Cronin_#11

Hannah Cronin

2022-11-21

```
library(lpSolveAPI)
AP = make.lp(0,7)
```

Define variables:

X1 = shift 1 X2 = shift 2 X3 = shift 3 X4 = shift 4 X5 = shift 5 X6 = shift 6 X7 = shift 7

minimize: C(cost) = 775 X1 + 800 X2 + 800 X3 + 800 X4 + 800 X5 + 775 X6 + 750 X7 Constraints: X2 + X3 + X4 + X5 + X6 >= 18 #Sunday workers X3 + X4 + X5 + X6 + X7 >= 27 #Monday workers X1 + X4 + X5 + X6 + X7 >= 22 #Tuesday workers X1 + X2 + X5 + X6 + X7 >= 26 #Wednesday workers X1 + X2 + X3 + X6 + X7 >= 25 #Thursday workers X1 + X2 + X3 + X4 + X7 >= 21 #Friday workers X1 + X2 + X3 + X4 + X5 >= 19 #Saturday workers

```
set.objfn(AP, c(775, 800, 800, 800, 800, 775, 750))
lp.control(AP, sense = 'min')
```

```
## $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"   "primal"
##
## $timeout
## [1] 0
##
## $verbose
## [1] "neutral"
```

```
set.type(AP, 1:7, type=c("integer"))
add.constraint(AP, c(0,1,1,1,1,1,0), ">=", 18) #Sunday workers
add.constraint(AP, c(0,0,1,1,1,1,1), ">=", 27) #Monday workers
add.constraint(AP, c(1,0,0,1,1,1,0), ">=", 22) #Tuesday workers
add.constraint(AP, c(1,1,0,0,1,1,1), ">=", 26) #Wednesday workers
add.constraint(AP, c(1,1,1,0,0,1,1), ">=", 25) #Thursday workers
add.constraint(AP, c(1,1,1,1,0,0,1), ">=", 21) #Friday workers
add.constraint(AP, c(1,1,1,1,1,0,0), ">=", 19) #Saturday workers
```

```
Colnames = c('Shift1','Shift2','Shift3','Shift4','Shift5','Shift6','Shift7')
Rownames = c('Sunday workers','Monday workers','Tuesday workers','Wednesday workers','Th
ursday workers','Friday workers','Saturday workers')
dimnames(AP) = list(Rownames,Colnames)
```

```
AP
```

```
## Model name:
##                     Shift1  Shift2  Shift3  Shift4  Shift5  Shift6  Shift7
## Minimize              775     800     800     800     800     775     750
## Sunday workers          0       1       1       1       1       1       0  >=  18
## Monday workers          0       0       1       1       1       1       1  >=  27
## Tuesday workers         1       0       0       1       1       1       0  >=  22
## Wednesday workers       1       1       0       0       1       1       1  >=  26
## Thursday workers        1       1       1       0       0       1       1  >=  25
## Friday workers          1       1       1       1       0       0       1  >=  21
## Saturday workers        1       1       1       1       1       0       0  >=  19
## Kind                  Std     Std     Std     Std     Std     Std     Std
## Type                  Int     Int     Int     Int     Int     Int     Int
## Upper                 Inf     Inf     Inf     Inf     Inf     Inf     Inf
## Lower                   0       0       0       0       0       0       0
```

```
solve(AP)
```

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

```
get.objective(AP)
```

```
## [1] 26425
```

```
get.variables(AP)
```

```
## [1] 7 0 3 2 7 6 9
```

```
get.constraints(AP)
```

```
## [1] 18 27 22 29 25 21 19
```

Min(Cost) = $26,425

Min = 7 X1 + 0 X2 + 3 X3 + 2 X4 + 7 X5 + 6 X6 + 9 X7

7 Shift1 Workers, 0 Shift2 Workers, 3 Shift3 Workers, 2 Shift4 Workers, 7 Shift5 Workers, 6 Shift6 Workers, 9 Shift 7 Workers

There would be 18 workers on Sunday, 27 workers on Monday, 22 Workers on Tuesday, 26 Workers on

Wednesday,25 Workers on Thursday, 21 Workers on Friday, and 19 Workers on Saturday

##This solution is all integer-based (no decimals as you cannot hire partial people).