QMM-ASSIGNMENT1_64018

Yeswanth Siripurapu

2023-09-24

#Summary: #a) A maximum revenue of $1780 can be achieved by making 40 artisanal truffles, #12 handmade chocolate nuggets, and 4 chocolate bars. #b) Chocolate bars, handmade chocolate nuggets, and artisanal truffles #constrain binding. #c)In terms of fliesability, artisanal truffles have a shadow price of $2 and a #range of 47.5 to 51.6 pounds. #c) Made-to-Order Chocolate Nuggets: Shadow Price = $30, #Range of Fiesability = 30 to 52 Pounds e) #Chocolate Bars: Shadow Price = $6, Range of Fiesability = 29.1 to 50 Pounds #d) Range of Optimality: Artisanal Truffles = $20 to $38, #Handmade Chocolate Nuggets = $22.5 to $26.67 #and Chocolate Bars = $18.75 to $35.00

```r
#Load lpSolveAPI
library(lpSolveAPI)



# make an lp object with 0 constraints and 3 decision variables
lprec <- make.lp(0, 3)

# Now create the objective function. The default is a minimization problem.
set.objfn(lprec, c(35, 25, 20))

# 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"
```

```r
# Add the four constraints
add.constraint(lprec, c(1, 1/2, 1), "<=", 50)
add.constraint(lprec, c(1, 2/3, 1/2), "<=", 50)
add.constraint(lprec, c(1/2, 2/3, 1/2), "<=", 30)
add.constraint(lprec, c(0, 1, 0), ">=", 10)


# Set bounds for variables.
set.bounds(lprec, lower = c(0, 0, 0), columns = c(1, 2, 3))


# To identify the variables and constraints, we can
# set variable names and name the constraints
RowNames <- c("CacaoButter", "Honey", "DiaryCream", "NUggetsOrder")
ColNames <- c("AritisanTruffel", "ChocalateNuggets", "ChocalateBars")
dimnames(lprec) <- list(RowNames, ColNames)


# Now, print out the model
lprec
```

```
## Model name:
##                 AritisanTruffel  ChocalateNuggets     ChocalateBars
## Maximize                     35                25                20
## CacaoButter                   1               0.5                 1  <=  50
## Honey                         1    0.666666666667               0.5  <=  50
## DiaryCream                  0.5    0.666666666667               0.5  <=  30
## NUggetsOrder                  0                 1                 0  >=  10
## Kind                        Std               Std               Std
## Type                       Real              Real              Real
## Upper                       Inf               Inf               Inf
## Lower                         0                 0                 0
```

```r
#The model can also be saved to a file"
write.lp(lprec, filename = "chocalte.lp", type = "lp")
#Here we are solving now the Above LP Problem

solve(lprec)
```

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

```r
#The result above shows that there was a successful solution rather than that
#the answer is 0. We now present the objective function's value as well as the
#variables.
get.objective(lprec)
```

```
## [1] 1780
```

```r
varV <- get.variables(lprec)
#The revenue is 1780, the first variable's value is 40, and the second variable

#variable name next to the answer, reading the output has an issue. The order in
#which the variables occur in the LP formulation determines the order in which
#they are output.
#For us, it was handcrafted chocolate nuggets, artisan truffles, and ultimately
```

```
#chocolate bars. Before examining other output values, think about using a
#different method to input the problem formuation. We will construct a text file
#containing the problem formulation in the LP format. The write.lp statement was
#used to create an lp file as well. Let's now examine the choclate.lp file.
#A file can be double-clicked.


y <- read.lp("chocalte.lp") # create an lp object y
y # display y
```

```
## Model name:
##                 AritisanTruffel  ChocalateNuggets     ChocalateBars
## Maximize                     35                25                20
## CacaoButter                    1               0.5                 1  <=  50
## Honey                        1    0.666666666667               0.5  <=  50
## DiaryCream                 0.5    0.666666666667               0.5  <=  30
## NUggetsOrder                 0                 1                 0  >=  10
## Kind                       Std               Std               Std
## Type                      Real              Real              Real
## Upper                      Inf               Inf               Inf
## Lower                        0                 0                 0
```

```
solve(y)
```

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

```
get.objective(y) # get objective value
```

```
## [1] 1780
```

```
get.variables(y) # get values of decision variables
```

```
## [1] 40 12  4
```

```
get.constraints(y) # get constraint RHS values
```

```
## [1] 50 50 30 12
```

```
#2.Report the shadow price and the range of feasibility of each binding
#constraint.
get.sensitivity.rhs(lprec) # get shadow prices
```

```
## $duals
## [1]  2 30  6  0  0  0  0
##
## $dualsfrom
## [1]  4.750000e+01  3.000000e+01  2.916667e+01 -1.000000e+30 -1.000000e+30
## [6] -1.000000e+30 -1.000000e+30
##
## $dualstill
## [1] 5.166667e+01 5.200000e+01 5.000000e+01 1.000000e+30 1.000000e+30
## [6] 1.000000e+30 1.000000e+30
```

```r
get.sensitivity.obj(lprec) # get reduced cost
```

```
## $objfrom
## [1] 20.00 22.50 18.75
## 
## $objtill
## [1] 38.00000 26.66667 35.00000
```

```r
#3.If the local store increases the daily order to 25 pounds of chocolatenuggets
#how much of each product should Francesco make?


#Make an lp object with 0 constraints and 3 decision variables
lprec <- make.lp(0, 3)

#Now create the objective function. The default is a minimization problem.
set.objfn(lprec, c(35, 25, 20))

#As the default is a minimization problem, we change the direction to set
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 four constraintsUpdatedconstraintsfrom 10to25 Poundsforfourthconstraint

add.constraint(lprec, c(1, 1/2, 1), "<=", 50)
add.constraint(lprec, c(1, 2/3, 1/2), "<=", 50)
add.constraint(lprec, c(1/2, 2/3, 1/2), "<=", 30)
add.constraint(lprec, c(0, 1, 0), ">=", 25)

#Set bounds for variables.
set.bounds(lprec, lower = c(0, 0, 0), columns = c(1, 2, 3))


#To identify the variables and constraints, we can set variable names and name
#constraints
RowNames <- c("CacaoButter", "Honey", "DiaryCream", "NUggetsOrder")
ColNames <- c("AritisanTruffel", "ChocalateNuggets", "ChocalateBars")
dimnames(lprec) <- list(RowNames, ColNames)
```

```
# Now, print out the model
lprec
```

```
## Model name:
##                AritisanTruffel  ChocalateNuggets     ChocalateBars
## Maximize                    35                25                20
## CacaoButter                  1               0.5                 1  <=  50
## Honey                        1    0.666666666667               0.5  <=  50
## DiaryCream                 0.5    0.666666666667               0.5  <=  30
## NUggetsOrder                 0                 1                 0  >=  25
## Kind                       Std               Std               Std
## Type                      Real              Real              Real
## Upper                      Inf               Inf               Inf
## Lower                        0                 0                 0
```

```
write.lp(lprec, filename = "chocalte.lp", type = "lp")

#To identify the variables and constraints, we can set variable names and name
#constraints

solve(lprec)
```

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

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

```
## [1] 1558.333
```

```
y <- read.lp("chocalte.lp") # create an lp object y
y # display y
```

```
## Model name:
##                AritisanTruffel  ChocalateNuggets     ChocalateBars
## Maximize                    35                25                20
## CacaoButter                  1               0.5                 1  <=  50
## Honey                        1    0.666666666667               0.5  <=  50
## DiaryCream                 0.5    0.666666666667               0.5  <=  30
## NUggetsOrder                 0                 1                 0  >=  25
## Kind                       Std               Std               Std
## Type                      Real              Real              Real
## Upper                      Inf               Inf               Inf
## Lower                        0                 0                 0
```

```
solve(lprec)
```

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

```
get.objective(lprec) # get objective value
```

```
## [1] 1558.333
```

```r
get.variables(lprec) # get values of decision variables
```

```
## [1] 26.66667 25.00000  0.00000
```

```r
get.constraints(lprec) # get constraint RHS values
```

```
## [1] 39.16667 43.33333 30.00000 25.00000
```