

# Análise da fila de atendimento

## Carga e Transformação

```
#le dados de entrada
data <- read.csv("data.csv", stringsAsFactors = F, sep = ";")
#data <- head(data, 10)

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

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

data2 <- read.csv("trab1.csv", stringsAsFactors = F, sep = ";")
data2$chegada.tmstamp <- as.POSIXct(strptime(with(data2, paste(Data, Hora.Chegada)), "%d/%m/%Y %H:%M:%S"))
data2$chamada.tmstamp <- as.POSIXct(strptime(with(data2, paste(Data, Hora.Chegada)), "%d/%m/%Y %H:%M:%S"))
data2$duracao.atend <- as.POSIXct(strptime(data2$Tempo.de.atendimento, "%H:%M:%S"))
data2$duracao.minutos <- as.numeric(data2$duracao.atend-as.POSIXlt("2016-05-12 00:00:00 BRT"))/60
```

## Distribuição de chegada

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

#ajusta as diferenças de chegada que tenham mais de 6 horas (dia anterior) usando a respectiva moda
temp <- data[data$timediff<21600, ]
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
temp <- ddply(temp, .(hora=format(fulldate, "%H")), summarize, moda=Mode(timediff))
data[data$timediff>21600, "timediff"] <- temp[match(format(data[data$timediff>21600, "fulldate"], "%H"),
remove(temp, Mode)
#summary(data$timediff)

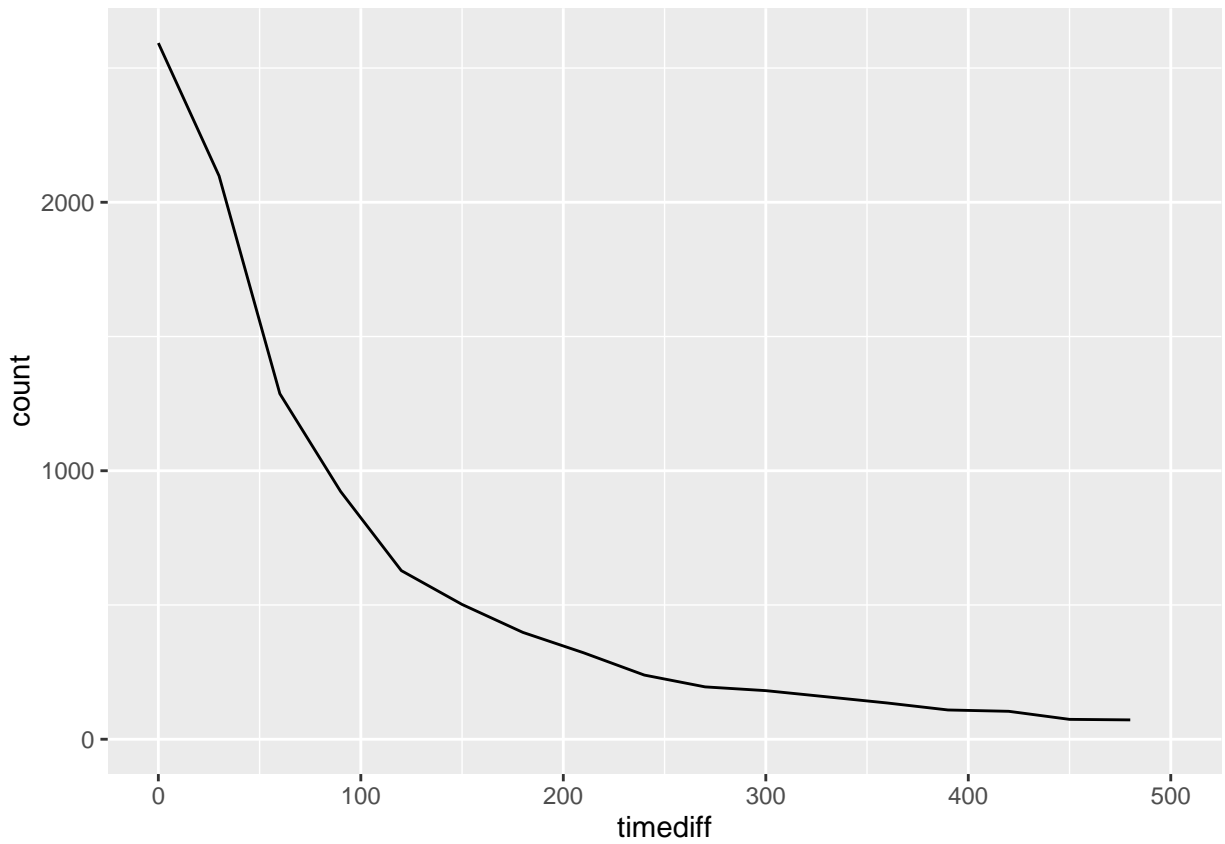
#cria um arquivo com 5000 amostras para ajuste de curva no easyFit
#write.csv(sample(data[, "timediff"], 5000), file="timediff.csv")

#hist(data[, "timediff"], breaks=seq(0, 7200, 30), xlim=c(0,1200))
#hist(data[format(data$fulldate, "%H")=="13", "timediff"], breaks=seq(0, 7200, 30), xlim=c(0,1200))
#hist(data[format(data$fulldate, "%H")=="19", "timediff"], breaks=seq(0, 7200, 30), xlim=c(0,1200))

ggplot(data, aes(timediff)) +
  geom_freqpoly(binwidth = 30) + xlim(0,500)
```

```
## Warning: Removed 635 rows containing non-finite values (stat_bin).
```

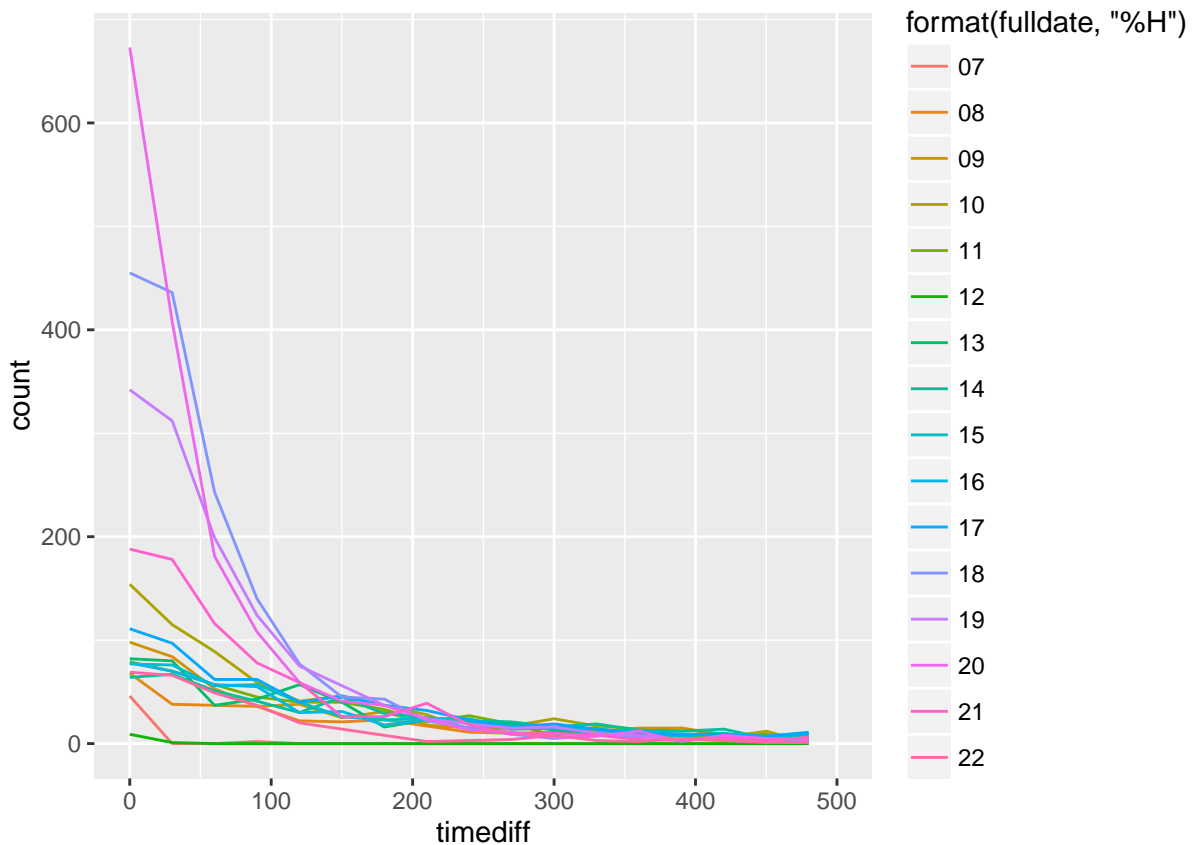
```
## Warning: Removed 3 rows containing missing values (geom_path).
```



```
ggplot(data, aes(timediff, colour = format(fulldate, "%H"))) +  
  geom_freqpoly(binwidth = 30) + xlim(0,500)
```

```
## Warning: Removed 635 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 48 rows containing missing values (geom_path).
```

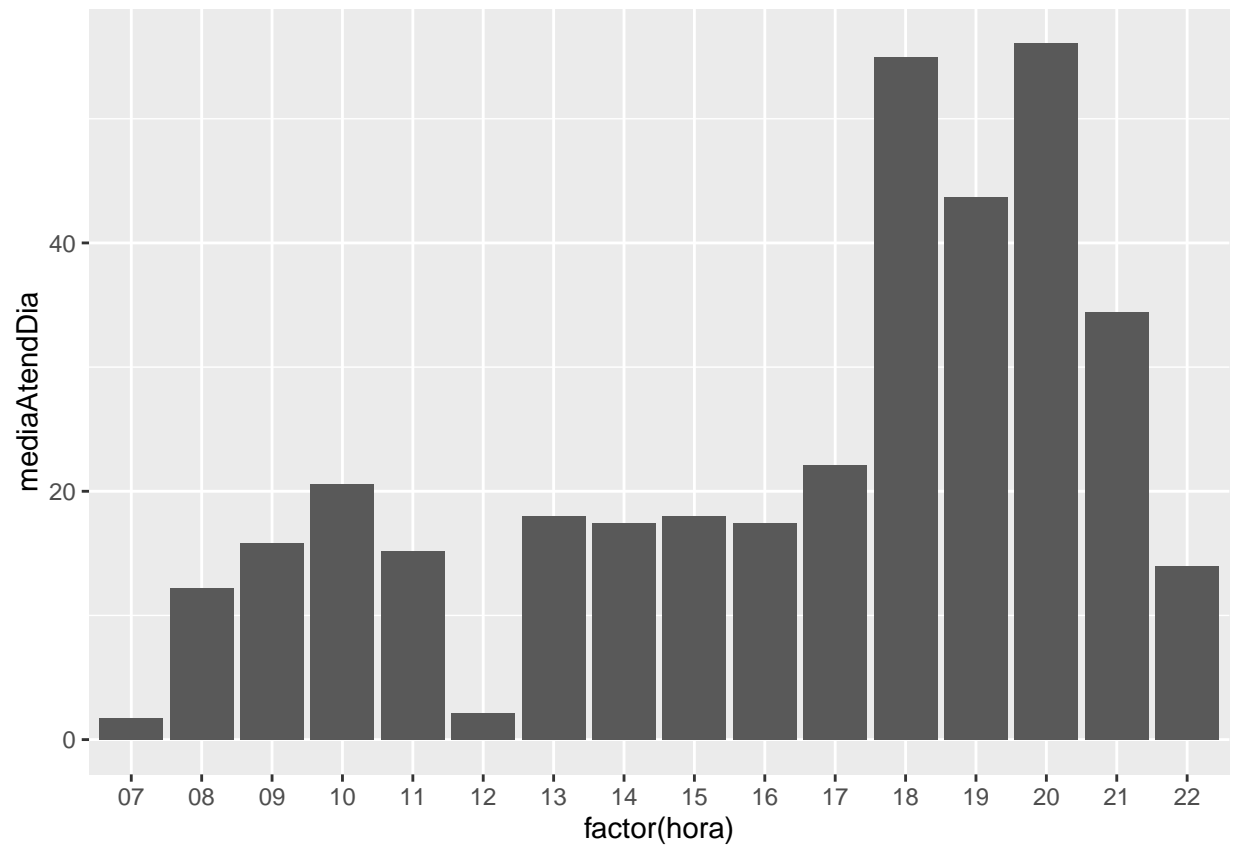


## Sumarização e visualização

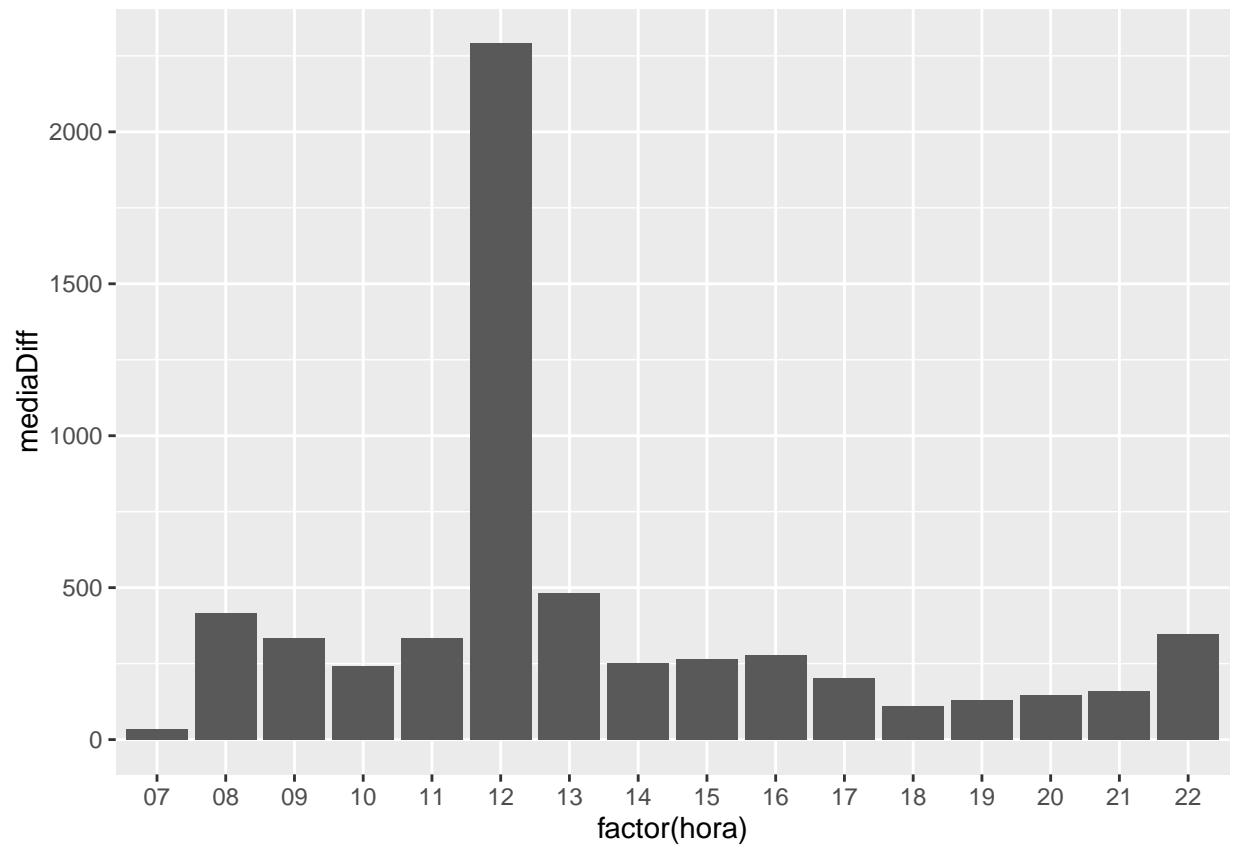
### Atendimentos por hora

```
#sumarização de atendimentos por hora
atend.hora.dia <- ddply(data, .(hora=format(fulldate, "%H"), dia=format(fulldate, "%d-%m-%Y")), summarize,
atend.hora.dia.medio <- ddply(atend.hora.dia, .(hora), summarize, mediaDiff=mean(meanTimeDiff), numDias=

#visualiza chegadas por hora
c <- ggplot(atend.hora.dia.medio, aes(x=factor(hora), y=mediaAtendDia))
c + geom_bar(stat = "identity")
```



```
#visualiza intervalo medio de chegada por hora  
c <- ggplot(atend.hora.dia.medio, aes(x=factor(hora), y=mediaDiff))  
c + geom_bar(stat = "identity")
```



```
#atendimentos sem duração informada:
nrow(data2)-sum(!is.na(data2$duracao.minutos))
```

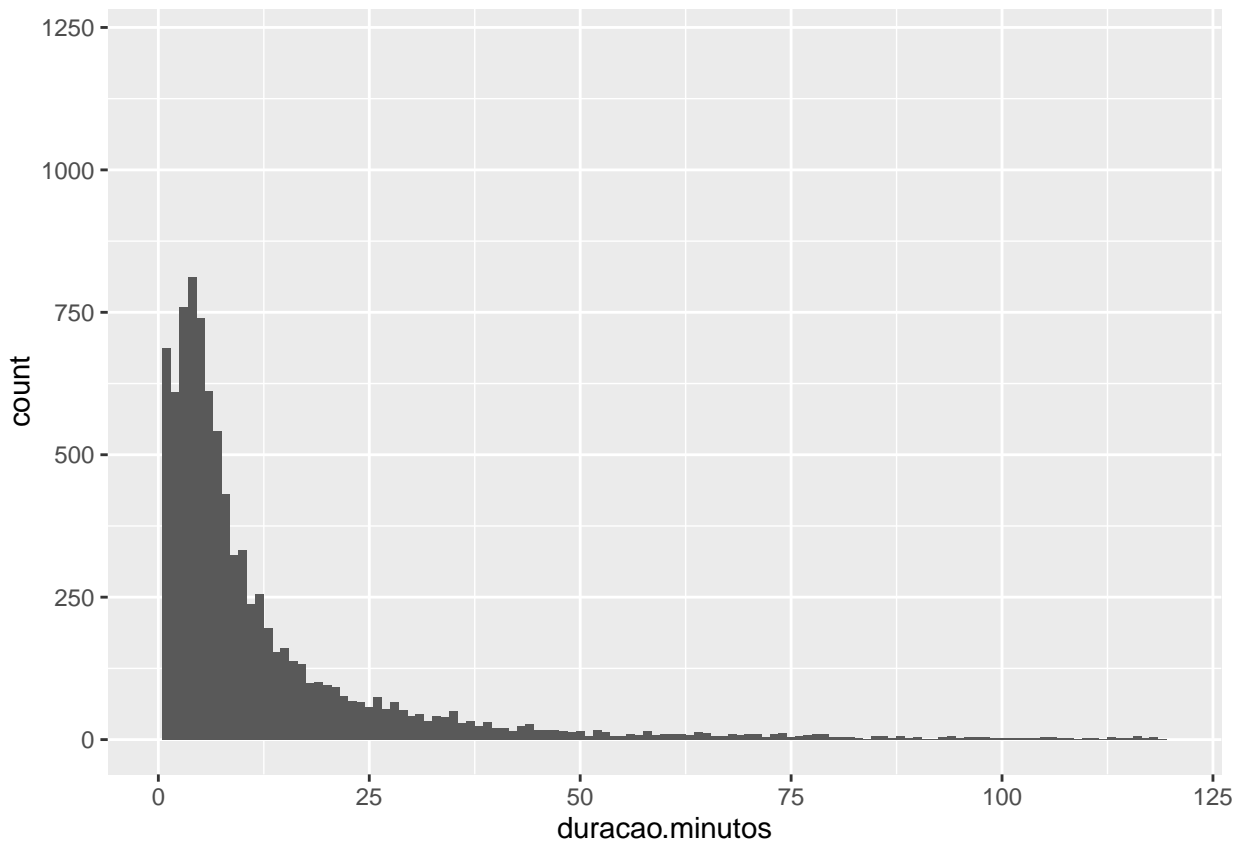
```
## [1] 575
```

```
#quantis da duracao
quantile(data2$duracao.minutos, probs = seq(from=0.1, to=1, by=0.1), na.rm = T)
```

```
##          10%          20%          30%          40%          50%          60%
##  0.3666667  1.7200000  3.2966667  4.5666667  6.0666667  7.9500000
##          70%          80%          90%         100%
## 11.1866667 16.9633333 31.0666667 494.4666667
```

```
ggplot(data2, aes(duracao.minutos)) +
  geom_histogram(binwidth = 1) + xlim(0,120)
```

```
## Warning: Removed 716 rows containing non-finite values (stat_bin).
```



data2\$duracao.minutos