batToken

Question 1:

library(readr)

## Warning: package 'readr' was built under R version 3.5.3

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(ggplot2)  
library(fitdistrplus)

## Warning: package 'fitdistrplus' was built under R version 3.5.3

## Loading required package: MASS

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

## Loading required package: survival

## Loading required package: npsurv

## Loading required package: lsei

bat <- read\_delim('networkbatTX.txt', delim = " ", col\_names = F)

## Parsed with column specification:  
## cols(  
## X1 = col\_double(),  
## X2 = col\_double(),  
## X3 = col\_double(),  
## X4 = col\_double()  
## )

print(bat)

## # A tibble: 327,854 x 4  
## X1 X2 X3 X4  
## <dbl> <dbl> <dbl> <dbl>  
## 1 79820 5 1524611245 2.67e21  
## 2 40002 351140 1524611310 9.45e20  
## 3 351141 238381 1524611390 2.41e20  
## 4 6 351143 1524611836 2.48e21  
## 5 351141 219077 1524611836 3.44e20  
## 6 5 351144 1524611875 6.24e20  
## 7 215733 351145 1524611886 5.00e19  
## 8 351146 351147 1524612030 2.50e20  
## 9 5 118 1524612101 3.51e22  
## 10 5 105551 1524612101 1.43e22  
## # ... with 327,844 more rows

names(bat) <- c('fromID', 'toID', 'unixTime', 'tokenAmount')  
print(bat)

## # A tibble: 327,854 x 4  
## fromID toID unixTime tokenAmount  
## <dbl> <dbl> <dbl> <dbl>  
## 1 79820 5 1524611245 2.67e21  
## 2 40002 351140 1524611310 9.45e20  
## 3 351141 238381 1524611390 2.41e20  
## 4 6 351143 1524611836 2.48e21  
## 5 351141 219077 1524611836 3.44e20  
## 6 5 351144 1524611875 6.24e20  
## 7 215733 351145 1524611886 5.00e19  
## 8 351146 351147 1524612030 2.50e20  
## 9 5 118 1524612101 3.51e22  
## 10 5 105551 1524612101 1.43e22  
## # ... with 327,844 more rows

decimals<-10^18  
supply<- 1.5\*10^9  
totalSupply<- decimals \* supply  
print(totalSupply)

## [1] 1.5e+27

filteredBat <- filter(bat,tokenAmount < totalSupply)  
print(filteredBat)

## # A tibble: 327,848 x 4  
## fromID toID unixTime tokenAmount  
## <dbl> <dbl> <dbl> <dbl>  
## 1 79820 5 1524611245 2.67e21  
## 2 40002 351140 1524611310 9.45e20  
## 3 351141 238381 1524611390 2.41e20  
## 4 6 351143 1524611836 2.48e21  
## 5 351141 219077 1524611836 3.44e20  
## 6 5 351144 1524611875 6.24e20  
## 7 215733 351145 1524611886 5.00e19  
## 8 351146 351147 1524612030 2.50e20  
## 9 5 118 1524612101 3.51e22  
## 10 5 105551 1524612101 1.43e22  
## # ... with 327,838 more rows

filteredBat <- filter(bat,fromID != toID)  
print(filteredBat)

## # A tibble: 317,133 x 4  
## fromID toID unixTime tokenAmount  
## <dbl> <dbl> <dbl> <dbl>  
## 1 79820 5 1524611245 2.67e21  
## 2 40002 351140 1524611310 9.45e20  
## 3 351141 238381 1524611390 2.41e20  
## 4 6 351143 1524611836 2.48e21  
## 5 351141 219077 1524611836 3.44e20  
## 6 5 351144 1524611875 6.24e20  
## 7 215733 351145 1524611886 5.00e19  
## 8 351146 351147 1524612030 2.50e20  
## 9 5 118 1524612101 3.51e22  
## 10 5 105551 1524612101 1.43e22  
## # ... with 317,123 more rows

NoOfOutliers <- count(bat)-count(filteredBat);  
print(NoOfOutliers)

## n  
## 1 10721

result <-filteredBat %>% count(fromID,toID, sort = FALSE)  
names(result) <- c('fromID', 'toID', 'Occurences')  
names(result)

## [1] "fromID" "toID" "Occurences"

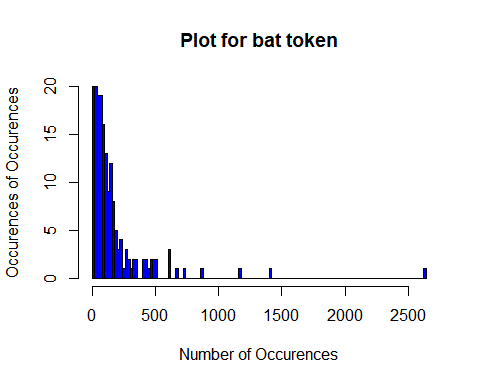
sum(result$Occurences)

## [1] 317133

result$Occ = 1  
result\_new <- aggregate(result$Occ, by=list(result$Occurences), FUN=sum)  
  
names(result\_new) <- c('Number','Occurences')  
head(result\_new)

## Number Occurences  
## 1 1 157220  
## 2 2 20990  
## 3 3 6493  
## 4 4 2757  
## 5 5 1469  
## 6 6 825

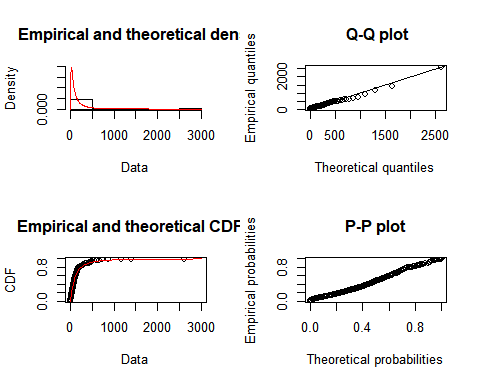
hist(result\_new$Number, breaks = 100, col = c("blue"), xlab = "Number of Occurences", ylab = "Occurences of Occurences",main = "Plot for bat token")



fit.exp.result <- fitdist(result\_new$Number, 'exp')  
fit.gamma.result <- fitdist(result\_new$Number, 'gamma',lower = c(0, 0), start = list(scale = 1, shape = 1))  
fit.geometric.result <- fitdist(result\_new$Number, 'geom')  
fit.log.result <- fitdist(result\_new$Number, 'logis')  
fit.lnorm.result <- fitdist(result\_new$Number, 'lnorm')  
fit.nbinom.result <- fitdist(result\_new$Number, 'nbinom')  
fit.norm.result <- fitdist(result\_new$Number, 'norm')  
fit.pois.result <- fitdist(result\_new$Number, 'pois')  
fit.unif.result <- fitdist(result\_new$Number, 'unif')  
fit.weibull.result <- fitdist(result\_new$Number, 'weibull')  
gofstat(list(fit.weibull.result, fit.gamma.result, fit.lnorm.result, fit.exp.result, fit.log.result, fit.pois.result))

## Goodness-of-fit statistics  
## 1-mle-weibull 2-mle-gamma 3-mle-lnorm  
## Kolmogorov-Smirnov statistic 0.09264762 0.1091445 0.05864597  
## Cramer-von Mises statistic 0.33488301 0.4598649 0.16653606  
## Anderson-Darling statistic 1.95215865 2.3737594 0.96295862  
## 4-mle-exp 5-mle-logis 6-mle-pois  
## Kolmogorov-Smirnov statistic 0.1181292 0.2127198 0.646231  
## Cramer-von Mises statistic 0.6465903 1.6148370 22.183829  
## Anderson-Darling statistic 3.0716912 11.2372824 Inf  
##   
## Goodness-of-fit criteria  
## 1-mle-weibull 2-mle-gamma 3-mle-lnorm  
## Akaike's Information Criterion 2193.151 2198.499 2182.973  
## Bayesian Information Criterion 2199.525 2204.873 2189.348  
## 4-mle-exp 5-mle-logis 6-mle-pois  
## Akaike's Information Criterion 2198.969 2383.641 40462.96  
## Bayesian Information Criterion 2202.156 2390.016 40466.15

plot(fit.lnorm.result)

 Question 2:

names(filteredBat) <- c('fromID', 'toID', 'TimeStamp', 'TokenAmount')  
filteredBat

## # A tibble: 317,133 x 4  
## fromID toID TimeStamp TokenAmount  
## <dbl> <dbl> <dbl> <dbl>  
## 1 79820 5 1524611245 2.67e21  
## 2 40002 351140 1524611310 9.45e20  
## 3 351141 238381 1524611390 2.41e20  
## 4 6 351143 1524611836 2.48e21  
## 5 351141 219077 1524611836 3.44e20  
## 6 5 351144 1524611875 6.24e20  
## 7 215733 351145 1524611886 5.00e19  
## 8 351146 351147 1524612030 2.50e20  
## 9 5 118 1524612101 3.51e22  
## 10 5 105551 1524612101 1.43e22  
## # ... with 317,123 more rows

filteredBat$TokenAmount<-(filteredBat$TokenAmount)/decimals  
Time<-as.Date(as.POSIXct(filteredBat$TimeStamp, origin="1970-01-01"))  
filteredBat$TimeStamp<-Time  
filteredBat

## # A tibble: 317,133 x 4  
## fromID toID TimeStamp TokenAmount  
## <dbl> <dbl> <date> <dbl>  
## 1 79820 5 2018-04-24 2675.  
## 2 40002 351140 2018-04-24 945.  
## 3 351141 238381 2018-04-24 241.  
## 4 6 351143 2018-04-24 2475.  
## 5 351141 219077 2018-04-24 344.  
## 6 5 351144 2018-04-24 624.  
## 7 215733 351145 2018-04-24 50   
## 8 351146 351147 2018-04-24 250   
## 9 5 118 2018-04-24 35134.  
## 10 5 105551 2018-04-24 14267.  
## # ... with 317,123 more rows

BatTokenData <- read\_delim('bat', delim = "\t", col\_names = T)

## Parsed with column specification:  
## cols(  
## Date = col\_character(),  
## Open = col\_double(),  
## High = col\_double(),  
## Low = col\_double(),  
## Close = col\_double(),  
## Volume = col\_number(),  
## `Market Cap` = col\_character()  
## )

BatTokenData

## # A tibble: 410 x 7  
## Date Open High Low Close Volume `Market Cap`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 07/15/2018 0.318 0.342 0.308 0.317 8434180 317,590,000   
## 2 07/14/2018 0.335 0.344 0.308 0.318 15608800 334,512,000   
## 3 07/13/2018 0.269 0.341 0.265 0.336 16786200 269,151,000   
## 4 07/12/2018 0.269 0.270 0.251 0.268 4768230 268,646,000   
## 5 07/11/2018 0.243 0.274 0.243 0.269 4486670 243,195,000   
## 6 07/10/2018 0.262 0.263 0.238 0.243 2751840 262,173,000   
## 7 07/09/2018 0.274 0.281 0.264 0.265 2762680 274,015,000   
## 8 07/08/2018 0.270 0.279 0.264 0.274 2346660 269,778,000   
## 9 07/07/2018 0.265 0.270 0.255 0.269 3189520 264,982,000   
## 10 07/06/2018 0.241 0.272 0.237 0.262 4473600 241,166,000   
## # ... with 400 more rows

names(BatTokenData)<-c("TimeStamp","Open","High", "Low", "Close","Volume","MarketCap")  
BatTokenData$TimeStamp<-as.Date(BatTokenData$TimeStamp,"%m/%d/%Y")  
BatTokenData$MarketCap <- as.numeric(gsub(",","",BatTokenData$MarketCap))

## Warning: NAs introduced by coercion

BatTokenData$MarketCap<-as.double(BatTokenData$MarketCap)  
BatTokenData

## # A tibble: 410 x 7  
## TimeStamp Open High Low Close Volume MarketCap  
## <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2018-07-15 0.318 0.342 0.308 0.317 8434180 317590000  
## 2 2018-07-14 0.335 0.344 0.308 0.318 15608800 334512000  
## 3 2018-07-13 0.269 0.341 0.265 0.336 16786200 269151000  
## 4 2018-07-12 0.269 0.270 0.251 0.268 4768230 268646000  
## 5 2018-07-11 0.243 0.274 0.243 0.269 4486670 243195000  
## 6 2018-07-10 0.262 0.263 0.238 0.243 2751840 262173000  
## 7 2018-07-09 0.274 0.281 0.264 0.265 2762680 274015000  
## 8 2018-07-08 0.270 0.279 0.264 0.274 2346660 269778000  
## 9 2018-07-07 0.265 0.270 0.255 0.269 3189520 264982000  
## 10 2018-07-06 0.241 0.272 0.237 0.262 4473600 241166000  
## # ... with 400 more rows

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

filteredBat <- filteredBat[order(-filteredBat$TokenAmount),]  
filteredBat

## # A tibble: 317,133 x 4  
## fromID toID TimeStamp TokenAmount  
## <dbl> <dbl> <date> <dbl>  
## 1 351440 363461 2018-03-29 1.16e59  
## 2 351440 363620 2018-03-30 1.16e59  
## 3 351440 363626 2018-03-30 1.16e59  
## 4 315614 363671 2018-03-30 1.16e59  
## 5 315614 363683 2018-03-30 1.16e59  
## 6 466763 351438 2017-05-31 5.00e 8  
## 7 215078 357055 2017-07-24 3.66e 8  
## 8 460722 460723 2017-06-21 3.66e 8  
## 9 460722 465482 2017-05-29 1.14e 8  
## 10 349691 324440 2018-01-09 3.20e 7  
## # ... with 317,123 more rows

top\_buyers<-head(filteredBat, 25)  
top\_buyers

## # A tibble: 25 x 4  
## fromID toID TimeStamp TokenAmount  
## <dbl> <dbl> <date> <dbl>  
## 1 351440 363461 2018-03-29 1.16e59  
## 2 351440 363620 2018-03-30 1.16e59  
## 3 351440 363626 2018-03-30 1.16e59  
## 4 315614 363671 2018-03-30 1.16e59  
## 5 315614 363683 2018-03-30 1.16e59  
## 6 466763 351438 2017-05-31 5.00e 8  
## 7 215078 357055 2017-07-24 3.66e 8  
## 8 460722 460723 2017-06-21 3.66e 8  
## 9 460722 465482 2017-05-29 1.14e 8  
## 10 349691 324440 2018-01-09 3.20e 7  
## # ... with 15 more rows

BatTokenData

## # A tibble: 410 x 7  
## TimeStamp Open High Low Close Volume MarketCap  
## <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2018-07-15 0.318 0.342 0.308 0.317 8434180 317590000  
## 2 2018-07-14 0.335 0.344 0.308 0.318 15608800 334512000  
## 3 2018-07-13 0.269 0.341 0.265 0.336 16786200 269151000  
## 4 2018-07-12 0.269 0.270 0.251 0.268 4768230 268646000  
## 5 2018-07-11 0.243 0.274 0.243 0.269 4486670 243195000  
## 6 2018-07-10 0.262 0.263 0.238 0.243 2751840 262173000  
## 7 2018-07-09 0.274 0.281 0.264 0.265 2762680 274015000  
## 8 2018-07-08 0.270 0.279 0.264 0.274 2346660 269778000  
## 9 2018-07-07 0.265 0.270 0.255 0.269 3189520 264982000  
## 10 2018-07-06 0.241 0.272 0.237 0.262 4473600 241166000  
## # ... with 400 more rows

TopBuyerData<-join(top\_buyers, BatTokenData)

## Joining by: TimeStamp

TopBuyerData

## fromID toID TimeStamp TokenAmount Open High Low  
## 1 351440 363461 2018-03-29 1.157921e+59 0.223284 0.224641 0.186172  
## 2 351440 363620 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 3 351440 363626 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 4 315614 363671 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 5 315614 363683 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 6 466763 351438 2017-05-31 5.000000e+08 NA NA NA  
## 7 215078 357055 2017-07-24 3.663500e+08 0.127953 0.168806 0.121486  
## 8 460722 460723 2017-06-21 3.663500e+08 0.190874 0.196532 0.183393  
## 9 460722 465482 2017-05-29 1.136500e+08 NA NA NA  
## 10 349691 324440 2018-01-09 3.200000e+07 0.853708 0.980702 0.816096  
## 11 378810 378811 2018-02-14 2.208355e+07 0.347633 0.402257 0.346107  
## 12 49 242976 2017-06-01 2.000000e+07 0.168230 0.175339 0.146197  
## 13 359616 462989 2017-06-03 1.920000e+07 0.156534 0.185107 0.154692  
## 14 447266 359931 2017-08-28 1.907440e+07 0.210025 0.236254 0.204585  
## 15 395185 411414 2017-06-01 1.625918e+07 0.168230 0.175339 0.146197  
## 16 255914 49 2017-05-31 1.600000e+07 NA NA NA  
## 17 395185 458944 2017-06-03 1.600000e+07 0.156534 0.185107 0.154692  
## 18 459679 359080 2017-06-12 1.561952e+07 0.182321 0.212659 0.152264  
## 19 92515 255914 2017-05-31 1.500000e+07 NA NA NA  
## 20 359931 403875 2018-01-11 1.400000e+07 0.800209 0.800209 0.620202  
## 21 411414 403686 2017-06-29 1.397752e+07 0.174455 0.174849 0.152339  
## 22 354421 251107 2017-06-24 1.384889e+07 0.201123 0.203272 0.176831  
## 23 314743 361104 2018-03-13 1.308356e+07 0.260966 0.267216 0.245337  
## 24 462757 248910 2017-06-07 1.280000e+07 0.259213 0.280704 0.228324  
## 25 462729 462757 2017-06-01 1.280000e+07 0.168230 0.175339 0.146197  
## Close Volume MarketCap  
## 1 0.188864 4810120 223284000  
## 2 0.185911 4906680 188415000  
## 3 0.185911 4906680 188415000  
## 4 0.185911 4906680 188415000  
## 5 0.185911 4906680 188415000  
## 6 NA NA NA  
## 7 0.144172 4271920 127953000  
## 8 0.187384 3436530 190874000  
## 9 NA NA NA  
## 10 0.891962 86957600 853708000  
## 11 0.395896 4580830 347633000  
## 12 0.171036 4373940 NA  
## 13 0.166049 4738010 156534000  
## 14 0.227662 3780050 210025000  
## 15 0.171036 4373940 NA  
## 16 NA NA NA  
## 17 0.166049 4738010 156534000  
## 18 0.182798 7175260 182321000  
## 19 NA NA NA  
## 20 0.653207 34119300 800209000  
## 21 0.154906 1880280 174455000  
## 22 0.186039 2256800 201123000  
## 23 0.246668 4214560 260966000  
## 24 0.238590 11149700 259213000  
## 25 0.171036 4373940 NA

#TopBuyerData$percentage<-(TopBuyerData$TokenAmount/TopBuyerData$MarketCap)\*100  
#TopBuyerData<- subset(TopBuyerData,percentage<100)  
TopUniqueBuyers<-unique(TopBuyerData)  
TopUniqueBuyers

## fromID toID TimeStamp TokenAmount Open High Low  
## 1 351440 363461 2018-03-29 1.157921e+59 0.223284 0.224641 0.186172  
## 2 351440 363620 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 3 351440 363626 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 4 315614 363671 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 5 315614 363683 2018-03-30 1.157921e+59 0.188415 0.190857 0.176944  
## 6 466763 351438 2017-05-31 5.000000e+08 NA NA NA  
## 7 215078 357055 2017-07-24 3.663500e+08 0.127953 0.168806 0.121486  
## 8 460722 460723 2017-06-21 3.663500e+08 0.190874 0.196532 0.183393  
## 9 460722 465482 2017-05-29 1.136500e+08 NA NA NA  
## 10 349691 324440 2018-01-09 3.200000e+07 0.853708 0.980702 0.816096  
## 11 378810 378811 2018-02-14 2.208355e+07 0.347633 0.402257 0.346107  
## 12 49 242976 2017-06-01 2.000000e+07 0.168230 0.175339 0.146197  
## 13 359616 462989 2017-06-03 1.920000e+07 0.156534 0.185107 0.154692  
## 14 447266 359931 2017-08-28 1.907440e+07 0.210025 0.236254 0.204585  
## 15 395185 411414 2017-06-01 1.625918e+07 0.168230 0.175339 0.146197  
## 16 255914 49 2017-05-31 1.600000e+07 NA NA NA  
## 17 395185 458944 2017-06-03 1.600000e+07 0.156534 0.185107 0.154692  
## 18 459679 359080 2017-06-12 1.561952e+07 0.182321 0.212659 0.152264  
## 19 92515 255914 2017-05-31 1.500000e+07 NA NA NA  
## 20 359931 403875 2018-01-11 1.400000e+07 0.800209 0.800209 0.620202  
## 21 411414 403686 2017-06-29 1.397752e+07 0.174455 0.174849 0.152339  
## 22 354421 251107 2017-06-24 1.384889e+07 0.201123 0.203272 0.176831  
## 23 314743 361104 2018-03-13 1.308356e+07 0.260966 0.267216 0.245337  
## 24 462757 248910 2017-06-07 1.280000e+07 0.259213 0.280704 0.228324  
## 25 462729 462757 2017-06-01 1.280000e+07 0.168230 0.175339 0.146197  
## Close Volume MarketCap  
## 1 0.188864 4810120 223284000  
## 2 0.185911 4906680 188415000  
## 3 0.185911 4906680 188415000  
## 4 0.185911 4906680 188415000  
## 5 0.185911 4906680 188415000  
## 6 NA NA NA  
## 7 0.144172 4271920 127953000  
## 8 0.187384 3436530 190874000  
## 9 NA NA NA  
## 10 0.891962 86957600 853708000  
## 11 0.395896 4580830 347633000  
## 12 0.171036 4373940 NA  
## 13 0.166049 4738010 156534000  
## 14 0.227662 3780050 210025000  
## 15 0.171036 4373940 NA  
## 16 NA NA NA  
## 17 0.166049 4738010 156534000  
## 18 0.182798 7175260 182321000  
## 19 NA NA NA  
## 20 0.653207 34119300 800209000  
## 21 0.154906 1880280 174455000  
## 22 0.186039 2256800 201123000  
## 23 0.246668 4214560 260966000  
## 24 0.238590 11149700 259213000  
## 25 0.171036 4373940 NA

message("K=",nrow(TopUniqueBuyers))

## K=25

cor.test(TopUniqueBuyers$Open,TopUniqueBuyers$Volume)

##   
## Pearson's product-moment correlation  
##   
## data: TopUniqueBuyers$Open and TopUniqueBuyers$Volume  
## t = 8.4894, df = 19, p-value = 6.862e-08  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.7433774 0.9546628  
## sample estimates:  
## cor   
## 0.8895894

cor.test(TopUniqueBuyers$Open,TopUniqueBuyers$MarketCap)

##   
## Pearson's product-moment correlation  
##   
## data: TopUniqueBuyers$Open and TopUniqueBuyers$MarketCap  
## t = Inf, df = 16, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 1 1  
## sample estimates:  
## cor   
## 1

cor.test(TopUniqueBuyers$Open,TopUniqueBuyers$TokenAmount)

##   
## Pearson's product-moment correlation  
##   
## data: TopUniqueBuyers$Open and TopUniqueBuyers$TokenAmount  
## t = -0.80535, df = 19, p-value = 0.4306  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.5687625 0.2712792  
## sample estimates:  
## cor   
## -0.1816842

linearModel <- lm(formula=Open ~ Volume+MarketCap+TokenAmount, data=TopUniqueBuyers)   
linearModel

##   
## Call:  
## lm(formula = Open ~ Volume + MarketCap + TokenAmount, data = TopUniqueBuyers)  
##   
## Coefficients:  
## (Intercept) Volume MarketCap TokenAmount   
## -3.774e-17 -1.328e-24 1.000e-09 -1.271e-76

summary(linearModel)

## Warning in summary.lm(linearModel): essentially perfect fit: summary may be  
## unreliable

##   
## Call:  
## lm(formula = Open ~ Volume + MarketCap + TokenAmount, data = TopUniqueBuyers)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.588e-16 -1.009e-17 3.086e-18 1.765e-17 9.525e-17   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -3.774e-17 3.000e-17 -1.258e+00 0.229   
## Volume -1.328e-24 1.380e-24 -9.620e-01 0.352   
## MarketCap 1.000e-09 1.360e-25 7.355e+15 <2e-16 \*\*\*  
## TokenAmount -1.271e-76 2.472e-76 -5.140e-01 0.615   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.281e-17 on 14 degrees of freedom  
## (7 observations deleted due to missingness)  
## Multiple R-squared: 1, Adjusted R-squared: 1   
## F-statistic: 8.773e+31 on 3 and 14 DF, p-value: < 2.2e-16

plot(linearModel)

