trabajo02

Magdalena Ceballos Rodríguez

2024-10-31

Contents

1 Introducción

La toma de decisiones en escenarios complejos donde intervienen múltiples criterios es un desafío común en muchas áreas, desde la gestión de proyectos hasta la selección de inversiones y la planificación estratégica. El análisis de decisión multicriterio (MCDM, por sus siglas en inglés) proporciona un marco robusto para abordar estos problemas, permitiendo evaluar y priorizar alternativas basadas en una variedad de criterios cualitativos y cuantitativos.

En este trabajo, se aborda un problema de decisión multicriterio aplicando varias técnicas vistas en clase: el Proceso de Análisis Jerárquico (AHP), el método Electre y el método Promethee, entre otros. Cada uno de estos métodos se emplea para estructurar, analizar y resolver el problema elegido, destacando sus enfoques, ventajas y limitaciones.

El objetivo de este estudio es comparar cómo cada técnica proporciona insights y resultados en el proceso de decisión, y discutir la consistencia y viabilidad de las soluciones obtenidas. Además, se documentará la metodología y los pasos seguidos, complementando el análisis con gráficos y material visual que facilite la comprensión del proceso y los resultados.

Finalmente, se presentarán las conclusiones y observaciones obtenidas a partir de la aplicación de estas técnicas, resaltando las implicaciones prácticas y las posibles recomendaciones para futuros análisis de decisiones.

2 Planteamiento del Problema:

Elección de Plataforma de Streaming

En la actualidad, las plataformas de streaming son una de las principales formas de consumir contenido multimedia. Sin embargo, la gran cantidad de opciones disponibles genera un desafío para los usuarios a la hora de elegir cuál es la plataforma que mejor satisface sus necesidades. En este caso, vamos a analizar cuál es la mejor plataforma de streaming, entre tres opciones populares: **Netflix**, **Prime** y **Disney+**, según varios criterios de decisión.

Criterios de Decisión:

Para resolver el problema, se han definido los siguientes criterios de decisión, basados en las características que los usuarios suelen considerar al elegir una plataforma de streaming:

1. **Precio**: El costo mensual de la suscripción a la plataforma.
2. **Variedad de Contenido**: La cantidad y diversidad de contenido disponible en la plataforma (películas, series, documentales, etc.).
3. **Calidad de Video**: La calidad de resolución de video disponible, como 4K, HDR, etc.
4. **Facilidad de Uso**: La interfaz de usuario y facilidad de navegación en la plataforma.
5. **Compatibilidad de Dispositivos**: La disponibilidad de la plataforma en diferentes dispositivos como Smart TVs, computadoras, teléfonos móviles, consolas de videojuegos, etc.
6. **Recomendaciones Personalizadas**: La calidad y precisión del sistema de recomendaciones de contenido que ofrece la plataforma.
7. **Exclusivos y Originales**: La presencia de contenido exclusivo o de producción propia que solo esté disponible en esa plataforma.

Alternativas:

Las alternativas a evaluar son las siguientes plataformas de streaming:

1. **Netflix**
2. **Prime**
3. **Disney+**

Valores Numéricos de los Criterios

Criterio	Precio	Variedad	Calidad	Facilidad	Compatibilidad	Recomendaciones	Originales
Precio (USD/mes)	1	1/3	1/4	1/4	1/5	1/3	1/5
Variedad	3	1	2	2	1	2	1
Calidad	4	1/2	1	2	2	3	2
Facilidad	4	1/2	1/2	1	1	2	1
Compatibilidad	5	1	1/2	1	1	2	1/2
Recomendaciones	3	1/2	1/3	1/2	1/2	1	1/2
Originales	5	1	1/2	1	2	2	1

Para simplificar, utilizaremos una escala de 1 a 5, donde 1 es muy malo y 5 es excelente.

Criterio	Netflix	Prime	Disney+
Precio (USD/mes)	3	5	4
Variedad de Contenido	5	4	4
Calidad de Video	5	3	5
Facilidad de Uso	5	4	5
Compatibilidad de Dispositivos	5	4	5
Recomendaciones Personalizadas	5	3	4
Exclusivos y Originales	5	3	4

Comparaciones por pares

1. Precio

Criterio	Netflix	Prime	Disney+
Netflix	1	3/2	5/2
Prime	2/3	1	3/2
Disney+	2/5	2/3	1

2. Variedad de Contenido

Criterio	Netflix	Prime	Disney+
Netflix	1	3/2	3
Prime	2/3	1	2

Criterio	Netflix	Prime	Disney+
Disney+	1/3	1/2	1

3. Calidad de Video

Criterio	Netflix	Prime	Disney+
Netflix	1	3/2	5/2
Prime	2/3	1	2
Disney+	2/5	1/2	1

4. Facilidad de Uso

Criterio	Netflix	Prime	Disney+
Netflix	1	3/2	5/2
Prime	2/3	1	3/2
Disney+	2/5	2/3	1

5. Compatibilidad de Dispositivos

Criterio	Netflix	Prime	Disney+
Netflix	1	3/2	3
Prime	2/3	1	2
Disney+	1/3	1/2	1

6. Recomendaciones Personalizadas

Criterio	Netflix	Prime	Disney+
Netflix	1	3/2	5/2
Prime	2/3	1	3/2
Disney+	2/5	2/3	1

7. Originales y Exclusivos

Criterio	Netflix	Prime	Disney+
Netflix	1	2	3
Prime	1/2	1	3/2
Disney+	1/3	2/3	1

3 Resolución con ayuda de las funciones R definidas en clase

```
## Warning: package 'diagram' was built under R version 4.3.2
```

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

```
## Warning: package 'shape' was built under R version 4.3.2
```

Con ayuda de las funciones R de clase

Se usan las funciones R en "teoriadecision_funciones_multicriterio.R":

- Método 1:

– multicriterio.crea.matrizvaloraciones_mej()

– multicriterio.metodoAHP.variante1.autovectormayorautovalor()

– multicriterio.metodoAHP.pesosglobales_entabla()

- Método 2:

– multicriterio.crea.matrizvaloraciones_mej()

– multicriterio.metodoAHP.variante3.completo()

3.1 Método 1

```
# Vector de la diagonal superior
vector_valores = c(
  1/3, 1/4, 1/4, 1/5, 1/3, 1/5,  # Comparaciones de Precio
  2,   2,   1,   2,   1,         # Comparaciones de Variedad
  2,   2,   3,   2,              # Comparaciones de Calidad
  1,   2,   1,                   # Comparaciones de Facilidad
  2,   1/2,                      # Comparaciones de Compatibilidad
  1/2                            # Comparaciones de Recomendaciones
)


matriz_criterios = multicriterio.crea.matrizvaloraciones_mej(
  vector_valores,
  numalternativas = 7,
  v.nombres.alternativas = c("Precio", "Variedad", "Calidad", "Facilidad",
                             "Compatibilidad", "Recomendaciones", "Originales")
)

matriz_precio=multicriterio.crea.matrizvaloraciones_mej(c(3/2,5/2,3/2),
    numalternativas = 3,v.nombres.alternativas =c("Netflix","Prime","Disney+"))


matriz_variedad=multicriterio.crea.matrizvaloraciones_mej(c(3/2,3,2),
                numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


matriz_calidad=multicriterio.crea.matrizvaloraciones_mej(c(3/2,5/2,2),
```

```
              numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


matriz_facilidad=multicriterio.crea.matrizvaloraciones_mej(c(3/2,5/2,3/2),
                  numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


matriz_compatibilidad=multicriterio.crea.matrizvaloraciones_mej(c(3/2,3,2),
            numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


matriz_recomendaciones=multicriterio.crea.matrizvaloraciones_mej(c(3/2,5/2,3/2),
                       numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


matriz_originales=multicriterio.crea.matrizvaloraciones_mej(c(2,3,3/2),
           numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))
```

Calculamos los pesos locales:

```
pesos_mcriterios = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_criterios)

pesos_mprecio = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_precio)

pesos_mvariedad = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_variedad)

pesos_mcalidad = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_calidad)

pesos_mfacilidad = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_facilidad)

pesos_mcompatibilidad = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_compatibilidad

pesos_mrecomendacionnes = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_recomendaci

pesos_moriginales = multicriterio.metodoAHP.variante1.autovectormayorautovalor(matriz_originales)
```

Calculamos ahora los pesos globales:

```
pesos=multicriterio.metodoAHP.pesosglobales_entabla(
  pesos_mcriterios$valoraciones.ahp,rbind(
  pesos_mprecio$valoraciones.ahp,pesos_mvariedad$valoraciones.ahp,
  pesos_mcalidad$valoraciones.ahp,
  pesos_mfacilidad$valoraciones.ahp,pesos_mcompatibilidad$valoraciones.ahp,
  pesos_mrecomendacionnes$valoraciones.ahp,
  pesos_moriginales$valoraciones.ahp))

pesos
```

```
##                     Precio  Variedad   Calidad Facilidad Compatibilidad
## Netflix          0.48591874 0.5000000 0.4796500 0.4859187      0.5000000
## Prime            0.31276626 0.3333333 0.3398028 0.3127663      0.3333333
## Disney+          0.20131500 0.1666667 0.1805472 0.2013150      0.1666667
```

```
## Ponder.Criterios 0.03924589 0.2069066 0.2220623 0.1343915      0.1420409
##                  Recomendaciones Originales Ponderadores Globales
## Netflix               0.48591874  0.5454545             0.4997535
## Prime                 0.31276626  0.2727273             0.3190111
## Disney+               0.20131500  0.1818182             0.1812354
## Ponder.Criterios      0.08212692  0.1732259                    NA
```

MEJOR DECISIÓN :ALTERNATIVA A (NETFLIX) (peso global del 49.97%)

3.2 Método 2

Con la función: multicriterio.metodoAHP.variante3.completo()

```
Xmat.precio=multicriterio.crea.matrizvaloraciones(c(3/2,5/2,3/2),
           numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


Xmat.variedad=multicriterio.crea.matrizvaloraciones(c(3/2,3,2),
         numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


Xmat.calidad=multicriterio.crea.matrizvaloraciones(c(3/2,5/2,2),
        numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


Xmat.facilidad=multicriterio.crea.matrizvaloraciones(c(3/2,5/2,3/2),
         numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


Xmat.compatibilidad=multicriterio.crea.matrizvaloraciones(c(3/2,3,2),
      numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


Xmat.recomendaciones=multicriterio.crea.matrizvaloraciones(c(3/2,5/2,3/2),
        numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


Xmat.originales=multicriterio.crea.matrizvaloraciones(c(2,3,3/2),
           numalternativas = 3,v.nombres.alternativas = c("Netflix","Prime","Disney+"))


num.alt=3
num.crit=7
Xmatriznivel2=array(NA,dim=c(num.alt,num.alt,num.crit))
Xmatriznivel2[,,1]=Xmat.precio
Xmatriznivel2[,,2]=Xmat.variedad
Xmatriznivel2[,,3]=Xmat.calidad
Xmatriznivel2[,,4]=Xmat.facilidad
Xmatriznivel2[,,5]=Xmat.compatibilidad
Xmatriznivel2[,,6]=Xmat.recomendaciones
Xmatriznivel2[,,7]=Xmat.originales

pesoscomp=multicriterio.metodoAHP.variante3.completo(matriz_criterios,Xmatriznivel2)
```

```
pesoscomp$pesos.globales_entabla
```

```
##                      Precio  Variedad   Calidad Facilidad Compatibilidad
##                  0.33333333 0.3333333 0.3333333 0.3333333      0.3333333
##                  0.33333333 0.3333333 0.3333333 0.3333333      0.3333333
##                  0.33333333 0.3333333 0.3333333 0.3333333      0.3333333
## Ponder.Criterios 0.03938043 0.2045323 0.2205426 0.1348805      0.1438524
##                  Recomendaciones Originales Ponderadores Globales
##                       0.33333333  0.3333333             0.3333333
##                       0.33333333  0.3333333             0.3333333
##                       0.33333333  0.3333333             0.3333333
## Ponder.Criterios      0.08288564  0.1739260                    NA
```

3.3 Estudio de la inconsistencia con funciones R de clase

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

```
## [1] "Consistencia aceptable" "0.0419"
```

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

```
## [1] "Consistencia aceptable" "0.0011"
```

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

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

```
InconsistenciaC = multicriterio.metodoAHP.coef.inconsistencia(matriz_calidad)
c(InconsistenciaC$mensaje, round(InconsistenciaA$RI.coef.inconsistencia,4) )
```

```
## [1] "Consistencia aceptable" "0.0011"
```

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

```
## [1] "Consistencia aceptable" "0.0011"
```

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

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

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

```
## [1] "Consistencia aceptable" "0.0011"
```

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

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

4 Con ayuda de las funciones R de clase (diagram)

Se usan las funciones R en "teoriadecision_funciones_multicriterio_diagram.R":

- Método 1 (diagram):

– multicriterio.crea.matrizvaloraciones_mej()

– multicriterio.metodoahp.diagrama()

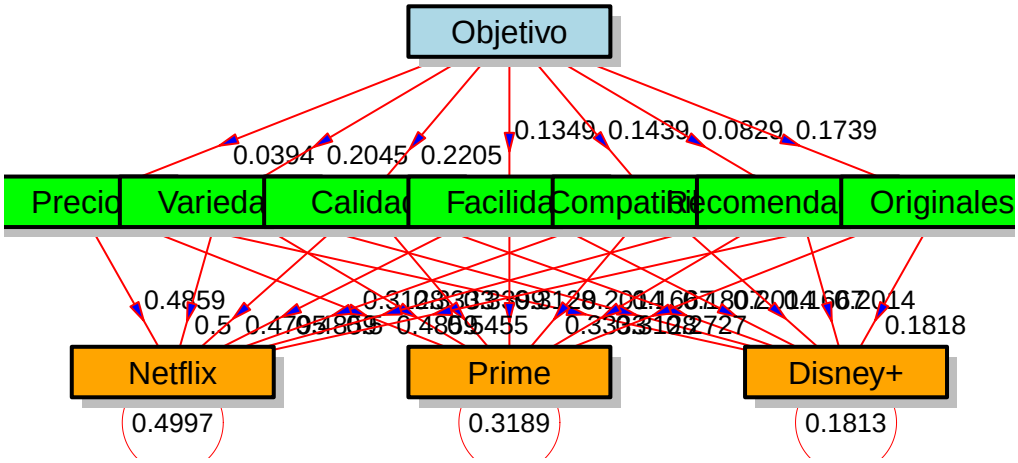
```
matriznivel=array(NA,dim=c(3,3,7))
matriznivel[,,1]=matriz_precio
matriznivel[,,2]=matriz_variedad
matriznivel[,,3]=matriz_calidad
matriznivel[,,4]=matriz_facilidad
matriznivel[,,5]=matriz_compatibilidad
matriznivel[,,6]=matriz_recomendaciones
matriznivel[,,7]=matriz_originales

dimnames(matriznivel)[[1]] = c("Netflix", "Prime", "Disney+")
dimnames(matriznivel)[[2]] = c("Netflix", "Prime", "Disney+")
dimnames(matriznivel)[[3]] = c("Precio", "Variedad", "Calidad",
        "Facilidad", "Compatibilidad", "Recomendaciones", "Originales")
matriznivel
```

```
## , , Precio
##
##           Netflix     Prime Disney+
## Netflix 1.0000000 1.5000000     2.5
## Prime   0.6666667 1.0000000     1.5
## Disney+ 0.4000000 0.6666667     1.0
##
## , , Variedad
##
##           Netflix Prime Disney+
## Netflix 1.0000000   1.5       3
## Prime   0.6666667   1.0       2
## Disney+ 0.3333333   0.5       1
##
## , , Calidad
##
##           Netflix Prime Disney+
```

```
## Netflix 1.0000000   1.5     2.5
## Prime   0.6666667   1.0     2.0
## Disney+ 0.4000000   0.5     1.0
##
## , , Facilidad
##
##           Netflix     Prime Disney+
## Netflix 1.0000000 1.5000000     2.5
## Prime   0.6666667 1.0000000     1.5
## Disney+ 0.4000000 0.6666667     1.0
##
## , , Compatibilidad
##
##           Netflix Prime Disney+
## Netflix 1.0000000   1.5       3
## Prime   0.6666667   1.0       2
## Disney+ 0.3333333   0.5       1
##
## , , Recomendaciones
##
##           Netflix     Prime Disney+
## Netflix 1.0000000 1.5000000     2.5
## Prime   0.6666667 1.0000000     1.5
## Disney+ 0.4000000 0.6666667     1.0
##
## , , Originales
##
##           Netflix     Prime Disney+
## Netflix 1.0000000 2.0000000     3.0
## Prime   0.5000000 1.0000000     1.5
## Disney+ 0.3333333 0.6666667     1.0
```

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



MEJOR DECISIÓN: ALTERNATIVA A (peso global del 49.97%)

5 Uso del paquete AHP (Analytic Hierarchy Process)

Se usan las funciones R:

- Método ahp:

– datos = Load(fichero_ahp)

– Calculate(datos)

– Visualize(datos)

– AnalyzeTable(datos, variable = “priority”) (pesos locales)

– AnalyzeTable(datos) (contribuciones)

Escribo el fichero “ahp24.ahp” con los datos.

Version: 2.0

Alternatives: &alternatives

Netflix:

Prime:

Disney+:

Goal:

name: Elegir la mejor Plataforma de Streaming

preferences:

pairwise:

- [Precio, VariedadDeContenido, 1]
- [Precio, CalidadDeVideo, 1]
- [Precio, FacilidadDeUso, 3/2]
- [VariedadDeContenido, CalidadDeVideo, 1]
- [VariedadDeContenido, FacilidadDeUso, 1]
- [CalidadDeVideo, FacilidadDeUso, 3/2]

```
- [CompatibilidadDeDispositivos, RecomendacionesPersonalizadas, 1]
- [CompatibilidadDeDispositivos, OriginalesYExclusivos, 1]
```

children:

Precio:

preferences:

pairwise:

- [Netflix,Prime,3/2]
- [Netflix,Disney+,5/2]
  - [Prime,Disney+,3/2]
    children: *alternatives

VariedadDeContenido:

preferences:

pairwise:

- [Netflix,Prime,3/2]

```
- [Netflix,Disney+,3]
- [Prime,Disney+,2]
  children: *alternatives
```

CalidadDeVideo:

preferences:

pairwise:

- [Netflix,Prime,3/2]

```
- [Netflix,Disney+,5/2]
- [Prime,Disney+,2]
  children: *alternatives
```

FacilidadDeUso:

preferences:

pairwise:

- [Netflix,Prime,3/2]
- [Netflix,Disney+,5/2]
- [Prime,Disney+,3/2]
  children: *alternatives

CompatibilidadDeDispositivos:

preferences:

pairwise:

- [Netflix,Prime,3/2]
- [Netflix,Disney+,3]
- [Prime,Disney+,2]
  children: *alternatives

RecomendacionesPersonalizadas:

preferences:

pairwise:

- [Netflix,Prime,3/2]
- [Netflix,Disney+,5/2]
- [Prime,Disney+,3/2]
  children: *alternatives

OriginalesYExclusivos:

preferences:

pairwise:

- [Netflix,Prime,2]

```
- [Netflix,Disney+,3]

- [Prime,Disney+,3/2]

  children: *alternatives
```

```
library(ahp)
#Paso 1. Cargar el modelo

ahp_ej = ahp::Load("streaming_decision.ahp")
ahp_ej
```

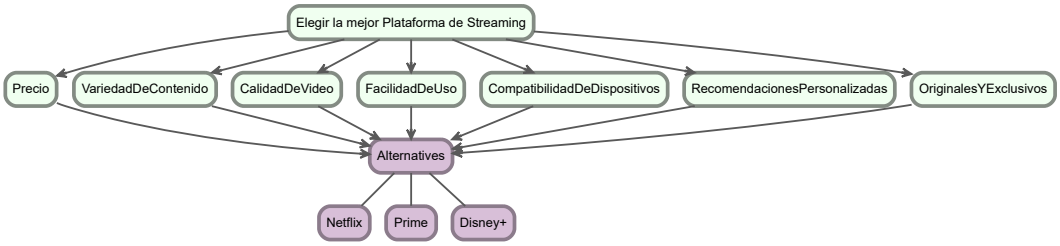
```
##                                levelName
## 1  Elegir la mejor Plataforma de Streaming
## 2    ¦--Precio
## 3    ¦   ¦--Netflix
## 4    ¦   ¦--Prime
## 5    ¦   °--Disney+
## 6    ¦--VariedadDeContenido
## 7    ¦   ¦--Netflix
## 8    ¦   ¦--Prime
## 9    ¦   °--Disney+
## 10   ¦--CalidadDeVideo
## 11   ¦   ¦--Netflix
## 12   ¦   ¦--Prime
## 13   ¦   °--Disney+
## 14   ¦--FacilidadDeUso
## 15   ¦   ¦--Netflix
## 16   ¦   ¦--Prime
## 17   ¦   °--Disney+
## 18   ¦--CompatibilidadDeDispositivos
## 19   ¦   ¦--Netflix
## 20   ¦   ¦--Prime
## 21   ¦   °--Disney+
## 22   ¦--RecomendacionesPersonalizadas
## 23   ¦   ¦--Netflix
## 24   ¦   ¦--Prime
## 25   ¦   °--Disney+
## 26   °--OriginalesYExclusivos
## 27       ¦--Netflix
## 28       ¦--Prime
## 29       °--Disney+
```

```
#Paso 2. Calcular las prioridades
Calculate(ahp_ej)
print(ahp_ej, priority = function(x) x$parent$priority["Total", x$name])
```

```
##                                levelName  priority
## 1  Elegir la mejor Plataforma de Streaming        NA
## 2    ¦--Precio                               0.1514119
## 3    ¦   ¦--Netflix                         0.4859187
## 4    ¦   ¦--Prime                           0.3127663
## 5    ¦   °--Disney+                         0.2013150
```

```
## 6   ¦--VariedadDeContenido               0.1423082
## 7   ¦   ¦--Netflix                       0.5000000
## 8   ¦   ¦--Prime                         0.3333333
## 9   ¦   °--Disney+                       0.1666667
## 10  ¦--CalidadDeVideo                    0.1514119
## 11  ¦   ¦--Netflix                       0.4796500
## 12  ¦   ¦--Prime                         0.3398028
## 13  ¦   °--Disney+                       0.1805472
## 14  ¦--FacilidadDeUso                    0.1279434
## 15  ¦   ¦--Netflix                       0.4859187
## 16  ¦   ¦--Prime                         0.3127663
## 17  ¦   °--Disney+                       0.2013150
## 18  ¦--CompatibilidadDeDispositivos      0.1423082
## 19  ¦   ¦--Netflix                       0.5000000
## 20  ¦   ¦--Prime                         0.3333333
## 21  ¦   °--Disney+                       0.1666667
## 22  ¦--RecomendacionesPersonalizadas     0.1423082
## 23  ¦   ¦--Netflix                       0.4859187
## 24  ¦   ¦--Prime                         0.3127663
## 25  ¦   °--Disney+                       0.2013150
## 26  °--OriginalesYExclusivos             0.1423082
## 27      ¦--Netflix                       0.5454545
## 28      ¦--Prime                         0.2727273
## 29      °--Disney+                       0.1818182
```

```r
#Paso 3. Visualizar la jerarquía
Visualize(ahp_ej)
```




```
#Paso 4
Analyze(ahp_ej)
```

```
##                                       Weight Netflix Prime Disney+
## 1 Elegir la mejor Plataforma de Streaming 100.0%   49.7% 31.7%   18.6%
## 2  ¦--Precio                               15.1%    7.4%  4.7%    3.0%
## 3  ¦--CalidadDeVideo                       15.1%    7.3%  5.1%    2.7%
## 4  ¦--CompatibilidadDeDispositivos         14.2%    7.1%  4.7%    2.4%
## 5  ¦--RecomendacionesPersonalizadas        14.2%    6.9%  4.5%    2.9%
## 6  ¦--OriginalesYExclusivos                14.2%    7.8%  3.9%    2.6%
## 7  ¦--VariedadDeContenido                  14.2%    7.1%  4.7%    2.4%
## 8  °--FacilidadDeUso                       12.8%    6.2%  4.0%    2.6%
##   Inconsistency
## 1          0.3%
## 2          0.1%
## 3          0.4%
## 4          0.0%
## 5          0.1%
## 6          0.0%
## 7          0.0%
## 8          0.1%
```

```r
#Paso 5. Analizar con Tabla Mejorada
aa = AnalyzeTable(ahp_ej)
formattable::as.htmlwidget(aa)
```

	Weight	Netflix	Prime	Disney+	Inconsistency
Elegir la mejor Plataforma de Streaming	100.0%	49.7%	31.7%	18.6%	0.3%
Precio	15.1%	7.4%	4.7%	3.0%	0.1%
CalidadDeVideo	15.1%	7.3%	5.1%	2.7%	0.4%
CompatibilidadDeDispositivos	14.2%	7.1%	4.7%	2.4%	0.0%
RecomendacionesPersonalizadas	14.2%	6.9%	4.5%	2.9%	0.1%
OriginalesYExclusivos	14.2%	7.8%	3.9%	2.6%	0.0%
VariedadDeContenido	14.2%	7.1%	4.7%	2.4%	0.0%
FacilidadDeUso	12.8%	6.2%	4.0%	2.6%	0.1%

Aplicación Shiny de la librería ahp:

A continuación se recogen imágenes que ilustran el uso de la aplicación Shiny que contiene la librería ahp después de ejecutar la instrucción R:

6 Aplicacion método ELECTRE

Resumen de Pesos Asignados:

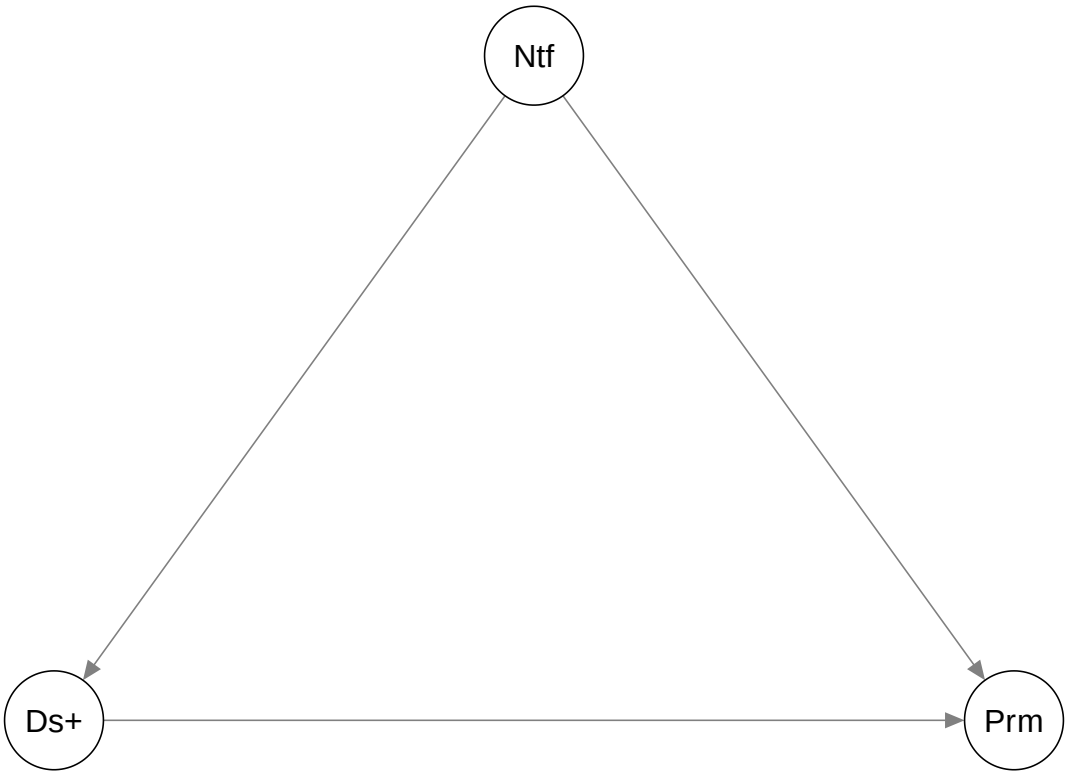
Criterio	Peso
Precio	20%
Variedad de Contenido	25%
Calidad de Video	20%
Facilidad de Uso	15%
Compatibilidad de Dispositivos	10%
Recomendaciones Personalizadas	5%
Exclusivos y Originales	5%

```
matriz_decision=multicriterio.crea.matrizdecision(
  c(3,5,5,5,5,5,5,5,4,3,4,4,4,3,3,4,4,5,5,5,4,4),numalternativas = 3,
  numcriterios = 7,v.nombresalt = c("Netflix","Prime","Disney+"),
  v.nombrescri = c("Precio", "Variedad", "Calidad","Facilidad",
"Compatibilidad", "Recomendaciones", "Originales"))
matriz_decision
```

```
##         Precio Variedad Calidad Facilidad Compatibilidad Recomendaciones
## Netflix      3        5       5         5              5               5
## Prime        5        4       3         4              4               3
## Disney+      4        4       5         5              5               4
##         Originales
## Netflix          5
## Prime            3
## Disney+          4
```

```
salida=multicriterio.metodoELECTRE_I(matriz_decision,pesos.criterios
=c(0.20,0.25,0.20,0.15,0.10,0.5,0.5),nivel.concordancia.minimo.alpha = 0.7,
no.se.compensan = c(Inf,Inf,Inf,Inf,Inf,Inf,Inf),que.alternativas = TRUE)
```

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




```
salida$nucleo_aprox
```

```
## Netflix
##       1
```

```
r1=func_ELECTRE_Completo(salida)

#r1$MIndices$KE %>% save_kable("test1.png")
#include_graphics("test1.png")

#r1$MConcordancia$KE %>% save_kable("test2.png")
#include_graphics("test2.png")

#r1$MDiscordancia$KE %>% save_kable("test3.png")
#include_graphics("test3.png")

#r1$MSuperacion$KE %>% save_kable("test4.png")
#include_graphics("test4.png")
```