Tenxpay 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('networktenxpayTX.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: 329,737 x 4
##         X1      X2         X3      X4
##      <dbl>   <dbl>      <dbl>   <dbl>
##  1     560    1452 1524611450 1.73e20
##  2 2011173 2011174 1524611865 4.56e20
##  3   75989 1822217 1524612292 5.80e20
##  4   40002 6382858 1524612655 4.48e20
##  5      17 2029263 1524612851 5.00e21
##  6  222770 4848204 1524612957 3.28e20
##  7      17    1148 1524613473 4.81e21
##  8   76011   76012 1524613896 5.20e19
##  9 6382859  104531 1524614072 6.08e19
## 10     187 3169275 1524614361 5.55e19
## # ... with 329,727 more rows
```

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

```
## # A tibble: 329,737 x 4
##     fromID    toID   unixTime tokenAmount
##      <dbl>   <dbl>      <dbl>       <dbl>
##  1     560    1452 1524611450     1.73e20
##  2 2011173 2011174 1524611865     4.56e20
##  3   75989 1822217 1524612292     5.80e20
##  4   40002 6382858 1524612655     4.48e20
##  5      17 2029263 1524612851     5.00e21
##  6  222770 4848204 1524612957     3.28e20
##  7      17    1148 1524613473     4.81e21
##  8   76011   76012 1524613896     5.20e19
##  9 6382859  104531 1524614072     6.08e19
## 10     187 3169275 1524614361     5.55e19
## # ... with 329,727 more rows
```

```
decimals<-10^18
supply<- 205218255.948577763364408207
totalSupply<- decimals * supply
print(totalSupply)
```

```
## [1] 2.052183e+26
```

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

```
## # A tibble: 329,736 x 4
##     fromID    toID   unixTime tokenAmount
##      <dbl>   <dbl>      <dbl>       <dbl>
##  1     560    1452 1524611450     1.73e20
##  2 2011173 2011174 1524611865     4.56e20
##  3   75989 1822217 1524612292     5.80e20
##  4   40002 6382858 1524612655     4.48e20
##  5      17 2029263 1524612851     5.00e21
##  6  222770 4848204 1524612957     3.28e20
##  7      17    1148 1524613473     4.81e21
##  8   76011   76012 1524613896     5.20e19
##  9 6382859  104531 1524614072     6.08e19
## 10     187 3169275 1524614361     5.55e19
## # ... with 329,726 more rows
```

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

```
## # A tibble: 319,656 x 4
##     fromID    toID   unixTime tokenAmount
##      <dbl>   <dbl>      <dbl>       <dbl>
##  1     560    1452 1524611450     1.73e20
##  2 2011173 2011174 1524611865     4.56e20
##  3   75989 1822217 1524612292     5.80e20
##  4   40002 6382858 1524612655     4.48e20
##  5      17 2029263 1524612851     5.00e21
##  6  222770 4848204 1524612957     3.28e20
##  7      17    1148 1524613473     4.81e21
##  8   76011   76012 1524613896     5.20e19
##  9 6382859  104531 1524614072     6.08e19
## 10     187 3169275 1524614361     5.55e19
## # ... with 319,646 more rows
```

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

```
##       n
## 1 10081
```

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

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

```r
sum(result$Occurences)
```

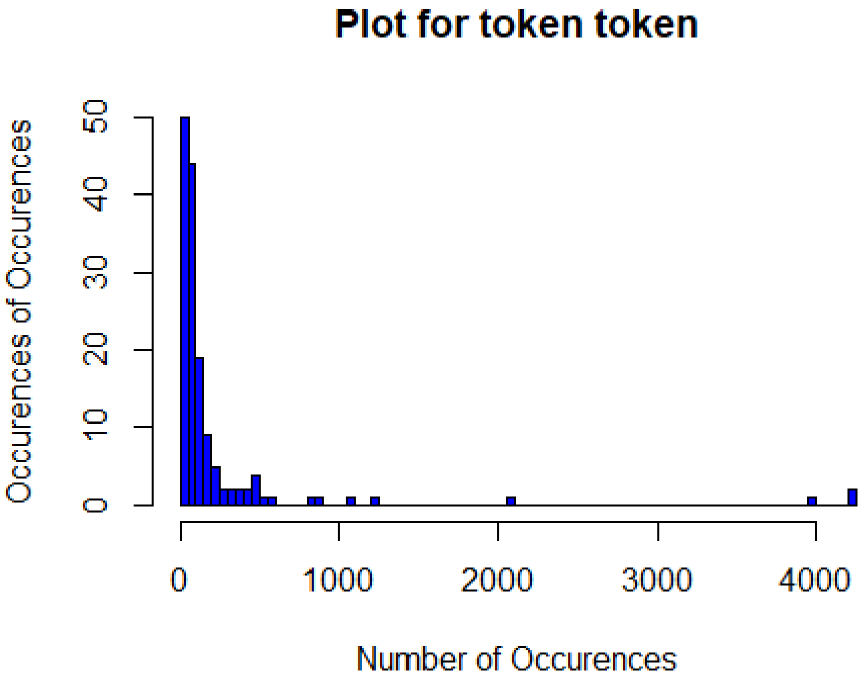
```
## [1] 319656
```

```r
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     131029
## 2      2      27246
## 3      3       8532
## 4      4       3875
## 5      5       2066
## 6      6       1258

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

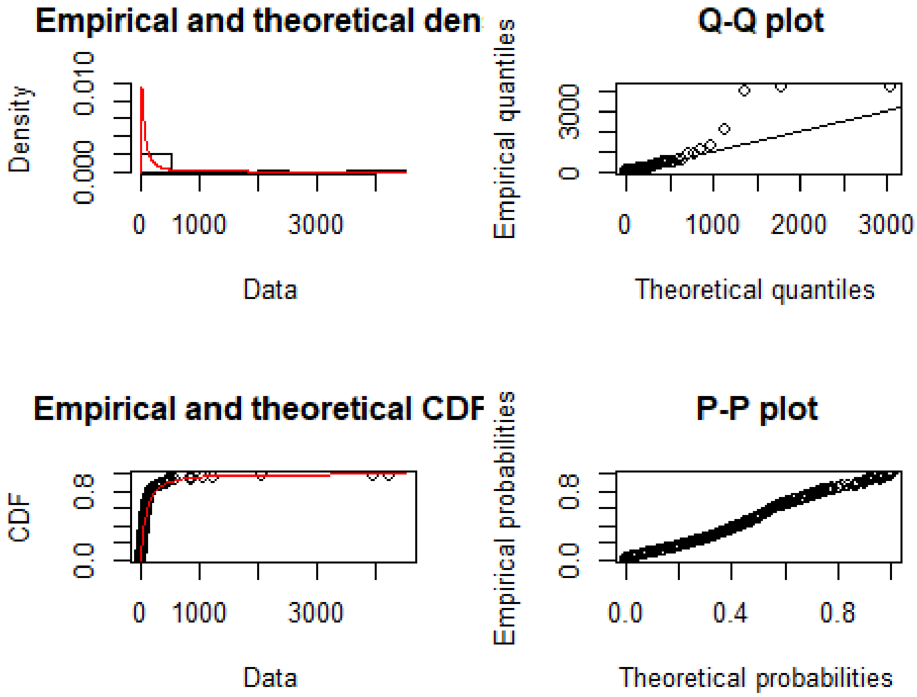


```
fit.exp.result <- fitdist(result_new$Number, 'exp')
fit.gamma.result <- fitdist(result_new$Number, 'gamma',lower = c(0, 0), start
= list(scale = 1, shape = 1))
fit.geometric.result <- fitdist(result_new$Number, 'geom')
fit.log.result <- fitdist(result_new$Number, 'logis')
fit.lnorm.result <- fitdist(result_new$Number, 'lnorm')
fit.nbinom.result <- fitdist(result_new$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     0.1361148   0.1980425  0.07326387
## Cramer-von Mises statistic       0.9125615   1.7294835  0.23198072
## Anderson-Darling statistic       5.3549628   8.9203993  1.29183124
##                               4-mle-exp 5-mle-logis
## Kolmogorov-Smirnov statistic  0.2872441   0.3147257
## Cramer-von Mises statistic    4.6133664   3.5188683
## Anderson-Darling statistic   22.0856395  20.1473062
##
## Goodness-of-fit criteria
##                                 1-mle-weibull 2-mle-gamma 3-mle-lnorm
## Akaike's Information Criterion      1850.388    1879.366    1809.956
## Bayesian Information Criterion      1856.396    1885.374    1815.964
##                                 4-mle-exp 5-mle-logis
## Akaike's Information Criterion  1916.776    2160.562
## Bayesian Information Criterion  1919.780    2166.570

plot(fit.lnorm.result)
```


Empirical and theoretical den
Density
Data
Q-Q plot
Empirical quantiles
Theoretical quantiles
Empirical and theoretical CDF
CDF
Data
P-P plot
Empirical probabilities
Theoretical probabilities