Integer Programming Assignment

Priyanka Jonnala

2023-11-20

Installing and loading the required libraries

```
#install.packages("lpSolve")
library(lpSolve)
#install.packages("lpSolveAPI")
library(lpSolveAPI)
```

This table below provides an estimate of the number of workers needed each day of the week.

```
Day_of_the_week = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
Workers_required = c(18,27,22,26,25,21,19)
Worker_per_day = data.frame(Day_of_the_week, Workers_required)
Worker_per_day
```

```
##   Day_of_the_week Workers_required
## 1          Sunday               18
## 2          Monday               27
## 3         Tuesday               22
## 4       Wednesday               26
## 5        Thursday               25
## 6          Friday               21
## 7        Saturday               19
```

Package handlers at AP are guaranteed a five-day work week with two consecutive days off. The base wage for the handlers is $750 per week. Workers working on Saturday or Sunday receive an additional $25 per day. The possible shifts and salaries for package handlers are:

```
Shifts = c(1,2,3,4,5,6,7)
Days_off = c("Sunday & Monday", "Monday & Tuesday", "Tuesday & Wednesday", "Wednesday & Thursday", "Thu
Wages = c(775, 800, 800, 800, 800, 775, 750)
shift_wage = data.frame(Shifts, Days_off, Wages)
shift_wage
```

```
##   Shifts             Days_off Wages
## 1      1      Sunday & Monday   775
## 2      2     Monday & Tuesday   800
## 3      3  Tuesday & Wednesday   800
## 4      4 Wednesday & Thursday   800
## 5      5    Thursday & Friday   800
## 6      6    Friday & Saturday   775
## 7      7    Saturday & Sunday   750
```

Defining the decision variables:

w1 = no.of workers in shift1

w2 = no.of workers in shift2

w3 = no.of workers in shift3

w4 = no.of workers in shift4

w5 = no.of workers in shift5

w6 = no.of workers in shift6

w7 = no.of workers in shift7

The Objective function is

Z = 775(w1) + 800(w2) + 800(w3) + 800(w4) + 800(w5) + 775(w6) + 750(w7)

Constraints:

Sunday : 0(w1)+1(w2)+1(w3)+1(w4)+1(w5)+1(w6)+0(w7) >= 18

Monday : 0(w1)+0(w2)+1(w3)+1(w4)+1(w5)+1(w6)+1(w7) >= 27

Tuesday : 1(w1)+0(w2)+0(w3)+1(w4)+1(w5)+1(w6)+1(w7) >= 22

Wednesday : 1(w1)+1(w2)+0(w3)+0(w4)+1(w5)+1(w6)+1(w7) >= 26

Thursday : 1(w1)+1(w2)+1(w3)+0(w4)+0(w5)+1(w6)+1(w7) >= 25

Friday : 1(w1)+1(w2)+1(w3)+1(w4)+0(w5)+0(w6)+1(w7) >= 21

Saturday : 1(w1)+1(w2)+1(w3)+1(w4)+1(w5)+0(w6)+0(w7) >= 19

Non-negativity Constraints: w1,w2,w3,w4,w5,w6,w7 >= 0

```
solvelp = make.lp(7, 7)
#Setting the objective function
set.objfn(solvelp, c(775, 800, 800, 800, 800, 775, 750))
#Setting the objective function to "min"
lp.control(solvelp, 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"
```

Adding constraints

```
set.row(solvelp,1,c(0,1,1,1,1,1,0))
set.row(solvelp,2,c(0,0,1,1,1,1,1))
set.row(solvelp,3,c(1,0,0,1,1,1,1))
```

```
set.row(solvelp,4,c(1,1,0,0,1,1,1))
set.row(solvelp,5,c(1,1,1,0,0,1,1))
set.row(solvelp,6,c(1,1,1,1,0,0,1))
set.row(solvelp,7,c(1,1,1,1,1,0,0))
```

Adding the right hand side coefficients.

```
rhs<-c(18,27,22,26,25,21,19)
set.rhs(solvelp,rhs)
```

Setting the Constraint types

```
set.constr.type(solvelp,c(">=", ">=", ">=", ">=", ">=", ">=", ">="))
set.bounds(solvelp,lower = rep(0,7))
```

Giving row names

```
lp.rownames <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
dimnames(solvelp) = list(lp.rownames,(1:7))
```

Column names

```
lp.colnames <- c("w1","w2","w3","w4","w5","w6","w7")
set.type(solvelp, columns = 1:7, type = "integer")
solvelp
```

```
## Model name:
##              1    2    3    4    5    6    7
## Minimize   775  800  800  800  800  775  750
## Sunday       0    1    1    1    1    1    0  >=  18
## Monday       0    0    1    1    1    1    1  >=  27
## Tuesday      1    0    0    1    1    1    1  >=  22
## Wednesday    1    1    0    0    1    1    1  >=  26
## Thursday     1    1    1    0    0    1    1  >=  25
## Friday       1    1    1    1    0    0    1  >=  21
## Saturday     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
```

Solving the LP Problem

```
solve(solvelp)
```

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

Finding the objective function of the LP Problem

```r
get.objective(solvelp)
```

```
## [1] 25675
```

```r
round(get.variables(solvelp))
```

```
## [1]  2  4  5  0  8  1 13
```

```r
get.constraints(solvelp)
```

```
## [1] 18 27 24 28 25 24 19
```

```r
Availability = matrix(c(0,4,5,0,8,1,0,
                 0,0,5,0,8,1,13,
                 2,0,0,0,8,1,13,
                 2,4,0,0,8,1,13,
                 2,4,5,0,0,1,13,
                 2,4,5,0,0,0,13,
                 2,4,5,0,8,0,0),ncol=7,byrow=TRUE)

colnames(Availability) = c("Shift1", "Shift2", "Shift3", "Shift4", "Shift5", "Shift6", "Shift7")

rownames(Availability) = c('Sunday', 'Monday','Tuesday','Wednesday','Thursday','Friday','Saturday')

Availability
```

```
##           Shift1 Shift2 Shift3 Shift4 Shift5 Shift6 Shift7
## Sunday         0      4      5      0      8      1      0
## Monday         0      0      5      0      8      1     13
## Tuesday        2      0      0      0      8      1     13
## Wednesday      2      4      0      0      8      1     13
## Thursday       2      4      5      0      0      1     13
## Friday         2      4      5      0      0      0     13
## Saturday       2      4      5      0      8      0      0
```

This shows the availability of workers each day.

```r
rowSums(Availability)
```

```
##    Sunday    Monday   Tuesday Wednesday  Thursday    Friday  Saturday
##        18        27        24        28        25        24        19
```

CONCLUSION

From the above formulation of the problem we can observe that

The total cost is $25,675

The optimal no.of workers available each day is:

Sunday : 18

Monday : 27

Tuesday : 24

Wednesday : 28

Thursday : 25

Friday : 24

Saturday : 19