

# **Selección de la ciudad europea más sostenible para vivir**

Benedetta Palmieri

## **Table of contents**

<b>1 Presentación</b>	<b>2</b>
<b>2 Método AHP</b>	<b>6</b>
2.1 Resolución con las funciones . . . . .	6
2.1.1 Estudio de la inconsistencia . . . . .	9
2.1.2 Pesos locales . . . . .	10
2.1.3 Pesos globales . . . . .	12
2.1.4 En forma de diagrama . . . . .	13
2.2 Resolución con paquete AHP . . . . .	14
<b>3 Método AHP - MULTINIVEL</b>	<b>21</b>
3.1 Resolución con las funciones . . . . .	23
3.2 Resolución con paquete AHP . . . . .	29
<b>4 Método ELECTRE</b>	<b>35</b>
4.1 Iteración 1 . . . . .	36
4.2 Iteración 2: reducir alternativas . . . . .	37
4.3 Iteración 3: reducir alpha . . . . .	38
<b>5 Método PROMETHEE</b>	<b>39</b>
5.1 Promethee I: . . . . .	40
5.2 Promethee II: . . . . .	42
5.3 Promethee I (medias): . . . . .	44
5.4 Promethee II (medias): . . . . .	45
<b>6 Conclusión</b>	<b>47</b>

# 1 Presentación

En la actualidad, las ciudades europeas afrontan el reto de avanzar hacia un desarrollo más sostenible, buscando un equilibrio entre el crecimiento económico, la protección del medio ambiente y el bienestar de sus habitantes. La sostenibilidad urbana se ha convertido, por tanto, en un tema clave de las políticas públicas y es un factor determinante en la calidad de vida de las personas.

El objetivo de este trabajo es **evaluar y comparar el nivel de sostenibilidad de varias ciudades europeas**, con el fin de seleccionar aquella que puede considerarse la **más sostenible para vivir**. Para ello, se aplican diferentes **métodos de decisión multicriterio** (AHP, ELECTRE y PROMETHEE) que permiten ponderar factores ambientales, sociales y económicos.

Los criterios utilizados se basan en indicadores objetivos y reconocidos a nivel internacional, obtenidos de la base de datos pública **Numbeo**, que ofrece información sobre coste de vida, poder adquisitivo, seguridad, calidad del sistema sanitario y nivel de contaminación en las principales ciudades del mundo.

Este tema resulta relevante por su importancia social y ambiental en un contexto donde el cambio climático y la calidad de vida son preocupaciones crecientes.

Se seleccionaron cinco capitales europeas (Copenhague, Berlín, París, Madrid y Roma) por representar distintas zonas y realidades de Europa. Los seis criterios elegidos (Cost of Living, Rent, Purchasing Power, Safety, Health Care y Pollution) permiten evaluar de forma integral la sostenibilidad y la calidad de vida en cada ciudad.

## Fuente de datos:

Los valores de los índices proceden de la base de datos Numbeo (consultada en inicio noviembre 2025). Es importante mencionar que los datos pueden variar ligeramente debido a las actualizaciones continuas: <https://www.numbeo.com/>

## Criterios (índices) y alternativas (ciudades) consideradas:

Table 1: Criterios considerados, escala e interpretación

Criterio	Escala	Tipo	Interpretación
Cost of Living Index	IR	Costo (min)	Mide el costo medio de bienes y servicios; valores altos indican ciudades más caras.
Rent Index	IR	Costo (min)	Refleja el nivel medio de los alquileres; valores altos significan alquileres más elevados.
Purchasing Power Index	IR	Beneficio (max)	Evaluá el poder adquisitivo medio en relación con los precios locales; valores altos indican mayor capacidad económica.
Safety Index	0–100	Beneficio (max)	Representa la percepción de seguridad ciudadana; valores altos indican ciudades más seguras.

<b>Health Care Index</b>	0–100	Beneficio (max)	Evalúa la calidad percibida del sistema sanitario (acceso, competencia, infraestructura).
<b>Pollution Index</b>	0–100	Costo (min)	Mide el nivel de contaminación ambiental; valores altos implican peores condiciones medioambientales.

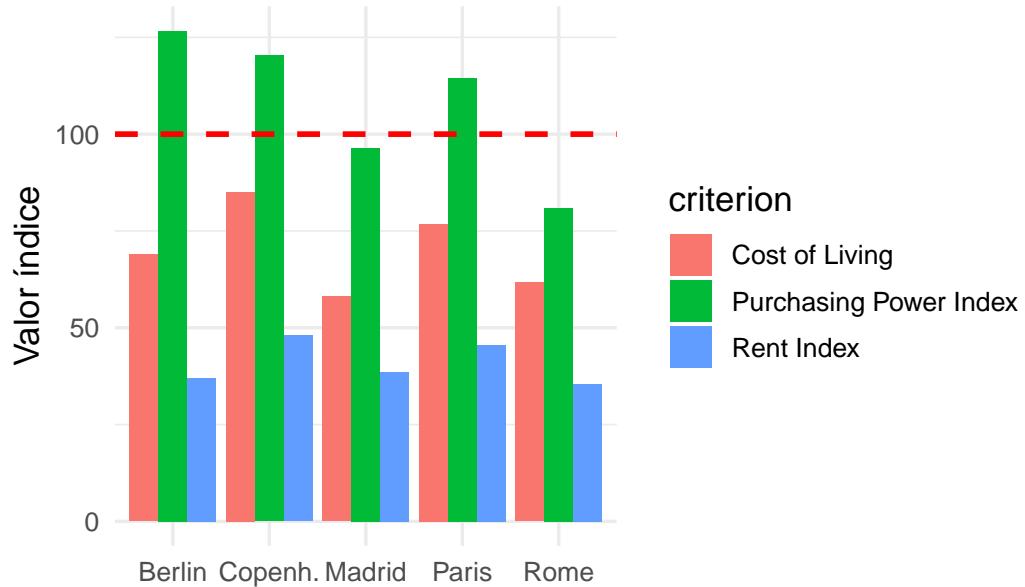
En escala orginal, IR indica indice relativo con base Nueva York = 100. Los criterios de costos son da minimizar, mientras que los de beneficios son da maximizar.

Table 2: Datos de las ciudades para cada índice

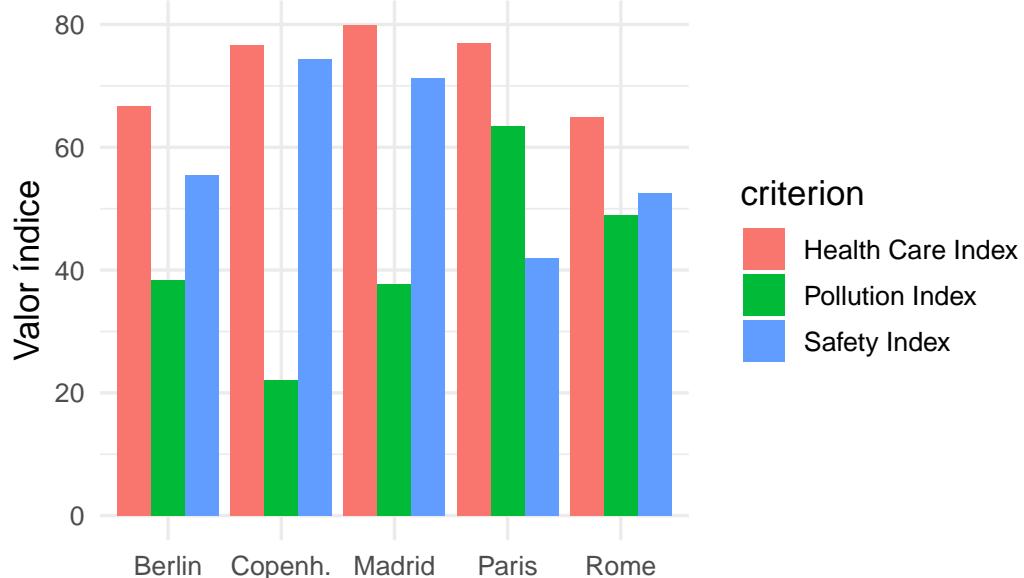
City	Cost.Living	Rent	Purchasing.Pwr	Safety	Health.Care	Pollution
Copenhagen	85.1	48.0	120.3	74.3	76.7	22.1
Berlin	68.9	37.1	126.7	55.5	66.7	38.4
Paris	76.8	45.5	114.5	42.0	76.9	63.5
Madrid	58.1	38.5	96.5	71.3	79.9	37.7
Rome	61.7	35.4	81.0	52.6	64.9	48.9

Representación gráfica para observar las diferencias entre las ciudades de forma visual.

## Comparación por los criterios en escala IR



## Comparación por los criterios en escala 0–100



Transformo los criterios para que todos sean de **maximizar**, colocando un signo negativo delante de los valores que inicialmente correspondían a criterios de coste.

Table 3: Datos transformados a criterios de maximización

City	Cost.Living	Rent	Purchasing.Pwr	Safety	Health.Care	Pollution
Copenhagen	-85.1	-48.0	120.3	74.3	76.7	-22.1
Berlin	-68.9	-37.1	126.7	55.5	66.7	-38.4
Paris	-76.8	-45.5	114.5	42.0	76.9	-63.5
Madrid	-58.1	-38.5	96.5	71.3	79.9	-37.7
Rome	-61.7	-35.4	81.0	52.6	64.9	-48.9

Normalizo los valores con el **método Nadir**, para que todos los criterios sean expresados en una escala común entre 0 y 1.

```
X = matrix( c(-85.1, -68.9, -76.8, -58.1, -61.7,
              -48.0, -37.1, -45.5, -38.5, -35.4,
              120.3, 126.7, 114.5, 96.5, 81.0,
              74.3, 55.5, 42.0, 71.3, 52.6,
              76.7, 66.7, 76.9, 79.9, 64.9,
              -22.1, -38.4, -63.5, -37.7, -48.9), nrow=5, ncol=6, byrow=F);
colnames(X) = c('Cost of Living Index', 'Rent Index', 'Purchasing Power Index',
               'Safety Index', 'Health care Index', 'Pollution Index');
rownames(X) = c('Copenhagen', 'Berlin', 'Paris', 'Madrid', 'Rome');
```

```

vcol.max = apply(X,2,max)
vcol.min = apply(X,2,min)

m.vol.max = matrix(vcol.max,5,6,byrow=T)
m.vol.min = matrix(vcol.min,5,6,byrow=T)

nueva.X = (X-m.vol.min)/(m.vol.max-m.vol.min)

```

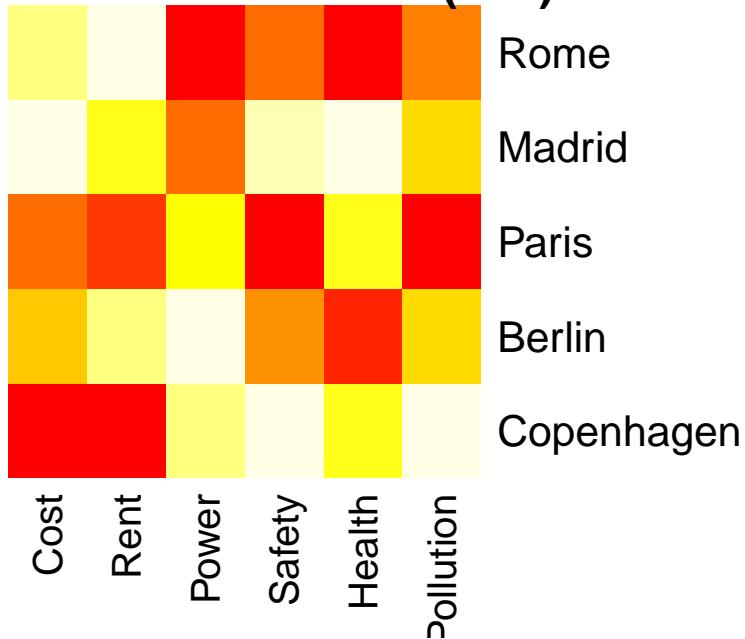
Table 4: Matriz de valores normalizados

	Cost of Living Index	Rent Index	Purchasing Power Index	Safety Index	Health care Index	Pollution Index
Copenhagen	0.0000000	0.0000000	0.8599562	1.0000000	0.7866667	1.0000000
Berlin	0.6000000	0.8650794	1.0000000	0.4179567	0.1200000	0.6062802
Paris	0.3074074	0.1984127	0.7330416	0.0000000	0.8000000	0.0000000
Madrid	1.0000000	0.7539683	0.3391685	0.9071207	1.0000000	0.6231884
Rome	0.8666667	1.0000000	0.0000000	0.3281734	0.0000000	0.3526570

#### Rapresentación grafica de los valores normalizados.

Los valores proximos a 1 indican un mejor comportamiento, mientras que los valores proximos a 0 representan un rendimiento mas bajo en el criterio correspondiente.

#### Valores normalizados (0-1)



Se cargan los archivos que contienen las funciones necesarias para el análisis.

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

Loading required package: shape

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

## 2 Método AHP

### 2.1 Resolución con las funciones

Para construir las matrices de comparación entre las ciudades para cada criterio, se transformaron los valores de los índices de Numbeo a la escala de Saaty (1–9), asignando un valor entre 1 y 3 cuando la diferencia es moderada, 5 cuando es fuerte, y entre 7 y 9 cuando es muy fuerte o extrema.

En la matriz de comparación entre los criterios, los valores asignados reflejan la importancia relativa de cada criterio según la escala de Saaty (1–9), considerando su relevancia en la evaluación de la sostenibilidad urbana.

```
tb_criterios = multicriterio.crea.matrizvaloraciones_mej(
  c(2, 1/2, 1/2, 1/2, 1/3,
    1/3, 1/4, 1/4, 1/6,
    1/2, 1/2, 1/2,
    1, 1/2,
    1),
  numalternativas = 6,
  v.nombres.alternativas =
  c('Cost of Living Index','Rent Index',
    'Purchasing Power Index', 'Safety Index',
    'Health care Index','Pollution Index')
);

tb_cost = multicriterio.crea.matrizvaloraciones_mej(
  c(1/3, 1/2, 1/5, 1/4,
    3, 1/3, 1/2,
    1/4, 1/3,
    2),
```

```

        numalternativas = 5,
        v.nombres.alternativas =
        c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_rent = multicriterio.crea.matrizvaloraciones_mej(
    c(1/3, 1/2, 1/3, 1/4,
      3, 2, 1/2,
      1/2, 1/3,
      1/2),
    numalternativas = 5,
    v.nombres.alternativas =
    c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_purch = multicriterio.crea.matrizvaloraciones_mej(
    c(1/2, 2, 5, 6,
      3, 6, 7,
      3, 5,
      2),
    numalternativas = 5,
    v.nombres.alternativas =
    c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_safety = multicriterio.crea.matrizvaloraciones_mej(
    c(4, 5, 2, 5,
      2, 1/3, 1,
      1/3, 1/2,
      3),
    numalternativas = 5,
    v.nombres.alternativas =
    c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_health = multicriterio.crea.matrizvaloraciones_mej(
    c(3, 1, 1/2, 4,
      1/3, 1/4, 2,
      1/2, 4,
      5),
    numalternativas = 5,
    v.nombres.alternativas =
    c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

```

```

tb_pollution = multicriterio.crea.matrizvaloraciones_mej(
    c(3,7,3,5,
      5,1,3,
      1/5,1/3,
      2),
    numalternativas = 5,
    v.nombres.alternativas =
    c("Copenhagen","Berlin", "Paris", "Madrid", "Rome"));

```

Table 5: Matriz de criterios

	Cost of Living Index	Rent Index	Purchasing Power Index	Safety Index	Health care Index	Pollution Index
<b>Cost of Living Index</b>	1.0	2	0.5000000	0.50	0.50	0.3333333
<b>Rent Index</b>	0.5	1	0.3333333	0.25	0.25	0.1666667
<b>Purchasing Power Index</b>	2.0	3	1.0000000	0.50	0.50	0.5000000
<b>Safety Index</b>	2.0	4	2.0000000	1.00	1.00	0.5000000
<b>Health care Index</b>	2.0	4	2.0000000	1.00	1.00	1.0000000
<b>Pollution Index</b>	3.0	6	2.0000000	2.00	1.00	1.0000000

Table 6: Matriz Cost Of Living Index

	Copenhagen	Berlin	Paris	Madrid	Rome
<b>Copenhagen</b>	1	0.3333333	0.5	0.2000000	0.2500000
<b>Berlin</b>	3	1.0000000	3.0	0.3333333	0.5000000
<b>Paris</b>	2	0.3333333	1.0	0.2500000	0.3333333
<b>Madrid</b>	5	3.0000000	4.0	1.0000000	2.0000000
<b>Rome</b>	4	2.0000000	3.0	0.5000000	1.0000000

Table 7: Matriz Rent Index

	Copenhagen	Berlin	Paris	Madrid	Rome
<b>Copenhagen</b>	1	0.3333333	0.5	0.3333333	0.2500000
<b>Berlin</b>	3	1.0000000	3.0	2.0000000	0.5000000
<b>Paris</b>	2	0.3333333	1.0	0.5000000	0.3333333
<b>Madrid</b>	3	0.5000000	2.0	1.0000000	0.5000000
<b>Rome</b>	4	2.0000000	3.0	2.0000000	1.0000000

Table 8: Matriz Purchasing Power Index

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	1.0000000	0.5000000	2.0000000	5.0	6
Berlin	2.0000000	1.0000000	3.0000000	6.0	7
Paris	0.5000000	0.3333333	1.0000000	3.0	5
Madrid	0.2000000	0.1666667	0.3333333	1.0	2
Rome	0.1666667	0.1428571	0.2000000	0.5	1

Table 9: Matriz Safety Index

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	1.00	4.0	5	2.0000000	5.0
Berlin	0.25	1.0	2	0.3333333	1.0
Paris	0.20	0.5	1	0.3333333	0.5
Madrid	0.50	3.0	3	1.0000000	3.0
Rome	0.20	1.0	2	0.3333333	1.0

Table 10: Matriz Health Index

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	1.0000000	3.0	1.0000000	0.50	4
Berlin	0.3333333	1.0	0.3333333	0.25	2
Paris	1.0000000	3.0	1.0000000	0.50	4
Madrid	2.0000000	4.0	2.0000000	1.00	5
Rome	0.2500000	0.5	0.2500000	0.20	1

Table 11: Matriz Pollution Index

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	1.0000000	3.0000000	7	3.0	5.0000000
Berlin	0.3333333	1.0000000	5	1.0	3.0000000
Paris	0.1428571	0.2000000	1	0.2	0.3333333
Madrid	0.3333333	1.0000000	5	1.0	2.0000000
Rome	0.2000000	0.3333333	3	0.5	1.0000000

### 2.1.1 Estudio de la inconsistencia

```
Inc_criterios = multicriterio.metodoAHP.coef.inconsistencia(tb_criterios)
c(Inc_criterios$mensaje, round(Inc_criterios$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0132"
```

```
Inc_cost = multicriterio.metodoAHP.coef.inconsistencia(tb_cost)
c(Inc_cost$mensaje, round(Inc_cost$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0269"
```

```
Inc_rent = multicriterio.metodoAHP.coef.inconsistencia(tb_rent)
c(Inc_rent$mensaje, round(Inc_rent$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.024"
```

```
Inc_purch = multicriterio.metodoAHP.coef.inconsistencia(tb_purch)
c(Inc_purch$mensaje, round(Inc_purch$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0203"
```

```
Inc_safety = multicriterio.metodoAHP.coef.inconsistencia(tb_safety)
c(Inc_safety$mensaje, round(Inc_safety$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0175"
```

```
Inc_health = multicriterio.metodoAHP.coef.inconsistencia(tb_health)
c(Inc_health$mensaje, round(Inc_health$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0119"
```

```
Inc_pollution = multicriterio.metodoAHP.coef.inconsistencia(tb_pollution)
c(Inc_pollution$mensaje, round(Inc_pollution$RI.coef.inconsistencia,4))
```

```
[1] "Consistencia aceptable" "0.0242"
```

El estudio de la inconsistencia sirve para verificar la coherencia entre los juicios de las matrices AHP. En este caso, todos los coeficientes de inconsistencia son bajos y están por debajo de 0.1; por tanto los juicios pueden considerarse coherentes.

### 2.1.2 Pesos locales

Cálculo de los **pesos locales** de los criterios entre sí y de las alternativas dentro de cada criterio.

```
p_criterios = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_criterios)
p_criterios$valoraciones.ahp
```

Cost of Living Index	Rent Index Purchasing Power Index	
0.09401851	0.04888298 0.13674234	
Safety Index	Health care Index	Pollution Index
0.20253783	0.22648965	0.29132868

```
p_cost = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_cost)
p_cost$valoraciones.ahp
```

Copenhagen	Berlin	Paris	Madrid	Rome
0.06093247	0.17485588	0.08988382	0.41456792	0.25975991

```
p_rent = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_rent)
p_rent$valoraciones.ahp
```

Copenhagen	Berlin	Paris	Madrid	Rome
0.07223072	0.26652950	0.10904732	0.18321436	0.36897810

```
p_purch = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_purch)
p_purch$valoraciones.ahp
```

Copenhagen	Berlin	Paris	Madrid	Rome
0.28228349	0.43589670	0.17182955	0.06671487	0.04327540

```
p_safety = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_safety)
p_safety$valoraciones.ahp
```

Copenhagen	Berlin	Paris	Madrid	Rome
0.45134332	0.10933371	0.07001099	0.26442217	0.10488981

```
p_health = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_health)
p_health$valoraciones.ahp
```

```
Copenhagen      Berlin      Paris      Madrid      Rome
0.23028345 0.09095138 0.23028345 0.38962764 0.05885407
```

```
p_pollution = multicriterio.metodoAHP.variente1.autovectormayorautovalor(
    tb_pollution)
p_pollution$valoraciones.ahp
```

```
Copenhagen      Berlin      Paris      Madrid      Rome
0.47263799 0.20447773 0.04305389 0.18613571 0.09369468
```

### 2.1.3 Pesos globales

Calculo de los **pesos globales**:

```
p_global = multicriterio.metodoAHP.pesosglobales_entabla(
    p_criterios$valoraciones.ahp,
    rbind(p_cost$valoraciones.ahp, p_rent$valoraciones.ahp,
          p_purch$valoraciones.ahp, p_safety$valoraciones.ahp,
          p_health$valoraciones.ahp, p_pollution$valoraciones.ahp)
    )
p_global
```

	Cost of Living Index	Rent Index	Purchasing Power Index
Copenhagen	0.06093247	0.07223072	0.28228349
Berlin	0.17485588	0.26652950	0.43589670
Paris	0.08988382	0.10904732	0.17182955
Madrid	0.41456792	0.18321436	0.06671487
Rome	0.25975991	0.36897810	0.04327540
Ponder.Criterios	0.09401851	0.04888298	0.13674234
	Safety Index	Health care Index	Pollution Index
Copenhagen	0.45134332	0.23028345	0.47263799
Berlin	0.10933371	0.09095138	0.20447773
Paris	0.07001099	0.23028345	0.04305389
Madrid	0.26442217	0.38962764	0.18613571
Rome	0.10488981	0.05885407	0.09369468
Ponder.Criterios	0.20253783	0.22648965	0.29132868
	Ponderadores Globales		
Copenhagen	0.3291237		
Berlin	0.1913880		
Paris	0.1161572		

Madrid	0.2530847
Rome	0.1102465
Ponder.Criterios	NA

Los pesos globales representan la importancia compleja de cada alternativa con respecto al objetivo final.

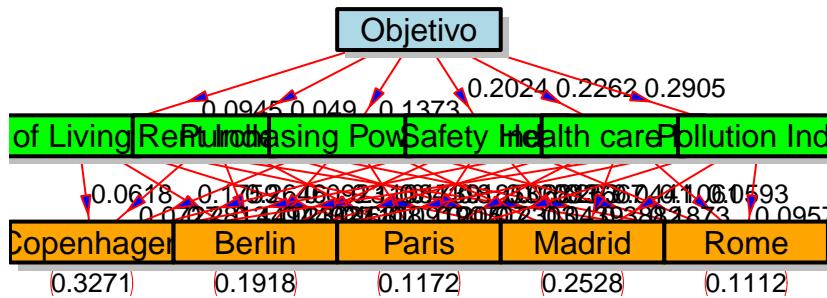
**Copenhagen** es la alternativa que presenta el mayor peso global, lo que indica que es la opción más favorable según el método AHP.

#### 2.1.4 En forma de diagrama

```
matriznivel = array(NA, dim = c(5,5,6))
matriznivel[,,1] = tb_cost
matriznivel[,,2] = tb_rent
matriznivel[,,3] = tb_purch
matriznivel[,,4] = tb_safety
matriznivel[,,5] = tb_health
matriznivel[,,6] = tb_pollution
dimnames(matriznivel)[[1]] = c("Copenhagen","Berlin","Paris","Madrid","Rome")
dimnames(matriznivel)[[2]] = c("Copenhagen","Berlin","Paris","Madrid","Rome")

multicriterio.metodoahp.diagrama(tb_criterios,matriznivel)
```

## Estructura Jerárquica (AHP)



**Copenhagen** es la alternativa que presenta el mayor peso global, lo que indica que es la opción más favorable según el método AHP.

## 2.2 Resolución con paquete AHP

```
library(ahp)
ahp_ej = ahp::Load("ej01_ahp.ahp")
ahp_ej
```

```
      levelName
1 Elegir la ciudad europea más sostenible para vivir
2   |--Cost of Living Index
3     |--Copenhagen
4     |--Berlin
5     |--Paris
6     |--Madrid
7     |--Rome
8   |--Rent Index
9     |--Copenhagen
10    |--Berlin
11    |--Paris
```

```

12 |     |--Madrid
13 |     |--Rome
14 |--Purchasing Power Index
15 |     |--Copenhagen
16 |     |--Berlin
17 |     |--Paris
18 |     |--Madrid
19 |     |--Rome
20 |--Safety Index
21 |     |--Copenhagen
22 |     |--Berlin
23 |     |--Paris
24 |     |--Madrid
25 |     |--Rome
26 |--Health care Index
27 |     |--Copenhagen
28 |     |--Berlin
29 |     |--Paris
30 |     |--Madrid
31 |     |--Rome
32 |--Pollution Index
33 |     |--Copenhagen
34 |     |--Berlin
35 |     |--Paris
36 |     |--Madrid
37 |     |--Rome

```

```

# Calcular las prioridades
Calculate(ahp_ej)
print(ahp_ej, priority = function(x) x$parent$priority["Total", x$name])

```

	levelName	priority
1	Elegir la ciudad europea más sostenible para vivir	NA
2	--Cost of Living Index	0.09401851
3	--Copenhagen	0.06093247
4	--Berlin	0.17485588
5	--Paris	0.08988382
6	--Madrid	0.41456792
7	--Rome	0.25975991
8	--Rent Index	0.04888298
9	--Copenhagen	0.07223072
10	--Berlin	0.26652950

```

11 |     |--Paris          0.10904732
12 |     |--Madrid         0.18321436
13 |     |--Rome           0.36897810
14 |--Purchasing Power Index 0.13674234
15 |     |--Copenhagen    0.28228349
16 |     |--Berlin          0.43589670
17 |     |--Paris           0.17182955
18 |     |--Madrid          0.06671487
19 |     |--Rome            0.04327540
20 |--Safety Index          0.20253783
21 |     |--Copenhagen    0.45134332
22 |     |--Berlin          0.10933371
23 |     |--Paris           0.07001099
24 |     |--Madrid          0.26442217
25 |     |--Rome            0.10488981
26 |--Health care Index    0.22648965
27 |     |--Copenhagen    0.23028345
28 |     |--Berlin          0.09095138
29 |     |--Paris           0.23028345
30 |     |--Madrid          0.38962764
31 |     |--Rome            0.05885407
32 |--Pollution Index      0.29132868
33 |     |--Copenhagen    0.47263799
34 |     |--Berlin          0.20447773
35 |     |--Paris           0.04305389
36 |     |--Madrid          0.18613571
37 |     |--Rome            0.09369468

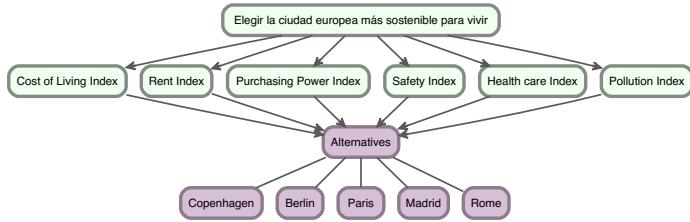
```

```

# Visualizar la jerarquía
Visualize(ahp_ej)

```

Google Chrome was not found. Try setting the `CHROMOTE\_CHROME` environment variable to the ex



```
# Analizar
Analyze(ahp_ej,variable = "priority" )
```

		Priority	Copenhagen	Madrid
1	Elegir la ciudad europea más sostenible para vivir	100.0%	NA	NA
2	--Pollution Index	29.1%	47.3%	18.6%
3	--Health care Index	22.6%	23.0%	39.0%
4	--Safety Index	20.3%	45.1%	26.4%
5	--Purchasing Power Index	13.7%	28.2%	6.7%
6	--Cost of Living Index	9.4%	6.1%	41.5%
7	°--Rent Index	4.9%	7.2%	18.3%
	Berlin	NA	NA	Inconsistency
1	NA	NA	NA	1.3%
2	20.4%	4.3%	9.4%	2.4%
3	9.1%	23.0%	5.9%	1.2%
4	10.9%	7.0%	10.5%	1.8%
5	43.6%	17.2%	4.3%	2.1%
6	17.5%	9.0%	26.0%	2.7%
7	26.7%	10.9%	36.9%	2.4%

```
Analyze(ahp_ej)
```

			Weight	Copenhagen	Madrid
1	Elegir la ciudad europea más sostenible para vivir		100.0%	32.9%	25.3%
2	--Pollution Index		29.1%	13.8%	5.4%
3	--Health care Index		22.6%	5.2%	8.8%
4	--Safety Index		20.3%	9.1%	5.4%
5	--Purchasing Power Index		13.7%	3.9%	0.9%
6	--Cost of Living Index		9.4%	0.6%	3.9%
7	--Rent Index		4.9%	0.4%	0.9%
	Berlin	Paris	Rome	Inconsistency	
1	19.1%	11.6%	11.0%	1.3%	
2	6.0%	1.3%	2.7%	2.4%	
3	2.1%	5.2%	1.3%	1.2%	
4	2.2%	1.4%	2.1%	1.8%	
5	6.0%	2.3%	0.6%	2.1%	
6	1.6%	0.8%	2.4%	2.7%	
7	1.3%	0.5%	1.8%	2.4%	

```
# Analizar con Tabla Mejorada
aa = AnalyzeTable(ahp_ej)
formattable::as.htmlwidget(aa)
```

	Weight	Copenhagen	Madrid	Berlin	Paris	Rome	Inconsistency
Elegir la ciudad europea más sostenible para vivir	100.0%	32.9%	25.3%	19.1%	11.6%	11.0%	1.3%
Pollution Index	29.1%	13.8%	5.4%	6.0%	1.3%	2.7%	2.4%
Health care Index	22.6%	5.2%	8.8%	2.1%	5.2%	1.3%	1.2%
Safety Index	20.3%	9.1%	5.4%	2.2%	1.4%	2.1%	1.8%
Purchasing Power Index	13.7%	3.9%	0.9%	6.0%	2.3%	0.6%	2.1%
Cost of Living Index	9.4%	0.6%	3.9%	1.6%	0.8%	2.4%	2.7%
Rent Index	4.9%	0.4%	0.9%	1.3%	0.5%	1.8%	2.4%

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

	Priority	Copenhagen	Madrid	Berlin	Paris	Rome	Inconsistency
Elegir la ciudad europea más sostenible para vivir	100.0%						1.3%
Pollution Index	29.1%	47.3%	18.6%	20.4%	4.3%	9.4%	2.4%
Health care Index	22.6%	23.0%	39.0%	9.1%	23.0%	5.9%	1.2%
Safety Index	20.3%	45.1%	26.4%	10.9%	7.0%	10.5%	1.8%
Purchasing Power Index	13.7%	28.2%	6.7%	43.6%	17.2%	4.3%	2.1%
Cost of Living Index	9.4%	6.1%	41.5%	17.5%	9.0%	26.0%	2.7%
Rent Index	4.9%	7.2%	18.3%	26.7%	10.9%	36.9%	2.4%

Los resultados del método AHP muestran que los criterios con mayor peso global son el Pollution Index (29.1%), el Health Care Index (22.6%) y el Safety Index (20.3%), lo que indica que los aspectos ambientales y sociales tienen una influencia más fuerte en la decisión que los factores puramente económicos.

En cuanto a las alternativas, la ciudad con mayor peso global es **Copenhagen (32.9%)**, seguida de Madrid (25.3%) y Berlín (19.1%). París (11.6%) y Roma (11.0%) se sitúan en las últimas posiciones.

Estos resultados reflejan que Copenhagen destaca por su bajo nivel de contaminación, alta seguridad y buena calidad del sistema sanitario, compensando su mayor coste de vida.

Las inconsistencias de todas las matrices se mantienen por debajo del umbral de 0.1, lo que confirma la coherencia de los juicios emitidos.

### 3 Método AHP - MULTINIVEL

Dado que el “**Cost of Living Index**” incluye dos componentes principales:

- “**Groceries Index**”: evalúa el costo de los productos de supermercado en una ciudad en comparación con Nueva York.
- “**Restaurants Index**”: compara los precios de comidas y bebidas en restaurantes y bares entre una ciudad y Nueva York, reflejando el gasto al comer fuera.

(En ambos casos se utiliza el índice IR relativo y ambos son costos que deben minimizarse)

Por esta razón, transformo el ejercicio **AHP** en un **AHP multinivel**, incorporando estos dos subcriterios dentro del criterio principal Cost of Living Index.

Table 12: Datos de la ciudades para cada indice

CITY	Cost.Living	Grocery	Restaurant	Rent	Purch.Pwr	Safety	Health.care	Pollution
Copenh.	85.1	75.4	101.6	48.0	120.3	74.3	76.7	22.1
Berlin	68.9	60.7	67.2	37.1	126.7	55.5	66.7	38.4
Paris	76.8	75.6	70.8	45.5	114.5	42.0	76.9	63.5
Madrid	58.1	53.0	60.8	38.5	96.5	71.3	79.9	37.7
Rome	61.7	62.8	62.9	35.4	81.0	52.6	64.9	48.9

Transformo los criterios para que todos sean de **maximizar**, colocando un signo negativo delante de los valores que inicialmente correspondían a criterios de costo.

Table 13: Datos transformados a criterios de maximización

CITY	Cost.Living	Grocery	Restaurant	Rent	Purch.Pwr	Safety	Health.care	Pollution
Copenh.	-85.1	-75.4	-101.6	-48.0	120.3	74.3	76.7	-22.1
Berlin	-68.9	-60.7	-67.2	-37.1	126.7	55.5	66.7	-38.4
Paris	-76.8	-75.6	-70.8	-45.5	114.5	42.0	76.9	-63.5
Madrid	-58.1	-53.0	-60.8	-38.5	96.5	71.3	79.9	-37.7
Rome	-61.7	-62.8	-62.9	-35.4	81.0	52.6	64.9	-48.9

Normalizo los valores con el **método Nadir**, para que todos los criterios sean expresados en una escala común entre 0 y 1.

```
X = matrix( c(-85.1, -68.9, -76.8, -58.1, -61.7,
-75.4, -60.7, -75.6, -53.0, -62.8,
-101.6, -67.2, -70.8, -60.8, -62.9,
-48.0, -37.1, -45.5, -38.5, -35.4,
120.3, 126.7, 114.5, 96.5, 81.0,
74.3, 55.5, 42.0, 71.3, 52.6,
```

```

    76.7, 66.7, 76.9, 79.9, 64.9,
    -22.1, -38.4, -63.5, -37.7, -48.9), nrow=5, ncol=8, byrow=F);
colnames(X) = c('Cost of Living Index', 'Grocery Index', 'Restaurant Index',
               'Rent Index', 'Purchasing Power Index', 'Safety Index',
               'Health care Index', 'Pollution Index');
rownames(X) = c('Copenhagen', 'Berlin', 'Paris', 'Madrid', 'Rome');

vcol.max = apply(X, 2, max)
vcol.min = apply(X, 2, min)

m.vol.max = matrix(vcol.max, 5, 8, byrow=T)
m.vol.min = matrix(vcol.min, 5, 8, byrow=T)

nueva.X.ml = (X - m.vol.min) / (m.vol.max - m.vol.min)

```

Table 14: Matriz de valores normalizados

	Cost of Living Index	Grocery Index	Restaurant Index	Rent Index	Purchasing Power Index	Safety Index	Health care Index	Pollution Index
Copenhagen	0.0000000	0.0088496	0.0000000	0.0000000	0.8599562	1.0000000	0.7866667	1.0000000
Berlin	0.6000000	0.6592920	0.8431373	0.8650794	1.0000000	0.4179567	0.1200000	0.6062802
Paris	0.3074074	0.0000000	0.7549020	0.1984127	0.7330416	0.0000000	0.8000000	0.0000000
Madrid	1.0000000	1.0000000	1.0000000	0.7539683	0.3391685	0.9071207	1.0000000	0.6231884
Rome	0.8666667	0.5663717	0.9485294	1.0000000	0.0000000	0.3281734	0.0000000	0.3526570

## Representación grafica

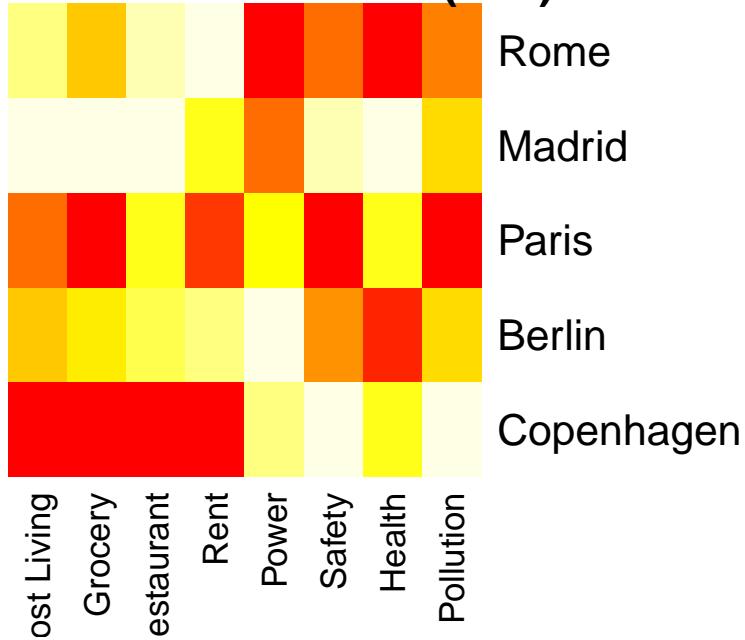
```

nueva.X2.ml <- nueva.X.ml
colnames(nueva.X2.ml) <- c("Cost Living", "Grocery", "Restaurant",
                           "Rent", "Power", "Safety", "Health", "Pollution")

heatmap(
  nueva.X2.ml,
  Rowv = NA, Colv = NA,
  col = heat.colors(20),
  scale = "none",
  main = "Valores normalizados (0-1)",
)

```

## Valores normalizados (0-1)



### 3.1 Resolución con las funciones

Se reutiliza parte del código utilizado en el método AHP (versión no multínivel), presentandolo de forma más compacta y omitiendo la impresión y las comprobaciones ya realizadas en la sección anterior.

```
tb_criterios = multicriterio.crea.matrizvaloraciones_mej(
    c(2, 1/2, 1/2, 1/2, 1/3,
      1/3, 1/4, 1/4, 1/6,
      1/2, 1/2, 1/2,
      1, 1/2,
      1),
    numalternativas = 6,
    v.nombres.alternativas =
    c('Cost of Living Index', 'Rent Index',
      'Purchasing Power Index', 'Safety Index',
      'Health care Index','Pollution Index')
  );

tb_cost = multicriterio.crea.matrizvaloraciones_mej(
    c(1/3, 1/2, 1/5, 1/4,
      3, 1/3, 1/2,
```

```

    1/4, 1/3,
    2),
numalternativas = 5,
v.nombres.alternativas =
c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_rent = multicriterio.crea.matrizvaloraciones_mej(
    c(1/3, 1/2, 1/3, 1/4,
      3, 2, 1/2,
      1/2, 1/3,
      1/2),
numalternativas = 5,
v.nombres.alternativas =
c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_purch = multicriterio.crea.matrizvaloraciones_mej(
    c(1/2, 2, 5, 6,
      3, 6, 7,
      3, 5,
      2),
numalternativas = 5,
v.nombres.alternativas =
c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_safety = multicriterio.crea.matrizvaloraciones_mej(
    c(4, 5, 2, 5,
      2, 1/3, 1,
      1/3, 1/2,
      3),
numalternativas = 5,
v.nombres.alternativas =
c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_health = multicriterio.crea.matrizvaloraciones_mej(
    c(3, 1, 1/2, 4,
      1/3, 1/4, 2,
      1/2, 4,
      5),
numalternativas = 5,

```

```

v.nombres.alternativas =
c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

tb_pollution = multicriterio.crea.matrizvaloraciones_mej(
  c(3,7,3,5,
    5,1,3,
    1/5,1/3,
    2),
  numalternativas = 5,
  v.nombres.alternativas =
  c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

# pesos locales
p_criterios = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
  tb_criterios)
p_cost = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_cost)
p_rent = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_rent)
p_purch = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_purch)
p_safety = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_safety)
p_health = multicriterio.metodoAHP.variante1.autovectormayorautovalor(tb_health)
p_pollution = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
  tb_pollution)

```

A continuación, se incorporan los datos necesarios para el **caso multinivel** del método AHP.

Tablas de los **subcriterios**:

```

tb_subcriterios = multicriterio.crea.matrizvaloraciones_mej(
  c(2),
  numalternativas = 2,
  v.nombres.alternativas = c('Groceries Index',
  'Restaurants Index')
  );

tb_grocery = multicriterio.crea.matrizvaloraciones_mej(
  c(1/3, 1, 1/5, 1/3,
    3, 1/3, 1,
    1/5, 1/3,
    3),
  numalternativas = 5,
  v.nombres.alternativas =
  c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

```

```

tb_restaurant = multicriterio.crea.matrizvaloraciones_mej(
    c(1/5, 1/4, 1/7, 1/6,
      2, 1/3, 1/2,
      1/4, 1/3,
      2),
    numalternativas = 5,
    v.nombres.alternativas =
    c("Copenhagen", "Berlin", "Paris", "Madrid", "Rome"));

```

Table 15: Matriz de subcriterios

	Groceries Index	Restaurants Index
Groceries Index	1.0	2
Restaurants Index	0.5	1

Table 16: Matriz Groceries Index

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	1	0.3333333	1	0.2000000	0.3333333
Berlin	3	1.0000000	3	0.3333333	1.0000000
Paris	1	0.3333333	1	0.2000000	0.3333333
Madrid	5	3.0000000	5	1.0000000	3.0000000
Rome	3	1.0000000	3	0.3333333	1.0000000

Table 17: Matriz Restaurant Index

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	1	0.2	0.25	0.1428571	0.1666667
Berlin	5	1.0	2.00	0.3333333	0.5000000
Paris	4	0.5	1.00	0.2500000	0.3333333
Madrid	7	3.0	4.00	1.0000000	2.0000000
Rome	6	2.0	3.00	0.5000000	1.0000000

### Estudio de la inconsistencia:

```

Inc_subcriterios = multicriterio.metodoAHP.coef.inconsistencia(tb_subcriterios)

Inc_grocery = multicriterio.metodoAHP.coef.inconsistencia(tb_grocery)
c(Inc_grocery$mensaje, round(Inc_grocery$RI.coef.inconsistencia, 4))

```

```
[1] "Consistencia aceptable" "0.0124"
```

```
Inc_restaurant = multicriterio.metodoAHP.coef.inconsistencia(tb_restaurant)
c(Inc_restaurant$mensaje, round(Inc_restaurant$RI.coef.inconsistencia,4))
```

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

Calculo de los **pesos locales**:

```
p_subcriterios = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_subcriterios)
p_subcriterios$valoraciones.ahp
```

```
Groceries Index Restaurants Index
0.6666667      0.3333333
```

```
p_grocery = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_grocery)
p_grocery$valoraciones.ahp
```

```
Copenhagen      Berlin      Paris      Madrid      Rome
0.07284055 0.19391621 0.07284055 0.46648647 0.19391621
```

```
p_restaurant = multicriterio.metodoAHP.variante1.autovectormayorautovalor(
    tb_restaurant)
p_restaurant$valoraciones.ahp
```

```
Copenhagen      Berlin      Paris      Madrid      Rome
0.03987621 0.16621481 0.10597795 0.42103402 0.26689701
```

Calculo de los **pesos globales**:

```
pc <- p_criterios$valoraciones.ahp # pesos criterios
ps <- p_subcriterios$valoraciones.ahp # pesos subcriterios

p_hojas <- c(
  "Groceries Index" = pc["Cost of Living Index"] * ps["Groceries Index"],
  "Restaurants Index" = pc["Cost of Living Index"] * ps["Restaurants Index"],
```

```

"Rent Index" = pc["Rent Index"],
"Purchasing Power Index" = pc["Purchasing Power Index"],
"Safety Index" = pc["Safety Index"],
"Health care Index" = pc["Health care Index"],
"Pollution Index" = pc["Pollution Index"]
)

M_alt_hojas <- rbind(
  p_grocery$valoraciones.ahp,
  p_restaurant$valoraciones.ahp,
  p_rent$valoraciones.ahp,
  p_purch$valoraciones.ahp,
  p_safety$valoraciones.ahp,
  p_health$valoraciones.ahp,
  p_pollution$valoraciones.ahp
)
rownames(M_alt_hojas) <- names(p_hojas)

#pond.globales = p_hojas %*% M_alt_hojas
#pond.globales

p_global <- multicriterio.metodoAHP.pesosglobales_entabla(p_hojas, M_alt_hojas)
p_global

```

	Groceries Index.Cost of Living Index	Restaurants Index.Cost of Living Index	Rent Index.Rent Index
Copenhagen	0.07284055	0.03987621	0.07223072
Berlin	0.19391621	0.16621481	0.26652950
Paris	0.07284055	0.10597795	0.10904732
Madrid	0.46648647	0.42103402	0.18321436
Rome	0.19391621	0.26689701	0.36897810
Ponder.Criterios	0.06267901	0.03133950	0.04888298
	Purchasing Power Index.Purchasing Power Index		
Copenhagen	0.28228349		
Berlin	0.43589670		
Paris	0.17182955		

Madrid		0.06671487
Rome		0.04327540
Ponder.Criterios		0.13674234
	Safety Index.	Safety Index
Copenhagen	Health care	Index
Berlin	Index	.Health care
Paris	0.45134332	Index
Madrid	0.10933371	0.23028345
Rome	0.07001099	0.09095138
Ponder.Criterios	0.26442217	0.23028345
	0.10488981	0.38962764
	0.20253783	0.05885407
		0.22648965
	Pollution Index.	Pollution Index
Copenhagen	Ponderadores Globales	
Berlin	0.47263799	0.3292102
Paris	0.20447773	0.1923118
Madrid	0.04305389	0.1155933
Rome	0.18613571	0.2565415
Ponder.Criterios	0.09369468	0.1063432
	0.29132868	NA

Los resultados del AHP multinivel muestran que, al incorporar los subcriterios dentro de “Cost of Living”, las prioridades globales se mantienen estables y **Copenhagen** sigue siendo la alternativa con el mayor peso total.

### 3.2 Resolución con paquete AHP

```
# Cargar el modelo:
library(ahp)
ahp_ej_m = ahp::Load("ej02_ahp.ahp")
ahp_ej_m
```

```
levelName
1 Elegir la ciudad europea más sostenible para vivir
2   |--Cost of Living Index
3     |--Groceries Index
4       |--Copenhagen
5       |--Berlin
6       |--Paris
7       |--Madrid
8       |--Rome
9     |--Restaurants Index
10    |--Copenhagen
```

```

11 |      |--Berlin
12 |      |--Paris
13 |      |--Madrid
14 |      |--Rome
15 |--Rent Index
16 |      |--Copenhagen
17 |      |--Berlin
18 |      |--Paris
19 |      |--Madrid
20 |      |--Rome
21 |--Purchasing Power Index
22 |      |--Copenhagen
23 |      |--Berlin
24 |      |--Paris
25 |      |--Madrid
26 |      |--Rome
27 |--Safety Index
28 |      |--Copenhagen
29 |      |--Berlin
30 |      |--Paris
31 |      |--Madrid
32 |      |--Rome
33 |--Health care Index
34 |      |--Copenhagen
35 |      |--Berlin
36 |      |--Paris
37 |      |--Madrid
38 |      |--Rome
39 |--Pollution Index
40 |      |--Copenhagen
41 |      |--Berlin
42 |      |--Paris
43 |      |--Madrid
44 |      |--Rome

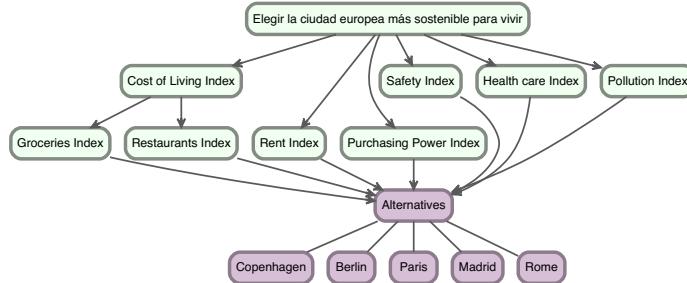
# Calcular las prioridades:
Calculate(ahp_ej_m)
print(ahp_ej_m, priority = function(x) x$parent$priority["Total", x$name])

```

	levelName	priority
1 Elegir la ciudad europea más sostenible para vivir		NA
2  --Cost of Living Index		0.09401851

3	--Groceries Index	0.66666667
4	--Copenhagen	0.07284055
5	--Berlin	0.19391621
6	--Paris	0.07284055
7	--Madrid	0.46648647
8	°--Rome	0.19391621
9	°--Restaurants Index	0.33333333
10	--Copenhagen	0.03987621
11	--Berlin	0.16621481
12	--Paris	0.10597795
13	--Madrid	0.42103402
14	°--Rome	0.26689701
15	--Rent Index	0.04888298
16	--Copenhagen	0.07223072
17	--Berlin	0.26652950
18	--Paris	0.10904732
19	--Madrid	0.18321436
20	°--Rome	0.36897810
21	--Purchasing Power Index	0.13674234
22	--Copenhagen	0.28228349
23	--Berlin	0.43589670
24	--Paris	0.17182955
25	--Madrid	0.06671487
26	°--Rome	0.04327540
27	--Safety Index	0.20253783
28	--Copenhagen	0.45134332
29	--Berlin	0.10933371
30	--Paris	0.07001099
31	--Madrid	0.26442217
32	°--Rome	0.10488981
33	--Health care Index	0.22648965
34	--Copenhagen	0.23028345
35	--Berlin	0.09095138
36	--Paris	0.23028345
37	--Madrid	0.38962764
38	°--Rome	0.05885407
39	°--Pollution Index	0.29132868
40	--Copenhagen	0.47263799
41	--Berlin	0.20447773
42	--Paris	0.04305389
43	--Madrid	0.18613571
44	°--Rome	0.09369468

```
# Visualizar la jerarquía
Visualize(ahp_ej_m)
```



```
# Analizar
Analyze(ahp_ej_m,variable = "priority" ,sort="orig")
```

		Priority	Copenhagen	Berlin
1	Elegir la ciudad europea más sostenible para vivir	100.0%	NA	NA
2	--Cost of Living Index	9.4%	NA	NA
3	--Groceries Index	66.7%	7.3%	19.4%
4	`--Restaurants Index	33.3%	4.0%	16.6%
5	--Rent Index	4.9%	7.2%	26.7%
6	--Purchasing Power Index	13.7%	28.2%	43.6%
7	--Safety Index	20.3%	45.1%	10.9%
8	--Health care Index	22.6%	23.0%	9.1%
9	`--Pollution Index	29.1%	47.3%	20.4%
	Paris   Madrid   Rome   Inconsistency			
1	NA        NA        NA        1.3%			
2	NA        NA        NA        0.0%			

3	7.3%	46.6%	19.4%	1.3%
4	10.6%	42.1%	26.7%	2.8%
5	10.9%	18.3%	36.9%	2.4%
6	17.2%	6.7%	4.3%	2.1%
7	7.0%	26.4%	10.5%	1.8%
8	23.0%	39.0%	5.9%	1.2%
9	4.3%	18.6%	9.4%	2.4%

```
Analyze(ahp_ej_m)
```

			Weight	Copenhagen	Madrid
1	Elegir la ciudad europea más sostenible para vivir		100.0%	32.9%	25.7%
2	--Pollution Index		29.1%	13.8%	5.4%
3	--Health care Index		22.6%	5.2%	8.8%
4	--Safety Index		20.3%	9.1%	5.4%
5	--Purchasing Power Index		13.7%	3.9%	0.9%
6	--Cost of Living Index		9.4%	0.6%	4.2%
7	--Groceries Index		6.3%	0.5%	2.9%
8	`--Restaurants Index		3.1%	0.1%	1.3%
9	--Rent Index		4.9%	0.4%	0.9%
	Berlin	Paris	Rome	Inconsistency	
1	19.2%	11.6%	10.6%	1.3%	
2	6.0%	1.3%	2.7%	2.4%	
3	2.1%	5.2%	1.3%	1.2%	
4	2.2%	1.4%	2.1%	1.8%	
5	6.0%	2.3%	0.6%	2.1%	
6	1.7%	0.8%	2.1%	0.0%	
7	1.2%	0.5%	1.2%	1.3%	
8	0.5%	0.3%	0.8%	2.8%	
9	1.3%	0.5%	1.8%	2.4%	

```
# Analizar con tabla mejorada
aa = AnalyzeTable(ahp_ej_m)
formattable::as.htmlwidget(aa)
```

	Weight	Copenhagen	Madrid	Berlin	Paris	Rome	Inconsistency
Elegir la ciudad europea más sostenible para vivir	100.0%	32.9%	25.7%	19.2%	11.6%	10.6%	1.3%
Pollution Index	29.1%	13.8%	5.4%	6.0%	1.3%	2.7%	2.4%
Health care Index	22.6%	5.2%	8.8%	2.1%	5.2%	1.3%	1.2%
Safety Index	20.3%	9.1%	5.4%	2.2%	1.4%	2.1%	1.8%
Purchasing Power Index	13.7%	3.9%	0.9%	6.0%	2.3%	0.6%	2.1%
Cost of Living Index	9.4%	0.6%	4.2%	1.7%	0.8%	2.1%	0.0%
Groceries Index	6.3%	0.5%	2.9%	1.2%	0.5%	1.2%	1.3%
Restaurants Index	3.1%	0.1%	1.3%	0.5%	0.3%	0.8%	2.8%
Rent Index	4.9%	0.4%	0.9%	1.3%	0.5%	1.8%	2.4%

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

	Priority	Copenhagen	Berlin	Paris	Madrid	Rome	Inconsistency
Elegir la ciudad europea más sostenible para vivir	100.0%						1.3%
Cost of Living Index	9.4%						0.0%
Groceries Index	66.7%	7.3%	19.4%	7.3%	46.6%	19.4%	1.3%
Restaurants Index	33.3%	4.0%	16.6%	10.6%	42.1%	26.7%	2.8%
Rent Index	4.9%	7.2%	26.7%	10.9%	18.3%	36.9%	2.4%
Purchasing Power Index	13.7%	28.2%	43.6%	17.2%	6.7%	4.3%	2.1%
Safety Index	20.3%	45.1%	10.9%	7.0%	26.4%	10.5%	1.8%
Health care Index	22.6%	23.0%	9.1%	23.0%	39.0%	5.9%	1.2%
Pollution Index	29.1%	47.3%	20.4%	4.3%	18.6%	9.4%	2.4%

**Copenhagen** mantiene la primera posición (32.9%), seguida de Madrid (25.7%) y Berlin (19.2%).

Este resultado confirma la robustez del modelo: al introducir subcriterios, el ranking global apenas varía respecto al AHP básico.

Las inconsistencias son bajas (<0.1), lo que garantiza la coherencia y fiabilidad de los juicios emitidos.

Ahora, para validar la solución obtenida, se resuelve el mismo problema utilizando los métodos de superación.

## 4 Método ELECTRE

El método ELECTRE se basa en dos conceptos fundamentales:

- **Concordancia:** muestra cuánto una opción es al menos tan buena como la otra con la que se compara, considerando los pesos de los criterios.
- **Discordancia:** evita que una alternativa sea preferida a otra cuando en algún criterio es demasiado desfavorable.

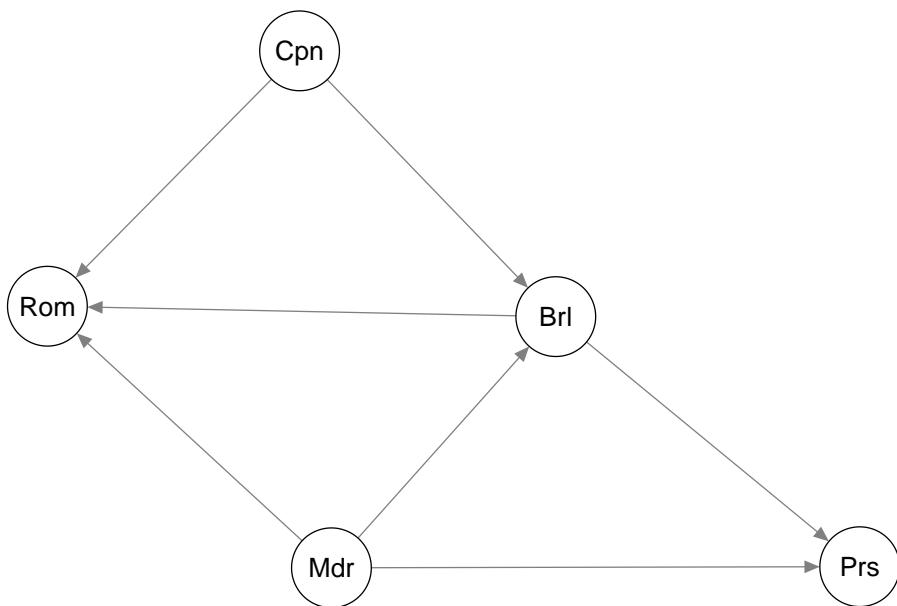
En este trabajo, se ha:

- fijado un nivel de concordancia mínimo alpha = 0.7
- elegido pesos para los criterios de forma que se enfatice la sostenibilidad, pero sin ignorar la economía.
- establecido vetos en los criterios “Safety” y “Pollution”, considerados claves para la sostenibilidad.

#### 4.1 Iteración 1

```
sol.Electre = multicriterio.metodoELECTRE_I(
  nueva.X,
  pesos.criterios = c(0.05, 0.05, 0.10,
                      0.25, 0.25, 0.30),
  nivel.concordancia.minimo.alpha = 0.7,
  no.se.compensan = c(Inf, Inf, Inf, 0.40, Inf, 0.40),
  que.alternativas = TRUE)

qgraph::qgraph(sol.Electre$relacion.dominante)
```



```
sol.Electre$nucleo_aprox
```

Copenhagen	Madrid
1	4

Despues la primera iteracion, se obtiene un nucleo con dos alternativas (Copenhagen y Madrid).

Para intentar quedarse con una única alternativa óptima se puede:

- reducir el grafo a las alterantivas en el nucleo y/o
- reducir el valor del alpha [0.5,1)

## 4.2 Iteración 2: reducir alternativas

```
sol.Electre = multicriterio.metodoELECTRE_I(
  nueva.X,
  pesos.criterios = c(0.05, 0.05, 0.10,
                      0.25, 0.25, 0.30),
  nivel.concordancia.minimo.alpha = 0.7,
  no.se.compensan = c(Inf, Inf, Inf, 0.40, Inf, 0.40),
  que.alternativas = c(1,4))

qgraph::qgraph(sol.Electre$relacion.dominante)
```



```
sol.Electre$nucleo_aprox
```

Copenhagen	Madrid
1	2

### 4.3 Iteración 3: reducir alpha

```
sol.Electre = multicriterio.metodoELECTRE_I(  
    nueva.X,  
    pesos.criterios = c(0.05, 0.05, 0.10,  
                        0.25, 0.25, 0.30),  
    nivel.concordancia.minimo.alpha = 0.6,  
    no.se.compensan = c(Inf, Inf, Inf, 0.40, Inf, 0.40),  
    que.alternativas = c(1,4))  
  
qgraph::qgraph(sol.Electre$relacion.dominante)
```



```
sol.Electre$nucleo_aprox
```

Copenhagen

1

Al reducir el grafo a las alternativas del núcleo no se obtiene una única alternativa, por lo que se reduce el valor de alpha (en este caso hasta 0.6) hasta lograr una sola alternativa óptima, que resulta ser Copenhagen.

El método ELECTRE identifica a **Copenhagen** como la ciudad más sostenible.

## 5 Método PROMETHEE

El metodo PROMETHEE se basa en la comparación por pares de alternativas. A diferencia de ELECTRE, no utiliza vetos, porque evalua la preferencia de una alternativa sobre otra mediante funciones específicas definidas para cada criterio. Ademas, permite obtener un ordenamiento parcial (PROMETHEE I) y un ordenamiento total (PROMETHEE II) de las alternativas.

Elección de las funciones de preferencia para los distintos criterios:

Table 18: Criterios e funciones asociadas

Criterio	Función	Interpretación
<b>Cost, Rent</b>	Lineal con indiferencia (5)	Las pequeñas diferencias no tienen peso (q), luego el crecimiento es lineal hasta alcanzar una referencia completa (p).
<b>Purchasing Power, Pollution</b>	Lineal (3)	Cada diferencia cuenta de forma proporcional hasta llegar a p.
<b>Safety, Health Care</b>	Por niveles (4)	Por debajo de q hay indiferencia, entre q y p una preferencia débil o constante, y por encima de p una preferencia completa.

### 5.1 Promethee I:

Primero se aplica el metodo PROMETHEE I, que construye un ordenamiento parcial entre las alternativas. El flujo positivo ( $\phi_+$ ) representa el grado en que una ciudad domina a las demás, el flujo negativo ( $\phi_-$ ) indica cuánto una ciudad es dominada por las otras. Por tanto, una alternativa con mayor  $\phi_+$  y menor  $\phi_-$  se considera más preferible. El grafo dirigido permite visualizar estas relaciones de dominancia parcial entre las alternativas.

```

pesos.criterios = pesos.criterios = c(0.05, 0.05, 0.10,
                                         0.25, 0.25, 0.30);
tab.fpref = matrix( c(5, 0.05, 0.25, 0,
                      5, 0.05, 0.25, 0,
                      3, 0, 0.20, 0,
                      4, 0.03, 0.15, 0,
                      4, 0.03, 0.10, 0,
                      3, 0, 0.15, 0), ncol=4, byrow=T )
tab.fpref

```

```

[,1] [,2] [,3] [,4]
[1,] 5 0.05 0.25 0
[2,] 5 0.05 0.25 0
[3,] 3 0.00 0.20 0
[4,] 4 0.03 0.15 0
[5,] 4 0.03 0.10 0
[6,] 3 0.00 0.15 0

```

```

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

```

```

$tabla.indices
      Copenhagen    Berlin    Paris    Madrid    Rome
Copenhagen 0.00000000 0.80000000 0.6134573 0.52500000 0.9000000
Berlin     0.17002188 0.00000000 0.7500000 0.11527778 0.7750000
Paris      0.08710317 0.25000000 0.0000000 0.10000000 0.3500000
Madrid     0.35000000 0.58381643 0.9000000 0.00000000 0.9208333
Rome       0.10000000 0.07123016 0.6500000 0.04900794 0.0000000

$vflujos.ent
Copenhagen    Berlin    Paris    Madrid    Rome
2.8384573 1.8102997 0.7871032 2.7546498 0.8702381

$vflujos.sal
Copenhagen    Berlin    Paris    Madrid    Rome
0.7071251 1.7050466 2.9134573 0.7892857 2.9458333

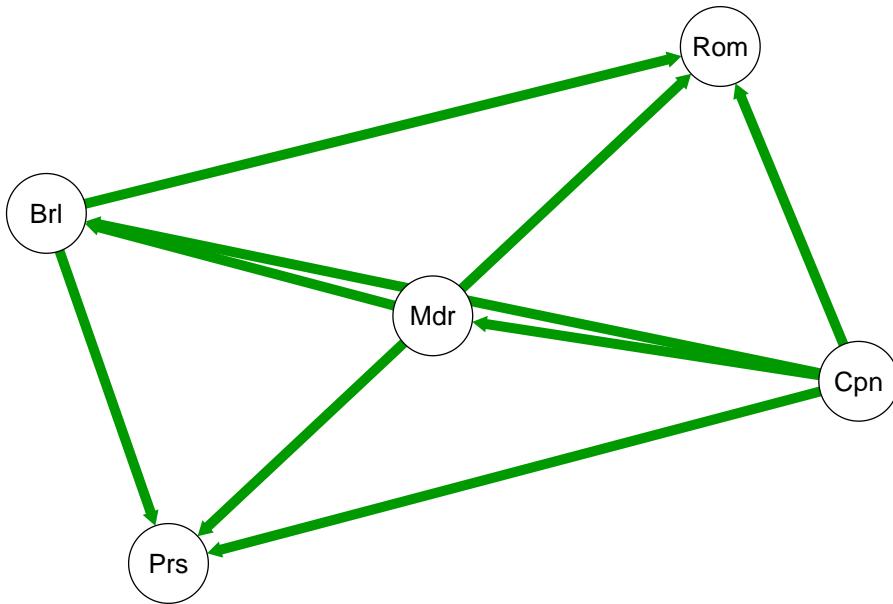
$tablarelacionsuper
      Copenhagen    Berlin    Paris    Madrid    Rome
Copenhagen      0.5    1.0    1.0    1.0    1.0
Berlin          0.0    0.5    1.0    0.0    1.0
Paris           0.0    0.0    0.5    0.0    0.0
Madrid          0.0    1.0    1.0    0.5    1.0
Rome            0.0    0.0    0.0    0.0    0.5

require("qgraph")

```

Loading required package: qgraph

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



En este caso, se observe que **Copenhagen** domina la otras alternativas, mostrando el maximo valor de phi+ y el minimo de phi- .

## 5.2 Promethee II:

Despues se aplica el metodo PROMETHEE II, que transforma el ordenamiento parcial en un ordenamiento total mediante el flujo neto. El flujo neto se define como la diferencia entre el flujo positivo y el negativo. Este flujo define la posicion global de cada alternativa: cuanto mayor sea el flujo, mejor será la alternativa.

```

tab.Pthee.ii = multicriterio.metodo.promethee_ii(nueva.X, pesos.criterios,
                                                 tab.fpref)
tab.Pthee.ii

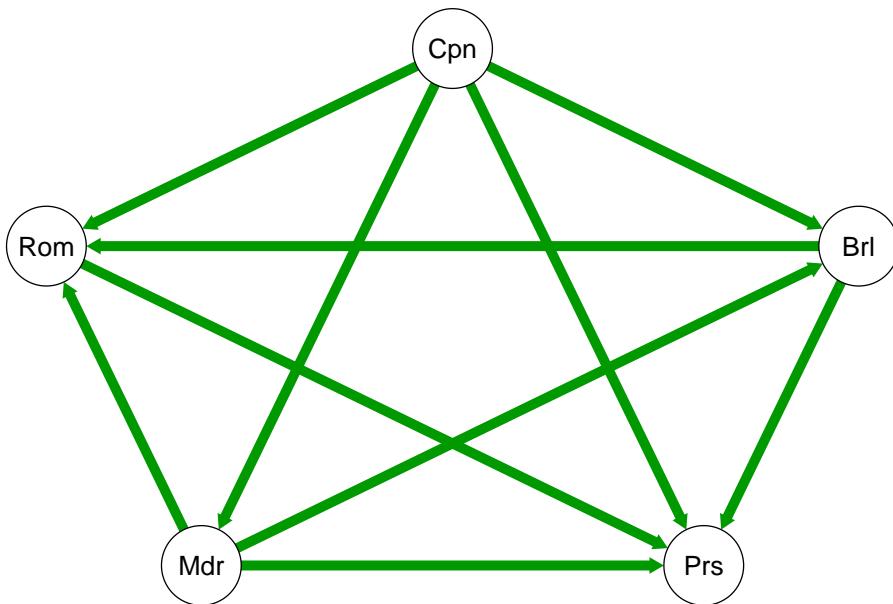
```

	Copenhagen	Berlin	Paris	Madrid	Rome
Copenhagen	0.0000000	0.8000000	0.6134573	0.5250000	0.9000000
Berlin	0.17002188	0.0000000	0.7500000	0.11527778	0.7750000
Paris	0.08710317	0.2500000	0.0000000	0.1000000	0.3500000
Madrid	0.3500000	0.58381643	0.9000000	0.0000000	0.9208333
Rome	0.1000000	0.07123016	0.6500000	0.04900794	0.0000000

```
$vflujos.netos
Copenhagen Berlin Paris Madrid Rome
2.1313323 0.1052531 -2.1263542 1.9653640 -2.0755952
```

```
$tablarelacionsuper
Copenhagen Berlin Paris Madrid Rome
Copenhagen 0.5 1.0 1.0 1.0 1.0
Berlin 0.0 0.5 1.0 0.0 1.0
Paris 0.0 0.0 0.5 0.0 0.0
Madrid 0.0 1.0 1.0 0.5 1.0
Rome 0.0 0.0 1.0 0.0 0.5
```

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



```
# ordenacion final alternativas del Metodo Promethee II
order(tab.Pthee.ii$vflujos.netos, decreasing=T)
```

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

Los flujos netos resultan en el siguiente ranking: **Copenhagen**>Madrid>Berlin>Roma>Paris.

### 5.3 Promethee I (medias):

En el caso anterior, los flujos se calculaban como sumas, mientras ahora se calculan como el promedio de las preferencias, es decir, se dividen por n-1. Si todo va bien, el orden no cambiará, ya que dividir por n-1 solo normaliza los valores entre 0 y 1.

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

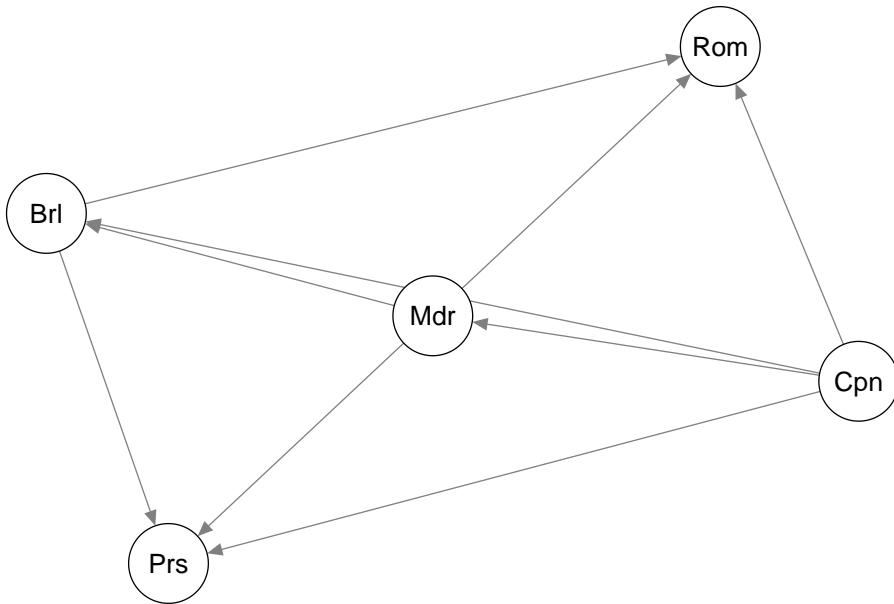
$tabla.indices
      Copenhagen      Berlin      Paris      Madrid      Rome
Copenhagen 0.00000000 0.80000000 0.6134573 0.52500000 0.9000000
Berlin     0.17002188 0.00000000 0.7500000 0.11527778 0.7750000
Paris      0.08710317 0.25000000 0.0000000 0.10000000 0.3500000
Madrid     0.35000000 0.58381643 0.9000000 0.00000000 0.9208333
Rome       0.10000000 0.07123016 0.6500000 0.04900794 0.0000000

$vflujos.ent
Copenhagen      Berlin      Paris      Madrid      Rome
0.7096143 0.4525749 0.1967758 0.6886624 0.2175595

$vflujos.sal
Copenhagen      Berlin      Paris      Madrid      Rome
0.1767813 0.4262616 0.7283643 0.1973214 0.7364583

$tablarelacionsuper
      Copenhagen      Berlin      Paris      Madrid      Rome
Copenhagen          0          1          1          1          1
Berlin              0          0          1          0          1
Paris               0          0          0          0          0
Madrid              0          1          1          0          1
Rome                0          0          0          0          0

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



#### 5.4 Promethee II (medias):

```

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

```

```

$tabla.indices
      Copenhagen      Berlin      Paris      Madrid      Rome
Copenhagen 0.00000000 0.80000000 0.6134573 0.52500000 0.9000000
Berlin     0.17002188 0.00000000 0.7500000 0.11527778 0.7750000
Paris      0.08710317 0.25000000 0.0000000 0.10000000 0.3500000
Madrid     0.35000000 0.58381643 0.9000000 0.00000000 0.9208333
Rome       0.10000000 0.07123016 0.6500000 0.04900794 0.0000000

```

```

$vflujos.netos
      Copenhagen      Berlin      Paris      Madrid      Rome
Copenhagen 0.53283307 0.02631327 -0.53158854 0.49134101 -0.51889881

```

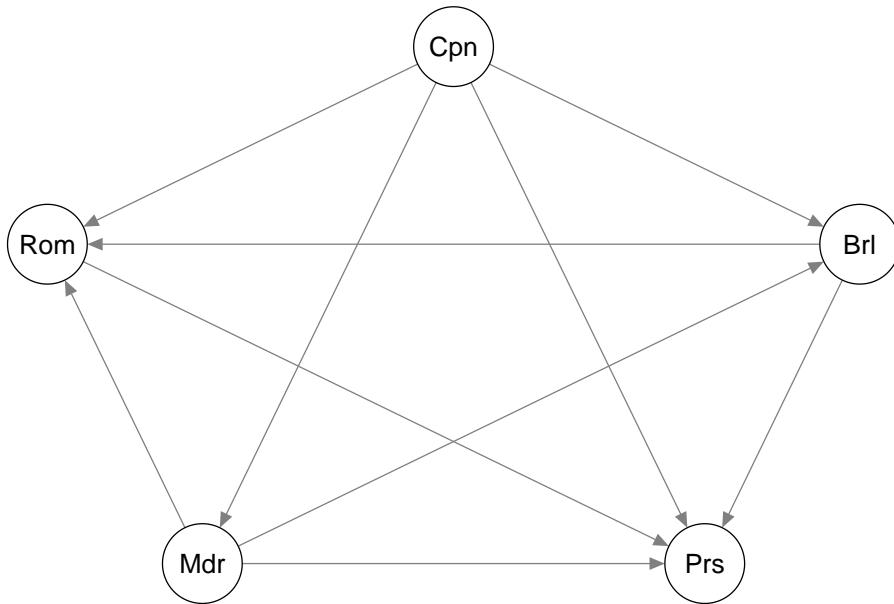
```

$tablarelacionsuper
      Copenhagen Berlin Paris Madrid Rome
Copenhagen      0     1     1     1     1

```

Berlin	0	0	1	0	1
Paris	0	0	0	0	0
Madrid	0	1	1	0	1
Rome	0	0	1	0	0

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



```
# ordenacion final alternativas del Metodo Promethee II
order(tab.Pthee.ii_med$vflujos.netos, decreasing=T)
```

[1] 1 4 2 5 3

### Comparativa Promethee II: sin medias y con medias

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

[1] 1 4 2 5 3

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

[1] 1 4 2 5 3

Se nota que el orden no ha cambiado.

Se concluye que segundo el metodo PROMETHEE, **Copenhagen** es la mejor opcion.

## 6 Conclusión

Los tres métodos aplicados (AHP, ELECTRE, PROMETHEE) coincidieron en identificar **Copenhagen** como la **ciudad más sostenible para vivir**, reforzando así la robustez y consistencia de la decision multicriterio.

En PROMETHEE y ELECTRE se han utilizado pesos diferentes a los de AHP para evaluar la robustez de la decisión y priorizar más los criterios de sostenibilidad y bienestar (Safety, Health Care, Pollution), reduciendo el peso de los aspectos económicos (Cost of Living, Rent, Purchasing Power).