

## metodo\_electre

**Nota:** Se han modificado las funciones para mejorar la presentación de los resultados en tablas mediante el paquete `kableExtra`, ya que el código funcionaba con la versión 1.3.4 del paquete `kableExtra`, con la versión 1.4.x han cambiado algunas funciones y parámetros y el código original no funcionaba correctamente. Se ha hecho una pequeña modificación, principalmente, en lugar de usar la función `kable()` ahora se usa la función `kbl()` en las funciones `func_ELECTRE_resTDiscordancia`, `func_ELECTRE_resTConcordancia` y `func_ELECTRE_resTSuperacion`.

```
fic_mult = here::here("teoriadecision_funciones_multicriterio.R")
source(file = fic_mult)
fic_mult_u = here::here("teoriadecision_funciones_multicriterio_utiles.R")
source(file = fic_mult_u)
```

Creamos la matriz de decisión:

```
t = multicriterio.crea.matrizdecision(
  c(-200, -100, -200, -550, 5,
    -5, -50, 0, -30, 7,
    -50, -5, 0, -100, 3,
    -50, -5, 0, -150, 3),
  numalternativas = 4, numcriterios = 5)
crisub = c(0.023, 0.063, 0.174, 0.106, 0.64)
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)
```

Mostramos los resultados de los tres tipos de tablas:

```
s1 = func_ELECTRE_resTDiscordancia(soll)
kableExtra::save_kable(s1$KE, file = "electreTDisc.html")
webshot::webshot("electreTDisc.html", file = "electreTDisc.png")
```

Alts	Inds	A1	A2	A3	A4
A1	I-		1,2,3,4,5	1,2,3,4	1,2,3,4
A1	Djk		195(100),50(20),200(Inf),520(Inf),2(2)	150(100),95(20),200(Inf),450(Inf)	150(100),95(20),200(Inf),400(Inf)
A1	TD	T	F	F	F
A2	I-			2	2
A2	Djk	NA		45(20)	45(20)
A2	TD	T	T	F	F
A3	I-	5	1,4,5		
A3	Djk	2(2)	45(100),70(Inf),4(2)		NA
A3	TD	F	F	T	T
A4	I-	5	1,4,5	4	
A4	Djk	2(2)	45(100),120(Inf),4(2)	50(Inf)	
A4	TD	F	F	T	T

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

```
## usar si falla paquete "webshot"
## webshot2::webshot("electreTDisc.html", file = "electreTDisc.png")
```

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

Alts	Inds	A1	A2	A3	A4
A1	Ijk	1	0	0.6362	0.6362
A1	IGjk	NaN	0	1.7486	1.7486
A1	TC	F	F	F	F
A2	Ijk	1	1	0.9374	0.9374
A2	IGjk	Inf	NaN	12.2063	12.2063
A2	TC	T	F	T	T
A3	Ijk	0.3638	0.2356	1	1
A3	IGjk	0.5719	0.0819	NaN	Inf
A3	TC	F	F	F	T
A4	Ijk	0.3638	0.2356	0.8946	1
A4	IGjk	0.5719	0.0819	0	NaN
A4	TC	F	F	F	F

ELECTRE: Test Concordancia

```
s1s = func_ELECTRE_resTSuperacion(sol1)
kableExtra::save_kable(s1s$KE, file = "electreTSup.html")
webshot::webshot("electreTSup.html", file = "electreTSup.png")
```

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

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

**Nota:** Una vez que se crean los gráficos png se pueden comentar las instrucciones `kableExtra::save_kable` y `webshot::webshot` para evitar recrear los gráficos en cada compilación del documento. Además, se podrían borrar los ficheros html generados si no se desean conservar (solo se usan para crear los png), es decir, borrar los ficheros: `electreTDisc.html`, `electreTConc.html` y `electreTSup.html`.

Por ejemplo, se podría usar el gráfico png de cualquier tabla, además sin mostrar el código R, como se muestra a continuación:

Alts	Inds	A1	A2	A3	A4
A1	I-		1,2,3,4,5	1,2,3,4	1,2,3,4
A1	Djk		195(100),50(20),200(Inf),520(Inf),2(2)	150(100),95(20),200(Inf),450(Inf)	150(100),95(20),200(Inf),400(Inf)
A1	TD	T	F	F	F
A2	I-			2	2
A2	Djk	NA		45(20)	45(20)
A2	TD	T	T	F	F
A3	I-	5	1,4,5		
A3	Djk	2(2)	45(100),70(Inf),4(2)		NA
A3	TD	F	F	T	T
A4	I-	5	1,4,5	4	
A4	Djk	2(2)	45(100),120(Inf),4(2)	50(Inf)	
A4	TD	F	F	T	T

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

Alts	Inds	A1	A2	A3	A4
A1	I-		1,2,3,4,5	1,2,3,4	1,2,3,4
A1	Djk		195(100),50(20),200(Inf),520(Inf),2(2)	150(100),95(20),200(Inf),450(Inf)	150(100),95(20),200(Inf),400(Inf)
A1	TD	T	F	F	F
A2	I-			2	2
A2	Djk	NA		45(20)	45(20)
A2	TD	T	T	F	F
A3	I-	5	1,4,5		
A3	Djk	2(2)	45(100),70(Inf),4(2)		NA
A3	TD	F	F	T	T
A4	I-	5	1,4,5	4	
A4	Djk	2(2)	45(100),120(Inf),4(2)	50(Inf)	
A4	TD	F	F	T	T

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

O también:

Alts	Inds	A1	A2	A3	A4
A1	I-		1,2,3,4,5	1,2,3,4	1,2,3,4
A1	Djk		195(100),50(20),200(Inf),520(Inf),2(2)	150(100),95(20),200(Inf),450(Inf)	150(100),95(20),200(Inf),400(Inf)
A1	TD	T	F	F	F
A2	I-			2	2
A2	Djk	NA		45(20)	45(20)
A2	TD	T	T	F	F
A3	I-	5	1,4,5		
A3	Djk	2(2)	45(100),70(Inf),4(2)		NA
A3	TD	F	F	T	T
A4	I-	5	1,4,5	4	
A4	Djk	2(2)	45(100),120(Inf),4(2)	50(Inf)	
A4	TD	F	F	T	T

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

Se ha usado el siguiente código markdown:

```

```{r}
#| echo: false
#| out-width: "80%"
s1 = func_ELECTRE_resTDiscordancia(sol1)
kableExtra::save_kable(s1$KE, file = "electreTDisc.html")
webshot::webshot("electreTDisc.html", file = "electreTDisc.png")
```

```{r}
#| echo: false
#| out-width: "80%"
# s1 = func_ELECTRE_resTDiscordancia(sol1)
# kableExtra::save_kable(s1$KE, file = "electreTDisc.html")
# webshot::webshot("electreTDisc.html", file = "electreTDisc.png")
include_graphics("electreTDisc.png")
```

```

0 también:

