

R Notebook

A model to calculate the optimal number of workers in an 8-hour shift. The workers change shifts either at the beginning of the shift or at the middle, so that at every given shift workers from the current shift and the previous shift are present. We know the mean value of the required number of workers, but this requirement is stochastic. We suppose that it follows a Poisson distribution. For example if a certain 4 hour window requires 9 workers, then

```
cat('95% of the time the required workes will be less than', qpois(0.95,9),  
    'and more than', qpois(0.95,9,lower.tail = F))
```

```
## 95% of the time the required workes will be less than 14 and more than 4
```

Assume that we have the following mean rates and we set the constraint so that 60% of the time they will be satisfied.

```
mean_rates = c(9,22,25,17,30,13)  
CHANCECTR = 0.6 #60% of the times the requirements will be satisfied. The higher the bigger the value.
```

The variables $X[i] \geq 0, i = 1 : 6$ the number of workers

The constraints: The available staff should be more than the effective constraints $AvailStaff(X,i) \geq VaR(RequiredStaff; ChanceCtr), i = 1 : 6$

The objective $\min(\sum_i X[i])$

```
library(data.table)  
library(lpSolveAPI)  
library(kableExtra)  
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following objects are masked from 'package:data.table':  
##  
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,  
##     yday, year  
  
## The following objects are masked from 'package:base':  
##  
##     date, intersect, setdiff, union  
  
library(stringr)  
  
DF = data.frame(val = 1:6,  
                dt_start = paste0(as.character(seq(0,20,by=4)),':','00') ,  
                dt_end = paste0(as.character(seq(4,24,by=4)),':','00') ,  
                effective_rate_ctr = sapply(mean_rates, function(x) qpois(CHANCECTR,x)),  
                mean_rates = mean_rates,  
                conf_95_lower = sapply(mean_rates, function(x) qpois(0.95,x)),  
                conf_95_uper = sapply(mean_rates, function(x) qpois(0.95,x, lower.tail=F)),  
                stringsAsFactors = FALSE)
```

val	dt_start	dt_end	effective_rate_ctr	mean_rates	conf_95_lower	conf_95_uper
1	0:00	4:00	10	9	14	4
2	4:00	8:00	23	22	30	15
3	8:00	12:00	26	25	33	17
4	12:00	16:00	18	17	24	11
5	16:00	20:00	31	30	39	21
6	20:00	24:00	14	13	19	7

```
DF %>%
  kbl() %>%
  kable_styling() %>%
  column_spec(4, background = "red")
```

```
initialise_model = function(n){
  nVar = n
  nCtr = nVar
  lprec <- make.lp(nCtr,nVar)
  return(list('nVar' = nVar,
             'nCtr' = nCtr,
             'model' = lprec))
}

#Build variables
build_variables = function(nVar){
  VarX_col_idx = 1:nVar
  obj = rep(1,nVar)
  return(list('variables_index' = VarX_col_idx,
             'objective_coefficient' = obj))
}

#At each 4-hour the workers of the current and previous shift are present
AvailStaff = function(X,i,n){
  ifelse(i>1,X[i] + X[i-1],X[1] + X[n])
}

#Set constraints
build_constraints = function(nCtr, nVar,model){
  row_lb = matrix(NA, nrow = nCtr)
  row_ub = matrix(NA, nrow = nCtr)
  for(i in 1:nVar){
    row_ub[i] = -DF[i,"effective_rate_ctr"]
    if(i>1){
      set.row(model, i,c(-1,-1),
              indices = c(i,i-1))
    }else{
      set.row(model, i,c(-1,-1),
              indices = c(i,nVar))
    }
  }
}

return(list("row_lb" = row_lb,
           "row_ub" = row_ub))
```

```

}

set_controls = function(lprec, row_lb, row_ub,
                        nCtr,obj, varIdx){
  #Controls
  #set lhs and rhs
  set.constr.value(lprec,
                  lhs = row_lb,
                  rhs = row_ub, 1:nCtr)
  set.objfn(lprec, obj)
  set.type(lprec,
           columns = varIdx, type = c("integer"))
}

#solve the model
solve_program = function(lprec){
  solStatusCode = solve.lpExtPtr(lprec)
  sols <- get.variables(lprec)
  minimizer <- get.objective(lprec)

  return(list('status' = solStatusCode,
             'sololutions' = sols,
             'minimiser' = minimizer)
  )
}

```

Solving the program

```

init_mod = initialise_model(nrow(DF))
variables_objective = build_variables(init_mod$nVar)
constraints = build_constraints(init_mod$nCtr,
                              init_mod$nVar, init_mod$model)

set_controls(init_mod$model, constraints$row_lb,
            constraints$row_ub, init_mod$nCtr,
            variables_objective$objective_coefficient,
            variables_objective$variables_index)

# write model to model.lp
#write.lp(lprec, "MaintOpt.lp", type = c("lp"))

solution = solve_program(init_mod$model)

```

Calculate the available staff (from the two shifts) and the chances that the day will require no more workers than the solution

```

available_staff = sapply(1:length(mean_rates),
                        function(x) AvailStaff(solution$sololutions, x, length(mean_rates)) )

chances = sapply(1:length(mean_rates), function(x) ppois(available_staff[x],
                                                         mean_rates[x]))

```

val	dt_start	dt_end	effective_rate_ctr	mean_rates	conf_95_lower	conf_95_uper	solution	available_sta
1	0:00	4:00	10	9	14	4	10	1
2	4:00	8:00	23	22	30	15	25	3
3	8:00	12:00	26	25	33	17	1	2
4	12:00	16:00	18	17	24	11	17	1
5	16:00	20:00	31	30	39	21	14	3
6	20:00	24:00	14	13	19	7	0	1

```
DF$solution = solution$soloutions
DF$available_staff= available_staff
DF$chances = chances
```

```
DF %>%
  kbl() %>%
  kable_styling() %>%
  column_spec(4, background = "red") %>%
  column_spec(8, background = "green") %>%
  column_spec(9, background = "green") %>%
  column_spec(10, background = "green")
```