

Storj Token

```
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

token <- read_delim('networkstorjTX.txt', delim = " ", col_names = F)

## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   X2 = col_double(),
##   X3 = col_double(),
##   X4 = col_double()
## )

print(token)
```

```
## # A tibble: 406,131 x 4
##       X1       X2       X3       X4
##   <dbl> <dbl> <dbl> <dbl>
## 1 6103619 5607448 1524611290 2307690000000
## 2      17  930403 1524611565  987000000000
## 3  930403      5 1524612137  987000000000
## 4 148332 1757147 1524612282  80312639339
## 5      17     573 1524612421 212877929273
## 6     573      5 1524612737 212877929273
## 7      17   87160 1524613395  997000000000
## 8       5   39994 1524613542  203413000000
## 9      82     46 1524614390 2061064700000
## 10     171     573 1524614582 2026000000000
## # ... with 406,121 more rows

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

## # A tibble: 406,131 x 4
##   fromID   toID   unixTime tokenAmount
##   <dbl> <dbl> <dbl> <dbl>
## 1 6103619 5607448 1524611290 2307690000000
## 2      17  930403 1524611565  987000000000
## 3  930403      5 1524612137  987000000000
## 4 148332 1757147 1524612282  80312639339
## 5      17     573 1524612421 212877929273
## 6     573      5 1524612737 212877929273
## 7      17   87160 1524613395  997000000000
## 8       5   39994 1524613542  203413000000
## 9      82     46 1524614390 2061064700000
## 10     171     573 1524614582 2026000000000
## # ... with 406,121 more rows

decimals<-10^8
supply<- 424999998.00001337
totalSupply<- decimals * supply
print(totalSupply)

## [1] 4.25e+16

filteredtoken <- filter(token,tokenAmount < totalSupply)
print(filteredtoken)

## # A tibble: 406,078 x 4
##   fromID   toID   unixTime tokenAmount
##   <dbl> <dbl> <dbl> <dbl>
## 1 6103619 5607448 1524611290 2307690000000
## 2      17  930403 1524611565  987000000000
## 3  930403      5 1524612137  987000000000
## 4 148332 1757147 1524612282  80312639339
## 5      17     573 1524612421 212877929273
```

```

## 6      573      5 1524612737 212877929273
## 7      17     87160 1524613395 99700000000
## 8      5     39994 1524613542 20341300000
## 9      82      46 1524614390 206106470000
## 10     171     573 1524614582 202600000000
## # ... with 406,068 more rows

filteredtoken <- filter(token,fromID != toID)
print(filteredtoken)

## # A tibble: 402,453 x 4
##   fromID   toID   unixTime tokenAmount
##   <dbl>   <dbl>   <dbl>     <dbl>
## 1 6103619 5607448 1524611290 230769000000
## 2      17  930403 1524611565  98700000000
## 3  930403      5 1524612137  98700000000
## 4 148332 1757147 1524612282  80312639339
## 5      17      573 1524612421 212877929273
## 6      573      5 1524612737 212877929273
## 7      17     87160 1524613395 99700000000
## 8      5     39994 1524613542 20341300000
## 9      82      46 1524614390 206106470000
## 10     171     573 1524614582 202600000000
## # ... with 402,443 more rows

NoOfOutliers <- count(token)-count(filteredtoken);
print(NoOfOutliers)

##      n
## 1 3678

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

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

sum(result$Occurences)

## [1] 402453

result$Occ = 1
result_new <- aggregate(result$Occ, by=list(result$Occurences), FUN=sum)

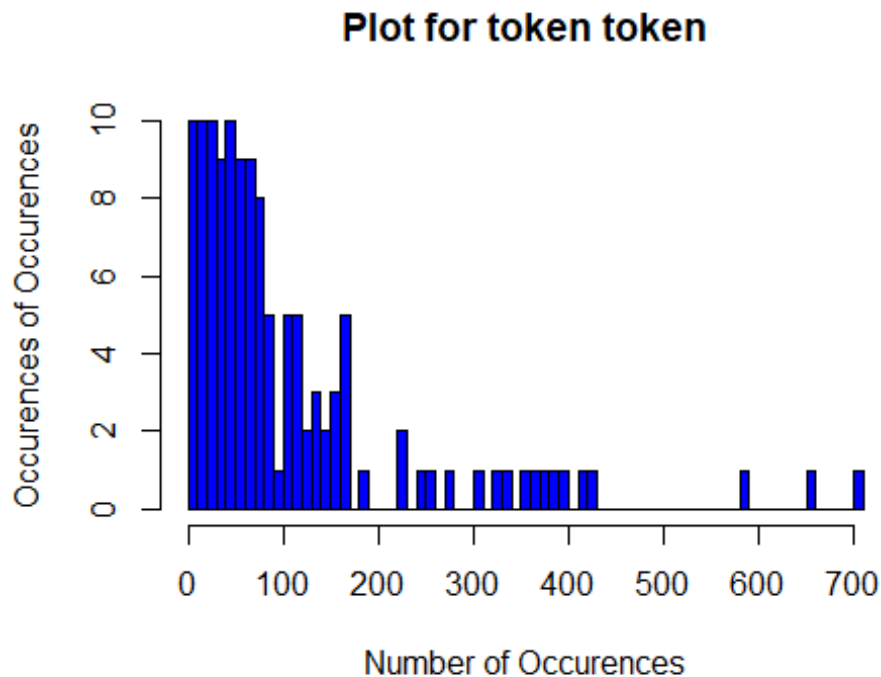
names(result_new) <- c('Number','Occurences')
head(result_new)

##   Number Occurences
## 1      1     198834
## 2      2     44017
## 3      3     14089
## 4      4       4207

```

```
## 5      5      2216
## 6      6      1396
```

```
hist(result_new$Number, breaks = 100, col = c("blue"), xlab = "Number of
Occurences", ylab = "Occurences of Occurences",main = "Plot for token 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))
```

```
## Goodness-of-fit statistics
##
## 1-mle-weibull 2-mle-gamma 3-mle-lnorm
## Kolmogorov-Smirnov statistic 0.08070105 0.09181971 0.06026925
## Cramer-von Mises statistic 0.15736680 0.19431060 0.10223441
## Anderson-Darling statistic 0.93148358 1.07418524 0.63858344
##
## 4-mle-exp 5-mle-logis
```

```
## Kolmogorov-Smirnov statistic 0.09605241 0.1917018
## Cramer-von Mises statistic 0.21412425 0.8888001
## Anderson-Darling statistic 1.15387288 7.1571758
##
## Goodness-of-fit criteria
##
## 1-mle-weibull 2-mle-gamma 3-mle-lnorm
## Akaike's Information Criterion 1429.939 1430.627 1430.977
## Bayesian Information Criterion 1435.596 1436.283 1436.633
##
## 4-mle-exp 5-mle-logis
## Akaike's Information Criterion 1428.733 1540.632
## Bayesian Information Criterion 1431.561 1546.289
```

```
plot(fit.lnorm.result)
```

