Assignment 3

Solving Question 1

Loading packages

```
library(lpSolve)
library(lpSolveAPI)
```

creating lp with 9 variables and 0 constraints

```
lp<- make.lp(0,9)
```

##Specify the objective function

```
set.objfn(lp, c(420,420,420,360,360,360,300,300,300))
lp.control(lp, 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(lp, c(1,1,1,0,0,0,0,0,0), "<=", 750 )
add.constraint(lp, c(0,0,0,1,1,1,0,0,0), "<=", 900)
add.constraint(lp, c(0,0,0,0,0,0,1,1,1), "<=", 450)
add.constraint(lp, c(20,15,12,0,0,0,0,0,0), "<=", 13000)
add.constraint(lp, c(0,0,0,20,15,12,0,0,0), "<=", 12000)
add.constraint(lp, c(0,0,0,0,0,0,20,15,12), "<=", 5000)
add.constraint(lp, c(1,1,1,0,0,0,0,0,0), "<=", 900)
add.constraint(lp, c(0,0,0,1,1,1,0,0,0), "<=", 1200)
add.constraint(lp, c(0,0,0,0,0,0,1,1,1), "<=", 750)
add.constraint(lp ,c(900,-750,0,900,-750,0,900,-750,0), "=", 0)
add.constraint(lp ,c(0,450,-900,0,450,-900,0,450,-900), "=", 0)
add.constraint(lp ,c(450,0,-750,450,0,-750,450,0,-750),"=",0)
```

Assignining rownames and column names

```
RowNames <-c("1-ProductionCapacity","2-ProductionCapacity","3-ProductionCapacity",
             "1-StorageSpace","2-StorageSpace","3-StorageSpace",
             "ForecastLarge","ForecastMedium","ForecastSmall",
             "PercentCapP1andP2","PercentCapP2andP3","PercentCapP1andP3")

ColNames <- c("1-PlantLarge","2-PlantLarge","3-PlantLarge",
              "1-PlantMedium","2-PlantMedium","3-PlantMedium",
              "1-PlantSmall","2-PlantSmall","3-PlantSmall")
```

Getting Constraints

```
solve(lp)
```

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

```
get.objective(lp)
```

```
## [1] 699026.5
```

```
get.constraints(lp)
```

```
##  [1]   750.0000   858.4071   250.0000 13000.0000 12000.0000  5000.0000
##  [7]   750.0000   858.4071   250.0000     0.0000     0.0000     0.0000
```

```
dimnames(lp) <- list(RowNames, ColNames)
lp
```

```
## Model name:
##   a linear program with 9 decision variables and 12 constraints
```

Solving Question 2

Identifying the shadow prices, dual solution, and reduced costs

```
# Reduced Costs
get.sensitivity.obj(lp)
```

```
## $objfrom
## [1]  4.200e+02  4.125e+02  4.200e+02 -1.000e+30  3.600e+02  3.480e+02  2.800e+02
## [8] -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 3.15e+02
## [9] 3.24e+02
```

```
#Shadow Prices
get.sensitivity.rhs(lp)
```

```
## $duals
##  [1]  60.0000000   0.0000000   0.0000000  22.3008850  22.3008850  19.3008850
##  [7]   0.0000000   0.0000000   0.0000000  -0.0339823   0.0000000  -0.1231858
## [13]   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
## [19]   0.0000000 -15.0000000 -24.0000000
##
## $dualsfrom
##  [1]  7.287611e+02 -1.000000e+30 -1.000000e+30  1.216129e+04  9.120000e+03
##  [6]  1.904762e+03 -1.000000e+30 -1.000000e+30 -1.000000e+30  0.000000e+00
## [11] -1.000000e+30  0.000000e+00 -1.000000e+30 -1.000000e+30 -1.000000e+30
## [16] -8.628319e+01 -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 1.267143e+04
##  [6] 7.285714e+03 1.000000e+30 1.000000e+30 1.000000e+30 0.000000e+00
## [11] 1.000000e+30 0.000000e+00 1.000000e+30 1.000000e+30 1.000000e+30
## [16] 6.371681e+01 1.000000e+30 1.000000e+30 1.000000e+30 1.150442e+02
## [21] 1.438053e+02
```

```
#Dual solution
get.dual.solution(lp)
```

```
##  [1]   1.0000000  60.0000000   0.0000000   0.0000000  22.3008850  22.3008850
##  [7]  19.3008850   0.0000000   0.0000000   0.0000000  -0.0339823   0.0000000
## [13]  -0.1231858   0.0000000   0.0000000   0.0000000   0.0000000   0.0000000
## [19]   0.0000000   0.0000000 -15.0000000 -24.0000000
```

Solving Question 3

Identifying the sensitivity of the above prices and costs.

```
Sensivity<-data.frame(get.sensitivity.rhs(lp)$duals[1:21],get.sensitivity.rhs(lp)$dualsfrom[1:21
],get.sensitivity.rhs(lp)$dualstill[1:21])
names(Sensivity)<-c("Price","low","High")
```

Specifying the range of shadow prices and reduced cost within which the optimal solution will not change.

```
Sensivity
```

Price	low	High
<dbl>	<dbl>	<dbl>

Price <dbl>	low <dbl>	High <dbl>
60.0000000	7.287611e+02	8.075221e+02
0.0000000	-1.000000e+30	1.000000e+30
0.0000000	-1.000000e+30	1.000000e+30
22.3008850	1.216129e+04	1.335821e+04
22.3008850	9.120000e+03	1.267143e+04
19.3008850	1.904762e+03	7.285714e+03
0.0000000	-1.000000e+30	1.000000e+30
0.0000000	-1.000000e+30	1.000000e+30
0.0000000	-1.000000e+30	1.000000e+30
-0.0339823	0.000000e+00	0.000000e+00

1-10 of 21 rows Previous **1** 2 3 Next

solving Question 4

Formulating the dual of the above problem.

```
lpdual <- make.lp(0,12)
set.objfn(lpdual, c(750,900,450,13000,12000,5000,900,1200,750,0,0,0))
lp.control(lpdual,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(lpdual ,c(1,0,0,20,0,0,1,0,0,900,0,450), ">=", 420)
add.constraint(lpdual ,c(0,1,0,0,20,0,1,0,0,-750,450,0), ">=", 420)
add.constraint(lpdual ,c(0,0,1,0,0,20,1,0,0,0,-900,-750), ">=", 420)
add.constraint(lpdual ,c(1,0,0,15,0,0,0,1,0,900,0,450), ">=", 360)
add.constraint(lpdual ,c(0,1,0,0,15,0,0,1,0,-750,450,0), ">=", 360)
add.constraint(lpdual ,c(0,0,1,0,0,15,0,1,0,0,-900,-750), ">=", 360)
add.constraint(lpdual ,c(1,0,0,12,0,0,0,0,1,900,0,450), ">=", 300)
add.constraint(lpdual ,c(0,1,0,0,12,0,0,0,1,-750,450,0), ">=", 300)
add.constraint(lpdual ,c(0,0,1,0,0,12,0,0,1,0,-900,-750), ">=", 300)
```

Getting constraints

```
solve(lpdual)
```

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

```
get.objective(lpdual)
```

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

```
get.variables(lpdual)
```

```
##  [1]  0.0000000  0.0000000  0.0000000 12.0000000 20.0000000 60.0000000
##  [7]  0.0000000  0.0000000  0.0000000  0.2000000  0.4666667  0.0000000
```

```
get.constraints(lpdual)
```

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

In a primal-dual pair of linear pro grammings, if either the primal or the dual problem has an optimal solution,

then the other does also, and the two optimal objective values are equal. Both the primal and the dual LP yield the same objective value. This is a consequence of the Strong Duality Theorem.

(Primal optimal) = (dual optimal) < -Strong dual theorem.