

R Notebook

Problema

Alberto está de viaje y está alojado temporalmente en una casa y mañana por la mañana parte a otra zona del país que está visitando. Por ser la última noche quiere salir a cenar fuera y de paso, si lo ve conveniente probar un helado porque al parecer en esa zona y los pueblos de alrededor hacen unos helados excelentes. Tiene 3 opciones:

1) Cenar en un restaurante al norte de la ciudad por 30 euros y, si quiere helado coger el coche y conducir al pueblo del norte, gastando así 5 euros más en gasolina, y pagar 5 euros por un helado; si no lo quiere se volvería andando.

2) Cenar en un restaurante a las afueras, por la zona sur, por 15 euros y coger el coche gastando 15 euros en gasolina para tomarse un helado en la heladería anterior por lo que gastaría otros 5 euros por él. Igualmente, si rechaza la opción del helado, vuelve andando.

3) Cenar al lado del apartamento por 25 euros y tomarse un helado en la mejor heladería de la ciudad en la cual se gastaría otros 20 euros adicionales por el pack de un helado en el momento en el local y llevarse una tarrina extra.

¿Cuál es la mejor opción para minimizar los costes?

- Solución usando la función creada:
 - Planteamiento:
 - Un decisor
 - Modelo de costes (desfavorable)
 - Alternativas:
 - d1= "Cenar zona norte y tomar helado"
 - d2= "Cenar zona sur y tomar helado"
 - d3= "Cenar cerca apartamento y tomar helado"
 - Estados de la naturaleza:
 - e1= "Toma helado"
 - e2= "No toma helado"
 - Valoraciones:

- Estado e1: representado en euros
 - $x_{1,1} = \$30 + 10$
 - $x_{2,1} = \$15 + 20$
 - $x_{3,1} = \$25 + 20$
- Estado e2: representado en euros
 - $x_{2,1} = \$30$
 - $x_{2,2} = \$15$
 - $x_{2,3} = \$25$
- Introducción de datos:

```
source("teoriadecision_funciones_incertidumbre.R")
tabla=crea.tablaX(c(40,30,
                   35,15,
                   45,25),numalternativas = 3, numestados = 2)
criterio.Hurwicz.General2 = function(tablaX,alfa=0.3,favorable=TRUE) {
  # si alfa es un escalar entre 0 y 1 lo obtiene para ese único valor
  # si alfa es igual a un número mayor que 1, lo usa para obtener
  cálculos para dividir el rango 0-1
  X = tablaX;
  if (favorable) {
    Altmin = apply(X,MARGIN=1,min);
    Altmax= apply(X,MARGIN=1,max);
    if (alfa<=1) {
      valfa = seq(from=0,to=1,by=(alfa));
    } else {
      valfa = seq(from=0,to=1,by=(1/alfa)); ## alfa: 100, 200,
    }
    vHurwicz = rep(0,length(valfa))
    Alt_vHurwicz = rep(0,length(valfa))
    ##
    alfab = valfa[1];
    vAlth = alfab * Altmax + (1-alfab) * Altmin;
    vHurwicz[1] = max(vAlth);
    Alt_vHurwicz[1] = which.max(vAlth);
    Alt_vHurwicz_g = which.max.general(vAlth);

    alfasCambio=c(alfab);
    alternativas=c(which.max(vAlth));

    for (i in 2:length(valfa)) {
      alfab = valfa[i];
      vAlth = alfab * Altmax + (1-alfab) * Altmin;
      vHurwicz[i] = max(vAlth);
```

```

    Alt_vHurwicz[i] = which.max(vAlth);
    Alt_vHurwicz_g = which.max.general(vAlth);

    if (Alt_vHurwicz[i]!=Alt_vHurwicz[i-1]) {
      alfasCambio=c(alfasCambio,alfab);
      alternativas=c(alternativas, which.max(vAlth));
    }
  }
  metodo = 'favorable';
} else {
  Altmin = apply(X,MARGIN=1,min);
  Altmax= apply(X,MARGIN=1,max);
  if (alfa<=1) {
    valfa = seq(from=0,to=1,by=(alfa));
  } else {
    valfa = seq(from=0,to=1,by=(1/alfa)); ## alfa: 100, 200,
  }
  vHurwicz = rep(0,length(valfa))
  Alt_vHurwicz = rep(0,length(valfa))

  alfab = valfa[1];
  vAlth = (1-alfab) * Altmax + alfab * Altmin;
  vHurwicz[1] = min(vAlth);
  Alt_vHurwicz[1] = which.min(vAlth);
  Alt_vHurwicz_g = which.min.general(vAlth);

  alfasCambio=c(alfab);
  alternativas=c(which.min(vAlth));

  for (i in 2:length(valfa)) {
    alfab = valfa[i];
    vAlth = (1-alfab) * Altmax + alfab * Altmin;
    vHurwicz[i] = min(vAlth);
    Alt_vHurwicz[i] = which.min(vAlth);
    Alt_vHurwicz_g = which.min.general(vAlth);
    if (Alt_vHurwicz[i]!=Alt_vHurwicz[i-1]) {
      alfasCambio=c(alfasCambio,alfab);
      alternativas=c(alternativas, which.min(vAlth));
    }
  }
  metodo = 'desfavorable';
}
resultados = list();
resultados$criterio = 'Hurwicz';
resultados$alfa = alfa;
resultados$metodo = metodo;
resultados$tablaX = tablaX;
resultados$ValorAlternativas = vAlth;
resultados$ValorOptimo = vHurwicz;

```

```

resultados$alternativasPorIntervalos = alternativas;
resultados$alfas = alfasCambio;

if (length(valfa)==1) {
  resultados$AlternativaOptima = Alt_vHurwicz_g;
} else {
  resultados$AlternativaOptima = Alt_vHurwicz;
}

return(resultados);
}

criterio.Hurwicz.General2(tabla, favorable = FALSE)

## $criterio
## [1] "Hurwicz"
##
## $alfa
## [1] 0.3
##
## $metodo
## [1] "desfavorable"
##
## $tablaX
##      e1 e2
## d1 40 30
## d2 35 15
## d3 45 25
##
## $ValorAlternativas
## d1 d2 d3
## 31 17 27
##
## $ValorOptimo
## [1] 35 29 23 17
##
## $alternativasPorIntervalos
## d2
## 2
##
## $alfas
## [1] 0
##
## $AlternativaOptima
## [1] 2 2 2 2

```

- Solucion con el criterio de Laplace:

```
criterio.Laplace(tabla, favorable=FALSE)
```

```
## $criterio
## [1] "Laplace"
##
## $metodo
## [1] "desfavorable"
##
## $tablaX
##      e1 e2
## d1 40 30
## d2 35 15
## d3 45 25
##
## $ValorAlternativas
## d1 d2 d3
## 35 25 35
##
## $ValorOptimo
## [1] 25
##
## $AlternativaOptima
## d2
## 2
```

- Solución usando el criterio optimista:

```
criterio.Optimista(tabla, favorable=FALSE)
```

```
## $criterio
## [1] "Optimista"
##
## $metodo
## [1] "desfavorable"
##
## $tablaX
##      e1 e2
## d1 40 30
## d2 35 15
## d3 45 25
##
## $ValorAlternativas
## d1 d2 d3
## 30 15 25
##
## $ValorOptimo
## [1] 15
##
## $AlternativaOptima
## d2
## 2
```

- Solución usando el criterio Punto Ideal:

```
criterio.PuntoIdeal(tabla,favorable=FALSE)

## $criterio
## [1] "Punto Ideal"
##
## $metodo
## [1] "desfavorable"
##
## $tablaX
##      e1 e2
## d1 40 30
## d2 35 15
## d3 45 25
##
## $Mejores
## e1 e2
## 35 15
##
## $ValorAlternativas
## [1] 15.81139 0.00000 14.14214
##
## $ValorOptimo
## [1] 0
##
## $AlternativaOptima
## [1] 2
```

- Solución usando el criterio de Savage:

```
criterio.Savage(tabla,favorable=FALSE)

## $criterio
## [1] "Savage"
##
## $metodo
## [1] "desfavorable"
##
## $tablaX
##      e1 e2
## d1 40 30
## d2 35 15
## d3 45 25
##
## $Mejores
## e1 e2
## 35 15
##
## $Pesos
##      e1 e2
```

```
## d1  5 15
## d2  0  0
## d3 10 10
##
## $ValorAlternativas
## d1 d2 d3
## 15  0 10
##
## $ValorOptimo
## [1] 0
##
## $AlternativaOptima
## d2
## 2
```

- Solución usando el criterio de Wald:

```
criterio.Wald(tabla,favorable=FALSE)
```

```
## $criterio
## [1] "Wald"
##
## $metodo
## [1] "desfavorable"
##
## $tablaX
##      e1 e2
## d1 40 30
## d2 35 15
## d3 45 25
##
## $ValorAlternativas
## d1 d2 d3
## 40 35 45
##
## $ValorOptimo
## [1] 35
##
## $AlternativaOptima
## d2
## 2
```