DAR Defi Team Assignment 2 (Fall 2021)

DeFi Reserve Coins

Duke Kwon

09/08/2021

Prepare Transaction Data and Explore

We begin by loading our prepared AAVE transaction data into a dataframe. The dataset has over 400,000 rows, and 27 columns.

We are directly loading the dataframe from an Rds archive instead of a CSV file to conserve space.

```r
#load Rds (binary version of csv file) into dataframe
# Assumes this notebook is in: ~/IDEA-Blockchain/DefiResearch/StudentNotebooks/Assignment02
df<-read_rds('../../Data/transactions.Rds')

# Let's take a quick look at the first few observation
head(df)
```

```
##       amount borrowRate borrowRateMode   onBehalfOf         pool reserve
## 1   41501.63   6.274937       Variable 8.502518e+47 1.034668e+48     DAI
## 2 7000000.00   2.589628       Variable 4.635974e+47 1.034668e+48    USDT
## 3   15000.00   8.802541       Variable 3.735263e+47 1.034668e+48    USDC
## 4    8193.19  48.747052         Stable 6.896232e+47 1.034668e+48    USDC
## 5   11000.00   3.225055       Variable 1.089455e+48 1.034668e+48    USDT
## 6   40000.00   5.739208       Variable 2.178337e+47 1.034668e+48    USDT
##    timestamp         user   type reservePriceETH reservePriceUSD  amountUSD
## 1 1621340435 8.502518e+47 borrow    2.852900e+14       0.9948044   41286.00
## 2 1622477822 4.635974e+47 borrow    3.812835e+14       1.0000000 7000000.00
## 3 1619775984 3.735263e+47 borrow    3.611000e+14       1.0043389   15065.08
## 4 1615481632 6.896232e+47 borrow    5.562201e+14       0.9993909    8188.20
## 5 1626914745 1.089455e+48 borrow    4.971100e+14       1.0000000   11000.00
## 6 1620936688 2.178337e+47 borrow    2.725248e+14       1.0000000   40000.00
##   collateralAmount collateralReserve principalAmount principalReserve
## 1               NA                                NA
## 2               NA                                NA
## 3               NA                                NA
## 4               NA                                NA
## 5               NA                                NA
## 6               NA                                NA
##   reservePriceETHPrincipal reservePriceUSDPrincipal reservePriceETHCollateral
## 1                       NA                       NA                        NA
## 2                       NA                       NA                        NA
## 3                       NA                       NA                        NA
## 4                       NA                       NA                        NA
## 5                       NA                       NA                        NA
## 6                       NA                       NA                        NA
```

```
##   reservePriceUSDCollateral amountUSDPincipal amountUSDCollateral
## 1                        NA                NA                  NA
## 2                        NA                NA                  NA
## 3                        NA                NA                  NA
## 4                        NA                NA                  NA
## 5                        NA                NA                  NA
## 6                        NA                NA                  NA
##   borrowRateModeFrom borrowRateModeTo stableBorrowRate variableBorrowRate
## 1                                                   NA                 NA
## 2                                                   NA                 NA
## 3                                                   NA                 NA
## 4                                                   NA                 NA
## 5                                                   NA                 NA
## 6                                                   NA                 NA
```

Now look at the summaries to see the types, values, and missingness (NA's) of the data.

```r
summary(df)
```

```
##      amount            borrowRate       borrowRateMode     onBehalfOf
##  Min.   :        0   Min.   :    0.0          :386542   Min.   :2.578e+33
##  1st Qu.:       24   1st Qu.:    3.3   Stable  : 18408   1st Qu.:4.174e+47
##  Median :     1427   Median :    3.9   Variable: 76569   Median :7.522e+47
##  Mean   :   191103   Mean   :    9.5                     Mean   :7.592e+47
##  3rd Qu.:    24382   3rd Qu.:   10.8                     3rd Qu.:1.168e+48
##  Max.   :600000000   Max.   :10002.0                     Max.   :1.461e+48
##  NA's   :7289        NA's   :386542                      NA's   :7289
##       pool              reserve         timestamp              user
##  Min.   :9.862e+47   USDC   :105937   Min.   :1.607e+09   Min.   :2.578e+33
##  1st Qu.:1.035e+48   WETH   :105279   1st Qu.:1.615e+09   1st Qu.:4.199e+47
##  Median :1.035e+48   USDT   : 58266   Median :1.621e+09   Median :8.697e+47
##  Mean   :1.034e+48   DAI    : 55211   Mean   :1.620e+09   Mean   :8.082e+47
##  3rd Qu.:1.035e+48   LINK   : 26404   3rd Qu.:1.624e+09   3rd Qu.:1.173e+48
##  Max.   :1.035e+48   WBTC   : 26344   Max.   :1.629e+09   Max.   :1.461e+48
##                      (Other):104078
##           type         reservePriceETH     reservePriceUSD
##  borrow     : 94977   Min.   :1.000e+00   Min.   :0.000e+00
##  deposit    :192006   1st Qu.:2.865e+14   1st Qu.:1.000e+00
##  liquidation:  6289   Median :4.652e+14   Median :1.000e+00
##  redeem     :126705   Mean   :3.458e+23   Mean   :6.774e+08
##  repay      : 60542   3rd Qu.:9.411e+14   3rd Qu.:1.000e+00
##  swap       :  1000   Max.   :1.647e+28   Max.   :4.252e+13
##                       NA's   :7289        NA's   :7289
##    amountUSD          collateralAmount  collateralReserve principalAmount
##  Min.   :        0   Min.   :      0          :475230    Min.   :      0
##  1st Qu.:       70   1st Qu.:      1   WETH   :  2665    1st Qu.:    962
##  Median :     5836   Median :     14   LINK   :  1312    Median :   4362
##  Mean   :   245851   Mean   :   5451   WBTC   :   686    Mean   :  66005
##  3rd Qu.:    49871   3rd Qu.:    250   AAVE   :   333    3rd Qu.:  21533
##  Max.   :754379487   Max.   :4638724   UNI    :   230    Max.   :4475668
##  NA's   :7289        NA's   :475230    (Other):  1063    NA's   :475230
##  principalReserve reservePriceETHPrincipal reservePriceUSDPrincipal
##         :475230   Min.   :1.000e+00        Min.   :    0.0
##  USDC   :  2142   1st Qu.:4.062e+14        1st Qu.:    1.0
##  USDT   :  1549   Median :4.682e+14        Median :    1.0
```

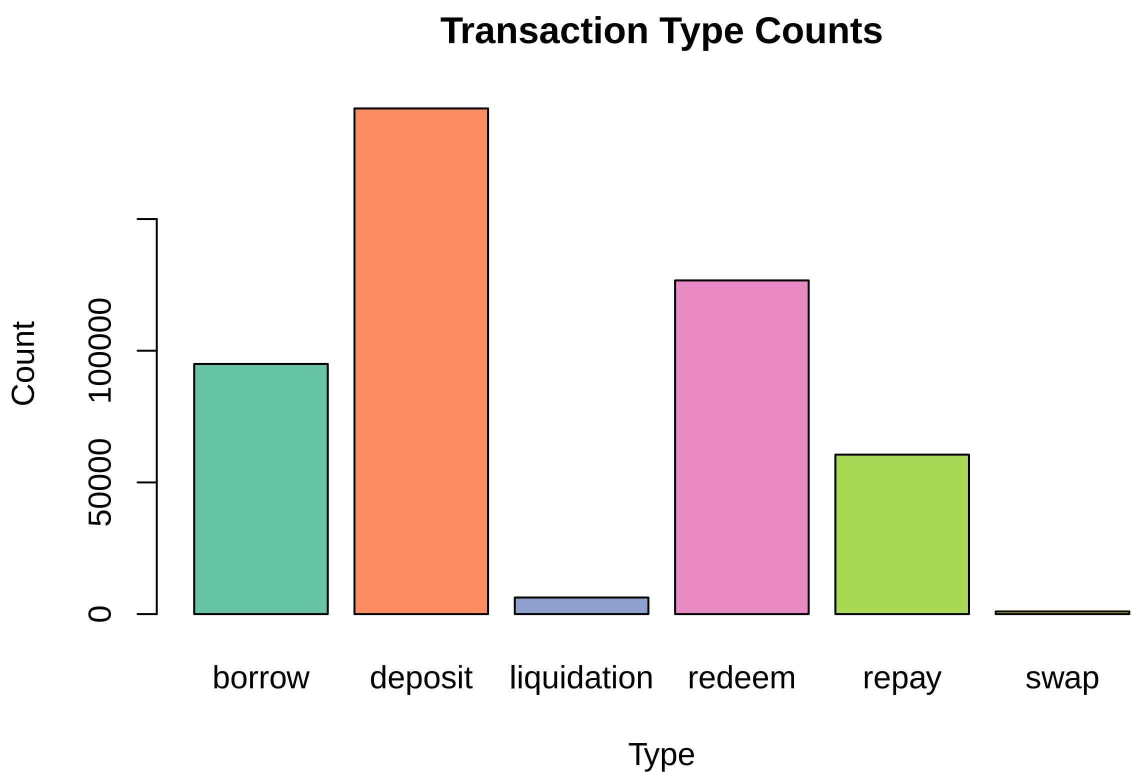
```
##  DAI     :  1459   Mean   :1.556e+17        Mean   :  295.6
##  GUSD    :   242   3rd Qu.:5.363e+14        3rd Qu.:    1.0
##  TUSD    :   175   Max.   :4.203e+19        Max.   :83819.1
##  (Other):   722   NA's   :475230           NA's   :475230
##  reservePriceETHCollateral reservePriceUSDCollateral amountUSDPincipal
##  Min.   :1.000e+00         Min.   :0.000e+00         Min.   :      0
##  1st Qu.:1.000e+00         1st Qu.:0.000e+00         1st Qu.:   1022
##  Median :5.110e+14         Median :1.000e+00         Median :   4481
##  Mean   :2.177e+21         Mean   :4.543e+06         Mean   :  67361
##  3rd Qu.:1.110e+16         3rd Qu.:2.600e+01         3rd Qu.:  22066
##  Max.   :9.116e+23         Max.   :2.509e+09         Max.   :4571839
##  NA's   :475230            NA's   :475230            NA's   :475230
##  amountUSDCollateral borrowRateModeFrom borrowRateModeTo  stableBorrowRate
##  Min.   :      0             :480519            :480519   Min.   :  0.0
##  1st Qu.:      0     Stable  :   471    Stable  :   529   1st Qu.:  9.0
##  Median :    476     Variable:   529    Variable:   471   Median : 10.9
##  Mean   :  37060                                          Mean   : 11.7
##  3rd Qu.:   7457                                          3rd Qu.: 12.0
##  Max.   :5029023                                          Max.   :154.7
##  NA's   :475230                                           NA's   :480519
##  variableBorrowRate
##  Min.   :  0.0
##  1st Qu.:  3.8
##  Median :  3.9
##  Mean   :  5.7
##  3rd Qu.:  5.1
##  Max.   :148.7
##  NA's   :480519
```

First we'll do some preliminary analysis before we ask detailed questions.

Analyze Transaction Types

Let's examine the different types of *transactions* present in the data. We'll make a simple bar plot to visualize the number of each transaction types. "Deposit" is the most common type of transaction, whereas "swaps" are the most rare.

```r
#set color palette
colors = brewer.pal(6,"Set2")

#create barplot
barplot(table(df$type), main='Transaction Type Counts', xlab='Type',ylab='Count',col=colors)
```

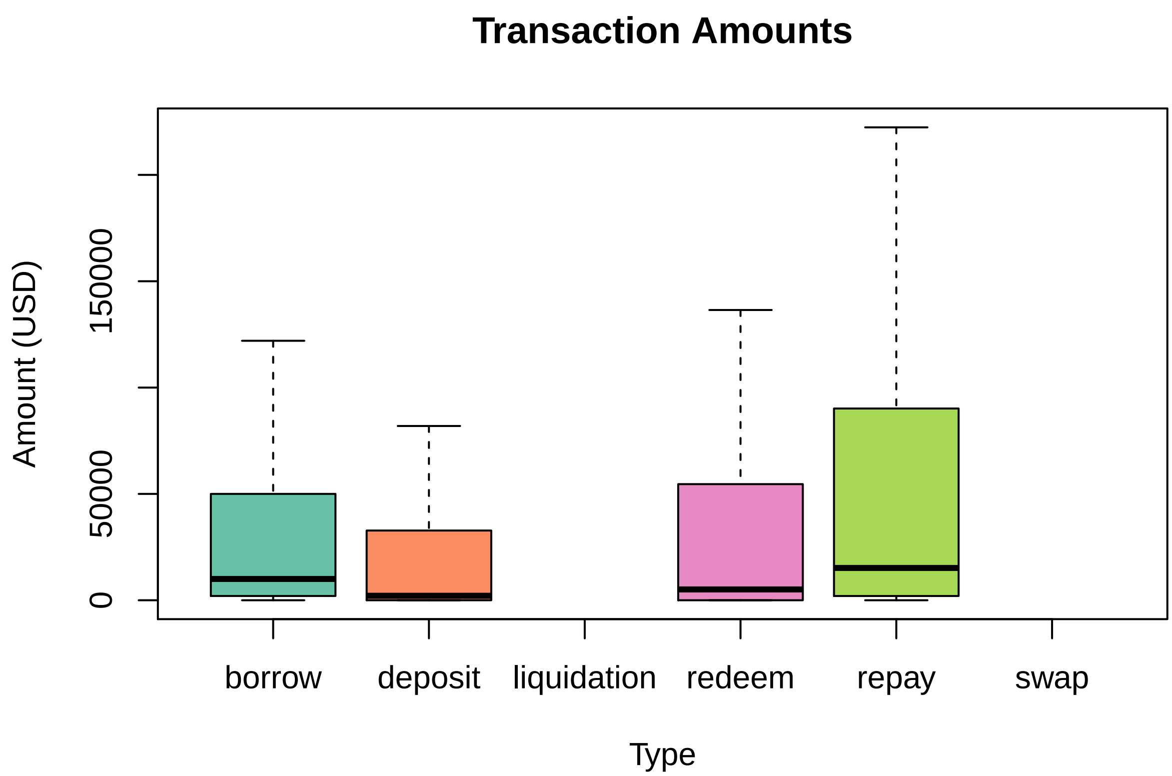


There are more "deposits" than "borrows," because users often need to overcollateralize for loans.

Now we'll examine the amount of US dollars being used in the different types of transactions. We create box plots for the four types of transactions that have the "amount" feature associated with them, and we visualize the distribution of that column for the different transactions.

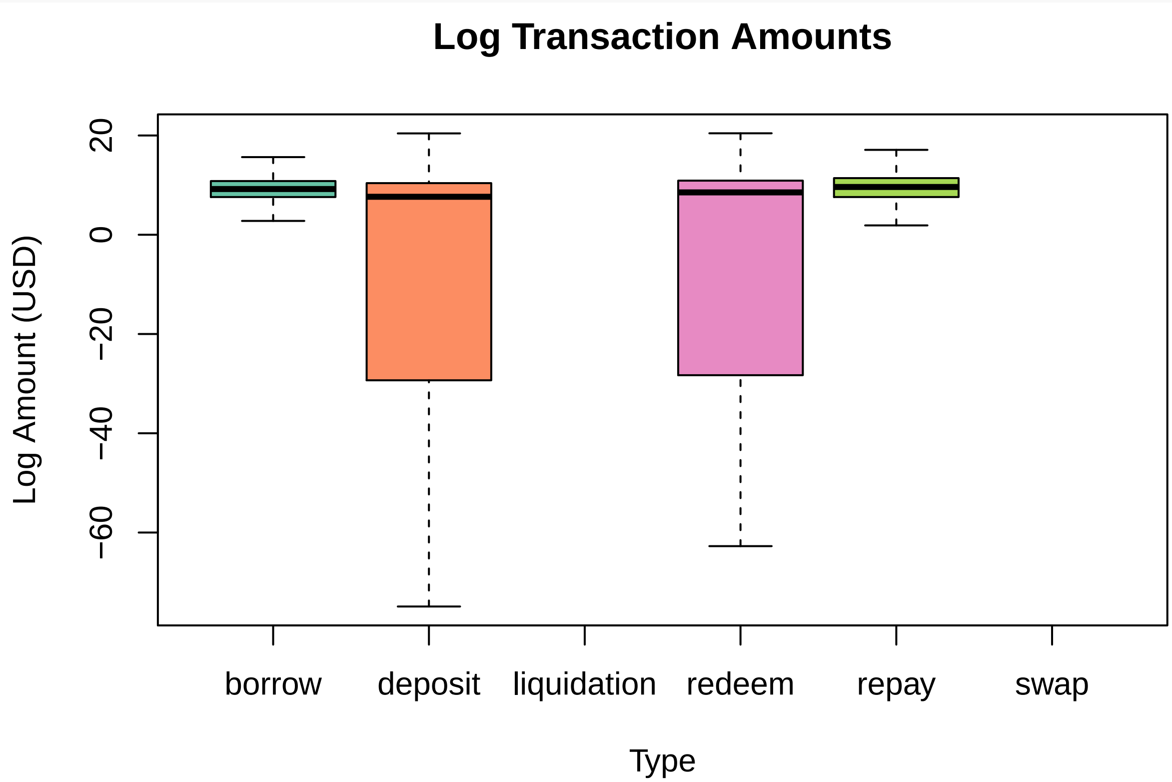
We can see that most transactions are completed with very little money.

```
#create boxplot
boxplot(amountUSD~type,data=df,outline=FALSE,col=colors,
        main="Transaction Amounts",xlab="Type",ylab="Amount (USD)")
```



We do find some very large amounts, so it's helpful to look at this on a log scale.

```
boxplot(log(amountUSD)~type,data=df,outline=FALSE,col=colors,
        main="Log Transaction Amounts",xlab="Type",ylab="Log Amount (USD)")
```



Observation: *There are many borrows and repays with high transactions amounts, but deposits and redeems* *have much lower transactions amounts.*

Examine Reserve Coins

There are 50 different "Reserve" coins used in transactions in AAVE. Let's create a table of those reserve coins with at least 500 transactions and rank order them by their volume.

```
# Use deplyr to drop NA reserves, add the counts and then kep only the top 20
reservecoins <- df %>%  drop_na(reserve) %>%
  count(reserve) %>%
  arrange(-n) %>%
  head(20)

# Add the rank to help keep track of the reserve coins
reservecoins <- reservecoins %>%
  mutate(rank=1:nrow(reservecoins),.before=reserve)

# List the results nicely with kable()
kable(reservecoins)
```

rank	reserve	n
1	USDC	105937
2	WETH	105279
3	USDT	58266
4	DAI	55211
5	LINK	26404
6	WBTC	26344
7	AAVE	12174
8	CRV	10593
9	UNI	7547
10	XSUSHI	7337
11	SNX	6938
12	SUSD	6542
13		6289
14	GUSD	6009
15	YFI	5919
16	BUSD	4863
17	TUSD	3317
18	BAL	3152
19	MKR	3101
20	REN	2638

Let's look at the number of transactions types for each currency.

```
#TopcoinSummary <- df %>% filter(reserve %in% reservecoins$reserve) %>%
#  group_by(reserve == "SNX") %>%
#  count(type) %>%
#  mutate(percent = n/sum(n)*100)
#kable(TopcoinSummary)


TopcoinSummary <- df %>% filter(reserve %in% reservecoins$reserve) %>%
  group_by(reserve) %>%
  count(type) %>%
```

```r
  mutate(percent = n/sum(n)*100)

kable(TopcoinSummary)
```

reserve	type	n	percent
	liquidation	6289	100.0000000
AAVE	borrow	2	0.0164285
AAVE	deposit	7028	57.7295876
AAVE	redeem	5141	42.2293412
AAVE	repay	3	0.0246427
BAL	borrow	215	6.8210660
BAL	deposit	2171	68.8769036
BAL	redeem	612	19.4162437
BAL	repay	154	4.8857868
BUSD	borrow	1685	34.6493934
BUSD	deposit	1135	23.3395024
BUSD	redeem	836	17.1910343
BUSD	repay	1207	24.8200699
CRV	borrow	1054	9.9499670
CRV	deposit	5780	54.5643349
CRV	redeem	2607	24.6105919
CRV	repay	1152	10.8751062
DAI	borrow	14133	25.5981598
DAI	deposit	18552	33.6019996
DAI	redeem	13381	24.2361124
DAI	repay	8895	16.1109199
DAI	swap	250	0.4528083
GUSD	borrow	2282	37.9763688
GUSD	deposit	1493	24.8460642
GUSD	redeem	967	16.0925279
GUSD	repay	1267	21.0850391
LINK	borrow	1321	5.0030298
LINK	deposit	15270	57.8321466
LINK	redeem	8713	32.9987881
LINK	repay	1097	4.1546735
LINK	swap	3	0.0113619
MKR	borrow	188	6.0625605
MKR	deposit	1766	56.9493712
MKR	redeem	986	31.7961948
MKR	repay	159	5.1273783
MKR	swap	2	0.0644953
REN	borrow	196	7.4298711
REN	deposit	1417	53.7149356
REN	redeem	840	31.8423048
REN	repay	183	6.9370735
REN	swap	2	0.0758150
SNX	borrow	433	6.2409916
SNX	deposit	4002	57.6823292
SNX	redeem	2052	29.5762468
SNX	repay	451	6.5004324
SUSD	borrow	1277	19.5200245
SUSD	deposit	2403	36.7318863
SUSD	redeem	1781	27.2240905

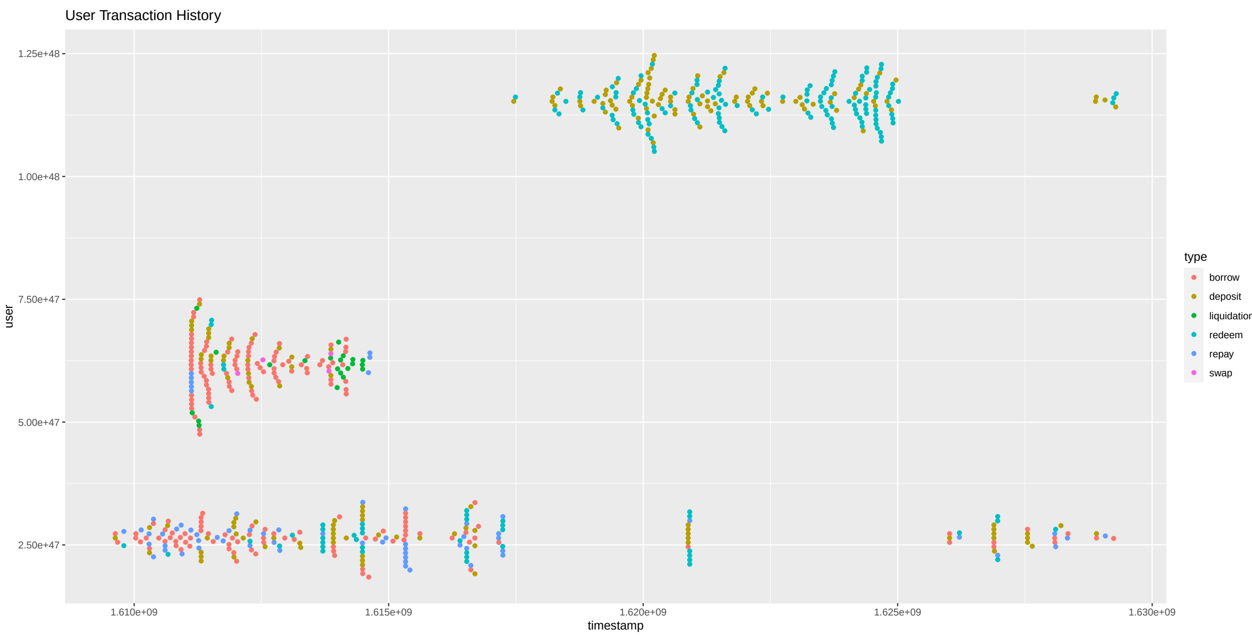
reserve	type	n	percent
SUSD	repay	1081	16.5239988
TUSD	borrow	991	29.8763943
TUSD	deposit	853	25.7160084
TUSD	redeem	661	19.9276455
TUSD	repay	796	23.9975882
TUSD	swap	16	0.4823636
UNI	borrow	567	7.5129190
UNI	deposit	3912	51.8351663
UNI	redeem	2540	33.6557573
UNI	repay	527	6.9829071
UNI	swap	1	0.0132503
USDC	borrow	35469	33.4812200
USDC	deposit	27586	26.0400049
USDC	redeem	22131	20.8907181
USDC	repay	20326	19.1868752
USDC	swap	425	0.4011818
USDT	borrow	22332	38.3276697
USDT	deposit	12593	21.6129475
USDT	redeem	10349	17.7616449
USDT	repay	12719	21.8291971
USDT	swap	273	0.4685408
WBTC	borrow	2082	7.9031278
WBTC	deposit	13994	53.1202551
WBTC	redeem	8442	32.0452475
WBTC	repay	1816	6.8934103
WBTC	swap	10	0.0379593
WETH	borrow	7234	6.8712659
WETH	deposit	56373	53.5462913
WETH	redeem	35505	33.7246744
WETH	repay	6155	5.8463701
WETH	swap	12	0.0113983
XSUSHI	borrow	242	3.2983508
XSUSHI	deposit	4382	59.7246831
XSUSHI	redeem	2454	33.4469129
XSUSHI	repay	259	3.5300532
YFI	borrow	403	6.8085825
YFI	deposit	2976	50.2787633
YFI	redeem	2146	36.2561243
YFI	repay	394	6.6565298

Look at Sample User Transaction Histories

Finally, we will examine the transaction history of different users. To do this, we will select 3 random users from the data who have completed between 100 and 300 transactions. Then, we create swarmplots displaying the different types of transactions those users made over time.

```
#set seed
set.seed(1)

# Select three random users that have between 100 and 300 transactions
users<-vector(length=3)
count<-0
```

```r
while(count<=3){
  success<-FALSE
  while(!success){
    #get random user
    ruser<-sample(df$user,1)

    #check for valid number of transactions
    length<-nrow(filter(df,user==ruser))
    if (length>100 && length<300){
      users[count]=ruser
      success<-TRUE
      count<-count+1
    }
  }
}
df.rusers<-filter(df, user %in%users)

# Create a "swarmplot"

ggplot(df.rusers,aes(user, timestamp,color=type)) +
        geom_beeswarm(cex=1)+
        coord_flip()+
        ggtitle("User Transaction History")
```



Observation: *Users have very different transactions patterns, which we will try to better understand.*

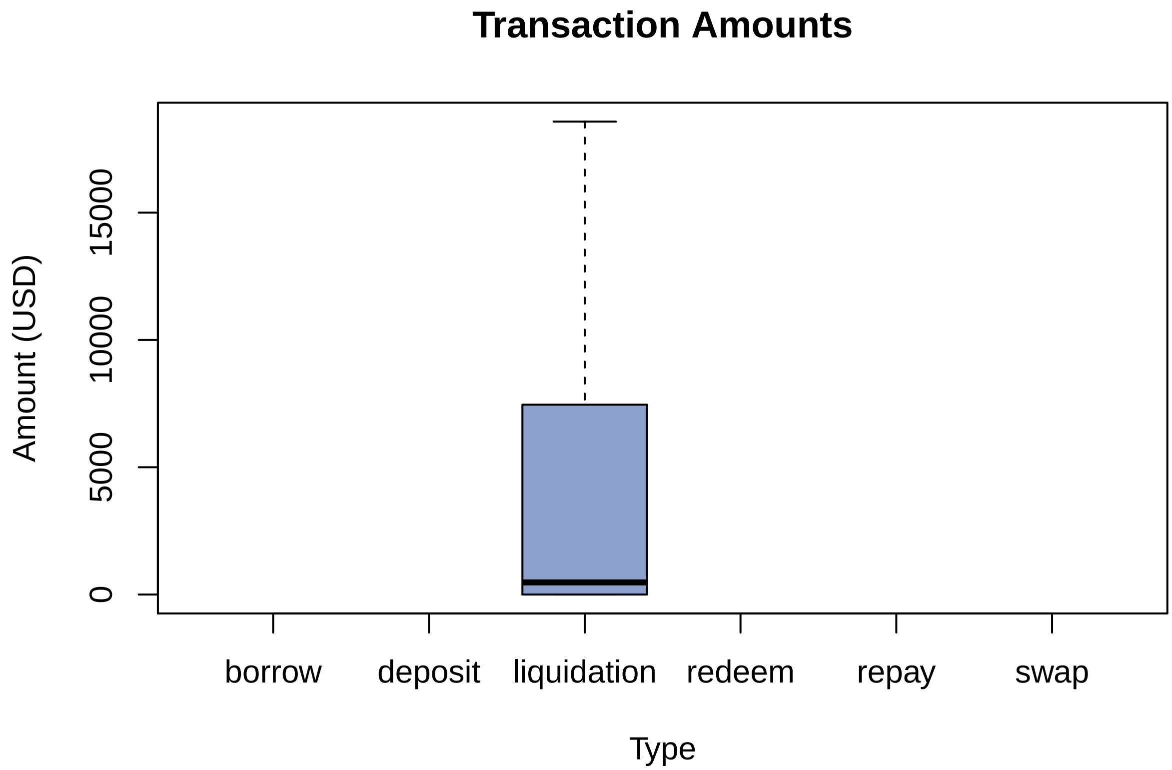
Let's do some exploring of the columns, since most of it is left unexplained. More specifically, let's take a look at amountUSDCollateral which is what I assume is the collateral put down by a borrower (in USD units).

```r
droppedcollat_full <- drop_na(df, amountUSDCollateral) # Remove NA items.
nonNAlength <- length(droppedcollat_full$amountUSDCollateral)
numNA <- length(df$amountUSDCollateral) - nonNAlength
sprintf("Number of N/A: %d, Number non-N/A: %d, Fraction: %f",numNA, nonNAlength, nonNAlength/numNA)
```

```
## [1] "Number of N/A: 475230, Number non-N/A: 6289, Fraction: 0.013234"
```

Odd that the number of non-N/A is so slim. Taking a look at the box plot of types tells us a lot more:

```r
boxplot(amountUSDCollateral~type,data=droppedcollat_full,outline=FALSE,col=colors,
        main="Transaction Amounts",xlab="Type",ylab="Amount (USD)")
```



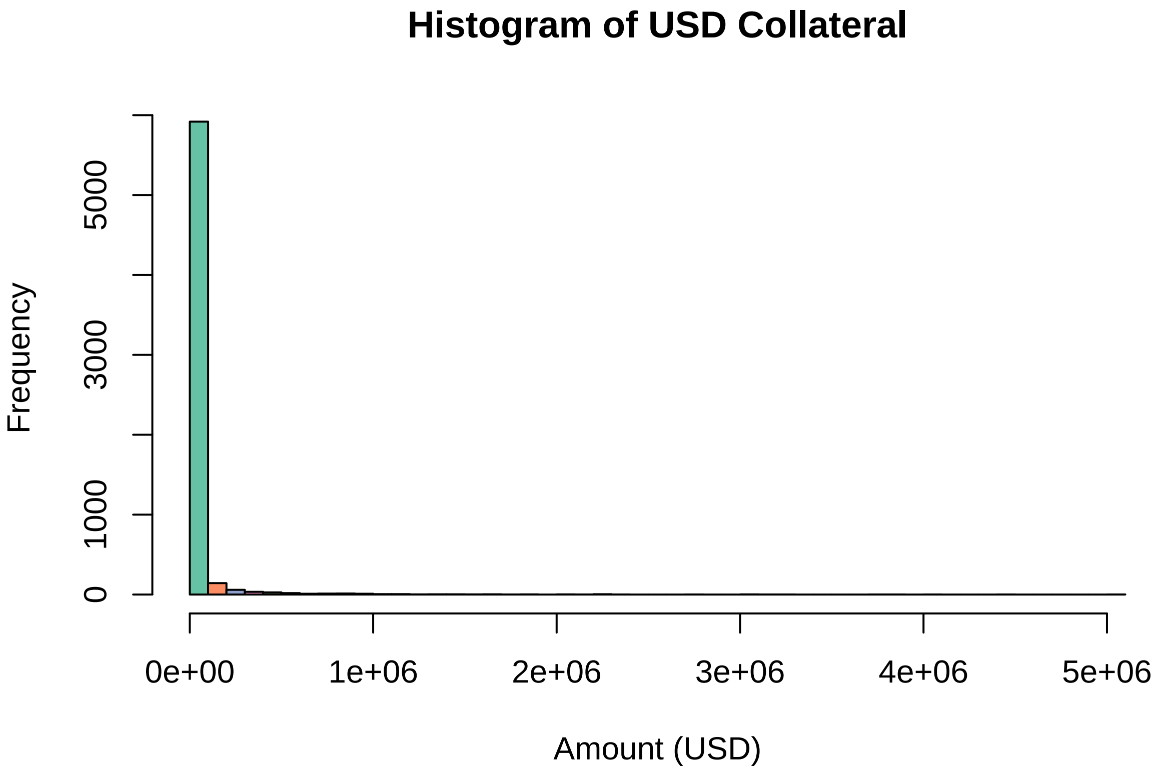
We see that these data points are likely just the liquidation in USD amount. Let's take a look at the overall values it takes on.

```r
# Take a look at the first 100 lowest
head(df[order(df$amountUSDCollateral),]$amountUSDCollateral, 100)
```

```
##   [1] 4.770112e-20 5.997939e-20 1.029948e-18 1.584272e-18 2.065990e-18
##   [6] 2.075215e-18 3.212388e-18 4.082090e-18 4.143902e-18 4.383437e-18
##  [11] 5.710297e-18 5.997939e-18 8.284528e-18 8.739470e-18 1.015305e-17
##  [16] 1.050894e-17 1.521751e-17 1.587503e-17 1.656905e-17 1.747892e-17
##  [21] 1.804793e-17 1.837928e-17 3.175005e-17 3.210753e-17 3.309996e-17
##  [26] 3.495782e-17 3.523883e-17 4.051611e-17 4.061692e-17 4.679315e-17
##  [31] 5.129117e-17 5.703784e-17 5.851092e-17 6.319520e-17 6.350002e-17
##  [36] 6.472775e-17 6.619980e-17 7.033510e-17 7.155240e-17 7.332113e-17
##  [41] 7.353641e-17 7.370632e-17 7.400488e-17 8.547401e-17 8.645241e-17
##  [46] 9.144959e-17 9.358623e-17 9.570547e-17 1.013899e-16 1.050938e-16
##  [51] 1.146598e-16 1.153353e-16 1.158542e-16 1.223440e-16 1.263882e-16
##  [56] 1.264038e-16 1.265827e-16 1.279406e-16 1.290922e-16 1.323996e-16
##  [61] 1.332625e-16 1.371782e-16 1.388186e-16 1.406700e-16 1.429720e-16
##  [66] 1.435987e-16 1.474126e-16 1.480097e-16 1.493615e-16 1.510555e-16
##  [71] 1.581759e-16 1.642312e-16 1.727173e-16 1.729048e-16 1.827309e-16
##  [76] 1.871573e-16 1.871724e-16 1.929308e-16 2.042679e-16 2.076713e-16
##  [81] 2.109226e-16 2.128683e-16 2.129609e-16 2.135648e-16 2.141564e-16
##  [86] 2.150857e-16 2.171442e-16 2.195820e-16 2.200354e-16 2.206686e-16
##  [91] 2.233677e-16 2.260725e-16 2.266131e-16 2.271079e-16 2.283382e-16
##  [96] 2.288628e-16 2.315837e-16 2.316386e-16 2.330899e-16 2.333277e-16
```

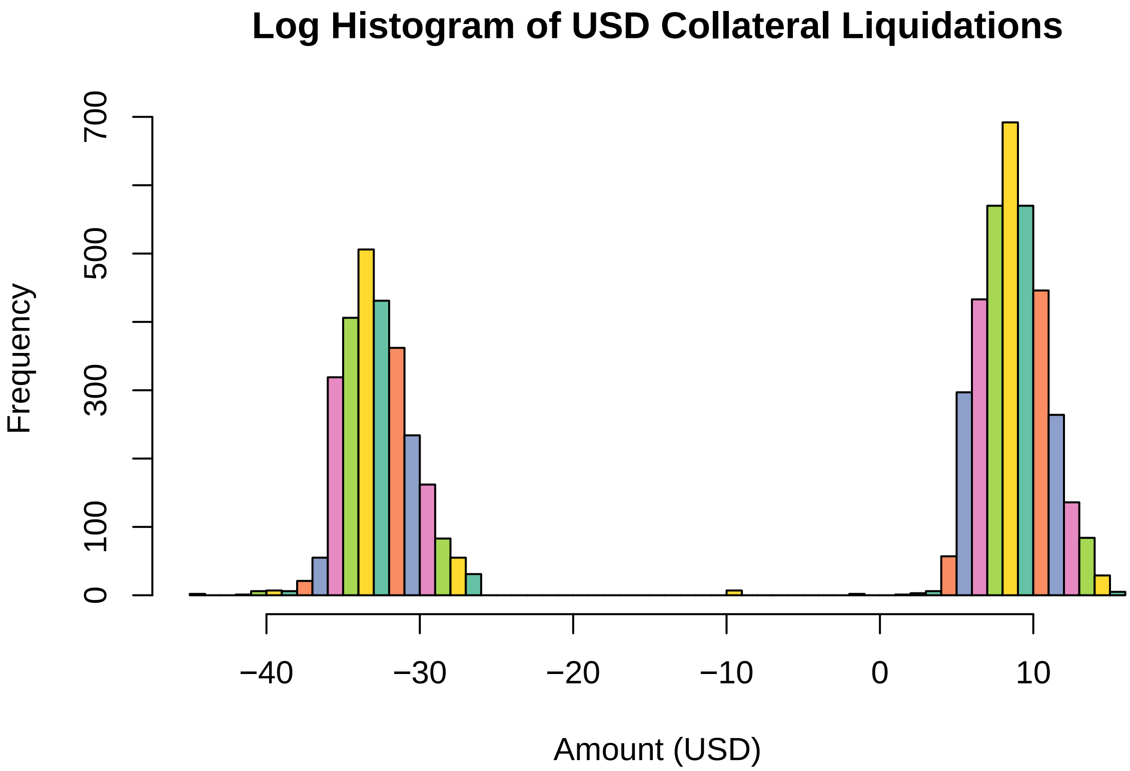
The values are effectively 0. It might be a good idea to take a look at the spread of the values.

```
hist(droppedcollat_full$amountUSDCollateral,breaks = 50,col=colors,
        main="Histogram of USD Collateral",xlab="Amount (USD)",ylab="Frequency")
```



Let's set an arbitrary threshold to see how it's distributed.

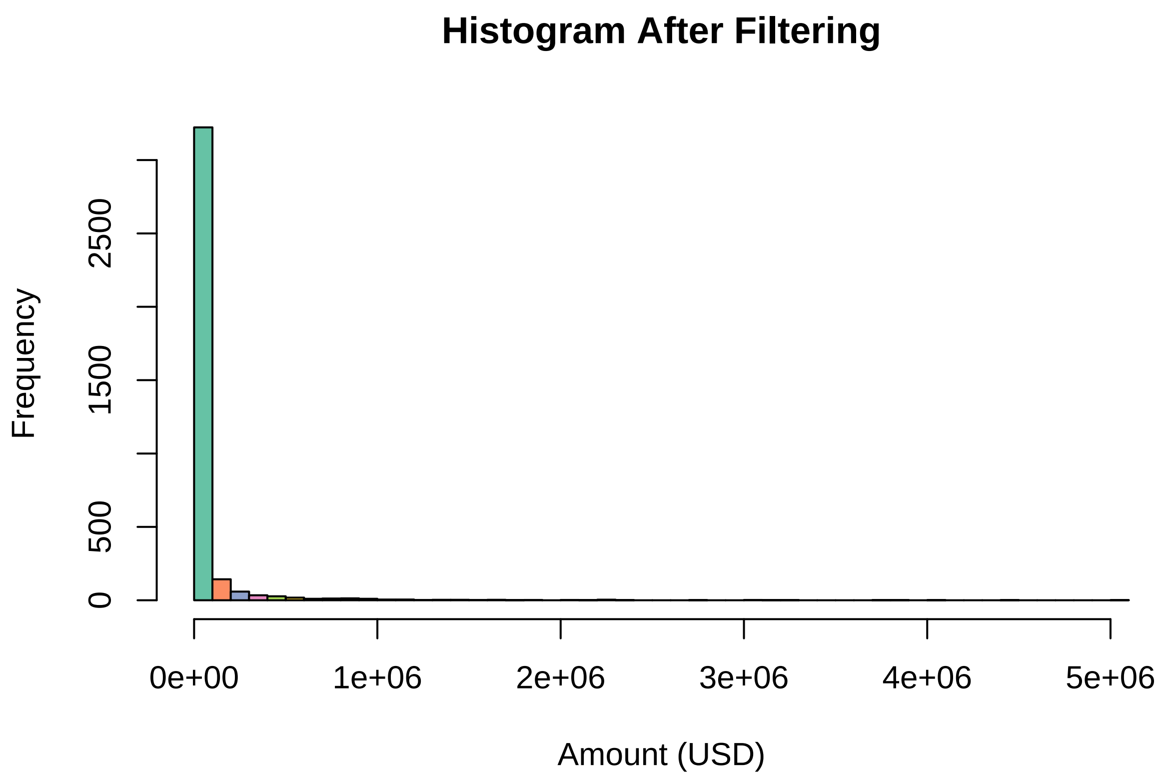
```
eps = 1 # filter out small dollar values to see evaluate the spread.
droppedcollat = droppedcollat_full
hist(log(droppedcollat$amountUSDCollateral),breaks =50,col=colors,
        main="Log Histogram of USD Collateral Liquidations",xlab="Amount (USD)",ylab="Frequency")
```



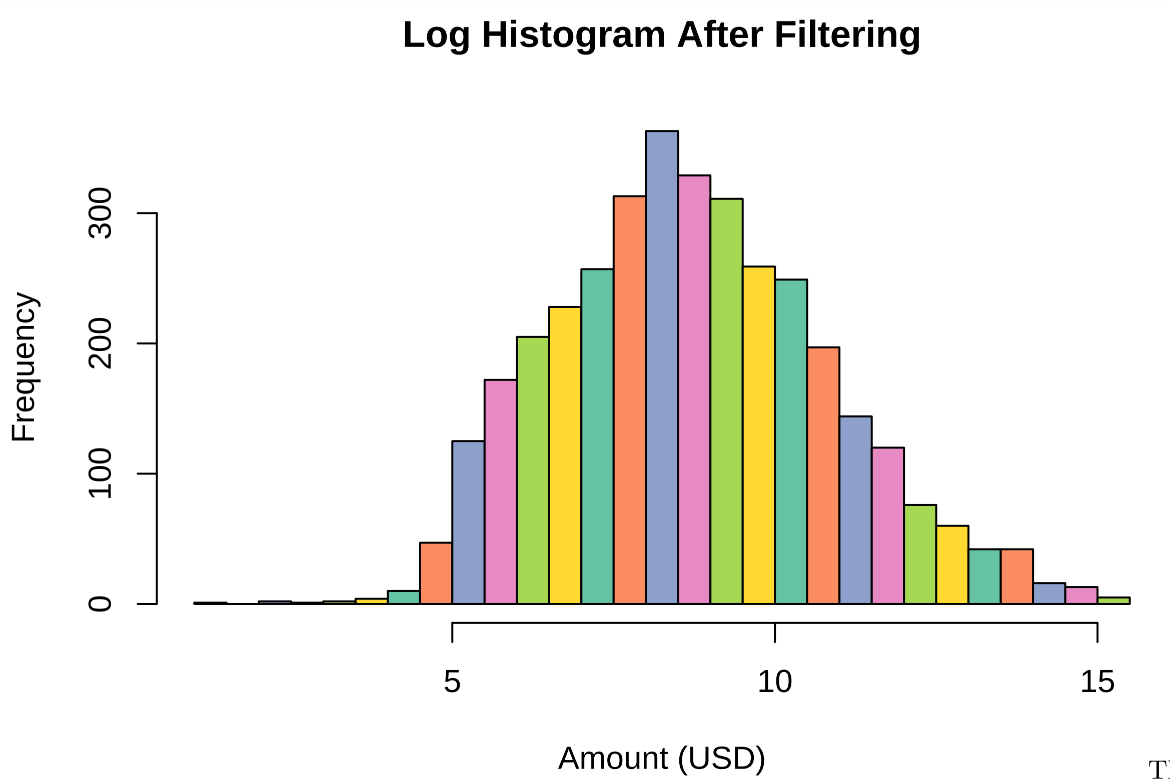

```r
num_thresholded = length(droppedcollat$amountUSDCollateral[droppedcollat$amountUSDCollateral <= eps])
droppedcollat$amountUSDCollateral[droppedcollat$amountUSDCollateral <= eps] <- 0
zero_droppedcollat <- droppedcollat[droppedcollat$amountUSDCollateral > eps,]
sprintf("Amount below eps = %d removed: %d, Total fraction removed: %f", eps, num_thresholded, num_thre
```

```
## [1] "Amount below eps = 1 removed: 2696, Total fraction removed: 0.428685"
```

```r
hist(zero_droppedcollat$amountUSDCollateral,breaks =50,col=colors,
        main="Histogram After Filtering",xlab="Amount (USD)",ylab="Frequency")
```




```r
hist(log(zero_droppedcollat$amountUSDCollateral),breaks =50,col=colors,
        main="Log Histogram After Filtering",xlab="Amount (USD)",ylab="Frequency")
```



The fraction of liquidations below or equal to eps = \$1 is about 42.9%, which is very interesting. Usually we can take a look at the 3rd and 4th moments for skew information, but taking a look at the mean, median gap, and variance tells us quite a bit. Also, the log plot is enlightening as to how the data is distributed - it's bimodal.

It'd be interesting to see where these ~0 dollar liquidations are coming from.

```
mn = mean(droppedcollat$amountUSDCollateral)
med = median(droppedcollat$amountUSDCollateral)
vr = var(droppedcollat$amountUSDCollateral)
sprintf("mean: %f, median: %f, variance: %f", mn, med, vr)
```

```
## [1] "mean: 37060.250366, median: 476.232627, variance: 42670202743.255920"
```

Despite the majority of liquidations being quite low, the few extremely large liquidations push the mean up quite a bit. Exploring the behavior of the users that were liquidated with high USD collateral values may be interesting to look into.