DeteccaoFraudePropaganda

Eduardo de Souza Dias

11/30/2019

Projeto 01 - Curso BigDataRAzure da DSA (parte da Formação cientista de dados)

Detecção de Fraudes em cliques de propaganda

```
#Definindo diretório de trabalho
setwd("C:/Cursos/FCD/01-BigDataRAzure/Cap20-ProjetosFeedback/Projeto01-DeteccaodeFaudePropaganda")

#Carregando alguns pacotes básicos
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(readr)
library(tidyr)
library(ggplot2)
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(lubridate)
```

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

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

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
source("Tools.R")

#Carregando o dataset de treino
df <- fread("train_sample.csv", colClasses=c(rep('factor', 5), 'character', rep('factor',2)))
View(df)
summary(df)
```

```
##        ip             app            device            os
##  5348   :  669   3      :18279   1      :94338   19     :23870
##  5314   :  616   12     :13198   2      : 4345   13     :21223
##  73487  :  439   2      :11737   0      :  541   17     : 5232
##  73516  :  399   9      : 8992   3032   :  371   18     : 4830
##  53454  :  280   15     : 8595   3543   :  151   22     : 4039
##  114276 :  219   18     : 8315   3866   :   93   10     : 2816
##  (Other):97378   (Other):30884   (Other):  161   (Other):37990
##     channel       click_time                  attributed_time  is_attributed
##  280    : 8114   Length:100000                         :99773   0:99773
##  245    : 4802   Class :character   2017-11-06 17:19:04:    1   1:  227
##  107    : 4543   Mode  :character   2017-11-06 21:30:47:    1
##  477    : 3960                      2017-11-06 23:16:28:    1
##  134    : 3224                      2017-11-06 23:58:31:    1
##  259    : 3130                      2017-11-07 00:15:11:    1
##  (Other):72227                      (Other)            :  222
```

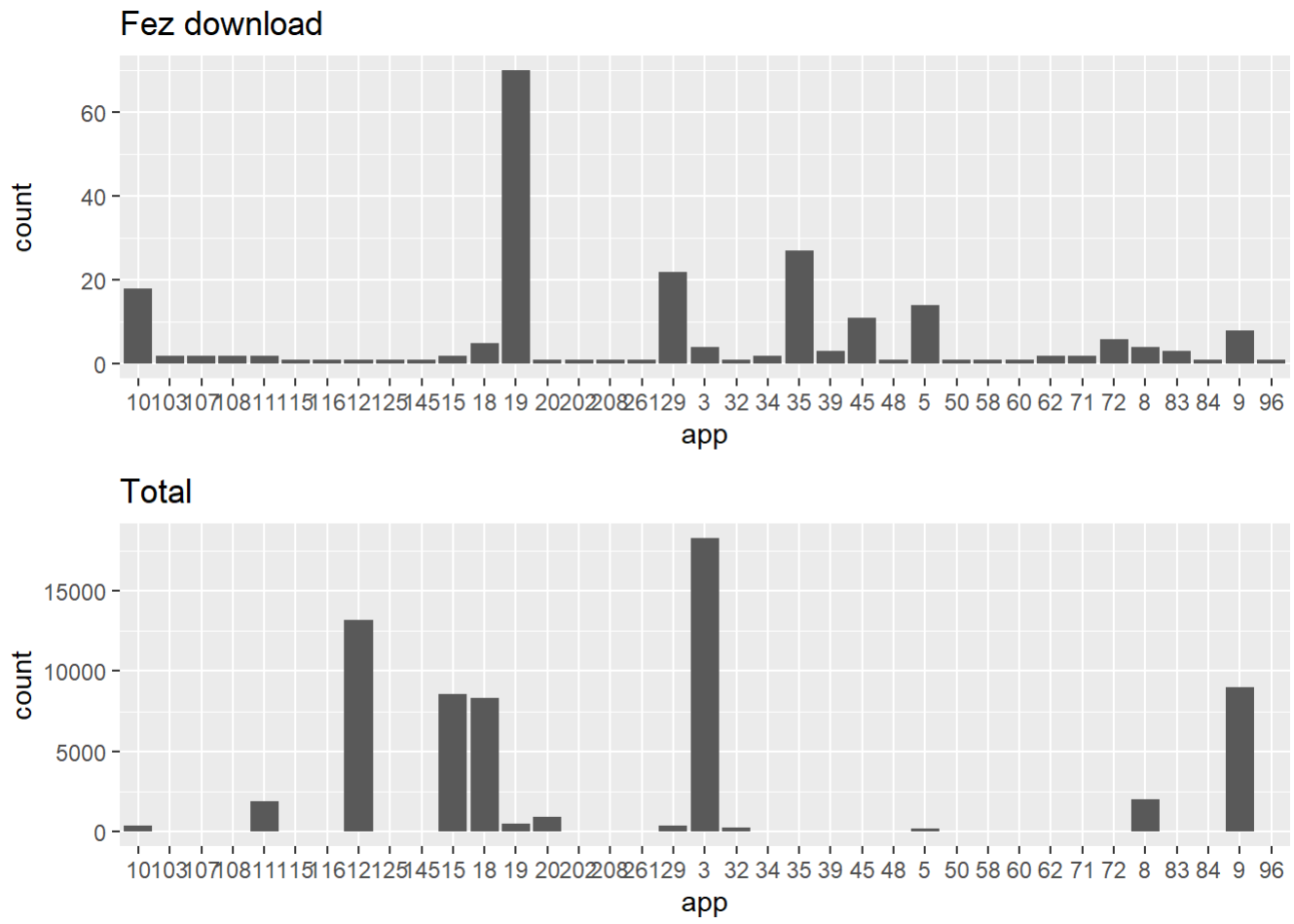
```
#Primeiramente, vou retirar do dataset a coluna ip
#que não representa uma informação útil para o modelo, sendo apenas uma
#identificação do usuário, assim como a coluna attributed_time
#que indica o horário do download, sendo uma variável pós fato
df$ip <- NULL
df$attributed_time <- NULL
View(df)

#Procurando por valores NA
qtNA_df <- df[rowSums(is.na(df)) > 0,]
qtNA_df
```

```
## Empty data.table (0 rows and 6 cols): app,device,os,channel,click_time,is_attributed
```

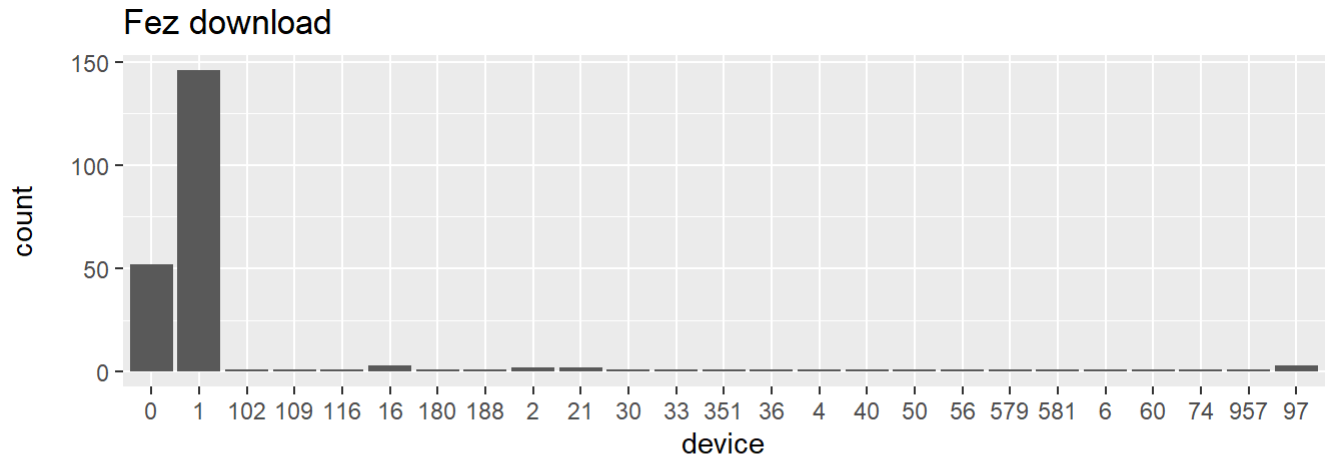
```
rm(qtNA_df)

#Como não há valores NA, quebrar a data em dia, hora e minuto
CriaHoraMinDia <- function(df){
  df <- mutate(df,hora = as.factor(hour(df$click_time)))
  df <- mutate(df,dia = as.factor(day(df$click_time)))
  df <- mutate(df,min = as.factor(minute(df$click_time)))
  return(df)
}
df <- CriaHoraMinDia(df)
str(df)
```

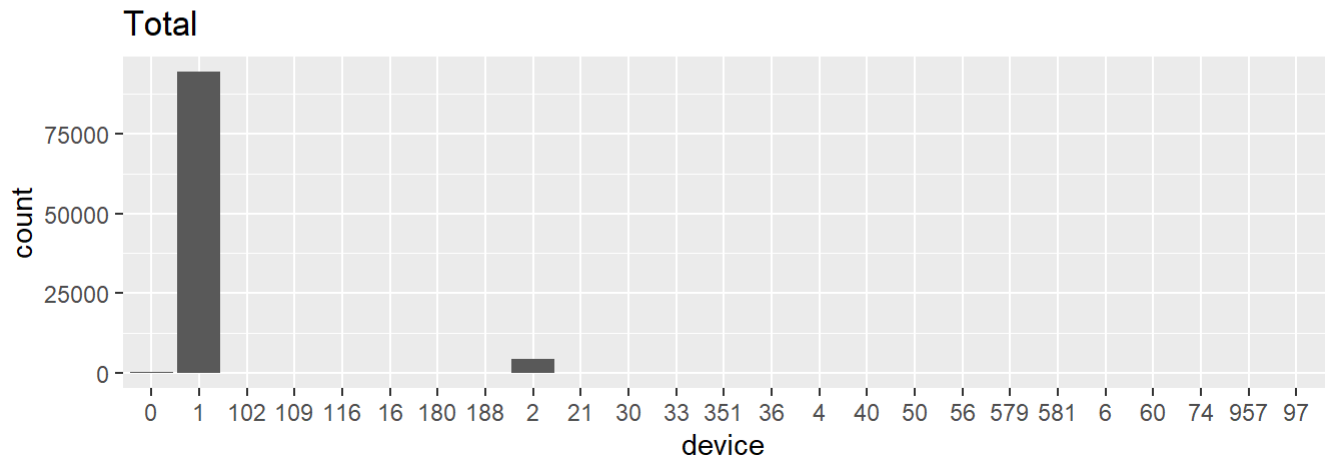
```
## 'data.frame':    100000 obs. of  9 variables:
##  $ app          : Factor w/ 161 levels "1","10","100",..: 19 68 19 25 19 80 1 153 55 80 ...
##  $ device       : Factor w/ 100 levels "0","1","100",..: 2 2 2 2 2 2 2 2 32 2 ...
##  $ os           : Factor w/ 130 levels "0","1","10","100",..: 22 37 44 22 2 37 37 57 54 44
...
##  $ channel      : Factor w/ 161 levels "101","105","107",..: 159 64 47 146 37 9 24 126 99 24
...
##  $ click_time   : chr  "2017-11-07 09:30:38" "2017-11-07 13:40:27" "2017-11-07 18:05:24" "201
7-11-07 04:58:08" ...
##  $ is_attributed: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ hora         : Factor w/ 24 levels "0","1","2","3",..: 10 14 19 5 10 2 2 11 10 13 ...
##  $ dia          : Factor w/ 4 levels "6","7","8","9": 2 2 2 2 4 4 4 2 3 3 ...
##  $ min          : Factor w/ 60 levels "0","1","2","3",..: 31 41 6 59 1 23 18 2 36 36 ...
```

```
#Vamos verificar a distribuicao de classes
df %>%
  group_by(is_attributed) %>%
  summarise(total = n()/nrow(df)*100) %>%
  View()
#Existem poucas classes 1 (apenas 0.22%). Teremos que arrumar isso no futuro para o treinamento
 do modelo

#Verificando as variáveis
#Criandos subdataset com downloads efetuados apenas
df_downloaded <- filter(df, is_attributed == 1)

#App
  PlotarAtributos("app", df_downloaded$app)
```

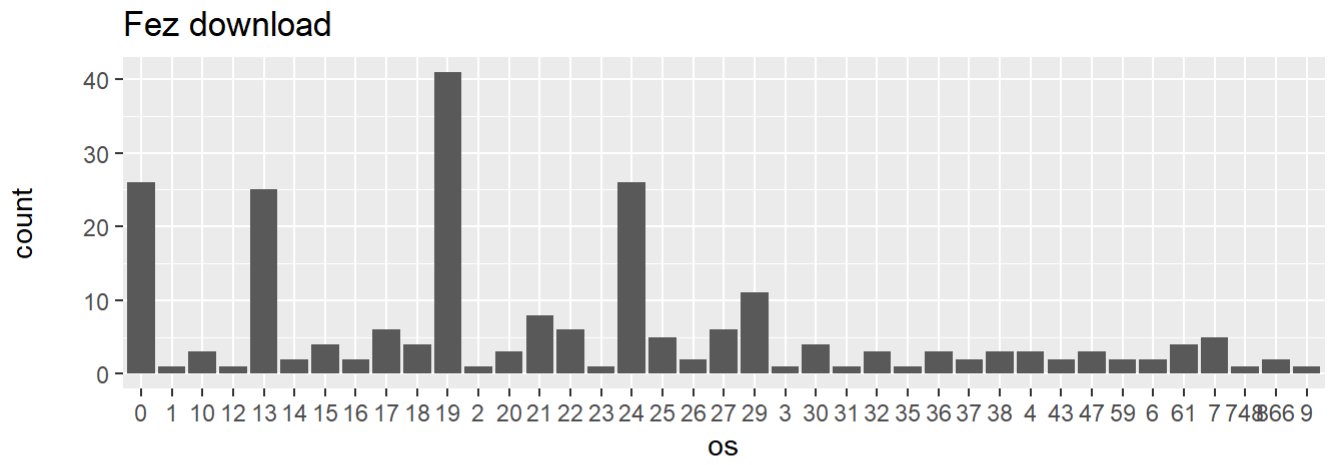


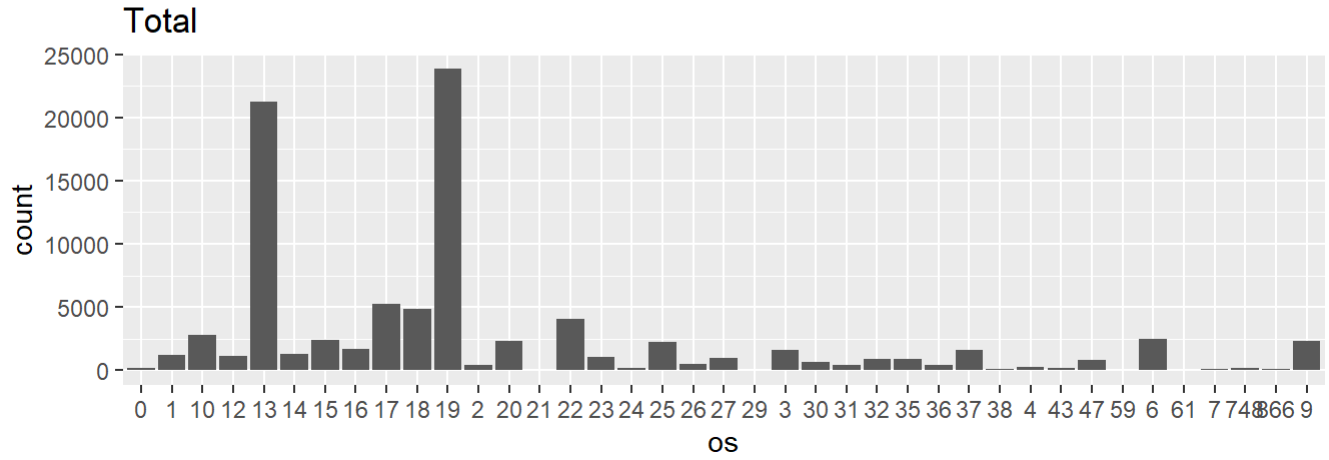

```
  #O app 19, 29 e 35 concentram grande parte dos downloads da base, mesmo sendo
  #minoria na base total. Isso indica uma forte relação entre estes app e o download

#Device
  PlotarAtributos("device", df_downloaded$device)
```



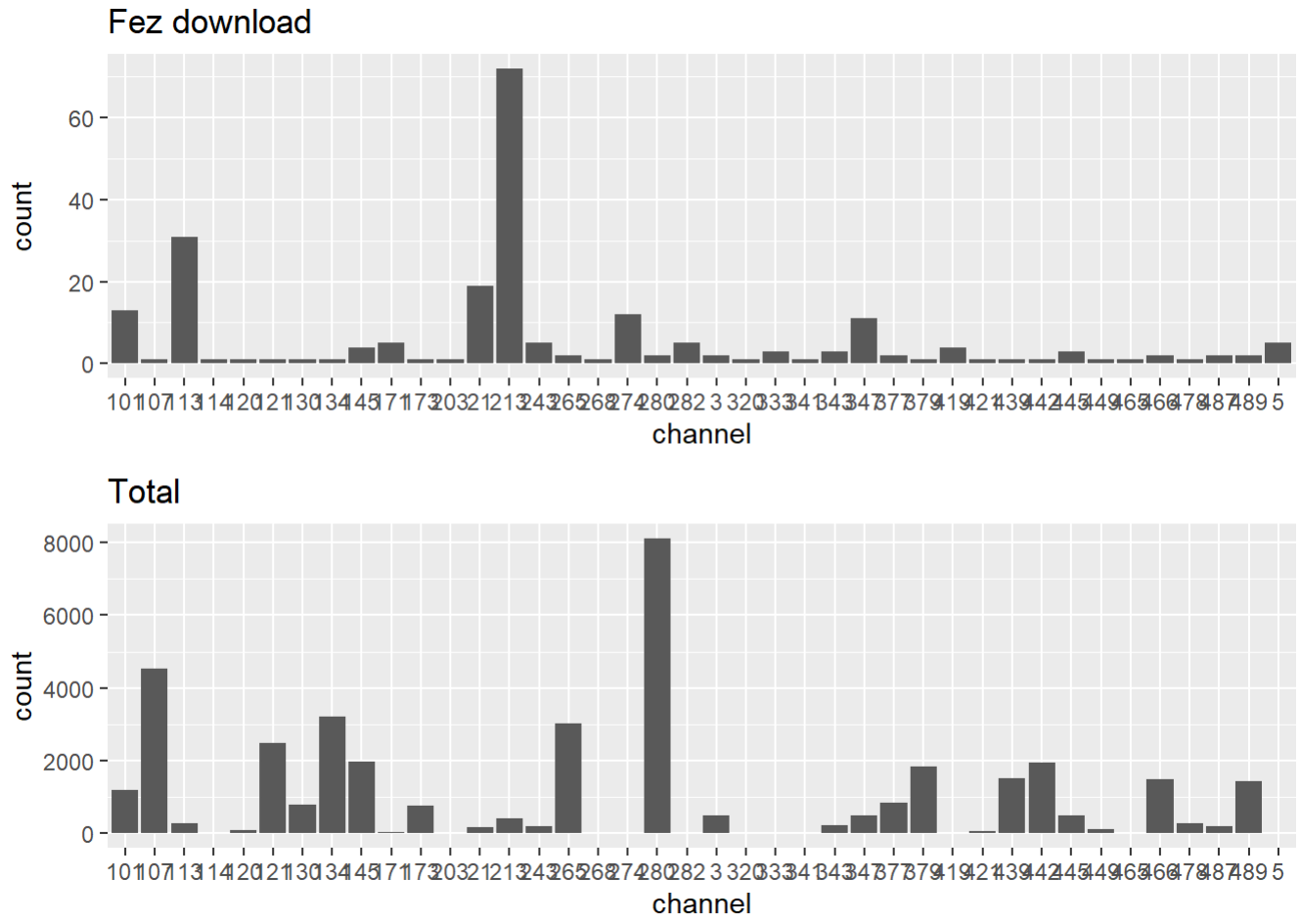



```
  #O device 0 representa grande parte dos downloads, mesmo aparecendo pouco na base

#OS
  PlotarAtributos("os", df_downloaded$os)
```

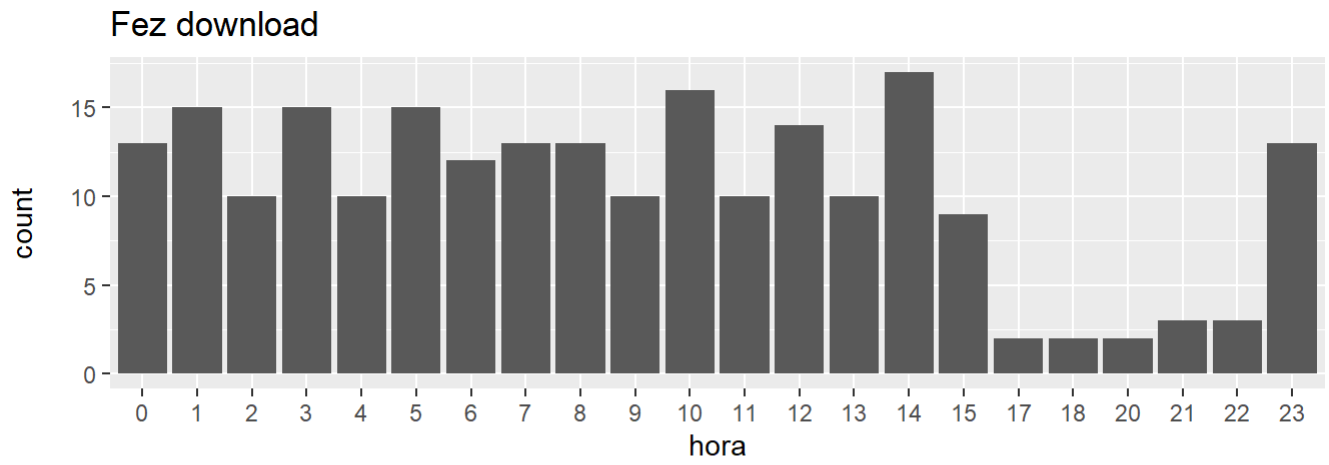


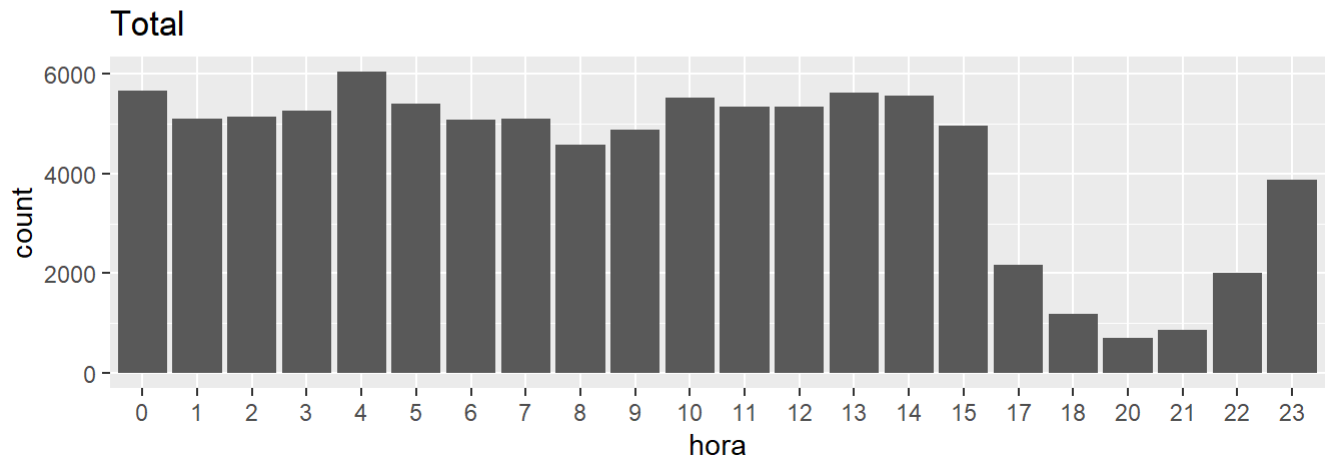



```
  #Os os 0 e 24 representam boa parte dos downloads, mesmo aparecendo pouco na base
  #O restante parece exibir mesma frequencia

#channel
  PlotarAtributos("channel", df_downloaded$channel)
```

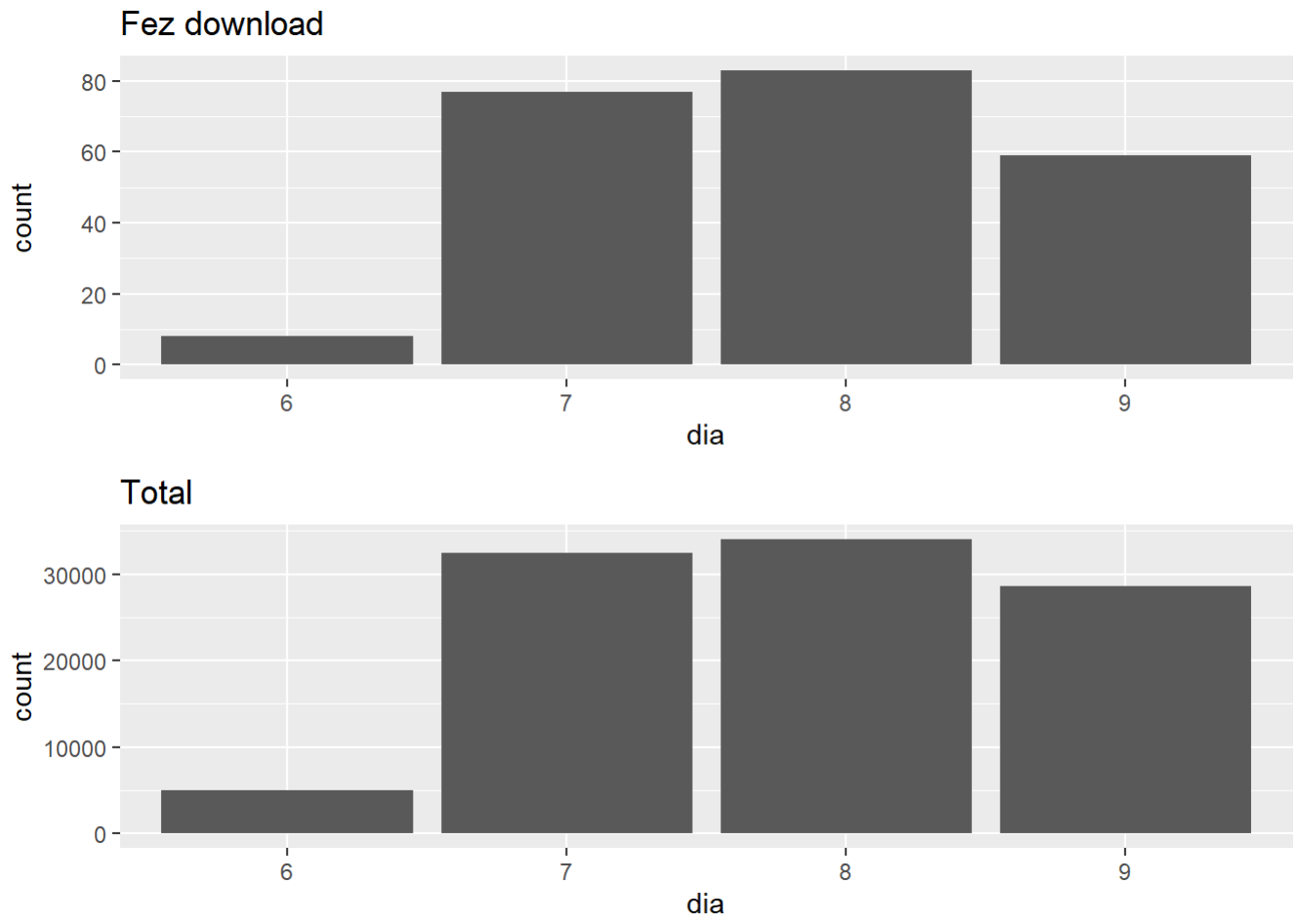



```
  #O channels 101, 113 e 213 representam boa parte dos downloads, mesmo aparecendo pouco na base

#click_time
  PlotarAtributos("hora", df_downloaded$hora)
```

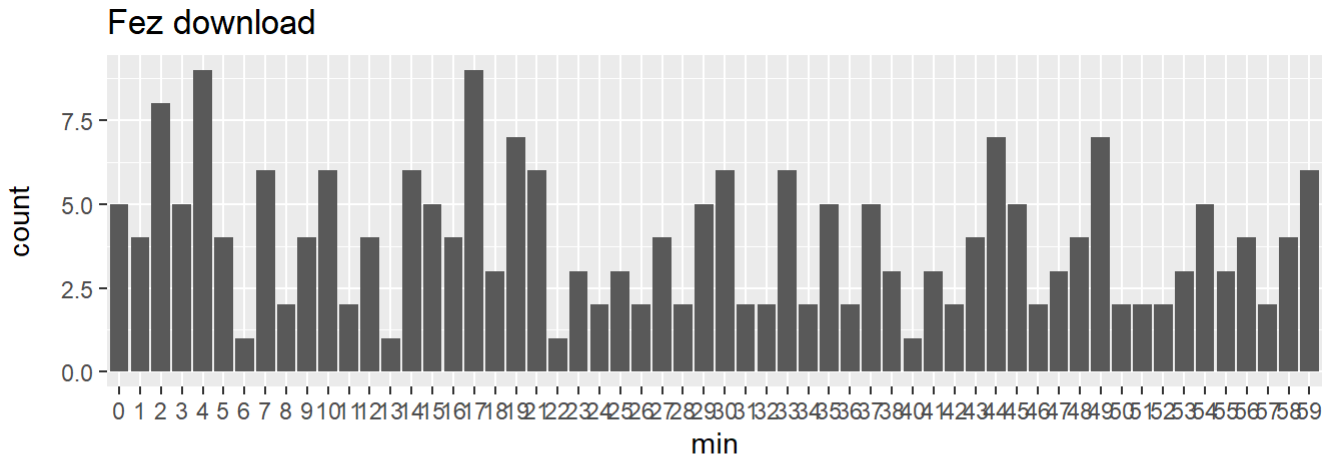


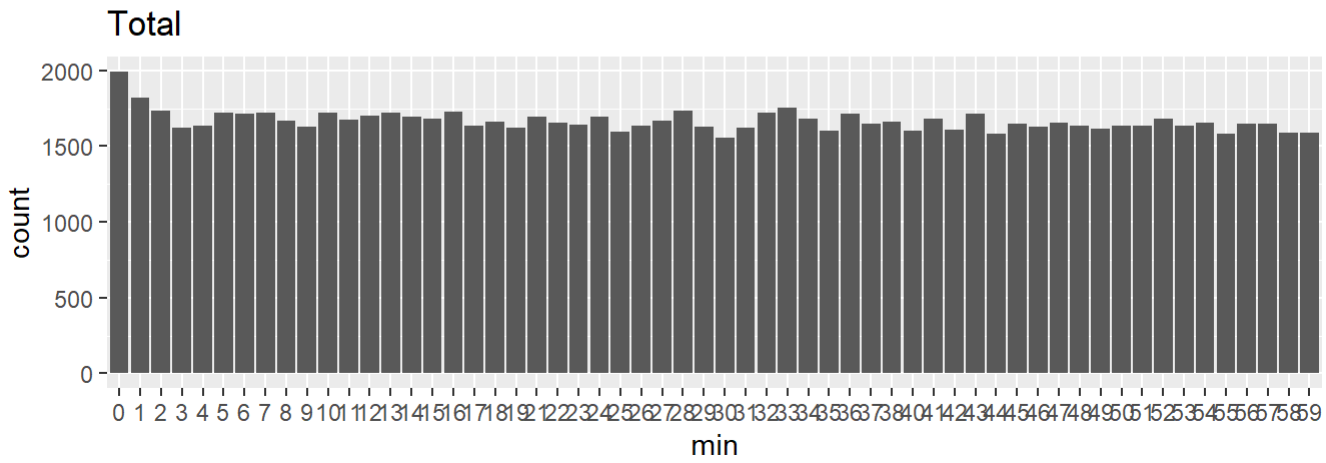



```
PlotarAtributos("dia", df_downloaded$dia)
```




```
PlotarAtributos("min", df_downloaded$min)
```






```
  #Nao parece haver grande correlacao entre o dia ou hora do download, sendo a
  #distribuicao das classes parececidas, porem os minutos parecem apresentar
  #alguma relacao

#MODELO
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

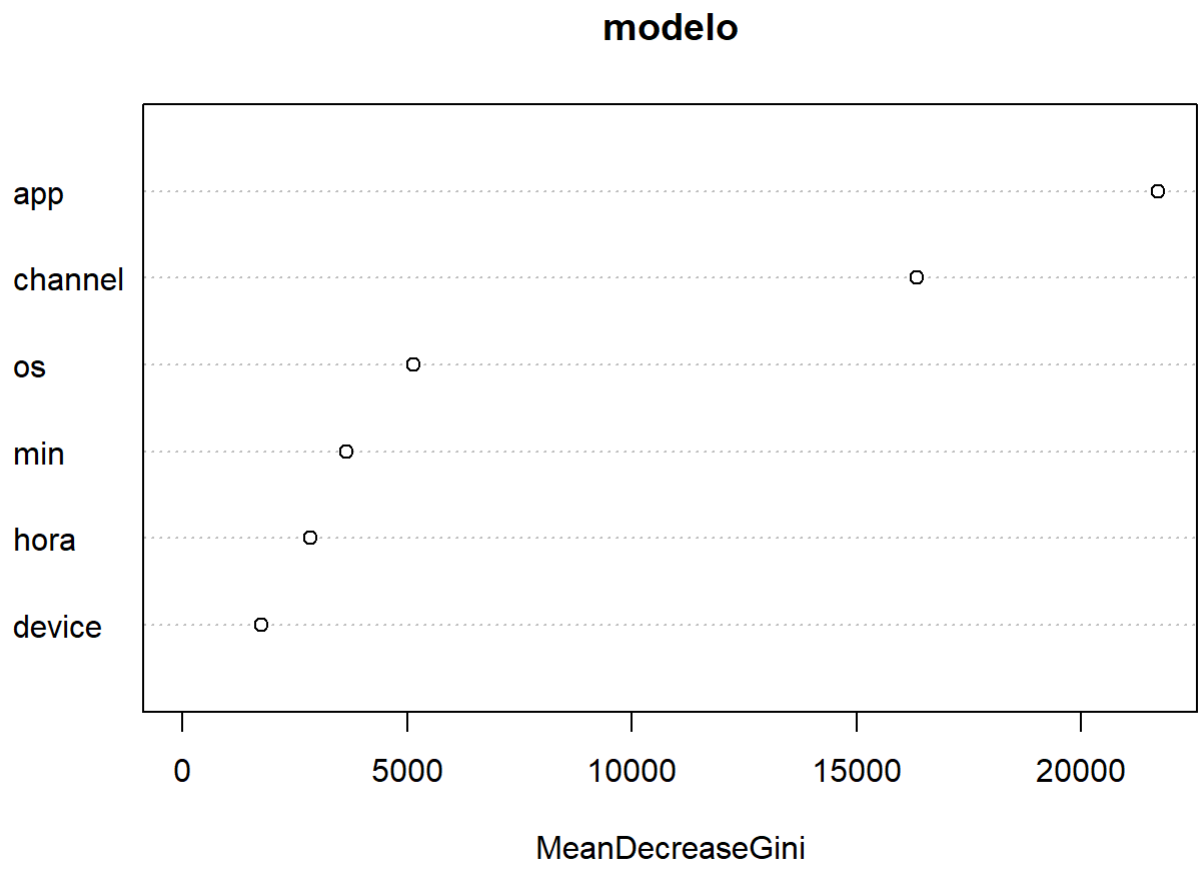
```
library(caret)
```

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

```
#Separei o dataset em treino, validacao e teste. Como o dataset esta desbalanceado (predominanci
a da classe 0)
#e temos uma base muito grande, vou gerar o dataset de treino e validacao balanceados baseados n
a quantidade de
#classes 1, gerando as classes 0 randomicamente. Isso reduzira o numero de linhas totais para tr
eino, porem ainda
#mantendo uma quantidade suficiente de exemplos para treinamento. Ja o dataset de test eu deixar
ei com a mesma
#distribuicao original, dado que ele é a representação original do dataset, onde o modelo seria
 aplicado em produção

#A separacao do dataset foi feita no script GenerateDataSet.R
#Apos separacao dos datasets, foi necessario agrupar os elementos
#dada a grande quantidade de variaveis diferentes do tipo fator,
#o que prejudica a criacao do modelo. O agrupamento foi feito
#no script GenerateDataSetsGroup.R. Estes datasets agrupados
#sao as bases finais que utilizaremos para criacao do modelo

trainSet <- fread("trainSetGroup.csv", colClasses=c(rep('factor', 5), rep('integer', 3)))
validationSet <- fread("validationSetGroup.csv", colClasses=c(rep('factor', 5), rep('integer', 3
)))
testSet <- fread("testSetGroup.csv", colClasses=c(rep('factor', 5), rep('integer', 3)))

#Nas fases de teste, verifiquei durante o Feature Selection com Random Forest que
#a variavel dia representava pouca importancia para o modelo. Com isso, vou apaga-la da base

trainSet$dia <- NULL
validationSet$dia <- NULL
testSet$dia <- NULL
trainSet <- distinct(trainSet)
validationSet <- distinct(validationSet)
testSet <- distinct(testSet)

levels(validationSet$device) <- levels(trainSet$device)

#Treinando o modelo
modelo <- randomForest( is_attributed ~ .,
                        data = trainSet)
previsoes <- data.frame(observado = validationSet$is_attributed,
                        previsto = predict(modelo, newdata = validationSet))
confusionMatrix(previsoes$observado, previsoes$previsto)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 68027  3466
##          1 13091 28908
##
##                Accuracy : 0.8541
##                  95% CI : (0.852, 0.8562)
##     No Information Rate : 0.7147
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6716
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8386
##             Specificity : 0.8929
##          Pos Pred Value : 0.9515
##          Neg Pred Value : 0.6883
##              Prevalence : 0.7147
##          Detection Rate : 0.5994
##    Detection Prevalence : 0.6299
##       Balanced Accuracy : 0.8658
##
##        'Positive' Class : 0
##
```

```
varImpPlot(modelo)
```




```
#Apesar da boa acuracia, a taxa de falsos positivos esta muito alta 63%
#Tentarei agora o modelo de regressao logistica
modeloRL <- glm(formula = is_attributed ~ ., data = trainSet, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
previsoesRL <- data.frame(observado = validationSet$is_attributed,
                          previsto = as.factor(round(predict(modeloRL, newdata = validationSet, ty
pe="response"))))
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
confusionMatrix(previsoesRL$observado, previsoesRL$previsto)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 68567  2926
##          1 14807 27192
##
##                Accuracy : 0.8438
##                  95% CI : (0.8416, 0.8459)
##     No Information Rate : 0.7346
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6441
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8224
##             Specificity : 0.9028
##          Pos Pred Value : 0.9591
##          Neg Pred Value : 0.6474
##              Prevalence : 0.7346
##          Detection Rate : 0.6042
##    Detection Prevalence : 0.6299
##       Balanced Accuracy : 0.8626
##
##        'Positive' Class : 0
##
```

```
#A regressao logistica teve o mesmo problema da random florest, muitos falsos positivos
#Vou testar agora o modelo de Random Florest com o dataset de teste
levels(testSet$device) <- levels(trainSet$device)
previsoes <- data.frame(observado = testSet$is_attributed,
                        previsto = predict(modelo, newdata = testSet))
confusionMatrix(previsoes$observado, previsoes$previsto)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction       0       1
##          0 1230563  147296
##          1   11252   26189
##
##                Accuracy : 0.888
##                  95% CI : (0.8875, 0.8885)
##     No Information Rate : 0.8774
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2141
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9909
##             Specificity : 0.1510
##          Pos Pred Value : 0.8931
##          Neg Pred Value : 0.6995
##              Prevalence : 0.8774
##          Detection Rate : 0.8695
##    Detection Prevalence : 0.9735
##       Balanced Accuracy : 0.5709
##
##        'Positive' Class : 0
##
```

```
#A acuracia aumentou um pouco, porem fruto do desbalanceamento dos dados (98% das classes 0).
```