Assignment2QT#3

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

```
library(lpSolveAPI)
```

Let us set up the problem. Note that we have two decision variables, and three constraints. In the first formulation, we will directly create the objective function and constraints

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

```
## Warning: package 'lpSolveAPI' is in use and will not be installed
```

```
library(lpSolveAPI)
```

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

# Now create the objective function. The default is a minimization problem.
set.objfn(lprec, c(420, 360, 300, 420, 360, 300, 420, 360,300 ))
```

```
library(lpSolveAPI)
# As the default is a minimization problem, we change the direction to set maximization
lp.control(lprec,sense='max')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $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"
```

```
# Add the three Squared Footage constraints
# Plant production by day
```

```
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)

#Land storage for products by day
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 for products by day
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)

# plants with the same % capacity
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)

RN<-c("Ccon1","Ccon2","Ccon3","SCon1", "SCon2","SCon3","saCon1","saCon2","saCon3","%C1","%C2")
CN<-c("P1L","P1M","P1S","P2L","P2M","P2S","P3L","P3M","P3S")
dimnames(lprec)<-list(RN, CN)
lprec
```

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

We now solve the above LP problem

```
‘‘‘r
solve(lprec)
```

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

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

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

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

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