

OPTIMIZACIÓN EN R: INSTRUCTIVO BÁSICO

Considere el siguiente problema de optimización:

Una compañía de *call centers* cuenta con dos servidores principales, localizados en San Francisco y Houston. Dichos servidores atienden las tres áreas geográficas en la que la compañía dividió el país: Costa Este, Centro y Costa Oeste. Los servidores reciben las llamadas de los clientes y las dirigen a los operadores telefónicos para que estos las atiendan. La red de banda ancha permite que cualquier llamada de cualquier área se conecte a cualquiera de los servidores. El problema que debe resolver la dirección de sistemas de la compañía tiene que ver con la asignación de los servidores a las llamadas de las áreas geográficas. En las tablas siguientes se puede apreciar la disponibilidad de procesamiento de cada servidor (en Terabytes), y las demandas de cada cliente (en Terabytes).

Servidor	Capacidad de procesamiento [TB]
San Francisco	350
Houston	600

Área geográfica	Demanda [TB]
Costa Este	325
Centro	300
Costa Oeste	275

Por cada Terabyte que se atienda en cada servidor de cada área geográfica se incurre en un costo (relacionado con la electricidad necesaria refrigerar los equipos de cómputo y mantener la infraestructura de red), que se muestra a continuación.

Servidor / Región	Costa Este	Centro	Costa Oeste
San Francisco	225	153	162
Houston	225	162	126

Se desea, con un modelo de optimización, minimizar el costo total de despacho de las llamadas, sujeto a las condiciones anteriormente descritas.

El anterior problema es un caso particular del problema general de encontrar la solución óptima de un problema de Programación Lineal.

Recuerde que la estructura general de un problema de programación lineal es:

Max **CX**

Sujeto a

AX ≤ **b**

X ≥ **0**

En la anterior nomenclatura se tiene que:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix}$$
 es el vector de las variables de decisión.

$\mathbf{C} = [C_1 \ C_2 \ \dots \ C_n]$ es el vector de coeficientes de la función objetivo.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \text{ es la matriz de coeficientes de las restricciones.}$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \text{ es el vector de recursos (o de lados derechos de las restricciones)}$$

A continuación se usará el ejemplo para programar el problema de optimización. Considere primero su formulación matricial:

$$Ciudades = \{San Francisco (SF), Houston (H)\}$$

$$Área geográfica = \{Costa Este (CE), Centro (C), Costa Oeste (CO)\}$$

En cada declaración se siguió el orden:

$$SF-CE \quad SF-C \quad SF-CO \quad H-CE \quad H-C \quad H-CO$$

El problema es:

$$\text{Min } [225 \quad 153 \quad 162 \quad 225 \quad 162 \quad 126] \begin{bmatrix} X_{SF-CE} \\ X_{SF-C} \\ X_{SF-CO} \\ X_{H-CE} \\ X_{H-C} \\ X_{H-CO} \end{bmatrix}$$

Sujeto a

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{SF-CE} \\ X_{SF-C} \\ X_{SF-CO} \\ X_{H-CE} \\ X_{H-C} \\ X_{H-CO} \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \geq \\ \geq \\ \geq \\ \geq \end{matrix} \begin{bmatrix} 350 \\ 600 \\ 325 \\ 300 \\ 275 \end{bmatrix}$$

$$\begin{bmatrix} X_{SF-CE} \\ X_{SF-C} \\ X_{SF-CO} \\ X_{H-CE} \\ X_{H-C} \\ X_{H-CO} \end{bmatrix} \begin{matrix} \geq \\ \geq \\ \geq \\ \geq \\ \geq \\ \geq \end{matrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

La esencia de la optimización en R es declarar cada una de las matrices y vectores anteriores, y luego ingresarlas a una función que correrá la optimización.

La función `lp()` de la librería `lpSolve` recibe los siguientes argumentos para correr la optimización:

- `direction`: dirección de la optimización. Debe escribirse "max" si se va a maximizar o "min" si se va a minimizar.
- `objective.in`: vector **C** de coeficientes de las variables de decisión en la función objetivo.

- `const.mat`: matriz **A** de coeficientes de las variables de decisión en las restricciones.
- `const.dir`: vector de strings en los que se indicará el signo de las restricciones. Cada elemento del vector debe ser "<=", "=", o ">=".
- `const.rhs`: vector **b** de recursos o de lados derechos de las restricciones.
- `transpose.constraints=TRUE`: argumento por defecto que indica que la matriz **A** será traspuesta dentro del código de optimización. Deje esta opción en `TRUE`.
- `int.vec`: vector que indicará el índice de las variables que deben ser enteras. Si, por ejemplo, las variables en la posición 3 y 5 deben ser enteras, se debe ingresar `c(3, 5)`.
- `binary.vec`: vector que indicará el índice de las variables que deben ser binarias. Si, por ejemplo, las variables en la posición 2 y 4 deben ser enteras, se debe ingresar `c(2, 4)`.
- `all.int`: escriba `TRUE` si todas las variables de su problema deben ser enteras. En ese caso no debe ingresar nada en el parámetro `int.vec`. En caso contrario, escriba `FALSE`.
- `all.bin`: escriba `TRUE` si todas las variables de su problema deben ser binarias. En ese caso no debe ingresar nada en el parámetro `binary.vec`. En caso contrario, escriba `FALSE`.

Para usar la función `lp()`, siga los pasos que se describen a continuación y que se ejemplifican con el problema inicial de este documento:

PASO 0. Instalar la librería lpSolve

La función `lp()` no pertenece al conjunto de funciones estándar de R. Por lo anterior se hace necesario instalar el paquete `lpSolve`. Realice este paso una sola vez. La librería quedará instalada en su equipo y no es necesario que la instale para correr futuros programas.

```
install.packages("lpSolve")
```

PASO 1. Cargar la librería lpSolve

Como disciplina general, cargue primero la librería que va a utilizar antes de programar otras líneas de código.

```
library(lpSolve)
```

PASO 2. Declare las matrices y vectores que definen el problema de optimización

Tenga presente que en cada una de las declaraciones deberá seguirse el mismo orden de las variables. Esto debido a la forma matricial en el R leerá el problema. En cada declaración se siguió el orden:

SF-CE SF-C SF-CO H-CE H-C H-CO

```
#Vector C
```

```
C=c(225,153,162,225,162,126)
```

```
[1] 225 153 162 225 162 126
```

```
#Matriz A
```

```
A=matrix(c(1,1,1,0,0,0,0,0,0,1,1,1,1,0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,1),
nrow=5,ncol=6,byrow=TRUE)
```

```

      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]    0    0    0    1    1    1
[3,]    1    0    0    1    0    0
[4,]    0    1    0    0    1    0
[5,]    0    0    1    0    0    1

```

```
#Vector b
```

```
b=c(350,600,325,300,275)
```

```
[1] 350 600 325 300 275
```

```
#Vector dir
```

```
dir=c("<=", "<=", ">=", ">=", ">=")
```

```
[1] "<=" "<=" ">=" ">=" ">="
```

PASO 3. Ingrese los argumentos anteriores a la función `lp()` . Asigne los resultados de esta función a un objeto.

```
Opt<-lp(direction="min", objective.in=C, const.mat=A, const.dir=dir,
const.rhs=b, transpose.constraints = TRUE, all.int=FALSE, all.bin=FALSE)
```

Note que en ningún argumento se tuvo que especificar que las variables son mayores o iguales a cero. La función `lp()` asume esta restricción por defecto.

PASO 4. Extraiga del objeto los valores de las variables de decisión y de la función objetivo.

```
Opt$solution
```

```
[1] 50 300 0 275 0 275
```

```
Opt$objval
```

```
[1] 153675
```

Puede visualizar otros elementos que puede extraer usando la función `attributes(Opt)`.

Teniendo en cuenta el orden en el que se definieron las variables de decisión:

SF-CE SF-C SF-CO H-CE H-C H-CO

La solución óptima del problema indica que:

$$X_{SF-CE} = 50 \quad X_{SF-C} = 300 \quad X_{SF-CO} = 0 \quad X_{H-CE} = 275 \quad X_{H-C} = 0 \quad X_{H-CO} = 275$$

El costo de esta asignación óptima es de \$153675.

Importar los datos desde archivos csv

Recuerde que en R se tiene la posibilidad de importar datos desde archivos csv. Al leerlos, R guarda los datos en dataframes. Es posible convertir los dataframes en matrices o extraer filas o columnas de los dataframes y convertirlas en vectores. Los siguientes comandos pueden serle útil para importar y transformar datos provenientes de archivos csv:

- Leer datos: `read.csv("Ruta del archivo csv que contiene datos")`
 - Ejemplo:
`datos=read.csv("C:/Users/Invitado/Documentos/archivo.csv")`
El objeto `datos` es el dataframe en el que R almacenará la información contenida en el archivo csv.
- Convertir un dataframe en una matriz: `data.matrix(Dataframe que se convertirá en matriz)`
 - Ejemplo: `matriz=data.matrix(datos)`
El objeto `datos` es el dataframe. El objeto `matriz` es la matriz resultante.
- Extraer una fila de una matriz: `matriz[índice de la fila,]`
 - Ejemplo: `vector=matriz[1,]`
El objeto `vector` es el resultante de extraer la fila 1 de `matriz`.
- Extraer una columna de una matriz: `matriz[,índice de la columna]`
 - Ejemplo: `vector=matriz[,1]`
El objeto `vector` es el resultante de extraer la columna 1 de `matriz`.

Más información:

Puede obtener más detalles sobre el funcionamiento de la función `lp()` usando el comando `help(lp)` o consultando material en la web.