

Trabajo02 - Teoría de la Decisión

Paula M^a Galindo Casas

- **Elección de la ubicación para abrir un nuevo restaurante**
- **Resolución con método Promethee (medias y Windows)**
- **Resolución con método Electre**
- **Resolución con método AHP**
 - Con funciones R
 - Mediante la librería AHP
- **Conclusiones finales**

Elección de la ubicación para abrir un nuevo restaurante

Una cadena de restaurantes está planeando expandir su negocio en la ciudad de Sevilla. Después de analizar varias zonas, la empresa ha seleccionado cinco ubicaciones donde podría abrir su próximo local. Sin embargo, cada una de estas ubicaciones presenta ventajas y desventajas en diferentes aspectos, por lo que la cadena desea tomar una decisión final basada en cinco criterios principales: facilidad en la accesibilidad para los clientes (puntuándose del 1 al 10, siendo 10 una accesibilidad excelente), costo mensual (en €) de alquiler del nuevo local, el tráfico de personas en la ubicación (puntuándose del 1 al 10, donde 10 representa el tráfico más alto de clientes), la visibilidad del local en avenidas principales o zonas concurridas (que se mide del 1 al 10, siendo 10 la mejor visibilidad) y la competencia en la zona con otros restaurantes (medida del 1 al 10, siendo 1 la máxima competencia).

Las cinco ubicaciones consideradas son:

Ubicación A, se trata de una zona céntrica con las siguientes características:

- Accesibilidad: 8 (muy buena)
- Costo de alquiler: 3000€
- Tráfico: 9, ya que se trataría de una zona peatonal al alcance de todos los ciudadanos.
- Visibilidad: 8 (muy buena)
- Competencia: 6

Ubicación B, se trata de un área comercial con las siguientes características:

- Accesibilidad: 6 (buena)
- Costo de alquiler: 2500€
- Tráfico: 7, se sitúa en buena zona comercial pero con competidores cercanos.
- Visibilidad: 7 (buena)
- Competencia: 5

Ubicación C, se trata de una zona de oficinas, donde trabajan muchas personas, con las siguientes características:

- Accesibilidad: 4 (baja)
- Costo de alquiler: 2000€
- Tráfico: 5, ya que el tráfico se desarrolla principalmente en el horario laboral.
- Visibilidad: 6 (buena)
- Competencia: 4

Ubicación D, se trata de un área universitaria, con las siguientes características:

- Accesibilidad: 7 (buena)
- Costo de alquiler: 2700€
- Tráfico: 8, ya que los estudiantes generan gran afluencia en horas de comida y tarde.
- Visibilidad: 7 (buena)
- Competencia: 6

Ubicación E, se trata de una zona residencial de alto nivel económico, con las siguientes características:

- Accesibilidad: 9 (muy buena)
- Costo de alquiler: 3500€
- Tráfico: 6, es una zona tranquila y con menor flujo peatonal.
- Visibilidad: 9 (muy buena)
- Competencia: 3

Los pesos que la cadena de restaurantes establece a los criterios estudiados serían los siguientes:

- **Accesibilidad:** 1, ya que es un aspecto importante pero no crucial.
- **Costo de alquiler:** 2, es el factor más relevante para la toma de la decisión final.
- **Tráfico:** 1.5, ya que es un factor clave para el éxito del restaurante.
- **Visibilidad:** 1.2, es importante que se sitúe en una zona visible para el público.
- **Competencia:** 0.8, no es el factor más importante para la cadena de restaurantes.

Para la decisión final es importante maximizar la accesibilidad, el tráfico de clientes, la visibilidad y competencia mientras que el costo de alquiler debe ser mínimo. Para recopilar toda esta información, se muestra la siguiente tabla con las valoraciones de las alternativas en cada criterio y los correspondientes parámetros para desarrollar el problema.

Criterios	Min/Max	Alternativas					Parámetros		
		Ubicación A	Ubicación B	Ubicación C	Ubicación D	Ubicación E	Tipo	q	p
Accesibilidad	Max	8	6	4	7	9	I	1	1
Costo de alquiler	Min	3000	2500	2000	2700	3500	V	0	2
Tráfico	Max	9	7	5	8	6	II	1	1.5
Visibilidad	Max	8	7	6	7	9	I	1	1.2
Competencia	Max	6	5	4	6	3	IV	1	0.8

Resuelve este problema mediante los métodos Promethee (aplicando también los de medias y Windows), Electre (iniciando el proceso con un $\alpha = 0.7$ y $d = (\infty, \infty, \infty, \infty, \infty)$) y AHP (con funciones R y con ayuda de la librería AHP).

Antes de empezar, importamos los scripts R donde se cargarán todas las funciones que necesitaremos para resolver el problema:

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

```
## Loading required package: shape
```

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

Resolución con método Promethee (medias y Windows)

Creemos la matriz de decisión teniendo en cuenta los criterios de maximizar o minimizar. En este último caso, los valores serán negativos.

```
matrizDecision = multicriterio.crea.matrizdecision(c(8, -3000, 9, 8, 6,
                                                    6, -2500, 7, 7, 5,
                                                    4, -2000, 5, 6, 4,
                                                    7, -2700, 8, 7, 6,
                                                    9, -3500, 6, 9, 3),
                                                    numalternativas = 5, numcriterios = 5,
                                                    v.nombresalt = c("Ubicación A","Ubicación B","Ubicación C",
                                                            "Ubicación D","Ubicación E"),
                                                    v.nombrescri = c("Accesibilidad","Costo alquiler",
                                                            "Tráfico","Visibilidad","Competencia"))

matrizDecision
```

```
##      Accesibilidad Costo alquiler Tráfico Visibilidad Competencia
## Ubicación A      8      -3000      9      8      6
## Ubicación B      6      -2500      7      7      5
## Ubicación C      4      -2000      5      6      4
## Ubicación D      7      -2700      8      7      6
## Ubicación E      9      -3500      6      9      3
```

Introducimos la tabla de preferencias:

```
tab.fpref = matrix(c(1, 1, 1, 2,
                    5, 0, 2, 4,
                    2, 1, 1.5, 3,
                    1, 1, 1.2, 1,
                    4, 1, 0.8, 3),
                  ncol = 4, byrow = T)
colnames(tab.fpref) <- c("num.pref", "qi", "pi", "si")
tab.fpref
```

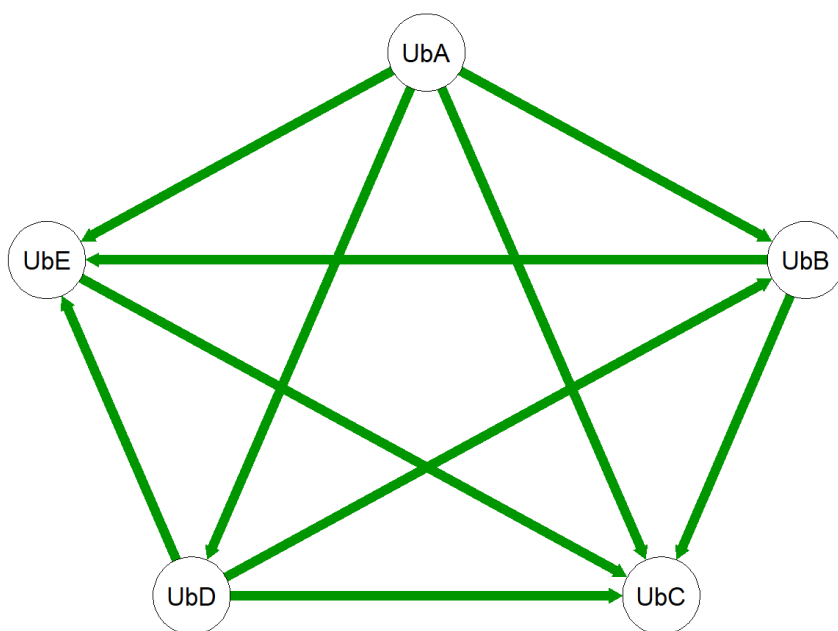
```
##      num.pref qi  pi si
## [1,]      1  1  1.0  2
## [2,]      5  0  2.0  4
## [3,]      2  1  1.5  3
## [4,]      1  1  1.2  1
## [5,]      4  1  0.8  3
```

Ahora ya podemos aplicar el método **Promethee I**.

```
tab.Pthee.i <- multicriterio.metodo.promethee_i(matrizDecision,  
pesos.criterios = tab.fpref[,3]/sum(tab.fpref[,3]),  
tab.fpref = tab.fpref)  
tab.Pthee.i
```

```
## $tabla.indices
##          Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      0.0000000      0.5692308      0.6923077      0.3384615      0.6615385
## Ubicación B      0.3076923      0.0000000      0.5692308      0.3076923      0.4307692
## Ubicación C      0.3076923      0.3076923      0.0000000      0.3076923      0.3076923
## Ubicación D      0.3076923      0.1538462      0.6923077      0.0000000      0.6615385
## Ubicación E      0.3384615      0.3384615      0.3384615      0.3384615      0.0000000
##
## $vflujos.ent
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
##      2.261538      1.615385      1.230769      1.815385      1.353846
##
## $vflujos.sal
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
##      1.261538      1.369231      2.292308      1.292308      2.061538
##
## $tablarelacionsupera
##          Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A          0.5          1.0          1.0          1.0          1.0
## Ubicación B          0.0          0.5          1.0          0.0          1.0
## Ubicación C          0.0          0.0          0.5          0.0          0.0
## Ubicación D          0.0          1.0          1.0          0.5          1.0
## Ubicación E          0.0          0.0          1.0          0.0          0.5
```

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



CONCLUSIÓN: Para el método Promethee I, la ubicación A supera al resto de alternativas completamente (1.0) y además, vemos claramente en el gráfico, que es el nodo del que salen todas las flechas. La ubicación B supera a las ubicaciones C y E. La ubicación C no supera a ninguna otra ubicación. La ubicación D supera completamente a B, C y E, y por último, la ubicación E sólo supera a C, observándose gráficamente el nodo de C es al que le llegan las flechas de todas las alternativas.

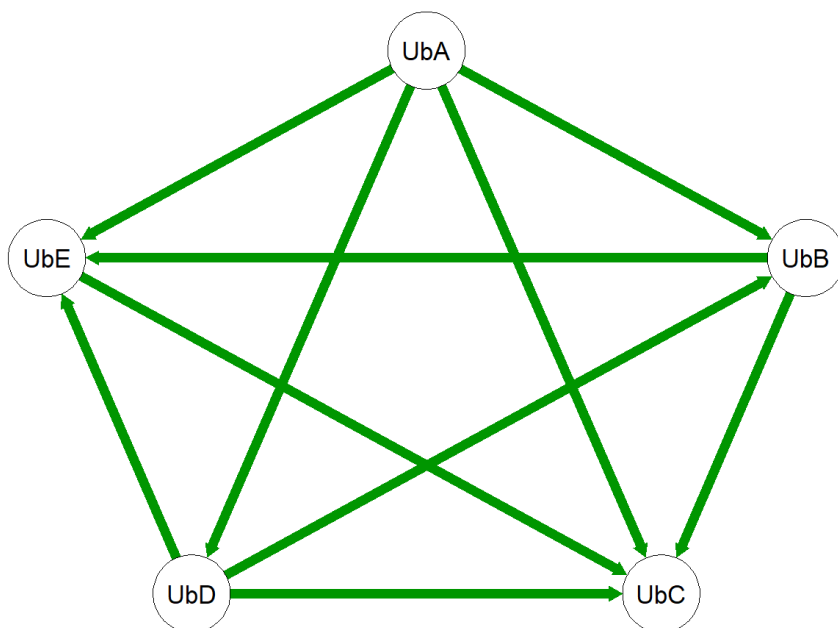
Como conclusión podemos decir que la ubicación A parece ser la más fuerte porque es la que supera completamente a todas las demás y la más débil.

Para llegar a una visión más clara aplicamos **Promethee II**.

```
tab.Pthee.ii <- multicriterio.metodo.promethee_ii(matrizDecision,  
                                                  pesos.criterios = tab.fpref[,3]/sum(tab.fpref[,3]),  
                                                  tab.fpref = tab.fpref)  
  
tab.Pthee.ii
```

```
## $tabla.indices
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A 0.0000000 0.5692308 0.6923077 0.3384615 0.6615385
## Ubicación B 0.3076923 0.0000000 0.5692308 0.3076923 0.4307692
## Ubicación C 0.3076923 0.3076923 0.0000000 0.3076923 0.3076923
## Ubicación D 0.3076923 0.1538462 0.6923077 0.0000000 0.6615385
## Ubicación E 0.3384615 0.3384615 0.3384615 0.3384615 0.0000000
##
## $vflujos.netos
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 1.0000000 0.2461538 -1.0615385 0.5230769 -0.7076923
##
## $tablarelacionsupera
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A 0.5 1.0 1.0 1.0 1.0
## Ubicación B 0.0 0.5 1.0 0.0 1.0
## Ubicación C 0.0 0.0 0.5 0.0 0.0
## Ubicación D 0.0 1.0 1.0 0.5 1.0
## Ubicación E 0.0 0.0 1.0 0.0 0.5
```

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



CONCLUSIÓN: dados los flujos netos (diferencia entre el flujo de entrada y de salida) se muestra claramente que la mejor alternativa es la ubicación A, que tiene el valor de flujo neto más alto (1.0000000) y domina a las otras ubicaciones. La ubicación D es preferida sobre B pero no tanto como A, ya que tiene un flujo neto más bajo con valor de 0.5230769. La peor alternativa, sin duda, será la ubicación C que posee un flujo neto negativo, de -1.0615385, indicando que es desfavorecida en comparación con las restantes.

La ordenación final de las alternativas según el método Promethee II será:

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

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

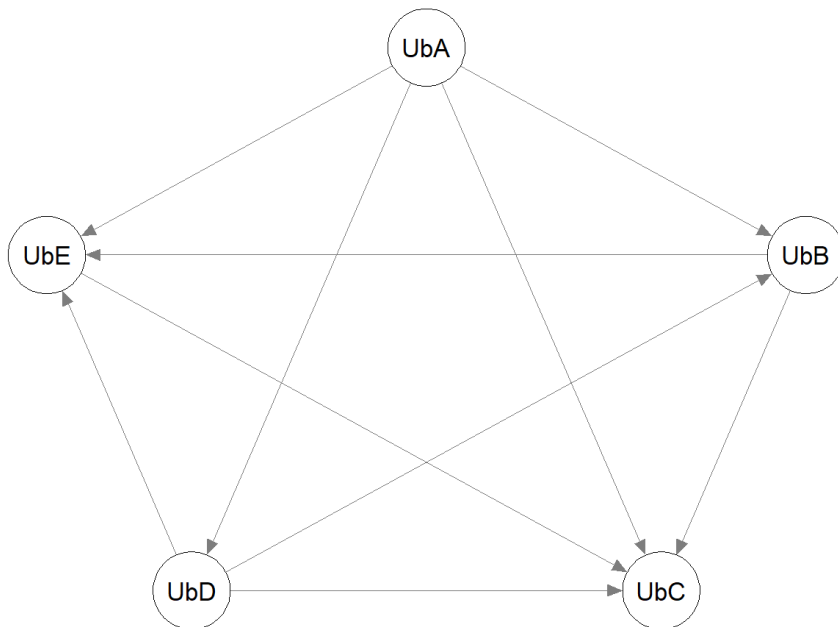
Finalmente podemos decir que la mejor decisión para la cadena será abrir el nuevo restaurante en la ubicación A, una zona céntrica de la ciudad.

Ahora lo analizamos mediante el método **Promethee I medias**.

```
tab.Pthee.i_medias <- multicriterio.metodo.promethee_i_med(matrizDecision,
  pesos.criterios = tab.fpref[,3]/sum(tab.fpref[,3]),
  tab.fpref = tab.fpref)
tab.Pthee.i_medias
```

```
## $tabla.indices
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A 0.000000 0.5692308 0.6923077 0.3384615 0.6615385
## Ubicación B 0.3076923 0.0000000 0.5692308 0.3076923 0.4307692
## Ubicación C 0.3076923 0.3076923 0.0000000 0.3076923 0.3076923
## Ubicación D 0.3076923 0.1538462 0.6923077 0.0000000 0.6615385
## Ubicación E 0.3384615 0.3384615 0.3384615 0.3384615 0.0000000
##
## $vflujos.ent
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.5653846 0.4038462 0.3076923 0.4538462 0.3384615
##
## $vflujos.sal
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.3153846 0.3423077 0.5730769 0.3230769 0.5153846
##
## $tablarelacionsupera
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A 0 1 1 1 1
## Ubicación B 0 0 1 0 1
## Ubicación C 0 0 0 0 0
## Ubicación D 0 1 1 0 1
## Ubicación E 0 0 1 0 0
```

```
qgraph::qgraph(tab.Pthee.i_medias$tablarelacionsupera)
```



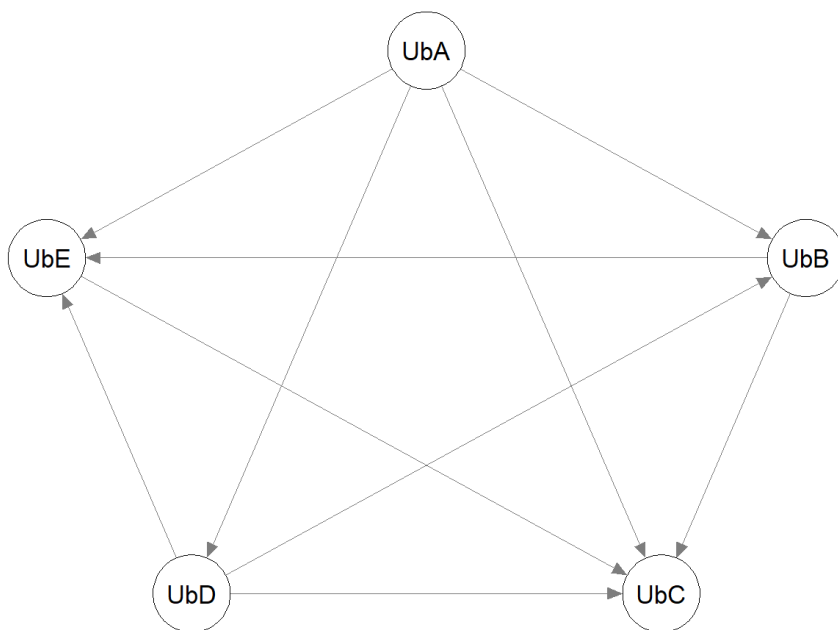
CONCLUSIÓN: se observa que la ubicación A sigue siendo es la opción más fuerte, ya que es la que domina a todas las demás y además gráficamente se ve claramente cómo domina en comparación con las restantes. La ubicación C es la alternativa más débil ya que no supera a ninguna otra.

Con el método **Promethee II medias**.

```
tab.Pthee.ii_medias <- multicriterio.metodo.promethee_ii_med(matrizDecision,
  pesos.criterios = tab.fpref[,3]/sum(tab.fpref[,3]),
  tab.fpref = tab.fpref)
tab.Pthee.ii_medias
```

```
## $tabla.indices
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A 0.0000000 0.5692308 0.6923077 0.3384615 0.6615385
## Ubicación B 0.3076923 0.0000000 0.5692308 0.3076923 0.4307692
## Ubicación C 0.3076923 0.3076923 0.0000000 0.3076923 0.3076923
## Ubicación D 0.3076923 0.1538462 0.6923077 0.0000000 0.6615385
## Ubicación E 0.3384615 0.3384615 0.3384615 0.3384615 0.0000000
##
## $vflujos.netos
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.25000000 0.06153846 -0.26538462 0.13076923 -0.17692308
##
## $tablarelacionsupera
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A 0 1 1 1 1
## Ubicación B 0 0 1 0 1
## Ubicación C 0 0 0 0 0
## Ubicación D 0 1 1 0 1
## Ubicación E 0 0 1 0 0
```

```
qgraph::qgraph(tab.Pthee.ii_medias$tablarelacionsupera)
```



La ordenación final de las alternativas según el método Promethee II medias será:

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

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

Los resultados coinciden con el método Promethee II analizado antes, la cadena deberá elegir la ubicación A, una zona céntrica de Sevilla, para abrir su nuevo restaurante. Esta alternativa es la que domina a todas las demás y presenta el mayor flujo neto, un valor de 0.25000000.

Hacemos ahora la resolución con **Promethee Windows**, donde los costos se muestran con signo negativo.

```
res=multicriterio.metodo.promethee_windows(matrizDecision, tab.fpref, tab.fpref[,3]/sum(tab.fpref[,3]))
res
```

```

## $Escenario
##
## Pesos                Criterio1          Criterio2
## Funciones Preferencias "Usual (1)"          "Linear (5)"
## Q: Indiferencia      "1"              "0"
## P: Preferencia       "1"              "2"
## S: Gausiano          "2"              "4"
## Minimo               "4"              "-3500"
## Maximo               "9"              "-2000"
## Media                "6.8"            "-2740"
## Desviacion Tipica    "1.72"           "500.4"
## Ubicación A          "8"              "-3000"
## Ubicación B          "6"              "-2500"
## Ubicación C          "4"              "-2000"
## Ubicación D          "7"              "-2700"
## Ubicación E          "9"              "-3500"
##
##                      Criterio3          Criterio4
## Pesos                "0.230769230769231" "0.184615384615385"
## Funciones Preferencias "U-shape (2)"          "Usual (1)"
## Q: Indiferencia      "1"              "1"
## P: Preferencia       "1.5"            "1.2"
## S: Gausiano          "3"              "1"
## Minimo               "5"              "6"
## Maximo               "9"              "9"
## Media                "7"              "7.4"
## Desviacion Tipica    "1.41"           "1.02"
## Ubicación A          "9"              "8"
## Ubicación B          "7"              "7"
## Ubicación C          "5"              "6"
## Ubicación D          "8"              "7"
## Ubicación E          "6"              "9"
##
##                      Criterio5
## Pesos                "0.123076923076923"
## Funciones Preferencias "Level (4)"
## Q: Indiferencia      "1"
## P: Preferencia       "0.8"
## S: Gausiano          "3"
## Minimo               "3"
## Maximo               "6"
## Media                "4.8"
## Desviacion Tipica    "1.17"
## Ubicación A          "6"
## Ubicación B          "5"
## Ubicación C          "4"
## Ubicación D          "6"
## Ubicación E          "3"
##
## $Acciones
##           Rango      Phi Phi.mas Phi.menos
## Ubicación A      1  0.2500  0.5654  0.3154
## Ubicación D      2  0.1308  0.4538  0.3231
## Ubicación B      3  0.0615  0.4038  0.3423
## Ubicación E      4 -0.1769  0.3385  0.5154
## Ubicación C      5 -0.2654  0.3077  0.5731

```

Se pueden ver los resultados por apartados de manera más clara en la siguiente salida:

```

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

```

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5
Preferencias					
Pesos	0.153846153846154	0.307692307692308	0.230769230769231	0.184615384615385	0.123076923076923
Funciones Preferencias	Usual (1)	Linear (5)	U-shape (2)	Usual (1)	Level (4)
Q: Indiferencia	1	0	1	1	1
P: Preferencia	1	2	1.5	1.2	0.8
S: Gausiano	2	4	3	1	3
Minimo	4	-3500	5	6	3

	Criterio1	Criterio2	Criterio3	Criterio4	Criterio5
Estadísticas					
Maximo	9	-2000	9	9	6
Media	6.8	-2740	7	7.4	4.8
Desviacion Tipica	1.72	500.4	1.41	1.02	1.17
Ubicación A	8	-3000	9	8	6
Evaluaciones					
Ubicación B	6	-2500	7	7	5
Ubicación C	4	-2000	5	6	4
Ubicación D	7	-2700	8	7	6
Ubicación E	9	-3500	6	9	3

La ordenación de las ubicaciones que se estudian (alternativas) son:

```
tabla_res$tabAcciones
```

	Rango	Phi	Phi.mas	Phi.menos
Ubicación A	1	0.2500	0.5654	0.3154
Ubicación D	2	0.1308	0.4538	0.3231
Ubicación B	3	0.0615	0.4038	0.3423
Ubicación E	4	-0.1769	0.3385	0.5154
Ubicación C	5	-0.2654	0.3077	0.5731

CONCLUSIÓN: de igual manera que concluimos en los métodos Promethee anteriores, podemos decir que la mejor alternativa que puede tomar la cadena para abrir un nuevo restaurante será la ubicación A, una zona céntrica de la ciudad, seguida de la ubicación D, B, E y quedando como peor alternativa la ubicación C, una zona de oficinas.

Resolución con método Electre

Partiendo de la matriz de decisión creada al principio donde los criterios de minimizar son valores negativos y los de maximizar son positivos, resolvemos el problema mediante el método Electre.

Matriz de decisión:

```
matrizDecision
```

```
##          Accesibilidad Costo alquiler Tráfico Visibilidad Competencia
## Ubicación A           8         -3000      9           8           6
## Ubicación B           6         -2500      7           7           5
## Ubicación C           4         -2000      5           6           4
## Ubicación D           7         -2700      8           7           6
## Ubicación E           9         -3500      6           9           3
```

Comenzamos el proceso con un valor $\alpha = 0.7$ y $d = (\infty, \infty, \infty, \infty, \infty)$.

```
solElectre_1 <- multicriterio.metodoELECTRE_I(matrizDecision,
                                              pesos.criterios = tab.fpref[,3]/sum(tab.fpref[,3]),
                                              nivel.concordancia.minimo.alpha = 0.7,
                                              no.se.compensan = c(Inf, Inf, Inf, Inf, Inf),
                                              que.alternativas = TRUE)

solElectre_1
```



```

## $datos
##
## Ubicación A      8.0000000 -3000.0000000 9.0000000 8.0000000 6.0000000
## Ubicación B      6.0000000 -2500.0000000 7.0000000 7.0000000 5.0000000
## Ubicación C      4.0000000 -2000.0000000 5.0000000 6.0000000 4.0000000
## Ubicación D      7.0000000 -2700.0000000 8.0000000 7.0000000 6.0000000
## Ubicación E      9.0000000 -3500.0000000 6.0000000 9.0000000 3.0000000
## pesos.criterios  0.1538462 0.3076923 0.2307692 0.1846154 0.1230769
## no.se.compensan  Inf          Inf          Inf          Inf          Inf
##
## $alpha
## [1] 0.7
##
## $Imas
## , , Accesibilidad
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      TRUE      FALSE
## Ubicación B FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación C FALSE      FALSE     FALSE     FALSE     FALSE
## Ubicación D FALSE      TRUE      TRUE      FALSE     FALSE
## Ubicación E TRUE       TRUE      TRUE      TRUE      FALSE
##
## , , Costo alquiler
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      FALSE     FALSE     FALSE     TRUE
## Ubicación B TRUE       FALSE     FALSE     TRUE      TRUE
## Ubicación C TRUE       TRUE      FALSE     TRUE      TRUE
## Ubicación D TRUE       FALSE     FALSE     FALSE     TRUE
## Ubicación E FALSE      FALSE     FALSE     FALSE     FALSE
##
## , , Tráfico
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      TRUE      TRUE
## Ubicación B FALSE      FALSE     TRUE      FALSE     TRUE
## Ubicación C FALSE      FALSE     FALSE     FALSE     FALSE
## Ubicación D FALSE      TRUE      TRUE      FALSE     TRUE
## Ubicación E FALSE      FALSE     TRUE      FALSE     FALSE
##
## , , Visibilidad
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      TRUE      FALSE
## Ubicación B FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación C FALSE      FALSE     FALSE     FALSE     FALSE
## Ubicación D FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación E TRUE       TRUE      TRUE      TRUE      FALSE
##
## , , Competencia
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      FALSE     TRUE
## Ubicación B FALSE      FALSE     TRUE      FALSE     TRUE
## Ubicación C FALSE      FALSE     FALSE     FALSE     TRUE
## Ubicación D FALSE      TRUE      TRUE      FALSE     TRUE
## Ubicación E FALSE      FALSE     FALSE     FALSE     FALSE
##
##
## $Iigual
## , , Accesibilidad
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A TRUE       FALSE     FALSE     FALSE     FALSE
## Ubicación B FALSE      TRUE      FALSE     FALSE     FALSE
## Ubicación C FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación D FALSE      FALSE     FALSE     TRUE      FALSE
## Ubicación E FALSE      FALSE     FALSE     FALSE     TRUE
##
## , , Costo alquiler
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A TRUE       FALSE     FALSE     FALSE     FALSE
## Ubicación B FALSE      TRUE      FALSE     FALSE     FALSE
## Ubicación C FALSE      FALSE     TRUE      FALSE     FALSE

```

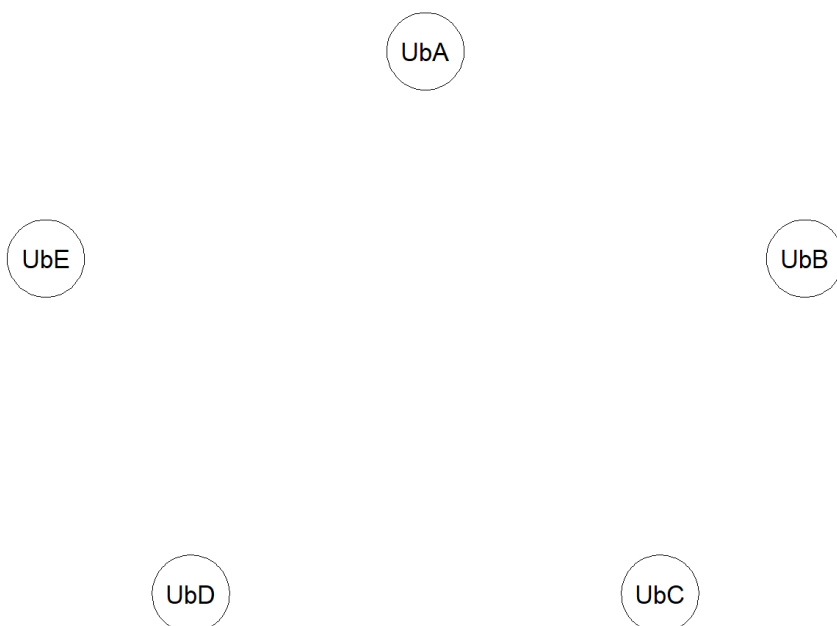
```

## Ubicación D      FALSE      FALSE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## , , Tráfico
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      FALSE      TRUE      FALSE      FALSE      FALSE
## Ubicación C      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación D      FALSE      FALSE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## , , Visibilidad
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      FALSE      TRUE      FALSE      TRUE      FALSE
## Ubicación C      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación D      FALSE      TRUE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## , , Competencia
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación B      FALSE      TRUE      FALSE      FALSE      FALSE
## Ubicación C      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación D      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## $Imenos
## , , Accesibilidad
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      TRUE
## Ubicación B      TRUE      FALSE      FALSE      TRUE      TRUE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      TRUE
## Ubicación D      TRUE      FALSE      FALSE      FALSE      TRUE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      FALSE
##
## , , Costo alquiler
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      TRUE      TRUE      TRUE      FALSE
## Ubicación B      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación C      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación D      FALSE      TRUE      TRUE      FALSE      FALSE
## Ubicación E      TRUE      TRUE      TRUE      TRUE      FALSE
##
## , , Tráfico
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      TRUE
## Ubicación D      TRUE      FALSE      FALSE      FALSE      FALSE
## Ubicación E      TRUE      TRUE      FALSE      TRUE      FALSE
##
## , , Visibilidad
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      TRUE
## Ubicación B      TRUE      FALSE      FALSE      FALSE      TRUE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      TRUE
## Ubicación D      TRUE      FALSE      FALSE      FALSE      TRUE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      FALSE
##
## , , Competencia
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      FALSE
## Ubicación D      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación E      TRUE      TRUE      TRUE      TRUE      FALSE

```

```
##
##
## $ind.concordancia
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A    1.0000000    0.6923077    0.6923077    0.6923077    0.6615385
## Ubicación B    0.3076923    1.0000000    0.6923077    0.4923077    0.6615385
## Ubicación C    0.3076923    0.3076923    1.0000000    0.3076923    0.4307692
## Ubicación D    0.4307692    0.6923077    0.6923077    1.0000000    0.6615385
## Ubicación E    0.3384615    0.3384615    0.5692308    0.3384615    1.0000000
##
## $ind.concordancia.gorro
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      NaN    2.2500000    2.2500000    1.8500000    1.9545455
## Ubicación B    0.4444444      NaN    2.2500000    0.6060606    1.9545455
## Ubicación C    0.4444444    0.4444444      NaN    0.4444444    0.7567568
## Ubicación D    0.5405405    1.6500000    2.2500000      NaN    1.9545455
## Ubicación E    0.5116279    0.5116279    1.321429    0.5116279      NaN
##
## $test.concordancia
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación B    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación C    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación D    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación E    FALSE      FALSE      FALSE      FALSE      FALSE
##
## $test.discordancia
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación B      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación C      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación D      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación E      TRUE      TRUE      TRUE      TRUE      TRUE
##
## $relacion.dominante
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación B    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación C    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación D    FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación E    FALSE      FALSE      FALSE      FALSE      FALSE
##
## $nucleo_aprox
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
##      1      2      3      4      5
```

```
qgraph::qgraph(solElectre_1$relacion.dominante)
```



solElectre_1\$nucleo_aprox

```
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
##          1           2           3           4           5
```

CONCLUSIÓN: en el núcleo aproximado se encuentran todas las ubicaciones. Por lo tanto volvemos a repetir el método para un valor más bajo del índice de concordancia mínimo. Probamos con $\alpha = 0.65$ manteniendo las mismas alternativas.

```
solElectre_2 <- multicriterio.metodoELECTRE_I(matrizDecision,  
                                              pesos.criterios = tab.fpref[,3]/sum(tab.fpref[,3]),  
                                              nivel.concordancia.minimo.alpha = 0.65,  
                                              no.se.compensan = c(Inf, Inf, Inf, Inf, Inf),  
                                              que.alternativas = TRUE)  
  
solElectre_2
```

```

## $datos
##
## Ubicación A      8.0000000 -3000.0000000 9.0000000 8.0000000 6.0000000
## Ubicación B      6.0000000 -2500.0000000 7.0000000 7.0000000 5.0000000
## Ubicación C      4.0000000 -2000.0000000 5.0000000 6.0000000 4.0000000
## Ubicación D      7.0000000 -2700.0000000 8.0000000 7.0000000 6.0000000
## Ubicación E      9.0000000 -3500.0000000 6.0000000 9.0000000 3.0000000
## pesos.criterios  0.1538462 0.3076923 0.2307692 0.1846154 0.1230769
## no.se.compensan  Inf          Inf          Inf          Inf          Inf
##
## $alpha
## [1] 0.65
##
## $Imas
## , , Accesibilidad
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      TRUE      FALSE
## Ubicación B FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación C FALSE      FALSE     FALSE     FALSE     FALSE
## Ubicación D FALSE      TRUE      TRUE      FALSE     FALSE
## Ubicación E TRUE       TRUE      TRUE      TRUE      FALSE
##
## , , Costo alquiler
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      FALSE     FALSE     FALSE     TRUE
## Ubicación B TRUE       FALSE     FALSE     TRUE      TRUE
## Ubicación C TRUE       TRUE      FALSE     TRUE      TRUE
## Ubicación D TRUE       FALSE     FALSE     FALSE     TRUE
## Ubicación E FALSE      FALSE     FALSE     FALSE     FALSE
##
## , , Tráfico
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      TRUE      TRUE
## Ubicación B FALSE      FALSE     TRUE      FALSE     TRUE
## Ubicación C FALSE      FALSE     FALSE     FALSE     FALSE
## Ubicación D FALSE      TRUE      TRUE      FALSE     TRUE
## Ubicación E FALSE      FALSE     TRUE      FALSE     FALSE
##
## , , Visibilidad
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      TRUE      FALSE
## Ubicación B FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación C FALSE      FALSE     FALSE     FALSE     FALSE
## Ubicación D FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación E TRUE       TRUE      TRUE      TRUE      FALSE
##
## , , Competencia
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A FALSE      TRUE      TRUE      FALSE     TRUE
## Ubicación B FALSE      FALSE     TRUE      FALSE     TRUE
## Ubicación C FALSE      FALSE     FALSE     FALSE     TRUE
## Ubicación D FALSE      TRUE      TRUE      FALSE     TRUE
## Ubicación E FALSE      FALSE     FALSE     FALSE     FALSE
##
##
## $Iigual
## , , Accesibilidad
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A TRUE       FALSE     FALSE     FALSE     FALSE
## Ubicación B FALSE      TRUE      FALSE     FALSE     FALSE
## Ubicación C FALSE      FALSE     TRUE      FALSE     FALSE
## Ubicación D FALSE      FALSE     FALSE     TRUE      FALSE
## Ubicación E FALSE      FALSE     FALSE     FALSE     TRUE
##
## , , Costo alquiler
##
##
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A TRUE       FALSE     FALSE     FALSE     FALSE
## Ubicación B FALSE      TRUE      FALSE     FALSE     FALSE
## Ubicación C FALSE      FALSE     TRUE      FALSE     FALSE

```

```

## Ubicación D      FALSE      FALSE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## , , Tráfico
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      FALSE      TRUE      FALSE      FALSE      FALSE
## Ubicación C      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación D      FALSE      FALSE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## , , Visibilidad
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      FALSE      TRUE      FALSE      TRUE      FALSE
## Ubicación C      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación D      FALSE      TRUE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## , , Competencia
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación B      FALSE      TRUE      FALSE      FALSE      FALSE
## Ubicación C      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación D      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      TRUE
##
## $Imenos
## , , Accesibilidad
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      TRUE
## Ubicación B      TRUE      FALSE      FALSE      TRUE      TRUE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      TRUE
## Ubicación D      TRUE      FALSE      FALSE      FALSE      TRUE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      FALSE
##
## , , Costo alquiler
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      TRUE      TRUE      TRUE      FALSE
## Ubicación B      FALSE      FALSE      TRUE      FALSE      FALSE
## Ubicación C      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación D      FALSE      TRUE      TRUE      FALSE      FALSE
## Ubicación E      TRUE      TRUE      TRUE      TRUE      FALSE
##
## , , Tráfico
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      TRUE
## Ubicación D      TRUE      FALSE      FALSE      FALSE      FALSE
## Ubicación E      TRUE      TRUE      FALSE      TRUE      FALSE
##
## , , Visibilidad
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      TRUE
## Ubicación B      TRUE      FALSE      FALSE      FALSE      TRUE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      TRUE
## Ubicación D      TRUE      FALSE      FALSE      FALSE      TRUE
## Ubicación E      FALSE      FALSE      FALSE      FALSE      FALSE
##
## , , Competencia
##
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación B      TRUE      FALSE      FALSE      TRUE      FALSE
## Ubicación C      TRUE      TRUE      FALSE      TRUE      FALSE
## Ubicación D      FALSE      FALSE      FALSE      FALSE      FALSE
## Ubicación E      TRUE      TRUE      TRUE      TRUE      FALSE

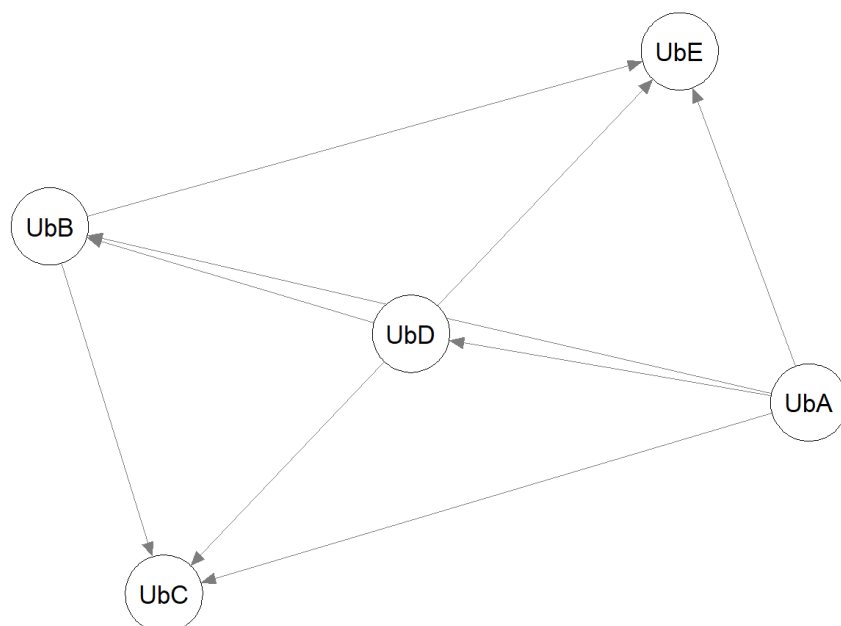
```

```

##
##
## $ind.concordancia
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A    1.0000000    0.6923077    0.6923077    0.6923077    0.6615385
## Ubicación B    0.3076923    1.0000000    0.6923077    0.4923077    0.6615385
## Ubicación C    0.3076923    0.3076923    1.0000000    0.3076923    0.4307692
## Ubicación D    0.4307692    0.6923077    0.6923077    1.0000000    0.6615385
## Ubicación E    0.3384615    0.3384615    0.5692308    0.3384615    1.0000000
##
## $ind.concordancia.gorro
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      NaN    2.2500000    2.2500000    1.8500000    1.9545455
## Ubicación B    0.4444444      NaN    2.2500000    0.6060606    1.9545455
## Ubicación C    0.4444444    0.4444444      NaN    0.4444444    0.7567568
## Ubicación D    0.5405405    1.6500000    2.2500000      NaN    1.9545455
## Ubicación E    0.5116279    0.5116279    1.321429    0.5116279      NaN
##
## $test.concordancia
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A     FALSE      TRUE      TRUE      TRUE      TRUE
## Ubicación B     FALSE     FALSE      TRUE     FALSE      TRUE
## Ubicación C     FALSE     FALSE     FALSE     FALSE     FALSE
## Ubicación D     FALSE      TRUE      TRUE     FALSE      TRUE
## Ubicación E     FALSE     FALSE     FALSE     FALSE     FALSE
##
## $test.discordancia
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación B      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación C      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación D      TRUE      TRUE      TRUE      TRUE      TRUE
## Ubicación E      TRUE      TRUE      TRUE      TRUE      TRUE
##
## $relacion.dominante
##      Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A     FALSE      TRUE      TRUE      TRUE      TRUE
## Ubicación B     FALSE     FALSE      TRUE     FALSE      TRUE
## Ubicación C     FALSE     FALSE     FALSE     FALSE     FALSE
## Ubicación D     FALSE      TRUE      TRUE     FALSE      TRUE
## Ubicación E     FALSE     FALSE     FALSE     FALSE     FALSE
##
## $nucleo_aprox
## Ubicación A
##      1

```

```
qgraph::qgraph(solElectre_2$relacion.dominante)
```



solElectre_2\$nucleo_aprox

Ubicación A
1

CONCLUSIÓN: para un valor $\alpha = 0.65$ hemos conseguido reducir el núcleo aproximado a una sola alternativa que se trata de la ubicación A. Por tanto podemos decir que la mejor decisión que puede tomar la cadena para abrir el nuevo restaurante es la ubicación A, una zona céntrica de la ciudad.

Resolución con método AHP

Para determinar la mejor ubicación, la cadena ha identificado las siguientes matrices de comparación por pares para cada criterio (accesibilidad, costo de alquiler, tráfico, visibilidad y competencia) para evaluar las ubicaciones A, B, C, D y E.

Accesibilidad	A	B	C	D	E	Costo alquiler	A	B	C	D	E	Tráfico	A	B	C	D	E
A	1	3	5	2	$\frac{1}{3}$	A	1	4	5	2	$\frac{1}{4}$	A	1	3	5	2	4
B	$\frac{1}{3}$	1	3	$\frac{1}{2}$	$\frac{1}{4}$	B	$\frac{1}{4}$	1	3	1	$\frac{1}{5}$	B	$\frac{1}{3}$	1	4	$\frac{1}{3}$	2
C	$\frac{1}{5}$	$\frac{1}{3}$	1	$\frac{1}{5}$	$\frac{1}{6}$	C	$\frac{1}{5}$	$\frac{1}{3}$	1	$\frac{1}{4}$	$\frac{1}{6}$	C	$\frac{1}{5}$	$\frac{1}{4}$	1	$\frac{1}{4}$	$\frac{1}{2}$
D	$\frac{1}{2}$	2	5	1	$\frac{1}{3}$	D	$\frac{1}{2}$	1	4	1	$\frac{1}{3}$	D	$\frac{1}{2}$	3	4	1	3
E	3	4	6	3	1	E	4	5	6	3	1	E	$\frac{1}{4}$	$\frac{1}{2}$	2	$\frac{1}{3}$	1

Visibilidad	A	B	C	D	E	Competencia	A	B	C	D	E
A	1	2	3	2	$\frac{1}{2}$	A	1	3	4	1	6
B	$\frac{1}{2}$	1	2	1	$\frac{1}{2}$	B	$\frac{1}{3}$	1	2	$\frac{1}{3}$	4
C	$\frac{1}{3}$	$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{3}$	C	$\frac{1}{4}$	$\frac{1}{2}$	1	$\frac{1}{4}$	2
D	$\frac{1}{2}$	1	2	1	$\frac{1}{3}$	D	1	3	4	1	6
E	2	2	3	3	1	E	$\frac{1}{6}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{6}$	1

También ha recogido sus prioridades en relación con los cinco criterios manejados.

	Accesibilidad	Costo alquiler	Tráfico	Visibilidad	Competencia
Accesibilidad	1	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	2
Costo alquiler	4	1	2	3	6
Tráfico	3	$\frac{1}{2}$	1	2	7
Visibilidad	2	$\frac{1}{3}$	$\frac{1}{2}$	1	3
Competencia	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{3}$	1

A partir de estos datos se pide:

- a. Identificar el diagrama de jerarquías de este problema de decisión.
- b. Calcular las ponderaciones asociadas a cada criterio manejado.
- c. Calcular las ponderaciones de las cinco alternativas en función de cada criterio.
- d. Verificar la consistencia de las preferencias de la cadena de restaurantes.
- e. Determinar la prioridad global de cada ubicación.

Empiezo creando las matrices anteriores para poder trabajar.

Matriz de criterios


```

nombre.criterios <-c("Accesibilidad","Costo alquiler","Tráfico","Visibilidad","Competencia")
Xmat.criterios= multicriterio.crea.matrizvaloraciones(c( 1, 1/4, 1/3, 1/2, 2,
4, 1, 2, 3, 6,
3, 1/2, 1, 2, 7,
2, 1/3, 1/2, 1, 3,
1/2, 1/6, 1/7, 1/3, 1),
numalternativas = 5,
v.nombres.alternativas = nombre.criterios)

Xmat.criterios

```

```

##          Accesibilidad Costo alquiler  Tráfico Visibilidad Competencia
## Accesibilidad      1.0      0.2500000 0.3333333  0.5000000      2
## Costo alquiler      4.0      1.0000000 2.0000000  3.0000000      6
## Tráfico             3.0      0.5000000 1.0000000  2.0000000      7
## Visibilidad         2.0      0.3333333 0.5000000  1.0000000      3
## Competencia         0.5      0.1666667 0.1428571  0.3333333      1

```

Matriz Alternativas/Criterios

• Accesibilidad

```

nombre.alternativas <- c("Ubicación A","Ubicación B","Ubicación C","Ubicación D","Ubicación E")
Xmat.accesibilidad = multicriterio.crea.matrizvaloraciones(c( 1, 3, 5, 2, 1/3,
1/3, 1, 3, 1/2, 1/4,
1/5, 1/3, 1, 1/5, 1/6,
1/2, 2, 5, 1, 1/3,
3, 4, 6, 3, 1),
numalternativas = 5,
nombre.alternativas)

Xmat.accesibilidad

```

```

##          Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      1.0000000  3.0000000      5      2.0  0.3333333
## Ubicación B      0.3333333  1.0000000      3      0.5  0.2500000
## Ubicación C      0.2000000  0.3333333      1      0.2  0.1666667
## Ubicación D      0.5000000  2.0000000      5      1.0  0.3333333
## Ubicación E      3.0000000  4.0000000      6      3.0  1.0000000

```

• Costo de alquiler

```

nombre.alternativas <- c("Ubicación A","Ubicación B","Ubicación C","Ubicación D","Ubicación E")
Xmat.costoAlquiler = multicriterio.crea.matrizvaloraciones(c( 1, 4, 5, 2, 1/4,
1/4, 1, 3, 1, 1/5,
1/5, 1/3, 1, 1/4, 1/6,
1/2, 1, 4, 1, 1/3,
4, 5, 6, 3, 1),
numalternativas = 5,
nombre.alternativas)

Xmat.costoAlquiler

```

```

##          Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## Ubicación A      1.00  4.0000000      5      2.00  0.2500000
## Ubicación B      0.25  1.0000000      3      1.00  0.2000000
## Ubicación C      0.20  0.3333333      1      0.25  0.1666667
## Ubicación D      0.50  1.0000000      4      1.00  0.3333333
## Ubicación E      4.00  5.0000000      6      3.00  1.0000000

```

• Tráfico

```

nombre.alternativas <- c("Ubicación A","Ubicación B","Ubicación C","Ubicación D","Ubicación E")
Xmat.trafico = multicriterio.crea.matrizvaloraciones(c( 1, 3, 5, 2, 4,
1/3, 1, 4, 1/3, 2,
1/5, 1/4, 1, 1/4, 1/2,
1/2, 3, 4, 1, 3,
1/4, 1/2, 2, 1/3, 1),
numalternativas = 5,
nombre.alternativas)

Xmat.trafico

```

##	Ubicación A	Ubicación B	Ubicación C	Ubicación D	Ubicación E
## Ubicación A	1.0000000	3.00	5	2.0000000	4.0
## Ubicación B	0.3333333	1.00	4	0.3333333	2.0
## Ubicación C	0.2000000	0.25	1	0.2500000	0.5
## Ubicación D	0.5000000	3.00	4	1.0000000	3.0
## Ubicación E	0.2500000	0.50	2	0.3333333	1.0

• Visibilidad

```

nombre.alternativas <- c("Ubicación A","Ubicación B","Ubicación C","Ubicación D","Ubicación E")
Xmat.visibilidad = multicriterio.crea.matrizvaloraciones(c( 1, 2, 3, 2, 1/2,
                                                           1/2, 1, 2, 1, 1/2,
                                                           1/3, 1/2, 1, 1/2, 1/3,
                                                           1/2, 1, 2, 1, 1/3,
                                                           2, 2, 3, 3, 1),
                                                           numalternativas = 5,
                                                           nombre.alternativas)
Xmat.visibilidad

```

##	Ubicación A	Ubicación B	Ubicación C	Ubicación D	Ubicación E
## Ubicación A	1.0000000	2.0	3	2.0	0.5000000
## Ubicación B	0.5000000	1.0	2	1.0	0.5000000
## Ubicación C	0.3333333	0.5	1	0.5	0.3333333
## Ubicación D	0.5000000	1.0	2	1.0	0.3333333
## Ubicación E	2.0000000	2.0	3	3.0	1.0000000

• Competencia

```

nombre.alternativas <- c("Ubicación A","Ubicación B","Ubicación C","Ubicación D","Ubicación E")
Xmat.competencia = multicriterio.crea.matrizvaloraciones(c( 1, 3, 4, 1, 6,
                                                           1/3, 1, 2, 1/3, 4,
                                                           1/4, 1/2, 1, 1/4, 2,
                                                           1, 3, 4, 1, 6,
                                                           1/6, 1/4, 1/2, 1/6, 1),
                                                           numalternativas = 5,
                                                           nombre.alternativas)
Xmat.competencia

```

##	Ubicación A	Ubicación B	Ubicación C	Ubicación D	Ubicación E
## Ubicación A	1.0000000	3.00	4.0	1.0000000	6
## Ubicación B	0.3333333	1.00	2.0	0.3333333	4
## Ubicación C	0.2500000	0.50	1.0	0.2500000	2
## Ubicación D	1.0000000	3.00	4.0	1.0000000	6
## Ubicación E	0.1666667	0.25	0.5	0.1666667	1

Con funciones R

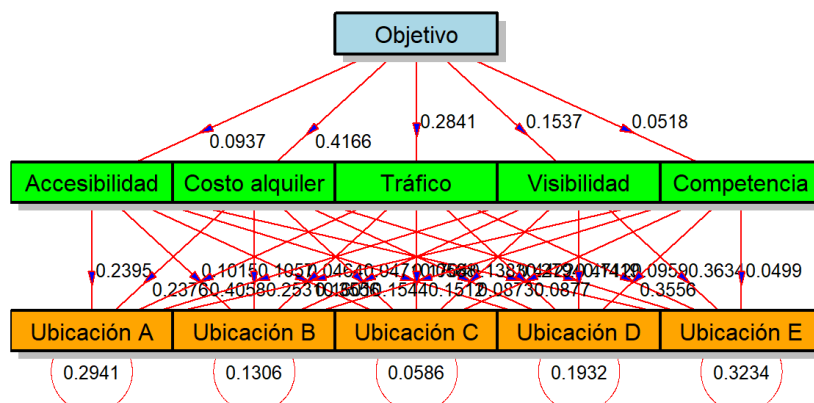
(a) Identificar el diagrama de jerarquías de este problema de decisión.

```

num.alt=5
num.cri=5
Xmatriznivel2 = array(NA,dim=c(num.alt,num.alt,num.cri))
Xmatriznivel2[,1] = Xmat.accesibilidad
Xmatriznivel2[,2] = Xmat.costoAlquiler
Xmatriznivel2[,3] = Xmat.trafico
Xmatriznivel2[,4] = Xmat.visibilidad
Xmatriznivel2[,5] = Xmat.competencia
dimnames(Xmatriznivel2)[[1]] = nombre.alternativas
dimnames(Xmatriznivel2)[[2]] = nombre.alternativas
dimnames(Xmatriznivel2)[[3]] = nombre.criterios
multicriterio.metodoahp.diagrama(Xmat.criterios,Xmatriznivel2)

```

Estructura Jerárquica (AHP)



En esta estructura jerárquica podemos ver claramente tres niveles: el primero define el objetivo principal del problema, el segundo nivel son los correspondientes a los criterios donde se reflejan los pesos de cada uno de ellos y el último nivel son las alternativas junto con sus pesos. Con solo mirarlo se puede decir que la mejor alternativa sería la ubicación E con un peso de 0.3234 y la peor es la ubicación C con un peso de 0.0586.

(b) Calcular las ponderaciones asociadas a cada criterio manejado.

```
pesos.criterios = multicriterio.metodoAHP.variante3.basico(Xmat.criterios)
pesos.criterios$valoraciones.ahp
```

```
## Accesibilidad Costo alquiler Tráfico Visibilidad Competencia
## 0.09372309 0.41664094 0.28410750 0.15372181 0.05180667
```

```
# Pesos ordenados
pesos.criterios$valoraciones.ahp.ordenadas
```

```
## Costo alquiler Tráfico Visibilidad Accesibilidad Competencia
## 0.41664094 0.28410750 0.15372181 0.09372309 0.05180667
```

CONCLUSIÓN: vemos que el costo de alquiler es el criterio más importante para seleccionar la ubicación del nuevo restaurante con una ponderación de 0.41664094 (41,66%), luego es el criterio que tiene más peso en la decisión final seguido del tráfico con una ponderación de 0.28410750 (28,41%). El criterio con menor peso es la competencia con una ponderación de 0.05180667 (5,18%), lo que indica que es el criterio menos importante para la cadena de restaurantes.

(c) Calcular las ponderaciones de las cinco alternativas en función de cada criterio.

Los pesos locales son:

- Accesibilidad

```
pesos.accesibilidad = multicriterio.metodoAHP.variante3.basico(Xmat.accesibilidad)
pesos.accesibilidad$valoraciones.ahp
```

```
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.23950111 0.10152524 0.04636878 0.17042797 0.44217689
```

```
# Pesos ordenados
pesos.accesibilidad$valoraciones.ahp.ordenadas
```

```
## Ubicación E Ubicación A Ubicación D Ubicación B Ubicación C
## 0.44217689 0.23950111 0.17042797 0.10152524 0.04636878
```

CONCLUSIÓN: podemos ver que para el criterio de accesibilidad, la ubicación E es la alternativa más favorable ya que es la que presenta mayor peso (0.44217689) entre las demás.

- Costo de alquiler

```
pesos.costoAlquiler = multicriterio.metodoAHP.variante3.basico(Xmat.costoAlquiler)
pesos.costoAlquiler$valoraciones.ahp
```

```
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.23764670 0.10572839 0.04712193 0.13833329 0.47116969
```

```
# Pesos ordenados
pesos.costoAlquiler$valoraciones.ahp.ordenadas
```

```
## Ubicación E Ubicación A Ubicación D Ubicación B Ubicación C
## 0.47116969 0.23764670 0.13833329 0.10572839 0.04712193
```

CONCLUSIÓN: podemos ver que para el criterio de accesibilidad, la ubicación E es la alternativa más favorable ya que es la que presenta mayor peso (0.47116969) entre las demás.

- Tráfico

```
pesos.trafico = multicriterio.metodoAHP.variante3.basico(Xmat.trafico)
pesos.trafico$valoraciones.ahp
```

```
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.40582873 0.16012005 0.05875963 0.27942166 0.09586993
```

```
# Pesos ordenados
pesos.trafico$valoraciones.ahp.ordenadas
```

```
## Ubicación A Ubicación D Ubicación B Ubicación E Ubicación C
## 0.40582873 0.27942166 0.16012005 0.09586993 0.05875963
```

CONCLUSIÓN: podemos ver que para el criterio de tráfico, la ubicación A es la alternativa más favorable ya que es la que presenta mayor peso (0.40582873) entre las demás.

- Visibilidad

```
pesos.visibilidad = multicriterio.metodoAHP.variante3.basico(Xmat.visibilidad)
pesos.visibilidad$valoraciones.ahp
```

```
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.25307110 0.15437646 0.08728438 0.14187646 0.36339161
```

```
# Pesos ordenados
pesos.visibilidad$valoraciones.ahp.ordenadas
```

```
## Ubicación E Ubicación A Ubicación B Ubicación D Ubicación C
## 0.36339161 0.25307110 0.15437646 0.14187646 0.08728438
```

CONCLUSIÓN: podemos ver que para el criterio de visibilidad, la ubicación E es la alternativa más favorable ya que es la que presenta mayor peso (0.36339161) entre las demás.

- Competencia

```
pesos.competencia = multicriterio.metodoAHP.variante3.basico(Xmat.competencia)
pesos.competencia$valoraciones.ahp
```

```
## Ubicación A Ubicación B Ubicación C Ubicación D Ubicación E
## 0.35559701 0.15117917 0.08771080 0.35559701 0.04991601
```

```
# Pesos ordenados
pesos.competencia$valoraciones.ahp.ordenadas
```

```
## Ubicación A Ubicación D Ubicación B Ubicación C Ubicación E
## 0.35559701 0.35559701 0.15117917 0.08771080 0.04991601
```

CONCLUSIÓN: podemos ver que para el criterio de competencia, la ubicación A es la alternativa más favorable ya que es la que presenta mayor peso (0.35559701) entre las demás.

(d) Verificar la consistencia de las preferencias de la cadena de restaurantes.

- Consistencia de los criterios:

```
multicriterio.metodoAHP.coef.inconsistencia(Xmat.criterios)
```

```
## $lambda
## [1] 5.064154
##
## $m
## [1] 5
##
## $CI.coef.inconsistencia
## [1] 0.01603842
##
## $CA.aleatorio
## [1] 1.12
##
## $RI.coef.inconsistencia
## [1] 0.01432002
##
## $mensaje
## [1] "Consistencia aceptable"
```

Tenemos una consistencia aceptable, luego las preferencias expresadas en la matriz de comparación de criterios con consistentes.

- Consistencia de la accesibilidad:

```
multicriterio.metodoAHP.coef.inconsistencia(Xmat.accesibilidad)
```

```
## $lambda
## [1] 5.179419
##
## $m
## [1] 5
##
## $CI.coef.inconsistencia
## [1] 0.04485479
##
## $CA.aleatorio
## [1] 1.12
##
## $RI.coef.inconsistencia
## [1] 0.04004892
##
## $mensaje
## [1] "Consistencia aceptable"
```

Resulta una consistencia aceptable.

- Consistencia del costo de alquiler:

```
multicriterio.metodoAHP.coef.inconsistencia(Xmat.costoAlquiler)
```

```
## $lambda
## [1] 5.284862
##
## $m
## [1] 5
##
## $CI.coef.inconsistencia
## [1] 0.07121547
##
## $CA.aleatorio
## [1] 1.12
##
## $RI.coef.inconsistencia
## [1] 0.06358524
##
## $mensaje
## [1] "Consistencia aceptable"
```

Resulta una consistencia aceptable.

- Consistencia del tráfico

```
multicriterio.metodoAHP.coef.inconsistencia(Xmat.trafico)
```

```
## $lambda
## [1] 5.157601
##
## $m
## [1] 5
##
## $CI.coef.inconsistencia
## [1] 0.03940037
##
## $CA.aleatorio
## [1] 1.12
##
## $RI.coef.inconsistencia
## [1] 0.0351789
##
## $mensaje
## [1] "Consistencia aceptable"
```

Resulta una consistencia aceptable.

- Consistencia de la visibilidad

```
multicriterio.metodoAHP.coef.inconsistencia(Xmat.visibilidad)
```

```
## $lambda
## [1] 5.078212
##
## $m
## [1] 5
##
## $CI.coef.inconsistencia
## [1] 0.01955299
##
## $CA.aleatorio
## [1] 1.12
##
## $RI.coef.inconsistencia
## [1] 0.01745803
##
## $mensaje
## [1] "Consistencia aceptable"
```

Resulta una consistencia aceptable.

- Consistencia de la competencia

```
multicriterio.metodoAHP.coef.inconsistencia(Xmat.competencia)
```

```
## $lambda
## [1] 5.058618
##
## $m
## [1] 5
##
## $CI.coef.inconsistencia
## [1] 0.01465438
##
## $CA.aleatorio
## [1] 1.12
##
## $RI.coef.inconsistencia
## [1] 0.01308427
##
## $mensaje
## [1] "Consistencia aceptable"
```

Resulta una consistencia aceptable.

(e) Determinar la prioridad global de cada ubicación

Los pesos globales son:

```
tabla.pesosglobales.ahp = multicriterio.metodoAHP.pesosglobales_entabla(
  pesos.criterios$valoraciones.ahp,
  rbind(pesos.accesibilidad$valoraciones.ahp,
        pesos.costoAlquiler$valoraciones.ahp,
        pesos.trafico$valoraciones.ahp,
        pesos.visibilidad$valoraciones.ahp,
        pesos.competencia$valoraciones.ahp))
knitr::kable(tabla.pesosglobales.ahp)
```

	Accesibilidad	Costo alquiler	Tráfico	Visibilidad	Competencia	Ponderadores Globales
Ubicación A	0.2395011	0.2376467	0.4058287	0.2530711	0.3555970	0.2940840
Ubicación B	0.1015252	0.1057284	0.1601200	0.1543765	0.1511792	0.1306205
Ubicación C	0.0463688	0.0471219	0.0587596	0.0872844	0.0877108	0.0586343
Ubicación D	0.1704280	0.1383333	0.2794217	0.1418765	0.3555970	0.1932259
Ubicación E	0.4421769	0.4711697	0.0958699	0.3633916	0.0499160	0.3234353
Ponder.Criterios	0.0937231	0.4166409	0.2841075	0.1537218	0.0518067	NA

Expresada en porcentajes:

```
knitr::kable(tabla.pesosglobales.ahp*100,digits = 2)
```

	Accesibilidad	Costo alquiler	Tráfico	Visibilidad	Competencia	Ponderadores Globales
Ubicación A	23.95	23.76	40.58	25.31	35.56	29.41
Ubicación B	10.15	10.57	16.01	15.44	15.12	13.06
Ubicación C	4.64	4.71	5.88	8.73	8.77	5.86
Ubicación D	17.04	13.83	27.94	14.19	35.56	19.32
Ubicación E	44.22	47.12	9.59	36.34	4.99	32.34
Ponder.Criterios	9.37	41.66	28.41	15.37	5.18	NA

Los pesos de las ramas son:

```
vPn1 = tabla.pesosglobales.ahp[6,1:5]
mPn2 = tabla.pesosglobales.ahp[1:5,1:5]
mPondRamasb = mPn2*matrix(vPn1,ncol=5,nrow=5,byrow = T)
mPondRamas = tabla.pesosglobales.ahp
mPondRamas[1:5,1:5] = mPondRamasb
mPondRamas[6,6] = 1
knitr::kable(mPondRamas)
```

	Accesibilidad	Costo alquiler	Tráfico	Visibilidad	Competencia	Ponderadores Globales
Ubicación A	0.0224468	0.0990133	0.1152990	0.0389025	0.0184223	0.2940840
Ubicación B	0.0095153	0.0440508	0.0454913	0.0237310	0.0078321	0.1306205
Ubicación C	0.0043458	0.0196329	0.0166941	0.0134175	0.0045440	0.0586343
Ubicación D	0.0159730	0.0576353	0.0793858	0.0218095	0.0184223	0.1932259
Ubicación E	0.0414422	0.1963086	0.0272374	0.0558612	0.0025860	0.3234353
Ponder.Criterios	0.0937231	0.4166409	0.2841075	0.1537218	0.0518067	1.0000000

Expresados en porcentajes:

```
knitr::kable(mPondRamas*100,digits = 2)
```

	Accesibilidad	Costo alquiler	Tráfico	Visibilidad	Competencia	Ponderadores Globales
Ubicación A	2.24	9.90	11.53	3.89	1.84	29.41
Ubicación B	0.95	4.41	4.55	2.37	0.78	13.06
Ubicación C	0.43	1.96	1.67	1.34	0.45	5.86
Ubicación D	1.60	5.76	7.94	2.18	1.84	19.32
Ubicación E	4.14	19.63	2.72	5.59	0.26	32.34
Ponder.Criterios	9.37	41.66	28.41	15.37	5.18	100.00

CONCLUSIÓN: finalmente se puede decir que la ubicación E, una zona residencial de alto nivel económico, es la mejor opción para abrir el nuevo restaurante, con una ponderación global de 0.3234353 (32,34%). Esto indica que, a partir de los criterios evaluados (accesibilidad, costo de alquiler, tráfico, visibilidad y competencia), esta ubicación es la que mejor cumple con los requisitos de la cadena en comparación con las demás alternativas que poseen ponderaciones globales más bajas. La peor alternativa para el nuevo restaurante sería la ubicación C, la zona de oficinas, con una ponderación global de 0.0586343 (5,86%).

Mediante la librería AHP

Los datos están recogidos en el archivo llamado “problema.ahp”. Lo cargamos:

```
library(ahp)
ahp_problema = ahp::Load("problema.ahp")
ahp_problema
```



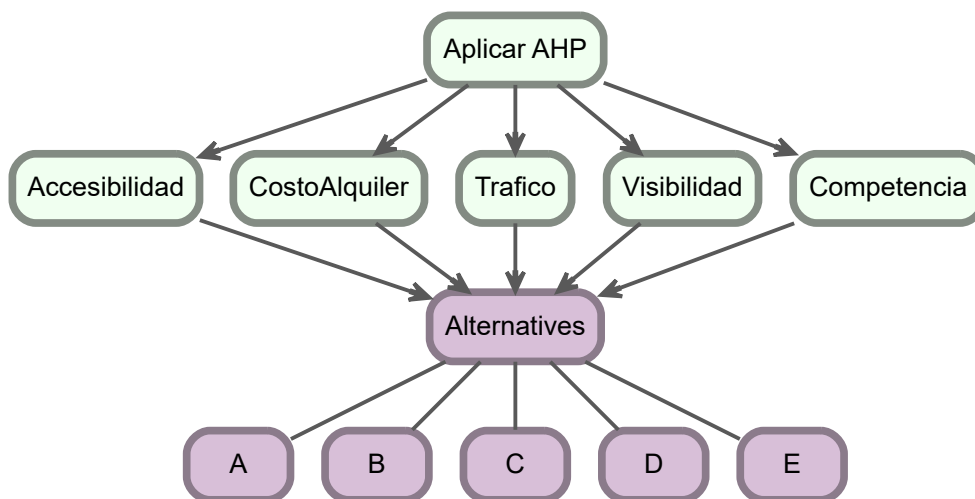
```

##          levelName
## 1  Aplicar AHP
## 2  |--Accesibilidad
## 3  |  |--A
## 4  |  |--B
## 5  |  |--C
## 6  |  |--D
## 7  |  °--E
## 8  |--CostoAlquiler
## 9  |  |--A
## 10 |  |--B
## 11 |  |--C
## 12 |  |--D
## 13 |  °--E
## 14 |--Trafico
## 15 |  |--A
## 16 |  |--B
## 17 |  |--C
## 18 |  |--D
## 19 |  °--E
## 20 |--Visibilidad
## 21 |  |--A
## 22 |  |--B
## 23 |  |--C
## 24 |  |--D
## 25 |  °--E
## 26 °--Competencia
## 27 |  |--A
## 28 |  |--B
## 29 |  |--C
## 30 |  |--D
## 31 |  °--E

```

(a) Identificar el diagrama de jerarquías de este problema de decisión.

```
Visualize(ahp_problema)
```



(b) Calcular las ponderaciones asociadas a cada criterio manejado.

(c) Calcular las ponderaciones de las cinco alternativas en función de cada criterio.

(d) Verificar la consistencia de las preferencias de la cadena de restaurantes.

```

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

```

```
##          levelName  priority
## 1  Aplicar AHP          NA
## 2  |--Accesibilidad 0.09302977
## 3  |  |--A         0.24004292
## 4  |  |--B         0.09840194
## 5  |  |--C         0.04523536
## 6  |  |--D         0.16585046
## 7  |  °--E         0.45046931
## 8  |--CostoAlquiler 0.41942741
## 9  |  |--A         0.23493381
## 10 |  |--B         0.09888539
## 11 |  |--C         0.04535304
## 12 |  |--D         0.13094012
## 13 |  °--E         0.48988763
## 14 |--Tráfico       0.28357223
## 15 |  |--A         0.40852660
## 16 |  |--B         0.15587545
## 17 |  |--C         0.05740228
## 18 |  |--D         0.28446482
## 19 |  °--E         0.09373086
## 20 |--Visibilidad   0.15282320
## 21 |  |--A         0.25285749
## 22 |  |--B         0.15300443
## 23 |  |--C         0.08666579
## 24 |  |--D         0.14097604
## 25 |  °--E         0.36649624
## 26 °--Competencia   0.05114739
## 27 |  |--A         0.35705813
## 28 |  |--B         0.14991754
## 29 |  |--C         0.08672279
## 30 |  |--D         0.35705813
## 31 |  °--E         0.04924341
```

Lo analizamos:

```
Analyze(ahp_problema,variable="priority")
```

##		Priority	E	A	D	B	C	Inconsistency
## 1	Aplicar AHP	100.0%	NA	NA	NA	NA	NA	1.4%
## 2	--CostoAlquiler	41.9%	49.0%	23.5%	13.1%	9.9%	4.5%	6.4%
## 3	--Tráfico	28.4%	9.4%	40.9%	28.4%	15.6%	5.7%	3.6%
## 4	--Visibilidad	15.3%	36.6%	25.3%	14.1%	15.3%	8.7%	1.8%
## 5	--Accesibilidad	9.3%	45.0%	24.0%	16.6%	9.8%	4.5%	4.0%
## 6	°--Competencia	5.1%	4.9%	35.7%	35.7%	15.0%	8.7%	1.3%

CONCLUSIÓN: en cuanto a los criterios podemos decir que el más importante para la cadena de restaurantes es el costo de alquiler con un 41.9%, seguido del tráfico (28.4%) y siendo la competencia el criterio con menor importancia en la decisión final (5.1%).

Si nos centramos en cada uno de los criterios por separados, observamos un desglose porcentual de preferencia para las alternativas. Por ejemplo, para el criterio "CostoAlquiler", la ubicación E es la que mayor peso tiene con un porcentaje de 49.0% mientras que la ubicación con menor peso es la C (4.5%). Para el criterio "Tráfico", la mejor alternativa es la ubicación A (40.9%) y la peor vuelve a ser la ubicación C.

En general, los valores de la inconsistencia son aceptables ya que se encuentran todos por debajo del 10%, esto indica que el análisis AHP es fiable.

(e) Determinar la prioridad global de cada ubicación

```
Analyze(ahp_problema)
```

##		Weight	E	A	D	B	C	Inconsistency
## 1	Aplicar AHP	100.0%	33.2%	29.4%	19.1%	12.6%	5.7%	1.4%
## 2	--CostoAlquiler	41.9%	20.5%	9.9%	5.5%	4.1%	1.9%	6.4%
## 3	--Tráfico	28.4%	2.7%	11.6%	8.1%	4.4%	1.6%	3.6%
## 4	--Visibilidad	15.3%	5.6%	3.9%	2.2%	2.3%	1.3%	1.8%
## 5	--Accesibilidad	9.3%	4.2%	2.2%	1.5%	0.9%	0.4%	4.0%
## 6	°--Competencia	5.1%	0.3%	1.8%	1.8%	0.8%	0.4%	1.3%

Analizamos con la tabla mejorada:

```
aa= AnalyzeTable(ahp_problema)
formattable::as.htmlwidget(aa)
```

	Weight	E	A	D	B	C	Inconsistency
Aplicar AHP	100.0%	33.2%	29.4%	19.1%	12.6%	5.7%	1.4%
CostoAlquiler	41.9%	20.5%	9.9%	5.5%	4.1%	1.9%	6.4%
Trafico	28.4%	2.7%	11.6%	8.1%	4.4%	1.6%	3.6%
Visibilidad	15.3%	5.6%	3.9%	2.2%	2.3%	1.3%	1.8%
Accesibilidad	9.3%	4.2%	2.2%	1.5%	0.9%	0.4%	4.0%
Competencia	5.1%	0.3%	1.8%	1.8%	0.8%	0.4%	1.3%

```
aab=AnalyzeTable(ahp_problema,variable="priority")
formattable::as.htmlwidget(aab)
```

	Priority	E	A	D	B	C	Inconsistency
Aplicar AHP	100.0%						1.4%
CostoAlquiler	41.9%	49.0%	23.5%	13.1%	9.9%	4.5%	6.4%
Trafico	28.4%	9.4%	40.9%	28.4%	15.6%	5.7%	3.6%
Visibilidad	15.3%	36.6%	25.3%	14.1%	15.3%	8.7%	1.8%
Accesibilidad	9.3%	45.0%	24.0%	16.6%	9.8%	4.5%	4.0%
Competencia	5.1%	4.9%	35.7%	35.7%	15.0%	8.7%	1.3%

CONCLUSIÓN: podemos concluir diciendo que la ubicación E, una zona residencial de alto nivel económico, es la mejor alternativa combinando los cinco criterios evaluados para la cadena de restaurantes, con una prioridad global de 33.2% sobre las demás. Esto se debe principalmente a su buen desempeño en los criterios de costo de alquiler (49.0%, la mejor puntuación) y accesibilidad (45.0%). La segunda mejor opción es la ubicación A, que presenta la segunda prioridad global más alta de 29.4%, destacándose en el criterio de tráfico con un 40.9%.

Se puede decir que la peor de las alternativas barajadas por la cadena es la ubicación C, la zona de oficinas, con el menor peso global de 5.7%, debido a su bajo rendimiento en la mayoría de los criterios evaluados.

Conclusiones finales

El objetivo de este problema ha sido elegir una de las cinco ubicaciones presentadas y evaluadas por un total de cinco criterios que toman más o menos importancia en la decisión final. Se ha demostrado que el costo de alquiler ha sido el criterio con más peso, por lo que es considerado el criterio más importante en la toma de la decisión, seguido del tráfico y siendo la competencia el menos relevante de todos.

Tras haber estudiado la situación por tres métodos distintos (Promethee, Electre y AHP), podemos ver que las tres conclusiones no coinciden en la misma alternativa. Mientras que para Promethee y Electre se demuestra que la ubicación A sería la mejor, para el método AHP la alternativa óptima será la ubicación E. Tomando en cuenta la conclusión de la mayoría de los criterios de selección, la decisión final de la cadena será abrir el nuevo restaurante en una zona céntrica de la ciudad de Sevilla.

Esto se debe a que según el método que usemos para resolver un mismo problema de decisión multicriterio, con las mismas alternativas y mismos criterios, los resultados pueden variar debido a la funcionalidad y enfoque de cada uno de ellos. Promethee compara alternativas usando funciones de preferencia y considerando pesos y umbrales. Por otro lado, Electre analiza la dominancia mediante concordancia y discordancia y el método AHP prioriza los criterios en una jerarquía, evaluando consistencia en las comparaciones por pares.