Practica 1 2024

Ignacio Fernández Contreras

ifcau3z@uma.es
Planificación de Proyectos y Análisis de Riesgos . E.T.S Informática.

1 Problema 1

Un pastelero dispone de 150 kg de harina, 22 kg de azúcar y 27.5 kg de mantequilla para elaborar los tipos de pasteles (A y B). Cada caja de pasteles de tipo A requiere 3 kg de harina, 1 kg de azúcar y 1 kg de mantequilla y su venta le reporta un beneficio de 20 euros. Cada caja de pasteles de tipo B requiere 6 kg de harina, 0.5 kg de azúcar y 1 kg de mantequilla y su venta le reporta un beneficio de 30 euros.¿Cuántas cajas de cada tipo debe elaborar el pastelero de manera que se maximicen sus ganancias? (Se supone en principio que también puede elaborar cajas incompletas, es decir, que no se trata de un problema de programación entera.). Define un modelo de PL y resuelve el problema con R (paquetes lpSolve y linprog) y PHPSimplex (http://www.phpsimplex.com/).

1.1 Paso 1: Definir las variables de decisión

Vamos a definir las variables de decisión para representar cuántas cajas de cada tipo de pastel el pastelero debe producir:

- x_1: Número de cajas de pasteles de tipo A
- x_2: Número de cajas de pasteles de tipo B

1.2 Paso 2: Definir la función objetivo

El objetivo es maximizar las ganancias del pastelero. Las ganancias provienen de la venta de cajas de pasteles A y B:

- Beneficio por cada caja de pastel tipo A: 20 euros
- Beneficio por cada caja de pastel tipo B: 30 euros

Por lo tanto, la función objetivo es:

$$\text{Maximizar Z} = 20\ x_1 + 30\ x_2$$

1.3 Paso 3: Definir las restricciones

Las restricciones vienen dadas por la cantidad de recursos disponibles (harina, azúcar y mantequilla) y el uso de estos recursos por cada tipo de pastel.

1. **Harina**: El pastelero tiene 150 kg de harina disponibles. Cada caja de pastel A usa 3 kg de harina y cada caja de pastel B usa 6 kg de harina:

$$3\ x_1 + 6\ x_2 \leq 150$$

2. **Azúcar**: El pastelero tiene 22 kg de azúcar disponibles. Cada caja de pastel A usa 1 kg de azúcar y cada caja de pastel B usa 0.5 kg de azúcar:

$$x_1 + 10.5\ x_2 \leq 22$$

3. **Mantequilla**: El pastelero tiene 27.5 kg de mantequilla disponibles. Cada caja de pastel A usa 1 kg de mantequilla y cada caja de pastel B usa 1 kg de mantequilla:

$$x_1 + x_2 \leq 27.5$$

4. **No negatividad:** No se pueden producir cantidades negativas de cajas de pasteles, por lo que las variables deben ser no negativas:

$$x_1 \geq,\ x_2 \geq 0$$

1.4 Paso 4: Modelo de programación lineal (PL)

El modelo de programación lineal queda como:

$$\text{Maximizar Z} = 20x_1 + 30x_2$$

Sujeto a:

$$\begin{gathered} 3x_1 + 6x_2 \leq 150, \\ \mathrm{x}_1 + 10{,}5x_2 \leq 22, \\ \mathrm{x}_1 + x_2 \leq 27{,}5, \\ \mathrm{x}_1, x_2 \geq 0 \end{gathered}$$

Usando R:

```
library(lpSolve)

# Coeficientes de la funcion objetivo (beneficios)
objetivo <- c(20, 30)

# Matriz de coeficientes de las restricciones
restricciones <- matrix(c(3, 6,    # Harina
                          1, 0.5, # Az car
                          1, 1),  # Mantequilla
                        nrow=3, byrow=TRUE)

# Lado derecho de las restricciones (recursos disponibles)
derecho <- c(150, 22, 27.5)

# Tipos de restricciones (todas son <=)
direccion <- c("<=", "<=", "<=")



# Solucion optima

optimo <- lp(direction = "max",
             objective.in = objetivo,
             const.mat = restricciones,
             const.dir = direccion,
             const.rhs = derecho)

cat("El beneficio maximo es de: ", optimo$objval)
cat("La solucion optima es: ", optimo$solution)
```

```
> cat("El beneficio máximo es de: ", optimo$objval)
El beneficio máximo es de:  775
> cat("La solución óptima es: ", optimo$solution)
La solución óptima es:  5 22.5
```

Usando linprog:

```
library(linprog)

# Coeficientes de la funcion objetivo (beneficios)
objetivo <- c(20, 30)

# Matriz de coeficientes de las restricciones
restricciones <- matrix(c(3, 6,   # Harina
                          1, 0.5, # Az car
                          1, 1),  # Mantequilla
                        nrow=3, byrow=TRUE)

# Lado derecho de las restricciones (recursos disponibles)
derecho <- c(150, 22, 27.5)

# Tipos de restricciones (todas son <=)
direccion <- c("<=", "<=", "<=")



# Solucion optima

optimo <- solveLP(objetivo,derecho, restricciones, maximum = TRUE, direccion)
cat("Ganancia m xima: ", optimo$opt)
cat("Cantidad de cajas del tipo A: ", optimo$solution[1])
cat("Cantidad de cajas del tipo B: ", optimo$solution[2])
```

```
> cat("Ganancia máxima: ", optimo$opt)
Ganancia máxima:  775
> cat("Cantidad de cajas del tipo A: ", optimo$solution[1])
Cantidad de cajas del tipo A:  5
> cat("Cantidad de cajas del tipo B: ", optimo$solution[2])
Cantidad de cajas del tipo B:  22.5
```

2 Problema 2

Distribución de almacenes y fábricas. Una empresa de distribución tiene tres almacenes que denominaremos A1, A2 y A3, y cuatro fábricas que denominaremos F1, F2, F3 y F4. Los almacenes A1, A2 y A3 tienen una capacidad máxima de 100, 200 y 150 piezas respectivamente. Las fábricas F1, F2, F3 y F4 tienen unos requisitos mínimos de materia primera de 100, 115, 80 y 105 piezas respectivamente. En la siguiente tabla tenemos los costes de transportar una pieza de materia prima desde cada uno de los almacenes a cada una de las fábricas, y el resumen de capacidades y demandas.

Se pide:

	F1	F2	F3	F4	Capacidad
A1	4	6	5	9	100
A2	8	7	9	4	200
A3	7	8	7	6	150
Demanda	100	115	80	105	

1. **Define un modelo de PL que permita encontrar cuántas piezas se han de transportar desde cada almacén a cada fábrica tal que asegure la demanda de las fábricas y el coste sea mínimo. Plantear el modelo en lenguaje R y resolverlo (paquetes lpSolve y linprog)**

Usando lpsolve:

```
library(lpSolve)

# Definir la matriz de costes (Almacenes x F bricas)
costes <- matrix(c(4, 6, 5, 9,
                   8, 7, 9, 4,
                   7, 8, 7, 6), nrow=3, byrow=TRUE)

direction = "max"

# Definir las capacidades de los almacenes
row.signs <- rep("<=", 3)
row.rhs <- c(100,200,150)

# Definir la demanda de las f bricas
col.signs <- rep(">=", 4)
col.rhs <- c(100, 115, 80, 105)


# Resolver el problema de transporte utilizando lp.transport
resultado <- lp.transport(cost.mat = costes,
                          direction = direction,
                          row.signs = row.signs,
                          row.rhs = row.rhs,
                          col.signs = col.signs,
                          col.rhs = col.rhs)

# Mostrar los resultados
print(resultado)
print(resultado$solution)  # Matriz de transporte  ptima
```

```
> # Mostrar los resultados
> print(resultado)


Results of Linear Programming / Linear Optimization

Objective function (Minimum): 0

Iterations in phase 1: 0
Iterations in phase 2: 0
Solution
  opt
1   0
2   0
3   0
4   0

Basic Variables
    opt
S 1 100
S 2 200
S 3 150

Constraints
  actual dir bvec free dual dual.reg
1      0  <=  100  100    0      100
2      0  <=  200  200    0      200
3      0  <=  150  150    0      150

All Variables (including slack variables)
    opt cvec min.c    max.c marg marg.reg
1     0  100    99 77.00000  100  21.4286
2     0  115    99 77.00000  115  16.6667
3     0   80    99 77.00000   80  20.0000
4     0  105    99 77.00000  105  11.1111
S 1 100    0    NA 11.66667    0       NA
S 2 200    0    NA  8.88889    0       NA
S 3 150    0    NA 11.42857    0       NA

> print(resultado$solution)  # Matriz de transporte óptima
1 2 3 4
0 0 0 0
>
```

Usando linprog:

```
library(linprog)

# Definir la matriz de costes (Almacenes x F bricas)
costes <- matrix(c(4, 6, 5, 9,
                   8, 7, 9, 4,
                   7, 8, 7, 6), nrow=3, byrow=TRUE)

# Definir las capacidades de los almacenes
row.signs <- rep("<=", 3)
row.rhs <- c(100,200,150)
```

```
# Definir la demanda de las f bricas
col.signs <- rep(">=", 4)
col.rhs <- c(100, 115, 80, 105)



resultado <- solveLP(col.rhs, row.rhs, costes, maximum = FALSE, row.signs)

summary(resultado)
```

```
> summary(resultado)


Results of Linear Programming / Linear Optimization

Objective function (Minimum): 0

Solution
  opt
1   0
2   0
3   0
4   0
```

3 Problema 3

Resolver el problema (paquetes lpSolve) de transporte definido por los siguientes datos:

3	4	6	8	9	30
2	2	4	5	5	80
2	2	2	3	3	10
3	3	2	4	2	60
10	50	20	80	20	

Las filas corresponden a los lugares de origen y las columnas a los lugares de destino. Los valores de la matriz son los costes de transporte desde cada origen a cada destino. El último elemento de cada fila es la oferta de producto en cada origen. El último elemento de cada columna es la demanda total en cada destino.

```
library(lpSolve)

# Definir la matriz de costes (Almacenes x F bricas)
costes <- matrix(c(3, 4, 6, 8, 9,
                   2, 2, 4, 5, 5,
                   2, 2, 2, 3, 3,
                   3, 3, 2, 4, 2), nrow=4, byrow=TRUE)

direction = "min"
```

```r
# Definir las capacidades de los almacenes
row.signs <- rep("<=", 4)
row.rhs <- c(30, 80, 10, 60)

# Definir la demanda de las f bricas
col.signs <- rep("=", 5)
col.rhs <- c(10, 50, 20, 80, 20)


# Resolver el problema de transporte utilizando lp.transport
resultado <- lp.transport(cost.mat = costes,
                          direction = direction,
                          row.signs = row.signs,
                          row.rhs = row.rhs,
                          col.signs = col.signs,
                          col.rhs = col.rhs)

# Mostrar los resultados
print(resultado)
print(resultado$solution)  # Matriz de transporte  ptima
```

```
> # Mostrar los resultados
> print(resultado)


Results of Linear Programming / Linear Optimization

Objective function (Minimum): 0

Iterations in phase 1: 0
Iterations in phase 2: 0
Solution
  opt
1   0
2   0
3   0
4   0

Basic Variables
    opt
S 1 100
S 2 200
S 3 150

Constraints
  actual dir bvec free dual dual.reg
1      0  <=  100  100    0      100
2      0  <=  200  200    0      200
3      0  <=  150  150    0      150

All Variables (including slack variables)
    opt cvec min.c    max.c marg marg.reg
1     0  100    99 77.00000  100  21.4286
2     0  115    99 77.00000  115  16.6667
3     0   80    99 77.00000   80  20.0000
4     0  105    99 77.00000  105  11.1111
S 1 100    0    NA 11.66667    0       NA
S 2 200    0    NA  8.88889    0       NA
S 3 150    0    NA 11.42857    0       NA

> print(resultado$solution)  # Matriz de transporte óptima
1 2 3 4
0 0 0 0
```

4 Problema 4

Resuelve el problema 1 y 2 con el paquete BOOT. Se muestra un ejemplo de uso (http://reyesestadistica.blogspot.com.es/2014/09/solucion-de-de-programacion-lineal-con.html): El problema a resolver es:
Maximizar: 200x1+6000x2+3000x3-200x4

Sujeto a:
800x1+6000x2+1000x3+400x4<=13800
50x1+3x2+150x3+100x4>=600
10x1+10x2+75x3+100x4>=300

150x1+35x2+75x3+5x4>=550

Programa R:
library(boot)
enj ← c(200, 6000, 3000, -200)
fat ← c(800, 6000, 1000, 400)
vitx ← c(50, 3, 150, 100)
vity ← c(10, 10, 75, 100)
vitz ← c(150, 35, 75, 5)
simplex(a = enj, A1 = fat, b1 = 13800, A2 = rbind(vitx, vity, vitz), b2 = c(600, 300, 550), maxi = TRUE)
Y el resultado que produce es el siguiente:

```
Linear Programming Results

Call : simplex(a = enj, A1 = fat, b1 = 13800, A2 = rbind(vitx, vity,
    vitz), b2 = c(600, 300, 550), maxi = TRUE)

Maximization Problem with Objective Function Coefficients
  x1   x2   x3   x4
 200 6000 3000 -200


Optimal solution has the following values
  x1   x2   x3   x4
 0.0  0.0 13.8  0.0
The optimal value of the objective  function is 41400.
```

Problema 1:

```
library(boot)

enj <- c(20, 30)
vitx <- c(3, 6)
vity <- c(1, 0.5)
vitz <- c(1, 1)

simplex(a = enj, A1 = rbind(vitx, vity, vitz), b1 = c(150, 22, 27.5), maxi = TRUE)
```

```
Linear Programming Results

Call : simplex(a = enj, A1 = rbind(vitx, vity, vitz), b1 = c(150, 22,
    27.5), maxi = TRUE)

Maximization Problem with Objective Function Coefficients
x1 x2
20 30


Optimal solution has the following values
  x1   x2
 5.0 22.5
The optimal value of the objective  function is 775.
```

Problema 2:

```
library(boot)

enj <- c(100, 115, 80, 105)
vitx <- c(4, 6, 5, 9)
vity <- c(8, 7, 9, 4)
vitz <- c(7, 8, 7, 6)

simplex(a = enj, A1 = rbind(vitx, vity, vitz), b1 = c(100, 200, 150), maxi = TRUE)
```

```
Linear Programming Results

Call : simplex(a = enj, A1 = rbind(vitx, vity, vitz), b1 = c(100, 200,
    150), maxi = TRUE)

Maximization Problem with Objective Function Coefficients
 x1  x2  x3  x4
100 115  80 105


Optimal solution has the following values
       x1        x2        x3        x4
19.230769  0.000000  0.000000  2.564103
The optimal value of the objective  function is 2192.30769230769.
```