

Trabajo 1 TD

Álvaro Fernández-Blanco Centeno

4/11/2021

Enunciado:

Una empresa textil sabe que se le pueden presentar 3 estados o situaciones posibles para el año que viene:

E1. La empresa tiene ventas bajas E2. La empresa tiene ventas medias E3. La empresa tiene ventas altas

Sin embargo, antes tiene que tomar una de las siguientes 3 decisiones

A1. Realizar una inversión elevada A2. Realizar una inversión media A3. Realizar una inversión baja

Los resultados de cada decisión (beneficios) en miles de euros varían, dependiendo del estado que finalmente se presente y se puede reflejar con la siguiente matriz:

¿Cuál es la mejor opción para maximizar los beneficios de la empresa?

```
library(knitr)
source("TDecisionCodigo/teoriadecision_funciones_incertidumbre.R")

vec=c(180,250,300,200,280,290,220,240,250)
tabla=matrix(vec,ncol = 3,nrow = 3, byrow = TRUE)
as.data.frame(tabla)

##      V1  V2  V3
## 1 180 250 300
## 2 200 280 290
## 3 220 240 250

colnames(tabla)=c("Ventas bajas","Ventas medias","Ventas altas")
rownames(tabla)=c("Inversión baja","Inversión media","Inversión alta")
tabla
```

```
##              Ventas bajas Ventas medias Ventas altas
## Inversión baja           180           250           300
## Inversión media          200           280           290
## Inversión alta           220           240           250
```

Solución:

Pasamos a resolver el problema con cada uno de los métodos vistos:

Criterio de Wald o pesimista:

```
mat=crea.tablaX(vec,numalternativas = 3,numestados = 3)
criterio.Wald(mat)

## $criterio
## [1] "Wald"
```

```
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1  e2  e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $ValorAlternativas
##   d1  d2  d3
## 180 200 220
##
## $ValorOptimo
## [1] 220
##
## $AlternativaOptima
## d3
## 3
```

Con este método, la alternativa que óptima es la tercera.

Criterio Optimista:

```
criterio.Optimista(mat)
```

```
## $criterio
## [1] "Optimista"
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1  e2  e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $ValorAlternativas
##   d1  d2  d3
## 300 290 250
##
## $ValorOptimo
## [1] 300
##
## $AlternativaOptima
## d1
## 1
```

Con este método, la alternativa que óptima es la primera.

Criterio Hurwicz:

```
criterio.Hurwicz(mat)
```

```
## $criterio
```

```
## [1] "Hurwicz"
##
## $alfa
## [1] 0.3
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1  e2  e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $ValorAlternativas
##   d1  d2  d3
## 216 227 229
##
## $ValorOptimo
## [1] 229
##
## $AlternativaOptima
## d3
## 3
```

Con este método, la alternativa que óptima es la tercera.

Criterio Savage:

```
criterio.Savage(mat)
```

```
## $criterio
## [1] "Savage"
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1  e2  e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $Mejores
##   e1  e2  e3
## 220 280 300
##
## $Pesos
##   e1 e2 e3
## d1 40 30 0
## d2 20 0 10
## d3 0 40 50
##
## $ValorAlternativas
## d1 d2 d3
## 40 20 50
```

```
##
## $ValorOptimo
## [1] 20
##
## $AlternativaOptima
## d2
## 2
```

Con este método, la alternativa que óptima es la segunda.

Criterio Laplace:

```
criterio.Laplace(mat)
```

```
## $criterio
## [1] "Laplace"
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1 e2 e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $ValorAlternativas
##      d1      d2      d3
## 243.3333 256.6667 236.6667
##
## $ValorOptimo
## [1] 256.6667
##
## $AlternativaOptima
## d2
## 2
```

Con este método, la alternativa que óptima es la segunda.

Criterio Punto Ideal:

```
criterio.PuntoIdeal(mat)
```

```
## $criterio
## [1] "Punto Ideal"
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1 e2 e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $Mejores
## e1 e2 e3
```

```
## 220 280 300
##
## $ValorAlternativas
##      d1      d2      d3
## 50.00000 22.36068 64.03124
##
## $ValorOptimo
## [1] 22.36068
##
## $AlternativaOptima
## d2
## 2
```

Con este método, la alternativa que óptima es la segunda.

Aplicamos ahora todos los criterios a la vez:

```
criterio.Todos(mat)
```

```
##           e1  e2  e3 Wald Optimista Hurwicz Savage Laplace Punto Ideal
## d1           180 250 300 180          300      216    40   243.3      50.00
## d2           200 280 290 200          290      227    20   256.7      22.36
## d3           220 240 250 220          250      229    50   236.7      64.03
## iAlt.Opt (fav.) -- -- --   d3          d1      d3    d2     d2          d2
```

Creamos la función pedida:

```
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);
}

```

La aplicamos al caso particular del enunciado:

```
criterio.Hurwicz.General2(mat)
```

```

## $criterio
## [1] "Hurwicz"
##
## $alfa
## [1] 0.3
##
## $metodo
## [1] "favorable"
##
## $tablaX
##      e1 e2 e3
## d1 180 250 300
## d2 200 280 290
## d3 220 240 250
##
## $ValorAlternativas
##  d1 d2 d3
## 288 281 247
##
## $ValorOptimo
## [1] 220 229 254 288
##
## $alternativasPorIntervalos
## d3 d2 d1
##  3  2  1
##
## $alfas
## [1] 0.0 0.6 0.9
##
## $AlternativaOptima
## [1] 3 3 2 1

```