ISHIKA NISHA

11/22/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
library(lpSolveAPI)
```

#Question 1

```
#find longest path

#Objective Function
# Max: 5x12+ 3x13+ 4x24+ 2x25+ 3x35+ 1x46+ 4x47+ 6x57+ 2x58+ 5x69+ 4x79+ 7x89
# Constraints:
# NODE1 5x12 + 3x13 = 1
# NODE2 5x12 - 4x24 - 2x25 = 0
# NODE3 3x13 - 3x35 = 0
# NODE4 4x24 - 1x46 - 4x47 = 0
# NODE5 2x25 + 3x25 - 6x57 - 2x58 = 0
# NODE6 1x46 - 5x69 = 0
# NODE7 4x47 + 6x57 - 4x79 = 0
# NODE8 2x58 - 7x89 = 0
# NODE9 5x69 + 4x79 + 7x89 = 1
#binx12,x13,x25,x24,x35,x47,x46,x58,x57,x79,x89

Assign1<- make.lp(0,12)

set.objfn(Assign1, c(5,3,4,2,3,1,4,6,2,5,4,7))

lp.control(Assign1, 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.constraint(Assign1, c(1,1,0,0,0,0,0,0,0,0,0,0), "=",1)
add.constraint(Assign1, c(1,0,-1,-1,0,0,0,0,0,0,0,0), "=",0)
add.constraint(Assign1, c(0,1,0,0,-1,0,0,0,0,0,0,0), "=",0)
add.constraint(Assign1, c(0,0,1,0,0,-1,-1,0,0,0,0,0), "=",0)
add.constraint(Assign1, c(0,0,0,1,1,0,0,-1,-1,0,0,0), "=",0)
add.constraint(Assign1, c(0,0,0,0,0,1,0,0,0,-1,0,0), "=",0)
add.constraint(Assign1, c(0,0,0,0,0,0,0,1,1,0,-1,0), "=",0)
add.constraint(Assign1, c(0,0,0,0,0,0,0,0,1,0,0,-1), "=",0)
add.constraint(Assign1, c(0,0,0,0,0,0,0,0,0,1,1,1), "=",1)

set.bounds(Assign1, lower = c(0,0,0,0,0,0,0,0,0,0,0,0), columns = 1:12)

solve(Assign1)

## [1] 0

get.objective(Assign1)

## [1] 17

get.variables(Assign1)

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

#longest path is x12 -> x25 -> x57 -> x79

2. (a) Let the dividend be D, growth rate be G and price per share be P

Return of stocks= (D/P) + G

So, return of stocks for S1= [[2(1+0.05)/40] + 0.05] *100 = 10.25

S2 = [[1.5(1+0.1)/50] + 0.10] *100 = 13.3

S3 = [[3.5(1+0.03)/80] + 0.03] *100 = 7.51

H1 = [[3(1+0.04)/60] + 0.04] *100 = 9.2

H2 = [[2(1+0.07)/45] + 0.07] *100 = 11.76

H3 = [[1(1+0.15)/60 + 0.15] *100 = 16.92

C1 = [[1.8(1+0.22)/30 + 0.22] *100 = 29.32

C2 = [0(7+0.25)/25] = 0

Maximum amount invested in 1 firm = 25,00,000 * 40% = 1,00,0000

Minimum invested in each stock = 1,00,000

Return on portfolio if invested equally = (10.25 + 13.3 + 7.51 + 9.2 + 11.76 + 16.92 + 29.32 + 0)/8 = 12.28

40% investment in (C1-C2), C1= 900000 C2= 100000

40% investment in (H1-H3), H1= 100000, H2= 100000, H3= 800000

Remaining balance in (S1-S3), S1= 100000, S2= 300000, S3= 100000

Maximum return in portfolio = 10.25*(0.1/2.5) + 13.3*(0.3/2.5) + 7.51*(0.1/2.5) + 9.2*(0.1/2.5) + 11.76*(0.1/2.5) + 16.92*(0.8/2.5) + 29.32*(0.9/2.5) + 0*(0.1/2.5) = 19.11%

Optimal number of shares to buy in each stock

S1= 100000/40 = 2500

S3= 100000/50 = 2000

S2= 300000/80 = 3750

H1= 100000/60 = 1666.67

H2 = 100000/45 = 2222.22

H3 = 800000/60 = 13333.33

C1 = 900000/30 = 30000

C2 = 100000/25 = 4000

(b) Removing the integer restrictions on the number of shares

Let’s take the optimal number of shares invested in

H1= 1660

H2= 2200

H3= 13000

The investment on H1 = 1660 * 60 = 99600

H2 = 2200 * 45 = 99000

H3 = 13000 * 60 = 780000

New total investment = 100000 + 100000 + 300000 + 99600 + 99000 + 780000 + 900000 + 100000 = 2478600

% change = (2500000 – 2478600)/2500000 = 0.85%

New optimal function = 19.04%

% change = (19.11 – 19.04)/19.11 = 0.36%

Solving by lpSolve

Question 2

```
library(lpSolveAPI)
```

Let the dividend be D, growth rate be G, and price per share be P # software companies

```
S1 <- as.integer(40) # rate .05 dividend 2
S1

## [1] 40

S2 <- as.integer(50) # rate .1 + dividend 1.5
S2

## [1] 50

S3 <- as.integer(80) # (80*.03) + 80 + 3.5
S3

## [1] 80
```

hardware companies

```r
H1 <- as.integer(60) # (60*.04) + 60 + 3
H1

## [1] 60

H2 <- as.integer(45) # (45*.07) + 45 + 2
H2

## [1] 45

H3 <- as.integer(60) # (60*.15) + 60 + 1
H3

## [1] 60
```

consulting companies

```r
C1 <- as.integer(30) # (30*.22) + 30 + 1.8
C1

## [1] 30

C2 <- as.integer(25) # (25*.25) + 25 + 0
C2

## [1] 25

Y1 <- (.05*2) + (.05*S1)
Y2 <- (.1*1.5) + (.1*S2)
Y3 <- .03*3.5 + (.03*S3)
Y4 <- .04*3 + (.04*H1)
Y5 <-.07*2 + (.07*H2)
Y6 <-.15*1 + (.15*H3)
Y7 <-.22*1.8 + (.22*C1)
Y8 <-.25*0 + (.25*C2)
Y1

## [1] 2.1

Y2

## [1] 5.15

Y3

## [1] 2.505

Y4

## [1] 2.52

Y5
```

```
## [1] 3.29
```

```
Y6
```

```
## [1] 9.15
```

```
Y7
```

```
## [1] 6.996
```

```
Y8
```

```
## [1] 6.25
```

```
2500000/1000
```

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

```
100000/1000
```

```
## [1] 100
```

```
2500*.4
```

```
## [1] 1000
```

Objective Function

```
# Max Y1+ Y2+ Y3+ Y4+ Y5+ Y6+ Y7+ Y8

# Constraints

# S1*Y1 >= 100
# S2*Y2 >= 100
# S3*Y3 >= 100
# H1*Y4 >= 100
# H2*Y5 >= 100
# H3*Y6 >= 100
# C1*Y7 >= 100
# C2*Y8 >= 100

#Constraints 1
# S1*Y1 + S2*Y2 + S3*Y3 <= 1000
# H1*Y4+ H2*Y5 + H3*Y6 <= 1000
# C1*Y7 + C2*Y8 <=1000
#Constraint 2
# Y1+Y2+Y3+Y4+Y5+Y6+Y7+Y8 = 2500

AX<- as.integer(2500)
typeof(AX)
```

```
## [1] "integer"
```

```r
RS <- as.integer(1000)
RS

## [1] 1000

DI <- as.integer(100)
DI

## [1] 100

Assign2<- make.lp(0,8)


set.objfn(Assign2, c(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8))

lp.control(Assign2, 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"
```

```
Assign2
```

```
## Model name:
##                C1     C2     C3     C4     C5     C6     C7     C8
## Maximize    2.1   5.15  2.505   2.52   3.29   9.15  6.996   6.25
## Kind        Std    Std    Std    Std    Std    Std    Std    Std
## Type       Real   Real   Real   Real   Real   Real   Real   Real
## Upper       Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
## Lower         0      0      0      0      0      0      0      0
```

#constraint

```
add.constraint(Assign2, c(S1,0,0,0,0,0,0,0), ">=",100)
add.constraint(Assign2, c(0,S2,0,0,0,0,0,0), ">=",100)
add.constraint(Assign2, c(0,0,S3,0,0,0,0,0), ">=",100)
add.constraint(Assign2, c(0,0,0,H1,0,0,0,0), ">=",100)
add.constraint(Assign2, c(0,0,0,0,H2,0,0,0), ">=",100)
```

```
add.constraint(Assign2, c(0,0,0,0,0,H3,0,0), ">=",100)
add.constraint(Assign2, c(0,0,0,0,0,0,C1,0), ">=",100)
add.constraint(Assign2, c(0,0,0,0,0,0,0,C2), ">=",100)
add.constraint(Assign2, c(1,1,1,0,0,0,0,0), "<=",1000)
add.constraint(Assign2, c(0,0,0,1,1,1,0,0), "<=", 1000)
add.constraint(Assign2, c(0,0,0,0,0,0,1,1), "<=", 1000)
add.constraint(Assign2, c(1,1,1,1,1,1,1,1), "=", AX)


solve(Assign2)

## [1] 0

get.objective(Assign2)

## [1] 18683.01

get.variables(Assign2)

## [1]   2.500000 496.250000   1.250000   1.666667   2.222222 996.111111
996.000000
## [8]   4.000000

#S2, H3 & C1 taking consideration
IN1 <- 496.25/2500
IN2 <- 996.11/2500
IN3 <- 996/2500
IN1+IN2+IN3

## [1] 0.995344

Assign3 <- make.lp(0,8)


set.objfn(Assign3, c(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8))

lp.control(Assign3, 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"
```

```
Assign3
```

```
## Model name:
##                C1      C2      C3      C4      C5      C6      C7      C8
## Maximize      2.1    5.15   2.505    2.52    3.29    9.15   6.996    6.25
## Kind          Std     Std     Std     Std     Std     Std     Std     Std
## Type         Real    Real    Real    Real    Real    Real    Real    Real
## Upper         Inf     Inf     Inf     Inf     Inf     Inf     Inf     Inf
## Lower           0       0       0       0       0       0       0       0
```

#Constraint

```
add.constraint(Assign3, c(S1,0,0,0,0,0,0,0), ">=",DI)
add.constraint(Assign3, c(0,S2,0,0,0,0,0,0), ">=",DI)
add.constraint(Assign3, c(0,0,S3,0,0,0,0,0), ">=",DI)
add.constraint(Assign3, c(0,0,0,H1,0,0,0,0), ">=",DI)
add.constraint(Assign3, c(0,0,0,0,H2,0,0,0), ">=",DI)
add.constraint(Assign3, c(0,0,0,0,0,H3,0,0), ">=",DI)
add.constraint(Assign3, c(0,0,0,0,0,0,C1,0), ">=",DI)
add.constraint(Assign3, c(0,0,0,0,0,0,0,C2), ">=",DI)
add.constraint(Assign3, c(1,1,1,0,0,0,0,0), "<=", RS)
add.constraint(Assign3, c(0,0,0,1,1,1,0,0), "<=", RS)
add.constraint(Assign3, c(0,0,0,0,0,0,1,1), "<=", RS)
add.constraint(Assign3, c(1,1,1,1,1,1,1,1), "=", AX)


solve(Assign3)
```

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

```
get.objective(Assign3)
```

```
## [1] 18683.01
```

```
get.variables(Assign3)
```

```
## [1]   2.500000 496.250000   1.250000   1.666667   2.222222 996.111111
996.000000
## [8]   4.000000
```

#solution(Comparision)

```
2.1*2+5.15*496+2.505*2 > 2.1*3+5.15*496+2.505*1
```

```
## [1] TRUE
```

```
2.52*2 + 3.29*2 + 9.15*996 > 2.52*1 + 3.29*3 + 9.15*996
```

```
## [1] FALSE

(2.1*2+5.15*496+2.505*2 + 2.52*1 + 3.29*3 + 9.15*996 + 6.996*996 + 6.25*4)

## [1] 18682.42

18683.01 - 18682.42

## [1] 0.59

#As per as the integer values, the optimal solution is
#.59(*1000) less
```