Transportation Problem LP

Melissa Paniagua

10/12/2020

Assignment 3 | Module 6

The purpose if this assignment is to solve a transportation problem, a special type of linear programming, utilizing R.

Heart Start Company

Heart Start produces automated external defibrillators (AEDs) in each of two different plants (A and B). The unit production costs and monthly production capacity of the two plants are indicated in the table below. The AEDs are sold through three wholesalers. The shipping cost from each plant to the warehouse of each wholesaler along with the monthly demand from each wholesaler are also indicated in the table.

How many AEDs should be produced in each plant, and how should they be distributed to each of the three wholesaler warehouses so as to minimize the combined cost of production and shipping?

Transportation Problem Formulation.

Let

X_{ij} = number of units produced on plant i and shipped to warehouse j, where i = A (Plant A), B (Plant B), and warehouse $j = 1$ (Warehouse 1), 2 (Warehouse 2), and 3 (Warehouse 3).

Minimize the combined cost of production and shipping

$Z = 622X_{A1} + 614X_{A2} + 630X_{A3} + 641X_{B1} + 645X_{B2} + 649X_{B3}$

Constraints:

Monthly production capacity per plant:

Plant A $= X_{A1} + X_{A2} + X_{A3} \leq 100$

Plant B $= X_{B1} + X_{B2} + X_{B3} \leq 120$

Monthly demand of units per warehouse:

Warehouse 1 $= X_{A1} + X_{B1} = 80$

Warehouse 2 $= X_{A2} + X_{B2} = 60$

Warehouse 3 $= X_{A3} + X_{B3} = 70$

All values must be greater or equal to zero

$X_{ij} \geq 0$

Solving this transportation problem utilizing R.

Add the LP objective function and constrains

To accomplish it, we will use the “lpSolveAPI” package in R.

```
# Import the lpSolve package.
library(lpSolveAPI)

# Create an lp object named lppoint with 0 constraints and 6 decision variables.
lppoint <- make.lp(0, 6)

# Set the objective function. Minimization problem is the default, so we do not
# have to change the direction.
set.objfn(lppoint, c(622, 614, 630,
                     641, 645, 649))

# Add the 9 constraints based on the number of units made on each plant
# and shipped to differente warehouses.

add.constraint(lppoint, c(1, 1, 1, 0, 0, 0), "<=", 100)
add.constraint(lppoint, c(0, 0, 0, 1, 1, 1), "<=", 120)

add.constraint(lppoint, c(1, 0, 0, 1, 0, 0), "=", 80)
add.constraint(lppoint, c(0, 1, 0, 0, 1, 0), "=", 60)
add.constraint(lppoint, c(0, 0, 1, 0, 0, 1), "=", 70)
```

Set bounds for variables.

Remember that all variables had to be non-negative. We don’t need to do it here, as that is the default, but we can set bounds explicitly.

```
# Set bounds for variables.
set.bounds(lppoint, lower = c(0, 0, 0, 0, 0, 0),
           columns = c(1:6)) #Not really needed

# To identify the variables and constraints, we can set variable names and name the constraints
RowNames <- c("MonthPrdCapPlantA", "MonthPrdCapPlantB",
              "MonthDemWH1", "MonthDemWH2", "MonthDemWH3")

ColNames <- c("PlantA_WH1", "PlantA_WH2", "PlantA_WH3",
              "PlantB_WH1", "PlantB_WH2", "PlantB_WH3")

dimnames(lppoint) <- list(RowNames, ColNames)

# Now, print out the model
lppoint
```

```
Model name:
                  PlantA_WH1  PlantA_WH2  PlantA_WH3  PlantB_WH1  PlantB_WH2  PlantB_WH3
Minimize                 622         614         630         641         645         649
MonthPrdCapPlantA          1           1           1           0           0           0  <=  100
MonthPrdCapPlantB          0           0           0           1           1           1  <=  120
MonthDemWH1                1           0           0           1           0           0   =   80
```

```
MonthDemWH2            0        1        0        0        1        0   =   60
MonthDemWH3            0        0        1        0        0        1   =   70
Kind                 Std      Std      Std      Std      Std      Std
Type                Real     Real     Real     Real     Real     Real
Upper                Inf      Inf      Inf      Inf      Inf      Inf
Lower                  0        0        0        0        0        0
```

Solve the LP problem structured above

```r
solve(lpproint)
```

```
[1] 0
```

As we can see, the result is "0," which means it was succefully solved.

The next step is to output the values

```r
# It will give us the minimum value for the optimal solution.
get.objective(lppoint)
```

```
[1] 132790
```

For our problem, $132, 790 is the minimum combined cost of production and shipping founded for the optimal solution.

```r
# It will give us the optimal number of units produced on each plant and shipped to each warehouse.
get.variables(lppoint)
```

```
[1]  0 60 40 80  0 30
```

In order to get the minimum combined cost of production and shipping, each plant should produce and ship the following amounts:

Plant A should produce 100 units. The company should ship 60 units of plant A to warehouse 2 and 40 units to warehouse 3.

Plant B should produce 110 units. The company should ship 80 units of plant B to warehouse 1 and 30 units to warehouse 3.

```r
# It will give us the optimal constrains
get.constraints(lppoint)
```

```
[1] 100 110  80  60  70
```

This output is essential it show us the correct values in order to satisfy the equality assumption. The equality assumption in a transportation problem says that the problem will have feasible solutions if and only if the total production = total demand.

So, the constrain on plant B must be reduced from 120 to 110 to have a feasible solution.

```r
# It will give us the reduced cost
get.sensitivity.objex(lppoint)
```

```
$objfrom
[1] -1.00e+30 -1.00e+30  6.18e+02 -1.00e+30  6.33e+02  6.49e+02

$objtill
[1] 6.22e+02 6.26e+02 6.30e+02 6.41e+02 1.00e+30 6.61e+02

$objfromvalue
[1]  4e+01 -1e+30 -1e+30 -1e+30  3e+01 -1e+30

$objtillvalue
[1] NA NA NA NA NA NA
```

```r
# It will give us the shadow prices
get.sensitivity.rhs(lppoint)
```

```
$duals
 [1] -19   0 641 633 649   0   0   0   0  12   0

$dualsfrom
 [1]  9e+01 -1e+30  0e+00  3e+01  4e+01 -3e+01 -1e+30 -1e+30 -1e+30 -4e+01
[11] -1e+30

$dualstill
 [1] 1.3e+02 1.0e+30 9.0e+01 7.0e+01 8.0e+01 4.0e+01 1.0e+30 1.0e+30 1.0e+30
[10] 3.0e+01 1.0e+30
```

Utilizing Dummy Variables Approach

```r
# Create an lp object named lppoint with 0 constraints and 8 decision variables.
lppoint_dummy <- make.lp(0, 8)

# Set the objective function. Minimization problem is the default, so we do not
# have to change the direction.
set.objfn(lppoint_dummy, c(622, 614, 630, 0,
                           641, 645, 649, 0))
```

Add the 8 constraints based on the number of units made and shipped. PLUS, the dummy variable needed, which will be a dummy warehouse 4, in order to satisfy the equallity assumplution total production = total demand

```r
add.constraint(lppoint_dummy, c(1, 1, 1, 1, 0, 0, 0, 0), "=", 100)
add.constraint(lppoint_dummy, c(0, 0, 0, 0, 1, 1, 1, 1), "=", 120)

add.constraint(lppoint_dummy, c(1, 0, 0, 0, 1, 0, 0, 0), "=", 80)
add.constraint(lppoint_dummy, c(0, 1, 0, 0, 0, 1, 0, 0), "=", 60)
add.constraint(lppoint_dummy, c(0, 0, 1, 0, 0, 0, 1, 0), "=", 70)
add.constraint(lppoint_dummy, c(0, 0, 0, 1, 0, 0, 0, 1), "=", 10) #Dummy Variable

# Set bounds for variables.
set.bounds(lppoint_dummy, lower = c(0, 0, 0, 0, 0, 0, 0, 0),
           columns = c(1:8)) #Not really needed

# To identify the variables and constraints, we can set variable names and name the constraints
RowNames_dummy <- c("MthPrdCapPltA", "MthPrdCapPltB",
              "MthDemWH1", "MthDemWH2", "MthDemWH3", "MthDemWH4")

ColNames_dummy <- c("PltA_WH1", "PltA_WH2", "PltA_WH3", "PltA_WH4",
              "PltB_WH1", "PltB_WH2", "PltB_WH3", "PltB_WH4")

dimnames(lppoint_dummy) <- list(RowNames_dummy, ColNames_dummy)

# Now, print out the model
lppoint_dummy
```

```
Model name:
               PltA_WH1  PltA_WH2  PltA_WH3  PltA_WH4  PltB_WH1  PltB_WH2  PltB_WH3  PltB_WH4
Minimize            622       614       630         0       641       645       649         0
MthPrdCapPltA         1         1         1         1         0         0         0         0  =  100
MthPrdCapPltB         0         0         0         0         1         1         1         1  =  120
MthDemWH1             1         0         0         0         1         0         0         0  =   80
MthDemWH2             0         1         0         0         0         1         0         0  =   60
MthDemWH3             0         0         1         0         0         0         1         0  =   70
MthDemWH4             0         0         0         1         0         0         0         1  =   10
Kind                Std       Std       Std       Std       Std       Std       Std       Std
Type               Real      Real      Real      Real      Real      Real      Real      Real
Upper               Inf       Inf       Inf       Inf       Inf       Inf       Inf       Inf
Lower                 0         0         0         0         0         0         0         0
```

```r
# Solve the transportation problem
solve(lppoint_dummy)
```

```
[1] 0
```

Answer is “0,” so it is correct.

```r
# It will give us the minimum value for the optimal solution.
get.objective(lppoint_dummy)
```

```
[1] 132790
```

Same optimal solution as the previous approach.

```r
# It will give us the optimal number of units produced on each plant and shipped to each warehouse.
get.variables(lppoint_dummy)
```

```
[1]  0 60 40  0 80  0 30 10
```

Here, the additional 10 units are logically sent to warehouse 4, which is needed to satisfy the equallity assumption. Warehouse 4 is not physical location, but it means 10 units must be shipped to any warehouse in order to get a feasible solution.

```r
# It will give us the optimal constrains
get.constraints(lppoint_dummy)
```

```
[1] 100 120  80  60  70  10
```

All the constrains are satisfied.

```r
# It will give us the reduced cost
get.sensitivity.objex(lppoint_dummy)
```

```
$objfrom
[1]  6.22e+02 -1.00e+30  6.18e+02 -1.90e+01 -1.00e+30  6.33e+02  6.49e+02
[8] -1.00e+30

$objtill
[1] 1.00e+30 6.26e+02 6.30e+02 1.00e+30 6.41e+02 1.00e+30 6.61e+02 1.90e+01

$objfromvalue
[1]  4e+01 -1e+30 -1e+30  1e+01 -1e+30  3e+01 -1e+30 -1e+30

$objtillvalue
[1] NA NA NA NA NA NA NA NA
```

```r
# It will give us the shadow prices
get.sensitivity.rhs(lppoint_dummy)
```

```
$duals
 [1]  614  633    8    0   16 -633    0    0    0   19    0   12    0    0

$dualsfrom
 [1]  1.0e+02  1.2e+02  8.0e+01 -1.0e+30  7.0e+01  1.0e+01 -3.0e+01 -1.0e+30
 [9] -1.0e+30 -3.0e+01 -1.0e+30 -4.0e+01 -1.0e+30 -1.0e+30

$dualstill
 [1] 1.0e+02 1.2e+02 8.0e+01 1.0e+30 7.0e+01 1.0e+01 4.0e+01 1.0e+30 1.0e+30
[10] 1.0e+01 1.0e+30 3.0e+01 1.0e+30 1.0e+30
```