

ELECCIÓN DEL MEJOR TIPO DE EMPLEO PARA COMPAGINAR CON LOS ESTUDIOS UNIVERSITARIOS

Marta Martín Fernández

2025-10-28

Contents

| | | |
|----------|--|-----------|
| 1 | INTRODUCCIÓN | 3 |
| 2 | DEFINICIÓN DEL PROBLEMA | 3 |
| 3 | ALTERNATIVAS (OPCIONES DE EMPLEO) | 3 |
| 4 | CRITERIOS Y SUBCRITERIOS | 4 |
| 5 | MÉTODO ELECTRE | 5 |
| 6 | CÁLCULOS EN EL MÉTODO ELECTRE I | 14 |
| 7 | MÉTODO PROMETHEE | 15 |
| 7.1 | REPRESENTACIÓN GRÁFICA DE LAS FUNCIONES DE PREFERENCIA | 16 |
| 7.2 | APLICACIÓN DEL MÉTODO PROMETHEE I | 22 |
| 7.3 | APLICACIÓN DEL MÉTODO PROMETHEE II | 24 |
| 7.3.1 | ORDENACIÓN FINAL DE LAS ALTERNATIVAS POR EL MÉTODO DE PROMETHEE II | 25 |
| 7.4 | MÉTODO PROMETHEE I MEDIAS | 25 |
| 7.5 | MÉTODO PROMETHEE II MEDIAS | 27 |
| 7.5.1 | ORDENACIÓN FINAL DE LAS ALTERNATIVAS POR EL MÉTODO DE PROMETHEE II MEDIAS | 28 |
| 7.5.2 | COMPARACIÓN MÉTODO PROMETHEE II SIN MEDIAS Y CON MEDIAS . . . | 28 |
| 8 | MÉTODO PROMETHEE WINDOWS | 29 |
| 9 | AHP CON R | 31 |
| 9.1 | Matriz de subcriterios. | 31 |
| 9.2 | Matrices de Alternativas/Criterios. | 31 |
| 9.2.1 | Criterio SALARIO | 31 |
| 9.2.2 | Criterio BENEFICIOS | 32 |
| 9.2.3 | Criterio CERCANÍA AL HOGAR | 32 |
| 9.2.4 | Criterio CARGA MENTAL/ESTRÉS | 32 |
| 9.2.5 | Criterio OPORTUNIDADES DESARROLLO PERSONAL | 32 |
| 9.2.6 | Criterio RELACIÓN CON LOS ESTUDIOS | 32 |
| 9.2.7 | Criterio SATISFACCIÓN PERSONAL | 33 |
| 9.2.8 | Criterio RELACIÓN CON COMPAÑEROS Y JEFES | 33 |
| 9.3 | EMPLEANDO LAS FUNCIONES VISTAS EN CLASE CON R | 33 |
| 9.3.1 | MÉTODO 1: | 33 |
| 9.3.2 | MÉTODO 2: | 39 |

| | | |
|-------|--------------------------------------|-----------|
| 9.4 | EMPLEANDO EL PAQUETE AHP | 40 |
| 9.4.1 | MÉTODO 1: | 40 |
| 10 | MÉTODO 2: USO DEL PAQUETE AHP | 43 |
| 11 | CONCLUSIONES GENERALES | 46 |

1 INTRODUCCIÓN

He elegido este tema porque es un problema que afecta a muchos estudiantes universitarios que buscan equilibrar sus estudios con un empleo que les permita cubrir sus gastos y adquirir experiencia laboral relevante. La elección del tipo de empleo adecuado puede influir significativamente en el rendimiento académico y en la calidad de vida del estudiante. Además, este tema es relevante en el contexto actual, donde muchos estudiantes enfrentan desafíos financieros y buscan oportunidades laborales que se adapten a sus horarios y necesidades. Al analizar las diferentes opciones de empleo, se pueden considerar factores como la flexibilidad horaria, la remuneración, la relevancia para la carrera académica y el ambiente laboral. Este análisis puede ayudar a los estudiantes a tomar decisiones informadas sobre qué tipo de empleo es más adecuado para ellos, permitiéndoles maximizar tanto su desarrollo académico como profesional.

Es un tema que me motiva debido a que ha sido una decisión que he tenido que tomar en el pasado, ya que necesitaba un empleo que pudiera compaginar con mi estudio en la universidad, y es una decisión que tienen que tomar mucho de los estudiantes universitarios año tras año.

2 DEFINICIÓN DEL PROBLEMA

El problema consiste en seleccionar el mejor tipo de empleo para un estudiante universitario que busca equilibrar sus estudios con un trabajo que le permita cubrir sus gastos y adquirir experiencia laboral relevante. Se deben considerar varios criterios y alternativas para tomar una decisión informada mediante el uso de las técnicas de toma de decisiones multicriterio (AHP, ELECTRE, PROMETHEE).

3 ALTERNATIVAS (OPCIONES DE EMPLEO)

Vamos a considerar 4 tipos de empleo comunes entre los estudiantes universitarios:

A1: EMPLEO EN HOSTELERÍA/SERVICIOS.

DESCRIPCIÓN: Camarero, repartidor o trabajador en un bar. Horarios fijos, a menudo suelen ser los fines de semana.

A2: PROFESOR PARTICULAR/TUTOR.

DESCRIPCIÓN: Dar clases particulares a otros estudiantes. Horarios flexibles, generalmente por las tardes o fines de semana cuando al estudiante le venga mejor por su organización.

A3: RRPP/PROMOTOR DE EVENTOS.

DESCRIPCIÓN: Trabajar en la promoción de eventos, productos o servicios, además de captar gente para el local en el que trabaje. Suelen ser trabajos con horarios nocturnos y fines de semana.

A4: DEPENDIENTE/A.

DESCRIPCIÓN: Es uno de los empleos más solicitados entre los estudiantes. Muchas tiendas de ropa, calzado y complementos ofrecen puestos de trabajo a media jornada, ideal para jóvenes estudiantes apasionados por la moda.

A5: CUIDADOR DE NIÑOS.

DESCRIPCIÓN: Fácil empleo para los estudiantes con horario flexible, que puede adaptarse a la organización del estudiante.

4 CRITERIOS Y SUBCRITERIOS

Hemos definido cuatro criterios principales y un total de 8 subcriterios para evaluar las alternativas de empleo:

C1. Remuneración e impacto financiero

- Subcriterios:

C1.1. Salario por hora. Tenemos que maximizar, queremos ganar el mayor dinero posible. ESCALA: Valor en euros por hora.

C1.2. Beneficios adicionales. Tenemos que maximizar, ya que si ganamos algo más aparte del salario normal es beneficioso para nosotros. ESCALA: Escala del 1 (bajo) al 5 (alto) para reflejar descuentos, propinas. . .

OBJETIVO: Maximizar la remuneración total y los beneficios financieros asociados al empleo, ya que vamos a tener más dinero para ahorrar y cubrir nuestras necesidades.

C2. Impacto académico

- Subcriterios:

C2.1. Cercanía al hogar. Tenemos que maximizar, para que este trabajo no suponga desplazarnos y perder más tiempo, ya que siendo estudiantes no es algo que nos sobre. ESCALA: Escala del 1 (lejos) al 10 (muy cerca/teletrabajo).

C2.2. Carga mental/estrés. Tenemos que minimizar, para que el trabajo no sea una carga en nuestra vida. ESCALA: Escala del 1 (muy bajo) al 10 (muy alto).

OBJETIVO: Minimizar, ya que buscamos que el impacto negativo del empleo en el rendimiento académico sea mínimo, y que el trabajo no suponga un estrés adicional en nuestra vida que pueda llegar a interferir con los estudios. Queremos sobre todo el bienestar del estudiante.

C3. Relevancia para la Carrera Académica

- Subcriterios:

C3.1. Oportunidades de desarrollo profesional. Tenemos que maximizar, para que nos beneficie todo lo posible. ESCALA: Escala del 1 (bajo) al 10 (alto).

C3.2. Relación con los estudios. Tenemos que maximizar, para que también nos ayude en nuestra formación. ESCALA: Escala del 1 (bajo) al 10 (alto).

OBJETIVO: Maximizar, ya que buscamos que el empleo tenga una relación directa con la carrera académica del estudiante, proporcionando experiencia relevante y oportunidades de desarrollo profesional que puedan beneficiar su futuro laboral.

C4. Desarrollo personal

- Subcriterios:

C4.1. Satisfacción personal. Tenemos que maximizar, para que el trabajo nos haga sentir bien en nuestro día a día y cuando vayamos. ESCALA: Escala del 1 (bajo) al 10 (alto).

C4.2. Relaciones con compañeros y jefes. Tenemos que maximizar, ya que es un aspecto importante siempre en un trabajo. Se trabaja mejor con personas con las que te lleves bien y tengas buena relación. ESCALA: Escala del 1 (bajo) al 10 (alto).

OBJETIVO: Maximizar, ya que buscamos que el empleo contribuya al crecimiento personal del estudiante, proporcionando un ambiente de trabajo positivo y oportunidades para desarrollar habilidades interpersonales y profesionales.

5 MÉTODO ELECTRE

Tenemos que maximizar los subcriterios C1.1, C1.2, C2.1, C3.1, C3.2, C4.1 y C4.2, y solo tenemos que minimizar el subcriterio C2.2.

Los pesos asignados a cada uno de estos subcriterios son: $W=(0.25, 0.10, 0.15, 0.15, 0.05, 0.05, 0.15, 0.10)$.

La matriz de decisión viene recogida en la siguiente tabla:

Table 1: Matriz de Decisión (Método ELECTRE)

| | Salario | Beneficios | Cercania | Estres | Desarrollo | Relacion_Estudios | Satisfaccion | Relaciones |
|-------------|---------|------------|----------|--------|------------|-------------------|--------------|------------|
| Hostelería | 8 | 2 | 3 | 7 | 4 | 1 | 6 | 8 |
| Tutor | 15 | 1 | 8 | 2 | 8 | 9 | 9 | 3 |
| RRPP | 10 | 5 | 5 | 8 | 5 | 2 | 7 | 10 |
| Dependiente | 9 | 1 | 7 | 6 | 5 | 3 | 5 | 7 |
| Cuidador | 12 | 3 | 7 | 5 | 3 | 5 | 8 | 5 |

Vamos a trabajar con un valor del umbral de concordancia (α) igual a 0.65, y un vector de umbrales de discordancia (d) igual a (6,4,5,4,6,4,5,5).

Ahora procedemos a aplicar el método ELECTRE para ordenar las alternativas.

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

metodo_electre = multicriterio.crea.matrizdecision(c(8,2,3,7,4,1,6,8,
15,1,8,2,8,9,9,3,
10,5,5,8,5,2,7,10,
9,1,7,6,5,3,5,7,
12,3,7,5,3,5,8,5),
numalternativas = 5,
numcriterios = 8)

metodo_electre

##      C1 C2 C3 C4 C5 C6 C7 C8
## a1  8  2  3  7  4  1  6  8
## a2 15  1  8  2  8  9  9  3
## a3 10  5  5  8  5  2  7 10
## a4  9  1  7  6  5  3  5  7
## a5 12  3  7  5  3  5  8  5

salida_electre= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10, 0.15, 0.15, 0.05, 0.05, 0.15, 0.10),
↪ nivel.concordancia.minimo.alpha=0.65,
↪ no.se.compensan=c(6,4,5,4,6,4,5,5),que.alternativas = TRUE)

salida_electre

## $datos
##           C1 C2 C3 C4 C5 C6 C7 C8
## a1      8.00 2.0 3.00 7.00 4.00 1.00 6.00 8.0
## a2     15.00 1.0 8.00 2.00 8.00 9.00 9.00 3.0
## a3     10.00 5.0 5.00 8.00 5.00 2.00 7.00 10.0
## a4      9.00 1.0 7.00 6.00 5.00 3.00 5.00 7.0
```

```

## a5          12.00 3.0 7.00 5.00 3.00 5.00 8.00 5.0
## pesos.criterios 0.25 0.1 0.15 0.15 0.05 0.05 0.15 0.1
## no.se.compensan 6.00 4.0 5.00 4.00 6.00 4.00 5.00 5.0
##
## $alpha
## [1] 0.65
##
## $Imas
## , , C1
##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 TRUE  FALSE TRUE  TRUE  TRUE
## a3 TRUE  FALSE FALSE TRUE  FALSE
## a4 TRUE  FALSE FALSE FALSE FALSE
## a5 TRUE  FALSE TRUE  TRUE  FALSE
##
## , , C2
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE  FALSE TRUE  FALSE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 TRUE  TRUE  FALSE TRUE  TRUE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5 TRUE  TRUE  FALSE TRUE  FALSE
##
## , , C3
##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 TRUE  FALSE TRUE  TRUE  TRUE
## a3 TRUE  FALSE FALSE FALSE FALSE
## a4 TRUE  FALSE TRUE  FALSE FALSE
## a5 TRUE  FALSE TRUE  FALSE FALSE
##
## , , C4
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE  FALSE TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 TRUE  TRUE  FALSE TRUE  TRUE
## a4 FALSE TRUE  FALSE FALSE TRUE
## a5 FALSE TRUE  FALSE FALSE FALSE
##
## , , C5
##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE FALSE TRUE
## a2 TRUE  FALSE TRUE  TRUE  TRUE
## a3 TRUE  FALSE FALSE FALSE TRUE
## a4 TRUE  FALSE FALSE FALSE TRUE
## a5 FALSE FALSE FALSE FALSE FALSE
##
## , , C6
##

```

```

##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 TRUE  FALSE TRUE  TRUE  TRUE
## a3 TRUE  FALSE FALSE FALSE FALSE
## a4 TRUE  FALSE TRUE  FALSE FALSE
## a5 TRUE  FALSE TRUE  TRUE  FALSE
##
## , , C7
##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE TRUE  FALSE
## a2 TRUE  FALSE TRUE  TRUE  TRUE
## a3 TRUE  FALSE FALSE TRUE  FALSE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5 TRUE  FALSE TRUE  TRUE  FALSE
##
## , , C8
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE  FALSE TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 TRUE  TRUE  FALSE TRUE  TRUE
## a4 FALSE TRUE  FALSE FALSE TRUE
## a5 FALSE TRUE  FALSE FALSE FALSE
##
## $Iigual
## , , C1
##
##      a1      a2      a3      a4      a5
## a1 TRUE  FALSE FALSE FALSE FALSE
## a2 FALSE TRUE  FALSE FALSE FALSE
## a3 FALSE FALSE TRUE  FALSE FALSE
## a4 FALSE FALSE FALSE TRUE  FALSE
## a5 FALSE FALSE FALSE FALSE TRUE
##
## , , C2
##
##      a1      a2      a3      a4      a5
## a1 TRUE  FALSE FALSE FALSE FALSE
## a2 FALSE TRUE  FALSE TRUE  FALSE
## a3 FALSE FALSE TRUE  FALSE FALSE
## a4 FALSE TRUE  FALSE TRUE  FALSE
## a5 FALSE FALSE FALSE FALSE TRUE
##
## , , C3
##
##      a1      a2      a3      a4      a5
## a1 TRUE  FALSE FALSE FALSE FALSE
## a2 FALSE TRUE  FALSE FALSE FALSE
## a3 FALSE FALSE TRUE  FALSE FALSE
## a4 FALSE FALSE FALSE TRUE  TRUE
## a5 FALSE FALSE FALSE TRUE  TRUE
##
## , , C4

```

```

##
##      a1      a2      a3      a4      a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C5
##
##      a1      a2      a3      a4      a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE  TRUE FALSE
## a4 FALSE FALSE  TRUE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C6
##
##      a1      a2      a3      a4      a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C7
##
##      a1      a2      a3      a4      a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
## , , C8
##
##      a1      a2      a3      a4      a5
## a1  TRUE FALSE FALSE FALSE FALSE
## a2 FALSE  TRUE FALSE FALSE FALSE
## a3 FALSE FALSE  TRUE FALSE FALSE
## a4 FALSE FALSE FALSE  TRUE FALSE
## a5 FALSE FALSE FALSE FALSE  TRUE
##
##
## $Imenos
## , , C1
##
##      a1      a2      a3      a4      a5
## a1 FALSE  TRUE  TRUE  TRUE  TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE  TRUE FALSE FALSE  TRUE
## a4 FALSE  TRUE  TRUE FALSE  TRUE
## a5 FALSE  TRUE FALSE FALSE FALSE
##

```



```

## , , C2
##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE TRUE FALSE TRUE
## a2 TRUE FALSE TRUE FALSE TRUE
## a3 FALSE FALSE FALSE FALSE FALSE
## a4 TRUE FALSE TRUE FALSE TRUE
## a5 FALSE FALSE TRUE FALSE FALSE
##
## , , C3
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE TRUE TRUE TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE TRUE FALSE TRUE TRUE
## a4 FALSE TRUE FALSE FALSE FALSE
## a5 FALSE TRUE FALSE FALSE FALSE
##
## , , C4
##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE TRUE FALSE FALSE
## a2 TRUE FALSE TRUE TRUE TRUE
## a3 FALSE FALSE FALSE FALSE FALSE
## a4 TRUE FALSE TRUE FALSE FALSE
## a5 TRUE FALSE TRUE TRUE FALSE
##
## , , C5
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE TRUE TRUE FALSE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE TRUE FALSE FALSE FALSE
## a4 FALSE TRUE FALSE FALSE FALSE
## a5 TRUE TRUE TRUE TRUE FALSE
##
## , , C6
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE TRUE TRUE TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE TRUE FALSE TRUE TRUE
## a4 FALSE TRUE FALSE FALSE TRUE
## a5 FALSE TRUE FALSE FALSE FALSE
##
## , , C7
##
##      a1      a2      a3      a4      a5
## a1 FALSE TRUE TRUE FALSE TRUE
## a2 FALSE FALSE FALSE FALSE FALSE
## a3 FALSE TRUE FALSE FALSE TRUE
## a4 TRUE TRUE TRUE FALSE TRUE
## a5 FALSE TRUE FALSE FALSE FALSE
##
## , , C8

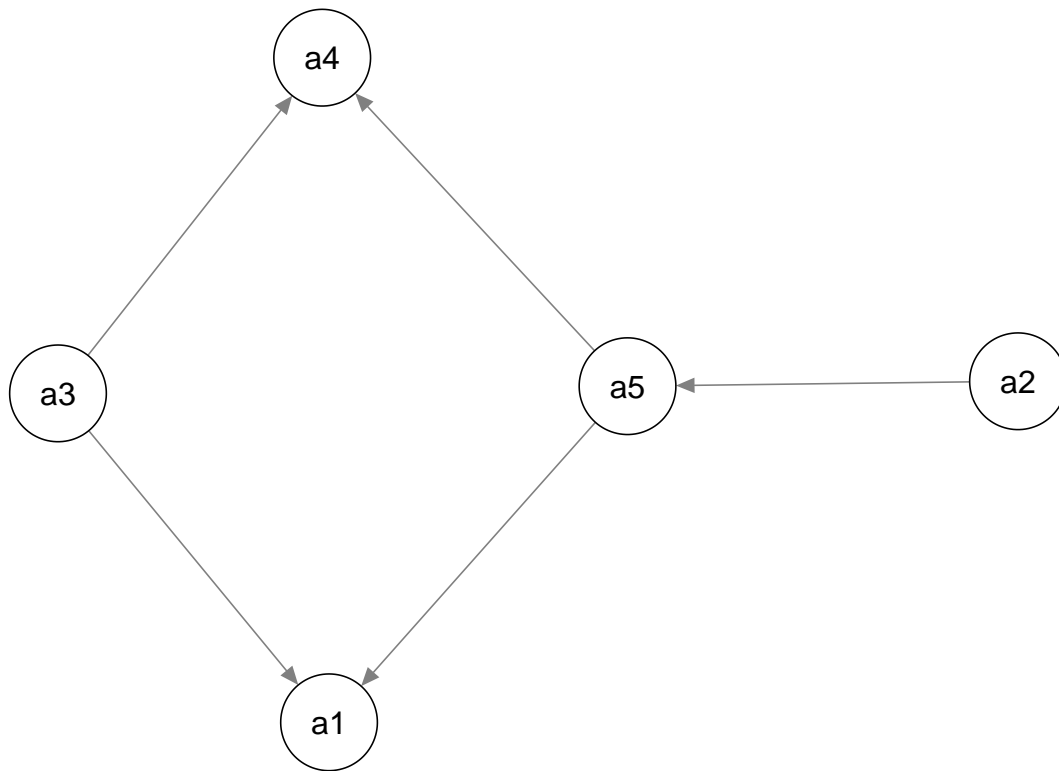
```

```

##
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE  TRUE FALSE FALSE
## a2  TRUE FALSE  TRUE  TRUE  TRUE
## a3 FALSE FALSE FALSE FALSE FALSE
## a4  TRUE FALSE  TRUE FALSE FALSE
## a5  TRUE FALSE  TRUE  TRUE FALSE
##
##
## $ind.concordancia
##      a1      a2      a3      a4      a5
## a1 1.00 0.35 0.00 0.50 0.30
## a2 0.65 1.00 0.65 0.75 0.65
## a3 1.00 0.35 1.00 0.80 0.40
## a4 0.50 0.35 0.25 1.00 0.45
## a5 0.70 0.35 0.60 0.70 1.00
##
## $ind.concordancia.gorro
##      a1      a2      a3      a4      a5
## a1      NaN 0.5384615 0.0000000 1.000000 0.4285714
## a2 1.857143      NaN 1.8571429 2.600000 1.8571429
## a3      Inf 0.5384615      NaN 3.750000 0.6666667
## a4 1.000000 0.3846154 0.2666667      NaN 0.5454545
## a5 2.333333 0.5384615 1.5000000 1.833333      NaN
##
## $test.concordancia
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2  TRUE FALSE  TRUE  TRUE  TRUE
## a3  TRUE FALSE FALSE  TRUE FALSE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5  TRUE FALSE FALSE  TRUE FALSE
##
## $test.discordancia
##      a1      a2      a3      a4      a5
## a1  TRUE FALSE  TRUE  TRUE FALSE
## a2 FALSE  TRUE FALSE FALSE  TRUE
## a3  TRUE FALSE  TRUE  TRUE  TRUE
## a4  TRUE FALSE FALSE  TRUE  TRUE
## a5  TRUE FALSE FALSE  TRUE  TRUE
##
## $relacion.dominante
##      a1      a2      a3      a4      a5
## a1 FALSE FALSE FALSE FALSE FALSE
## a2 FALSE FALSE FALSE FALSE  TRUE
## a3  TRUE FALSE FALSE  TRUE FALSE
## a4 FALSE FALSE FALSE FALSE FALSE
## a5  TRUE FALSE FALSE  TRUE FALSE
##
## $nucleo_aprox
## a2 a3
## 2 3

```

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



En este grafo de dominancia se representa visualmente la relación de superación obtenida al aplicar los test de concordancia y discordancia. Los resultados obtenidos son:

A2 domina a a5. A5 domina a las alternativas A1 y A4. A3 domina a las alternativas A4 y A1.

```
salida_electre$ nucleo_aprox
```

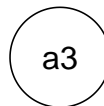
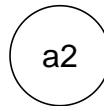
```
## a2 a3
## 2 3
```

Este es el núcleo aproximado, que es el conjunto de soluciones que no son dominadas por otras alternativas. El resultado es A2 y A3. Esto nos indica que las alternativas “Profesor particular/Tutor” y “RRPP/Promotor de eventos” son las más adecuadas, ya que ninguna de las otras opciones logró superarlas tras la aplicación de los subcriterios y umbrales.

Para intentar quedarnos con una única alternativa óptima, vamos a reducir el grafo a las alternativas en el núcleo aproximado. Aplicamos el método ELECTRE de nuevo para ver si una alternativa domina a la otra en este subconjunto.

```
salida_electre_b= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10),
→ nivel.concordancia.minimo.alpha=0.65,
→ no.se.compensan=c(6,4,5,4,6,4,5,5),que.alternativas = c(2,3))

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



El grafo nos indica que no hay dominancia de una alternativa sobre la otra, ya que no aparece ninguna flecha.

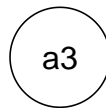
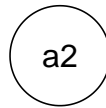
```
salida_electre_b$nucleo_aprox
```

```
## a2 a3  
## 1 2
```

Después de reducir el análisis al subconjunto A2,A3, seguimos teniendo que el núcleo aproximado final sigue siendo A2,A3. Los números 1 y 2 solo indican el orden de estas alternativas en el subconjunto.

Ahora vamos a probar a reducir el umbral alpha para ver si alguna alternativa domina a la otra e intentar resolver la incomparabilidad. Si ajustamos los umbrales podemos intentar forzar esta dominancia para seleccionar una única alternativa óptima.

```
salida_electre_c= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=  
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10),  
↪ nivel.concordancia.minimo.alpha=0.10,  
↪ no.se.compensan=c(6,4,5,4,6,4,5,5),que.alternativas = c(2,3))  
  
qgraph::qgraph(salida_electre_c$relacion.dominante)
```



Probando a reducir el valor de alpha numerosas veces hasta llegar a un límite extremadamente permisivo (0.10), sigue sin aparecer dominancia en el grafo. Esto indica que necesitamos modificar también los valores del vector de discordancia.

```
salida_electre_d= multicriterio.metodoELECTRE_I(metodo_electre, pesos.criterios=
c(0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10),
↪ nivel.concordancia.minimo.alpha=0.65,
↪ no.se.compensan=c(6,5,5,7,6,4,5,8),que.alternativas = c(2,3))

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



Tras probar una serie de valores para el vector de discordancia, se obtiene que A2 domina a A3 con $d=(6,5,5,7,6,4,5,8)$, que es el vector de discordancia mínimamente ajustado. Tras observar esta solución, podemos decir que la alternativa A2 domina a la alternativa A3, por lo que la mejor opción es “Profesor particular/Tutor”.

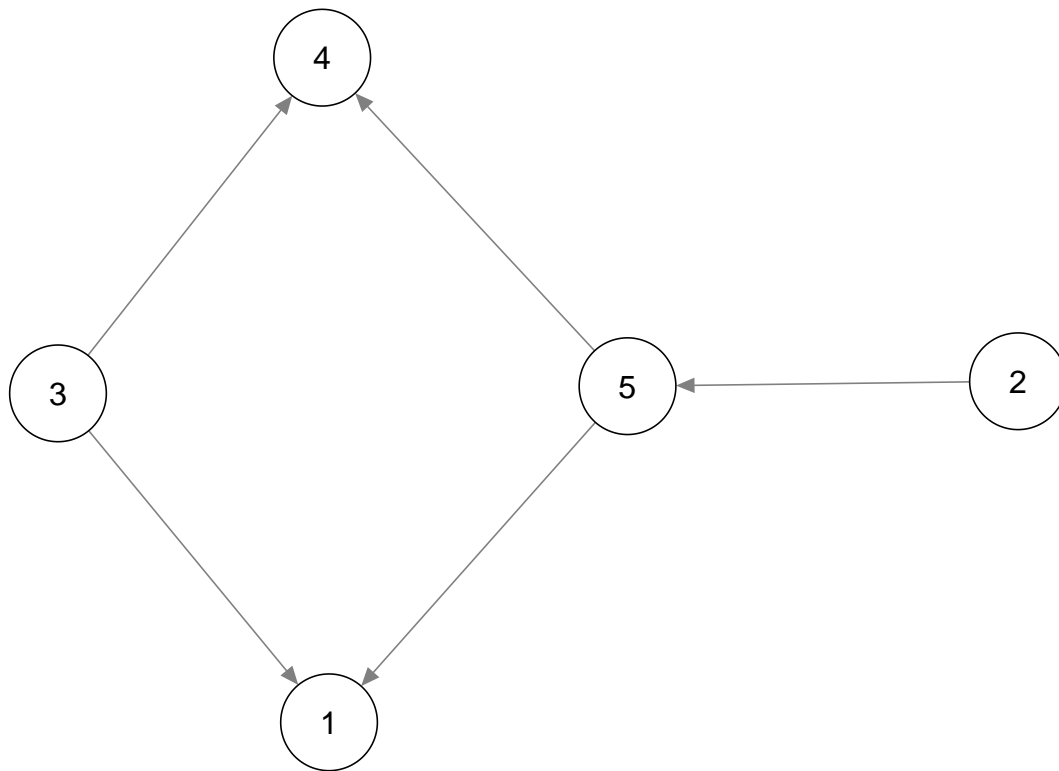
6 CÁLCULOS EN EL MÉTODO ELECTRE I

```
e1 = func_ELECTRE_Completo(salida_electre)
```

```
e1$Grafo
```

```
## De A
## 1 2 5
## 2 3 1
## 3 3 4
## 4 5 1
## 5 5 4
```

```
qgraph::qgraph(e1$Grafo)
```



```
e1$Nucleo
```

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

Con el método ELECTRE I nos salen los mismos resultados que con el método ELECTRE anterior. El núcleo sigue estando compuesto por las alternativas 2 y 3, y siguen existiendo las mismas dominancias entre las alternativas, como vemos con el código e1\$Grafo.

7 MÉTODO PROMETHEE

Tenemos que maximizar los subcriterios C1.1, C1.2, C2.1, C3.1, C3.2, C4.1 y C4.2, y solo tenemos que minimizar el subcriterio C2.2. Para crear la matriz para este método, multiplicamos la columna del subcriterio 2.2 por -1.

```

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

tabdecision_promethee = multicriterio.crea.matrizdecision(c( 8, 2, 3, -7, 4, 1, 6, 8, 15,
→ 1, 8, -2, 8, 9, 9, 3, 10, 5, 5, -8, 5, 2, 7, 10, 9, 1, 7, -6, 5, 3, 5, 7, 12, 3, 7,
→ -5, 3, 5, 8, 5), numalternativas = 5, numcriterios = 8,
→ v.nombresalt=c("Hostelería","Tutor","RRPP","Dependiente","Cuidador"),
→ v.nombrescri=c("Salario", "Beneficios", "Cercanía", "Estrés", "Desarrollo",
→ "Relación_Estudios", "Satisfacción", "Relaciones"))

tabdecision_promethee

```

```
##          Salario Beneficios Cercanía Estrés Desarrollo Relación_Estudios
## Hostelería      8          2          3      -7          4          1
## Tutor           15          1          8      -2          8          9
## RRPP            10          5          5      -8          5          2
## Dependiente     9          1          7      -6          5          3
## Cuidador        12          3          7      -5          3          5
##          Satisfacción Relaciones
## Hostelería      6          8
## Tutor           9          3
## RRPP            7          10
## Dependiente     5          7
## Cuidador        8          5
```

Vamos a asignar los siguientes pesos a los criterios: $W=(0.20, 0.15, 0.20, 0.15, 0.10, 0.05, 0.10, 0.05)$.

A continuación se define la matriz de funciones de preferencia denominada `tab.fpref` en el código R, esencial para aplicar este método PROMETHEE. Esta matriz indica el tipo de función, el umbral de indiferencia (q), el umbral de preferencia (p) y s .

```
pesos.criterios=c(0.20, 0.15, 0.20, 0.15, 0.10, 0.05, 0.10, 0.05)

tab.fpref = matrix(c(2,3,1,0,1,0,1,0,3,0,5,0,5,2,5,0,1,0,1,0,4,3,6,0,6,0,1,2,3,0,4,0),
  ↪ ncol=4, byrow=TRUE)

tab.fpref
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  2   3   1   0
## [2,]  1   0   1   0
## [3,]  3   0   5   0
## [4,]  5   2   5   0
## [5,]  1   0   1   0
## [6,]  4   3   6   0
## [7,]  6   0   1   2
## [8,]  3   0   4   0
```

7.1 REPRESENTACIÓN GRÁFICA DE LAS FUNCIONES DE PREFERENCIA

```
fpref.criterio_usual_di <- function(di) {
  # di = vaj - vah
  if (di <= 0) {
    res = 0
  } else {
    res = 1
  }
  return(res)
}

fpref.cuasi_criterio_di <- function(di, qi) {
  # di = vaj - vah
  if (di <= qi) {
    res = 0
  } else {
    res = 1
  }
}
```



```

    }
    return(res)
}

fpref.criterio_preflineal_di <- function(di, pi) {
  #  $di = v_{aj} - v_{ah}$ 
  if (di <= 0) {
    res = 0
  } else if (di > pi) {
    res = 1
  } else {
    res = di / pi
  }
  return(res)
}

fpref.criterio_nivel_di <- function(di, qi, pi) {
  #  $di = v_{aj} - v_{ah}$ 
  if (di <= qi) {
    res = 0
  } else if (di > pi) {
    res = 1
  } else {
    res = 0.5
  }
  return(res)
}

fpref.criterio_preflineal_indif_di <- function(di, qi, pi) {
  #  $di = v_{aj} - v_{ah}$ 
  if (di <= qi) {
    res = 0
  } else if (di > pi) {
    res = 1
  } else {
    res = (di - qi) / (pi - qi)
  }
  return(res)
}

fpref.criterio_gaussiano_di <- function(di, qi, pi, si) {
  #  $di = v_{aj} - v_{ah}$ .
  if (di <= 0) {
    res = 0
  } else {
    res = 1 - exp(-(di^2) / (2 * si^2))
  }
  return(res)
}

x = seq(-10,10,length.out=100)

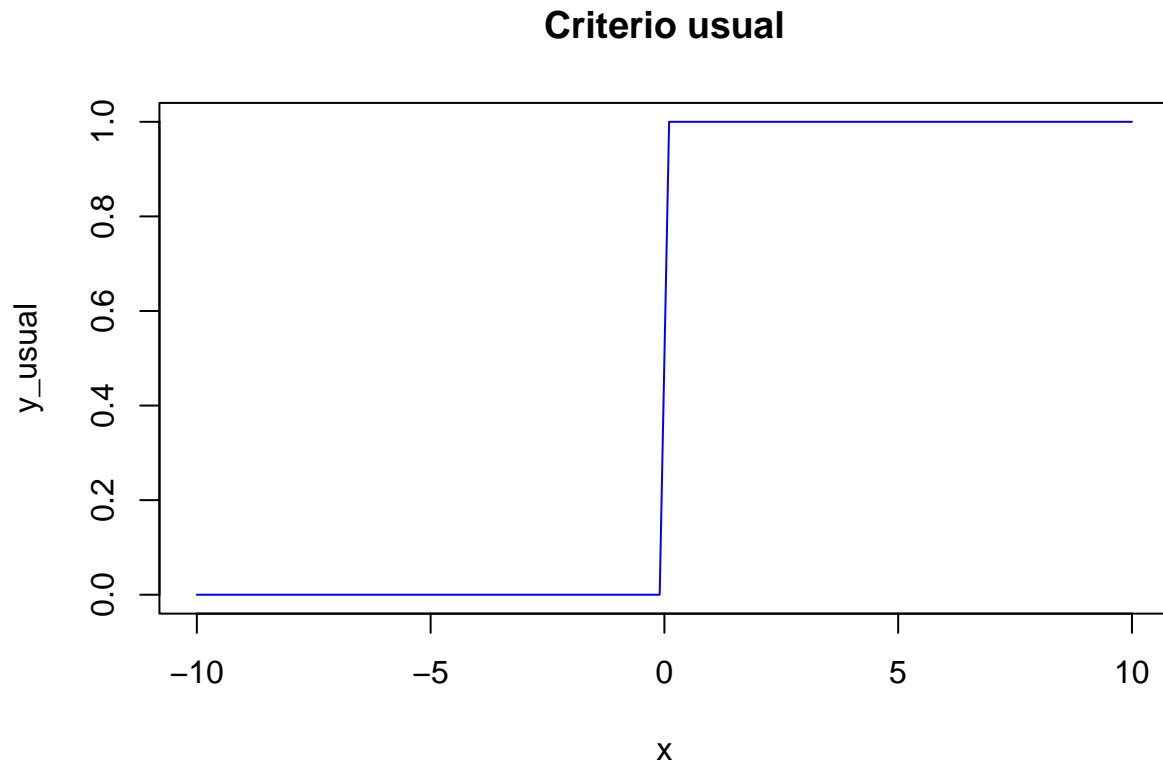
```

```

# GRÁFICA DEL CRITERIO USUAL

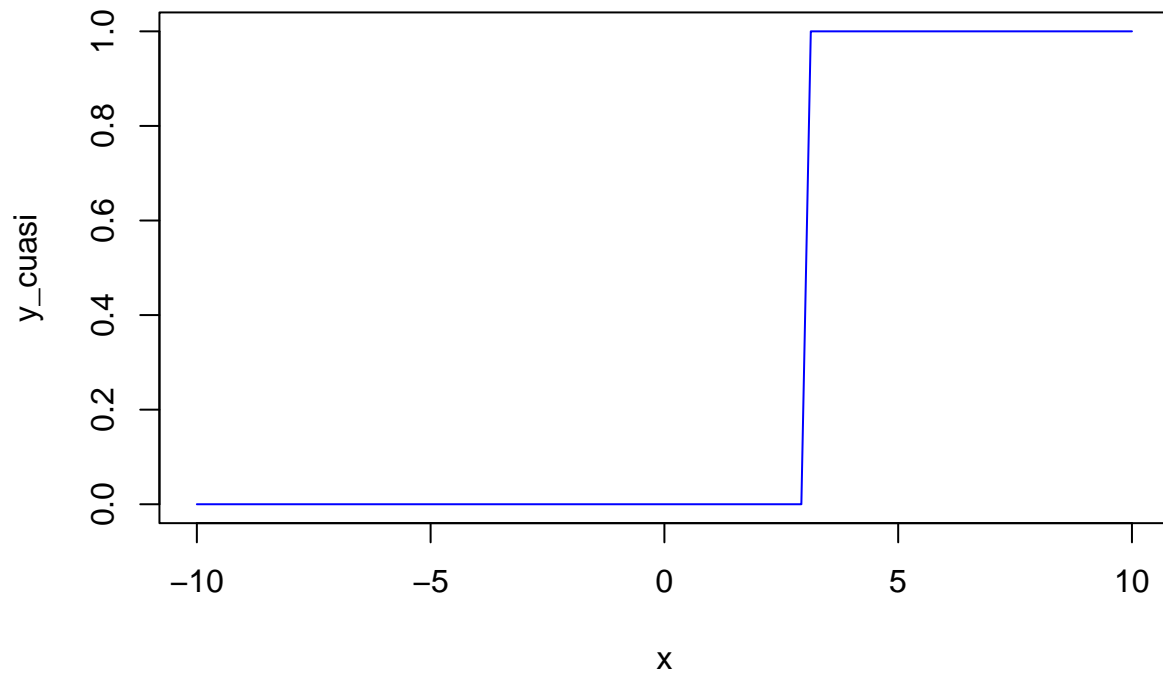
```

```
y_usual = sapply(x,fpref.criterio_usual_di)
plot(x,y_usual,type="l",col="blue",main="Criterio usual")
```



```
# GRÁFICA DEL CRITERIO CUASI CON Q=3
y_cuasi <- sapply(x, function(xx) fpref.cuasi_criterio_di(xx, qi = 3))
plot(x,y_cuasi,type="l",col="blue",main="Criterio cuasi con Q=3")
```

Criterio cuasi con Q=3



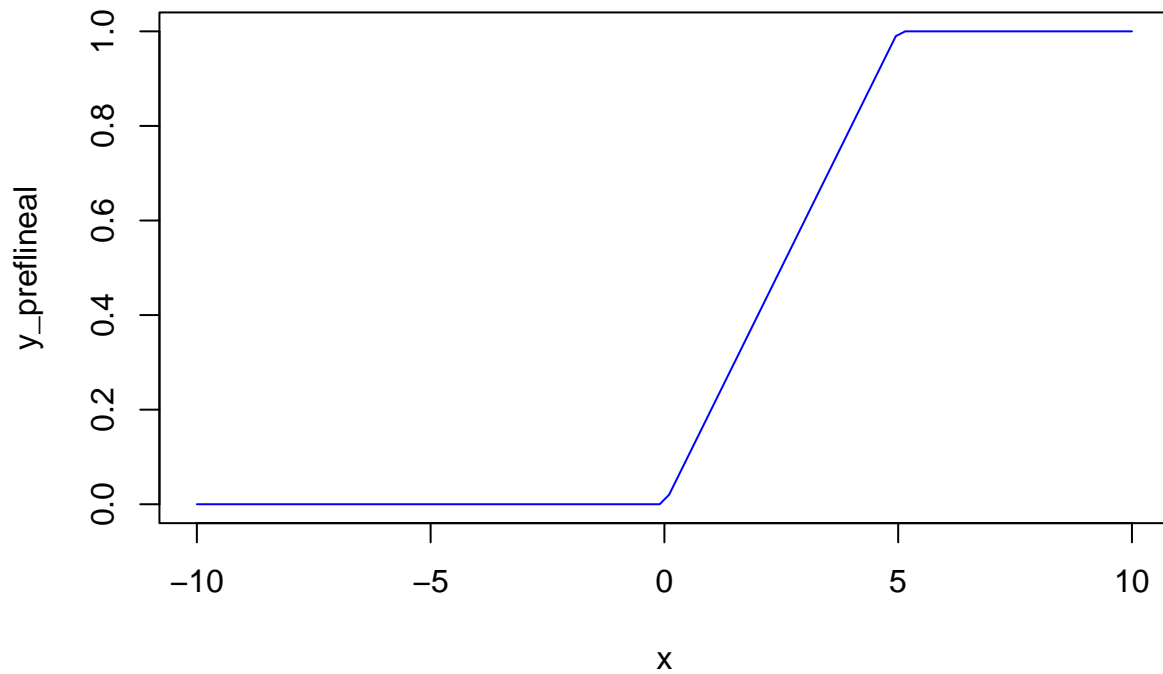
```
# Usamos q=3 porque consideramos que una diferencia en el salario de 1 o 2 euros no es  
↳ suficiente, pero 3 sí. Si un empleo paga 3 euros más que otro, ese empleo se  
↳ considera estrictamente superior.
```

```
# GRÁFICA DEL CRITERIO DE PREFERENCIA LINEAL CON P=5
```

```
y_prelineal <- sapply(x, function(xx) fpref.criterio_prelineal_di(xx, pi = 5))
```

```
plot(x,y_prelineal,type="l",col="blue",main="Criterio preferencia lineal con P=5")
```

Criterio preferencia lineal con P=5



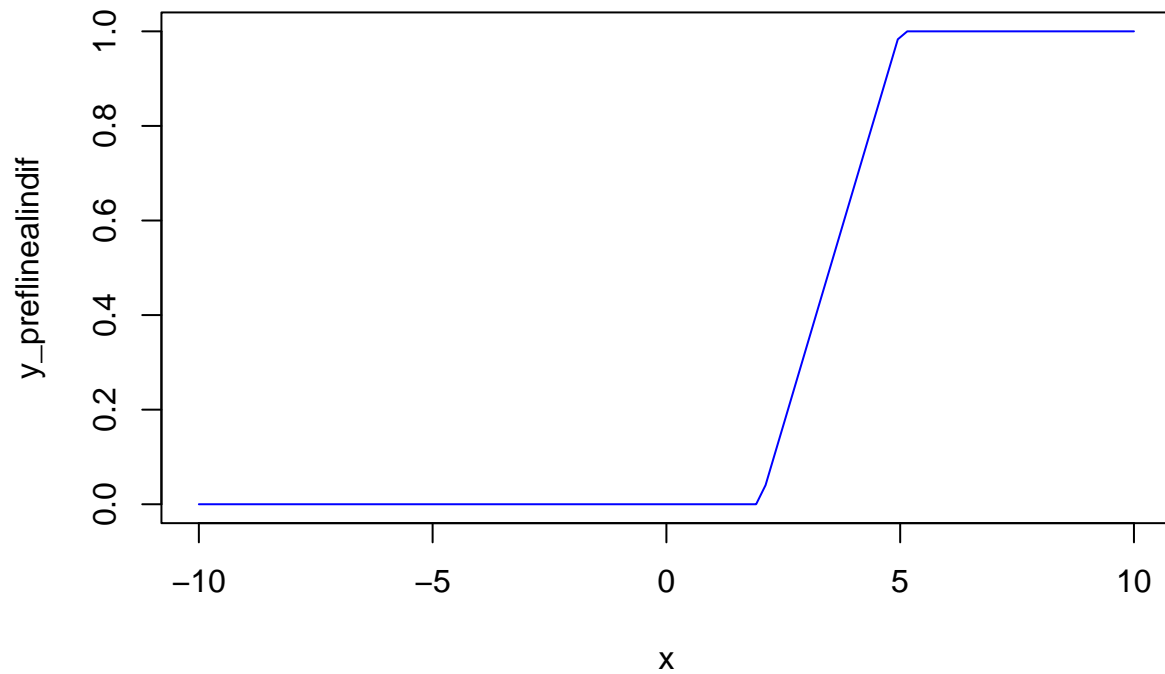
5 es la diferencia necesaria para considerar que un empleo es preferible a otro en términos de cercanía al hogar. Si un empleo está a 5 km o más cerca, se considera una ventaja significativa.

GRÁFICA DEL CRITERIO DE PREFERENCIA LINEAL INDIFERENTE CON Q=2,P=5

```
y_prelinealindif <- sapply(x, function(xx) fpref.criterio_prelineal_indif_di(xx, qi=2,  
↳ pi = 5))
```

```
plot(x,y_prelinealindif,type="l",col="blue",main="Criterio preferencia lineal  
↳ indiferente con Q=2,P=5")
```

Criterio preferencia lineal indiferente con Q=2,P=5



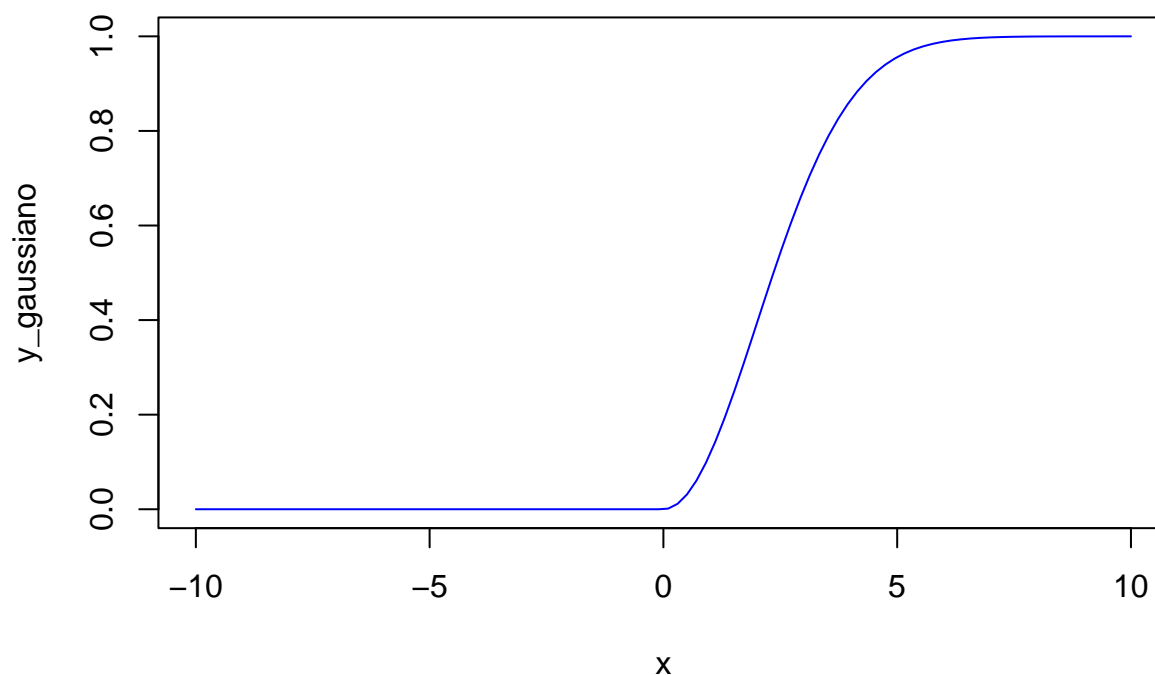
*# Una diferencia de hasta 2 puntos de carga mental/estrés en un empleo se considera
↳ indiferente. Una diferencia de 5 puntos o más ya es una ventaja significativa.*

GRÁFICA DEL CRITERIO GAUSSIANO

```
y_gaussiano <- sapply(x, function(xx) fpref.criterio_gaussiano_di(xx, qi= 0, pi = 1,  
↳ si=2))
```

```
plot(x,y_gaussiano,type="l",col="blue",main="Criterio gaussiano")
```

Criterio gaussiano



7.2 APLICACIÓN DEL MÉTODO PROMETHEE I

```
tab.promethee_i = multicriterio.metodo.promethee_i(tabdecision_promethee,
↳ pesos.criterios,tab.fpref)
```

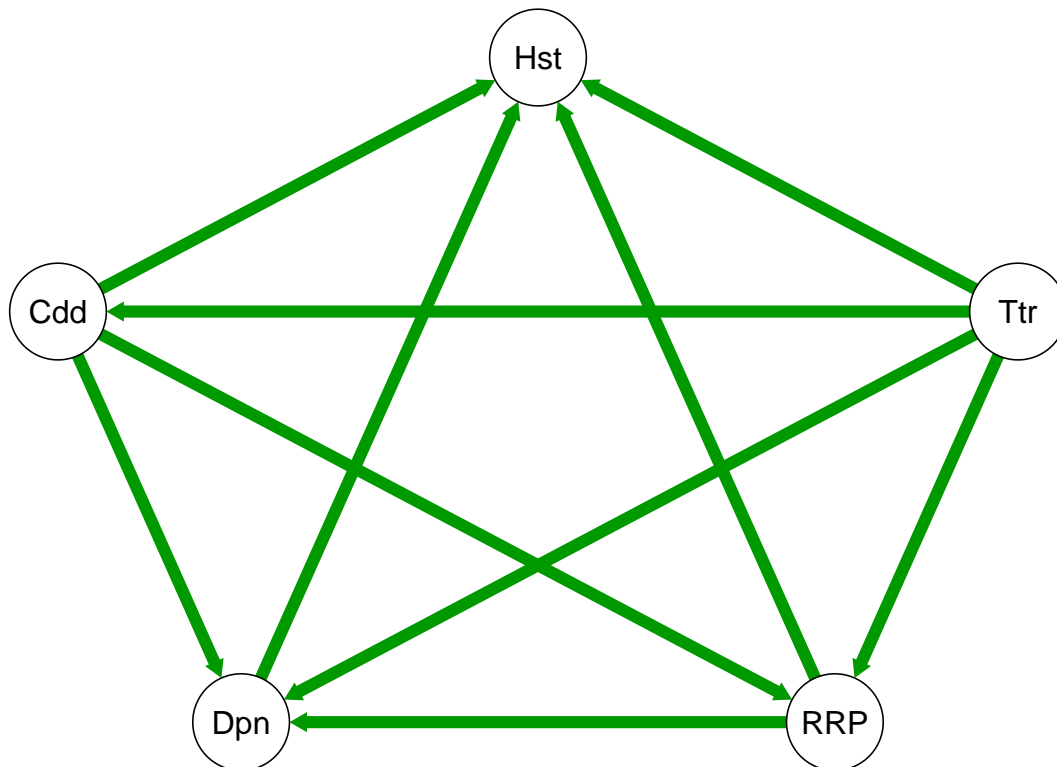
```
tab.promethee_i
```

```
## $tabla.indices
##      Hostelería Tutor      RRPP Dependiente  Cuidador
## Hostelería  0.0000000 0.200 0.0000000   0.1742503 0.1375000
## Tutor      0.7675348 0.000 0.6593469   0.5764665 0.2267503
## RRPP       0.3667503 0.200 0.0000000   0.2268469 0.3000000
## Dependiente 0.2600000 0.050 0.0800000   0.0000000 0.1250000
## Cuidador    0.5743469 0.175 0.1417503   0.2175348 0.0000000
##
## $vflujos.ent
##      Hostelería      Tutor      RRPP Dependiente  Cuidador
##      0.5117503    2.2300985    1.0935972    0.5150000    1.1086320
##
## $vflujos.sal
##      Hostelería      Tutor      RRPP Dependiente  Cuidador
##      1.9686320    0.6250000    0.8810972    1.1950985    0.7892503
##
## $tablarelacionsupera
##      Hostelería Tutor RRPP Dependiente Cuidador
## Hostelería      0.5  0.0  0.0          0.0      0.0
```

```
## Tutor          1.0  0.5  1.0          1.0    1.0
## RRPP           1.0  0.0  0.5          1.0    0.0
## Dependiente    1.0  0.0  0.0          0.5    0.0
## Cuidador       1.0  0.0  1.0          1.0    0.5
```

```
qgraph::qgraph(tab.promethee_i$tablarelacionsupera)
```

```
## Warning in abbreviate(colnames(input), 3): abreviatura utilizada con caracteres
## no ASCII
```



```
tab.promethee_i$tablarelacionsupera
```

```
##          Hostelería Tutor RRPP Dependiente Cuidador
## Hostelería      0.5  0.0  0.0          0.0    0.0
## Tutor           1.0  0.5  1.0          1.0    1.0
## RRPP            1.0  0.0  0.5          1.0    0.0
## Dependiente     1.0  0.0  0.0          0.5    0.0
## Cuidador        1.0  0.0  1.0          1.0    0.5
```

En el grafo se representa esta tabla de la relación que hay entre las alternativas. Donde aparezca un 1, significará que la alternativa de esa fila supera a la alternativa de esta columna. Por ejemplo, de la fila de “RRPP” sacamos que supera a “Hostelería” y “Dependiente”, por tanto, en la fila de “Hostelería” y “Dependiente” debería de aparecer un 0 en la columna de “RRPP”, como vemos que ocurre. Así vamos conociendo estas relaciones.

En el grafo, al igual que ocurre en la tabla anterior, vemos que todas las alternativas son superadas por al menos otra alternativa, menos “Profesor particular/Tutor”, que no es superada por ninguna. El método Promethee I nos dice que la mejor alternativa es esta.

7.3 APLICACIÓN DEL MÉTODO PROMETHEE II

```
tab.promethee_ii = multicriterio.metodo.promethee_ii(tabdecision_promethee,  
↳ pesos.criterios,tab.fpref)
```

```
tab.promethee_ii
```

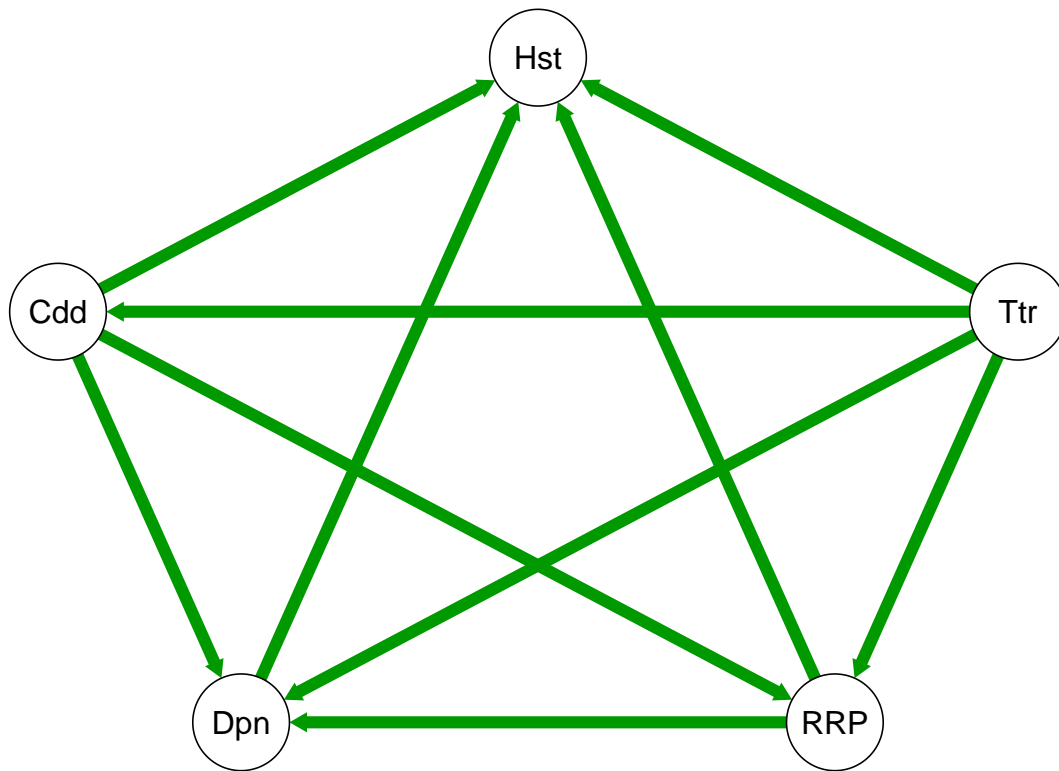
```
## $tabla.indices  
##  
## Hostelería Tutor RRPP Dependiente Cuidador  
## Hostelería 0.0000000 0.200 0.0000000 0.1742503 0.1375000  
## Tutor 0.7675348 0.000 0.6593469 0.5764665 0.2267503  
## RRPP 0.3667503 0.200 0.0000000 0.2268469 0.3000000  
## Dependiente 0.2600000 0.050 0.0800000 0.0000000 0.1250000  
## Cuidador 0.5743469 0.175 0.1417503 0.2175348 0.0000000  
##
```

```
## $vflujos.netos  
## Hostelería Tutor RRPP Dependiente Cuidador  
## -1.4568817 1.6050985 0.2125000 -0.6800985 0.3193817  
##
```

```
## $tablarelacionsupera  
## Hostelería Tutor RRPP Dependiente Cuidador  
## Hostelería 0.5 0.0 0.0 0.0 0.0  
## Tutor 1.0 0.5 1.0 1.0 1.0  
## RRPP 1.0 0.0 0.5 1.0 0.0  
## Dependiente 1.0 0.0 0.0 0.5 0.0  
## Cuidador 1.0 0.0 1.0 1.0 0.5
```

```
qgraph::qgraph(tab.promethee_ii$tablarelacionsupera)
```

```
## Warning in abbreviate(colnames(input), 3): abreviatura utilizada con caracteres  
## no ASCII
```

7.3.1 ORDENACIÓN FINAL DE LAS ALTERNATIVAS POR EL MÉTODO DE PROMETHEE II

```
order(tab.promethee_ii$vflujos.netos, decreasing = TRUE)
```

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

La mejor alternativa vuelve a ser la de “Profesor particular/Tutor”. Después le sigue “Cuidador de niños”, “RRPP/Promotor de eventos”, “Dependiente”, y por último como peor opción, “Empleo en hostelería/Servicios”.

7.4 MÉTODO PROMETHEE I MEDIAS

```
tab.promethee_i_med = multicriterio.metodo.promethee_i_med(tabdecision_promethee,
  ↪ pesos.criterios,tab.fpref)
```

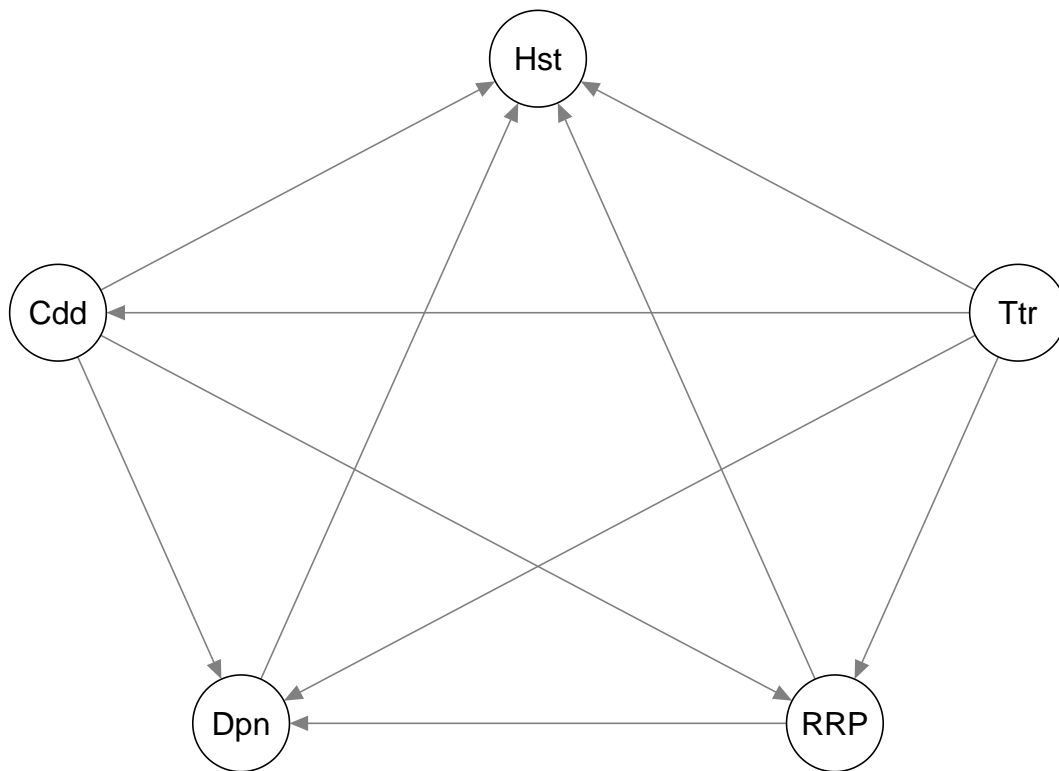
```
tab.promethee_i_med
```

```
## $tabla.indices
##           Hostelería Tutor      RRPP Dependiente  Cuidador
## Hostelería  0.0000000 0.200 0.0000000  0.1742503 0.1375000
## Tutor       0.7675348 0.000 0.6593469  0.5764665 0.2267503
## RRPP        0.3667503 0.200 0.0000000  0.2268469 0.3000000
## Dependiente 0.2600000 0.050 0.0800000  0.0000000 0.1250000
## Cuidador    0.5743469 0.175 0.1417503  0.2175348 0.0000000
##
## $vflujos.ent
```

```
## Hostelería Tutor RRPP Dependiente Cuidador
## 0.1279376 0.5575246 0.2733993 0.1287500 0.2771580
##
## $vflujos.sal
## Hostelería Tutor RRPP Dependiente Cuidador
## 0.4921580 0.1562500 0.2202743 0.2987746 0.1973126
##
## $tablarelacionsupera
## Hostelería Tutor RRPP Dependiente Cuidador
## Hostelería 0 0 0 0 0
## Tutor 1 0 1 1 1
## RRPP 1 0 0 1 0
## Dependiente 1 0 0 0 0
## Cuidador 1 0 1 1 0
```

```
qgraph::qgraph(tab.promethee_i_med$tablarelacionsupera)
```

```
## Warning in abbreviate(colnames(input), 3): abreviatura utilizada con caracteres
## no ASCII
```



Si volvemos a observar la tabla de las relaciones entre las alternativas, tenemos exactamente la misma tabla obtenida con el método promethee I sin medias. “Tutor” sigue siendo la mejor alternativa, como también podemos comprobar en el grafo, siendo la única alternativa que no es superada por ninguna otra.

7.5 MÉTODO PROMETHEE II MEDIAS

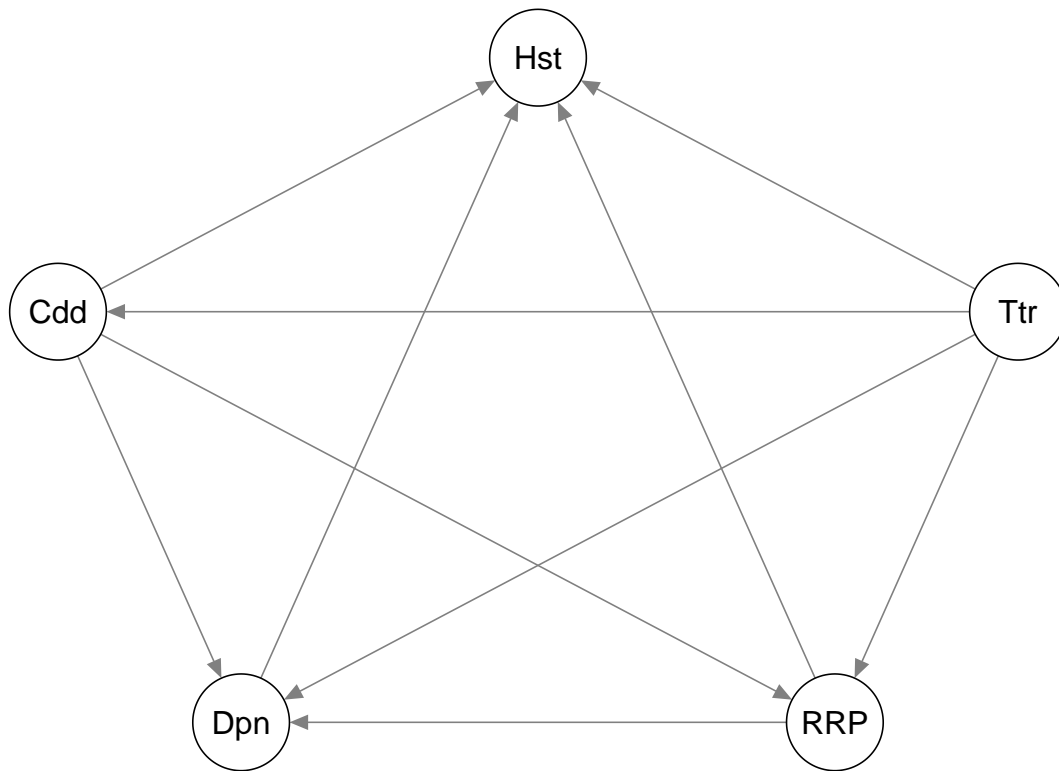
```
tab.promethee_ii_med = multicriterio.metodo.promethee_ii_med(tabdecision_promethee,  
↳ pesos.criterios,tab.fpref)
```

```
tab.promethee_ii_med
```

```
## $tabla.indices  
##           Hostelería Tutor      RRPP Dependiente  Cuidador  
## Hostelería  0.0000000 0.200 0.0000000  0.1742503 0.1375000  
## Tutor       0.7675348 0.000 0.6593469  0.5764665 0.2267503  
## RRPP        0.3667503 0.200 0.0000000  0.2268469 0.3000000  
## Dependiente 0.2600000 0.050 0.0800000  0.0000000 0.1250000  
## Cuidador    0.5743469 0.175 0.1417503  0.2175348 0.0000000  
##  
## $vflujos.netos  
##           Hostelería Tutor      RRPP Dependiente  Cuidador  
## -0.36422042 0.40127462 0.05312500 -0.17002462 0.07984542  
##  
## $tablarelacionsupera  
##           Hostelería Tutor RRPP Dependiente Cuidador  
## Hostelería      0      0      0           0         0  
## Tutor           1      0      1           1         1  
## RRPP            1      0      0           1         0  
## Dependiente     1      0      0           0         0  
## Cuidador        1      0      1           1         0
```

```
qgraph::qgraph(tab.promethee_ii_med$tablarelacionsupera)
```

```
## Warning in abbreviate(colnames(input), 3): abreviatura utilizada con caracteres  
## no ASCII
```



Obtenemos las mismas conclusiones que antes.

7.5.1 ORDENACIÓN FINAL DE LAS ALTERNATIVAS POR EL MÉTODO DE PROMETHEE II MEDIAS

```
order(tab.promethee_ii_med$vflujos.netos, decreasing = TRUE)
```

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

7.5.2 COMPARACIÓN MÉTODO PROMETHEE II SIN MEDIAS Y CON MEDIAS

```
order(tab.promethee_ii$vflujos.netos, decreasing = TRUE)
```

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

```
order(tab.promethee_ii_med$vflujos.netos, decreasing = TRUE)
```

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

Tanto con el método Promethee II con medias como con este mismo método sin medias, obtenemos la misma solución. Las mejores alternativas son, por orden: 1. Profesor particular/Tutor 2. Cuidador de niños 3. RRPP/Promotor de eventos 4. Dependiente 5. Empleo en hostelería/Servicios

8 MÉTODO PROMETHEE WINDOWS

He utilizado el software PROMETHEE Windows para corroborar los resultados obtenidos con R. He introducido los mismos datos de la matriz de decisión, los pesos de los criterios y las funciones de preferencia definidas anteriormente.

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

## Cargando paquete requerido: shape

tabdecision_promethee = multicriterio.crea.matrizdecision(c(8, 2, 3, -7, 4, 1, 6, 8, 15 ,
↪ 1, 8, -2, 8, 9, 9, 3, 10, 5, 5, -8, 5, 2, 7, 10, 9, 1, 7, -6, 5, 3, 5, 7, 12, 3, 7,
↪ -5, 3, 5, 8, 5), numalternativas = 5, numcriterios = 8, v.nombresalt=c("Hostelería",
↪ "Tutor", "RRPP", "Dependiente", "Cuidador"), v.nombrescri=c("Salario", "Beneficios",
↪ "Cercanía", "Estrés", "Desarrollo", "Relación_Estudios", "Satisfacción",
↪ "Relaciones"))

pesos.criterios01=c(0.20, 0.15 , 0.20, 0.15, 0.10, 0.05, 0.10 , 0.05)

tab.fpref01 = matrix(c(2,3,1,0,1,0,1,0,3,0,5,0,5,2,5,0,1,0,1,0,4,3,6,0,6,0,1,2,3,0,4,0),
↪ ncol=4, byrow=TRUE)

res = multicriterio.metodo.promethee_windows(tabdecision_promethee, tab.fpref01,
↪ pesos.criterios01)

res
```

```
## $Escenario
##
## Pesos          Criterio1      Criterio2      Criterio3      Criterio4
## Funciones Preferencias "U-shape (2)" "Usual (1)" "V-shape (3)" "Linear (5)"
## Q: Indiferencia      "3"          "0"          "0"          "2"
## P: Preferencia        "1"          "1"          "5"          "5"
## S: Gausiano           "0"          "0"          "0"          "0"
## Minimo                "8"          "1"          "3"          "-8"
## Maximo                "15"         "5"          "8"          "-2"
## Media                "10.8"       "2.4"        "6"          "-5.6"
## Desviacion Tipica     "2.48"       "1.5"        "1.79"       "2.06"
## Hostelería           "8"          "2"          "3"          "-7"
## Tutor                "15"         "1"          "8"          "-2"
## RRPP                 "10"         "5"          "5"          "-8"
## Dependiente          "9"          "1"          "7"          "-6"
## Cuidador             "12"         "3"          "7"          "-5"
##
## Criterio5      Criterio6      Criterio7      Criterio8
## Pesos          "0.1"      "0.05"      "0.1"      "0.05"
## Funciones Preferencias "Usual (1)" "Level (4)" "Gaussian (6)" "V-shape (3)"
## Q: Indiferencia      "0"        "3"        "0"        "0"
## P: Preferencia        "1"        "6"        "1"        "4"
## S: Gausiano           "0"        "0"        "2"        "0"
## Minimo                "3"        "1"        "5"        "3"
## Maximo                "8"        "9"        "9"        "10"
## Media                "5"        "4"        "7"        "6.6"
## Desviacion Tipica     "1.67"     "2.83"     "1.41"     "2.42"
## Hostelería           "4"        "1"        "6"        "8"
```

```
## Tutor          "8"          "9"          "9"          "3"
## RRPP           "5"          "2"          "7"          "10"
## Dependiente    "5"          "3"          "5"          "7"
## Cuidador       "3"          "5"          "8"          "5"
##
## $Acciones
##      Rango      Phi Phi.mas Phi.menos
## Tutor      1  0.4013  0.5575  0.1562
## Cuidador    2  0.0798  0.2772  0.1973
## RRPP        3  0.0531  0.2734  0.2203
## Dependiente 4 -0.1700  0.1288  0.2988
## Hosteleria   5 -0.3642  0.1279  0.4922
```

Fijándonos en los valores de Phi, tenemos el orden de preferencia de las alternativas. “Tutor” es la alternativa con el valor más alto, un 0.4013, seguida de “Cuidador”, “RRPP” y “Dependiente”. Por último, la peor opción es la de “Hostelería”, que tiene un valor de Phi de -0.3642.

Datos introducidos en el problema resuelto con Promethee:

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

| | Criterio1 | Criterio2 | Criterio3 | Criterio4 | Criterio5 | Criterio6 | Criterio7 | Criterio8 |
|------------------------|-------------|-----------|-------------|------------|-----------|-----------|--------------|-------------|
| Preferencias | | | | | | | | |
| Pesos | 0.2 | 0.15 | 0.2 | 0.15 | 0.1 | 0.05 | 0.1 | 0.05 |
| Funciones Preferencias | U-shape (2) | Usual (1) | V-shape (3) | Linear (5) | Usual (1) | Level (4) | Gaussian (6) | V-shape (3) |
| Q: Indiferencia | 3 | 0 | 0 | 2 | 0 | 3 | 0 | 0 |
| P: Preferencia | 1 | 1 | 5 | 5 | 1 | 6 | 1 | 4 |
| S: Gausiano | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 |
| Minimo | 8 | 1 | 3 | -8 | 3 | 1 | 5 | 3 |
| Estadísticas | | | | | | | | |
| Maximo | 15 | 5 | 8 | -2 | 8 | 9 | 9 | 10 |
| Media | 10.8 | 2.4 | 6 | -5.6 | 5 | 4 | 7 | 6.8 |
| Desviacion Tipica | 2.48 | 1.5 | 1.79 | 2.06 | 1.67 | 2.83 | 1.41 | 2.3 |
| Hosteleria | 8 | 2 | 3 | -7 | 4 | 1 | 6 | 8 |
| Evaluaciones | | | | | | | | |
| Tutor | 15 | 1 | 8 | -2 | 8 | 9 | 9 | 3 |
| RRPP | 10 | 5 | 5 | -8 | 5 | 2 | 7 | 10 |
| Dependiente | 9 | 1 | 7 | -6 | 5 | 3 | 5 | 7 |
| Cuidador | 12 | 3 | 7 | -5 | 3 | 5 | 8 | 5 |

Ordenación de las alternativas:

```
res02$tabAcciones
```

| | Rango | Phi | Phi.mas | Phi.menos |
|-------------|-------|---------|---------|-----------|
| Tutor | 1 | 0.4013 | 0.5575 | 0.1562 |
| Cuidador | 2 | 0.0798 | 0.2772 | 0.1973 |
| RRPP | 3 | 0.0531 | 0.2734 | 0.2203 |
| Dependiente | 4 | -0.1700 | 0.1288 | 0.2988 |
| Hostelería | 5 | -0.3642 | 0.1279 | 0.4922 |

En esta tabla volvemos a observar los valores de Phi obtenidos para cada alternativa, confirmando que “Tutor” es la mejor opción.

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

```
## [1] "Tutor"      "Cuidador"    "RRPP"        "Dependiente" "Hostelería"
```

9 AHP CON R

9.1 Matriz de subcriterios.

En la siguiente matriz se muestran las preferencias entre los distintos criterios.

| Criterio | Salario | Beneficios | Cercania | Estrés | Desarrollo | R.estudios | Satisfaccion | R.personal |
|---------------------|---------|------------|----------|--------|------------|------------|--------------|------------|
| <i>Salario</i> | 1 | 3 | 2 | 2 | 2 | 5 | 4 | 3 |
| <i>Beneficios</i> | 1/3 | 1 | 1/3 | 1/2 | 2 | 2 | 2 | 1/2 |
| <i>Cercania</i> | 1/2 | 3 | 1 | 1/2 | 2 | 4 | 2 | 3 |
| <i>Estrés</i> | 1/2 | 2 | 2 | 1 | 3 | 4 | 2 | 2 |
| <i>Desarrollo</i> | 1/2 | 1/2 | 1/2 | 1/3 | 1 | 2 | 2 | 3 |
| <i>R.estudios</i> | 1/5 | 1/2 | 1/4 | 1/4 | 1/2 | 1 | 1/3 | 1/3 |
| <i>Satisfaccion</i> | 1/4 | 1/2 | 1/2 | 1/2 | 1/2 | 3 | 1 | 3 |
| <i>R.personal</i> | 1/3 | 2 | 1/3 | 1/2 | 1/3 | 3 | 1/3 | 1 |

De la tabla podemos sacar conclusiones generales. Podemos comprobar que los criterios salario y estrés son más importantes que los demás, debido a que la cantidad de dinero que ganaría el estudiante trabajando es una motivación fundamental para escoger un trabajo u otro, y el estrés es otro factor importante teniendo en cuenta que es el riesgo más directo para el rendimiento académico. Otro criterio que observamos muy importante es la cercanía, ya que facilitará mucho al alumno la coordinación entre el trabajo y los estudios. Tener un trabajo lejos de casa puede incrementar el estrés y hacernos perder tiempo de estudio, afectándonos negativamente.

Por otro lado, la relación con los estudios se sitúa consistentemente como el subcriterio de menor prioridad. Esto indica que, para el estudiante de este análisis, las necesidades a corto plazo (cubrir gastos y asegurar el rendimiento académico mediante la gestión del estrés y el tiempo) superan la importancia de que el empleo esté directamente relacionado con su currículum o carrera académica.

Los criterios de satisfacción personal y relación con los compañeros del trabajo se mantienen en un nivel de prioridad intermedio o bajo. Aunque son esenciales para el bienestar del estudiante, no se consideran prioritarios.

9.2 Matrices de Alternativas/Criterios.

9.2.1 Criterio SALARIO

| | Salario | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|---------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | | 1 | 1/3 | 1/2 | 3 | 2 |
| <i>Tutor</i> | | 3 | 1 | 2 | 5 | 4 |
| <i>RRPP</i> | | 2 | 1/2 | 1 | 4 | 3 |
| <i>Dependiente</i> | | 1/3 | 1/5 | 1/4 | 1 | 1/2 |
| <i>Cuidador</i> | | 1/2 | 1/4 | 1/3 | 2 | 1 |

9.2.2 Criterio BENEFICIOS

| Beneficios extras | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 3 | 1/3 | 2 | 3 |
| <i>Tutor</i> | 1/3 | 1 | 1/5 | 1/3 | 1/2 |
| <i>RRPP</i> | 3 | 5 | 1 | 4 | 5 |
| <i>Dependiente</i> | 1/2 | 3 | 1/4 | 1 | 2 |
| <i>Cuidador</i> | 1/3 | 2 | 1/5 | 1/2 | 1 |

9.2.3 Criterio CERCANÍA AL HOGAR

| Cercanía | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 1/3 | 2 | 1/2 | 1/3 |
| <i>Tutor</i> | 3 | 1 | 4 | 3 | 1 |
| <i>RRPP</i> | 1/2 | 1/4 | 1 | 1/3 | 1/5 |
| <i>Dependiente</i> | 2 | 1/3 | 3 | 1 | 1/2 |
| <i>Cuidador</i> | 3 | 1 | 5 | 2 | 1 |

9.2.4 Criterio CARGA MENTAL/ESTRÉS

| Estrés | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 1/5 | 4 | 1/3 | 1/2 |
| <i>Tutor</i> | 5 | 1 | 7 | 3 | 2 |
| <i>RRPP</i> | 1/4 | 1/7 | 1 | 1/5 | 1/6 |
| <i>Dependiente</i> | 3 | 1/3 | 5 | 1 | 1/2 |
| <i>Cuidador</i> | 2 | 1/2 | 6 | 2 | 1 |

9.2.5 Criterio OPORTUNIDADES DESARROLLO PERSONAL

| Desarrollo personal | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|---------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 1/3 | 1/2 | 2 | 1/3 |
| <i>Tutor</i> | 3 | 1 | 2 | 4 | 1 |
| <i>RRPP</i> | 2 | 1/2 | 1 | 3 | 1/2 |
| <i>Dependiente</i> | 1/2 | 1/4 | 1/3 | 1 | 1/4 |
| <i>Cuidador</i> | 3 | 1 | 2 | 4 | 1 |

9.2.6 Criterio RELACIÓN CON LOS ESTUDIOS

| R.Estudios | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 1/5 | 1/3 | 1 | 1/4 |
| <i>Tutor</i> | 5 | 1 | 4 | 5 | 6 |
| <i>RRPP</i> | 3 | 1/4 | 1 | 3 | 2 |
| <i>Dependiente</i> | 1 | 1/5 | 1/3 | 1 | 1/4 |
| <i>Cuidador</i> | 4 | 1/6 | 1/2 | 4 | 1 |

9.2.7 Criterio SATISFACCIÓN PERSONAL

| Satisfacción | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 1/3 | 2 | 1/2 | 1/4 |
| <i>Tutor</i> | 3 | 1 | 4 | 3 | 1/2 |
| <i>RRPP</i> | 1/2 | 1/4 | 1 | 1/3 | 1/5 |
| <i>Dependiente</i> | 2 | 1/3 | 3 | 1 | 1/2 |
| <i>Cuidador</i> | 4 | 2 | 5 | 2 | 1 |

9.2.8 Criterio RELACIÓN CON COMPAÑEROS Y JEFES

| Relación | Hostelería | Tutor | RRPP | Dependiente | Cuidador |
|--------------------|------------|-------|------|-------------|----------|
| <i>Hostelería</i> | 1 | 2 | 3 | 1/2 | 1 |
| <i>Tutor</i> | 1/2 | 1 | 3 | 1/2 | 1/2 |
| <i>RRPP</i> | 1/3 | 1/3 | 1 | 1/5 | 1/4 |
| <i>Dependiente</i> | 2 | 2 | 5 | 1 | 1 |
| <i>Cuidador</i> | 1 | 2 | 4 | 1 | 1 |

9.3 EMPLEANDO LAS FUNCIONES VISTAS EN CLASE CON R

9.3.1 MÉTODO 1:

- multicriterio.crea.matrizvaloraciones_mej() – multicriterio.metodoAHP.variante1.autovectormayorautovalor()
- multicriterio.metodoAHP.pesosglobales_entabla()

```
tabla_criterios = multicriterio.crea.matrizvaloraciones_mej(c(3, 2, 2, 2, 5, 4, 3, 1/3,
↪ 1/2, 2, 2, 2, 1/2, 1/2, 2, 4, 2, 3, 3, 4, 2, 2, 2, 2, 3, 1/3, 1/3, 3),
↪ numalternativas = 8, v.nombres.alternativas = c("Salario", "Beneficios", "Cercanía",
↪ "Estrés", "Desarrollo", "Relación_Estudios", "Satisfacción", "Relaciones"))
```

tabla_criterios

```
##          Salario Beneficios Cercanía Estrés Desarrollo
## Salario      1.0000000      3.0 2.0000000 2.0000000 2.0000000
## Beneficios   0.3333333      1.0 0.3333333 0.5000000 2.0000000
## Cercanía     0.5000000      3.0 1.0000000 0.5000000 2.0000000
## Estrés       0.5000000      2.0 2.0000000 1.0000000 3.0000000
## Desarrollo   0.5000000      0.5 0.5000000 0.3333333 1.0000000
## Relación_Estudios 0.2000000      0.5 0.2500000 0.2500000 0.5000000
## Satisfacción 0.2500000      0.5 0.5000000 0.5000000 0.5000000
## Relaciones   0.3333333      2.0 0.3333333 0.5000000 0.3333333
##          Relación_Estudios Satisfacción Relaciones
## Salario              5      4.0000000 3.0000000
## Beneficios           2      2.0000000 0.5000000
## Cercanía             4      2.0000000 3.0000000
## Estrés               4      2.0000000 2.0000000
## Desarrollo           2      2.0000000 3.0000000
## Relación_Estudios    1      0.3333333 0.3333333
## Satisfacción         3      1.0000000 3.0000000
## Relaciones           3      0.3333333 1.0000000
```

```
tabsalario = multicriterio.crea.matrizvaloraciones_mej(c(1/3, 1/2, 3, 2, 2, 5, 4, 4, 3,
↪ 1/2), numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
↪ "Dependiente", "Cuidador" ))
```

```
tabsalario
```

```
##          Hostelería      Tutor      RRPP Dependiente Cuidador
## Hostelería  1.0000000 0.3333333 0.5000000          3      2.0
## Tutor      3.0000000 1.0000000 2.0000000          5      4.0
## RRPP       2.0000000 0.5000000 1.0000000          4      3.0
## Dependiente 0.3333333 0.2000000 0.2500000          1      0.5
## Cuidador   0.5000000 0.2500000 0.3333333          2      1.0
```

```
tabbeneficios = multicriterio.crea.matrizvaloraciones_mej(c(3, 1/3, 2, 3, 1/5, 1/3, 1/2,
↪ 4, 5, 2), numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor",
↪ "RRPP", "Dependiente", "Cuidador" ))
```

```
tabbeneficios
```

```
##          Hostelería Tutor      RRPP Dependiente Cuidador
## Hostelería  1.0000000      3 0.3333333  2.0000000      3.0
## Tutor      0.3333333      1 0.2000000  0.3333333      0.5
## RRPP       3.0000000      5 1.0000000  4.0000000      5.0
## Dependiente 0.5000000      3 0.2500000  1.0000000      2.0
## Cuidador   0.3333333      2 0.2000000  0.5000000      1.0
```

```
tabcercania = multicriterio.crea.matrizvaloraciones_mej(c(1/3, 2, 1/2, 1/3, 4, 3, 1, 1/3,
↪ 1/5, 1/2), numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor",
↪ "RRPP", "Dependiente", "Cuidador" ))
```

```
tabcercania
```

```
##          Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.0 0.3333333      2  0.5000000 0.3333333
## Tutor          3.0 1.0000000      4  3.0000000 1.0000000
## RRPP           0.5 0.2500000      1  0.3333333 0.2000000
## Dependiente    2.0 0.3333333      3  1.0000000 0.5000000
## Cuidador       3.0 1.0000000      5  2.0000000 1.0000000
```

```
tabestres = multicriterio.crea.matrizvaloraciones_mej(c(1/5, 4, 1/3, 1/2, 7, 3, 2, 1/5,
↪ 1/6, 1/2), numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor",
↪ "RRPP", "Dependiente", "Cuidador" ))
```

```
tabestres
```

```
##          Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.00 0.2000000      4  0.3333333 0.5000000
## Tutor          5.00 1.0000000      7  3.0000000 2.0000000
## RRPP           0.25 0.1428571      1  0.2000000 0.1666667
## Dependiente    3.00 0.3333333      5  1.0000000 0.5000000
## Cuidador       2.00 0.5000000      6  2.0000000 1.0000000
```

```
tabdesarrollpersonal = multicriterio.crea.matrizvaloraciones_mej(c(1/3, 1/2, 2, 1/3, 2,
↪ 4, 1, 3, 1/2, 1/4), numalternativas = 5, v.nombres.alternativas = c("Hostelería",
↪ "Tutor", "RRPP", "Dependiente", "Cuidador" ))
```

```
tabdesarrollpersonal
```

```
##          Hostelería      Tutor      RRPP Dependiente Cuidador
```

```
## Hostelería      1.0 0.3333333 0.5000000      2 0.3333333
## Tutor          3.0 1.0000000 2.0000000      4 1.0000000
## RRPP           2.0 0.5000000 1.0000000      3 0.5000000
## Dependiente    0.5 0.2500000 0.3333333      1 0.2500000
## Cuidador       3.0 1.0000000 2.0000000      4 1.0000000
```

```
tabrelacionestudios = multicriterio.crea.matrizvaloraciones_mej(c(1/5, 1/3, 1, 1/4, 4, 5,
↪ 6, 3, 2, 1/4), numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor",
↪ "RRPP", "Dependiente", "Cuidador" ))
```

```
tabrelacionestudios
```

```
##           Hostelería      Tutor      RRPP Dependiente Cuidador
## Hostelería      1 0.2000000 0.3333333      1      0.25
## Tutor           5 1.0000000 4.0000000      5      6.00
## RRPP            3 0.2500000 1.0000000      3      2.00
## Dependiente     1 0.2000000 0.3333333      1      0.25
## Cuidador        4 0.1666667 0.5000000      4      1.00
```

```
tabsatisfaccion = multicriterio.crea.matrizvaloraciones_mej(c(1/3, 2, 1/2, 1/4, 4, 3,
↪ 1/2, 1/3, 1/5, 1/2), numalternativas = 5, v.nombres.alternativas = c("Hostelería",
↪ "Tutor", "RRPP", "Dependiente", "Cuidador" ))
```

```
tabsatisfaccion
```

```
##           Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.0 0.3333333      2 0.5000000      0.25
## Tutor           3.0 1.0000000      4 3.0000000      0.50
## RRPP            0.5 0.2500000      1 0.3333333      0.20
## Dependiente     2.0 0.3333333      3 1.0000000      0.50
## Cuidador        4.0 2.0000000      5 2.0000000      1.00
```

```
tabrelaciontrabajo = multicriterio.crea.matrizvaloraciones_mej(c(2, 3, 1/2, 1, 3, 1/2,
↪ 1/2, 1/5, 1/4, 1), numalternativas = 5, v.nombres.alternativas = c("Hostelería",
↪ "Tutor", "RRPP", "Dependiente", "Cuidador" ))
```

```
tabrelaciontrabajo
```

```
##           Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.0000000 2.0000000      3      0.5      1.00
## Tutor           0.5000000 1.0000000      3      0.5      0.50
## RRPP            0.3333333 0.3333333      1      0.2      0.25
## Dependiente     2.0000000 2.0000000      5      1.0      1.00
## Cuidador        1.0000000 2.0000000      4      1.0      1.00
```

PESOS LOCALES:

```
pesos_locales_criterios =
↪ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabla_criterios)
```

```
pesos_locales_criterios
```

```
## $Xmat
##           Salario Beneficios Cercanía      Estrés Desarrollo
## Salario      1.0000000      3.0 2.0000000 2.0000000 2.0000000
## Beneficios    0.3333333      1.0 0.3333333 0.5000000 2.0000000
## Cercanía      0.5000000      3.0 1.0000000 0.5000000 2.0000000
```

```

## Estrés          0.5000000      2.0 2.0000000 1.0000000 3.0000000
## Desarrollo      0.5000000      0.5 0.5000000 0.3333333 1.0000000
## Relación_Estudios 0.2000000      0.5 0.2500000 0.2500000 0.5000000
## Satisfacción    0.2500000      0.5 0.5000000 0.5000000 0.5000000
## Relaciones      0.3333333      2.0 0.3333333 0.5000000 0.3333333
## Relación_Estudios Satisfacción Relaciones
## Salario          5      4.0000000 3.0000000
## Beneficios        2      2.0000000 0.5000000
## Cercanía          4      2.0000000 3.0000000
## Estrés            4      2.0000000 2.0000000
## Desarrollo        2      2.0000000 3.0000000
## Relación_Estudios 1      0.3333333 0.3333333
## Satisfacción      3      1.0000000 3.0000000
## Relaciones        3      0.3333333 1.0000000
##
## $autovalor
## [1] 8.737603
##
## $suma.autovector
## [1] 2.507146
##
## $normaeuclidea.autovector
## [1] 1
##
## $valoraciones.ahp
##          Salario      Beneficios      Cercanía      Estrés
##      0.25254914      0.09389823      0.16328555      0.18480131
##      Desarrollo Relación_Estudios Satisfacción Relaciones
##      0.10313929      0.03760313      0.08779230      0.07693105
##
## $valoraciones.ahp.ordenadas
##          Salario      Estrés      Cercanía      Desarrollo
##      0.25254914      0.18480131      0.16328555      0.10313929
##      Beneficios Satisfacción Relaciones Relación_Estudios
##      0.09389823      0.08779230      0.07693105      0.03760313
##
## $CI.coef.inconsistencia
## [1] 0.1053719
##
## $RI.coef.inconsistencia
## [1] 0.07473182
##
## $consistencia
## [1] "Consistencia aceptable"
##
## $tablaresumen
##          Salario Beneficios Cercanía Estrés Desarrollo
## Salario      1.0000000      3.0 2.0000000 2.0000000 2.0000000
## Beneficios    0.3333333      1.0 0.3333333 0.5000000 2.0000000
## Cercanía      0.5000000      3.0 1.0000000 0.5000000 2.0000000
## Estrés        0.5000000      2.0 2.0000000 1.0000000 3.0000000
## Desarrollo    0.5000000      0.5 0.5000000 0.3333333 1.0000000
## Relación_Estudios 0.2000000      0.5 0.2500000 0.2500000 0.5000000
## Satisfacción  0.2500000      0.5 0.5000000 0.5000000 0.5000000
## Relaciones    0.3333333      2.0 0.3333333 0.5000000 0.3333333

```

```
##          NA          NA          NA          NA          NA
## Relación_Estudios Satisfacción Relaciones
## Salario          5      4.0000000  3.0000000 8.737603
## Beneficios       2      2.0000000  0.5000000      NA
## Cercanía        4      2.0000000  3.0000000      NA
## Estrés          4      2.0000000  2.0000000      NA
## Desarrollo       2      2.0000000  3.0000000      NA
## Relación_Estudios 1      0.3333333  0.3333333      NA
## Satisfacción      3      1.0000000  3.0000000      NA
## Relaciones       3      0.3333333  1.0000000      NA
##          NA          NA          NA          NA
##          autovector.v prioridades.relativas
## Salario          0.63317755          0.25254914
## Beneficios       0.23541656          0.09389823
## Cercanía        0.40938070          0.16328555
## Estrés          0.46332387          0.18480131
## Desarrollo       0.25858526          0.10313929
## Relación_Estudios 0.09427653          0.03760313
## Satisfacción     0.22010812          0.08779230
## Relaciones       0.19287737          0.07693105
##          2.50714596          NA
```

```
pesos_locales_salario =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabsalario)

pesos_locales_salario$valoraciones.ahp
```

```
## Hostelería      Tutor      RRPP Dependiente      Cuidador
## 0.15992286 0.41853929 0.26251761 0.06176665 0.09725359
```

```
pesos_locales_beneficios =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabbeneficios)

pesos_locales_beneficios$valoraciones.ahp
```

```
## Hostelería      Tutor      RRPP Dependiente      Cuidador
## 0.21943787 0.06344222 0.48322220 0.14474391 0.08915380
```

```
pesos_locales_cercania =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabcercania)

pesos_locales_cercania$valoraciones.ahp
```

```
## Hostelería      Tutor      RRPP Dependiente      Cuidador
## 0.10600378 0.34220238 0.06347638 0.16641503 0.32190242
```

```
pesos_locales_estres =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabestres)

pesos_locales_estres$valoraciones.ahp
```

```
## Hostelería      Tutor      RRPP Dependiente      Cuidador
## 0.10155531 0.43319686 0.03918953 0.18277994 0.24327836
```

```
pesos_locales_desarrollo =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabdesarrollopersonal)
```

```
pesos_locales_desarrollo$valoraciones.ahp
```

```
## Hostelería Tutor RRPP Dependiente Cuidador
## 0.10934412 0.31918548 0.18400552 0.06827939 0.31918548
```

```
pesos_locales_relacionestudios =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabrelacionestudios)
```

```
pesos_locales_relacionestudios$valoraciones.ahp
```

```
## Hostelería Tutor RRPP Dependiente Cuidador
## 0.06211943 0.53140343 0.18810217 0.06211943 0.15625554
```

```
pesos_locales_satisfaccion =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabsatisfaccion)
```

```
pesos_locales_satisfaccion$valoraciones.ahp
```

```
## Hostelería Tutor RRPP Dependiente Cuidador
## 0.09629377 0.29338908 0.06119857 0.16160575 0.38751283
```

```
pesos_locales_relaciontrabajo =
↳ multicriterio.metodoAHP.variante1.autovectormayorautovalor(tabrelaciontrabajo)
```

```
pesos_locales_relaciontrabajo$valoraciones.ahp
```

```
## Hostelería Tutor RRPP Dependiente Cuidador
## 0.21818284 0.14276556 0.06113138 0.31649106 0.26142916
```

PESOS GLOBALES:

```
pesos_globales =
↳ multicriterio.metodoAHP.pesosglobales_entabla(pesos_locales_criterios$valoraciones.ahp,
↳ rbind(pesos_locales_salario$valoraciones.ahp, pesos_locales_beneficios$valoraciones.ahp,
↳ pesos_locales_cercania$valoraciones.ahp, pesos_locales_estres$valoraciones.ahp,
↳ pesos_locales_desarrollo$valoraciones.ahp,
↳ pesos_locales_relacionestudios$valoraciones.ahp,
↳ pesos_locales_satisfaccion$valoraciones.ahp,
↳ pesos_locales_relaciontrabajo$valoraciones.ahp))
```

```
pesos_globales
```

```
## Salario Beneficios Cercanía Estrés Desarrollo
## Hostelería 0.15992286 0.21943787 0.10600378 0.10155531 0.10934412
## Tutor 0.41853929 0.06344222 0.34220238 0.43319686 0.31918548
## RRPP 0.26251761 0.48322220 0.06347638 0.03918953 0.18400552
## Dependiente 0.06176665 0.14474391 0.16641503 0.18277994 0.06827939
## Cuidador 0.09725359 0.08915380 0.32190242 0.24327836 0.31918548
## Ponder.Criterios 0.25254914 0.09389823 0.16328555 0.18480131 0.10313929
## Relación_Estudios Satisfacción Relaciones
## Hostelería 0.06211943 0.09629377 0.21818284
## Tutor 0.53140343 0.29338908 0.14276556
## RRPP 0.18810217 0.06119857 0.06113138
## Dependiente 0.06211943 0.16160575 0.31649106
## Cuidador 0.15625554 0.38751283 0.26142916
## Ponder.Criterios 0.03760313 0.08779230 0.07693105
## Ponderadores Globales
```

| | |
|---------------------|-----------|
| ## Hostelería | 0.1359221 |
| ## Tutor | 0.3372343 |
| ## RRPP | 0.1654065 |
| ## Dependiente | 0.1380554 |
| ## Cuidador | 0.2233818 |
| ## Ponder.Criterios | NA |

Observando los pesos globales, podemos observar que la mejor alternativa es la de profesor particular o tutor, con un peso global del 33.7%. Después de esta, viene cuidador de niños con un peso de 22.34%, y RRPP con un 16.54%. Por último, las peores alternativas serían dependiente y hostelería, con unos pesos relativamente similares, 13.81% y 13.59%. Se vuelve a repetir que la mejor alternativa es la de Tutor, igual que hemos comprobado con los demás métodos.

9.3.2 MÉTODO 2:

```
matriz.criterios=multicriterio.crea.matrizvaloraciones(
  c(1, 3, 2, 2, 2, 5, 4, 3, 1/3, 1, 1/3, 1/2, 2, 2, 2, 1/2, 1/2, 3, 1, 1/2, 2, 4, 2, 3,
    ↪ 1/2, 2, 2, 1, 3, 4, 2, 2, 1/2, 1/2, 1/2, 1/3, 1, 2, 2, 3, 1/5, 1/2, 1/4, 1/4,
    ↪ 1/2, 1, 1/3, 1/3, 1/4, 1/2, 1/2, 1/2, 1/2, 3, 1, 3, 1/3, 2, 1/3, 1/2, 1/3, 3,
    ↪ 1/3, 1),
  numalternativas = 8, v.nombres.alternativas = c("Salario", "Beneficios", "Cercanía",
    ↪ "Estrés", "Desarrollo", "Relación_Estudios", "Satisfacción", "Relaciones"))

matriz.salario=multicriterio.crea.matrizvaloraciones(
  c(1, 1/3, 1/2, 3, 2, 3, 1, 2, 5, 4, 2, 1/2, 1, 4, 3, 1/3, 1/5, 1/4, 1, 1/2, 1/2, 1/4,
    ↪ 1/3, 2, 1),
  numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
    ↪ "Dependiente", "Cuidador" ))

matriz.beneficios=multicriterio.crea.matrizvaloraciones(
  c(1, 3, 1/3, 2, 3, 1/3, 1, 1/5, 1/3, 1/2, 3, 5, 1, 4, 5, 1/2, 3, 1/4, 1, 2, 1/3, 2,
    ↪ 1/5, 1/2, 1),
  numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
    ↪ "Dependiente", "Cuidador" ))

matriz.cercania=multicriterio.crea.matrizvaloraciones(
  c(1, 1/3, 2, 1/2, 1/3, 3, 1, 4, 3, 1, 1/2, 1/4, 1, 1/3, 1/5, 2, 1/3, 3, 1, 1/2, 3, 1,
    ↪ 5, 2, 1),
  numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
    ↪ "Dependiente", "Cuidador" ))

matriz.estres=multicriterio.crea.matrizvaloraciones(
  c(1, 1/5, 4, 1/3, 1/2, 5, 1, 7, 3, 2, 1/4, 1/7, 1, 1/5, 1/6, 3, 1/3, 5, 1, 1/2, 2,
    ↪ 1/2, 6, 2, 1),
  numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
    ↪ "Dependiente", "Cuidador" ))

matriz.desarrollpersonal=multicriterio.crea.matrizvaloraciones(
  c(1, 1/3, 1/2, 2, 1/3, 3, 1, 2, 4, 1, 2, 1/2, 1, 3, 1/2, 1/2, 1/4, 1/3, 1, 1/4, 3, 1,
    ↪ 2, 4, 1),
  numalternativas = 5, v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
    ↪ "Dependiente", "Cuidador" ))

matriz.relacionestudios=multicriterio.crea.matrizvaloraciones(
  c(1, 1/5, 1/3, 1, 1/4, 5, 1, 4, 5, 6, 3, 1/4, 1, 3, 2, 1, 1/5, 1/3, 1, 1/4, 4, 1/6,
    ↪ 1/2, 4, 1),
```

```

numalternativas = 5,v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
↪ "Dependiente", "Cuidador" ))

matriz.satisfaccion=multicriterio.crea.matrizvaloraciones(
  c(1, 1/3, 2, 1/2, 1/4, 3, 1, 4, 3, 1/2, 1/2, 1/4, 1, 1/3, 1/5, 2, 1/3, 3, 1, 1/2, 4,
↪ 2, 5, 2, 1),
  numalternativas = 5,v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
↪ "Dependiente", "Cuidador" ))

matriz.relaciontrabajo=multicriterio.crea.matrizvaloraciones(
  c(1, 2, 3, 1/2, 1, 1/2, 1, 3, 1/2, 1/2, 1/3, 1/3, 1, 1/5, 1/4, 2, 2, 5, 1, 1, 1, 2,
↪ 4, 1, 1),
  numalternativas = 5,v.nombres.alternativas = c("Hostelería", "Tutor", "RRPP",
↪ "Dependiente", "Cuidador" ))

num.alt=5
num.cri=8
matriznivel2=array(NA,dim=c(num.alt,num.alt,num.cri))

matriznivel2[,1] = matriz.salario
matriznivel2[,2] = matriz.beneficios
matriznivel2[,3] = matriz.cercania
matriznivel2[,4] = matriz.estres
matriznivel2[,5] = matriz.desarrollopersonal
matriznivel2[,6] = matriz.relacionestudios
matriznivel2[,7] = matriz.satisfaccion
matriznivel2[,8] = matriz.relaciontrabajo

solucion_metodo2 =
↪ multicriterio.metodoAHP.variante3.completo(matriz.criterios,matriznivel2)

solucion_metodo2$pesos.globales

```

```
## [1] 0.1375381 0.3342639 0.1651137 0.1391173 0.2239670
```

Obtenemos los pesos globales, y podemos volver a observar que la mejor alternativa es la de profesor particular o tutor, con un peso global de 33.43%. Después de esta, viene cuidador de niños con un peso de 22.4%, y RRPP con un 16.51%. Por último, las peores alternativas serían dependiente y hostelería, con unos pesos relativamente similares, 13.91% y 13.76%. Se vuelve a repetir que la mejor alternativa es la de Tutor, igual que hemos comprobado con los demás métodos.

9.4 EMPLEANDO EL PAQUETE AHP

9.4.1 MÉTODO 1:

– multicriterio.crea.matrizvaloraciones_mej() – multicriterio.metodoahp.diagrama()

En forma de diagrama:

```

source("teoriadecision_funciones_multicriterio_diagram.R")

matriznivel2_metodo3 = array(NA,dim=c(5,5,8))
# 5 filas
# 5 columnas
# 8 matrices 5x5

```



```
matriznivel2_metodo3[,1] = matriz.salario
matriznivel2_metodo3[,2] = matriz.beneficios
matriznivel2_metodo3[,3] = matriz.cercania
matriznivel2_metodo3[,4] = matriz.estres
matriznivel2_metodo3[,5] = matriz.desarrollpersonal
matriznivel2_metodo3[,6] = matriz.relacionestudios
matriznivel2_metodo3[,7] = matriz.satisfaccion
matriznivel2_metodo3[,8] = matriz.relaciontrabajo
```

```
row.names(matriznivel2_metodo3) = c("Hostelería", "Tutor", "RRPP", "Dependiente",
↪ "Cuidador")
```

```
colnames(matriznivel2_metodo3) = c("Hostelería", "Tutor", "RRPP", "Dependiente",
↪ "Cuidador")
```

```
matriznivel2_metodo3
```

```
## , , 1
##
##      Hostelería      Tutor      RRPP Dependiente Cuidador
## Hostelería  1.0000000 0.3333333 0.5000000          3      2.0
## Tutor      3.0000000 1.0000000 2.0000000          5      4.0
## RRPP       2.0000000 0.5000000 1.0000000          4      3.0
## Dependiente 0.3333333 0.2000000 0.2500000          1      0.5
## Cuidador   0.5000000 0.2500000 0.3333333          2      1.0
##
```

```
## , , 2
##
##      Hostelería Tutor      RRPP Dependiente Cuidador
## Hostelería  1.0000000      3 0.3333333  2.0000000      3.0
## Tutor      0.3333333      1 0.2000000  0.3333333      0.5
## RRPP       3.0000000      5 1.0000000  4.0000000      5.0
## Dependiente 0.5000000      3 0.2500000  1.0000000      2.0
## Cuidador   0.3333333      2 0.2000000  0.5000000      1.0
##
```

```
## , , 3
##
##      Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.0 0.3333333      2 0.5000000 0.3333333
## Tutor          3.0 1.0000000      4 3.0000000 1.0000000
## RRPP           0.5 0.2500000      1 0.3333333 0.2000000
## Dependiente    2.0 0.3333333      3 1.0000000 0.5000000
## Cuidador       3.0 1.0000000      5 2.0000000 1.0000000
##
```

```
## , , 4
##
##      Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.00 0.2000000      4 0.3333333 0.5000000
## Tutor          5.00 1.0000000      7 3.0000000 2.0000000
## RRPP           0.25 0.1428571      1 0.2000000 0.1666667
## Dependiente    3.00 0.3333333      5 1.0000000 0.5000000
## Cuidador       2.00 0.5000000      6 2.0000000 1.0000000
##
```

```
## , , 5
```

```

##
##      Hostelería      Tutor      RRPP Dependiente Cuidador
## Hostelería      1.0 0.3333333 0.5000000      2 0.3333333
## Tutor          3.0 1.0000000 2.0000000      4 1.0000000
## RRPP           2.0 0.5000000 1.0000000      3 0.5000000
## Dependiente    0.5 0.2500000 0.3333333      1 0.2500000
## Cuidador       3.0 1.0000000 2.0000000      4 1.0000000
##
## , , 6
##
##      Hostelería      Tutor      RRPP Dependiente Cuidador
## Hostelería      1 0.2000000 0.3333333      1 0.25
## Tutor          5 1.0000000 4.0000000      5 6.00
## RRPP           3 0.2500000 1.0000000      3 2.00
## Dependiente    1 0.2000000 0.3333333      1 0.25
## Cuidador       4 0.1666667 0.5000000      4 1.00
##
## , , 7
##
##      Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.0 0.3333333 2 0.5000000 0.25
## Tutor          3.0 1.0000000 4 3.0000000 0.50
## RRPP           0.5 0.2500000 1 0.3333333 0.20
## Dependiente    2.0 0.3333333 3 1.0000000 0.50
## Cuidador       4.0 2.0000000 5 2.0000000 1.00
##
## , , 8
##
##      Hostelería      Tutor RRPP Dependiente Cuidador
## Hostelería      1.0000000 2.0000000 3 0.5 1.00
## Tutor          0.5000000 1.0000000 3 0.5 0.50
## RRPP           0.3333333 0.3333333 1 0.2 0.25
## Dependiente    2.0000000 2.0000000 5 1.0 1.00
## Cuidador       1.0000000 2.0000000 4 1.0 1.00

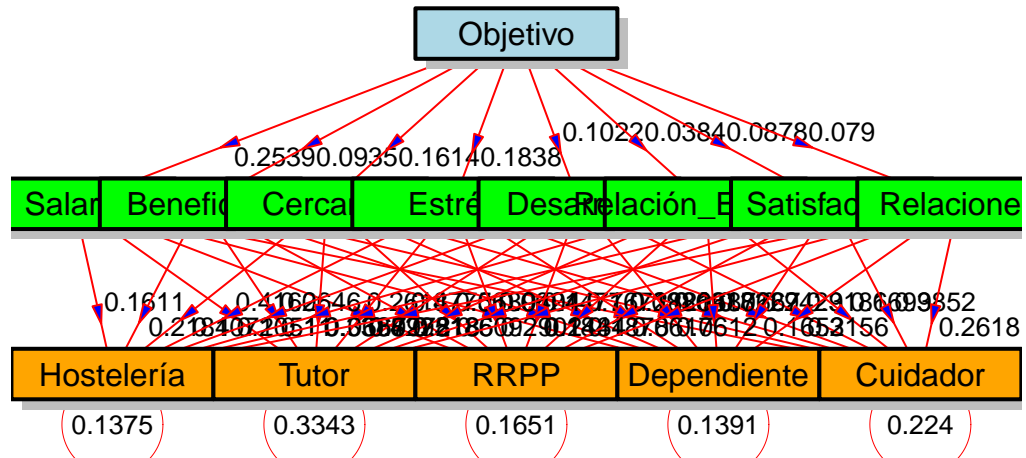
```

```

multicriterio.metodoahp.diagrama(matriz.criterios,matriznivel2_metodo3)

```

Estructura Jerárquica (AHP)



En el diagrama podemos observar que la mejor alternativa es la de profesor particular o tutor, con un peso global de 33.43%. Después de esta, viene cuidador de niños con un peso de 22.4%, y RRPP con un 16.51%. Por último, las peores alternativas serían dependiente y hostelería, con unos pesos relativamente similares, 13.91% y 13.75%. Se vuelve a repetir que la mejor alternativa es la de Tutor, igual que hemos comprobado con los demás métodos.

10 MÉTODO 2: USO DEL PAQUETE AHP

```
library(ahp)
#ahp::RunGUI()

datos = Load("ahptrabajo.ahp")
Calculate(datos)

Visualize(datos)
```

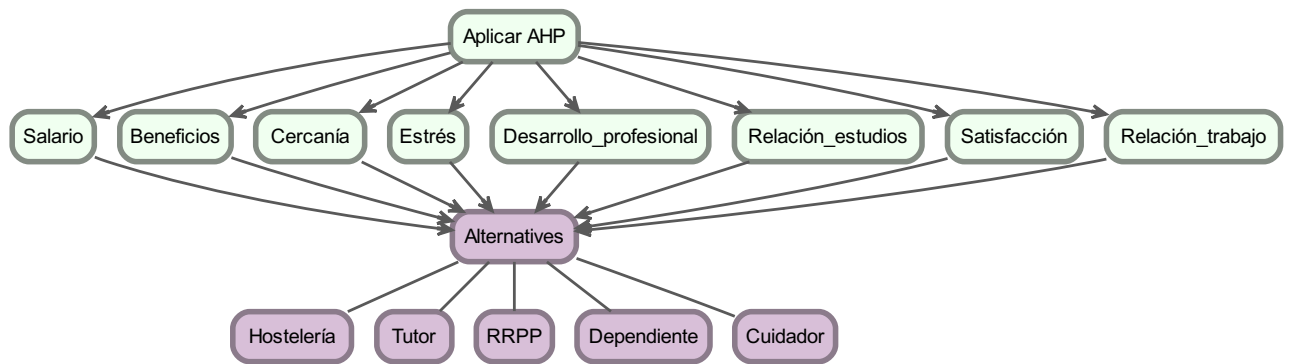


TABLA SOLUCIÓN:

```

# export_formattable(AnalyzeTable(datos), file = "tablatrabajo.png")
knitr::include_graphics("tablatrabajo.png")

```

| | Weight | Tutor | Cuidador | RRPP | Dependiente | Hostelería | Inconsistency |
|------------------------|--------|-------|----------|-------|-------------|------------|---------------|
| Aplicar AHP | 100.0% | 33.7% | 22.3% | 16.5% | 13.8% | 13.8% | 7.5% |
| Salario | 25.3% | 10.6% | 2.5% | 6.6% | 1.6% | 4.0% | 1.5% |
| Estrés | 18.5% | 8.0% | 4.5% | 0.7% | 3.4% | 1.9% | 4.4% |
| Cercanía | 16.3% | 5.6% | 5.2% | 1.0% | 2.7% | 1.9% | 2.7% |
| Desarrollo_profesional | 10.3% | 3.3% | 3.3% | 1.9% | 0.7% | 1.1% | 0.8% |
| Beneficios | 9.4% | 0.6% | 0.8% | 4.5% | 1.4% | 2.1% | 3.1% |
| Satisfacción | 8.8% | 2.6% | 3.4% | 0.5% | 1.4% | 0.8% | 3.0% |
| Relación_trabajo | 7.7% | 1.1% | 2.0% | 0.5% | 2.4% | 1.7% | 1.6% |
| Relación_estudios | 3.8% | 2.0% | 0.6% | 0.7% | 0.2% | 0.2% | 7.9% |

En la foto que hemos exportado vemos que la mejor alternativa es la de profesor particular/tutor, como en los demás métodos, con un peso global de 33.7%. Después le sigue cuidador, con un 22.3%, y RRPP, con un 16.5%. Por último, las peores alternativas son dependiente y hostelería, ambas con un peso global de 13.8%.

TABLA SOLUCIÓN DE PESOS LOCALES:

```
t2 = AnalyzeTable(datos, variable = "priority")
# export_formattable(t2, file = "tablabtrabajo.png")
knitr::include_graphics("tablabtrabajo.png")
```

| | Priority | Tutor | Cuidador | RRPP | Dependiente | Hostelería | Inconsistency |
|------------------------|----------|-------|----------|-------|-------------|------------|---------------|
| Aplicar AHP | 100.0% | | | | | | 7.5% |
| Salario | 25.3% | 41.9% | 9.7% | 26.3% | 6.2% | 16.0% | 1.5% |
| Estrés | 18.5% | 43.3% | 24.3% | 3.9% | 18.3% | 10.2% | 4.4% |
| Cercanía | 16.3% | 34.0% | 31.9% | 5.9% | 16.6% | 11.6% | 2.7% |
| Desarrollo_profesional | 10.3% | 31.9% | 31.9% | 18.4% | 6.8% | 10.9% | 0.8% |
| Beneficios | 9.4% | 6.3% | 8.9% | 48.3% | 14.5% | 21.9% | 3.1% |
| Satisfacción | 8.8% | 29.3% | 38.8% | 6.1% | 16.2% | 9.6% | 3.0% |
| Relación_trabajo | 7.7% | 14.3% | 26.1% | 6.1% | 31.6% | 21.8% | 1.6% |
| Relación_estudios | 3.8% | 53.1% | 15.6% | 18.8% | 6.2% | 6.2% | 7.9% |

En esta nueva tabla podemos corroborar la importancia de cada criterio para encontrar la mejor alternativa. El salario ha sido el factor más influyente, con un 25.3%, seguidos del estrés y la cercanía al hogar, con 18.5% y 16.3% respectivamente. El desarrollo profesional ocupa un 10.3% de la prioridad. Los siguientes factores importantes han sido los beneficios extras que puedes conseguir en el trabajo, con un 9.4%, la satisfacción personal con un 8.8%, y la relación dentro del trabajo con nuestros compañeros y jefes. El factor que menos importancia ha tenido a la hora de elegir la opción ha sido la relación del trabajo con los estudios que estamos realizando con un 3.8%, ya que en nuestro caso buscamos un trabajo para compaginar con los estudios y conseguir dinero fácil, no necesariamente de lo que hayamos estudiado.

Vamos a estudiar la inconsistencia de las matrices de datos que tenemos. El análisis de inconsistencia es la forma de validar subjetivamente la entrada de nuestro modelo AHP. Si todas las matrices son consistentes, el modelo es sólido y se puede confiar en las prioridades y conclusiones obtenidas.

```
pesos_locales_criterios$consistencia
```

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

```
pesos_locales_salario$consistencia
```

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

```

pesos_locales_beneficios$consistencia

## [1] "Consistencia aceptable"
pesos_locales_cercania$consistencia

## [1] "Consistencia aceptable"
pesos_locales_estres$consistencia

## [1] "Consistencia aceptable"
pesos_locales_relacionestudios$consistencia

## [1] "Consistencia aceptable"
pesos_locales_desarrollo$consistencia

## [1] "Consistencia aceptable"
pesos_locales_satisfaccion$consistencia

## [1] "Consistencia aceptable"
pesos_locales_relaciontrabajo$consistencia

## [1] "Consistencia aceptable"

```

El análisis de consistencia, llevado a cabo mediante el cálculo de la Razón de Consistencia (RC) para cada matriz, ha demostrado que todos los juicios de valor son coherentes y válidos. En la matriz de criterios, la matriz de comparación por pares de los 8 criterios nos da una razón de Consistencia (RC) inferior al 10%, lo que indica que la jerarquización de la importancia de los factores es altamente fiable. Respecto a las matrices de alternativas, las ocho matrices de comparación por pares para las alternativas en función de cada criterio también resultaron tener una RC inferior al 10%.

Dado que todas las matrices del modelo AHP presentan una consistencia aceptable, se valida la calidad de los juicios de valor emitidos. Esto asegura que no hay contradicciones significativas en las preferencias establecidas, lo cual confiere una sólida fiabilidad a los pesos locales y, consecuentemente, a los pesos globales que determinan que la alternativa de Tutor es la mejor opción.

Si hubiese habido inconsistencia, el resultado de que “Tutor” es la mejor opción habría tenido que ser revisado. Al ser consistente, puedes afirmar que el resultado (Tutor con el peso global más alto) refleja fielmente las preferencias del decisor.

11 CONCLUSIONES GENERALES

Hemos utilizado los métodos de Electre, Promethee y AHP para decidir sobre qué trabajo es mejor para un estudiante, teniendo en cuenta diferentes criterios que afectan a su vida. Todos los criterios coinciden en que la mejor opción de las 5 que estamos evaluando es ser profesor particular o tutor de niños, dar clases particulares, ya que su horario es más flexible y destaca favorablemente en prácticamente todos los criterios que hemos tenido en cuenta.

El peso que hemos usado en los métodos Electre y Promethee son diferentes a los pesos globales que nos ha dado el método AHP. Los pesos que he usado inicialmente fueron: (0.25, 0.10 , 0.15, 0.15, 0.05, 0.05, 0.15 , 0.10), y los pesos globales que nos ha dado AHP son: (25.3 , 9.4 , 16.3 , 18.5 , 10.3 , 3.8 , 8.8 , 7.7). Podemos observar que los pesos son ligeramente similares a los que supuse al principio, pero el método AHP le ha dado menos valor a la satisfacción y la relación con los compañeros y jefes del trabajo. También le ha dado más importancia sobre todo al desarrollo personal con el trabajo que escogamos.