DAR F21 Project Status Notebook: DeFi Coin Types

DeFi

Roman Vakhrushev (vakhrr)

09/30/2021

Contents

#Initial Attempt to Separate Coins

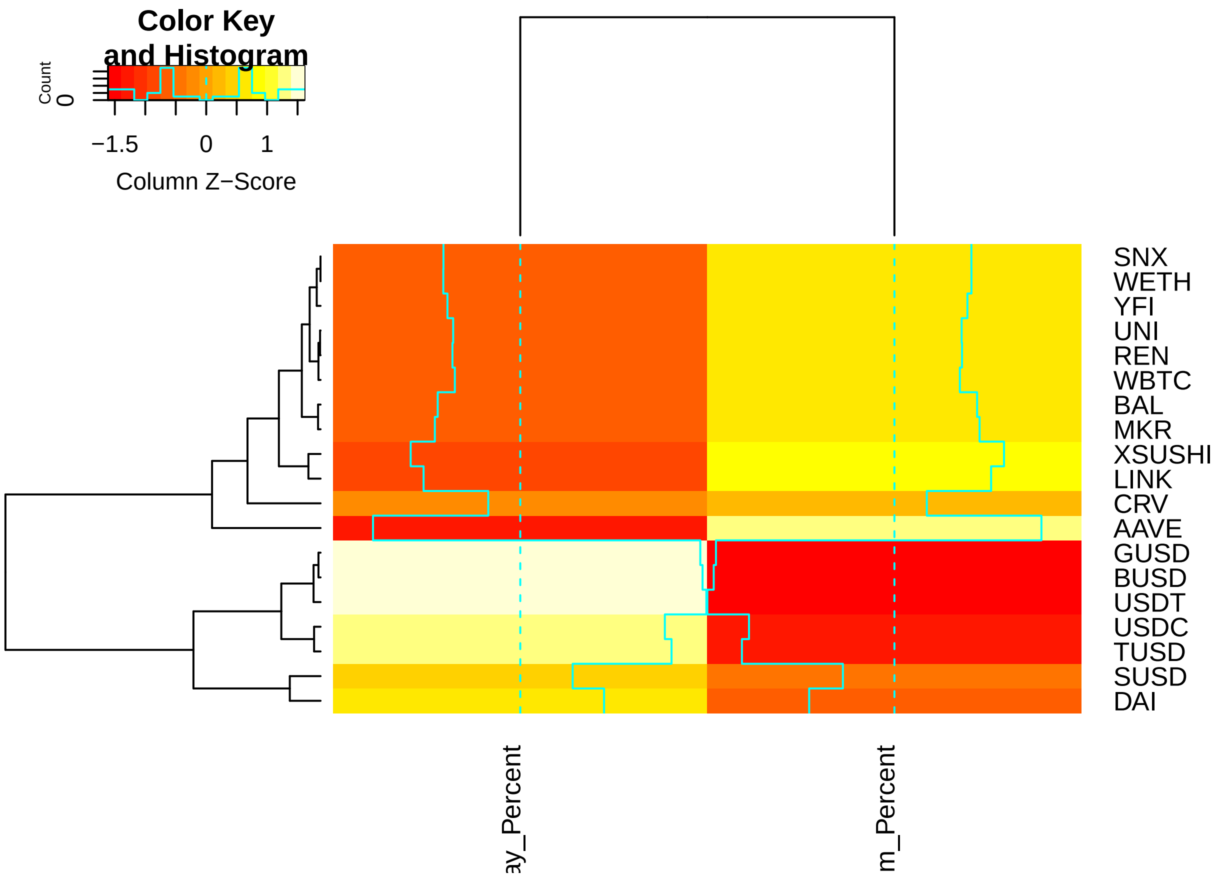
After looking into the data, I was very interested in different usage patterns that could be observed in stable and non-stable coins. The features that seemed to separate non-stable coins from stable were total percents of deposit + redeem and borrow + repay. Therefore, I created the features and

```
#Note: This code was the first attempt to do the coin analysis, it is rather long and inefficient. Th

#Read data, create initial summary
df<-read_rds('../../Data/transactions.Rds')
# 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

rank	reserve	n
14	GUSD	6009
15	YFI	5919
16	BUSD	4863
17	TUSD	3317
18	BAL	3152
19	MKR	3101
20	REN	2638

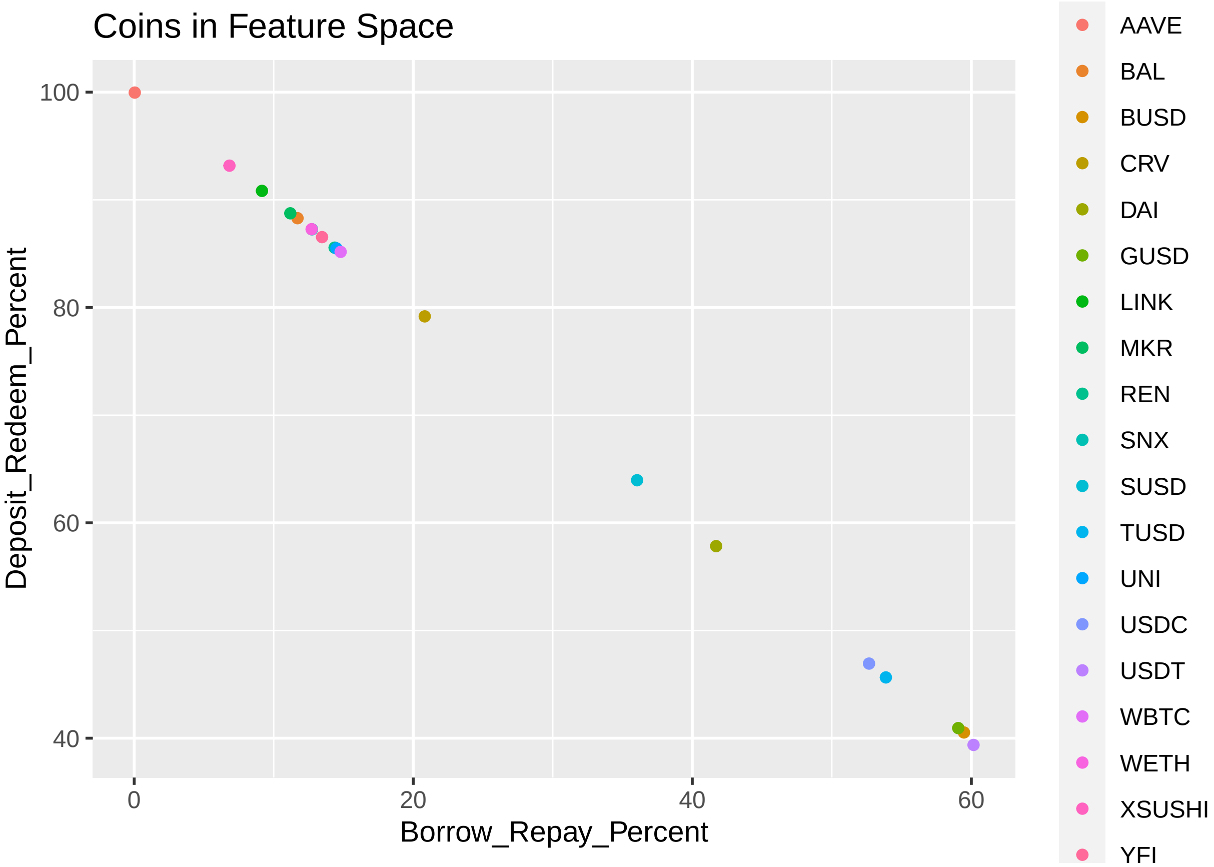
```r
CoinSummary <- df %>% filter(reserve %in% reservecoins$reserve) %>%
group_by(reserve) %>%
count(type) %>%
mutate(percent = n/sum(n)*100)

#Separate summary into different type of transactions
CoinSummaryDep <- CoinSummary %>% subset(type == 'deposit')
CoinSummaryBor <- CoinSummary %>% subset(type == 'borrow')
CoinSummaryRed <- CoinSummary %>% subset(type == 'redeem')
CoinSummaryRep <- CoinSummary %>% subset(type == 'repay')

#Create separate summaries for our features: Borrow+Repay and Deposit+Redeem
CoinSummaryBorRep <- rbind(CoinSummaryBor, CoinSummaryRep)
CoinSummaryDepRed <- rbind(CoinSummaryDep, CoinSummaryRed)

#Have to sum over correct column in order to compute percent correctly
CoinSummaryBorRep <- aggregate(percent ~ reserve, data=CoinSummaryBorRep, FUN=sum)
CoinSummaryDepRed <- aggregate(percent ~ reserve, data=CoinSummaryDepRed, FUN=sum)

#Merge summaries in order to make a one summary that contains all the information
CoinSummary <- merge(CoinSummaryBorRep,CoinSummaryDepRed, by="reserve") %>%
rename(Borrow_Repay_Percent = percent.x, Deposit_Redeem_Percent = percent.y)

#Have to create stability column manually in order to represent stable and non-stable coins
stability = c('non-stable','non-stable','stable','non-stable','stable','stable','non-stable','non-stal

#This piece of code makes the first column of the dataframe to be a rownames column,
#which is important for heatmaps
NewCoinSummary <- CoinSummary
rownames(NewCoinSummary) <- NewCoinSummary[,1] #Assigning row names from 1st column
NewCoinSummary[,1] <- NULL #Removing the first column

NewCoinSummary <- data.matrix(NewCoinSummary)

#Initial Heatmap that was constructed, includes only new features
heatmap.2(NewCoinSummary,scale="column",cexRow=1,cexCol=1)
```



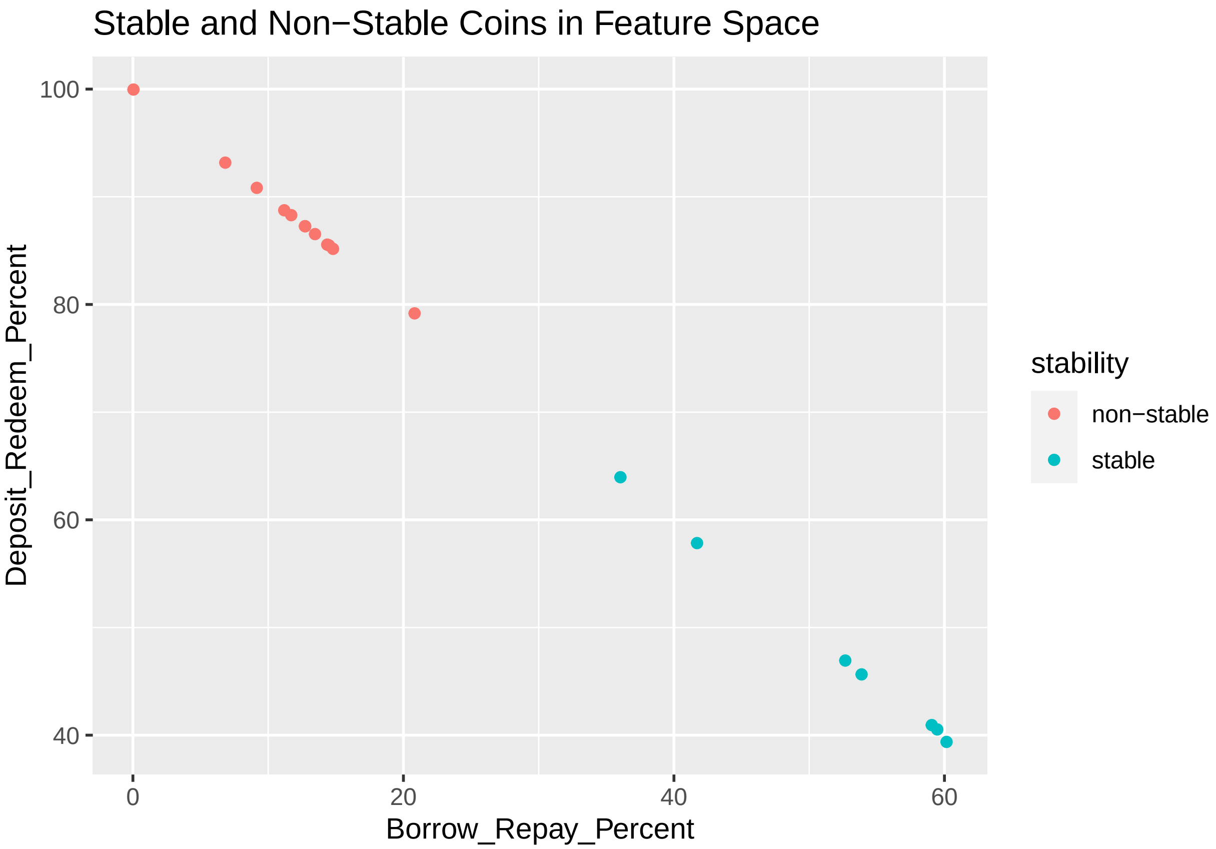
```r
CoinSummary$stability <- stability

#Table
kable(CoinSummary)
```

reserve	Borrow_Repay_Percent	Deposit_Redeem_Percent	stability
AAVE	0.0410711	99.95893	non-stable
BAL	11.7068528	88.29315	non-stable
BUSD	59.4694633	40.53054	stable
CRV	20.8250732	79.17493	non-stable
DAI	41.7090797	57.83811	stable
GUSD	59.0614079	40.93859	stable
LINK	9.1577034	90.83093	non-stable
MKR	11.1899387	88.74557	non-stable
REN	14.3669447	85.55724	non-stable
SNX	12.7414240	87.25858	non-stable
SUSD	36.0440232	63.95598	stable
TUSD	53.8739825	45.64365	stable
UNI	14.4958262	85.49092	non-stable
USDC	52.6680952	46.93072	stable
USDT	60.1568668	39.37459	stable
WBTC	14.7965381	85.16550	non-stable
WETH	12.7176360	87.27097	non-stable
XSUSHI	6.8284040	93.17160	non-stable
YFI	13.4651124	86.53489	non-stable

```
#Graphs
ggplot(CoinSummary,aes(Borrow_Repay_Percent, Deposit_Redeem_Percent, color=reserve)) + geom_point() +
```



```
ggplot(CoinSummary,aes(Borrow_Repay_Percent, Deposit_Redeem_Percent, color=stability)) + geom_point()
```



As we can see from the results, the created features completely separate the two groups (stable and non-stable coins) from each other. In fact even one feature is enough, the other one is just for convenience. Non-stable coins are mostly deposits+redeem and stable coins are mostly borrow+repay. Even more interestingly, there is a huge gap between the two groups.

It also seems like almost all non-stable coins perform similarly in terms of the features except for AAVE coin, which is also 100% of deposits+redeems, and CRV token, which did not achieve even 80% of deposits + borrows (all other non-stable coins are 85%+). One reason of why non-stable group could behave this way would be high risks associated with borrowing non-stable coins, which could potentially drop in value and lead to liquidation. So, non-stable coins are just used for depositing mostly and are not very popular for borrowers.

The stable group seems to be a bit more diverse. We have two coins that are around 60% of deposits+redeems (which is rather high for stable coins) these are DAI and SUSD. Then we have two other coins that are 45-50% deposits+redeems (TUSD,USDC) and the rest are about 40% deposits+redeems. Similarly, the reasons for these behaving in such a way could be similar to those for stable coins – risk. It would also be very interesting to see if borrow rates for different coins correlate with these trends (and to study how borrow rates are actually created and adjusted) as it could potentially affect the trends as well (theoretically, high availability of some coins should affect the borrow rates).

#More Detailed Analysis of Coin Types

This time I decided to extend the analysis to not only features that I created, but also to other simple features available in the data right away. So, I added such features as amount of transactions of different type, average value of transactions in usd, total value of transactions in usd. The analysis targeted to identify some other types of coins (bot just stable vs non-stable)

```
# Create new coin summary to study other types of coins
CoinSummary <- df %>% filter(reserve %in% reservecoins$reserve) %>%
group_by(reserve) %>%
```

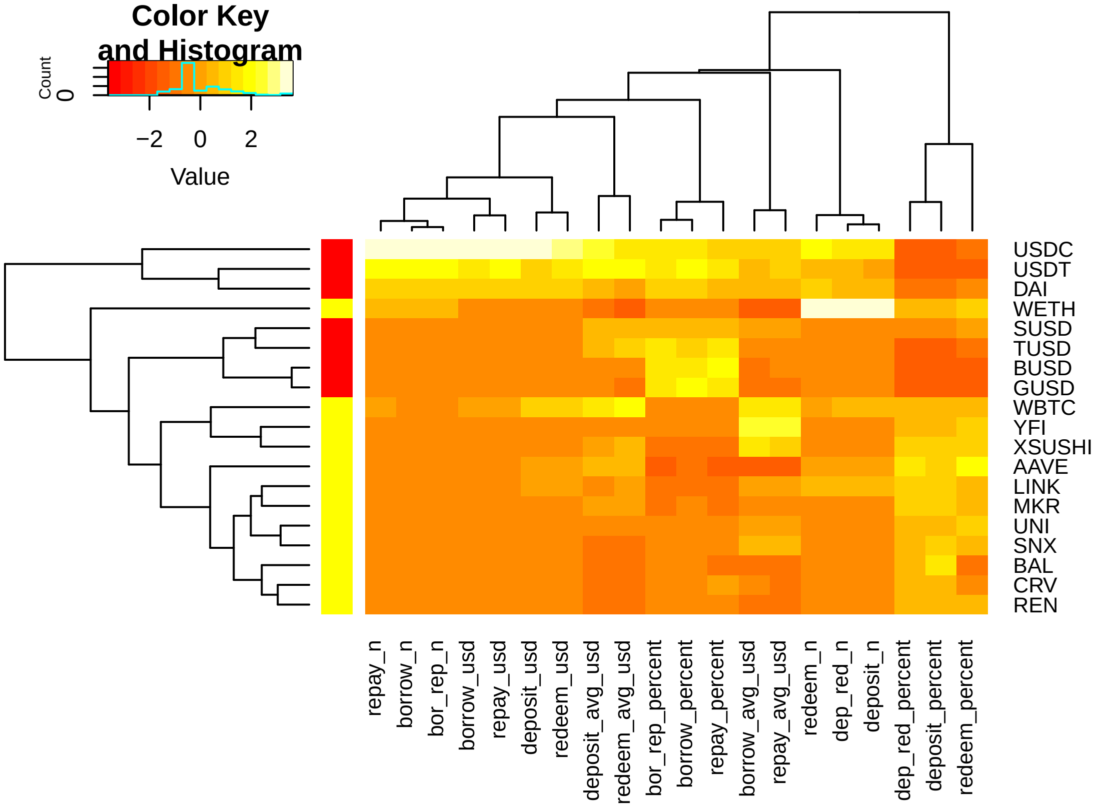
```r
count(type) %>%
mutate(percent = n/sum(n)*100)

#filter liquidations and swaps, group transactions by reserve and type and also create basic
#statistic for each group
CoinSummary2 <- df %>% filter(reserve %in% reservecoins$reserve) %>% filter(type != "liquidation" & ty
group_by(reserve,type) %>% summarize(avg_usd = mean(amountUSD),usd = sum(amountUSD),n=n()) %>% mutate
```

```
## `summarise()` has grouped output by 'reserve'. You can override using the `.groups` argument.
```

```r
#Now we have to move the data to wide format (for heatmap and table)
#These lines of code widen the data in appropriate format
CoinSummary2Wide <- CoinSummary2 %>% gather(variable, value, -(reserve:type)) %>%
  unite(temp, type, variable) %>%
  spread(temp, value)

#Create new features(borrow+repay, deposit+redeem and their variations)
CoinSummary2Wide <- CoinSummary2Wide %>% mutate(bor_rep_percent=borrow_percent+repay_percent, bor_rep_

kable(CoinSummary2Wide)
```

reserve	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
AAVE	2.17796 2	0.0164[illegible]	57.729[illegible]	42.229[illegible]	0.0246[illegible]	99.958[illegible]
BAL	74815.[illegible]	[illegible]	68.876[illegible]	19.416[illegible]	[illegible]	88.293[illegible]
BUSD	[illegible]	[illegible]	23.339[illegible]	17.191[illegible]	[illegible]	40.530[illegible]
CRV	104106[illegible]	[illegible]	54.564[illegible]	24.610[illegible]	[illegible]	79.174[illegible]
DAI	285831[illegible]	[illegible]	33.754[illegible]	24.346[illegible]	[illegible]	58.101[illegible]
GUSD	[illegible]	[illegible]	24.846[illegible]	16.092[illegible]	[illegible]	40.938[illegible]
LINK	168380[illegible]	[illegible]	57.838[illegible]	33.002[illegible]	[illegible]	90.841[illegible]
MKR	109391[illegible]	[illegible]	56.986[illegible]	31.816[illegible]	[illegible]	88.802[illegible]
REN	86000.[illegible]	[illegible]	53.755[illegible]	31.866[illegible]	[illegible]	85.622[illegible]
SNX	309288[illegible]	[illegible]	57.682[illegible]	29.576[illegible]	[illegible]	87.258[illegible]
SUSD	[illegible]	[illegible]	36.731[illegible]	27.224[illegible]	[illegible]	63.955[illegible]
TUSD	[illegible]	[illegible]	25.840[illegible]	20.024[illegible]	[illegible]	45.864[illegible]
UNI	208502[illegible]	[illegible]	51.842[illegible]	33.660[illegible]	[illegible]	85.502[illegible]
USDC	[illegible]	[illegible]	26.144[illegible]	20.974[illegible]	[illegible]	47.119[illegible]
USDT	[illegible]	[illegible]	21.714[illegible]	17.845[illegible]	[illegible]	39.559[illegible]
WBTC	[illegible]	[illegible]	53.140[illegible]	32.057[illegible]	[illegible]	85.197[illegible]
WETH	[illegible]	6.8720[illegible]	53.5524[illegible]	33.7285[illegible]	5.8470[illegible]	87.280[illegible]
XSUSHI	[illegible]	[illegible]	59.724[illegible]	33.446[illegible]	[illegible]	93.171[illegible]
YFI	579352[illegible]	[illegible]	50.278[illegible]	36.256[illegible]	[illegible]	86.534[illegible]

```r
#create row names for column names for heatmap
CoinSummary2Wide <- CoinSummary2Wide %>% rename(rowname = reserve) %>% column_to_rownames()

#create separate bar for heatmap to indicate stability
stability_colored <- stability %>% replace(stability == 'non-stable','yellow') %>% replace(stability =
heatmap.2(scale(CoinSummary2Wide),trace="none",RowSideColors = stability_colored,margin=c(10, 10))
```



```
#this is a regular version of heatmap with their scaling
#heatmap.2(CoinSummary2Wide,scale="column")
```

The following heatmap summarizes the coin data in more details than the old graph. The bar on the left indicates stability of coins: red means stable, yellow non-stable. In the map lighter colors mean "more", darker mean "less".

First of all, we see the clearly non-stable vs stable clusters we already saw on the graph. For example, the last 3 columns indicate that very well.

The first two rows show that USDC and USDT coins completely dominate the borrow-repay market. There are both a lot of transactions and a lot of usd amounts for borrow and repay for these coins, although percentages of these are similar to other stable coins.

Another observation is Wrapped Ether WETH, which completely dominates the redeem-deposit market in terms of transactions. There is some issue with usd amount for WETH in the data, so usd results for it are not correct. However, we clearly WETH is the most popular coin for depositing, although it is not used that much for borrowing (although total amount of borrows and repays is much higher compared to other non-stable coins). So good question is why people deposit WETH, what do they want to do with it? It may be used to be put as a collateral, but this data is currently unavailable.

There are some other "groups" that are hard to identify and understand. For example, we see the WBTC, YFI and XSUSHI have very high average borrow and repay transactions in usd, which is strange and interesting, knowing that total borrow and repay for these is very small.

Thoughts on Liquidation Prediction

```
head(df%>% filter(type == "liquidation"))
```

```
##    amount borrowRate borrowRateMode onBehalfOf         pool reserve  timestamp
## 1     NA         NA                         NA 1.034668e+48         1626124715
## 2     NA         NA                         NA 1.034668e+48         1619145033
## 3     NA         NA                         NA 1.034668e+48         1621319875
## 4     NA         NA                         NA 1.034668e+48         1614324006
## 5     NA         NA                         NA 1.034668e+48         1621788289
## 6     NA         NA                         NA 1.034668e+48         1621429473
##           user        type reservePriceETH reservePriceUSD amountUSD
## 1 2.976865e+47 liquidation              NA              NA        NA
## 2 3.748214e+47 liquidation              NA              NA        NA
## 3 1.130833e+48 liquidation              NA              NA        NA
## 4 9.560356e+45 liquidation              NA              NA        NA
## 5 6.451374e+45 liquidation              NA              NA        NA
## 6 1.460589e+48 liquidation              NA              NA        NA
##   collateralAmount collateralReserve principalAmount principalReserve
## 1     3.308552e-01              WETH    6.391700e+02             GUSD
## 2     3.382573e+04               ENJ    5.514457e+04             USDT
## 3     6.345434e+03              USDC    8.207972e-02              YFI
## 4     8.489321e+01              WETH    1.170341e+05             BUSD
## 5     3.749461e+03              LINK    5.206766e+04             USDC
## 6     3.182213e+02              AAVE    1.219451e+05             USDC
##   reservePriceETHPrincipal reservePriceUSDPrincipal reservePriceETHCollateral
## 1             3.268884e+14             6.625489e-01              1.000000e+00
## 2             4.568000e+14             1.000000e+00              4.568000e+14
## 3             2.068914e+19             7.317892e+04              2.810000e+14
## 4             6.908300e+14             1.012487e+00              1.000000e+00
## 5             5.739400e+14             1.012537e+00              8.767150e+15
## 6             4.373258e+14             9.722212e-01              1.843456e+17
##   reservePriceUSDCollateral amountUSDPincipal amountUSDCollateral
## 1              2.026835e-15          423.4814        6.705890e-16
## 2              5.576372e-01        55144.5691        1.886249e+04
## 3              9.939162e-01         6006.5051        6.306830e+03
## 4              1.465609e-15       118495.5106        1.244203e-13
## 5              1.546688e+01        52720.4229        5.799247e+04
## 6              4.098195e+02       118557.5599        1.304133e+05
##   borrowRateModeFrom borrowRateModeTo stableBorrowRate variableBorrowRate
## 1                                                   NA                 NA
## 2                                                   NA                 NA
## 3                                                   NA                 NA
## 4                                                   NA                 NA
## 5                                                   NA                 NA
## 6                                                   NA                 NA
```

```
#collateralAmount - amount of collateral claimed (in corresponding currency)
#c-l reserve - corresponding reserve
#principalAmount - same for principal claimed
#p-l reserve - same for reserve
#principal/collateral amounts/prices in usd/eth

head(df%>% filter(type == "borrow"))
```

```
##       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
```

```r
#onBehalfOf always same as user?

users<-vector(length=3)
count<-0
while(count<=1){
  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)
#kable(df.rusers)

BorrowSummarySum <- df %>% filter(type == "borrow") %>% group_by(reserve) %>% summarize(sum(amountUSD))
BorrowSummaryMean <- df %>% filter(type == "borrow") %>% group_by(reserve) %>%
summarize(mean(amountUSD))
kable(BorrowSummarySum)
```

reserve	sum(amountUSD)
AAVE	4.355919e+00
AmmDAI	1.111405e+07
AmmUSDC	3.336005e+07
AmmUSDT	2.235150e+07
AmmWBTC	7.377355e+06
AmmWETH	0.000000e+00
AMPL	9.522049e+06
BAL	1.608533e+07
BAT	5.216522e+06
BUSD	1.453950e+08
CRV	1.097278e+08
DAI	4.039650e+09
ENJ	1.699836e+07
GUSD	1.232664e+08
KNC	5.920488e+06
LINK	2.224313e+08
MANA	1.153632e+07
MKR	2.056552e+07
PAX	6.073554e+06
RAI	3.883547e+07
REN	1.685605e+07
RENFIL	3.026235e+06
SNX	1.339218e+08
SUSD	2.833338e+08
TUSD	1.374206e+08
UNI	1.182207e+08
USDC	1.300875e+10
USDT	5.793992e+09
WBTC	9.671563e+08
WETH	0.000000e+00
XSUSHI	1.041535e+08
YFI	2.334792e+08
ZRX	9.481052e+05

```
kable(BorrowSummaryMean)
```

reserve	mean(amountUSD)
AAVE	2.17796
AmmDAI	21009.53690

reserve	mean(amountUSD)
AmmUSDC	117880.03931
AmmUSDT	124174.97422
AmmWBTC	153694.88693
AmmWETH	0.00000
AMPL	21739.83824
BAL	74815.48914
BAT	33015.95973
BUSD	86287.81560
CRV	104106.11557
DAI	285831.06911
ENJ	72642.57824
GUSD	54016.82953
KNC	43533.00019
LINK	168380.99214
MANA	52437.83840
MKR	109391.06866
PAX	759194.21381
RAI	110328.02862
REN	86000.27362
RENFIL	104352.92464
SNX	309288.28960
SUSD	221874.51600
TUSD	138668.60380
UNI	208502.19388
USDC	366763.83530
USDT	259447.94696
WBTC	464532.30579
WETH	0.00000
XSUSHI	430386.33101
YFI	579352.73415
ZRX	12475.06871

I think one of potential interesting topics to consider in the future would be liquidation prediction. However, after looking into the data, I have so thoughts that we would have to address before going into that.

1. What events do we consider to be liquidations? Since there is no liquidation ID or anything similar, we have to understand when liquidation occurs in data and whether some group of liquidations (in a short period of time for one user) is to be considered one liquidation or several.

2. Could we identify the borrower and liquidator for the same event? Could be tract users that liquidate a lot?

3. What data do we want to predict liquidations from? It seems like currently we do not even know what resource (and how much) is being used for collateral when borrowing. This data is absolutely crucial and it is probably impossible to predict anything without it.