

Assignment2QT#1

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

```
library(lpSolveAPI)
```

Let us set up the problem. Note that we had 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)
```

```
# make an lp object with 0 constraints and 2 decision variables
```

```
lprec <- make.lp(0, 2)
```

```
# Now create the objective function. The default is a minimization problem.
```

```
set.objfn(lprec, c(32, 24))
```

```
# 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 Plant constraints
add.constraint(lprec, c(3, 2), "<=", 5000)
add.constraint(lprec, c(45, 4), "<=", 84000)

```

```

# Set bounds for variables.
# that all variables had to be non-negative. We don't need to do it here, as that is the default,
# we can set bounds explicitly
set.bounds(lprec, lower=c(0, 0), columns=c(1, 2)) #Not really needed

# To identify the variables and constraints, we can
# set variable names and name the constraints
RowNames<-c("Plant1", "Plant2")
ColNames<-c("Product1", "Product2")
dimnames(lprec)<-list(RowNames, ColNames)

# Now, print out the model
lprec

```

```

## Model name:
##           Product1  Product2
## Maximize      32      24
## Plant1         3       2  <=   5000
## Plant2        45       4  <=  84000
## Kind          Std      Std
## Type          Real     Real
## Upper         Inf      Inf
## Lower         0        0

```

We now solve the above LP problem

```
solve(lprec)
```

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

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

```
## [1] 60000
```

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

```
## [1] 0 2500
```