Projeto1

Thiago Barral

1/23/2020

Carregando os pacotes necessários

```
library(readr)
library(ggplot2)
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
```

```
library(gplots)
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(DMwR)
```

```
## Loading required package: lattice
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(e1071)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(caTools)
library(rmarkdown)
library(knitr)
library(tinytex)
```

Carregando os dados

Carregando os dados de trabalho. Fazendo uma análise inicial dos dados.

```
train_sample <- read_csv('train_sample.csv', col_names = TRUE)
```

```
## Parsed with column specification:
## cols(
##   ip = col_double(),
##   app = col_double(),
##   device = col_double(),
##   os = col_double(),
##   channel = col_double(),
##   click_time = col_datetime(format = ""),
##   attributed_time = col_datetime(format = ""),
##   is_attributed = col_double()
## )
```

```
str(train_sample)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 100000 obs. of  8 variables:
##  $ ip             : num  87540 105560 101424 94584 68413 ...
##  $ app            : num  12 25 12 13 12 3 1 9 2 3 ...
##  $ device         : num  1 1 1 1 1 1 1 1 2 1 ...
##  $ os             : num  13 17 19 13 1 17 17 25 22 19 ...
##  $ channel        : num  497 259 212 477 178 115 135 442 364 135 ...
##  $ click_time     : POSIXct, format: "2017-11-07 09:30:38" "2017-11-07 13:40:27" ...
##  $ attributed_time: POSIXct, format: NA NA ...
##  $ is_attributed  : num  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ip = col_double(),
##   ..   app = col_double(),
##   ..   device = col_double(),
##   ..   os = col_double(),
##   ..   channel = col_double(),
##   ..   click_time = col_datetime(format = ""),
##   ..   attributed_time = col_datetime(format = ""),
##   ..   is_attributed = col_double()
##   .. )
```

```r
summary(train_sample)
```

```
##        ip              app              device              os
##  Min.   :     9   Min.   :  1.00   Min.   :   0.00   Min.   :  0.00
##  1st Qu.: 40552   1st Qu.:  3.00   1st Qu.:   1.00   1st Qu.: 13.00
##  Median : 79827   Median : 12.00   Median :   1.00   Median : 18.00
##  Mean   : 91256   Mean   : 12.05   Mean   :  21.77   Mean   : 22.82
##  3rd Qu.:118252   3rd Qu.: 15.00   3rd Qu.:   1.00   3rd Qu.: 19.00
##  Max.   :364757   Max.   :551.00   Max.   :3867.00   Max.   :866.00
##
##     channel        click_time                  attributed_time
##  Min.   :  3.0   Min.   :2017-11-06 16:00:00   Min.   :2017-11-06 17:19:04
##  1st Qu.:145.0   1st Qu.:2017-11-07 11:34:09   1st Qu.:2017-11-07 11:50:27
##  Median :258.0   Median :2017-11-08 07:07:50   Median :2017-11-08 06:43:39
##  Mean   :268.8   Mean   :2017-11-08 06:29:52   Mean   :2017-11-08 07:04:12
##  3rd Qu.:379.0   3rd Qu.:2017-11-09 02:06:01   3rd Qu.:2017-11-09 01:42:52
##  Max.   :498.0   Max.   :2017-11-09 15:59:51   Max.   :2017-11-09 15:28:15
##                                                NA's   :99773
##  is_attributed
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.00227
##  3rd Qu.:0.00000
##  Max.   :1.00000
##
```

```r
table(train_sample$is_attributed)
```

```
##
##     0     1
## 99773   227
```

```r
prop.table(table(train_sample$is_attributed))
```

```
##
##       0       1
## 0.99773 0.00227
```

Limpando os dados

A variável 'os' representa a mesma coisa que a coluna device logo será retirada. A variável 'ip' serve simplesmente para identificação então não influenciará no estudo. A variável 'attributed_time' representa o horário no qual o usuário fez o download, então também será retirado pois não é representativa, visto que já temos a variável 'click_time'.

```r
train_sample$os <- NULL
train_sample$attributed_time <- NULL
train_sample$ip <- NULL
```

Mudando a variável do tempo

A variável 'click_time' tem informação da data e da hora exata que o usuário entrou no site, porém eu acretido que somente a hora seja um fator de relevância, visto que é algo que não e afetado pela sazonalidade e a exatidão de minutos e segundos não se faz necessária.

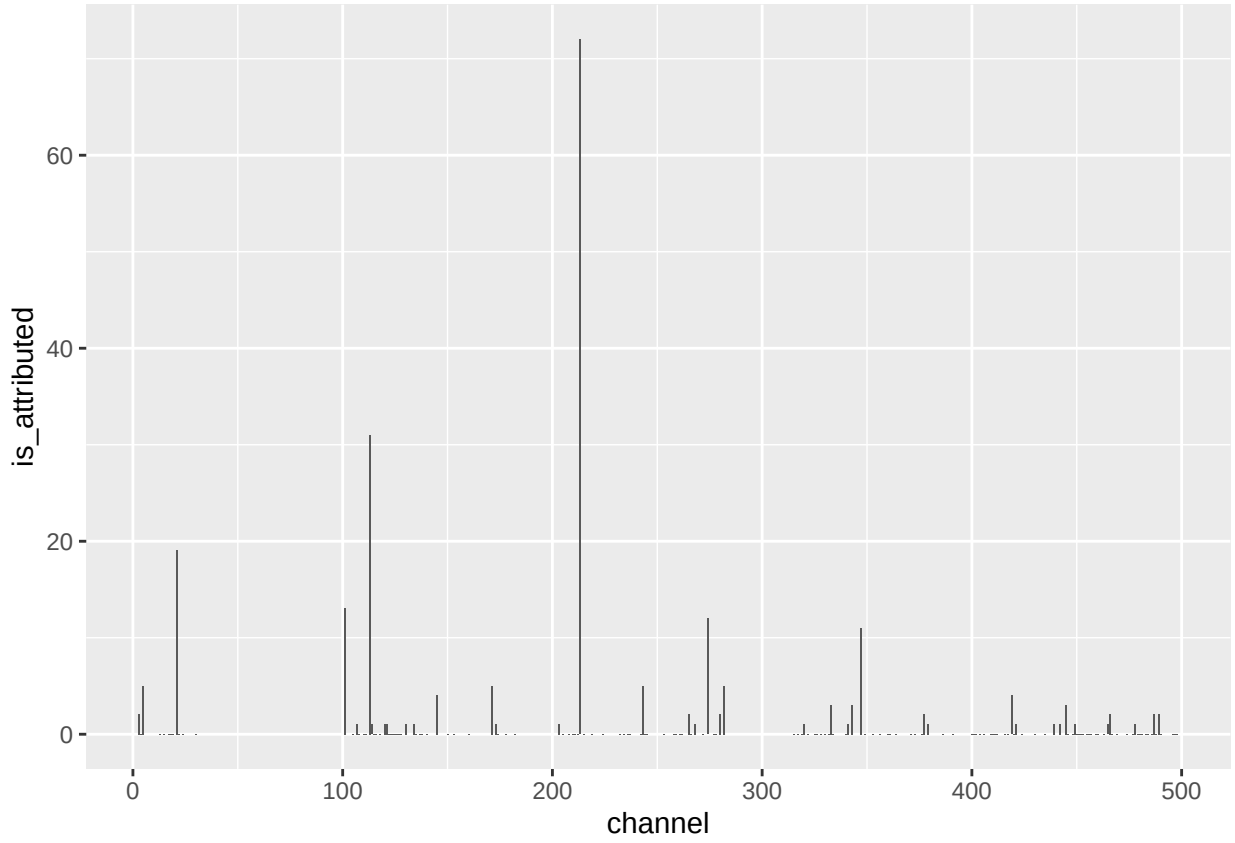
```
train_sample$hora <- hour(train_sample$click_time)
train_sample$click_time <- NULL
```

Gráficos

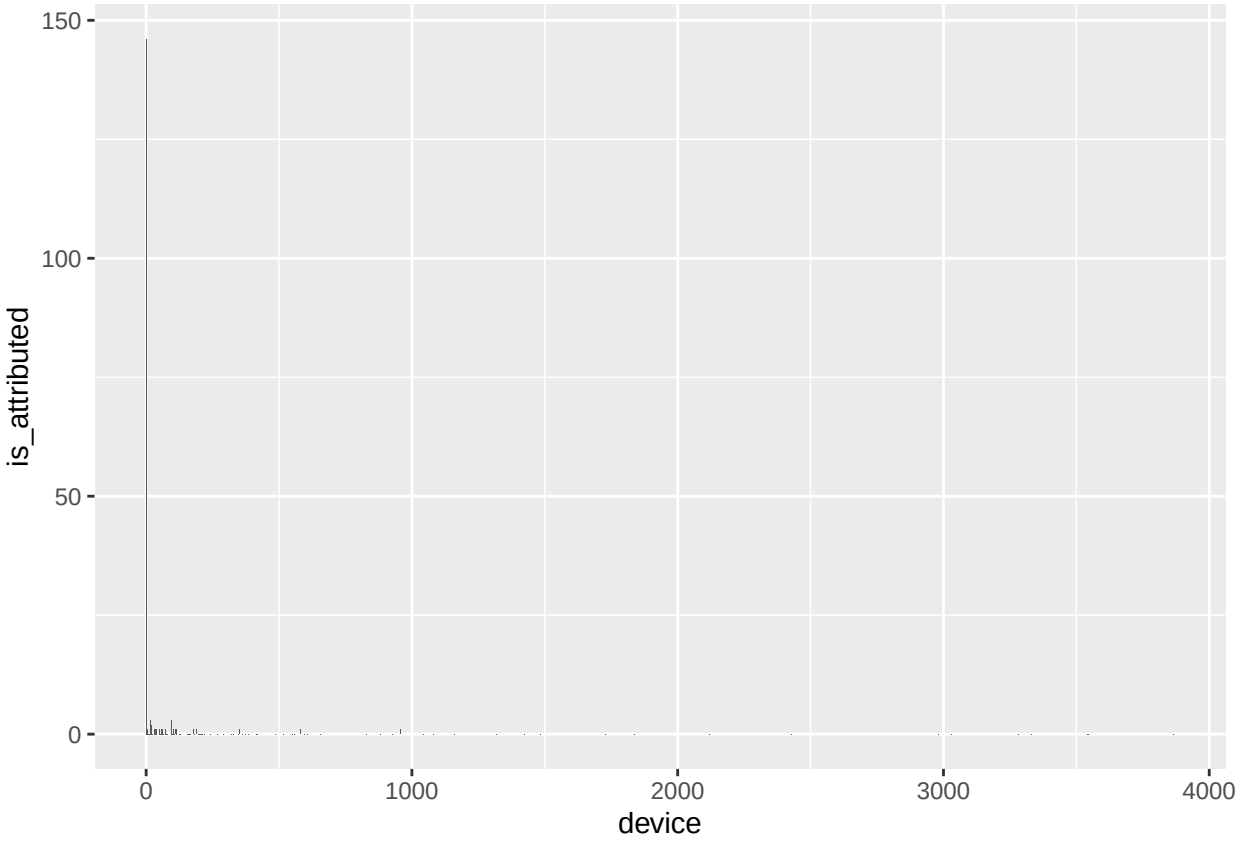
Análise visual para entender o comportamento das variáveis.

Transformando variáveis

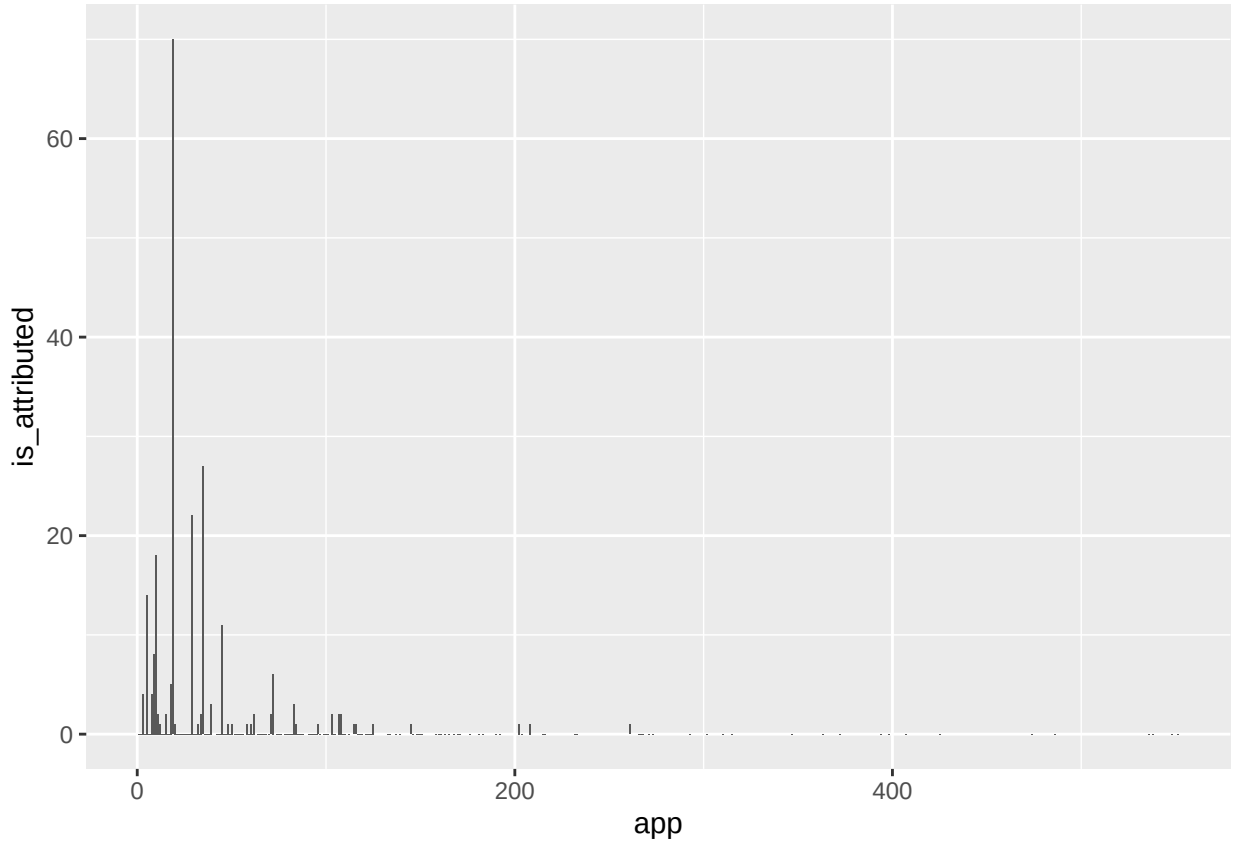
```
ggplot(train_sample, aes(x = channel, y = is_attributed)) +
  geom_bar(stat = 'identity')
```




```
ggplot(train_sample, aes(x = device, y = is_attributed)) +
  geom_bar(stat = 'identity')
```

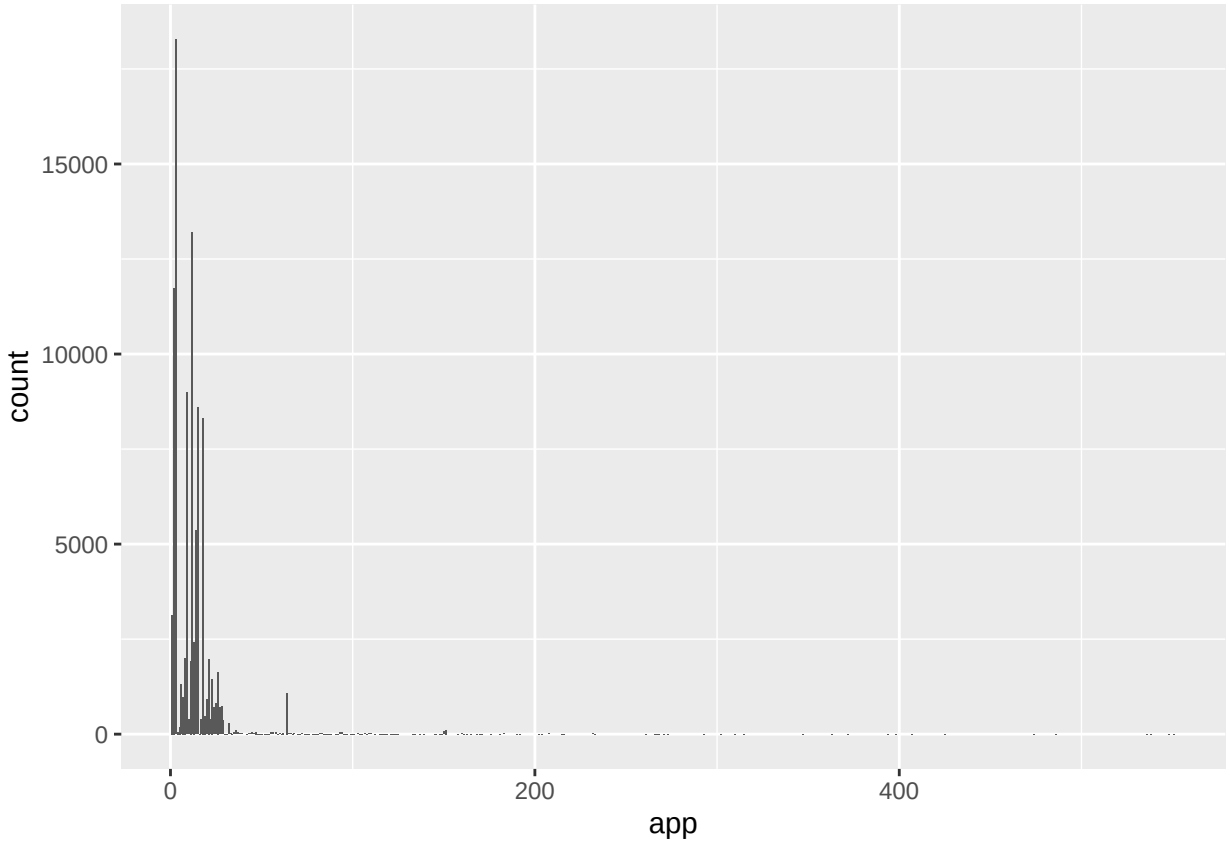



```
ggplot(train_sample, aes(x = app, y = is_attributed)) +
  geom_bar(stat = 'identity')
```



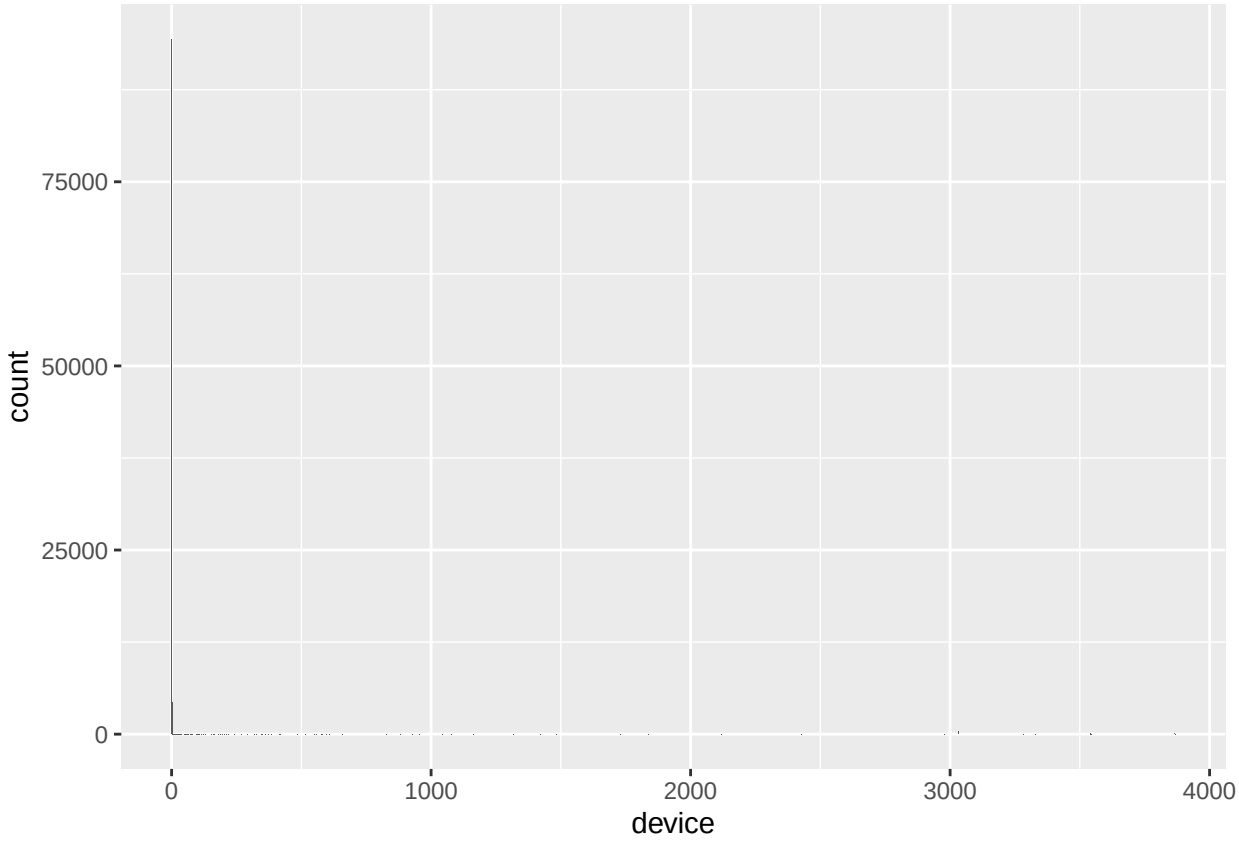

```
ggplot(train_sample, aes(x=app)) +
  geom_histogram(stat = 'count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



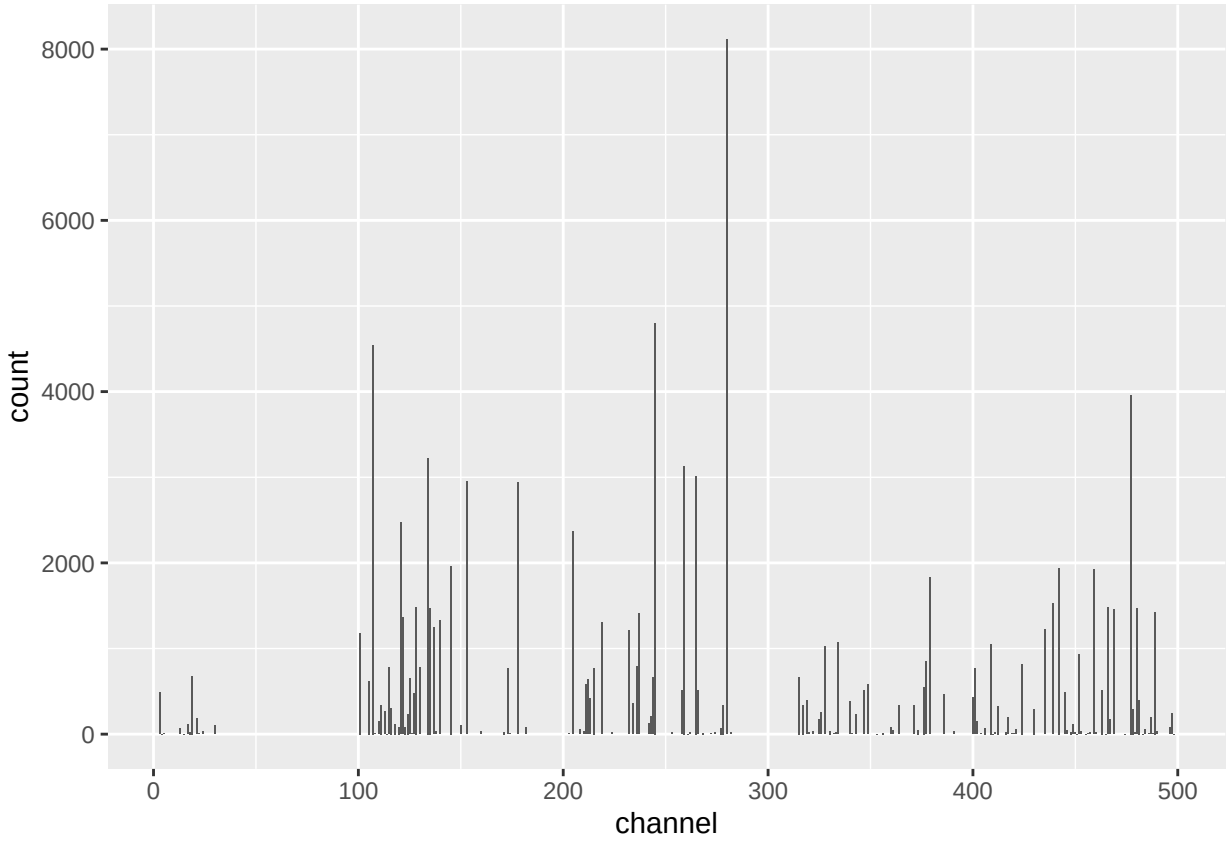

```
ggplot(train_sample, aes(x=device)) +
  geom_histogram(stat = 'count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



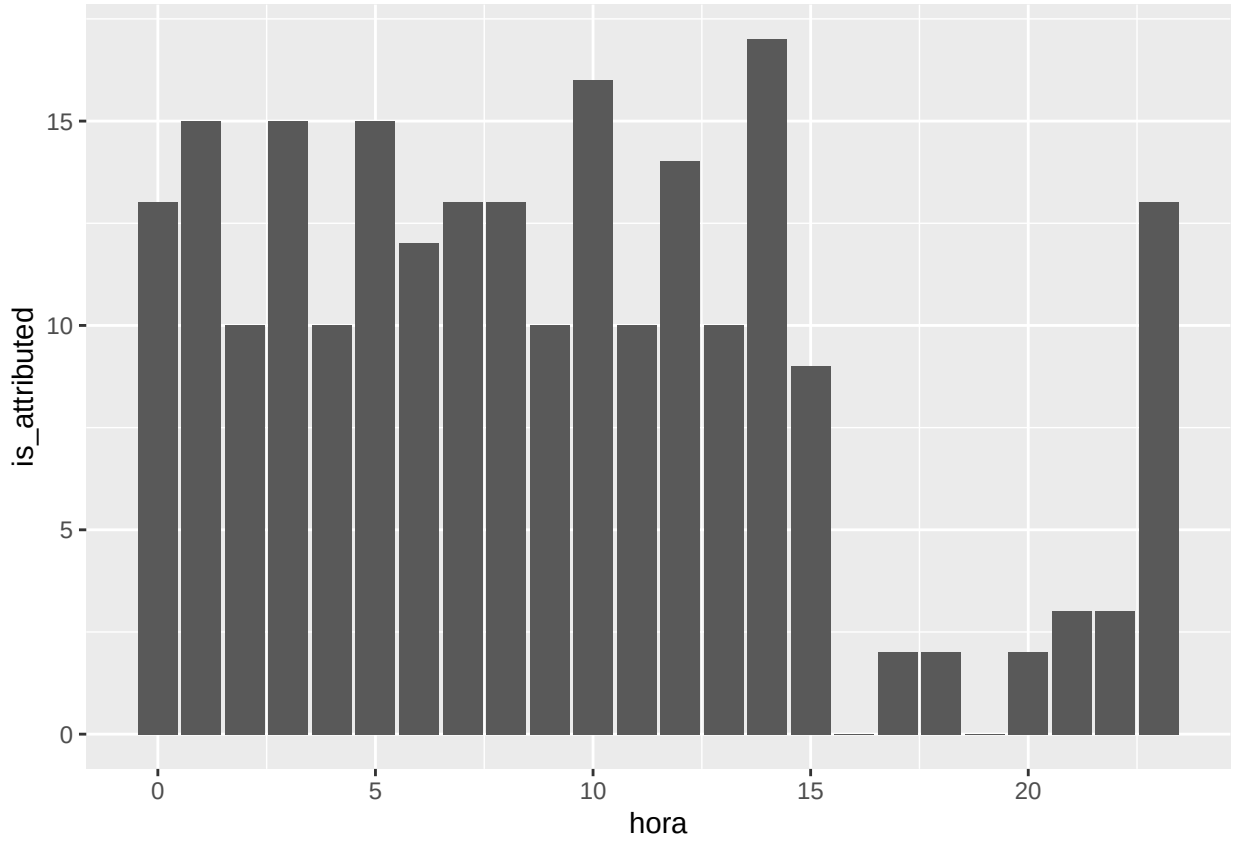

```
ggplot(train_sample, aes(x=channel)) +
  geom_histogram(stat = 'count')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
ggplot(train_sample, aes(x = hora, y = is_attributed)) +
  geom_histogram(stat = 'identity')
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



As variáveis 'is_attributed', 'app', 'device', 'channel' e 'hora' estão categorizadas como numéricas porém elas não representão números e sim classes, então transforei as 5 em factor.

```
train_sample$is_attributed <- as.factor(train_sample$is_attributed)
train_sample$app <- as.factor(train_sample$app)
train_sample$device <- as.factor(train_sample$device)
train_sample$channel <- as.factor(train_sample$channel)
train_sample$hora <- as.factor(train_sample$hora)
```

Dividindo dados em treino e teste

Realizei a divisão de 70% dos dados em dados de treino e os demais 30% em dados de teste.

```
train_sample$spl <- sample.split(train_sample$is_attributed, SplitRatio = 0.7)
table(train_sample$spl)
```

```
##
## FALSE  TRUE
## 30000 70000
```

```
treino_train <- dplyr::filter(train_sample, train_sample$spl == TRUE)
teste_train <- dplyr::filter(train_sample, train_sample$spl == FALSE)
table(treino_train$is_attributed)
```

```
##
##     0     1
## 69841   159
```

Balanceamento da variável alvo

Como os dados estão desbalenceados, ou seja, há muito mais dados referentes aos usuários que não baixaram o app, há a necessidade de se realizar o balanceamento para que não haja tendência no modelo.

```
train_sample_smote <- SMOTE(is_attributed~., as.data.frame(treino_train), perc.over = 5000, k = 5, perc
table(train_sample_smote$is_attributed)
```

```
##
##    0    1
## 7950 8109
```

```
respostas_train_teste <- teste_train$is_attributed
teste_train$is_attributed <- NULL
teste_train$spl <- NULL
```

Previsão

Utilizei o modelo Naive Bayes para prever quais os usuários são fradulentos.

```
nb <- naiveBayes(is_attributed~., treino_train)
prev <- predict(nb, teste_train)
```

Avaliação

O método utilizado para avaliação e o area under the curve (AUC).

```
roc(response =  as.numeric(respostas_train_teste), predictor = as.numeric(prev), auc = TRUE)
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
##
## Call:
## roc.default(response = as.numeric(respostas_train_teste), predictor = as.numeric(prev),     auc = TRU
##
## Data: as.numeric(prev) in 29932 controls (as.numeric(respostas_train_teste) 1) < 68 cases (as.numeri
## Area under the curve: 0.937
```