Análise da fila de atendimento (restrita ao horário de pico 18h)

Carga e Transformação

```
#le dados de entrada
data <- read.csv("data.csv", stringsAsFactors = F, sep = ";")

#dias com ocupação acima de 90% durante os dois intervalos estudados
highDays <- c("16-02-18", "16-02-19", "16-02-22", "16-02-23", "16-02-24", "16-02-25", "16-03-08", "16-0
#data <- head(data, 10)
#nrow(data)

#transforma data/hora de entrada em timestamp
data$arrivalTimestamp <- as.POSIXct(strptime(with(data, paste(Data, Hora.Chegada)), "%Y/%m/%d %H:%M:%S"
data$servStartTimestamp <- as.POSIXct(strptime(with(data, paste(Data, Hora.Chamada)), "%Y/%m/%d %H:%M:%

#calcula o turno
data$turno <- as.factor(floor(as.numeric(format(data$arrivalTimestamp, "%H"))/6))

data$prefer <- substr(data$Chamada, 2, 2)=="P"

#transforma Guiche em variavel categórica
data$Guiche <- as.factor(data$Guiche)

#extrai tipo de atendimento
data$Tipo <- as.factor(
  substr(data$Chamada, 1, attr(regexpr("^[A-Z]{1,2}", data$Chamada), "match.length")))
# table(data$Tipo)
# midpoints <- barplot(as.data.frame(table(data$Tipo))$Freq,
#         names.arg=as.data.frame(table(data$Tipo))$Var1)
# text(midpoints, 200, labels=as.data.frame(table(data$Tipo))$Freq)

#calcula o tempo na fila
data$waitingTime <- data$servStartTimestamp - data$arrivalTimestamp

#calcula o tempo de atendimento
data <- data[with(data, order(Guiche, servStartTimestamp)), ]
data <- ddply(data, .(format(servStartTimestamp, "%Y/%m/%d"), Guiche), mutate, servDuration = c(as.nume
data <- data[,-1]

#ordena por hora de chegada para cálculo da diferença de chegada
data <- data[order(data$arrivalTimestamp),]
data$timediff <- c(Inf, diff(data$arrivalTimestamp))

#filtra apenas os registros com tempo de servico válido
data <- data[!is.na(data$servDuration), ]

#todos os dados transformados
#write.csv(data, file="dataFull.csv", row.names = F)
```

```
#filtra apenas as chegadas occoridas entre 18:00:00 e 18:59:59
data <- data[format(data$arrivalTimestamp, "%H")=="18", ]
write.csv(data, file="data18.csv", row.names = F)
```

Percentual de atendimentos preferenciais: 0.012394

Distribuição de chegada

```
chegadas1 <-
  ggplot(data, aes(timediff, colour = format(arrivalTimestamp, "%H"))) +
  geom_freqpoly(aes(y = (..count..)/sum(..count..)), binwidth = 30) +
  xlim(0, 600) +
  ylab("Frequência Percentual") +
  xlab("Intervalo em segundos") +
  labs(colour = "Hora") +
  scale_y_continuous(labels = percent_format())
ggsave(filename = "chegadas118.png", plot = chegadas1)
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```

```
write.csv(data[,"timediff"], file="chegadasFiltro18.csv")
write.csv(data[format(data$arrivalTimestamp, "%y-%m-%d") %in% highDays,"timediff"], file="chegadasFiltr
data[format(data$arrivalTimestamp, "%y-%m-%d") %in% highDays,"timediff"]
```

```
##   [1] 252  91  53  56  17  62   7  55  11  24  20  61 149  34  33  43  27
##  [18]  50  26  57   4   4  97   6   9  19  32  15   5 109   5  32  12  14
##  [35]  36  86  19  50  89  76  69  18  10  25  26  44   8 115  18  87  18
##  [52]  30  89  65   5  37  71   8  40  25  22   8 156   6   3   6   3  11
##  [69]  14  13  16  33  16  34  17  23 159  17  15  30  11   8  84 112  39
##  [86]  27   6  42   7   5   9 112   8  13   8   6   2  15  16  29 539 130
## [103]   7 248   6  57 156  68  13 154  44   7   5  38  71 115  67   6  56
## [120]  35   5  78  35  22  54  20  13 126   6  35  46  17  21  25  32   5
## [137]  10  27  12   5  28  31  27 103 113  29  24   6   5   3  12  16  16
## [154]  89  55   8   6  10  23   7   9  40  34  59  75  20   8  56   7   5
## [171]  28  16  18  16  73  16  64  21  71 201   4 256  33   3  24  52 100
## [188]   5   3   4  34  69  20  33  26  96  85  10 105  35  39  47  90  63
## [205]  16  42  10 116  16   8  19  44  35  26  38  46  40  44  41  68 166
## [222]  68  28  22   5  61  64  28  12 175   8  11  22   5   5  18  13  21
## [239]  13   3   4   6   6   5  40  37  16  56  12  40  56   8  27  47  49
## [256]  26  11   6  19   3   3   5  27   6   5   7  16  49  59  19  10  12
## [273]  10  38  48  11   8  21   3  31  55   3  40   6  56   6   7 391   9
## [290] 214  73  38   4   6  11  20  41  93  44  30  20  80  22  77  94  52
## [307]  68   4  35  25  10 124  37  14  30 215  28   3  26  22  10  17   7
## [324]  91  36  73   7  10  25 108  13  39  52  30  83  86   7  93  81  13
## [341]  38   5  50  12   6   6   6  18  48  63  61   5  12   7   8  16   8
```

```
## [358]  39   6 172  14   8  39   7  23   4   8   5  31   5   6  11   5  21
## [375]  18  81   7  20   7   9  10   9  61   6   4   5   5   9   8   6 197
## [392]  17  49  23  38  62  90  83  33  47 197  19  96  27  41  17  49  87
## [409]  97  46 181  39  42 154  48  85  11   8  33  32   5  24   8  71   9
## [426]  39  87  60  44  11  90  13  17  28   8  64   3  78   6  12   7   6
## [443]  29  25  12  60  96  20   6   5  46  13  67  33  11   9   7  36   6
## [460]  37  15  66  12   4   4   4   4   8  11   5   9  88  10  46  27  29
## [477]  89   9  67  10  39   6 154  52  10 120   7  84  94 170   5 280  59
## [494]   5   6   4  39   6 139  30  36 113  37 118   4  62  30  32   8  12
## [511]   6  10  11  14  71  69  12 118   9   9  65  82  23 141  29   9  18
## [528]  15   7  15  94   5  10   4 107   8  16  21   5  16  25  70  22 130
## [545]   6  16  11  96  50   9  49  10  14  60  73   2   9   4  61   5  34
## [562]  37  44   7   3   5  87  14  12 323 480   8  20  74  55  79  42  58
## [579]  70  85 195 149  17  84  32  48  68  78   4  71  59  23   8  59  15
## [596]  57  71  42  42   5  35  33 136  80  18  46  31  16  28   9  97  42
## [613] 109  69  89  58 104   9  36  75 121  38  39  10  25   5 320 235 124
## [630]  70 164 456 104  91  71   7  56 360 206 235   4  34  83 113 430 156
## [647] 303  15 128   9  32   6 183 144  45  30 208 285  40  18  65 316 132
## [664]  16  16  13  59 152 250  16 268  30  99  63  15  61  33  17  18 191
## [681] 115   9  52  35  48  15 119  18  40   9  46 126  84   8  36  58  27
## [698]  29 249  51 223  48  43  27  31 315  84  44  17  15 199  49  19  12
## [715]  84 158 171  66   5  32  50 103   4  52  33  20 133  20  85   5 155
## [732]   4 117   5  15  93  22   7 105  13   8 216  63  87  91  70  13   5
## [749]  27  12  41  40  54
```

```
chegadas2 <- ggplot(data, aes(timediff)) +
  geom_histogram(aes(y = (..count..)/sum(..count..)), binwidth = 30) +
  #geom_freqpoly(aes(y = (..count..)/sum(..count..)), binwidth = 30) +
  xlim(0, 600) +
  ylab("Frequência Percentual") +
  xlab("Intervalo em segundos") +
  scale_y_continuous(labels = percent_format())
ggsave(filename = "chegadas218.png", plot = chegadas2)
```

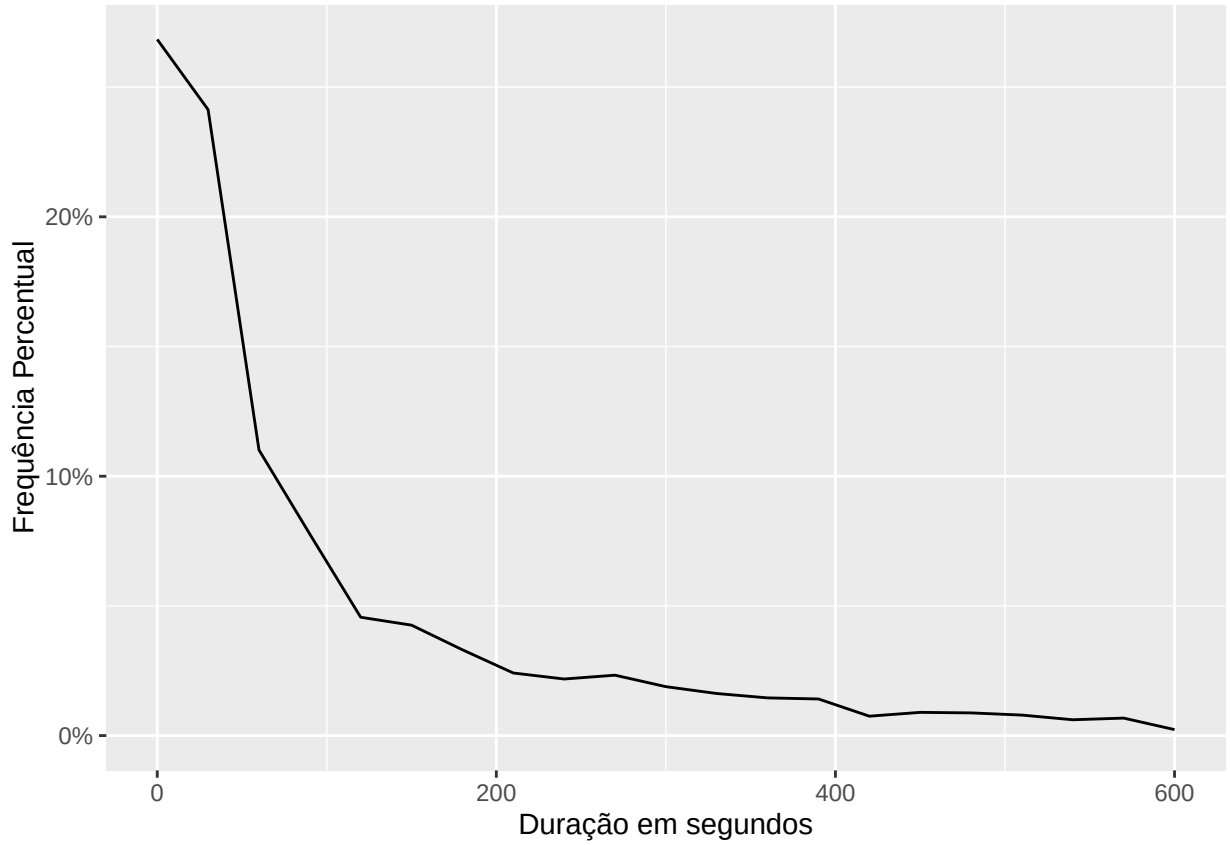
```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
#-> para os dados gerais, EasyFit retornou Fatigue Life (Birnbaum-Saunders) Distribution com shape=1.94
test <- rbisa(10000, scale=50.405, shape=1.9434)
ggplot(data.frame(test), aes(test)) +
  geom_freqpoly(aes(y = (..count..)/sum(..count..)), binwidth = 30) +
  xlim(0, 600) +
  ylab("Frequência Percentual") +
  xlab("Duração em segundos") +
  scale_y_continuous(labels = percent_format())
```

```
## Warning: Removed 510 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_path).
```



Sumarização e visualização

Chegadas por hora

```
#sumarização de chegadas por hora
chegadas.hora.dia <- ddply(data, .(hora=format(arrivalTimestamp, "%H"), dia=format(arrivalTimestamp, "%
chegadas.hora.dia.medio <- ddply(chegadas.hora.dia, .(hora), summarize, mediaDiff=mean(meanTimeDiff), nu

#visualiza quantidade de chegadas por hora
chegadas3 <-
  ggplot(chegadas.hora.dia.medio, aes(x=factor(hora), y=mediaAtendDia)) +
  geom_bar(stat = "identity") +
  ylab("Quantidade de média novos clientes") +
  xlab("Horas do dia")
ggsave(filename = "chegadas318.png", plot = chegadas3)
```

```
## Saving 6.5 x 4.5 in image
```

```
#visualiza intervalo medio de chegada por hora
chegadas4 <-
  ggplot(chegadas.hora.dia.medio, aes(x=factor(hora), y=as.POSIXct(mediaDiff, origin = "1970-01-01", tz=
  geom_bar(stat = "identity") +
```

```
  ylab("Intervalo médio entre chegadas (hh:mm:ss)") +
  xlab("Horas do dia") +
  scale_y_datetime(labels = date_format("%H:%M:%S"))
ggsave(filename = "chegadas418.png", plot = chegadas4)
```

```
## Saving 6.5 x 4.5 in image
```

```
quantile(floor(data$waitingTime/60), probs = seq(0, 1, 0.1), na.rm = T)
```

```
## Time differences in secs
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##    0    0    0    1    2    4    8   12   16   22   32
```

```
#percentual de registros com tempo de espera < 1min
sum(data$waitingTime<=60, na.rm = T)/sum(!is.na(data$waitingTime))*100
```

```
## [1] 29.68037
```

```
#percentual de registros com tempo de espera < 30min
sum(data$waitingTime<=60*30, na.rm = T)/sum(!is.na(data$waitingTime))*100
```

```
## [1] 99.54338
```

```
#indica atendimentos imediatos
data$atendImediato <- data$waitingTime <= 60

tempo_fila_1 <-
  ggplot(data[!data$atendImediato,], aes(floor(waitingTime/60))) +
  geom_histogram(aes(y = (..count..)/sum(..count..)), binwidth = 1, boundary = 1) +
  xlim(0, 60) +
  ylab("Frequência Percentual") +
  xlab("Duração em minutos") +
  scale_y_continuous(labels = percent_format())
ggsave(filename = "tempo_fila_118.png", plot = tempo_fila_1)
```

```
## Saving 6.5 x 4.5 in image
```

```
#atendimentos sem duração informada e com duração menor que 1min
#hist(data[data$servDuration < 60, "servDuration"])
#sum(data$servDuration < 60, na.rm = T)

#data$validServTime <- TRUE
sum(data$servDuration > 7200, na.rm = T)
```

```
## [1] 5
```

```
nrow(data)-sum(!is.na(data$servDuration))
```

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

```
data$validServTime <-
  with(data, !is.na(servDuration)
       #& servDuration >= 60 # opcionalmente  excluios atendimentos menores que 1 min (desistência?)
       & servDuration < 7200)
sum(data$validServTime)
```

```
## [1] 1528
```

```
#quantis da duracao
quantile(data[data$validServTime, "servDuration"], probs = seq(from=0.1, to=1, by=0.1), na.rm = T)
```

```
##    10%    20%    30%    40%    50%    60%    70%    80%    90%   100%
##   34.7  101.4  185.0  253.0  321.0  391.0  487.0  667.0 1066.9 7167.0
```

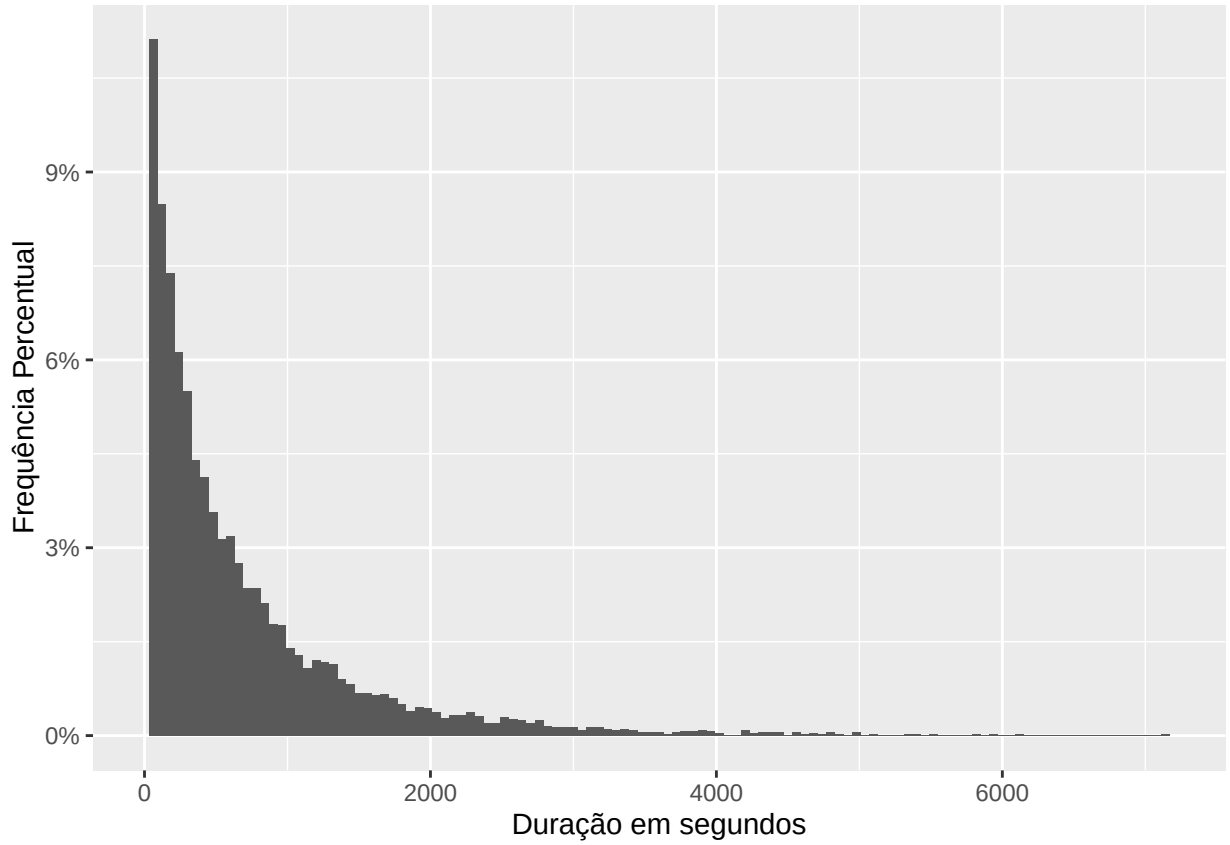
```
duracao_atend <-
  ggplot(data[data$validServTime, ], aes(floor(servDuration/60))) +
  geom_histogram(aes(y = (..count..)/sum(..count..)), binwidth = 1) +
  xlim(0, 60) +
  ylab("Frequência Percentual") +
  xlab("Duração em minutos") +
  scale_y_continuous(labels = percent_format(), limits = c(0,0.1))
ggsave(filename = "duracao_atend18.png", plot = duracao_atend)
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 7 rows containing non-finite values (stat_bin).
```

```
write.csv(data[data$validServTime, "servDuration"], file="duracao_atend18.csv")

#Weibull (data without atend < 60)
#test <- floor(rweibull(10000, 0.85283, scale=705.25) + 60)

#Log normal (data without atend < 60)
#test <- floor(rlnorm(10000, meanlog=6.0411, sdlog=1.1442) + 41.277)

#Dagum ("all" data)
test <- floor(rdagum(10000, scale = 818.72, shape1.a=0.39952, shape2.p=1.8476))

#Weibull ("all" data)
test <- floor(rweibull(10000, shape=0.8045, scale=570.75))

ggplot(data.frame(test), aes(test)) +
  geom_histogram(aes(y = (..count..)/sum(..count..)), binwidth = 60) +
  xlim(0, 7200) +
  ylab("Frequência Percentual") +
  xlab("Duração em segundos") +
  scale_y_continuous(labels = percent_format())
```

```
## Warning: Removed 7 rows containing non-finite values (stat_bin).
```




```
citation(package = "base", lib.loc = NULL)
```

```
##
## To cite R in publications use:
##
##   R Core Team (2016). R: A language and environment for
##   statistical computing. R Foundation for Statistical Computing,
##   Vienna, Austria. URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2016},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please
## cite it when using it for data analysis. See also
## 'citation("pkgname")' for citing R packages.
```