kaggle Search kaggle

Competitions Datasets Kernels Discussion Learn

Yash Mahendra

Exploratory Data Analysis

0
voters

last run 36 minutes ago · R notebook
using data from TalkingData AdTracking Fraud Detection Challenge · Private Make Public

Notebook Code Data (1) Output (1) Comments (0) Log Versions (2) Options Fork Notebook Edit Notebook

Tags Add Tag

Notebook

EDA in Kaggle Kernel

In [1]:

```
# This R environment comes with all of CRAN preinstalled, as well as many other helpful packages
# The environment is defined by the kaggle/rstats docker image: https://github.com/kaggle/docker-rstats
# For example, here's several helpful packages to load in

library(ggplot2) # Data visualization
library(readr) # CSV file I/O, e.g. the read_csv function

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

system("ls ../input")

# Any results you write to the current directory are saved as output.
```

In [2]:

```
library(data.table)
library(ggplot2)
library(DT)
library(magrittr)
library(corrplot)
library(Rmisc)
library(ggalluvial)
library(caret)
library(ModelMetrics)
require(scales)
library(irlba)
library(forcats)
library(forecast)
library(TSA)
library(zoo)
library(skimr)
library(fasttime)
library(gridExtra)
library(Amelia)
```

```
corrplot 0.84 loaded
Loading required package: lattice
Loading required package: plyr

Attaching package: ‘ModelMetrics’

The following objects are masked from ‘package:caret’:

    confusionMatrix, precision, recall, sensitivity, specificity

Loading required package: scales

Attaching package: ‘scales’

The following object is masked from ‘package:readr’:
    col_factor

Loading required package: Matrix

Attaching package: ‘forecast’

The following object is masked from ‘package:ggplot2’:

    autolayer

Loading required package: leaps
Loading required package: locfit
locfit 1.5-9.1   2013-03-22
Loading required package: mgcv
Loading required package: nlme

Attaching package: ‘nlme’

The following object is masked from ‘package:forecast’:

    getResponse

This is mgcv 1.8-23. For overview type 'help("mgcv-package")'.
Loading required package: tseries

Attaching package: ‘TSA’

The following object is masked from ‘package:readr’:

    spec

The following objects are masked from ‘package:stats’:

    acf, arima

The following object is masked from ‘package:utils’:

    tar


Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: Rcpp
##
## Amelia II: Multiple Imputation
## (Version 1.7.4, built: 2015-12-05)
## Copyright (C) 2005-2018 James Honaker, Gary King and Matthew Blackwell
## Refer to http://gking.harvard.edu/amelia/ for more information
##
```

In [3]:

```
# Lets use train data and we will later split it into training and testing
# Since the data is quite large, this approach can be implemented on larger data with server and cloud
train <- fread("../input/train_sample.csv", showProgress=F)
```

In [4]:

```
#Check the head of Train
head(train,5)
```

ip	app	device	os	channel	click_time	attributed_time	is_attributed
29540	3	1	42	489	2017-11-08 03:57:46		0
26777	11	1	25	319	2017-11-09 11:02:14		0
140926	12	1	13	140	2017-11-07 04:36:14		0
69375	2	1	19	377	2017-11-09 13:17:20		0
119166	9	2	15	445	2017-11-07 12:11:37		0

In [5]:

```
#Check data for Null Values
sapply(train, function(y) sum(is.na(y)))
```

```
             ip  0
            app  0
         device  0
             os  0
        channel  0
     click_time  0
attributed_time  0
  is_attributed  0
```

Check Factor Variable

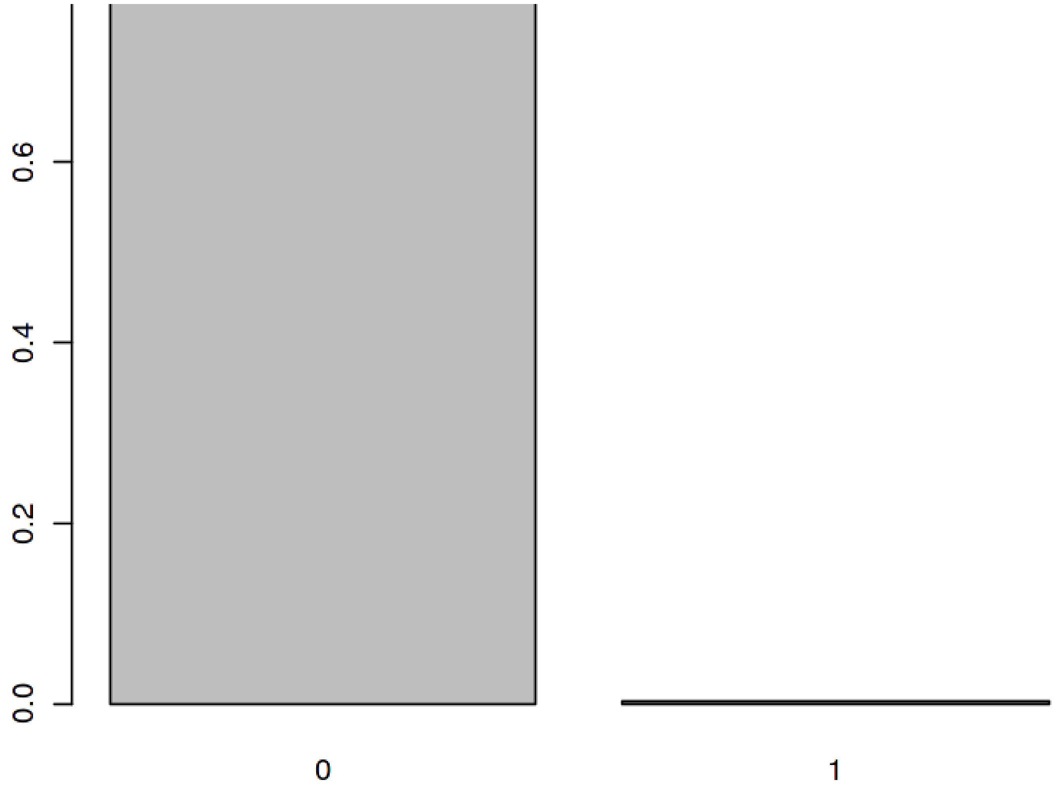
In [6]:

```
table(train$is_attributed)
```

```
    0     1 
99749   251 
```

In [7]:

```
barplot(prop.table(table(train$is_attributed)))
```



In [8]:

```
str(train)
```

```
Classes ‘data.table’ and 'data.frame':  100000 obs. of  8 variables:
 $ ip             : int  29540 26777 140926 69375 119166 126411 118315 34631 108040 14230 ...
 $ app            : int  3 11 12 2 9 13 1 12 12 3 ...
 $ device         : int  1 1 1 1 2 1 1 1 1 1 ...
 $ os             : int  42 25 13 19 15 17 13 6 19 13 ...
 $ channel        : int  489 319 140 377 445 477 153 140 265 19 ...
 $ click_time     : chr  "2017-11-08 03:57:46" "2017-11-09 11:02:14" "2017-11-07 04:36:14" "2017-11-09 1
3:17:20" ...
 $ attributed_time: chr  "" "" "" "" ...
 $ is_attributed  : int  0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

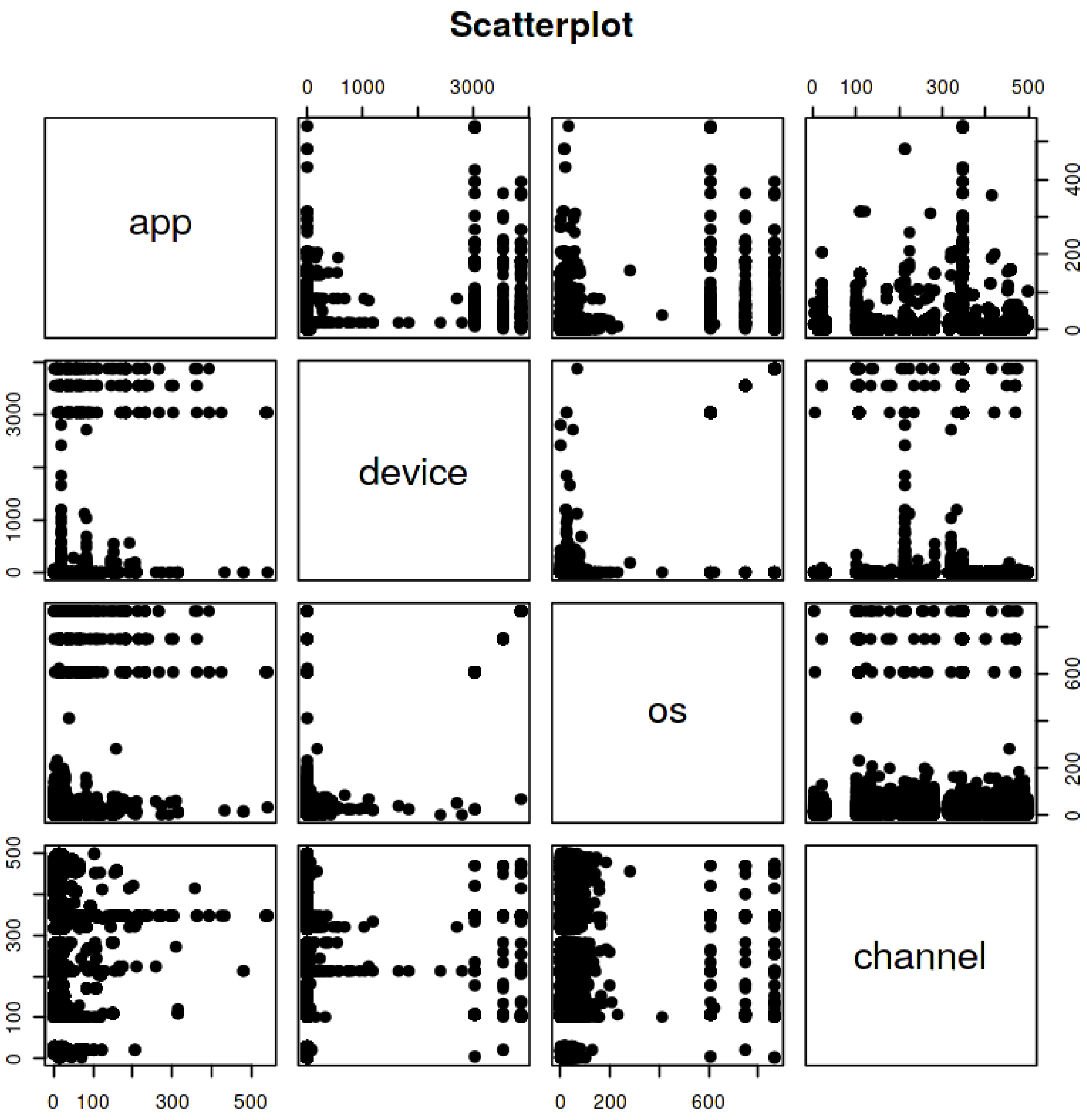
In [9]:

```
summary(train)
```

```
       ip              app             device              os        
 Min.   :     9   Min.   :  0.00   Min.   :   0.00   Min.   :  0.00  
 1st Qu.: 40316   1st Qu.:  3.00   1st Qu.:   1.00   1st Qu.: 13.00  
 Median : 79666   Median : 12.00   Median :   1.00   Median : 18.00  
 Mean   : 91092   Mean   : 12.03   Mean   :  22.39   Mean   : 22.84  
 3rd Qu.:118284   3rd Qu.: 15.00   3rd Qu.:   1.00   3rd Qu.: 19.00  
 Max.   :364759   Max.   :542.00   Max.   :3866.00   Max.   :866.00  
    channel       click_time        attributed_time    is_attributed    
 Min.   :  3.0   Length:100000      Length:100000      Min.   :0.00000  
 1st Qu.:140.0   Class :character   Class :character   1st Qu.:0.00000  
 Median :258.0   Mode  :character   Mode  :character   Median :0.00000  
 Mean   :268.7                                         Mean   :0.00251  
 3rd Qu.:379.0                                         3rd Qu.:0.00000  
 Max.   :498.0                                         Max.   :1.00000  
```

In [10]:

```
plot(train[,2:5], main="Scatterplot", pch=19)
```

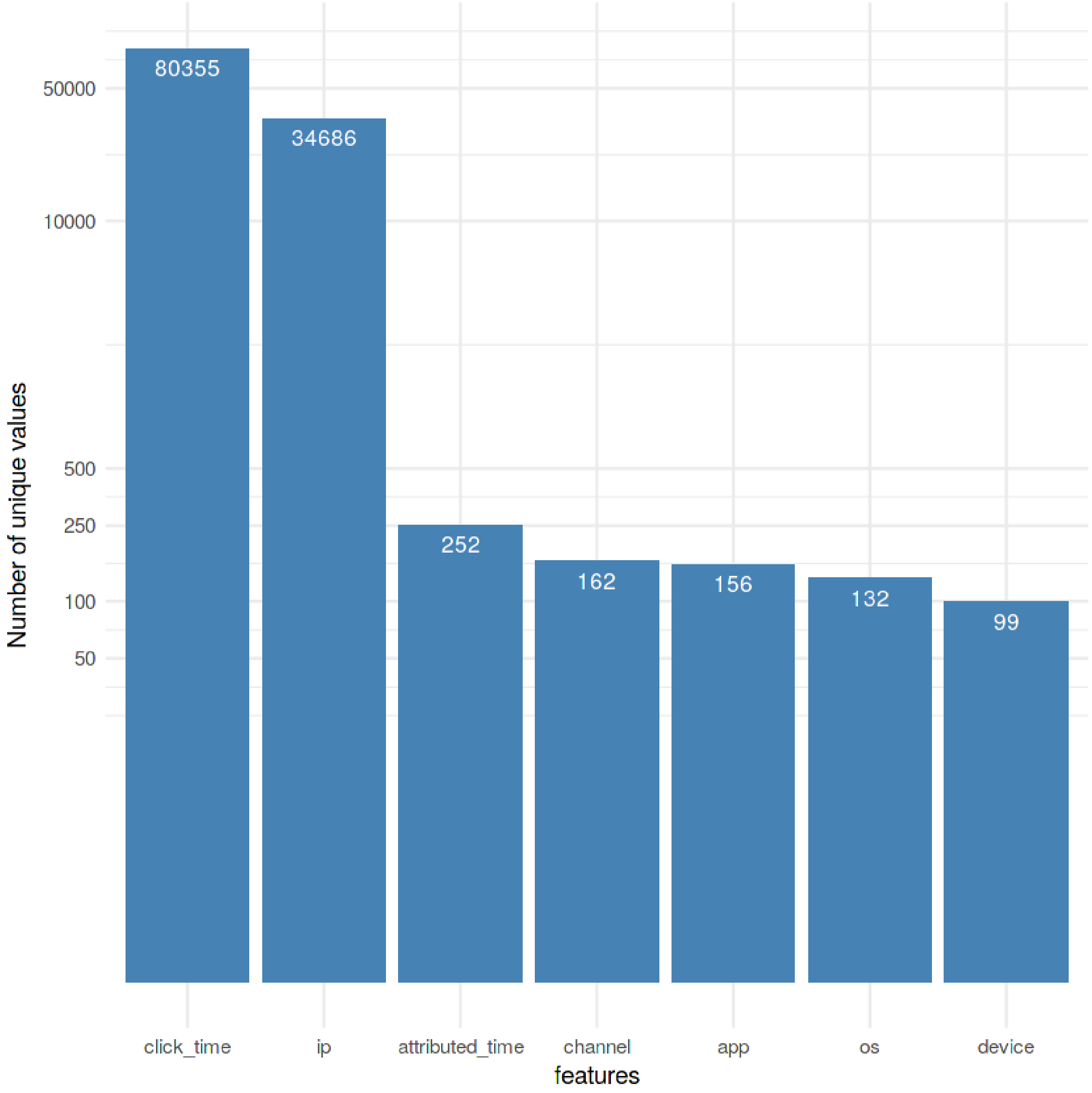


Let’s have a look at features counts:

In [11]:

```
fea <- c("os", "channel", "device", "app", "attributed_time", "click_time", "ip")
train[, lapply(.SD, uniqueN), .SDcols = fea] %>%
  melt(variable.name = "features", value.name = "unique_values") %>%
  ggplot(aes(reorder(features, -unique_values), unique_values)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  scale_y_log10(breaks = c(50,100,250, 500, 10000, 50000)) +
  geom_text(aes(label = unique_values), vjust = 1.6, color = "white", size=3.5) +
  theme_minimal() +
  labs(x = "features", y = "Number of unique values")
```

```
Warning message in melt.data.table(., variable.name = "features", value.name = "unique_values"):
“To be consistent with reshape2's melt, id.vars and measure.vars are internally guessed when both are 'N
ULL'. All non-numeric/integer/logical type columns are conisdered id.vars, which in this case are column
s []. Consider providing at least one of 'id' or 'measure' vars in future.”
```

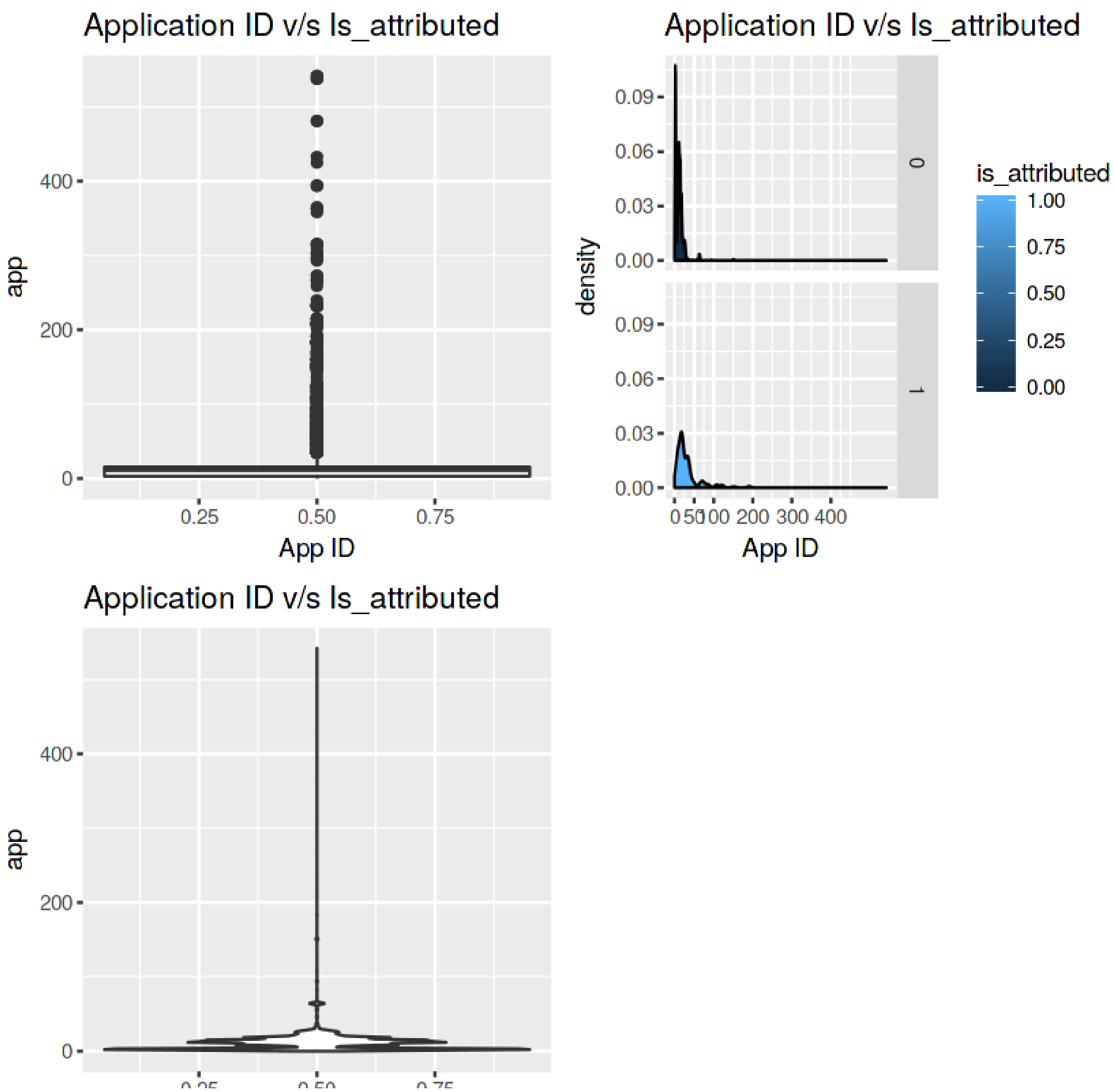


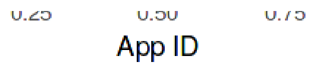
Checking Important Features

In [12]:

```
#Application ID vs is_attributed
p1=ggplot(train,aes(x=is_attributed,y=app,fill=is_attributed))+
  geom_boxplot()+
  ggtitle("Application ID v/s Is_attributed")+
  xlab("App ID") +
  labs(fill = "is_attributed")

p2=ggplot(train,aes(x=app,fill=is_attributed))+
  geom_density()+facet_grid(is_attributed~.)+
  scale_x_continuous(breaks = c(0,50,100,200,300,400))+
  ggtitle("Application ID v/s Is_attributed")+
  xlab("App ID") +
  labs(fill = "is_attributed")


p3=ggplot(train,aes(x=is_attributed,y=app,fill=is_attributed))+
  geom_violin()+
  ggtitle("Application ID v/s Is_attributed")+
  xlab("App ID") +
  labs(fill = "is_attributed")


grid.arrange(p1,p2,p3, nrow=2,ncol=2)
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```

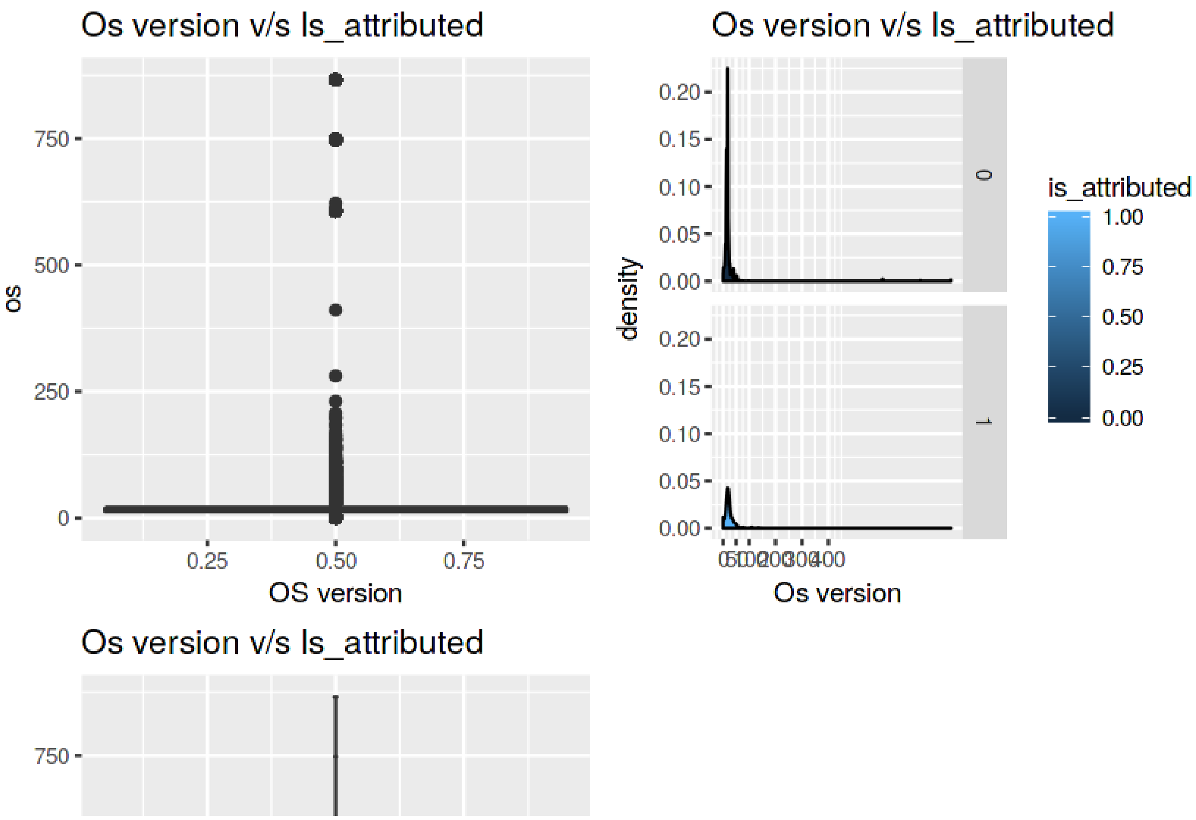


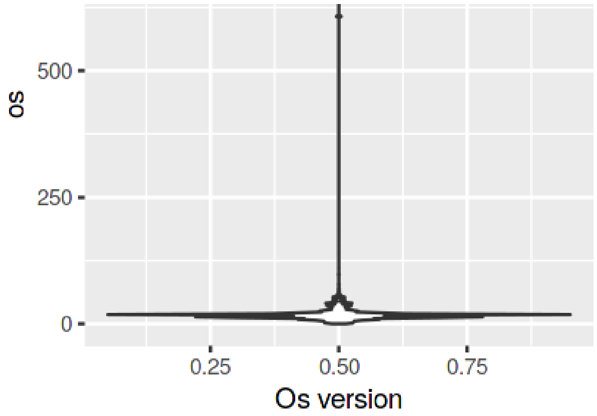


In [13]:

```
#App downloaded vs OS version id of user mobile phone
p4=ggplot(train,aes(x=is_attributed,y=os,fill=is_attributed))+
  geom_boxplot()+
  ggtitle("Os version v/s Is_attributed")+
  xlab("OS version") +
  labs(fill = "is_attributed")


p5=ggplot(train,aes(x=os,fill=is_attributed))+
  geom_density()+facet_grid(is_attributed~.)+
  scale_x_continuous(breaks = c(0,50,100,200,300,400))+
  ggtitle("Os version v/s Is_attributed ")+
  xlab("Os version") +
  labs(fill = "is_attributed")


p6=ggplot(train,aes(x=is_attributed,y=os,fill=is_attributed))+
  geom_violin()+
  ggtitle("Os version v/s Is_attributed")+
  xlab("Os version") +
  labs(fill = "is_attributed")


grid.arrange(p4,p5, p6, nrow=2,ncol=2)
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```

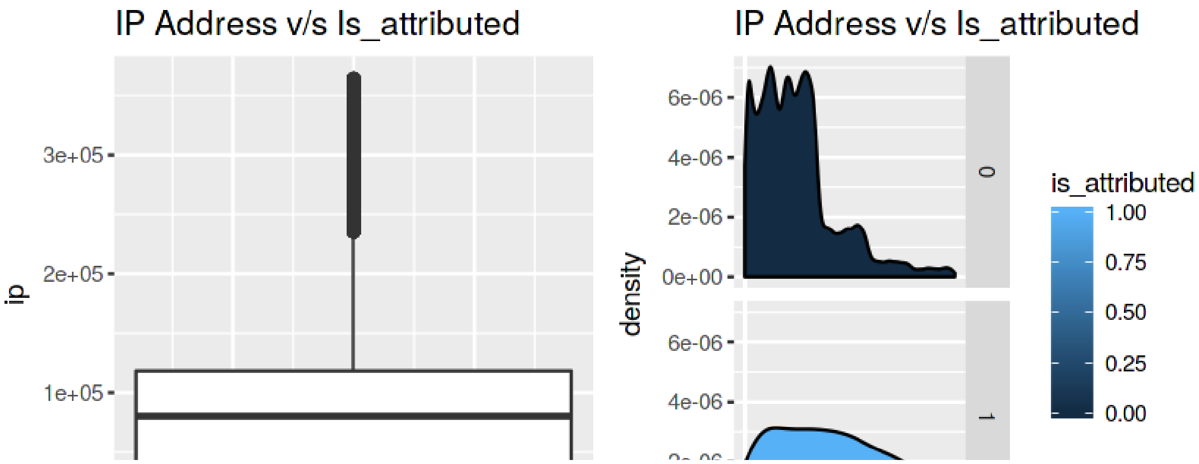


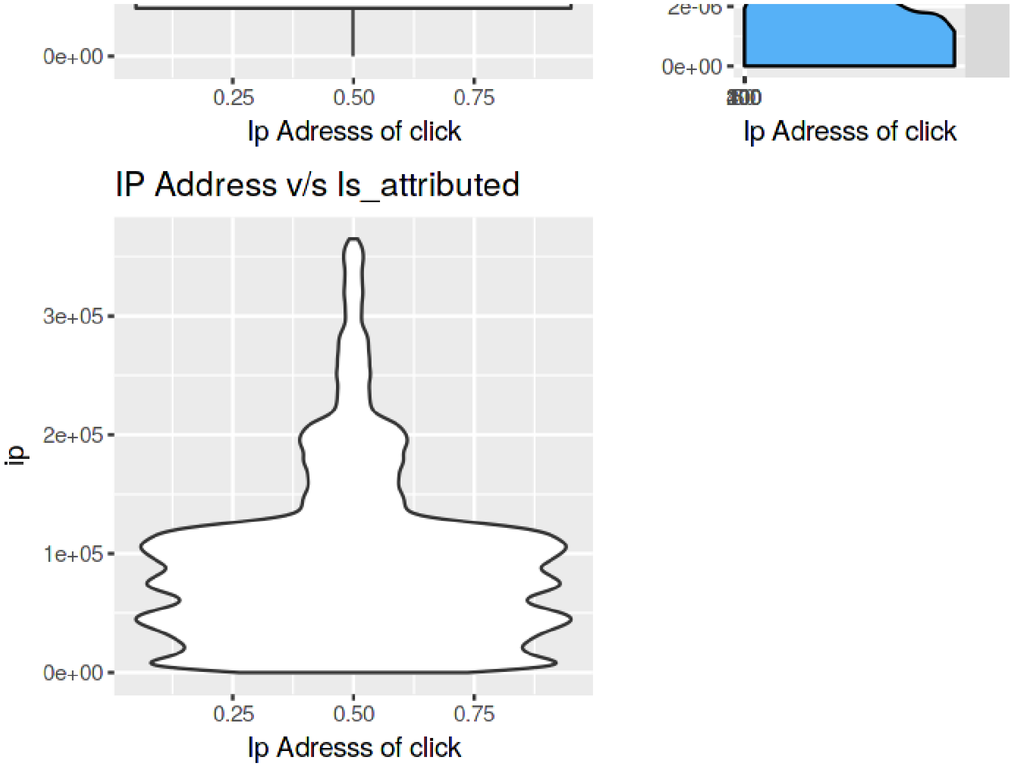


In [14]:

```
###App was downloaded v/s ip address of click.
p7=ggplot(train,aes(x=is_attributed,y=ip,fill=is_attributed))+
  geom_boxplot()+
  ggtitle("IP Address v/s Is_attributed")+
  xlab("Ip Adresss of click") +
  labs(fill = "is_attributed")


p8=ggplot(train,aes(x=ip,fill=is_attributed))+
  geom_density()+facet_grid(is_attributed~.)+
  scale_x_continuous(breaks = c(0,50,100,200,300,400))+
  ggtitle("IP Address v/s Is_attributed")+
  xlab("Ip Adresss of click") +
  labs(fill = "is_attributed")



p9=ggplot(train,aes(x=is_attributed,y=ip,fill=is_attributed))+
  geom_violin()+
  ggtitle("IP Address v/s Is_attributed")+
  xlab("Ip Adresss of click") +
  labs(fill = "is_attributed")

grid.arrange(p7,p8, p9, nrow=2,ncol=2)
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```

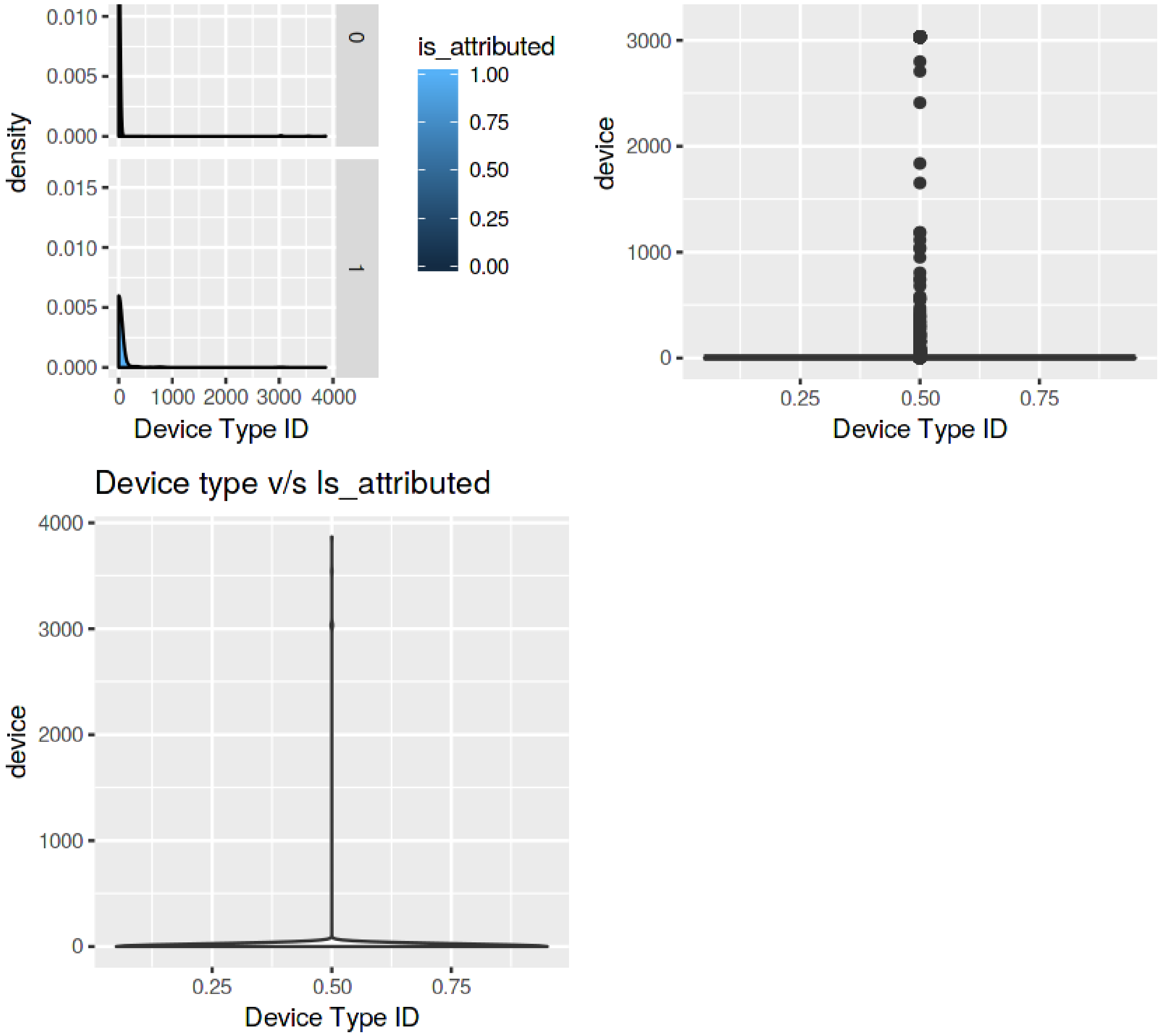




In [15]:

```
###App was downloaded v/s device type id of user mobile phone

p10=ggplot(train,aes(x=device,fill=is_attributed))+
  geom_density()+facet_grid(is_attributed~.)+
  ggtitle("Device type v/s Is_attributed")+
  xlab("Device Type ID") +
  labs(fill = "is_attributed")



p11=ggplot(train,aes(x=is_attributed,y=device,fill=is_attributed))+
  geom_boxplot()+
  ggtitle("Device type v/s Is_attributed")+
  xlab("Device Type ID") +
  labs(fill = "is_attributed")



p12=ggplot(train,aes(x=is_attributed,y=device,fill=is_attributed))+
  geom_violin()+
  ggtitle("Device type v/s Is_attributed")+
  xlab("Device Type ID") +
  labs(fill = "is_attributed")

grid.arrange(p10,p11, p12, nrow=2,ncol=2)
```

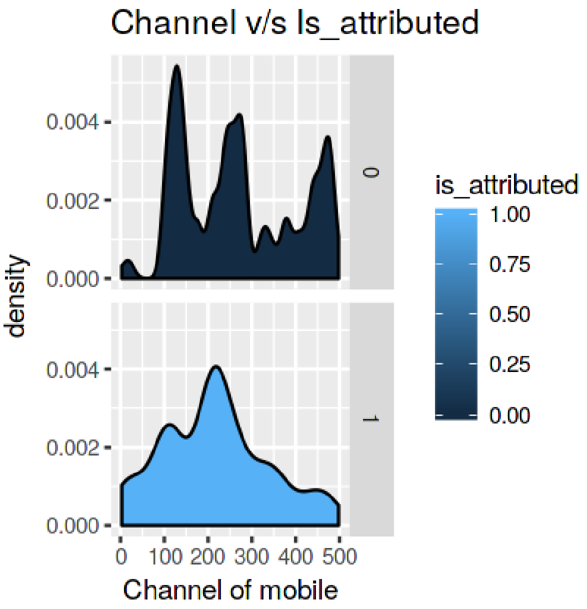
```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```

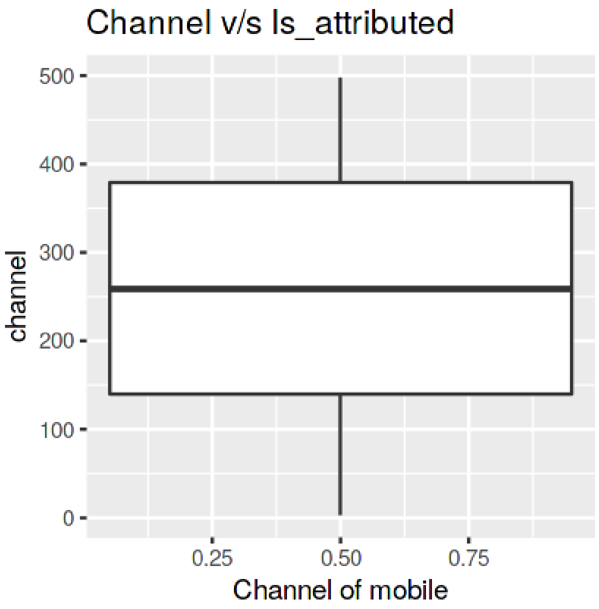


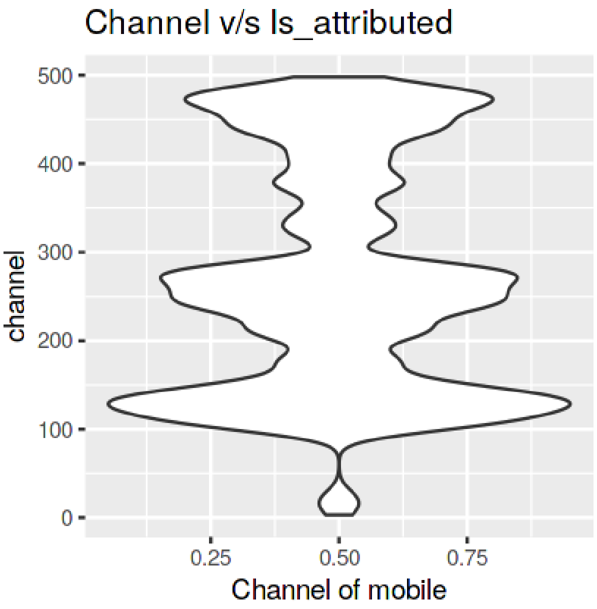
In [16]:

```
###App was downloaded v/s channel id of mobile ad publisher

p13=ggplot(train,aes(x=channel,fill=is_attributed))+
  geom_density()+facet_grid(is_attributed~.)+
  ggtitle("Channel v/s Is_attributed")+
  xlab("Channel of mobile") +
  labs(fill = "is_attributed")


p14=ggplot(train,aes(x=is_attributed,y=channel,fill=is_attributed))+
  geom_boxplot()+
  ggtitle("Channel v/s Is_attributed")+
  xlab("Channel of mobile") +
  labs(fill = "is_attributed")

p15=ggplot(train,aes(x=is_attributed,y=channel,fill=is_attributed))+
  geom_violin()+
  ggtitle("Channel v/s Is_attributed")+
  xlab("Channel of mobile") +
  labs(fill = "is_attributed")

grid.arrange(p13,p14, p15, nrow=2,ncol=2)
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```



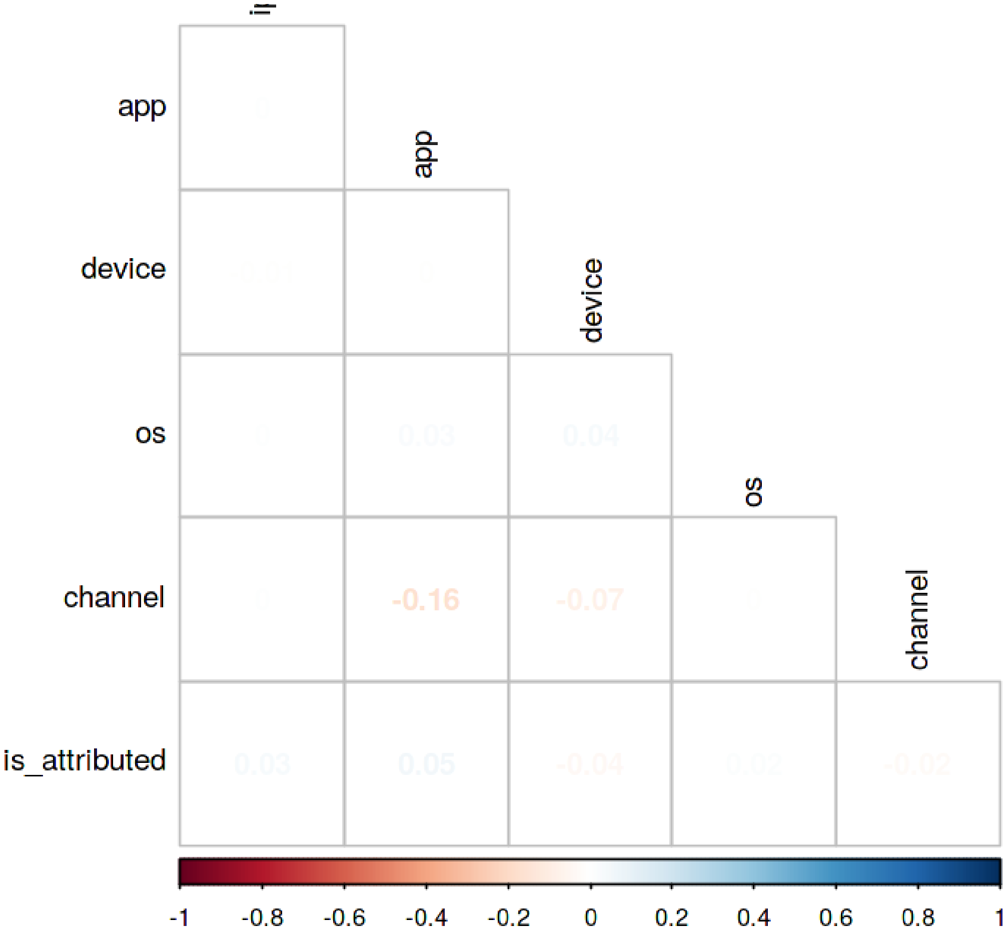




Correlation

In [17]:

```
train[, -c("click_time", "attributed_time"), with=F] %>%
  cor(method = "spearman") %>%
  corrplot(type="lower", method = "number", tl.col = "black", diag=FALSE)
```



End

Comments (0)

Sort by Select...

Click here to enter a comment...