**1.The UA insight trend we will provide here**

You are tasked with looking into the performance of a fictional freemium mobile game over a period of time. Please provide three (3) exploratory plots on the data that you find relevant along with brief summaries of your observations for each plot (free format, can be bullet points). You can modify and transform the attached dataset as you see fit; please detail any changes or calculations and provide brief reasoning. It is not necessary to use all columns provided in the dataset, aim rather to provide relevant plots that give an overview of game performance over time. Use a software of your choice for this task. If you use a scripting language, please attach the script to your answer.

# Jazz report on task analysis

In the overview of a game performance, we will take a look on daily install,D1_dau,Retention rate,ARPU and LTV

Read CSV into R

```
df <- read.csv(file="ua_analyst_task3.csv", header=TRUE, sep=",")
```

# 1.The trend of new installs

We want to know the trend of new installs.

Overview of Installs with line chart and linear fit line.

Load the `ggplot2` package to plot the line chart, and `scales` package is to Formate dates on X axis in `ggplot2`

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
require(scales)
```

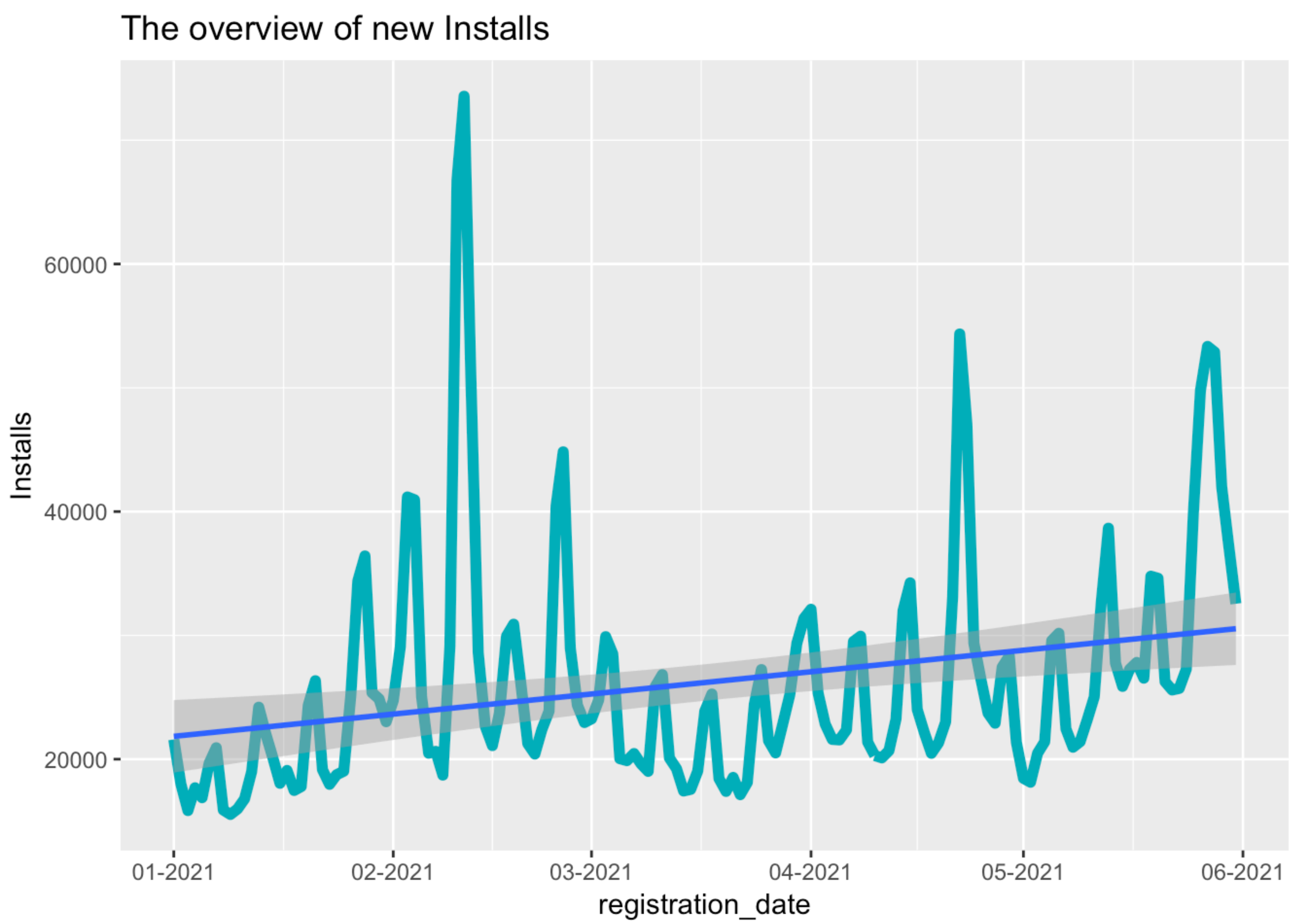
```
## Loading required package: scales
```

Define columns as the dataframe we want to plot, and using `as.Date` to transfer `df$registration_date` as a date column

```
df1 <- data.frame(df$registration_date,df$installs)
date <- as.Date(df$registration_date)
```

Using `ggplot()` to generate line chart. Add linear fit line with `method = "lm"`. `scale_x_dat` is the function we use to format date

```
ggplot(data = df1, aes(x = date, y = df$installs,group = 1))+
  geom_line(color = "#00AFBB", size = 2)+
  geom_smooth(method = "lm")+
  labs(x = "registration_date", y = "Installs",
       title = "The overview of new Installs")+
  scale_x_date(labels = date_format("%m-%Y"))
```




From Fitting line, we can assume that installs is getting more and more since 01-01-2021

# 2.The trend of D1_DAU

We want to know the trend of D1_DAU.

Overview of Installs with line chart and linear fit line.

Load the `ggplot2` package to plot the line chart, and `scales` package is to Formate dates on X axis in `ggplot2`

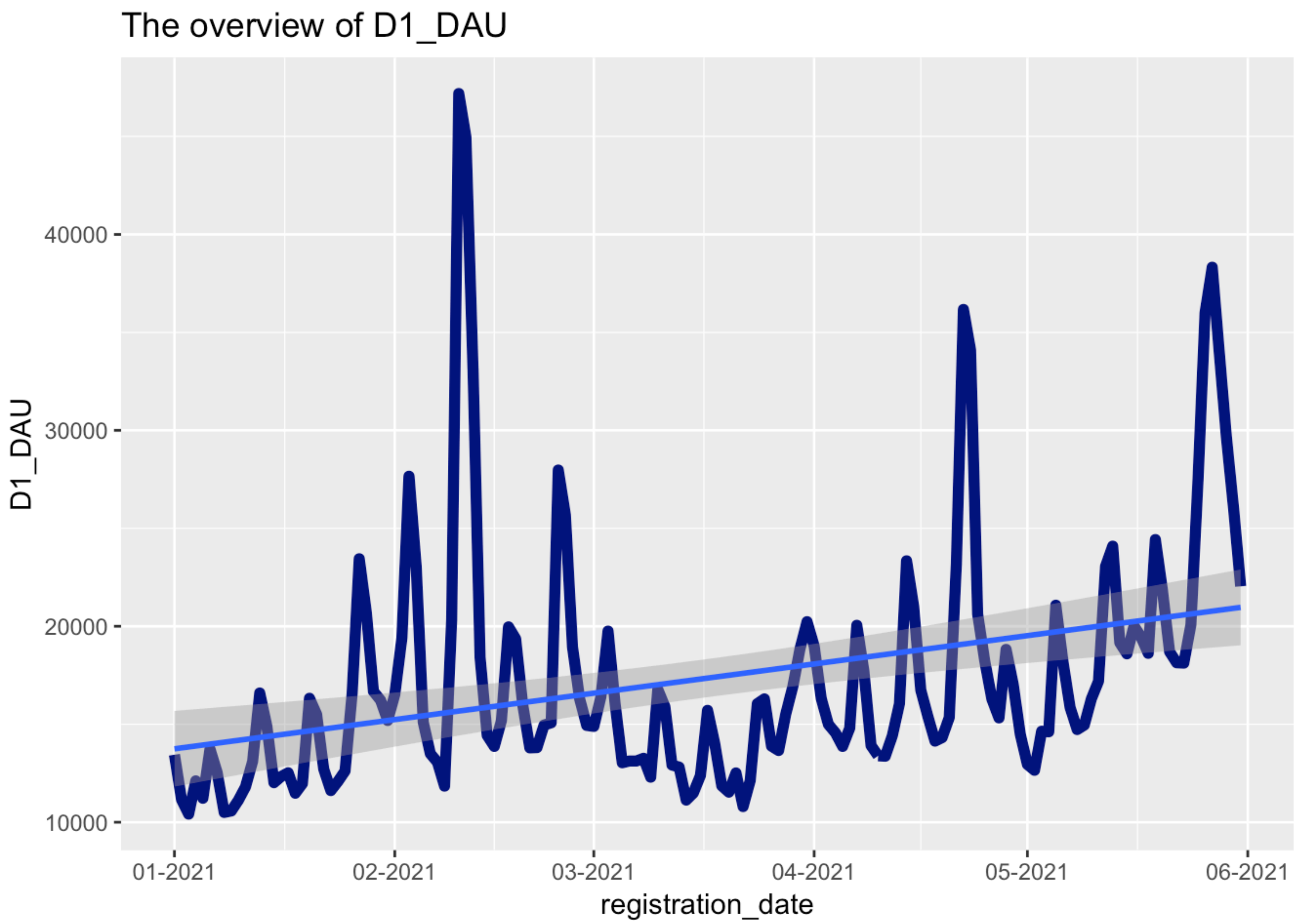
```
require(ggplot2)
require(scales)
```

Define columns as the dataframe we want to plot, and using `as.Date` to transfer `df$registration_date` as a date column

```
df2 <- data.frame(df$registration_date,df$d1_dau)
date2 <- as.Date(df$registration_date)
```

Using `ggplot()` to generate line chart. Add linear fit line with `method = "lm"`. `scale_x_dat` is the function we use to format date

```
ggplot(data = df2, aes(x = date2, y = df$d1_dau,group = 1))+
  geom_line(color = "navy", size = 2)+
  geom_smooth(method = "lm")+
  labs(x = "registration_date", y = "D1_DAU",
       title = "The overview of D1_DAU")+
  scale_x_date(labels = date_format("%m-%Y"))
```




From Fitting line, we can assume that D1_DAU is getting more and more since 01-01-2021

# 3.How is the performance on Retention_rate?

We want to know the performance of Retention_rate.

Firstly, we need to calculate the Retention_Rate

Assumption:installs mean day0 user who download the app and sign up the app on day0

add new column Retention_Day_1,Retention_Day_3,Retention_Day_7 add new column ,Retention_Day_14,Retention_Day_30,Retention_Day_60

```
df$Retention_Day_1<-(df$d1_dau/df$installs)
df$Retention_Day_3<-(df$d3_dau/df$installs)
df$Retention_Day_7<-(df$d7_dau/df$installs)
df$Retention_Day_14<-(df$d14_dau/df$installs)
df$Retention_Day_30<-(df$d30_dau/df$installs)
df$Retention_Day_60<-(df$d60_dau/df$installs)
```

In the regular situation, retention_rate should not fluctuate a lot in the overall time period.

As a result, we did the shapiro.test on d1,d3 and d7 retention_rate to know if they are normal distributed.

First, we want to know if d1_retention is normally distributed.

HO:d1_retention follow the normal distribution ,HA:Reject the normal distribution

```
shapiro.test(df$Retention_Day_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Retention_Day_1
## W = 0.96684, p-value = 0.001052
```

W = 0.96684, p-value = 0.001052<0.05, reject H0 Retention_Day_1 is not normally distributed.

Second, we want to know if d3_retention is normally distributed.

HO:d3_retention follow the normal distribution ,HA:Reject the normal distribution

```
shapiro.test(df$Retention_Day_3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Retention_Day_3
## W = 0.97297, p-value = 0.004504
```

W = 0.97297, p-value = 0.004504 < 0.05, reject H0 Retention_Day_1 is not normally distributed.

Third, we want to know if d7_retention is normally distributed.

HO:d7_retention follow the normal distribution ,HA:Reject the normal distribution

```
shapiro.test(df$Retention_Day_7)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$Retention_Day_7
## W = 0.99344, p-value = 0.7262
```

W = 0.99344, p-value = 0.7262 > 0.05, Accept H0 Retention_Day_7 is normally distributed.

Campaign d7_Retention_Rate performance overview

Import the campaign d7 data

```
residentuser<-data.frame(df$registration_date,df$installs,df$d7_dau,df$Retention_D
ay_7)
```

Using `qcc` package AS Quality Control Charts to perform statistical quality control

```
require(qcc)
```

```
## Loading required package: qcc
```

```
## Package 'qcc' version 2.7
```

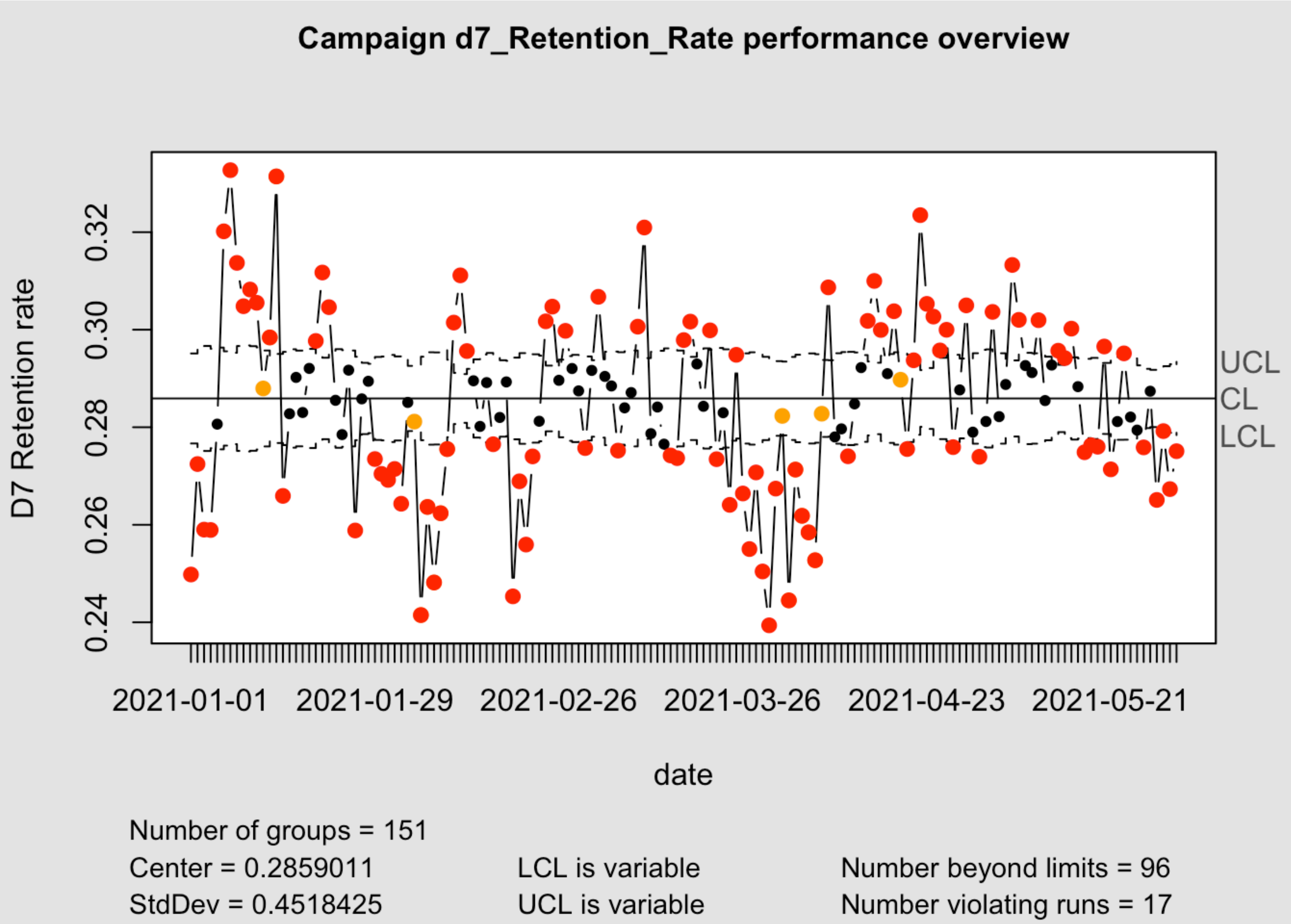
```
## Type 'citation("qcc")' for citing this R package in publications.
```

```
residentuser$date<-as.Date(df$registration_date)
attach(residentuser)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     date
```

'type = "p"' one-at-time data of a continuous process variable

```
sol<-qcc(df$d7_dau,df$installs,labels = date,
         type = "p",nsigmas = 3,
         title="Campaign d7_Retention_Rate performance overview",
         xlab="date",ylab="D7 Retention rate")
```




There are 96 days(red points) have abnormal situation.

From campaign retion_rate optimization point of view, we could check those time period with better retention rate performance.

Like the middle of 2021-01, and from middle of 2021-04 to middle of 2021-05.

Which campaign is the key player on d7_retention_rate during that time period? Is it because we using the new creative? If yes, could we using new creative on the future campaigns?

As for fraud prevention perspective, we should take care time periods on 2021-02-02 and 2021-04-07.

Because there are a lot of days d7_Retention_Rate perform below the three sigma low bar, we can suspect there might be some fraudulent behaviors happened.

Which publisher's installs contribute the most bad d7_Retention_Rate?

We should ask channel to add them into black list ,start the fraud investigation and pay us fraud rebate if those are fraud confirmed.

# 4.The trend of ARPDAU_Day_1

We want to know the trend of ARPDAU_Day_1.

Firstly, we need to calculate the ARPDAU_Day_1

```
df$ARPDAU_Day_1<-((df$d1_iap_revenue+df$d1_ads_revenue)/df$d1_dau)
```

Overview of ARPDAU_Day_1 with line chart and linear fit line

Load the `ggplot2` package to plot the line chart, and `scales` package is to Formate dates on X axis in `ggplot2`

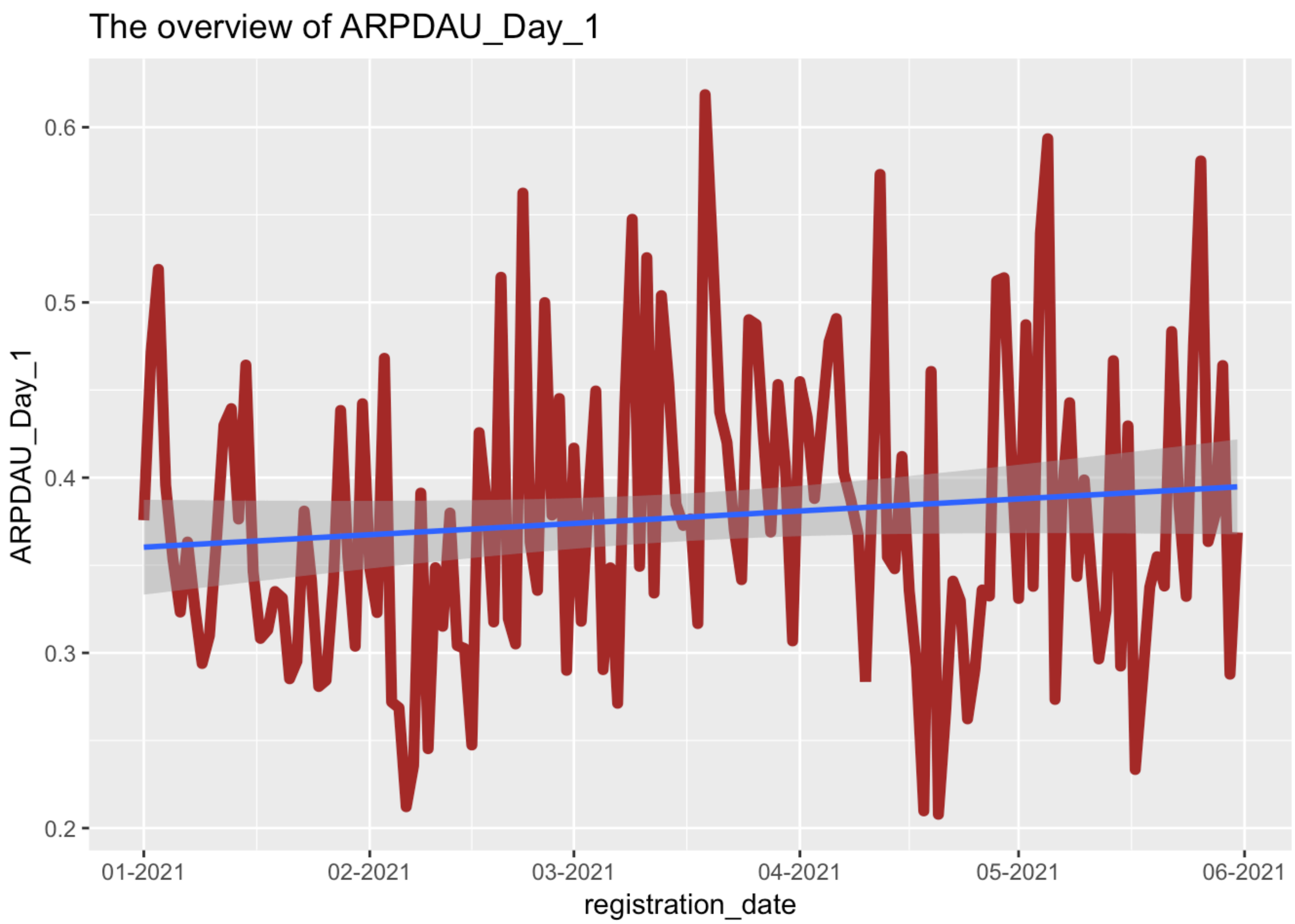
```
require(ggplot2)
require(scales)
```

Define columns as the dataframe we want to plot, and using `as.Date` to transfer `df$registration_date` as a date column

```
df4 <- data.frame(df$registration_date,df$ARPDAU_Day_1)
date4 <- as.Date(df$registration_date)
```

Using `ggplot()` to generate line chart. Add linear fit line with `method = "lm"` . `scale_x_dat` is the function we use to format date

```
ggplot(data = df4, aes(x = date4, y = df$ARPDAU_Day_1,group = 1))+
  geom_line(color = "brown", size = 2)+
  geom_smooth(method = "lm")+
  labs(x = "registration_date", y = "ARPDAU_Day_1",
       title = "The overview of ARPDAU_Day_1")+
  scale_x_date(labels = date_format("%m-%Y"))
```




From Fitting line, we can assume that ARPDAU_Day_1 is increasing slightly since 01-01-2021

# 5.The trend of LTV_Day_1

We want to know the trend of LTV_Day_1.

Firstly, we need to calculate the LTV_Day_1

CLTV calculation from AppLovin
`https://blog.applovin.com/must-know-kpis-measuring-mobile-games-performance/`

LTV=ARPUx(1/churn_rate)+(referral value)

churn_rate=1-Retention_rate,here we assume (referral value)=0

```
df$LTV_Day_1<-(df$ARPDAU_Day_1*(1/(1-df$Retention_Day_1)))
```

Overview of LTV_Day_1 with line chart and linear fit line

Load the `ggplot2` package to plot the line chart, and `scales` package is to Formate dates on X axis in `ggplot2`

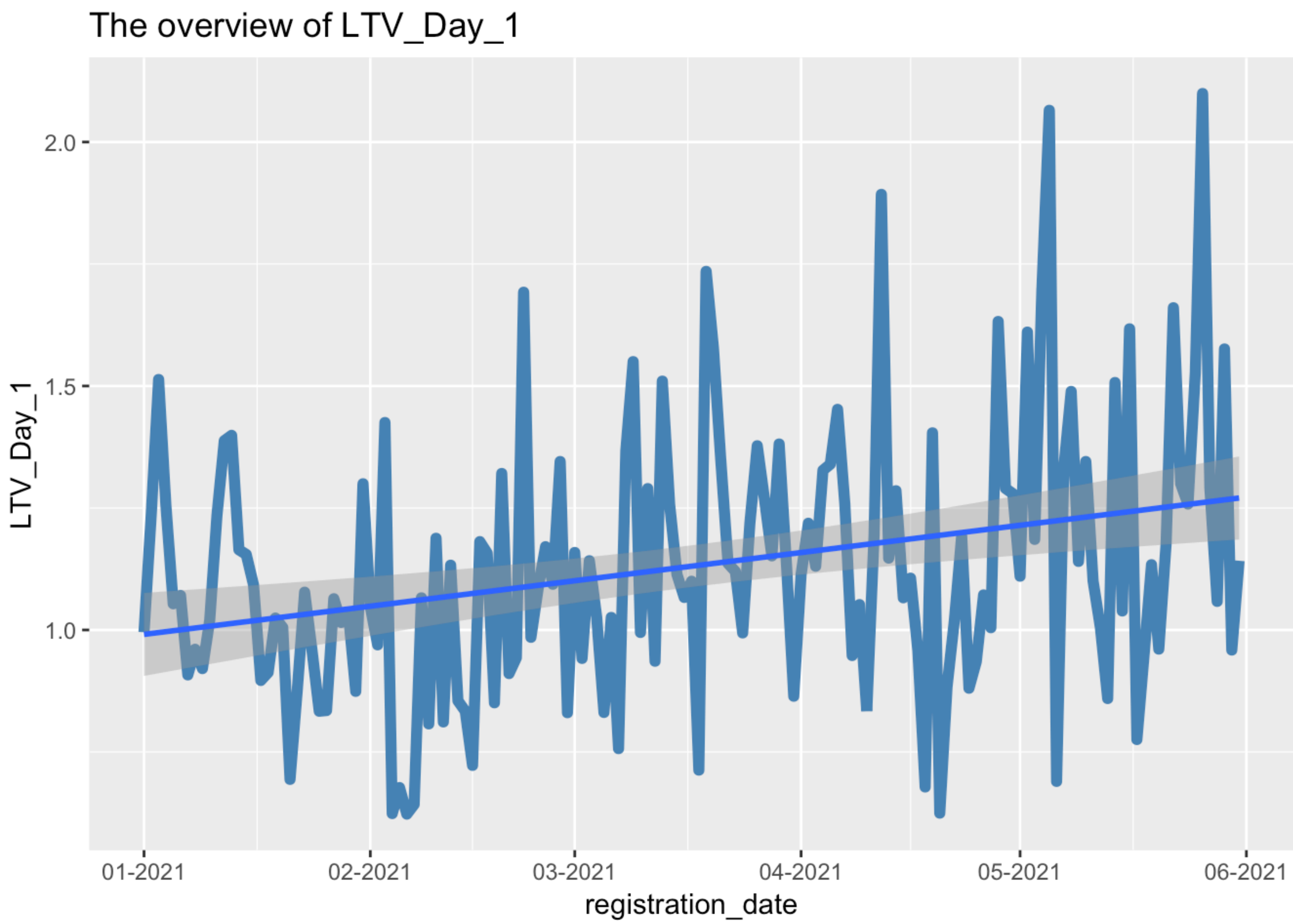
```
require(ggplot2)
require(scales)
```

Define columns as the dataframe we want to plot, and using `as.Date` to transfer `df$registration_date` as a date column

```
df5 <- data.frame(df$registration_date,df$LTV_Day_1)
date5 <- as.Date(df$registration_date)
```

Using `ggplot()` to generate line chart. Add linear fit line with `method = "lm"`. `scale_x_dat` is the function we use to format date

```
ggplot(data = df5, aes(x = date5, y = df$LTV_Day_1,group = 1))+
  geom_line(color = "steelblue", size = 2)+
  geom_smooth(method = "lm")+
  labs(x = "registration_date", y = "LTV_Day_1",
       title = "The overview of LTV_Day_1")+
  scale_x_date(labels = date_format("%m-%Y"))
```




From Fitting line, we can assume that LTV_Day_1 is increasing since 01-01-2021

**2.How is the campaign performance look like?**

The attached dataset contains fictional paid user acquisition campaigns for a game. Please evaluate the performance of the campaigns and identify “good”“,”average" and “bad” performing campaigns briefly providing your reasoning and bucketing criteria.

# Jazz report on task analysis

Because it’s really hard to compare the performance between different campaigns due to different size of spend and dau.

We decide to define a campaign user quality score(CUQ). The campaign user quality score consists of: CPM_Spend,CTR,Conversion_rate,Retention_Day_7,ARPDAU_D7,ROI_180,ROAS.

Here are the scoring rules.

We will use the weight score like below, it can adjust depend on our need.

Fraud_metric(-):CPM_Spend:10%,CTR:10%,Conversion_rate:10%

Key_index_metric(+~-):Retention_Day_7:10%,ARPDAU_D7:10%,ROI_180:20%,ROAS:20%

We take overall time period as comparison, all campaign CUQ set to 0 in the original.

campaign user quality score(CUQ)= 0 + Key_index_metric(+~-)+Fraud_metric(-)

As a result, you will be labeled

as “good” campaign if your CUQ>0,

“average”:CUQ=0,

“bad”:CUQ<0

## Read Task4 CSV into R

```
df <- read.csv(file="ua_analyst_task4_campaigns.csv", header=TRUE, sep=",")
```

# Date overview

Before we have a deep look on We will calculate
CPM_Spend,CTR,Conversion_rate,ARPDAU_D7,Retention_Day_7,ROI_180,ROAS

We will calculate every metric here.

```
df$CPM_Spend<- (df$spend*1000)/df$impressions
df$CTR<- df$clicks/df$impressions
df$Conversion_rate<- df$installs/df$clicks
df$CPI<-df$spend/df$installs
df$Retention_Day_7<-(df$d7_dau/df$installs)
df$ARPDAU_D7<-(df$d7_revenue/7/df$d7_dau)
```

### 1.ROI_180 performance evaluation

ROI_180 is one of important factors to evaluate the success of campaign.

Evaluate campaign performance from ROI_180-(LTV_180/CPI) perspective

We are wondering how is the each campaign current ROI performance?

Existing ROI = LTV_180 / current campaign CPI

Firstly, we need to calculate the Retention_Rate

add new column Retention_Day_1,Retention_Day_7,Retention_Day_14

add new column ,Retention_Day_30,Retention_Day_60,Retention_Day_120

add new column ,Retention_Day_150,Retention_Day_180

```
df$Retention_Day_1<-(df$d1_dau/df$installs)
df$Retention_Day_7<-(df$d7_dau/df$installs)
df$Retention_Day_14<-(df$d14_dau/df$installs)
df$Retention_Day_30<-(df$d30_dau/df$installs)
df$Retention_Day_60<-(df$d60_dau/df$installs)
df$Retention_Day_120<-(df$d120_dau/df$installs)
df$Retention_Day_150<-(df$d150_dau/df$installs)
df$Retention_Day_180<-(df$d180_dau/df$installs)
```

Assume LTV_180 = ARPDAU_180*Life_time(1-180)

This LTV function got from `Eric???s Seufert???s` lecture from GDC (Retention Approach) https://mobiledevmemo.com/two-methods-modeling-ltv-spreadsheet/ (https://mobiledevmemo.com/two-methods-modeling-ltv-spreadsheet/)

It assumes the retention function is a power function `(y=a*x^b)` and that ARPDAU is constant.

Assume retention rate will follow the power function y=ax^b, x:days since install,y:retention rate

Create a dataframe `r` to import all retention rate data

```
x<-c(1,7,14,30,60,120,150,180)
x
```

```
## [1]   1   7  14  30  60 120 150 180
```

```
r<-data.frame(df$campaign_id,df$Retention_Day_1,df$Retention_Day_7,df$Retention_Da
y_14,df$Retention_Day_30,df$Retention_Day_60,df$Retention_Day_120,df$Retention_Day
_150,df$Retention_Day_180)
class(r)
```

```
## [1] "data.frame"
```

```
as.numeric(r[r$df.campaign_id == 1,2:9])
```

```
## [1] 0.48262951 0.21176561 0.14766849 0.08052189 0.03151780 0.02188682
## [7] 0.01619316 0.01619316
```

Write an retention funtion to fetch retention from day1 to day180 by campaignID

```
retention<- function(x){
  r<-data.frame(df$campaign_id,df$Retention_Day_1,df$Retention_Day_7,df$Retention_
Day_14,df$Retention_Day_30,df$Retention_Day_60,df$Retention_Day_120,df$Retention_D
ay_150,df$Retention_Day_180)
  n<-as.numeric(r[r$df.campaign_id == x,2:9])
    return(n)
}
c<-retention(1)
c
```

```
## [1] 0.48262951 0.21176561 0.14766849 0.08052189 0.03151780 0.02188682
## [7] 0.01619316 0.01619316
```

## Life_time_model_function

Write an LT function which can provide us the Life_time(1-180) by integrating its area of distribution.

```
LT<- function(y){
  #Define days since install
  x<-c(1,7,14,30,60,120,150,180)
  r<-data.frame(df$campaign_id,df$Retention_Day_1,df$Retention_Day_7,df$Retention_
Day_14,df$Retention_Day_30,df$Retention_Day_60,df$Retention_Day_120,df$Retention_D
ay_150,df$Retention_Day_180)
  #Write an retention funtion to fetch retention from day1 to day180 by campaignID
  n<-as.numeric(r[r$df.campaign_id == y,2:9])
  #Find coefficient a,b of power function
  lmResult<-lm(log(n)~log(x))
  i<-as.numeric(coef(lmResult)["(Intercept)"])
  a<-exp(i)
  b<-as.numeric(coef(lmResult)["log(x)"])
  f <-function(x) a*(x^(b))
  #Calculate the area under power function, we can get the estimate lifetime
  #To calculate, integration value of a function, we first define a function (with
name f or some other name0 for the function as shown below.
  l<-integrate(f,1,180)$value
  return(l)
}
```

Calculate campaign1 to campaign20 Life_time

```
sapply(1:20,LT)
```

```
##  [1]  8.294319  6.504264 10.415370 10.577130  8.990027  9.538796  4.571853
##  [8] 10.876999 12.503182 10.164366  9.989207  9.234193  8.840356  6.300879
## [15]  4.361960  9.465114 13.830877 11.219470 14.505188 13.155585
```

Add Life_time_180 into the original data set

```
df$Life_time_180<-sapply(1:20,LT)
```

Calculate the ARPDAU_180

```
df$ARPDAU_Day_180<-((df$d180_revenue)/180/df$d180_dau)
```

Calculate the LTV_180

```
df$LTV_180<-df$ARPDAU_Day_180*df$Life_time_180
```

Calculate the CPI

```
df$CPI<-df$spend/df$installs
```

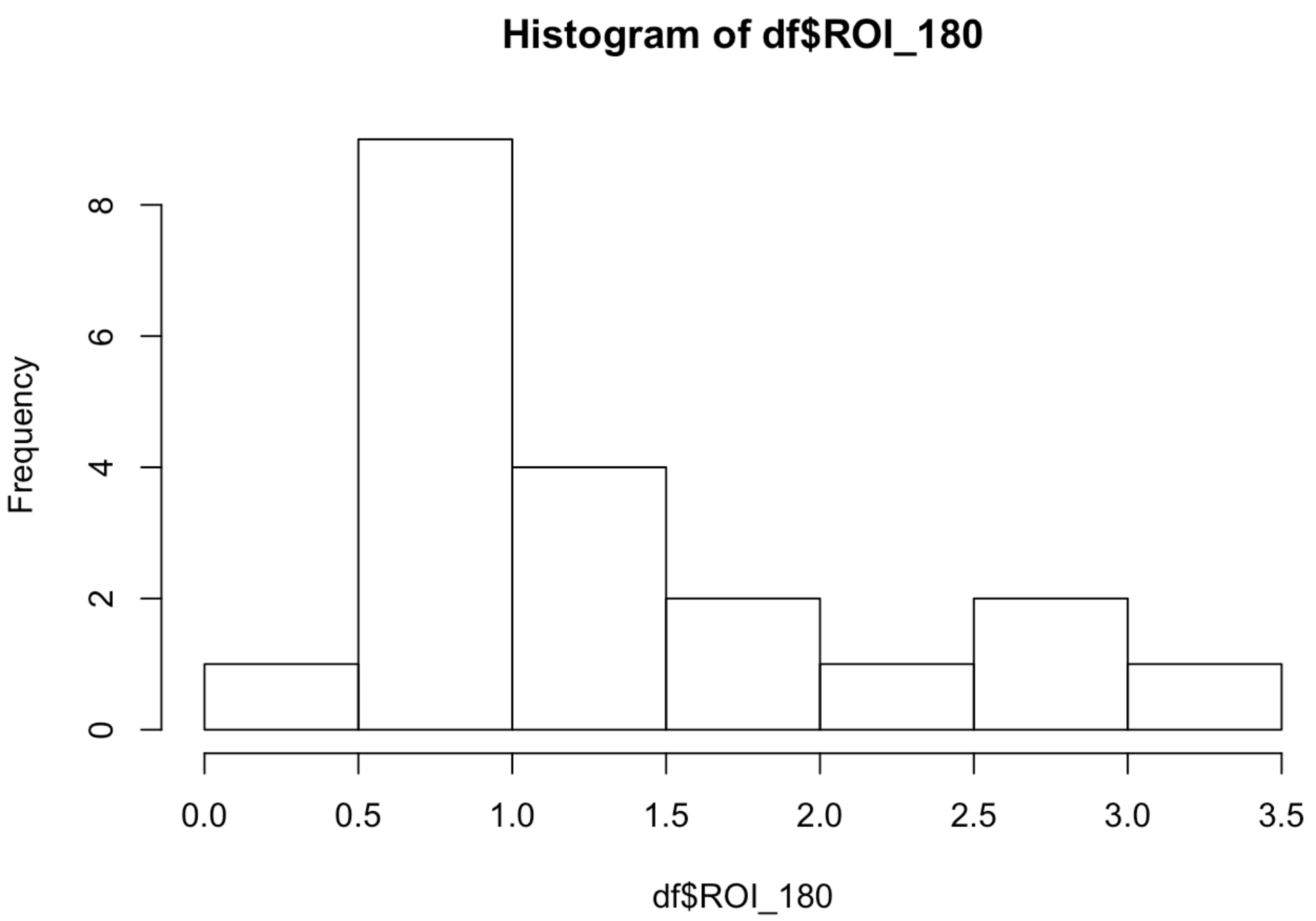
Calculate the ROI_180

```
df$ROI_180<-df$LTV_180/df$CPI
df$ROI_180
```

```
##  [1] 1.6764665 1.5203842 0.6843698 1.2587703 0.8861002 0.8408390 0.7958634
##  [8] 0.8516756 0.2971192 1.1063747 2.2906696 0.9461077 3.1847172 2.6827762
## [15] 1.1191030 2.8014313 1.2700406 0.8147215 0.8951216 0.8629128
```

First of all, we will use histogram to overview the ROI_180 distribution.

```
hist(df$ROI_180)
```




From plot, it looks more like the Skewed right (positive) distribution

We apply the function skewness from the e1071 package to compute the skewness coefficient of ROI_180.

```
require(e1071)
```

```
## Loading required package: e1071
```

```
## Warning: package 'e1071' was built under R version 3.5.2
```

```
sk = df$ROI_180
skewness(sk)
```

```
## [1] 1.054106
```

The skewness of sk is 1.054106. It indicates that the ROI_180 distribution is skewed towards the right.

The mean is on the right of the peak value.

However how do we use standard deviation to intepret the data even we have the right skewed distribution ?

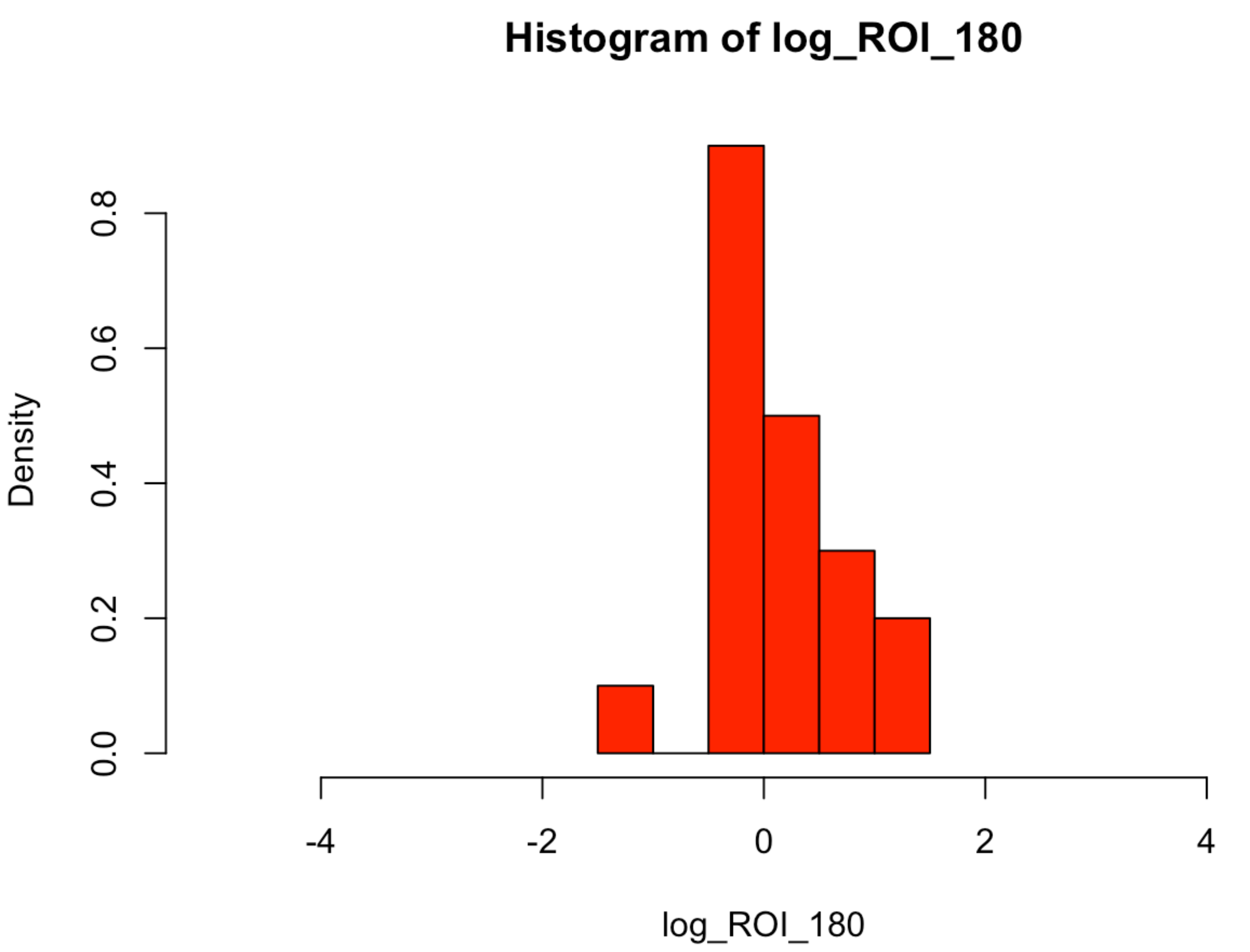
To use log transformation on data

```
w<-log(sk)
shapiro.test(w)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  w
## W = 0.92876, p-value = 0.1461
```

p-value = 0.1461 > 0.05, Accept H0 Log(ROI_180) is normally distributed.

```
log_ROI_180<- w
hist(log_ROI_180,col="red",freq=F,xlim=c(-5,5))
```



Nevertheless, we find there some missing data on histogram, we decide to use right skewed distribution rather than normal distribution.

Since this is a right skewed distribution, it doesn’t recommend to use regular boxplot to detect outlier.

Instead, we will use `adjbox` for Skew Distributions

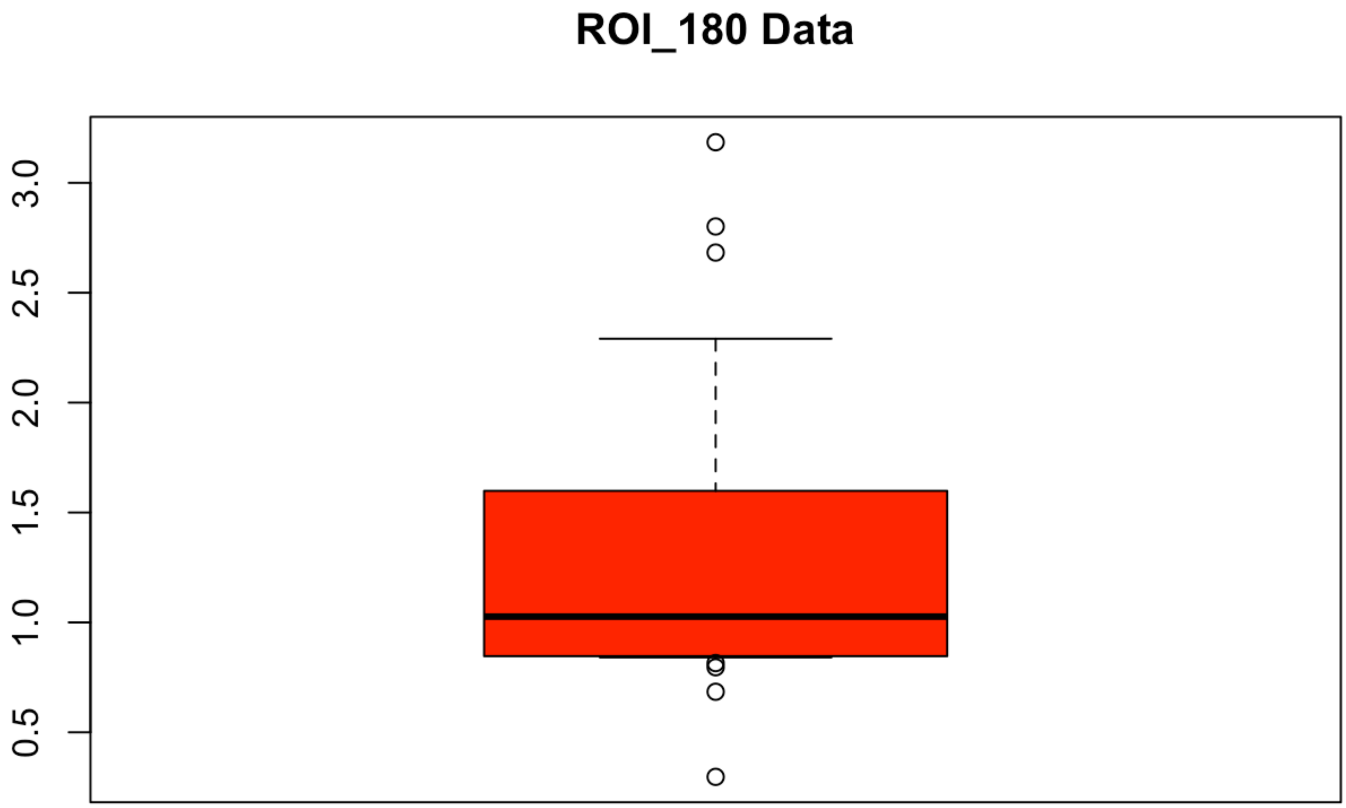
It’s boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

```
library(robustbase)
```

```
## Warning: package 'robustbase' was built under R version 3.5.2
```

```
#Since I can't find any good campaigns when coefficient = 1.5, I decide to use coe
fficient = 0.25 to find good and bad campaigns.
r_180<- data.frame(df$campaign_id,df$ROI_180)
adjbox(df$ROI_180,range = 0.25,col="red",main="ROI_180 Data")
```




```
adjboxStats(df$ROI_180,coef = 0.25)
```

```
## $stats
## [1] 0.8408390 0.8462573 1.0262412 1.5984253 2.2906696
##
## $n
## [1] 20
##
## $conf
## [1] 0.7605012 1.2919812
##
## $fence
## [1] 0.8189805 2.3984457
##
## $out
## [1] 0.6843698 0.7958634 0.2971192 3.1847172 2.6827762 2.8014313 0.8147215
```

```
adjboxStats(df$ROI_180,coef = 0.25)$out
```

```
## [1] 0.6843698 0.7958634 0.2971192 3.1847172 2.6827762 2.8014313 0.8147215
```

```
#Have an outlier detection with adjusted boxplot
length(unique(adjboxStats(df$ROI_180,coef = 0.25)$out))
```

```
## [1] 7
```

```
l6<-as.numeric(length(unique(adjboxStats(df$ROI_180,coef = 0.25)$out)))
```

Find out 7 campaigns ROI_180 lie outside Q1-0.25*exp(3M) and Q3+0.25*exp(3M)

Where M is an index of skewness of the uncontaminated part of the data

Details please refer user603 answer on https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data (https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data)

And adjboxStats description https://rdrr.io/rforge/robustbase/man/adjboxStats.html (https://rdrr.io/rforge/robustbase/man/adjboxStats.html)

Create function to filter those campaigns ID correspond values fall into outlier list.

```
ROI_180_O<-function(x){
  ROI_180_outlier<-as.numeric(adjboxStats(df$ROI_180,coef = 0.25)$out)
  df[df$ROI_180 == ROI_180_outlier[x],1]
}
ROI_180_O(1)
```

```
## [1] 3
```

```
sapply(1:l6,ROI_180_O)
```

```
## [1]  3  7  9 13 14 16 18
```

### Conclusion_1:

Campaign 13,14,16 have ROI_180 higher Q3+0.25*exp(3M) which mean we should take it plus +20%

Campaign 3,7,9,18 have ROI_180 lower Q1-0.25*exp(3M) which mean we should take it plus -20%

## 2.ROAS performance evaluation

I will evaluate campaign performance from ROAS perspective.

ROAS:Return of Ads Spend

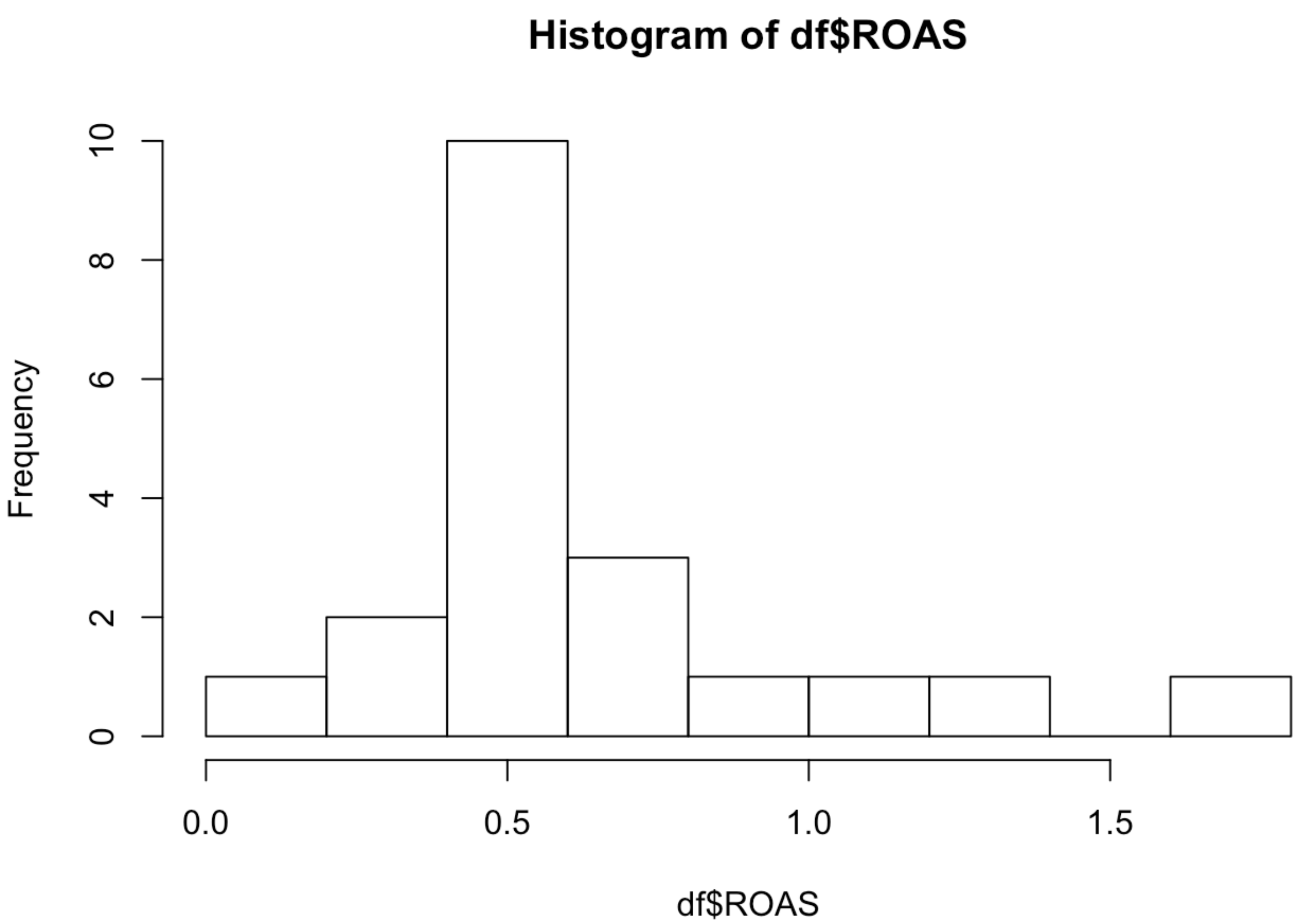
Definition:ROAS= Campaign_revenue_to_date / campaign_spend

To calculate the ROAS and add it into dataframe.

```
df$ROAS<-df$revenue_to_date/df$spend
```

Here, we will use histogram to overview the ROAS distribution

```
hist(df$ROAS)
```




From plot, it looks more like the Skewed right (positive) distribution We apply the function skewness from the `e1071` package to compute the skewness coefficient of ROAS.

```r
require(e1071)
skewness(df$ROAS)
```

```
## [1] 1.52701
```

The skewness of sk is 1.52701. It indicates that the ROAS distribution is skewed towards the right.

The mean is on the right of the peak value.

However how do we use standard deviation to intepret the data?

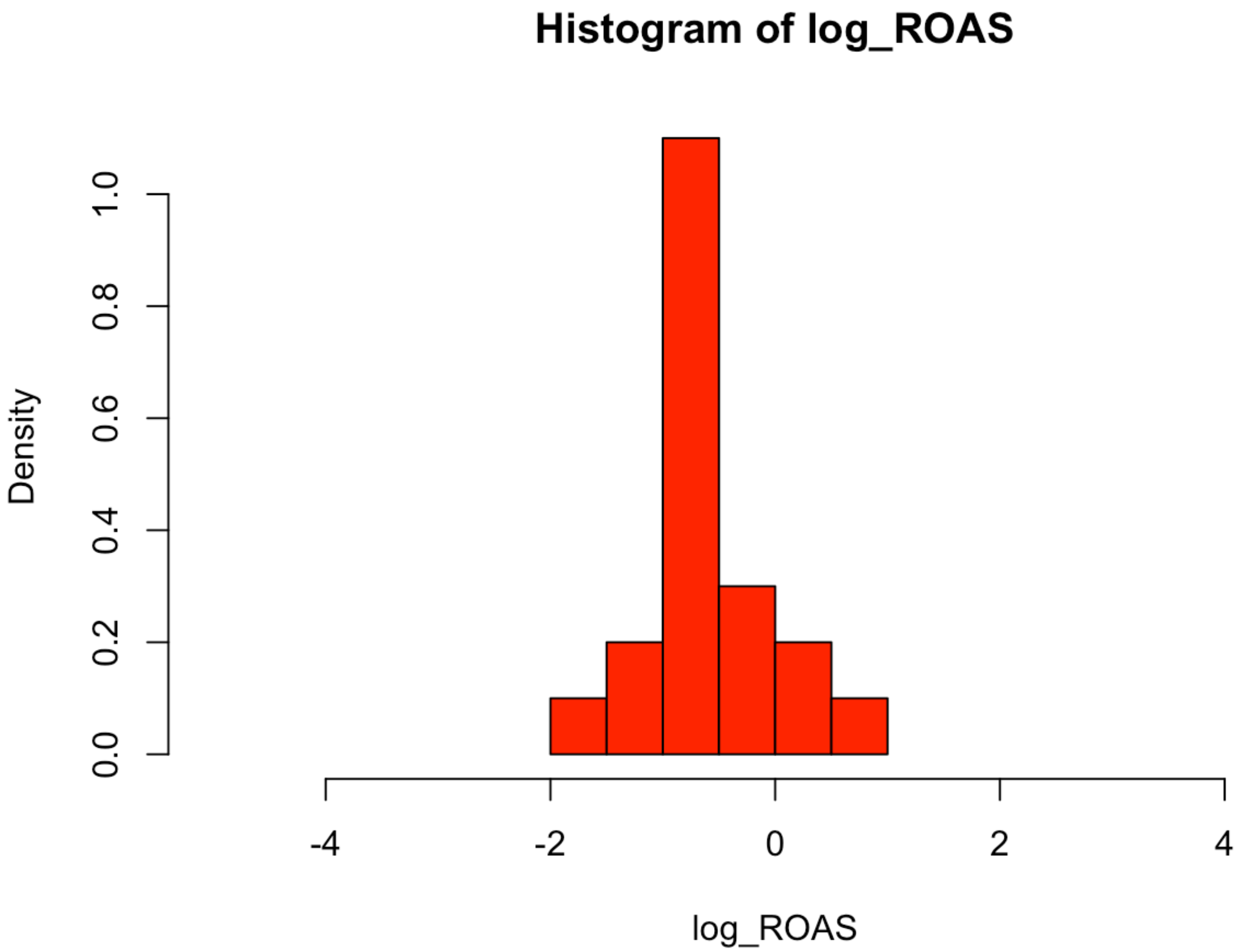
To use log transformation on data

```r
z<-log(df$ROAS)
shapiro.test(z)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  z
## W = 0.96193, p-value = 0.583
```

p-value = 0.583 > 0.05, Accept H0 Log(ROAS) is normally distributed.

```r
log_ROAS<- z
hist(log_ROAS,col="red",freq=F,xlim=c(-5,5))
```




Nevertheless, we still find out there some missing data on histogram,

As a result, we decide to use right skewed distribution rather than normal distribution.

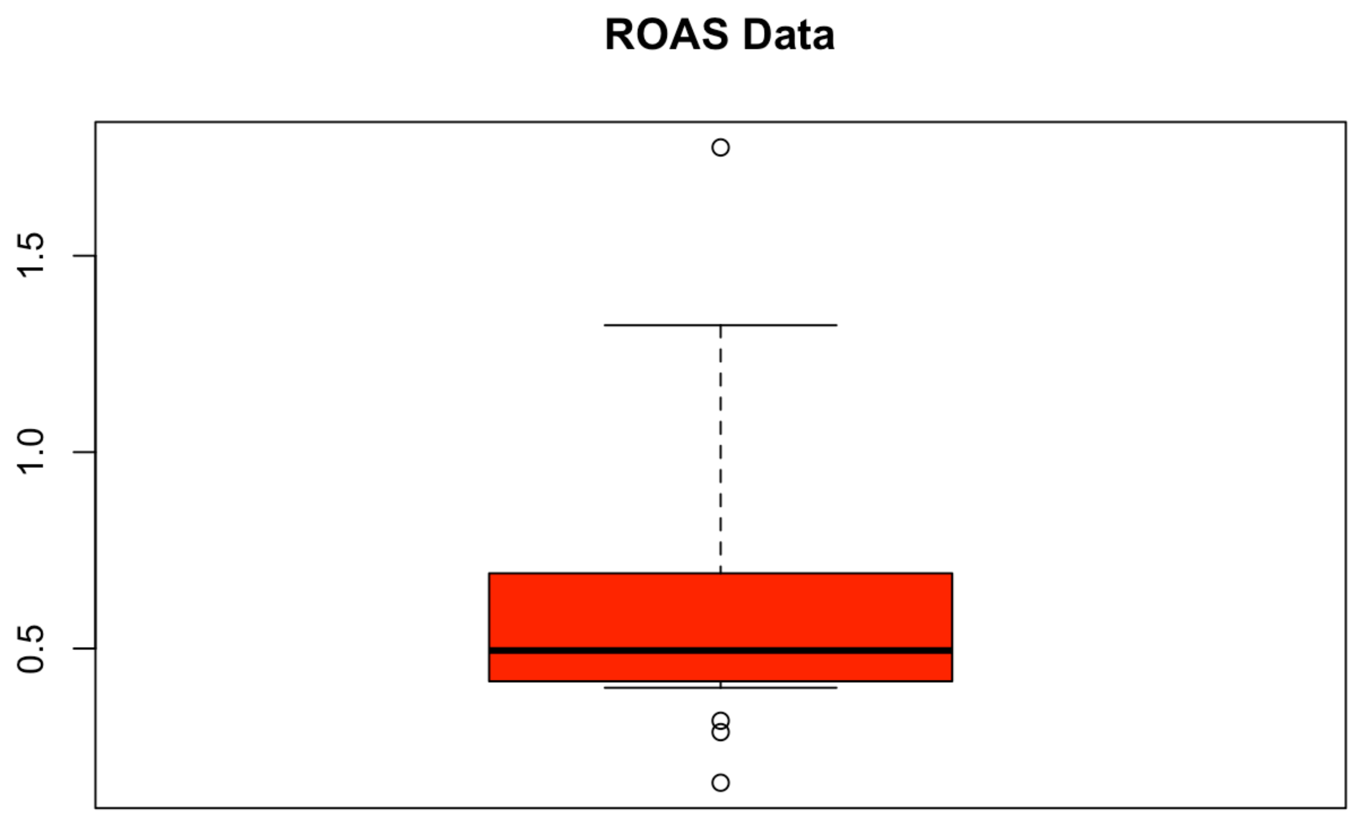
Since this is a right skewed distribution, it doesn't recommend to use regular boxplot to detect outlier.

Instead, we will use `adjbox` for Skew Distributions

It's boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

```
library(robustbase)
#Since I can't find any good campaigns when coefficient = 1.5, I decide to use coe
fficient = 1 to find good and bad campaigns.
adjbox(df$ROAS,range = 1,col="red",main="ROAS Data")
```




```
adjboxStats(df$ROAS,coef = 1)
```

```
## $stats
## [1] 0.4001575 0.4165895 0.4950473 0.6915407 1.3230051
##
## $n
## [1] 20
##
## $conf
## [1] 0.3979074 0.5921872
##
## $fence
## [1] 0.3576893 1.5647328
##
## $out
## [1] 0.2871515 0.3160798 0.1583612 1.7757609
```

```
adjboxStats(df$ROAS,coef = 1)$out
```

```
## [1] 0.2871515 0.3160798 0.1583612 1.7757609
```

```
#Have an outlier detection with adjusted boxplot
length(unique(adjboxStats(df$ROAS,coef = 1)$out))
```

```
## [1] 4
```

```
l7<-as.numeric(length(unique(adjboxStats(df$ROAS,coef = 1)$out)))
```

Find out three campaigns ROAS lie outside Q1-1.5*exp(3M) and another one outside of Q3+1.5*exp(3M)

Where M is an index of skewness of the uncontaminated part of the data

Details please refer user603 answer on https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data (https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data)

And adjboxStats description https://rdrr.io/rforge/robustbase/man/adjboxStats.html (https://rdrr.io/rforge/robustbase/man/adjboxStats.html)

Create function to filter those campaigns ID correspond values fall into outlier list.

```
ROAS_O<-function(x){
  ROAS_outlier<-as.numeric(adjboxStats(df$ROAS,coef = 1)$out)
  df[df$ROAS == ROAS_outlier[x],1]
}
ROAS_O(1)
```

```
## [1] 3
```

```
sapply(1:l7,ROAS_O)
```

```
## [1]  3  7  9 13
```

### Conclusion_2:

Campaign 13 have ROAS higher Q3+1*exp(3M) which mean we should take it plus +20%

Campaign 3,7,9 have ROAS lower Q1-1*exp(3M) which mean we should take it plus -20%

## 3.CPM_Spend performance evaluation

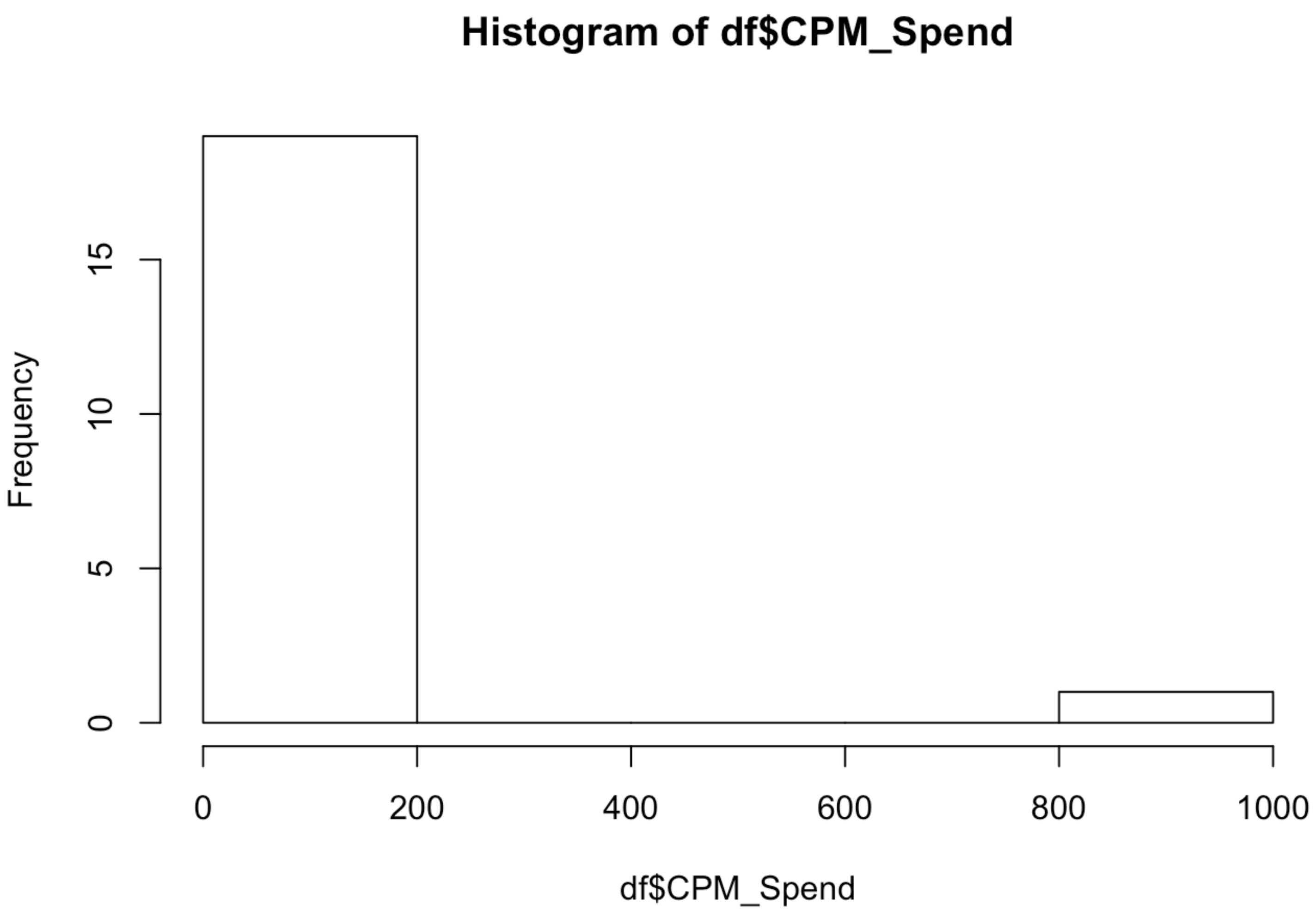
Have an overview with summary()

```
summary(df$CPM_Spend)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.11   16.71   23.02   70.03   29.61  884.61
```

Using histogram to overview the distribution

```
hist(df$CPM_Spend)
```




From plot, it looks more like the Skewed right (positive) distribution We apply the function skewness from the e1071 package to compute the skewness coefficient of eCPM_Spend.

```
require(e1071)
skewness(df$CPM_Spend)
```

```
## [1] 3.764521
```

The skewness of sk is 3.764521. It indicates that the CPM_Spend distribution is skewed towards the right.

The mean is on the right of the peak value.

Since this is a right skewed distribution, it doesn't recommend to use regular boxplot to detect outlier.

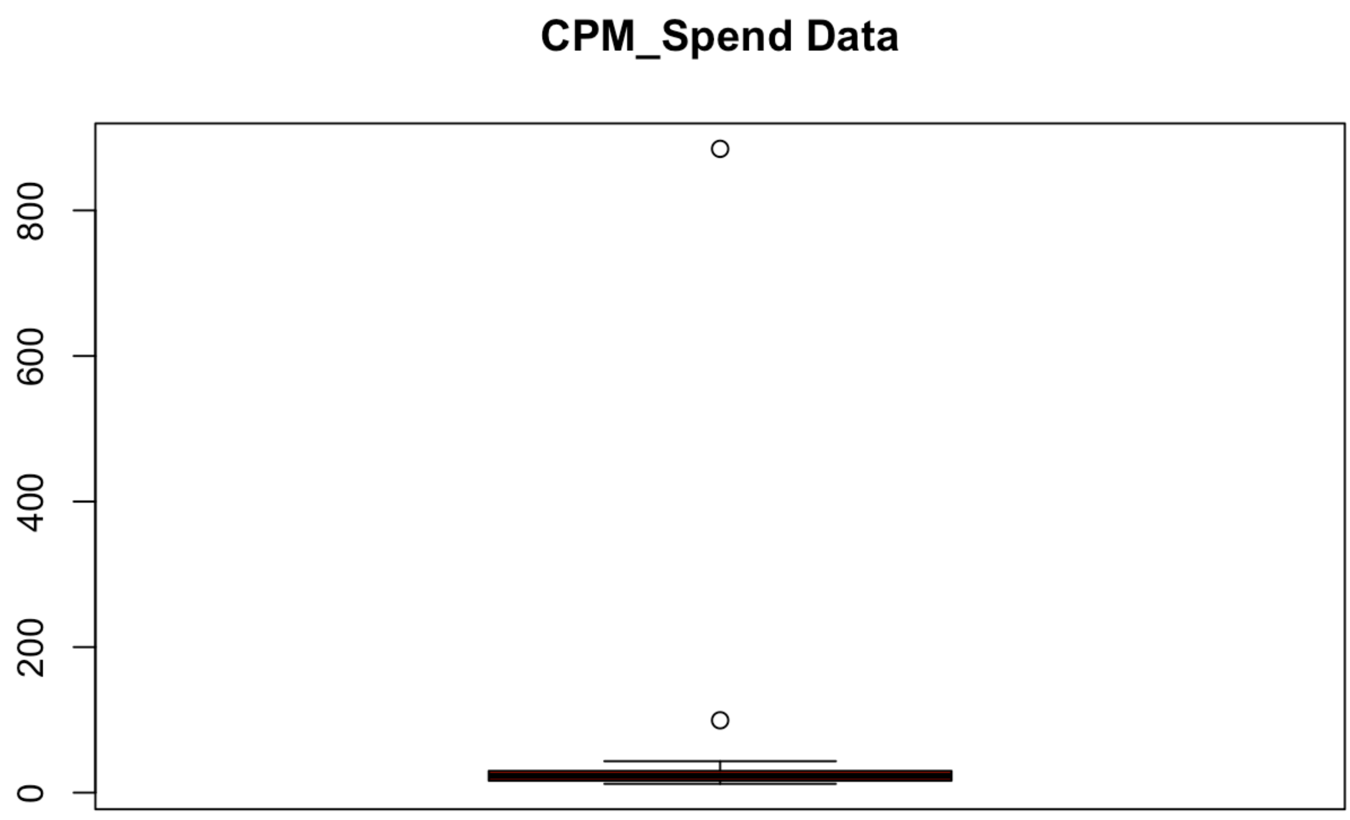
Instead, we will use `adjbox` for Skew Distributions

It's boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

Have an outlier detection with adjusted boxplot

```r
library(robustbase)
adjbox(df$CPM_Spend,col="red",main="CPM_Spend Data")
```




View the outlier with `adjboxStats()$out` function

```r
adjboxStats(df$CPM_Spend)$out
```

```
## [1] 884.6109  99.4243
```

```r
length(unique(adjboxStats(df$CPM_Spend)$out))
```

```
## [1] 2
```

```r
l1<-as.numeric(length(unique(adjboxStats(df$CPM_Spend)$out)))
```

Find out two campaigns CPM_spend lie outside Q3+1.5*exp(3M),

Where M is an index of skewness of the uncontaminated part of the data

Details please refer user603 answer on https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data (https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data)

And adjboxStats description https://rdrr.io/rforge/robustbase/man/adjboxStats.html (https://rdrr.io/rforge/robustbase/man/adjboxStats.html)

Create function to filter those campaigns ID correspond values fall into outlier list.

```
CSO<-function(x){
  CPM_spend_outlier<-as.numeric(adjboxStats(df$CPM_Spend)$out)
  df[df$CPM_Spend == CPM_spend_outlier[x],1]
}
CSO(1)
```

```
## [1] 14
```

```
sapply(1:l1,CSO)
```

```
## [1] 14 15
```

### Conclusion_3:

It’s not normal that CPM_Spend is so high(over 99).

And campaign ID 14 and 15 have CPM_Spend over Q3+1.5*exp(3M), which mean we should take them plus -10%.

## 4.CTR performance evaluation

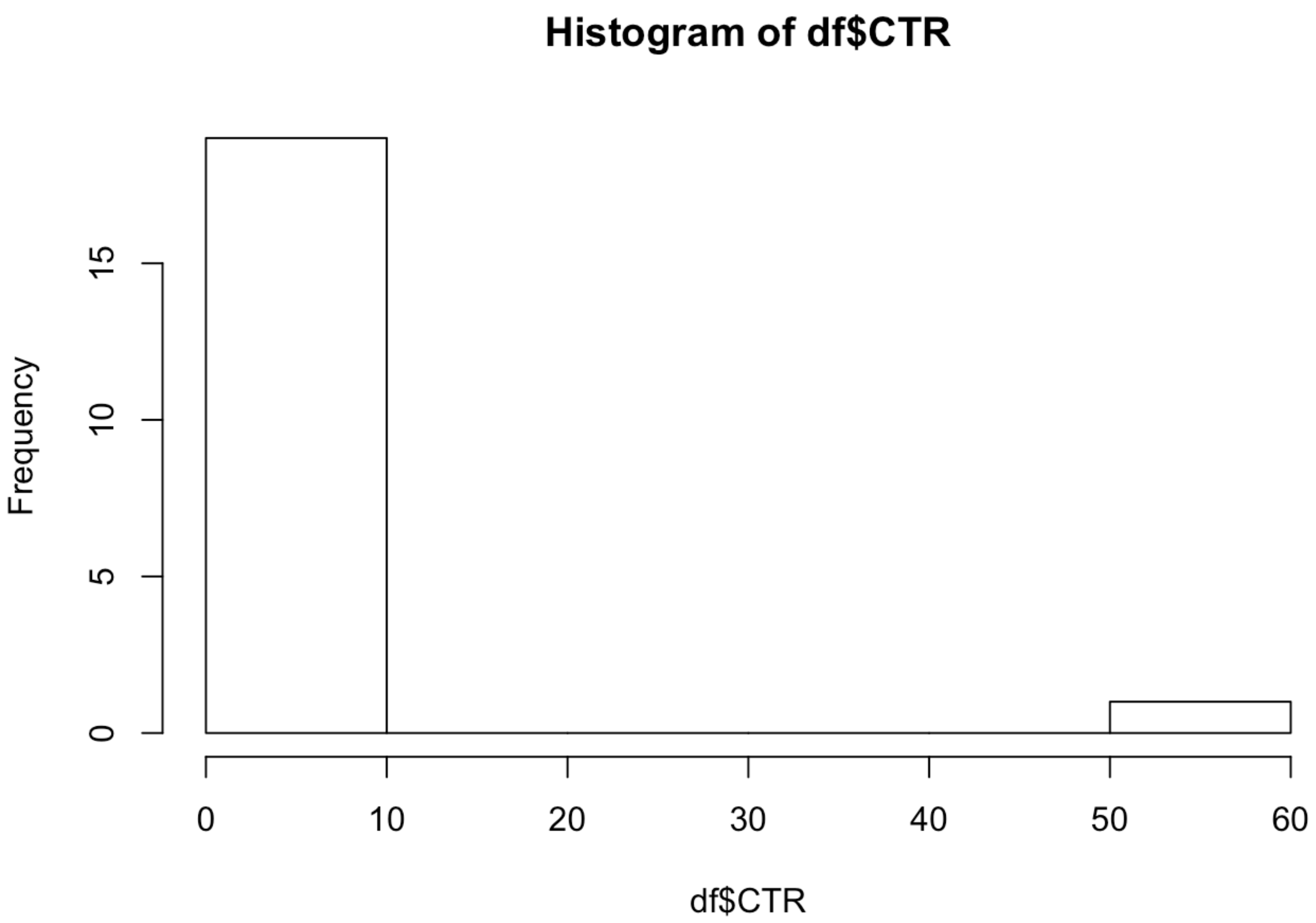
Have an overview with summary()

```
summary(df$CTR)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00353  0.02076  0.03127  3.09007  0.10969 58.90246
```

Using histogram to overview the distribution

```
hist(df$CTR)
```




From plot, it looks more like the Skewed right (positive) distribution We apply the function skewness from the e1071 package to compute the skewness coefficient of CTR.

```
require(e1071)
skewness(df$CTR)
```

```
## [1] 3.817522
```

The skewness of sk is 3.817522. It indicates that the CTR distribution is skewed towards the right.

The mean is on the right of the peak value.

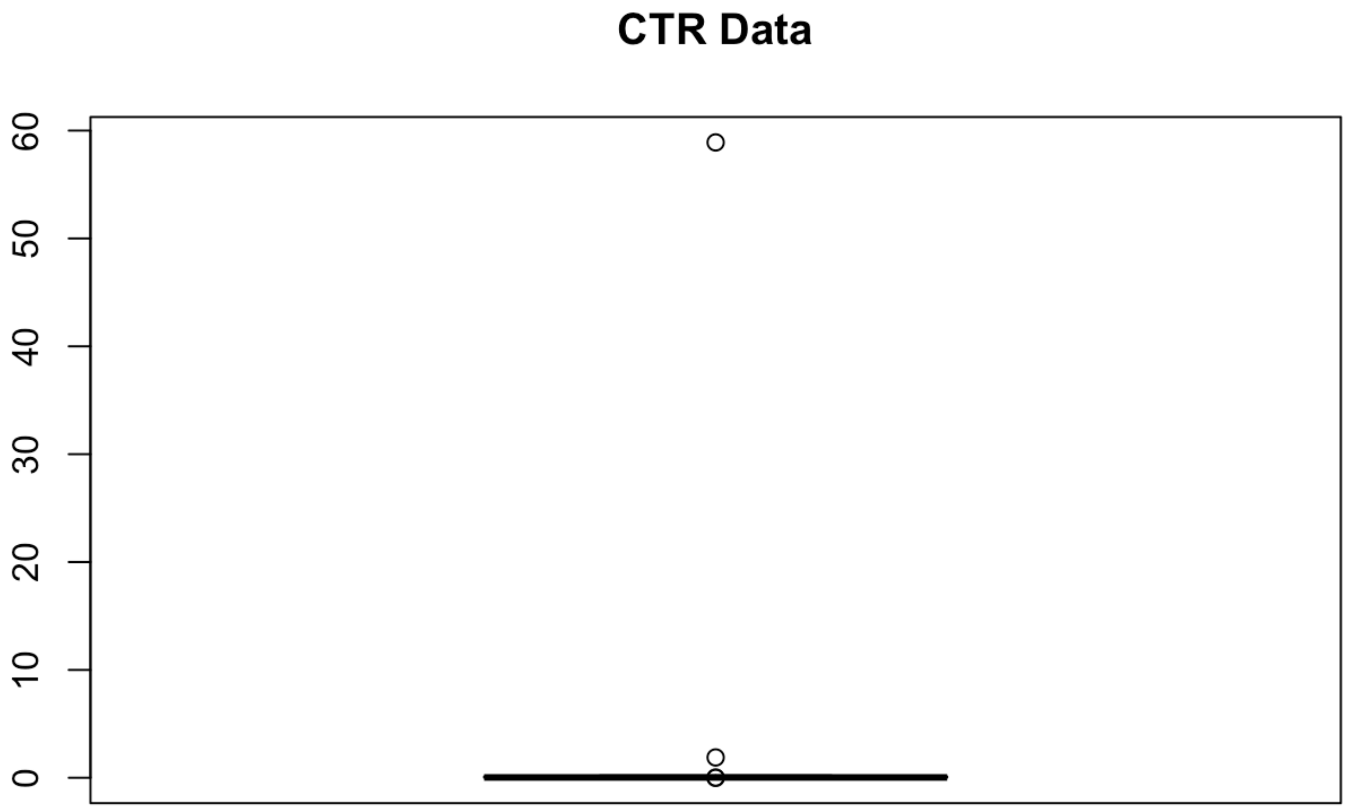
Since this is a right skewed distribution, it doesn't recommend to use regular boxplot to detect outlier.

Instead, we will use `adjbox` for Skew Distributions It's boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

Have an outlier detection with adjusted boxplot

```
require(robustbase)
adjbox(df$CTR,col="red",main="CTR Data")
```




```
adjboxStats(df$CTR)
```

```
## $stats
## [1] 0.01446149 0.02058014 0.03127485 0.11808796 0.21063882
##
## $n
## [1] 20
##
## $conf
## [1] -0.003174547  0.065724237
##
## $fence
## [1] 0.01330363 1.50655858
##
## $out
## [1]  0.009731068  0.003527742 58.902464434  1.884955540
```

```
adjboxStats(df$CTR)$out
```

```
## [1]  0.009731068  0.003527742 58.902464434  1.884955540
```

```
length(unique(adjboxStats(df$CTR)$out))
```

```
## [1] 4
```

```
l2<-as.numeric(length(unique(adjboxStats(df$CTR)$out)))
```

Find out two campaigns CTR lie outside Q3+1.5*exp(3M),another two campaigns CTR lie outside Q1-1.5*exp(3M)

Where M is an index of skewness of the uncontaminated part of the data

Details please refer user603 answer on https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data (https://stats.stackexchange.com/questions/13086/is-there-a-boxplot-variant-for-poisson-distributed-data)

And adjboxStats description https://rdrr.io/rforge/robustbase/man/adjboxStats.html (https://rdrr.io/rforge/robustbase/man/adjboxStats.html)

Create function to filter those campaigns ID correspond values fall into outlier list.

```
CTO<-function(x){
  CTR_outlier<-as.numeric(adjboxStats(df$CTR)$out)
  df[df$CTR == CTR_outlier[x],1]
}
CTO(1)
```

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

```
sapply(1:l2,CTO)
```

```
## [1]  1 10 14 15
```

### Conclusion_4:

Since it’s not normal that CTR over 100% and too low

Campaign 14 and 15 have CTR over Q3+1.5*exp(3M) which mean we should take them plus -10%.

Campaign 1 and 10 have CTR lower Q1-1.5*exp(3M) which mean we should take them plus -10%.

## 5.Conversion_rate performance evaluation

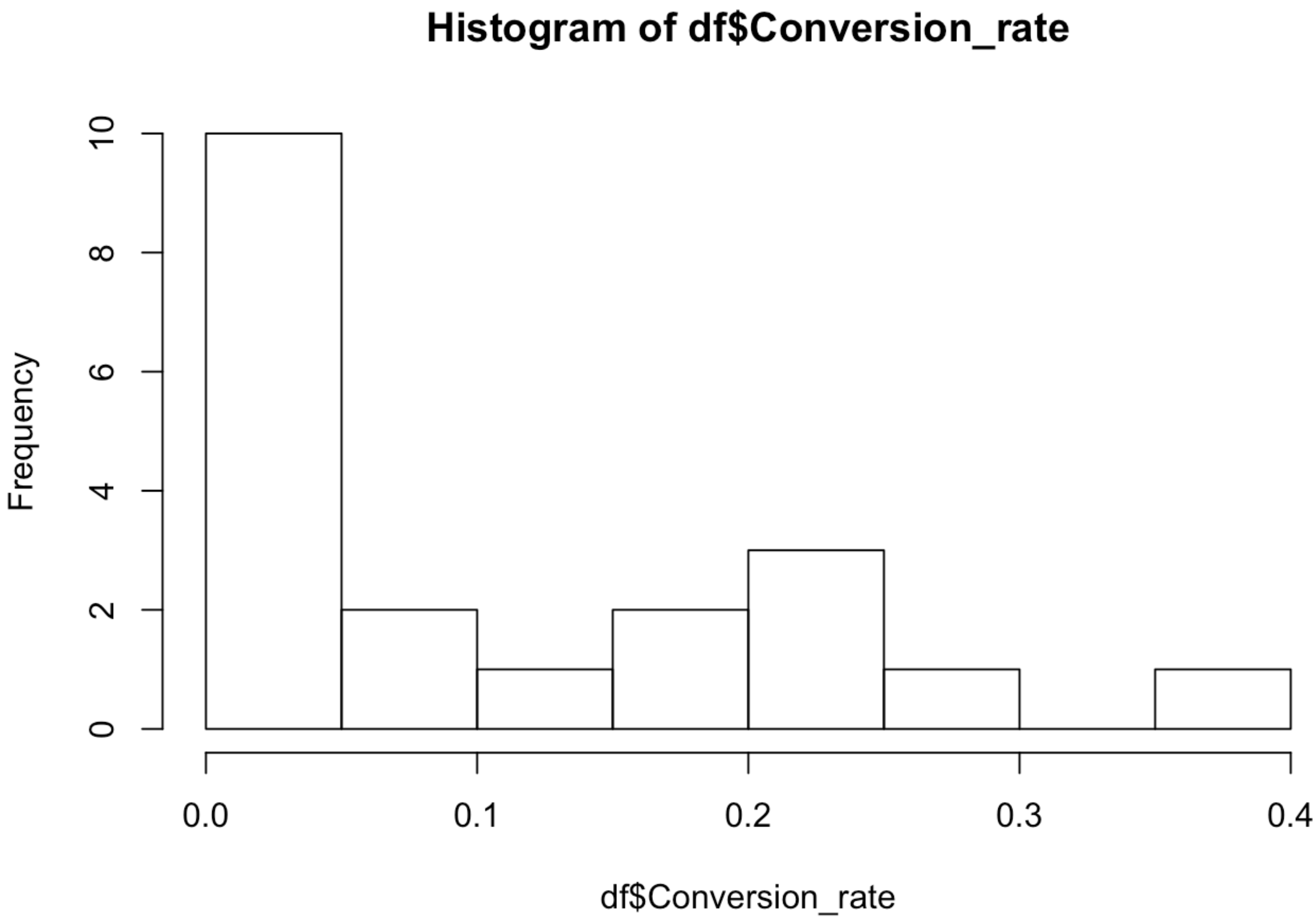
Have an overview with summary()

```
summary(df$Conversion_rate)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005361 0.032895 0.063229 0.114853 0.199958 0.357691
```

Using histogram to overview the distribution

```
hist(df$Conversion_rate)
```




From plot, it looks more like the Skewed right (positive) distribution We apply the function skewness from the e1071 package to compute the skewness coefficient of Conversion_rate.

```
require(e1071)
skewness(df$Conversion_rate)
```

```
## [1] 0.7519787
```

The skewness of sk is 0.7519787.

It indicates that the Conversion_rate distribution is skewed towards the right.

The mean is on the right of the peak value.

Since this is a right skewed distribution, it doesn't recommend to use regular boxplot to detect outlier.

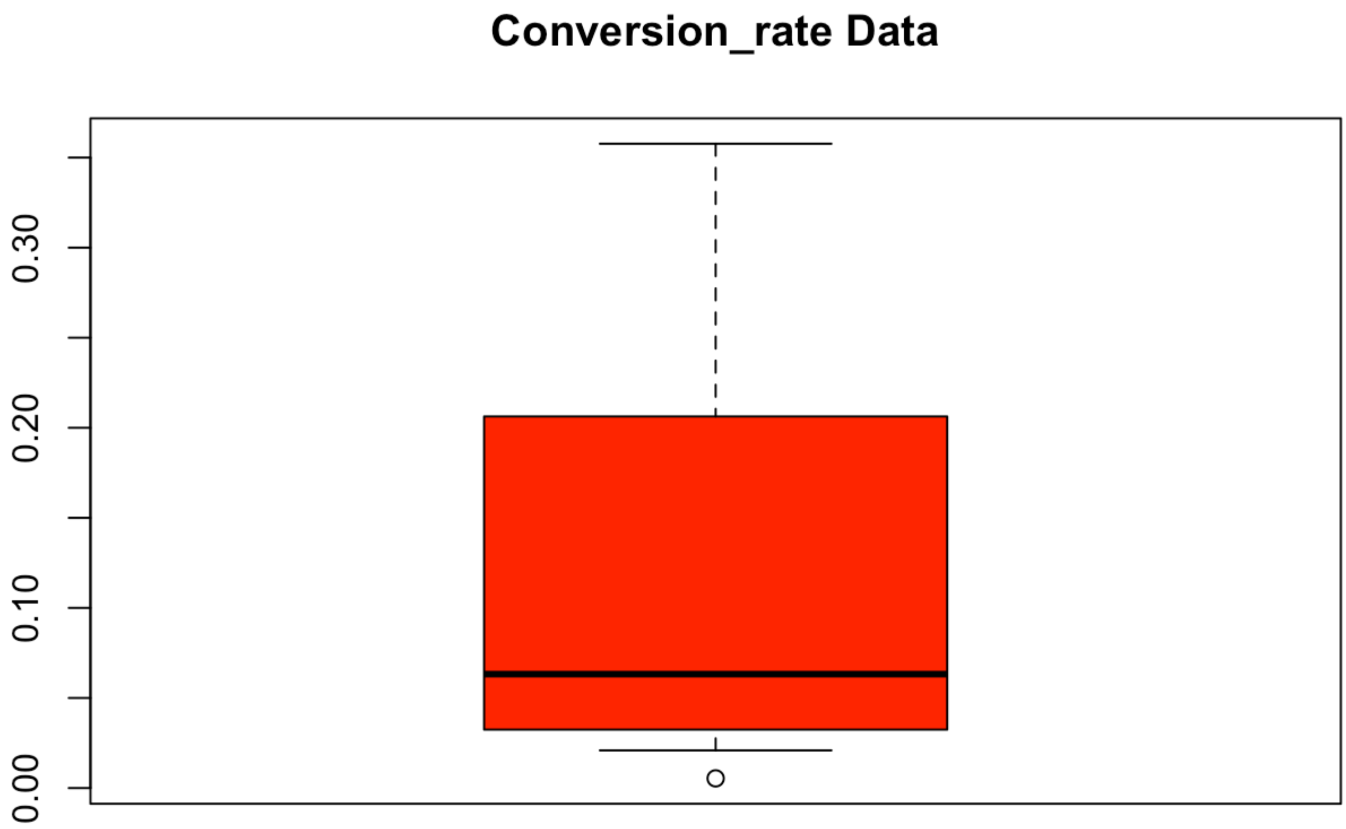
Instead, we will use `adjbox` for Skew Distributions

It's boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

Have an outlier detection with adjusted boxplot

```r
require(robustbase)
adjbox(df$Conversion_rate,col="red",main="Conversion_rate Data")
```




```r
adjboxStats(df$Conversion_rate)
```

```
## $stats
## [1] 0.02090167 0.03242073 0.06322950 0.20629655 0.35769099
##
## $n
## [1] 20
##
## $conf
## [1] 0.001799396 0.124659606
##
## $fence
## [1] 0.009777534 1.836999613
##
## $out
## [1] 0.005361067
```

```r
adjboxStats(df$Conversion_rate)$out
```

```
## [1] 0.005361067
```

```
length(unique(adjboxStats(df$Conversion_rate)$out))
```

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

```
l3<-as.numeric(length(unique(adjboxStats(df$Conversion_rate)$out)))
```

Find out one campaigns Conversion_rate lie outside Q1-1.5*exp(3M)

Create function to filter those campaigns ID correspond values fall into outlier list.

```
CVO<-function(x){
  Conversion_rate_outlier<-as.numeric(adjboxStats(df$Conversion_rate)$out)
  df[df$Conversion_rate == Conversion_rate_outlier[x],1]
}
CVO(1)
```

```
## [1] 14
```

```
sapply(1:l3,CVO)
```

```
## [1] 14
```

### Conclusion_5:

Since it's not normal that CVR too low.

Campaign 14 has Conversion_rate lower Q1-1.5*exp(3M) which mean we should take them plus -10%.

## 6.Retention_Day_7 performance evaluation

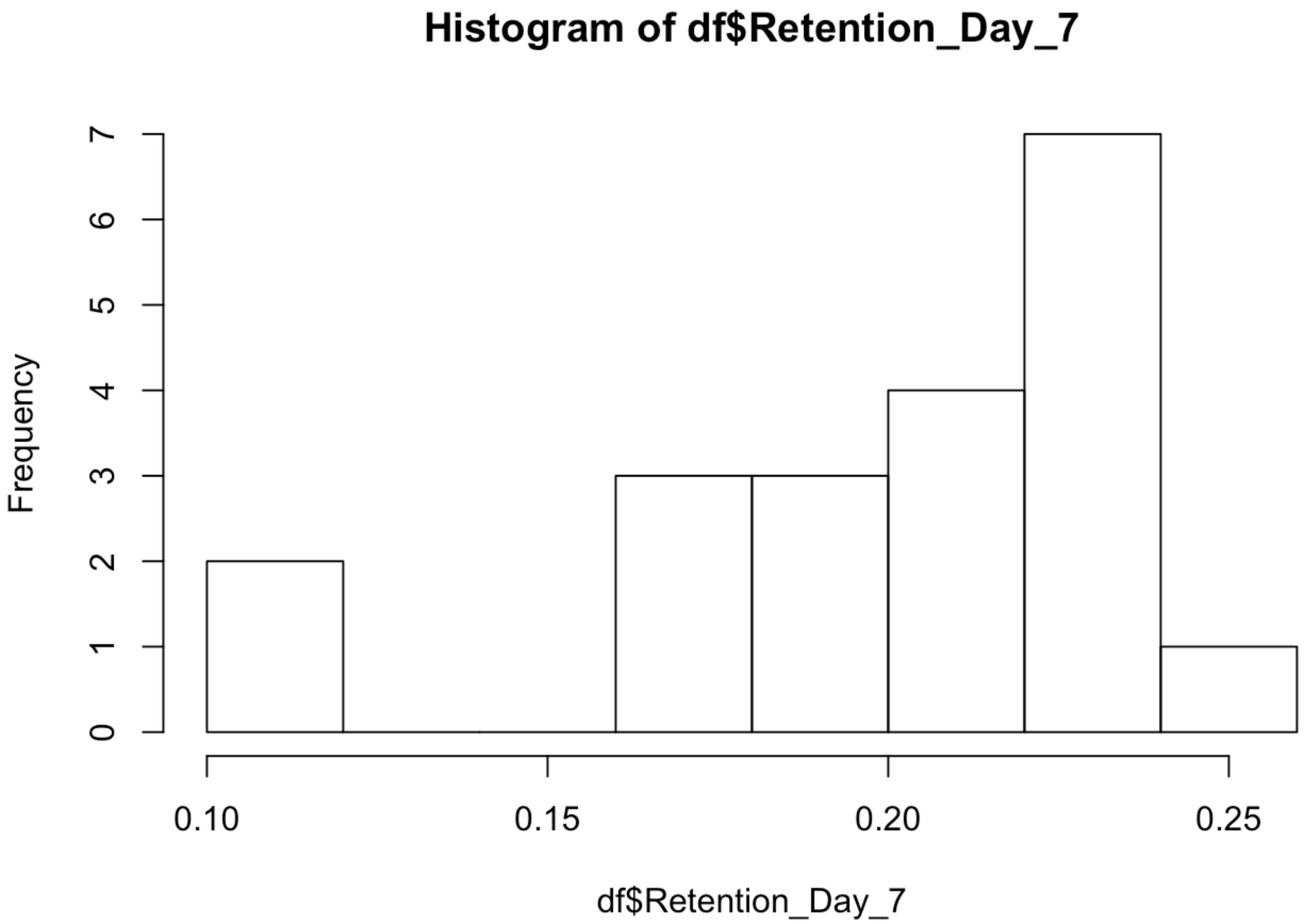
Have an overview with summary()

```
summary(df$Retention_Day_7)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.1027  0.1867  0.2083  0.1988  0.2293  0.2461
```

Using histogram to overview the distribution

```
hist(df$Retention_Day_7)
```




From plot, it looks more like the Skewed right (positive) distribution We apply the function skewness from the e1071 package to compute the skewness coefficient of Retention_Day_7.

```
require(e1071)
skewness(df$Retention_Day_7)
```

```
## [1] -1.116296
```

The skewness of sk is -1.116296.

It indicates that the Conversion_rate distribution is skewed towards the left.

The mean is on the left of the peak value.

Since this is a left skewed distribution, it doesn't recommend to use regular boxplot to detect outlier.

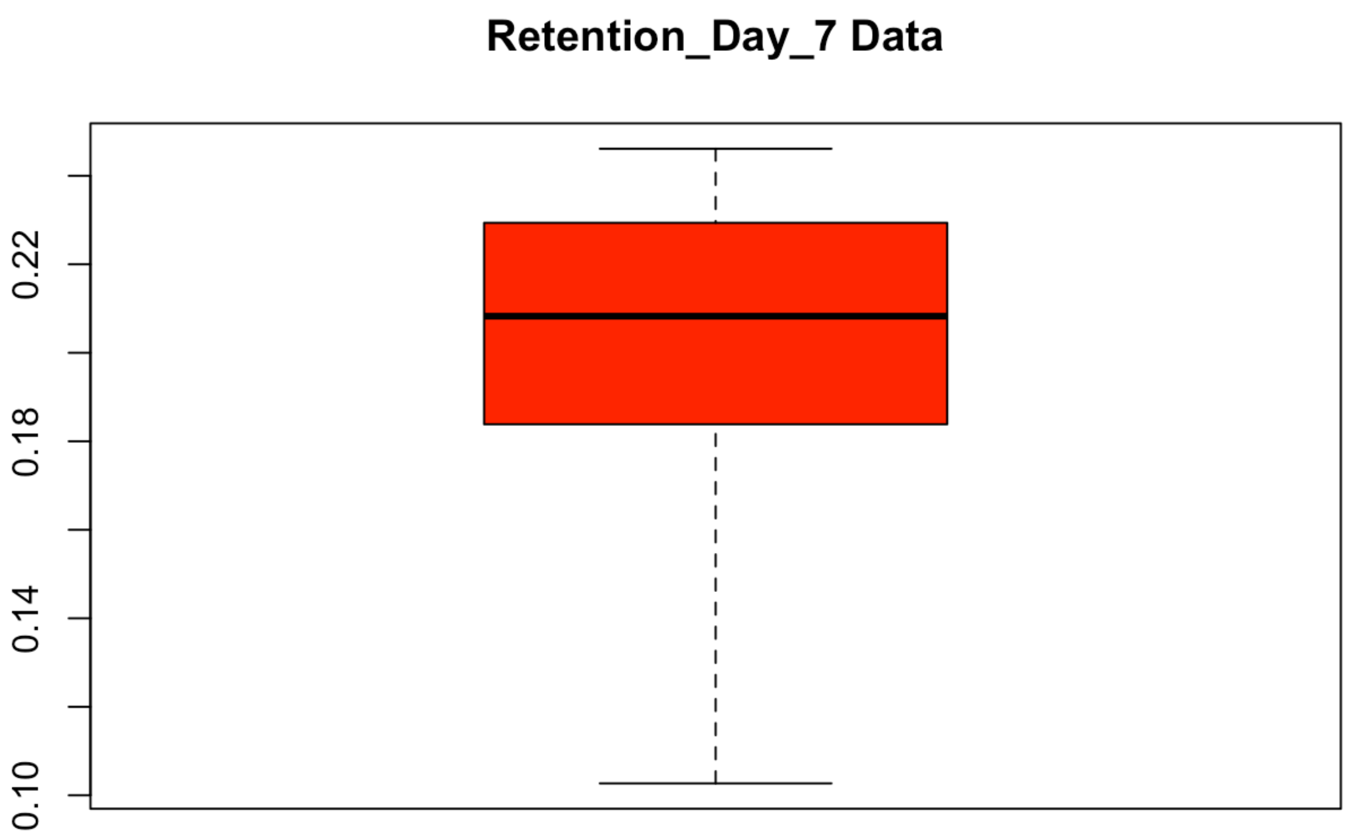
Instead, we will use `adjbox` for Skew Distributions

It's boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

Have an outlier detection with adjusted boxplot

```
require(robustbase)
adjbox(df$Retention_Day_7,col="red",main="Retention_Day_7 Data")
```




```
adjboxStats(df$Retention_Day_7)
```

```
## $stats
## [1] 0.1026962 0.1838462 0.2082698 0.2293687 0.2461117
##
## $n
## [1] 20
##
## $conf
## [1] 0.1921868 0.2243529
##
## $fence
## [1] 0.06803116 0.26312761
##
## $out
## numeric(0)
```

```
adjboxStats(df$Retention_Day_7)$out
```

```
## numeric(0)
```

```
length(unique(adjboxStats(df$Retention_Day_7)$out))
```

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

### Conclusion_6:

Find out 0 campaigns Retention_Day_7 lie outside Q1-1.5*exp(3M) and Q3+1.5*exp(3M)

## 7.ARPDAU_D7 performance evaluation

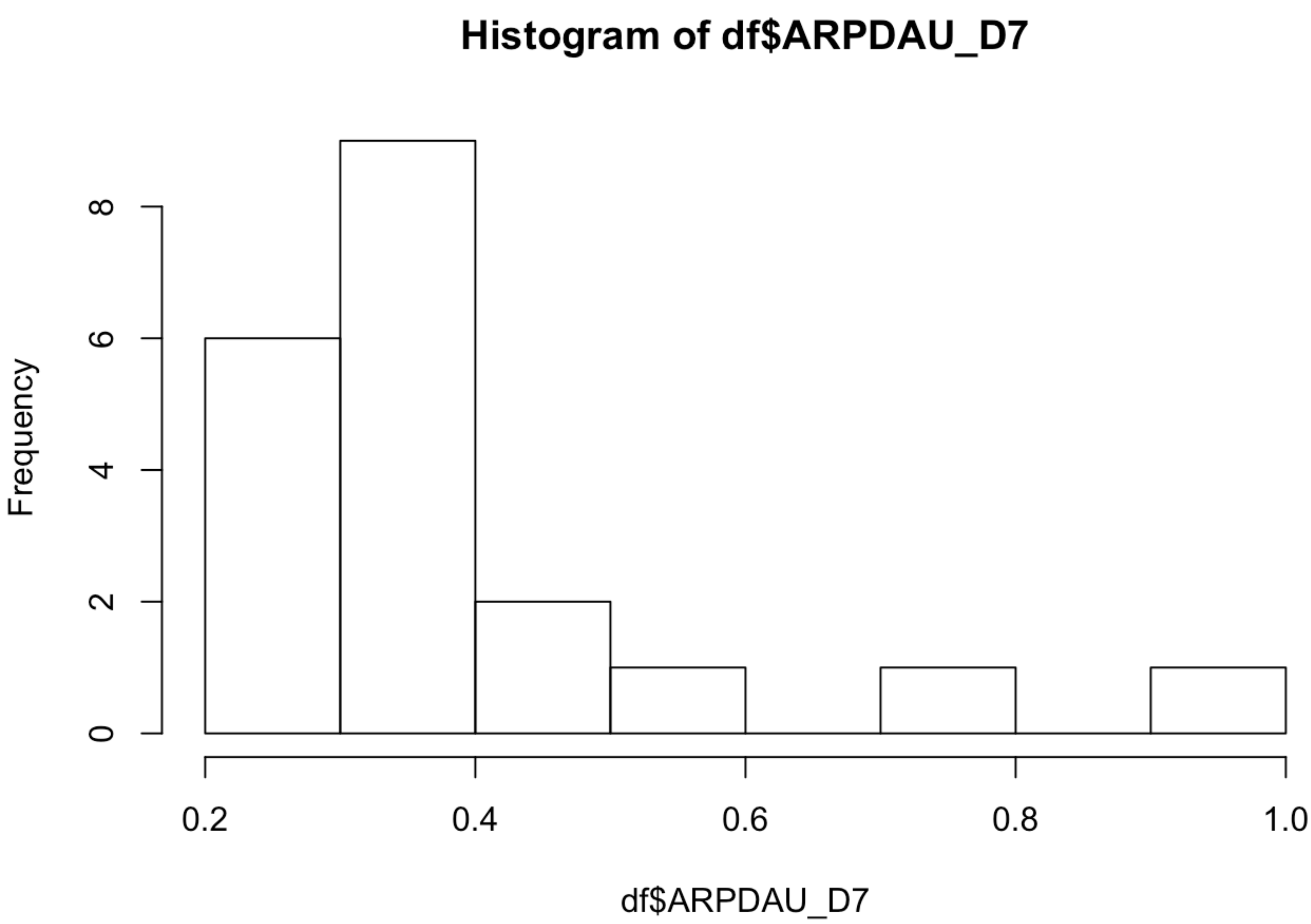
Have an overview with summary()

```
summary(df$ARPDAU_D7)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2024  0.2903  0.3394  0.3944  0.4115  0.9338
```

Using histogram to overview the distribution

```
hist(df$ARPDAU_D7)
```




From plot, it looks more like the Skewed right (positive) distribution

We apply the function skewness from the `e1071` package to compute the skewness coefficient of ARPDAU_D7.

```r
require(e1071)
skewness(df$ARPDAU_D7)
```

```
## [1] 1.624966
```

The skewness is 1.624966.

It indicates that the ARPDAU_D7 distribution is skewed towards the right.

The mean is on the right of the peak value.

Since this is a right skewed distribution, it doesn't recommend to use regular boxplot to detect outlier.

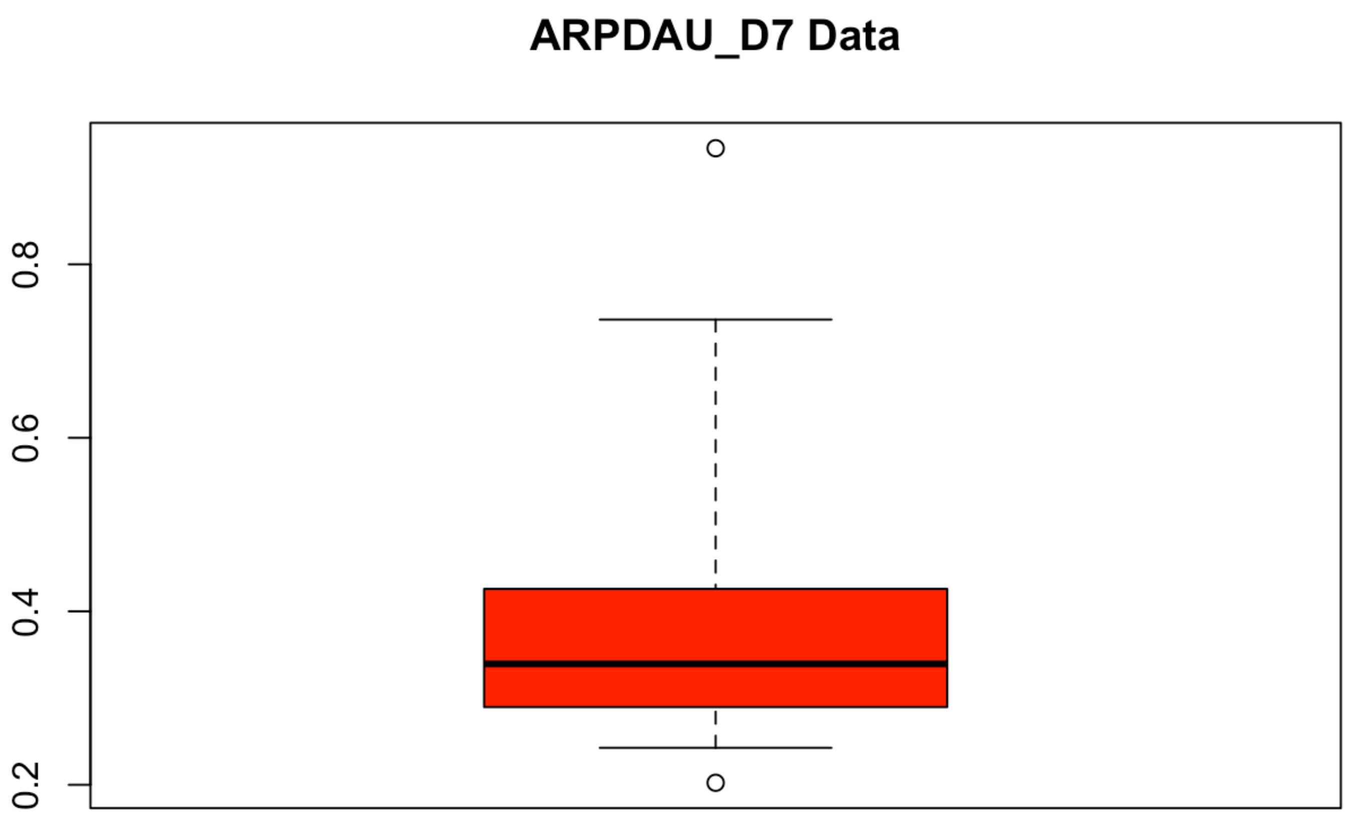
Instead, we will use `adjbox` for Skew Distributions

It's boxplot adjusted for skewed distributions as proposed in Hubert and Vandervieren (2004).

Further information, please refer https://rdrr.io/cran/robustbase/man/adjbox.html (https://rdrr.io/cran/robustbase/man/adjbox.html) and https://www.sciencedirect.com/science/article/pii/S0167947307004434 (https://www.sciencedirect.com/science/article/pii/S0167947307004434)

Have an outlier detection with adjusted boxplot

```r
require(robustbase)
adjbox(df$ARPDAU_D7,col="red",main="ARPDAU_D7 Data")
```



```r
adjboxStats(df$ARPDAU_D7)
```

```
## $stats
## [1] 0.2426627 0.2896294 0.3393850 0.4258307 0.7362935
##
## $n
## [1] 20
##
## $conf
## [1] 0.2912652 0.3875048
##
## $fence
## [1] 0.2282009 0.9289837
##
## $out
## [1] 0.2023675 0.9337911
```

```r
adjboxStats(df$ARPDAU_D7)$out
```

```
## [1] 0.2023675 0.9337911
```

```r
length(unique(adjboxStats(df$ARPDAU_D7)$out))
```

```
## [1] 2
```

```r
l8<-as.numeric(length(unique(adjboxStats(df$ARPDAU_D7)$out)))
```

Find out two campaigns ARPDAU_D7 lie outside Q1-1.5*exp(3M) and Q3+1.5*exp(3M)

Where M is an index of skewness of the uncontaminated part of the data.

Create function to filter those campaigns ID correspond values fall into outlier list.

```r
ARPD7O<-function(x){
  ARPDAU_D7_outlier<-as.numeric(adjboxStats(df$ARPDAU_D7)$out)
  df[df$ARPDAU_D7 == ARPDAU_D7_outlier[x],1]
}
ARPD7O(1)
```

```
## [1] 2
```

```r
sapply(1:l8,ARPD7O)
```

```
## [1]  2 10
```

## Conclusion_7:

Campaign 10 has ARPDAU_D7 higher Q3+1.5*exp(3M) which mean we should take them plus +10%.

Campaign 2 has ARPDAU_D7 lower Q1-1.5*exp(3M) which mean we should take them plus -10%.

### Overview of the conclusion

Based on Conclusion_1 to 7

Conclusion_1:ROI_180

Campaign 13,14,16 have ROI_180 higher Q3+0.25*exp(3M) which mean we should take it plus +20%

Campaign 3,7,9,18 have ROI_180 lower Q1-0.25*exp(3M) which mean we should take it plus -20%

Conclusion_2:ROAS

Campaign 13 have ROAS higher Q3+1*exp(3M) which mean we should take it plus +20%

Campaign 3,7,9 have ROAS lower Q1-1*exp(3M) which mean we should take it plus -20%

Conclusion_3:CPM_Spend

It’s not normal that CPM_Spend is so high(over 99)

And Campaign ID 14 and 15 have CPM_Spend over Q3+1.5*exp(3M), which mean we should take them plus -10%.

Conclusion_4:CTR

Since it’s not normal that CTR over 100% and too low

Campaign 14 and 15 have CTR over Q3+1.5*exp(3M) which mean we should take them plus -10%.

Campaign 1 and 10 have CTR lower Q1-1.5*exp(3M) which mean we should take them plus -10%.

Conclusion_5:CVR

Since it’s not normal that CVR too low

Campaign 14 has Conversion_rate lower Q1-1.5*exp(3M) which mean we should take them plus -10%.

Conclusion_6:Retention_Day_7

Find out 0 campaigns Retention_Day_7 lie outside Q1-1.5*exp(3M) and Q3+1.5*exp(3M)

Conclusion_7:ARPDAU_D7

Campaign 10 has ARPDAU_D7 higher Q3+1.5*exp(3M) which mean we should take them plus +10%.

Campaign 2 has ARPDAU_D7 lower Q1-1.5*exp(3M) which mean we should take them multiply -10%.

# Overall campaign score calculation:

Campaign ID 15:0-0.1 -0.1= -0.2

Campaign ID 14:0-0.1 -0.1 -0.1 =-0.3

Campaign ID 10:0-0.1+0.1=0

Campaign ID 1:0-0.1 = -0.1

Campaign ID 2:0-0.1 = -0.1

Campaign ID 13:0+0.2+0.2=0.4

Campaign ID 14:-0.3+0.2=-0.1

Campaign ID 16:0+0.2=0.2

Campaign ID 3:0-0.2-0.2=-0.4

Campaign ID 7:0-0.2-0.2=-0.4

Campaign ID 9:0-0.2-0.2=-0.4

Campaign ID 18:0-0.2=-0.2

**In the summary, we can label campaign ID = 13,16 as “Good” campaign due to positive Campaign score.**

**And campaign ID = 1,2,3,7,9,14,15,18 as “Bad” campaign due to negative Campaign score.**

**Others will label as “Average” campaign due to 0 Campaign score and fall into the standaed.**

**Further more discussion:**

1.We can create a weekly campaign score to evaluate the overall campaign performance.

2.It can be an automatic daily or weekly report if it’s necessary.

3.We can add other important metrics into campaign score depend on their correlation with ROI or business need.