mbruner3_mod4

Mark Bruner

9/21/2020

```r
library(lpSolveAPI)
```

Defining decision variables and objective function

```r
lprec <- make.lp(0, 9)
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"
```

```
set.objfn(lprec, c(420, 360, 300, 420, 360, 300, 420, 360, 300))
```

set constraints

Capacity Constraints

```
add.constraint(lprec, c(rep(1, 3)), indices = c(1, 2, 3), "<=", 750)
add.constraint(lprec, c(rep(1, 3)), indices = c(4, 5, 6), "<=", 900)
add.constraint(lprec, c(rep(1, 3)), indices = c(7, 8, 9), "<=", 450)
```

Square Footage

```
add.constraint(lprec, c(20, 15,12), indices = c(1, 2, 3), "<=", 13000)
add.constraint(lprec, c(20, 15, 12), indices = c(4, 5, 6), "<=", 12000)
add.constraint(lprec, c(20, 15, 12), indices = c(7, 8, 9), "<=", 5000)
```

Sales

```
add.constraint(lprec, c(rep(1, 3)), indices = c(1, 4, 7), "<=", 900)
add.constraint(lprec, c(rep(1, 3)), indices = c(2, 5, 8), "<=", 1200)
add.constraint(lprec, c(rep(1, 3)), indices = c(3, 6, 9), "<=", 750)
```

Same percentage of capacity

```
add.constraint(lprec, c(rep(900, 3), rep(-750, 3)), indices = c(1, 2, 3, 4, 5, 6), "=", 0)
add.constraint(lprec, c(rep(450, 3), rep(-750, 3)), indices = c(1, 2, 3, 7, 8, 9), "=", 0)

set.bounds(lprec, lower = c(rep(0, 9)), columns = c(1:9))
```

Decision Variable Names

```
RowNames <- c("Capacity 1", "Capacity 2", "Capacity 3", "Sqft1", "Sqft2", "Sqft3", "Sales 1", "Sales 2"
ColNames <- c("L1","M1","S1", "L2", "M2", "S2", "L3", "M3", "S3")
dimnames(lprec) <- list(RowNames, ColNames)
lprec
```

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

Solve LP model

```
solve(lprec)
```

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

Optimize Objective Function

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

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

Decision Variables

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

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

Interpreting the output from optimization routines

In order to satisfy the constraints on the LP Model, the Weigelt Corporation should produce the following quantities of L, M, and S at each factory:

Factory 1 (space = 13000 sq ft and capacity is 750 units.) L1 = 516.6667 M1 = 177.7778
S1 = 0.00000

Factory 2 (space = 12000 sq ft and capacity is 900 units.) L2 = 0.00000 M2 = 666.6667 S2 = 166.6667

Factory 3 (space = 5000 sq ft and capacity is 450 units.) L3 = 0 M3 = 0 S3 = 416.6667

Interpreting optimal objective function value:

If the above production occurs at each factory, optimal amount of money they can expect in profit per day. ~$696,000