BatToken

```
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
```

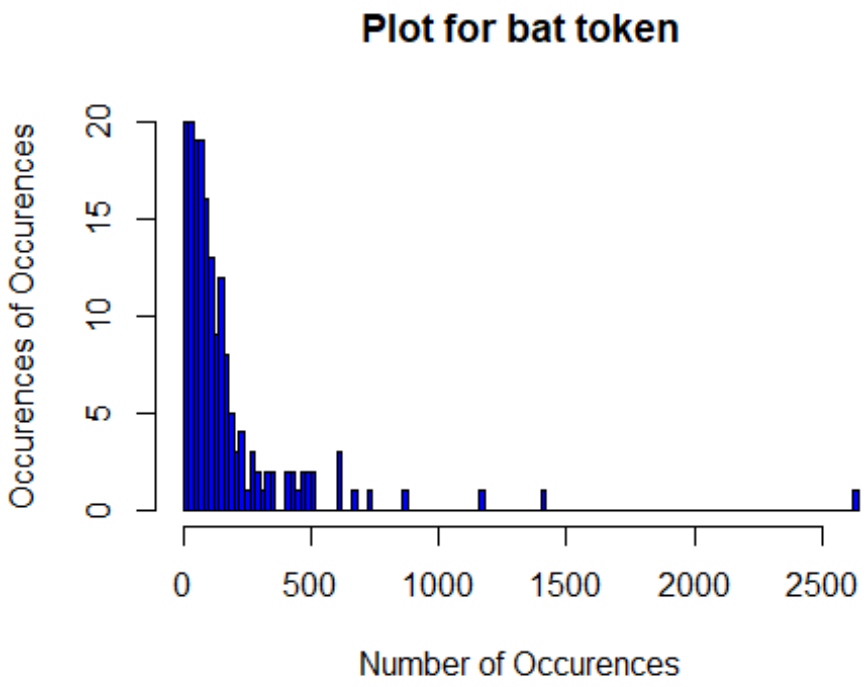
```
#names(Occ) <- c('Occurences','No')
result$Occ = 1
result_new <- aggregate(result$Occ, by=list(result$Occurences), FUN=sum)

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))

## 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
## Kolmogorov-Smirnov statistic 0.1181292   0.2127198
## Cramer-von Mises statistic   0.6465903   1.6148370
## Anderson-Darling statistic   3.0716912  11.2372824
##
## 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
## Akaike's Information Criterion  2198.969    2383.641
## Bayesian Information Criterion  2202.156    2390.016
```

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