# Assignment_2_QMM

Abhishek_Sau

29/09/2021

## R Markdown

**Question 3 - (Weigelt Production) The Weigelt Corporation has three branch plants with excess production capacity. Fortunately, the corporation has a new product ready to begin production, and all three plants have this capability, so some of the excess capacity can be used in this way. This product can be made in three sizes–large, medium, and small–that yield a net unit profit of $420, $360, and $300, respectively. Plants 1, 2, and 3 have the excess capacity to produce 750, 900, and 450 units per day of this product, respectively, regardless of the size or combination of sizes involved. The amount of available in-process storage space also imposes a limitation on the production rates of the new product. Plants 1, 2, and 3 have 13,000, 12,000, and 5,000 square feet, respectively, of in-process storage space available for a day's production of this product. Each unit of the large, medium, and small sizes produced per day requires 20, 15, and 12 square feet, respectively. Sales forecasts indicate that if available, 900, 1,200, and 750 units of the large, medium, and small sizes, respectively, would be sold per day. At each plant, some employees will need to be laid off unless most of the plant's excess production capacity can be used to produce the new product. To avoid layoffs if possible, management has decided that the plants should use the same percentage of their excess capacity to produce the new product. Management wishes to know how much of each of the sizes should be produced by each of the plants to maximize profit.**

a. Define the decision variables
b. Formulate a linear programming model for this problem.
c. Solve the problem using lpsolve, or any other equivalent library in R.

```
#install.packages("lpSolveAPI")
```

Loading the library

```
library(lpSolveAPI)

setwd("D:\\Study\\Assignments\\DMM\\Assignment 2")
```

#Answer 1 - Decision Variables- Let xij be the number of units of size j (j=Large , Medium, Small) product produced in plant i (i=1,2,3)

hence, the decision variables are x1l,x1m,x1s,x2l,x2m,x2s,x3l,x3m and x3s

```
# make an lp object with 0 constraints and 9 decision variables
lprec <- make.lp(0, 9)
```

```r
lprec
```

```
## Model name:
##   a linear program with 9 decision variables and 0 constraints
```

```r
# Create the objective function and since we need to maximize profit, change the sense to max.
set.objfn(lprec, c(420, 360, 300, 420, 360, 300, 420, 360, 300))
lp.control(lprec,sense='max')
```

```
## $anti.degen
## [1] "none"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##       epsb       epsd      epsel     epsint epsperturb   epspivot
##      1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
```

```
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```r
# Add the constraints

#Capacity Constraints
add.constraint(lprec, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 750)
add.constraint(lprec, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 900)
add.constraint(lprec, c(0, 0, 0, 0, 0, 0,1, 1, 1), "<=", 450)

#Storage COnstraints
add.constraint(lprec, c(20, 15, 12, 0, 0, 0, 0, 0, 0), "<=", 13000)
add.constraint(lprec, c(0, 0, 0, 20, 15, 12, 0, 0, 0), "<=", 12000)
add.constraint(lprec, c(0, 0, 0, 0, 0, 0, 20, 15, 12), "<=", 5000)

#Sales forecast constraints
add.constraint(lprec, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 900)
add.constraint(lprec, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 1200)
add.constraint(lprec, c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=", 750)

#Same Capacity % constraints
add.constraint(lprec, c(6, 6, 6, -5, -5, -5, 0, 0, 0), "=", 0)
add.constraint(lprec, c( 3, 3, 3, 0, 0, 0, -5, -5, -5), "=", 0)

#set.bounds(lprec, lower = c(0, 0, 0, 0, 0, 0, 0, 0, 0), columns = c(1, 2,3,4,5,6,7,8,9))

# To identify the variables and constraints, we can set variable names and name the constraints
RowNames <- c("CapCon1", "CapCon2", "CapCon3", "StoCon1", "StoCon2", "StoCon3", "SalCon1", "SalCon2", "
ColNames <- c("P1Large", "P1Medium", "P1Small", "P2Large", "P2Medium", "P2Small", "P3Large", "P3Medium"
dimnames(lprec) <- list(RowNames, ColNames)

lprec
```

```
## Model name:
##   a linear program with 9 decision variables and 11 constraints
```

```r
#Answer 3 = LP file created
#writing ot LP file
write.lp(lprec, filename = "A2QMM.lp", type = "lp")
```

```r
solve(lprec)
```

```
## [1] 0
```

```r
get.objective(lprec)
```

```
## [1] 696000
```

```r
get.variables(lprec)
```

```
## [1] 516.6667 177.7778   0.0000   0.0000 666.6667 166.6667   0.0000   0.0000
## [9] 416.6667
```

Answer - Maximum profit (z) = 696000$