Required Solution

> library(lpSolve)

> library(lpSolveAPI)

> # creating 9 variables and 0 constraints

> lpod <- make.lp(0,9)

> #creating objective function

> set.objfn(lpod,c(420,420,420,360,360,360,300,300,300))

> #maximize profit

> lp.control(lpod, 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"

> #Adding Constraints

> add.constraint(lpod, c(1,1,1,0,0,0,0,0,0), "<=", 750 )

> add.constraint(lpod, c(0,0,0,1,1,1,0,0,0), "<=", 900)

> add.constraint(lpod, c(0,0,0,0,0,0,1,1,1), "<=", 450)

> add.constraint(lpod, c(20,15,12,0,0,0,0,0,0), "<=", 13000)

> add.constraint(lpod, c(0,0,0,20,15,12,0,0,0), "<=", 12000)

> add.constraint(lpod, c(0,0,0,0,0,0,20,15,12), "<=", 5000)

> add.constraint(lpod, c(1,1,1,0,0,0,0,0,0), "<=", 900)

> add.constraint(lpod, c(0,0,0,1,1,1,0,0,0), "<=", 1200)

> add.constraint(lpod, c(0,0,0,0,0,0,1,1,1), "<=", 750)

> add.constraint(lpod ,c(900,-750,0,900,-750,0,900,-750,0), "=", 0)

> add.constraint(lpod ,c(0,450,-900,0,450,-900,0,450,-900), "=", 0)

> add.constraint(lpod ,c(450,0,-750,450,0,-750,450,0,-750),"=",0)

> #identifying the variable constrant

> RowNames <-c("PC1","PC2","PC3",

+ "SS1","SS2","SS3",

+ "FCL","FCM","FCS",

+ "PerCP1andP2","PerCP2andP3","PerCP1andP3")

> ColNames <- c("PL1","PL2","PL3",

+ "PM1","PM2","PM3",

+ "PS1","PS2","PS3")

> solve(lpod)

[1] 0

> get.objective(lpod)

[1] 699026.5

> get.constraints(lpod)

[1] 750.0000 858.4071 250.0000 13000.0000 12000.0000

[6] 5000.0000 750.0000 858.4071 250.0000 0.0000

[11] 0.0000 0.0000

> dimnames(lpod) <- list(RowNames, ColNames)

> lpod

Model name:

a linear program with 9 decision variables and 12 constraints

> ```

Error: attempt to use zero-length variable name

>

> #shadow prices, dual solution, and reduced costs

>

> ```{r}

Error: attempt to use zero-length variable name

> # identyfying reduced cost

> get.sensitivity.obj(lpod)

$objfrom

[1] 4.200e+02 4.125e+02 4.200e+02 -1.000e+30 3.600e+02

[6] 3.480e+02 2.800e+02 -1.000e+30 -1.000e+30

$objtill

[1] 4.40e+02 4.20e+02 4.32e+02 3.60e+02 3.75e+02 3.60e+02 1.00e+30

[8] 3.15e+02 3.24e+02

> #identiofying shadow cost

> get.sensitivity.rhs(lpod)

$duals

[1] 60.0000000 0.0000000 0.0000000 22.3008850 22.3008850

[6] 19.3008850 0.0000000 0.0000000 0.0000000 -0.0339823

[11] 0.0000000 -0.1231858 0.0000000 0.0000000 0.0000000

[16] 0.0000000 0.0000000 0.0000000 0.0000000 -15.0000000

[21] -24.0000000

$dualsfrom

[1] 7.287611e+02 -1.000000e+30 -1.000000e+30 1.216129e+04

[5] 9.120000e+03 1.904762e+03 -1.000000e+30 -1.000000e+30

[9] -1.000000e+30 0.000000e+00 -1.000000e+30 0.000000e+00

[13] -1.000000e+30 -1.000000e+30 -1.000000e+30 -8.628319e+01

[17] -1.000000e+30 -1.000000e+30 -1.000000e+30 -8.495575e+01

[21] -1.039823e+02

$dualstill

[1] 8.075221e+02 1.000000e+30 1.000000e+30 1.335821e+04

[5] 1.267143e+04 7.285714e+03 1.000000e+30 1.000000e+30

[9] 1.000000e+30 0.000000e+00 1.000000e+30 0.000000e+00

[13] 1.000000e+30 1.000000e+30 1.000000e+30 6.371681e+01

[17] 1.000000e+30 1.000000e+30 1.000000e+30 1.150442e+02

[21] 1.438053e+02

> # identyfing Dual solution

> get.dual.solution(lpod)

[1] 1.0000000 60.0000000 0.0000000 0.0000000 22.3008850

[6] 22.3008850 19.3008850 0.0000000 0.0000000 0.0000000

[11] -0.0339823 0.0000000 -0.1231858 0.0000000 0.0000000

[16] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

[21] -15.0000000 -24.0000000

> ```

Error: attempt to use zero-length variable name

>

> #sensitivity calculation

>

>

>

> ```{r}

Error: attempt to use zero-length variable name

> Sensivity<-data.frame(get.sensitivity.rhs(lpod)$duals[1:21],get.sensitivity.rhs(lpod)$dualsfrom[1:21],get.sensitivity.rhs(lpod)$dualstill[1:21])

> names(Sensivity)<-c("Price","low","High")

> Sensivity

Price low High

1 60.0000000 7.287611e+02 8.075221e+02

2 0.0000000 -1.000000e+30 1.000000e+30

3 0.0000000 -1.000000e+30 1.000000e+30

4 22.3008850 1.216129e+04 1.335821e+04

5 22.3008850 9.120000e+03 1.267143e+04

6 19.3008850 1.904762e+03 7.285714e+03

7 0.0000000 -1.000000e+30 1.000000e+30

8 0.0000000 -1.000000e+30 1.000000e+30

9 0.0000000 -1.000000e+30 1.000000e+30

10 -0.0339823 0.000000e+00 0.000000e+00

11 0.0000000 -1.000000e+30 1.000000e+30

12 -0.1231858 0.000000e+00 0.000000e+00

13 0.0000000 -1.000000e+30 1.000000e+30

14 0.0000000 -1.000000e+30 1.000000e+30

15 0.0000000 -1.000000e+30 1.000000e+30

16 0.0000000 -8.628319e+01 6.371681e+01

17 0.0000000 -1.000000e+30 1.000000e+30

18 0.0000000 -1.000000e+30 1.000000e+30

19 0.0000000 -1.000000e+30 1.000000e+30

20 -15.0000000 -8.495575e+01 1.150442e+02

21 -24.0000000 -1.039823e+02 1.438053e+02

> ```

Error: attempt to use zero-length variable name

>

> # solution of the dual

>

> ```{r}

Error: attempt to use zero-length variable name

> lpoddual <- make.lp(0,12)

> set.objfn(lpoddual, c(750,900,450,13000,12000,5000,900,1200,750,0,0,0))

> lp.control(lpoddual,sense='min',simplextype="dual")

$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] "minimize"

$simplextype

[1] "dual" "dual"

$timeout

[1] 0

$verbose

[1] "neutral"

> add.constraint(lpoddual ,c(1,0,0,20,0,0,1,0,0,900,0,450), ">=", 420)

> add.constraint(lpoddual ,c(0,1,0,0,20,0,1,0,0,-750,450,0), ">=", 420)

> add.constraint(lpoddual ,c(0,0,1,0,0,20,1,0,0,0,-900,-750), ">=", 420)

> add.constraint(lpoddual ,c(1,0,0,15,0,0,0,1,0,900,0,450), ">=", 360)

> add.constraint(lpoddual ,c(0,1,0,0,15,0,0,1,0,-750,450,0), ">=", 360)

> add.constraint(lpoddual ,c(0,0,1,0,0,15,0,1,0,0,-900,-750), ">=", 360)

> add.constraint(lpoddual ,c(1,0,0,12,0,0,0,0,1,900,0,450), ">=", 300)

> add.constraint(lpoddual ,c(0,1,0,0,12,0,0,0,1,-750,450,0), ">=", 300)

> add.constraint(lpoddual ,c(0,0,1,0,0,12,0,0,1,0,-900,-750), ">=", 300)

> '''

+ '''{r}

Error: unexpected string constant in:

"'''

'"

> solve(lpoddual)

[1] 0

> get.objective(lpoddual)

[1] 696000

> get.variables(lpoddual)

[1] 0.0000000 0.0000000 0.0000000 12.0000000 20.0000000

[6] 60.0000000 0.0000000 0.0000000 0.0000000 0.2000000

[11] 0.4666667 0.0000000

> get.constraints(lpoddual)

[1] 420 460 780 360 360 480 324 300 300

> ```

Maximize A= 420L1 +360M1 + 300S1 + 420L2 + 360M2 + 300S2 + 420L3 +360M3 + 300S3

Subject to: L1+M1+S1<=750

L2+M2+S2<=900

L3+M3+S3<=450

20L1+15M1+12S1<=13000

20L2+15M2+12S2<=12000

20L3+15M3+12S3<=5000

L1+L2+L3<=900

M1+M2+M3<=1200

S1+S2+S3<=750

Where L1, L2, L3, M1, M2, M3, S1, S2, S3>=0

Let x1, x2, x3, x4, x5, x6, x7, x8, x9 be the dual variables

So, the dual will be:

Minimize B= 750x1 + 900x4 + 450x7 + 13000x2 + 12000x5 + 5000x8 + 900x3 + 1200x6 + 750x9

Now determining the dual constraints

Constructing a matrix for the coefficients of the constraints

1 0 0 1 0 0 1 0 0

0 1 0 0 1 0 0 1 0

0 0 1 0 0 1 0 0 1

20 0 0 15 0 0 12 0 0

0 20 0 0 15 0 0 12 0

0 0 20 0 0 15 0 0 12

1 1 1 0 0 0 0 0 0

0 0 0 1 1 1 0 0 0

0 0 0 0 0 0 1 1 1

Taking the transpose of the above matrix

0 0 1 0 0 20 0 0 1

0 0 1 0 20 0 0 1 0

0 0 1 20 0 0 1 0 0

0 1 0 0 0 15 0 0 1

0 1 0 0 15 0 0 1 0

0 1 0 15 0 0 1 0 0

1 0 0 0 0 12 0 0 1

1 0 0 0 12 0 0 1 0

1 0 0 12 0 0 1 0 0

So, the dual constraints will be:

X3 + 20x6 + x9 >= 420

X3 + 20x5 + x8 >= 360

X3 + 20x4 + x7 >= 300

X2 + 15x6 + x9 >=420

X2 + 15x5 + x8 >= 360

X2 + 15x4 + x7 >= 300

X1 + 12x6 + x9 >= 420

X1 + 12x5 + x8 >= 360

X1 + 12x4 + x7 >= 300