

Marina Rivero

Marina está barajando la posibilidad de darse de alta en una plataforma para ver series y películas. Para ver cual es la que más se adapta a sus necesidades se ha informado y ha encontrado la siguiente información:

- Con la plataforma 1 podría ver todo el contenido que quiera (series, películas, etc). Esta plataforma tiene un precio de 20€ al mes. Si queremos añadirle una pantalla más, es decir, poder ver el contenido en dos dispositivos diferentes al mismo tiempo debemos pagar 5€ más al mes.
- Con la plataforma 2 tendría acceso a solo a series o películas por 12 euros. Si queremos tener todo el contenido debería pagar 10€ más cada mes. La ampliación a dos pantallas cuesta 3,5€ al mes.
- Con la plataforma 3 dispondríamos de series y películas en dos dispositivos por 23€ al mes.

¿Qué le aconsejaría si su objetivo es minimizar los costes?

Planteamiento

- Un decisor (Marina)
- Modelos de costos (desfavorable)
- Alternativas:
 - d1="Plataforma 1"
 - d2="Plataforma 2"
 - d3="Plataforma 3"
- Estados de la naturaleza:
 - e1="Series, películas y dos pantallas"
 - e2="Series y/o películas y una pantallas"
- Valoraciones (costos de los dos)
 - Estado e1:
 - $x_{1,1}$ \$ 25
 - $x_{2,1}$ \$ 25.5
 - $x_{3,1}$ \$ 23
 - Estado e2:
 - $x_{1,2}$ \$ 20
 - $x_{2,2}$ \$ 12

- \$x_{\{3,1\}}\$ 23

Resolucion

Como nos indica el enunciado del trabajo lo primero que tenemos que hacer es crear un enunciado de un problema, para después resolverlo tanto con las funciones dadas en el scrip “teoriadecision_funciones_incertidumbre.R” como la nueva función creada por nosotros.

Creamos la matriz de datos a la que posteriormente aplicaremos todos los métodos que contiene la librería “teoriadecision_funciones_incertidumbre.R”

```
tablaX=crea.tablaX(c(25, 20,
                    25.5, 12,
                    23, 23), numalternativas = 3, numestados = 2)
```

Aplicamos los métodos

```
sol=criterio.Todos(tablaX, alfa = 0.5, favorable = T)
sol
```

##		e1	e2	Wald	Optimista	Hurwicz	Savage	Laplace	Punto	Ideal
## d1		25.0	20	20	25.0	22.50	3.0	22.50		3.041
## d2		25.5	12	12	25.5	18.75	11.0	18.75		11.000
## d3		23.0	23	23	23.0	23.00	2.5	23.00		2.500
## iAlt.Opt (fav.)	<NA>	<NA>	d3		d2	d3	d3	d3		d3

La opción de la plataforma 3 es la mejor siguiendo todos los criterios, menos con el optimista, que nos quedaríamos con la plataforma 2

Aplicamos la función que hemos creado

```
# Función Principal (creada por el grupo)
intervalos.alfa= function(tablaX,favorable=TRUE) {

  alfa=seq(0,1,by=0.01) # Introducimos un conjunto de alfas que nos servirán
  # para saber cuándo cambia la alternativa óptima. Fijamos un valor de 0.01,
  # el cual indica cada cuánto se quiere que exista alfa.
  X = tablaX

  if(favorable){ #en el caso de que sea favorable
    Altmin = apply(X,MARGIN=1,min)
    Altmax= apply(X,MARGIN=1,max)

    # Como ya no tenemos un alfa, sino varios, debemos crear un bucle que
    # trabaje con todos los alfa. Como, además, Altmin y Altmax son vecto-
    # res, tenemos que crear listas que nos devuelvan, para cada elemento
    # de dichos vectores, cuáles son las alternativas asociadas a los
    # alfas.

    AltH=list()
    Hurwicz=list()
    Alt_Hurwicz=list()

    # Creamos el bucle:

    for(i in 1:length(alfa)){
      AltH[[i]] = alfa[i] * Altmax + (1-alfa[i]) * Altmin
```

```

    Hurwicz[[i]] = max(AltH[[i]])
    Alt_Hurwicz[[i]] = which.max.general(AltH[[i]])
  }

  metodo = 'favorable'

} else { #en caso de que no sea favorable
  Altmin = apply(X,MARGIN=1,min)
  Altmax= apply(X,MARGIN=1,max)

  AltH=list()
  Hurwicz=list()
  Alt_Hurwicz=list()

  for(i in 1:length(alfa)){
    AltH[[i]] = (1-alfa[i]) * Altmax + alfa[i] * Altmin
    Hurwicz[[i]] = min(AltH[[i]])
    Alt_Hurwicz[[i]] = which.min.general(AltH[[i]])
  }

  metodo = 'desfavorable'

}

resultados = list();
resultados$metodo = metodo;
resultados$ValorAlternativa = unlist(Hurwicz); # Valores que toma cada
# alfa en su alternativa óptima.
resultados$alfa = alfa; # Alfas usados.
resultados$AlternativaOptima = unlist(Alt_Hurwicz);
resultados$Solucion = unlist(Alt_Hurwicz);
names(resultados$Solucion)=alfa;
# distinct(as.data.frame(resultados,alfa));
prueba=cbind.data.frame(alfa,resultados$Solucion[1:length(alfa)]);
colnames(prueba)<-c("Alfa", "Solución");
rownames(prueba %>% distinct(Solución));
alternativas= prueba %>% distinct(Solución);
resultados$intervalos = alternativas
return(resultados)
}

intervalos.alfa(tablaX,FALSE)

## $metodo
## [1] "desfavorable"
##
## $ValorAlternativa
## [1] 23.000 23.000 23.000 23.000 23.000 23.000 23.000 23.000 23.000 23.000
## [11] 23.000 23.000 23.000 23.000 23.000 23.000 23.000 23.000 23.000 22.935
## [21] 22.800 22.665 22.530 22.395 22.260 22.125 21.990 21.855 21.720 21.585
## [31] 21.450 21.315 21.180 21.045 20.910 20.775 20.640 20.505 20.370 20.235
## [41] 20.100 19.965 19.830 19.695 19.560 19.425 19.290 19.155 19.020 18.885
## [51] 18.750 18.615 18.480 18.345 18.210 18.075 17.940 17.805 17.670 17.535
## [61] 17.400 17.265 17.130 16.995 16.860 16.725 16.590 16.455 16.320 16.185

```

```

## [71] 16.050 15.915 15.780 15.645 15.510 15.375 15.240 15.105 14.970 14.835
## [81] 14.700 14.565 14.430 14.295 14.160 14.025 13.890 13.755 13.620 13.485
## [91] 13.350 13.215 13.080 12.945 12.810 12.675 12.540 12.405 12.270 12.135
## [101] 12.000
##
## $alfa
## [1] 0.00 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.10 0.11 0.12 0.13 0.14
## [16] 0.15 0.16 0.17 0.18 0.19 0.20 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29
## [31] 0.30 0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.40 0.41 0.42 0.43 0.44
## [46] 0.45 0.46 0.47 0.48 0.49 0.50 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59
## [61] 0.60 0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69 0.70 0.71 0.72 0.73 0.74
## [76] 0.75 0.76 0.77 0.78 0.79 0.80 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89
## [91] 0.90 0.91 0.92 0.93 0.94 0.95 0.96 0.97 0.98 0.99 1.00
##
## $AlternativaOptima
## d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d3 d2 d2 d2 d2 d2 d2 d2
## 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2
## d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2 d2
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## $Solucion
## 0 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 0.11 0.12 0.13 0.14 0.15
## 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## 0.16 0.17 0.18 0.19 0.2 0.21 0.22 0.23 0.24 0.25 0.26 0.27 0.28 0.29 0.3 0.31
## 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2
## 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4 0.41 0.42 0.43 0.44 0.45 0.46 0.47
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## 0.48 0.49 0.5 0.51 0.52 0.53 0.54 0.55 0.56 0.57 0.58 0.59 0.6 0.61 0.62 0.63
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## 0.64 0.65 0.66 0.67 0.68 0.69 0.7 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## 0.8 0.81 0.82 0.83 0.84 0.85 0.86 0.87 0.88 0.89 0.9 0.91 0.92 0.93 0.94 0.95
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## 0.96 0.97 0.98 0.99 1
## 2 2 2 2 2
##
## $intervalos
## Solución
## 0 3
## 0.19 2

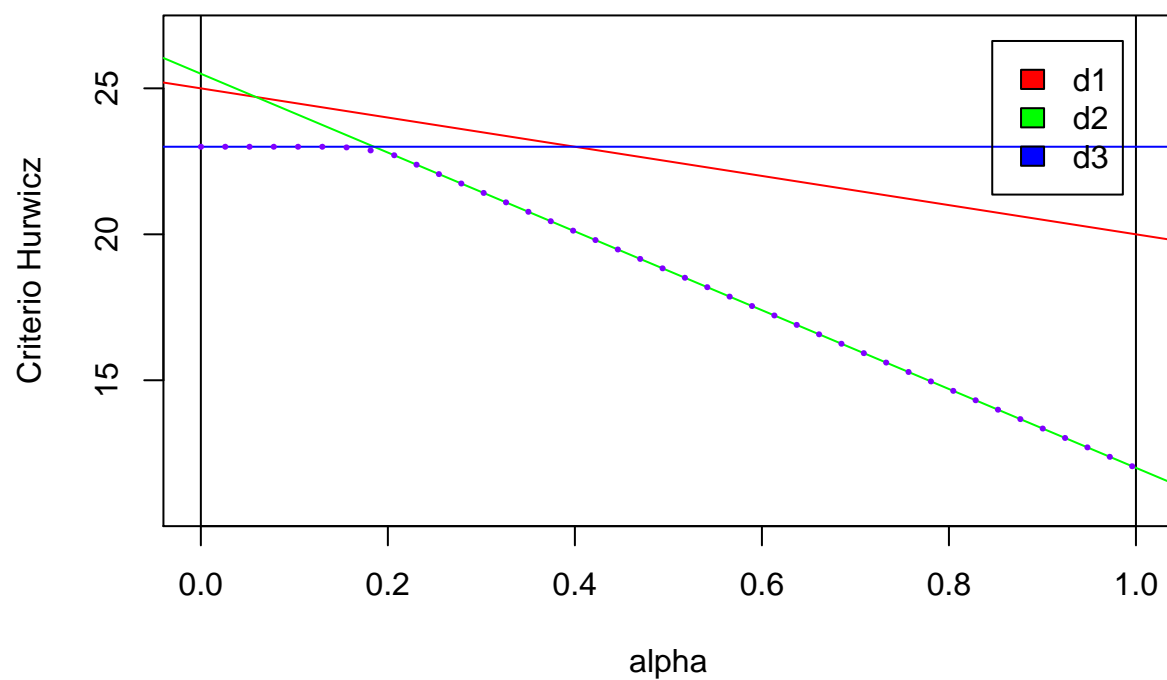
```

El resultado nos muestra que si alfa esta entre [0, 0.19) la mejor opción sera la plataforma 3, en cambio si alfa esta entre (0.19 y 1] la mejor opción sera la plataforma 2

Comprobamos estos resultados graficamente

```
dibuja.criterio.Hurwicz(tablaX,FALSE) # Comprobamos.
```

Criterio de Hurwicz (desfavorable – línea discontinua)



Vemos que los resultados coinciden con el resultado que nos da la función