

Teoria das filas

Modelo para se utilizar as funções de teoria das filas Aqui teremos a aplicação da teoria de filas

Modelo de filas

Aqui foi criado uma função com o pacote queueing onde colocaremos os vetores:

- Lambda: Volumetria da hora
- mu: Média da quantidade de chamadas atendidas por operador na hora específica
- hora: Hora específica do Lambda e mu

obs: Existem variações no modelo de fila, nesse exemplo estamos utilizando o modelo MMC

```
library(queueing)
library(dplyr)

## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

# Acerta Qtd. op ----
library(dplyr)
otimiza<- function(lambda,mu,c,hora){
  temp<-NewInput.MMC(lambda,mu, c)
  temp2<-QueueingModel(temp)
  data<- data.frame("Fluxo" = lambda,
                     "Taxa média atendimento"=mu,
                     "Tempo medio de espera na fila minutos" = round(temp2$Wq*60,3),
                     # "Prob do tempo de espera ser 0" = round(temp2$Pn,4),
                     # "Taxa de ocupacao " = temp2$R0,
                     # "Numero medio clientes na fila" =temp2$Lq,
                     # "Numero medio clientes na fila quando tem fila" = temp2$Lqq,
                     # "Variancia do numero de clientes na fila" = temp2$VN,
                     # "Variancia do tempo de espera no sistema" = round(temp2$VT*60,3),
                     # "Variancia do tempo de espera na fila" = round(temp2$VTq*60,3),
                     "Hora"= hora,
                     "Qtd_op" = c)
```

```

    return(data)
}

filtra = function(lambda,mu=mu,hora){
  <<-1:70
  c_filtrado<-which(c>lambda/mu)
  data.frame(lambda =lambda, mu = mu, c = c_filtrado, hora= hora)
}

```

Simulando dados

Simulando o numero de chegadas de clientes em um dia por hora, considerando que nosso serviço tem uma taxa de atendimento de $\mu = 3$. abaixo os dados

```

data = as.Date("2020-05-20")+1:2
Hmd = rep(8:17,2); Hmd %>% length()

## [1] 20

Lambda = rep(seq(60,150,by = 10),2)+rnorm(1,5,10)
mu = rep(3,20)
dados<-data.frame(data,Hmd,Lambda,mu);dados

##           data Hmd      Lambda mu
## 1 2020-05-21   8  75.11242  3
## 2 2020-05-22   9  85.11242  3
## 3 2020-05-21  10  95.11242  3
## 4 2020-05-22  11 105.11242  3
## 5 2020-05-21  12 115.11242  3
## 6 2020-05-22  13 125.11242  3
## 7 2020-05-21  14 135.11242  3
## 8 2020-05-22  15 145.11242  3
## 9 2020-05-21  16 155.11242  3
## 10 2020-05-22  17 165.11242  3
## 11 2020-05-21   8  75.11242  3
## 12 2020-05-22   9  85.11242  3
## 13 2020-05-21  10  95.11242  3
## 14 2020-05-22  11 105.11242  3
## 15 2020-05-21  12 115.11242  3
## 16 2020-05-22  13 125.11242  3
## 17 2020-05-21  14 135.11242  3
## 18 2020-05-22  15 145.11242  3
## 19 2020-05-21  16 155.11242  3
## 20 2020-05-22  17 165.11242  3

```

Aplicando a função

```

Presc_temp<-purrr::pmap_dfr(list(lambda = dados$Lambda,
                                    mu =dados$mu,hora = dados$Hmd),

```

```

        filtra)
Prescritivo<-purrr::pmap_dfr(Presc_temp, otimiza)
Prescritivo %>% head()

##      Fluxo Taxa.média.atendimento Tempo.media.de.espera.na.fila.minutos Hora
## 1 75.11242                      3                               16.415    8
## 2 75.11242                      3                                6.209    8
## 3 75.11242                      3                                3.122    8
## 4 75.11242                      3                                1.742    8
## 5 75.11242                      3                                1.020    8
## 6 75.11242                      3                                0.612    8
##   Qtd_op
## 1     26
## 2     27
## 3     28
## 4     29
## 5     30
## 6     31

```

Na função criada, teremos um range de possíveis valores ótimos de operadores com seus respectivos tempos médios de espera. Para simplificar, pegaremos os valores ótimos que sejam inferiores a 1 minuto

```

Prescritivo %>%filter(Tempo.media.de.espera.na.fila.minutos<=1) %>%
  group_by(Hora)%>%
  summarise(Tempo_medio = mean(Tempo.media.de.espera.na.fila.minutos),Qtd_op = median(Qtd_op))

## `summarise()` ungrouping output (override with `groups` argument)

## # A tibble: 10 x 3
##       Hora Tempo_medio Qtd_op
##   <int>     <dbl>   <dbl>
## 1     8     0.0384   50.5
## 2     9     0.057    52
## 3    10     0.0535   54
## 4    11     0.0769   55.5
## 5    12     0.0742   57.5
## 6    13     0.106    59
## 7    14     0.151    60.5
## 8    15     0.157    62.5
## 9    16     0.234    64
## 10   17     0.275    66

```