4) 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

Required solution: (1,2,3)

> library(lpSolveAPI)  
> setwd("~/Desktop/SEMESTER 1")  
> lpmod<- make.lp(0, 9)  
> lpmod  
Model name:  
  a linear program with 9 decision variables and 0 constraints  
> set.objfn(lpmod, c(420, 360, 300, 420, 360, 300, 420, 360, 300))  
> lp.control(lpmod,sense='max')  
$anti.degen  
[1] "none"  
  
$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(lpmod, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 750)  
> add.constraint(lpmod, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 900)  
> add.constraint(lpmod, c(0, 0, 0, 0, 0, 0,1, 1, 1), "<=", 450)  
> add.constraint(lpmod, c(20, 15, 12, 0, 0, 0, 0, 0, 0), "<=", 13000)  
> add.constraint(lpmod, c(0, 0, 0, 20, 15, 12, 0, 0, 0), "<=", 12000)  
> add.constraint(lpmod, c(0, 0, 0, 0, 0, 0, 20, 15, 12), "<=", 5000)  
> add.constraint(lpmod, c(1, 1, 1, 0, 0, 0, 0, 0, 0), "<=", 900)  
> add.constraint(lpmod, c(0, 0, 0, 1, 1, 1, 0, 0, 0), "<=", 12000)  
> add.constraint(lpmod, c(0, 0, 0, 0, 0, 0, 1, 1, 1), "<=", 750)  
> add.constraint(lpmod, c(6, 6, 6, -5, -5, -5, 0, 0, 0), "=", 0)  
> add.constraint(lpmod, c( 3, 3, 3, 0, 0, 0, -5, -5, -5), "=", 0)  
> set.bounds(lpmod, lower = c(0, 0, 0, 0, 0, 0, 0, 0, 0), columns = c(1, 2,3,4,5,6,7,8,9))

RowsNames <- c("CCN1", "CCN2", "CCN3", "SCN1", "SCN2", "SCN3", "S1CN1", "S1CN2", "S1CN3", "%C1", "%C2")  
> ColumnsNames <- c("Large1", "Medium1", "Small1", "Large2", "Medium2", "Small2", "Large3", "Medium3", "Small3")  
> dimnames(lpmod) <- list(RowsNames, ColumnsNames)

> lpmod  
Model name:  
  a linear program with 9 decision variables and 11 constraints  
> solve(lpmod)  
[1] 0

> get.objective(lpmod)

[1] 696000

> get.variables(lpmod)

[1] 516.6667 177.7778 0.0000 0.0000 666.6667 166.6667 0.0000

[8] 0.0000 416.6667

get.constraints(lpmod)  
 [1]   694.4444   833.3333   416.6667 13000.0000 12000.0000  
 [6]  5000.0000   694.4444   833.3333   416.6667     0.0000  
[11]     0.0000  
> get.sensitivity.rhs(lpmod)  
$duals  
 [1]    0    0    0   12   20   60    0    0    0  -12   84    0    0  
[14]  -24  -40    0    0 -360 -120    0

$dualsfrom  
 [1] -1.000000e+30 -1.000000e+30 -1.000000e+30  1.041667e+04  
 [5]  1.000000e+04  4.800000e+03 -1.000000e+30 -1.000000e+30  
 [9] -1.000000e+30 -3.333333e+02 -8.333333e+01 -1.000000e+30  
[13] -1.000000e+30 -8.611111e+02 -1.000000e+02 -1.000000e+30  
[17] -1.000000e+30 -5.000000e+01 -1.333333e+02 -1.000000e+30

$dualstill  
 [1] 1.000000e+30 1.000000e+30 1.000000e+30 1.388889e+04 1.250000e+04  
 [6] 5.400000e+03 1.000000e+30 1.000000e+30 1.000000e+30 1.666667e+02  
[11] 1.666667e+02 1.000000e+30 1.000000e+30 1.111111e+02 2.500000e+02  
[16] 1.000000e+30 1.000000e+30 2.500000e+01 6.666667e+01 1.000000e+30  
  
> get.sensitivity.obj(lpmod)  
$objfrom  
[1]  3.60e+02  3.45e+02 -1.00e+30 -1.00e+30  3.45e+02  2.52e+02  
[7] -1.00e+30 -1.00e+30  2.04e+02

$objtill  
[1] 4.60e+02 4.20e+02 3.24e+02 4.60e+02 4.20e+02 3.24e+02 7.80e+02  
[8] 4.80e+02 1.00e+30