Module 11 Assigment

Chris Hargis

2022-11-12

Objective Function:

MIN: Z = 775x1 + 800x2 + 800x3 + 800x4 + 800x5 + 775x6 + 750x7

Constraints:

x3 + x4 + x5 + x6 + x7 >= 27 #Monday

x1 + x4 + x5 + x6 + x7 >= 22 #Tuesday

x1 + x2 + x5 + x6 + x7 >= 26 #Wednesday

x1 + x2 + x3 + x6 + x7 >= 25 #Thursday

x1 + x2 + x3 + x4 + x7 >= 21 #Friday

x1 + x2 + x3 + x4 + x5 >= 19 #Saturday

x2 + x3 + x4 + x5 + x6 >= 18 #Sunday

```
library(lpSolveAPI)
```

```
## Warning: package 'lpSolveAPI' was built under R version 4.1.3
```

```
lprec = make.lp(0, 7)
```

Setting up the objective function

```
set.objfn(lprec, c(775, 800, 800, 800, 800, 775, 750))
lp.control(lprec, 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(lprec, 1:7, type = c("integer"))   #integers because you cant have
```

Constraints

```
add.constraint(lprec, c(0, 0, 1, 1, 1, 1, 1), ">=", 27) #Monday
add.constraint(lprec, c(1, 0, 0, 1, 1, 1, 1), ">=", 22) #Tuesday
add.constraint(lprec, c(1, 1, 0, 0, 1, 1, 1), ">=", 26) #Wednesday
add.constraint(lprec, c(1, 1, 1, 0, 0, 1, 1), ">=", 25) #Thursday
add.constraint(lprec, c(1, 1, 1, 1, 0, 0, 1), ">=", 21) #Friday
add.constraint(lprec, c(1, 1, 1, 1, 1, 0, 0), ">=", 19) #Saturday
add.constraint(lprec, c(0, 1, 1, 1, 1, 1, 0), ">=", 18) #Sunday
```

```
solve(lprec)
```

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

```
get.objective(lprec)
```

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

```
vars = get.variables(lprec)

vars
```

```
## [1]  5  1  5  0  8  4 10
```

Total Cost:

The total cost can be determined by using the below formula:

775x1 + 800x2 + 800x3 + 800x4 + 800x5 + 775x6 + 750x7 = Z

```
shift1_cost = vars[1] * 775     #Vars[x] indexes the vars variable we created

shift2_cost = vars[2] * 800

shift3_cost = vars[3] * 800

shift4_cost = vars[4] * 800

shift5_cost = vars[5] * 800

shift6_cost = vars[6] * 775

shift7_cost = vars[7] * 750

Week = c(shift1_cost, shift2_cost, shift3_cost, shift4_cost, shift5_cost, shift6_cost, shift7_cost)

Total_cost = sum(Week)

Total_cost
```

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

Workers available per day:

Monday:

Tuesday: 28

Wednesday: 28

Thursday: 25

Friday: 21

Saturday: 19

Sunday: 18