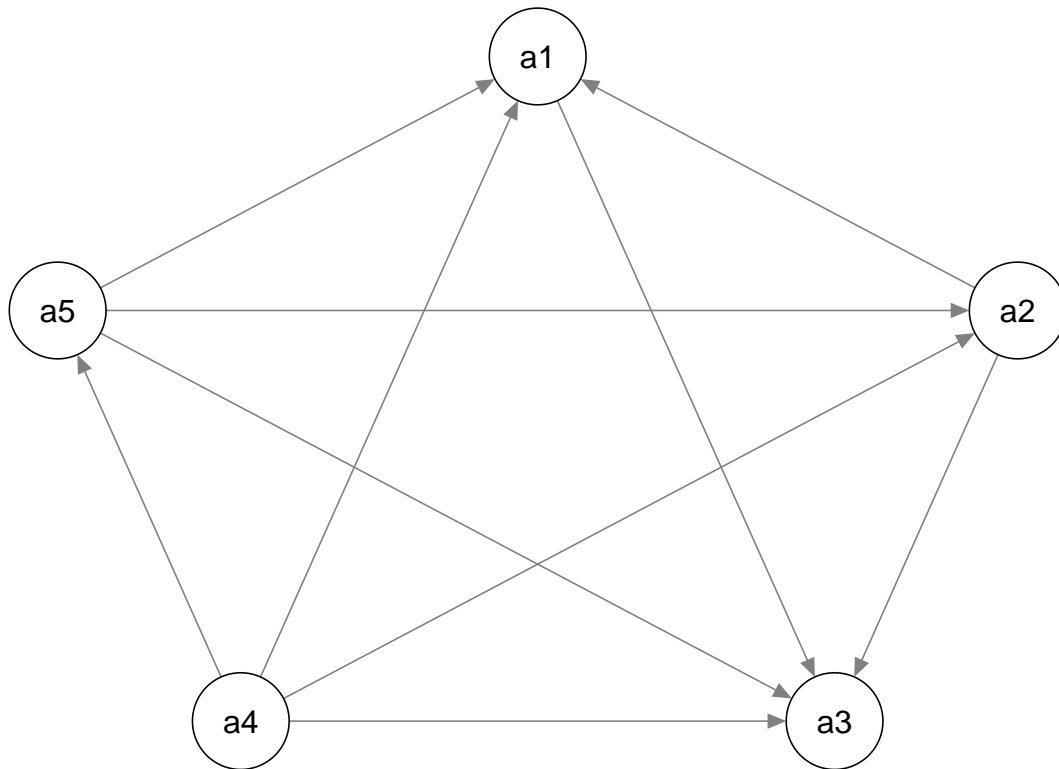
```

```

## $tabla.indices
##      a1      a2      a3      a4      a5
## a1 0.0000000 0.1818182 0.3636364 0.1818182 0.1818182
## a2 0.6363636 0.0000000 0.8181818 0.1818182 0.1818182
## a3 0.4090909 0.1818182 0.0000000 0.1818182 0.1818182
## a4 0.8181818 0.5909091 0.8181818 0.0000000 0.5909091
## a5 0.8181818 0.5909091 0.8181818 0.4090909 0.0000000
##
## $vflujos.netos
##      a1      a2      a3      a4      a5
## -0.44318182 0.06818182 -0.46590909 0.46590909 0.37500000
##
## $tablarelacionsupera
##      a1 a2 a3 a4 a5
## a1  0  0  1  0  0
## a2  1  0  1  0  0
## a3  0  0  0  0  0
## a4  1  1  1  0  1
## a5  1  1  1  0  0

```

```
qgraph (tab.Pthee.ii_med$tablarelacionsupera)
```



Ordenación final alternativas Mét. Promethee II (medias)

```
order(tab.Pthee.ii_med$vflujos.netos,decreasing = T)
```

```
## [1] 4 5 2 1 3
```

Comparativa Promethee II: sin medias y con medias

```
order(tab.Pthee.ii$vflujos.netos,decreasing = T)
```

```
## [1] 4 5 2 1 3
```

```
order(tab.Pthee.ii_med$vflujos.netos,decreasing = T)
```

```
## [1] 4 5 2 1 3
```

Resolución con Promethee Windows

```

res = multicriterio.metodo.promethee_windows(tab07, tab.fpref, pesos.criterios)
res = multicriterio.metodo.promethee_windows (tab07, tab.fpref, pesos.criterios,
fminmax = c("max", "max", "max", "max", "min"))

res02 = multicriterio.metodo.promethee_windows_kableExtra(res)
res02$tabEscenario

```

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5
Preferencias					
Min/Max	max	max	max	max	min
Pesos	0.227272727272727	0.227272727272727	0.181818181818182	0.181818181818182	0.181818181818182
Funciones Preferencias	Usual (1)	Usual (1)	Usual (1)	Usual (1)	Usual (1)
Q: Indiferencia	0	0	0	0	0
P: Preferencia	0	0	0	0	0
S: Gausiano	0	0	0	0	0
Estadísticas					
Minimo	100	15	4	25	25
Maximo	250	30	20	100	500
Media	170	23	10.6	59	225
Desviacion Tipica	60	5.1	5.95	25.77	180.28
Evaluaciones					
a1	100	15	7	40	50
a2	200	25	7	60	200
a3	100	20	4	25	25
a4	200	30	20	70	350
a5	250	25	15	100	500

```
res02$tabAcciones
```

	Rango	Phi	Phi.mas	Phi.menos
a4	1	0.4659	0.7045	0.2386
a5	2	0.3750	0.6591	0.2841
a2	3	0.0682	0.4545	0.3864
a1	4	-0.4432	0.2273	0.6705
a3	5	-0.4659	0.2386	0.7045

```
rownames(res$Acciones)
```

```
## [1] "a4" "a5" "a2" "a1" "a3"
```

EJERCICIO 8

```

tab08 <- multicriterio.crea.matrizdecision(c(-80,90,-6,-5.4,-8,5,
-65,58,-2,-9.7,-1,1,
-83,60,-4,-7.2,-4,7,
-40,80,-10,-7.5,-7,10,

```



```

-52,72,-6,-2.0,-3,8,
-94,96,-7-3.6,-5,6),
numalternativas=6,
numcriterios=6,
v.nombresalt=c('A1','A2','A3','A4','A5','A6'),
v.nombrescri=c('f1', 'f2','f3','f4', 'f5','f6'))

```

```

## Warning in matrix(vector_matporfilas, nrow = numalternativas, ncol =
## numcriterios, : data length [34] is not a sub-multiple or multiple of the
## number of rows [6]

```

```

pesos.criterios = c(1/6,1/6,1/6,1/6,1/6,1/6)
tab.fpref = matrix (c(2,10,1,0,
                      3,0,30,0,
                      5,0.5,5,0,
                      4,1,6,0,
                      1,0,1,0,
                      6,0,1,5),ncol=4,byrow=T)
tab.fpref

```

```

##      [,1] [,2] [,3] [,4]
## [1,]    2 10.0    1    0
## [2,]    3  0.0   30    0
## [3,]    5  0.5    5    0
## [4,]    4  1.0    6    0
## [5,]    1  0.0    1    0
## [6,]    6  0.0    1    5

```

PROMETHEE I

```

tab.Pthee.i = multicriterio.metodo.promethee_i(tab08,pesos.criterios,tab.fpref)
tab.Pthee.i

```

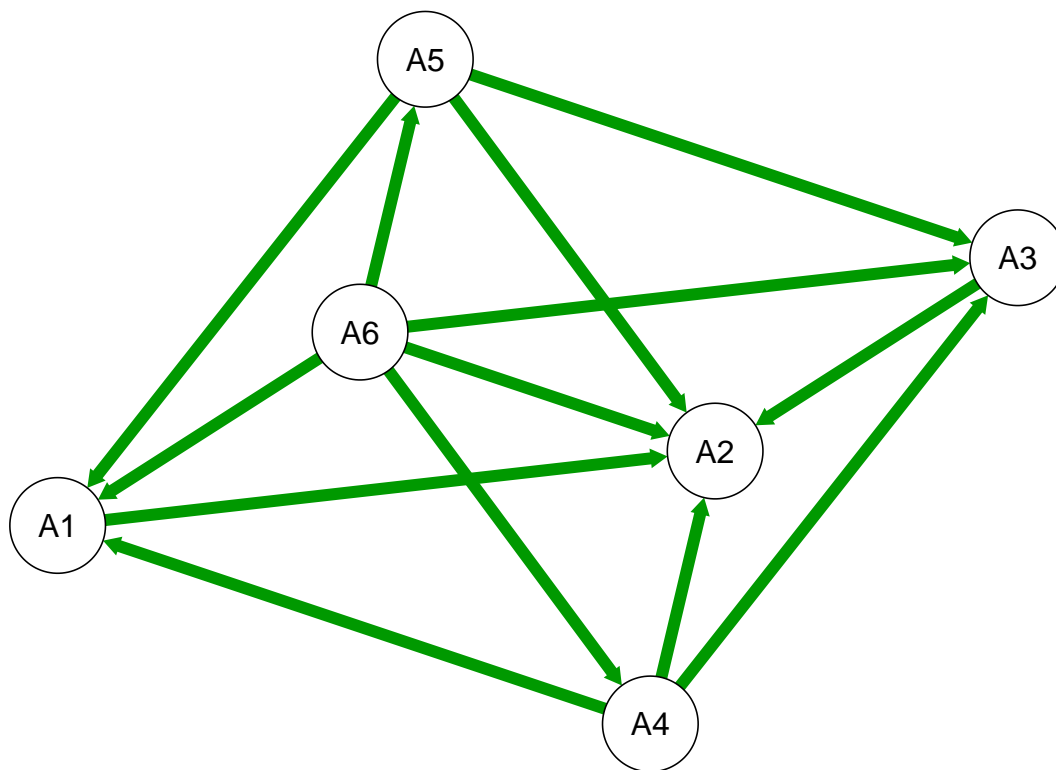
```

## $tabla.indices
##      A1      A2      A3      A4      A5      A6
## A1 0.0000000 0.6404851 0.21481481 0.2259259 0.3333333 0.3333333
## A2 0.3333333 0.0000000 0.09444444 0.2111111 0.2574020 0.2144444
## A3 0.5833327 0.4074074 0.00000000 0.2739775 0.1777778 0.2033333
## A4 0.4943254 0.5629630 0.33333333 0.0000000 0.3333333 0.1700000
## A5 0.5240741 0.5000000 0.50000000 0.2722222 0.0000000 0.2848148
## A6 0.5000000 0.6555556 0.56296296 0.5740741 0.5000000 0.0000000
##
## $vflujos.ent
##      A1      A2      A3      A4      A5      A6
## 1.747893 1.110735 1.645829 1.893955 2.081111 2.792593
##
## $vflujos.sal
##      A1      A2      A3      A4      A5      A6
## 2.435066 2.766411 1.705556 1.557311 1.601846 1.205926
##

```

```
## $tablarelacionsupera
##      A1  A2  A3  A4  A5  A6
## A1 0.5 1.0 0.0 0.0 0.0 0.0
## A2 0.0 0.5 0.0 0.0 0.0 0.0
## A3 0.0 1.0 0.5 0.0 0.0 0.0
## A4 1.0 1.0 1.0 0.5 0.0 0.0
## A5 1.0 1.0 1.0 0.0 0.5 0.0
## A6 1.0 1.0 1.0 1.0 1.0 0.5
```

```
require ("qgraph")
qgraph(tab.Pthee.i$tablarelacionsupera)
```



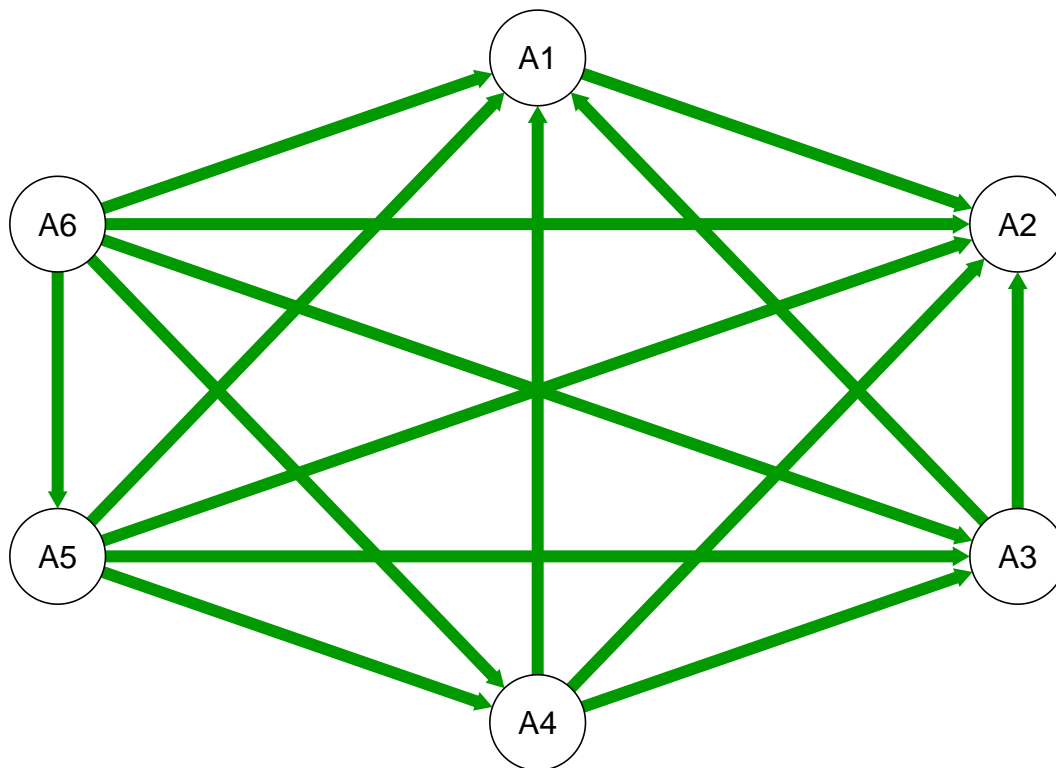
PROMETHEE II

```
tab.Pthee.ii = multicriterio.metodo.promethee_ii(tab08,pesos.criterios,tab.fpref)
tab.Pthee.ii
```

```
## $tabla.indices
##      A1      A2      A3      A4      A5      A6
## A1 0.0000000 0.6404851 0.21481481 0.2259259 0.3333333 0.3333333
## A2 0.3333333 0.0000000 0.09444444 0.2111111 0.2574020 0.2144444
## A3 0.5833327 0.4074074 0.00000000 0.2739775 0.1777778 0.2033333
## A4 0.4943254 0.5629630 0.33333333 0.0000000 0.3333333 0.1700000
## A5 0.5240741 0.5000000 0.50000000 0.2722222 0.0000000 0.2848148
## A6 0.5000000 0.6555556 0.56296296 0.5740741 0.5000000 0.0000000
##
```

```
## $vflujos.netos
##      A1      A2      A3      A4      A5      A6
## -0.68717304 -1.65567574 -0.05972679  0.33664418  0.47926472  1.58666667
##
## $tablarelacionsupera
##      A1  A2  A3  A4  A5  A6
## A1 0.5 1.0 0.0 0.0 0.0 0.0
## A2 0.0 0.5 0.0 0.0 0.0 0.0
## A3 1.0 1.0 0.5 0.0 0.0 0.0
## A4 1.0 1.0 1.0 0.5 0.0 0.0
## A5 1.0 1.0 1.0 1.0 0.5 0.0
## A6 1.0 1.0 1.0 1.0 1.0 0.5
```

```
qgraph(tab.Pthee.ii$tablarelacionsupera)
```



PROMETHEE I (medias)

```
tab.Pthee.i_med = multicriterio.metodo.promethee_i_med(tab08,pesos.criterios,tab.fpref)
tab.Pthee.i_med
```

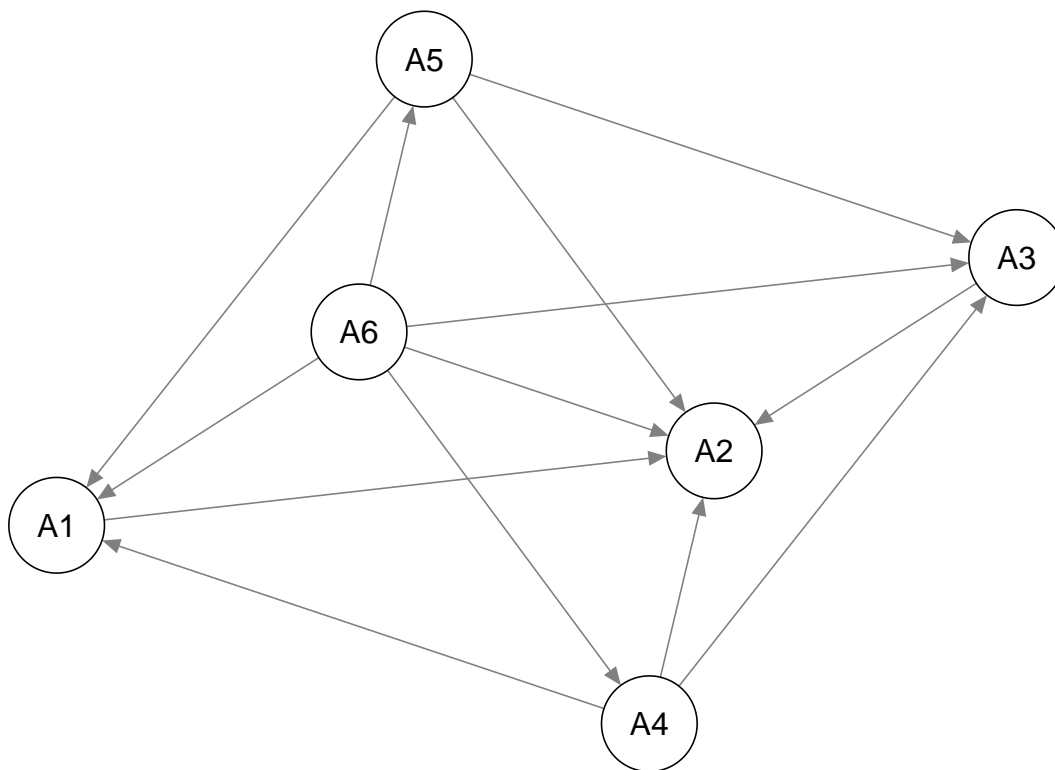
```
## $tabla.indices
##      A1      A2      A3      A4      A5      A6
## A1 0.0000000 0.6404851 0.21481481 0.2259259 0.3333333 0.3333333
## A2 0.3333333 0.0000000 0.09444444 0.2111111 0.2574020 0.2144444
## A3 0.5833327 0.4074074 0.00000000 0.2739775 0.1777778 0.2033333
## A4 0.4943254 0.5629630 0.33333333 0.0000000 0.3333333 0.1700000
```

```

## A5 0.5240741 0.5000000 0.5000000 0.2722222 0.0000000 0.2848148
## A6 0.5000000 0.6555556 0.56296296 0.5740741 0.5000000 0.0000000
##
## $vflujos.ent
##      A1      A2      A3      A4      A5      A6
## 0.3495785 0.2221471 0.3291658 0.3787910 0.4162222 0.5585185
##
## $vflujos.sal
##      A1      A2      A3      A4      A5      A6
## 0.4870131 0.5532822 0.3411111 0.3114622 0.3203693 0.2411852
##
## $tablarelacionsupera
##   A1 A2 A3 A4 A5 A6
## A1  0  1  0  0  0  0
## A2  0  0  0  0  0  0
## A3  0  1  0  0  0  0
## A4  1  1  1  0  0  0
## A5  1  1  1  0  0  0
## A6  1  1  1  1  1  0

```

```
qgraph (tab.Pthee.i_med$tablarelacionsupera)
```



PROMETHEE II (medias)

```

tab.Pthee.ii_med = multicriterio.metodo.promethee_ii_med(tab08,pesos.criterios,tab.fpref)
tab.Pthee.ii_med

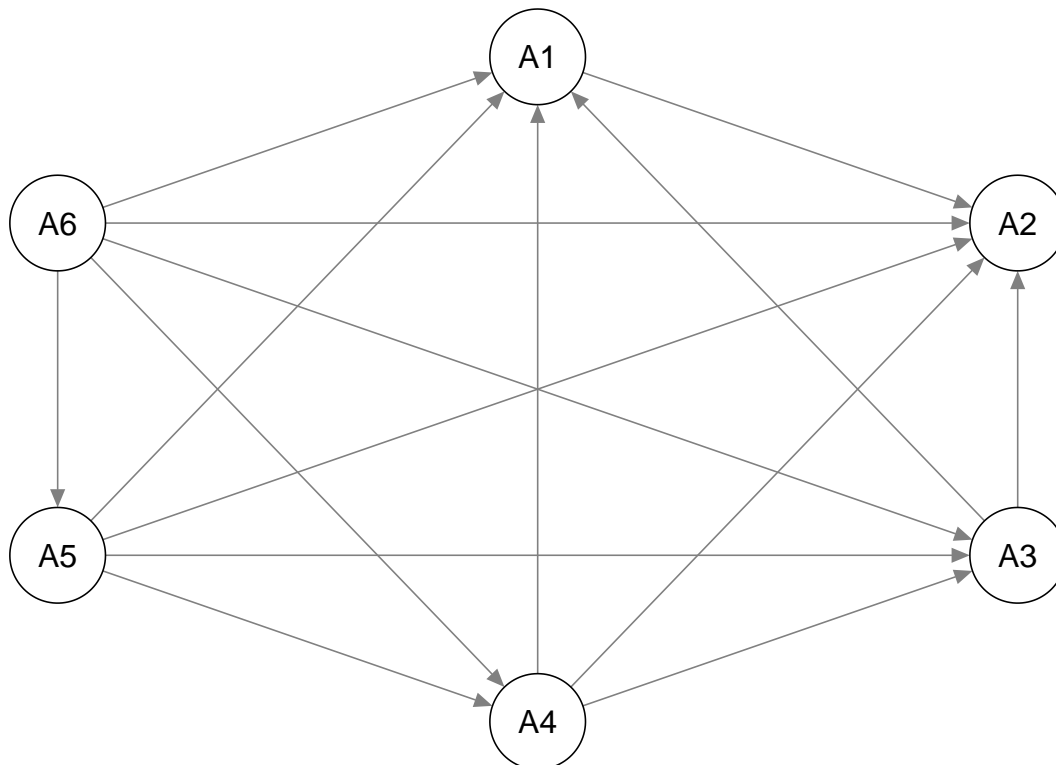
```

```

## $tabla.indices
##      A1      A2      A3      A4      A5      A6
## A1 0.0000000 0.6404851 0.21481481 0.2259259 0.3333333 0.3333333
## A2 0.3333333 0.0000000 0.09444444 0.2111111 0.2574020 0.2144444
## A3 0.5833327 0.4074074 0.00000000 0.2739775 0.1777778 0.2033333
## A4 0.4943254 0.5629630 0.33333333 0.0000000 0.3333333 0.1700000
## A5 0.5240741 0.5000000 0.50000000 0.2722222 0.0000000 0.2848148
## A6 0.5000000 0.6555556 0.56296296 0.5740741 0.5000000 0.0000000
##
## $vflujos.netos
##      A1      A2      A3      A4      A5      A6
## -0.13743461 -0.33113515 -0.01194536 0.06732884 0.09585294 0.31733333
##
## $tablarelacionsupera
##      A1 A2 A3 A4 A5 A6
## A1 0 1 0 0 0 0
## A2 0 0 0 0 0 0
## A3 1 1 0 0 0 0
## A4 1 1 1 0 0 0
## A5 1 1 1 1 0 0
## A6 1 1 1 1 1 0

```

```
qgraph (tab.Pthee.ii_med$tablarelacionsupera)
```



Ordenación final alternativas Mét. Promethee II (medias)

```
order(tab.Pthee.ii_med$vflujos.netos,decreasing = T)
```

```
## [1] 6 5 4 3 1 2
```

Comparativa Promethee II: sin medias y con medias

```
order(tab.Pthee.ii$vflujos.netos,decreasing = T)
```

```
## [1] 6 5 4 3 1 2
```

```
order(tab.Pthee.ii_med$vflujos.netos,decreasing = T)
```

```
## [1] 6 5 4 3 1 2
```

Resolución con Promethee Windows

```
res = multicriterio.metodo.promethee_windows(tab08, tab.fpref, pesos.criterios)
res = multicriterio.metodo.promethee_windows (tab08, tab.fpref, pesos.criterios,
fminmax = c("min", "max", "min", "min","min","max"))

res02 = multicriterio.metodo.promethee_windows_kableExtra(res)
res02$tabEscenario
```

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5
Preferencias					
Min/Max	min	max	min	min	min
Pesos	0.166666666666667	0.166666666666667	0.166666666666667	0.166666666666667	0.166666666666667
Funciones Preferencias	U-shape (2)	V-shape (3)	Linear (5)	Level (4)	Usual (1)
Q: Indiferencia	10	0	0.5	1	0
P: Preferencia	1	30	5	6	1
S: Gausiano	0	0	0	0	0
Estadísticas					
Minimo	-96	-10.6	2	-6	-10
Maximo	80	84	9.7	8	80
Media	-47.67	8.57	6.13	2.83	8.17
Desviacion Tipica	58.5	33.87	2.4	4.6	32.25
Evaluaciones					
A1	80	84	5.4	8	-5
A2	-58	-2	9.7	1	-1
A3	-60	-4	7.2	4	-7
A4	-80	-10	7.5	7	-10
A5	-72	-6	2	3	-8
A6	-96	-10.6	5	-6	80

```
res02$tabAcciones
```

	Rango	Phi	Phi.mas	Phi.menos
A6	1	0.3173	0.5585	0.2412
A5	2	0.0959	0.4162	0.3204
A4	3	0.0673	0.3788	0.3115
A3	4	-0.0119	0.3292	0.3411
A1	5	-0.1374	0.3496	0.4870
A2	6	-0.3311	0.2221	0.5533

```
rownames(res$Acciones)
```

```
## [1] "A6" "A5" "A4" "A3" "A1" "A2"
```