

# DAR F21 Project Status Notebook:

DeFi

Jaimin Vyas (vyasj2)

10/31/2021

## Contents

<b>Biweekly work summary</b>	<b>1</b>
<b>Personal Contribution</b>	<b>1</b>
<b>Discussion of primary findings</b>	<b>1</b>
<b>Next Steps</b>	<b>11</b>

## Biweekly work summary

- RCS ID: vyasj2
- Project Name: IDEA-Blockchain
- Summary of work for last 2 weeks

## Personal Contribution

- GitHub commits on branch dar-vyasj2
  - Worked on general survival functions, as well as a little bit of further analysis on liquidations.
- In this notebook, I look a bit deeper into the impact of change in price on rate of liquidation
- Working on solving GitHub Issue #80: Survival approach at the user level

## Discussion of primary findings

- What did you want to know? I wanted to see the relationship between rate of liquidation and fluctuation of price, and look at individual user liquidations, and perhaps building off of Roman's work to further analyze these liquidations and what causes them, so that we can model the survival of a borrow better on those variables.
- How did you go about finding it? Getting the number of liquidations every day, finding the most statistically significant liquidations by any one specific user, and thinking about how to combine with Roman's work.
- What did you find?

To begin, we install the necessary packages:

```
if (!require("survival")) {  
  install.packages("survival")  
  library(survival)  
}
```

```

## Loading required package: survival
if (!require("dplyr")) {
  install.packages("dplyr")
  library(dplyr)
}

## Loading required package: 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
if (!require("survminer")) {
  install.packages("survminer")
  library(survminer)
}

## Loading required package: survminer
## Loading required package: ggplot2
## Loading required package: ggpubr
##
## Attaching package: 'survminer'
## The following object is masked from 'package:survival':
##
##   myeloma
if (!require("ggplot2")) {
  install.packages("ggplot2")
  library(ggplot2)
}
if (!require("readr")) {
  install.packages("readr")
  library(readr)
}

## Loading required package: readr
if (!require("tidyr")) {
  install.packages("tidyr")
  library(tidyr)
}

## Loading required package: tidyr
if (!require("stringr")) {
  install.packages("stringr")
  library(stringr)
}

## Loading required package: stringr

```

```

if (!require(lubridate)) {
  install.packages("lubridate")
  library("lubridate")
}

```

```
## Loading required package: lubridate
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1
```

```
##
```

```
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      date, intersect, setdiff, union
```

Then, we load the data:

```
df <- read_rds("../Data/transactions2.rds")
```

```
df <- df %>%
```

```
  dplyr::mutate(date=as_datetime(timestamp))
```

```
head(df)
```

```
##      amount borrowRate borrowRateMode  onBehalfOf      pool reserve
## 1      15.00  0.2590658      Variable 1.117217e+48 1.034668e+48  WETH
## 2    41501.63  6.2749368      Variable 8.502518e+47 1.034668e+48  DAI
## 3 7000000.00  2.5896280      Variable 4.635974e+47 1.034668e+48  USDT
## 4    15000.00  8.8025409      Variable 3.735263e+47 1.034668e+48  USDC
## 5     8193.19 48.7470516      Stable 6.896232e+47 1.034668e+48  USDC
## 6    11000.00  3.2250550      Variable 1.089455e+48 1.034668e+48  USDT
##      timestamp      user  type reservePriceETH reservePriceUSD amountUSD
## 1 1633275840 1.168069e+48 borrow  1.0000000000      3421.8708189  51328.06
## 2 1621340435 8.502518e+47 borrow  0.0002852900      0.9948044    41286.00
## 3 1622477822 4.635974e+47 borrow  0.0003812835      1.0000000 7000000.00
## 4 1619775984 3.735263e+47 borrow  0.0003611000      1.0043389   15065.08
## 5 1615481632 6.896232e+47 borrow  0.0005562201      0.9993909    8188.20
## 6 1626914745 1.089455e+48 borrow  0.0004971100      1.0000000   11000.00
##      collateralAmount collateralReserve liquidator principalAmount
## 1                NA                NA                NA
## 2                NA                NA                NA
## 3                NA                NA                NA
## 4                NA                NA                NA
## 5                NA                NA                NA
## 6                NA                NA                NA
##      principalReserve reservePriceETHPrincipal reservePriceUSDPrincipal
## 1                NA                NA                NA
## 2                NA                NA                NA
## 3                NA                NA                NA
## 4                NA                NA                NA
## 5                NA                NA                NA
## 6                NA                NA                NA
##      reservePriceETHCollateral reservePriceUSDCollateral amountUSDPrincipal
## 1                NA                NA                NA
## 2                NA                NA                NA
```

```
## 3          NA          NA          NA
## 4          NA          NA          NA
## 5          NA          NA          NA
## 6          NA          NA          NA
## amountUSDCollateral borrowRateModeFrom borrowRateModeTo stableBorrowRate
## 1          NA          NA          NA
## 2          NA          NA          NA
## 3          NA          NA          NA
## 4          NA          NA          NA
## 5          NA          NA          NA
## 6          NA          NA          NA
## variableBorrowRate fromState toState protocolContract user_alias
## 1          NA          True Gladys Marquez
## 2          NA          False Angel Prather
## 3          NA          False Jack Crowley
## 4          NA          False Jim Dickens
## 5          NA          False Leonard Reyes
## 6          NA          False Jill Carn
## onBehalfOf_alias          datetime          date
## 1 Evelyn Terrazas 2021-10-03 15:44:00 2021-10-03 15:44:00
## 2   Angel Prather 2021-05-18 12:20:35 2021-05-18 12:20:35
## 3   Jack Crowley 2021-05-31 16:17:02 2021-05-31 16:17:02
## 4    Jim Dickens 2021-04-30 09:46:24 2021-04-30 09:46:24
## 5   Leonard Reyes 2021-03-11 16:53:52 2021-03-11 16:53:52
## 6     Jill Carn 2021-07-22 00:45:45 2021-07-22 00:45:45
```

We first want to get all of the prices for each reserve and the date at which that price was recorded.

```
reservePrices <- df %>%
  dplyr::select(reserve,reservePriceUSD,date)

head(reservePrices)
```

```
## reserve reservePriceUSD          date
## 1   WETH      3421.8708189 2021-10-03 15:44:00
## 2    DAI       0.9948044 2021-05-18 12:20:35
## 3   USDT      1.0000000 2021-05-31 16:17:02
## 4   USDC      1.0043389 2021-04-30 09:46:24
## 5   USDC      0.9993909 2021-03-11 16:53:52
## 6   USDT      1.0000000 2021-07-22 00:45:45
```

Now we filter these prices so that we get only WETH prices, and add a rolling average and rolling standard deviation with a 21-day window. This is just so that we can see the fluctuations in price, as well as an average and standard deviation of how the price changes over time.

```
wethPrice <- reservePrices %>%
  dplyr::filter(reserve=="WETH") %>%
  dplyr::mutate(sdPrice=roll::roll_sd(reservePriceUSD,width=21)) %>%
  dplyr::mutate(avgPrice=roll::roll_mean(reservePriceUSD,width=21))

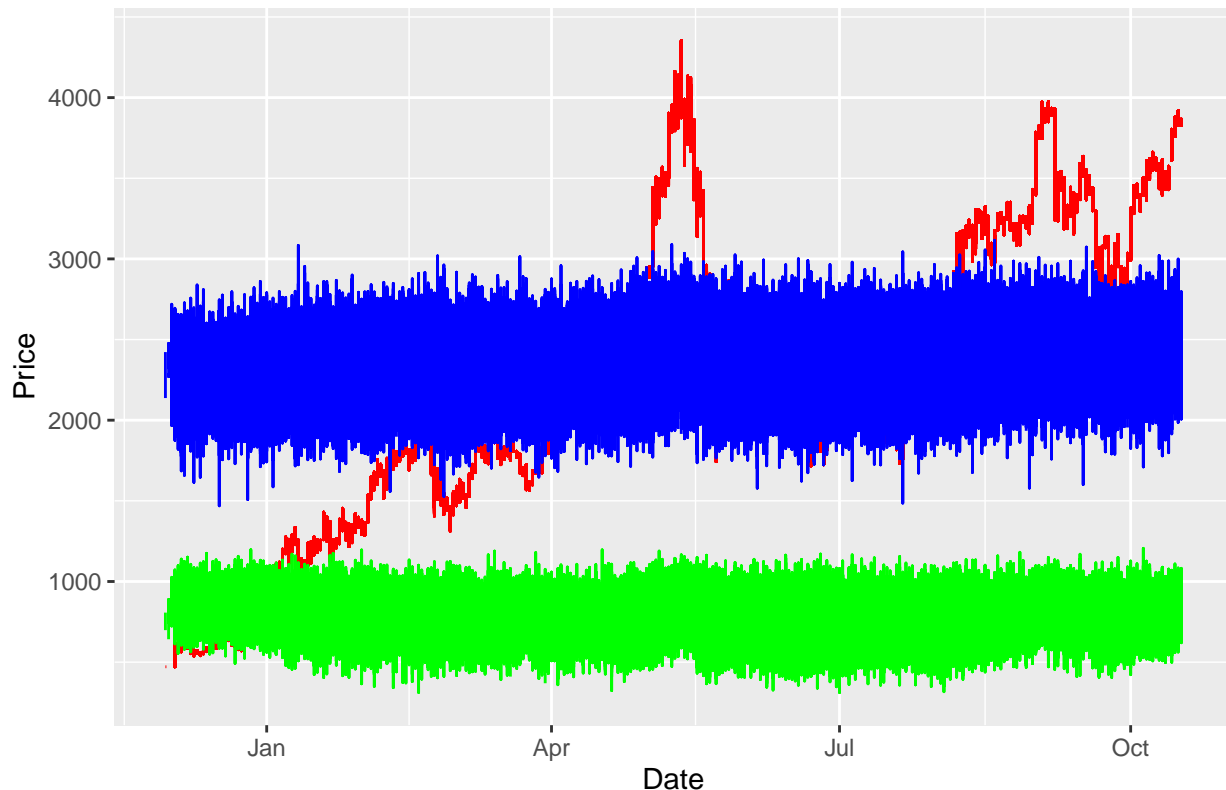
ggplot(wethPrice,aes(x=as.Date(date))) +
  geom_line(aes(y=reservePriceUSD), colour="red") +
  geom_line(aes(y=avgPrice), colour="blue") +
  geom_line(aes(y=sdPrice), colour="green") +
  ggtitle("WETH Price (red), 21-Day Rolling Avg (blue), and 21-Day Rolling StDev (green)") +
  labs(x="Date", y="Price", colour="Legend")
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

WETH Price (red), 21-Day Rolling Avg (blue), and 21-Day Rolling StDev (green)



It's immediately visible that the price of WETH is very highly volatile, because the rolling average and standard deviation seem to just be zigzags going back and forth. It is also clear to see that because the red line has a lot of hills and troughs. We won't use the rolling average or standard deviation going forward, because it doesn't seem to show anything that is valuable, other than what price they are centered around. At this point, I was wondering how the number of liquidations was impacted by this high volatility. I was also curious to see what the difference in liquidation price and borrow price was over time, because seeing the differences in prices on certain dates could provide more insight into what kinds of events impact liquidation, since if there is a low difference, then that means the price of WETH when liquidation occurred was not too far off from the initial borrow price. We can try and perform some interesting analyses on the collateral reserve when this happens.

```
borrows <- df %>%
  dplyr::filter(type=="borrow")

liquidations <- df %>%
  dplyr::filter(type=="liquidation")

wethLiquids <- liquidations %>%
  dplyr::filter(principalReserve=="WETH") %>%
  dplyr::select(user_alias,date,timestamp,type,liquidator,collateralAmount,collateralReserve,reservePri
  dplyr::rename(user=user_alias)
```

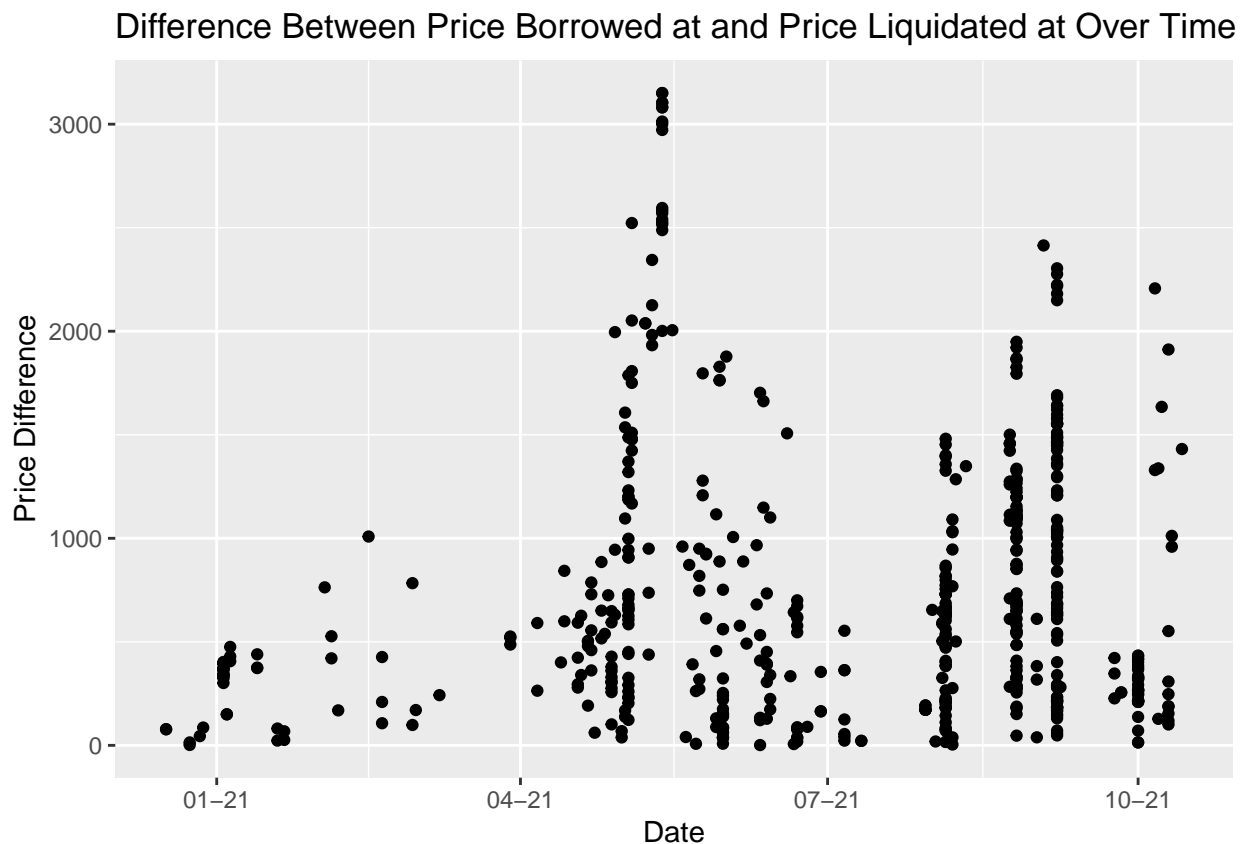
```

wethBorrows <- borrows %>%
  dplyr::filter(reserve=="WETH") %>%
  dplyr::select(onBehalfOf_alias,date,reservePriceUSD,timestamp,borrowRateMode,type) %>%
  dplyr::rename(user=onBehalfOf_alias)

wethBorrowLiquids <- left_join(wethBorrows,wethLiquids,by="user") %>%
  dplyr::rename(borrowDate=date.x) %>%
  dplyr::rename(liquidationDate=date.y) %>%
  dplyr::mutate(date=case_when(is.na(liquidationDate) ~ borrowDate, TRUE ~ liquidationDate)) %>%
  dplyr::mutate(priceDiff=reservePriceUSDPrincipal-reservePriceUSD) %>%
  dplyr::filter(priceDiff>0)

ggplot(wethBorrowLiquids,aes(x=as.Date(date),y=priceDiff)) +
  geom_point() +
  ggtitle("Difference Between Price Borrowed at and Price Liquidated at Over Time") +
  scale_x_date(date_labels="%m-%y") +
  xlab("Date") +
  ylab("Price Difference")

```



We can see that it seems to be fairly sporadic, but there are some clumps where there are a lot of low difference in prices, obviously at the China crypto ban date in May/June, as well as later on around August/September. If we plot this as a line graph and overlay it on the price of WETH, it might provide something more interesting.

```

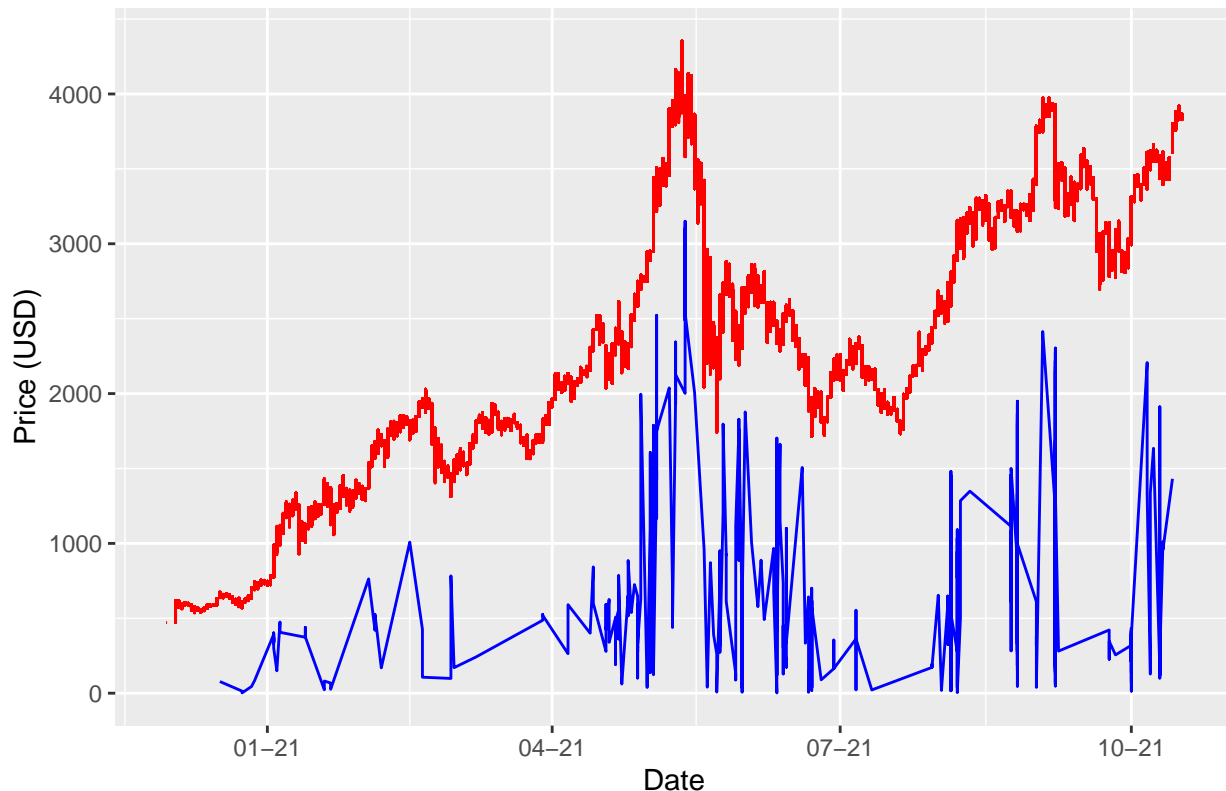
ggplot() +
  geom_line(data=wethPrice, aes(x=as.Date(date),y=reservePriceUSD), colour = "red") +
  geom_line(data=wethBorrowLiquids, aes(x=as.Date(date),y=priceDiff), colour = "blue") +
  scale_x_date(date_labels="%m-%y") +

```

```
xlab("Date") +
ylab("Price (USD)") +
ggtitle("Difference in Price Borrowed at and Price Liquidated at (Blue) and WETH Price Over Time (Red)")
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

### Difference in Price Borrowed at and Price Liquidated at (Blue) and WETH



Here we can see that the difference in price of reserve between borrow and liquidation somewhat seems to model the fluctuation of the price of WETH itself, which makes sense. The gap in between the two lines is the price that the specific loan was liquidated at. As we can see, sometimes that price is very high, and sometimes it is much lower. Now I started wondering about the impact of price on the number of liquidations. My prediction was that when the price of WETH dropped very low, there would be a spike in liquidations.

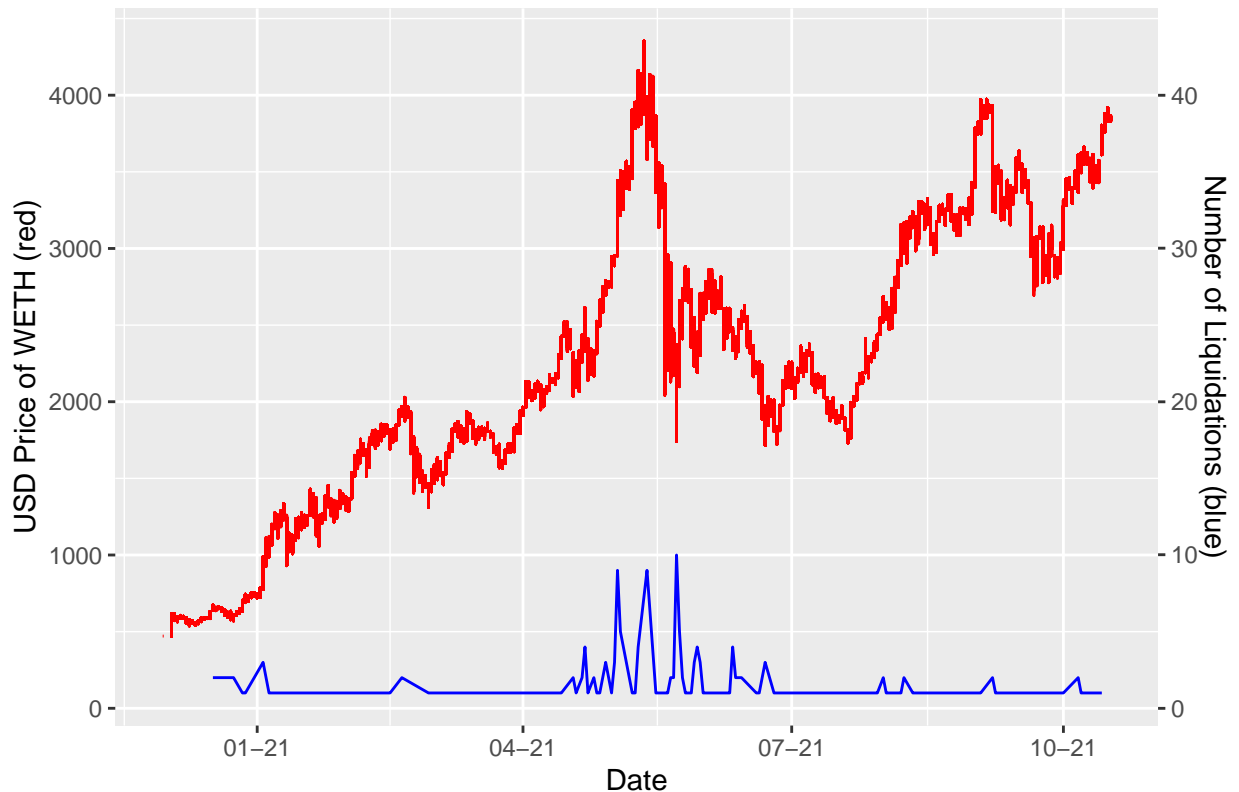
```
wethLiquidDateCounts <- wethLiquids %>%
  dplyr::group_by(as.Date(date)) %>%
  dplyr::summarise(date, n=n())
```

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

```
ggplot() +
  geom_line(data=wethPrice, aes(x=as.Date(date), y=reservePriceUSD), colour="red") +
  geom_line(data=wethLiquidDateCounts, aes(x=as.Date(date), y=100*n), colour="blue",) +
  scale_x_date(date_labels="%m-%y") +
  scale_y_continuous("USD Price of WETH (red)", sec.axis=sec_axis(~ . * 0.01, name = "Number of Liquidations"))
xlab("Date") +
ggtitle("Price of WETH and Number of Liquidations by Date")
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

Price of WETH and Number of Liquidations by Date



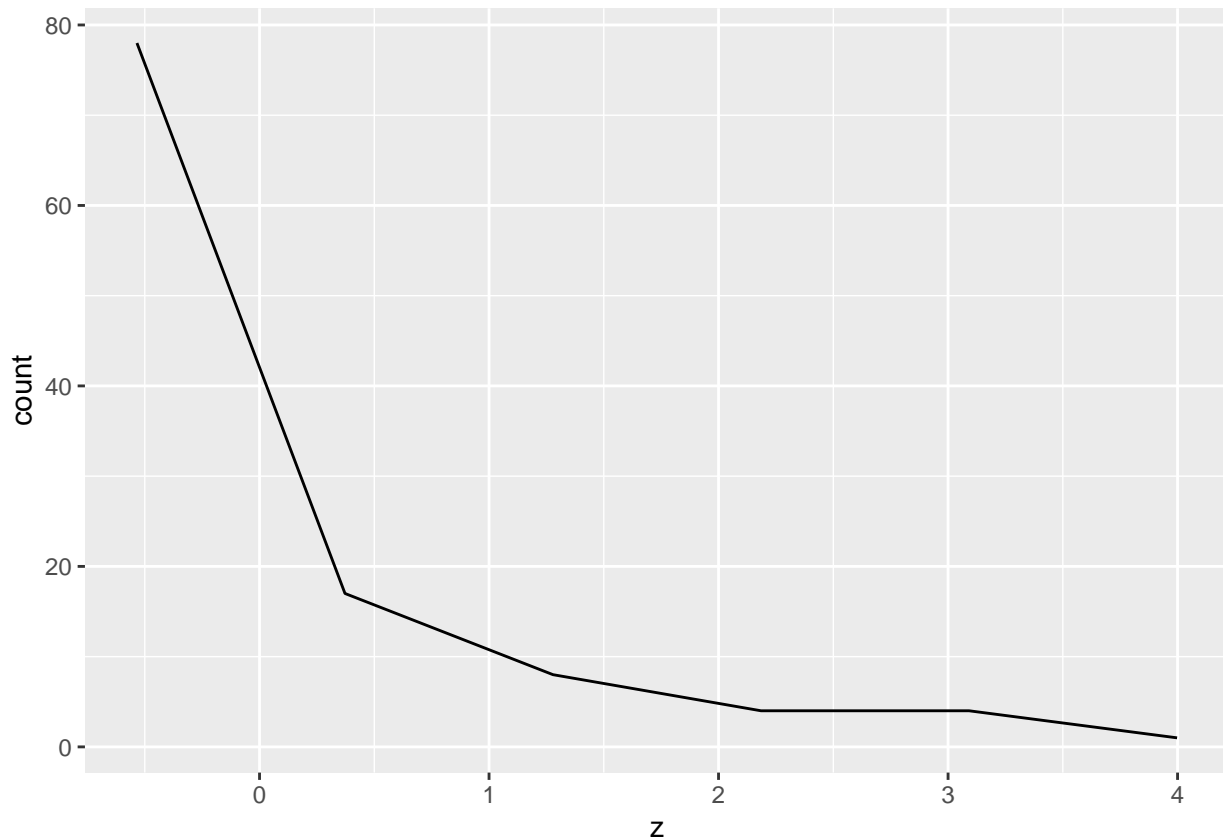
As we can see, the graph above confirms this hypothesis. There were more liquidations at the peak price, but that is most likely because of the China crypto ban that occurred around the same time. This further shows the impact of world events on cryptocurrency prices. Now I started thinking about individual users' liquidations, and I decided to standardize the number of liquidations each user had, so that we could find statistically significant liquidations, and analyze the characteristics about those loans so that we might get a better idea about what causes liquidations and what factors to consider when making a borrow.

```
wethLiquidUserCounts <- wethLiquids %>%
  dplyr::count(user) %>%
  dplyr::arrange(n)

wethLiquidUserCounts <- wethLiquidUserCounts %>%
  dplyr::arrange(n) %>%
  dplyr::mutate(z=(n-mean(wethLiquidUserCounts$n))/sd(wethLiquidUserCounts$n))

ggplot(wethLiquidUserCounts, aes(x=z)) + geom_line(stat = "count")
```





```
stat.sig <- wethLiquidUserCounts %>%
  dplyr::filter(z>=2)

stat.sig
```

```
##           user n      z
## 1 Anthony Hines 4 2.185084
## 2 Brenda Thompson 4 2.185084
## 3 Debra Meinhart 4 2.185084
## 4 Holly Childs 4 2.185084
## 5 Bridget Hensley 5 3.091489
## 6 Ivan Brown 5 3.091489
## 7 Lester Hines 5 3.091489
## 8 Richard Garnett 5 3.091489
## 9 Odis Wight 6 3.997894
```

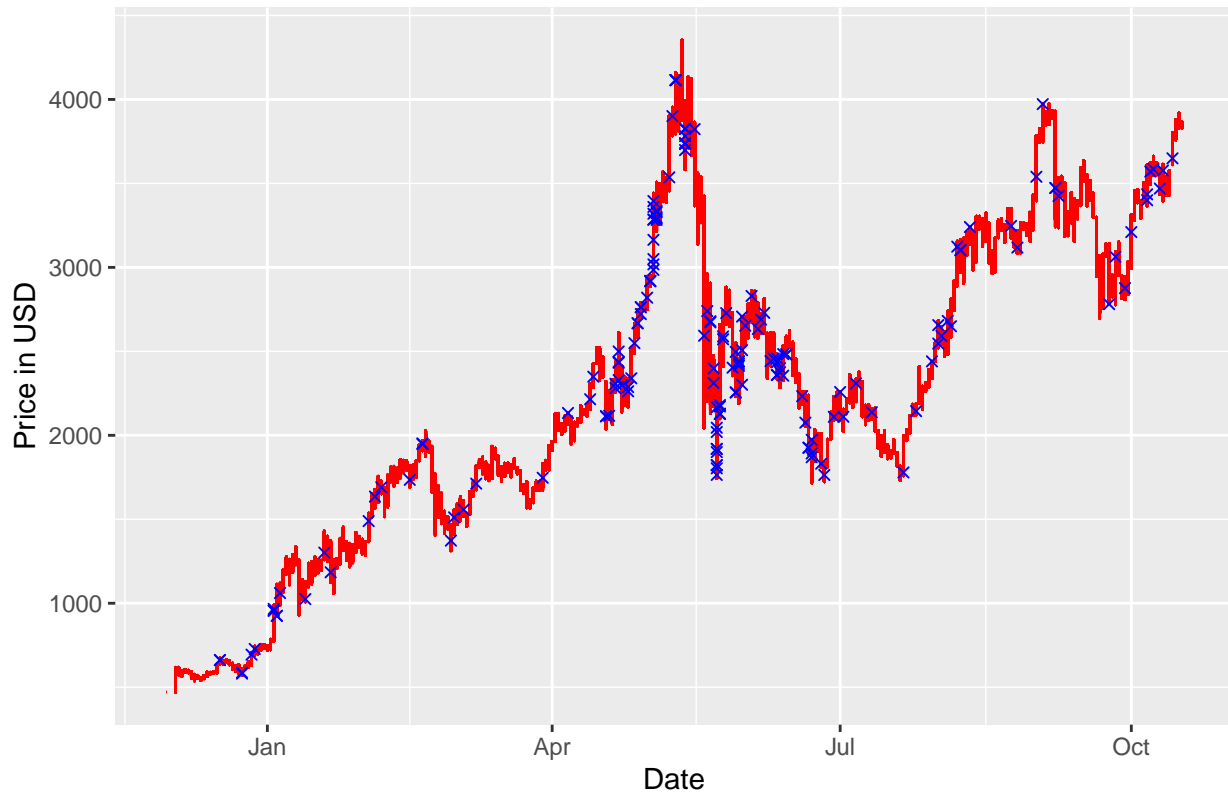
The dataframe above contains the names of users who had a statistically significant number of liquidations using WETH as the principal, and now we will graph the prices along the price curve of WETH.

```
wethLiquids.sig <- wethLiquids %>%
  dplyr::filter(user %in% stat.sig$user)

ggplot() +
  geom_line(data=wethPrice, aes(x=as.Date(date),y=reservePriceUSD), colour="red") +
  geom_point(data=wethLiquids, aes(x=as.Date(date),y=reservePriceUSDPrincipal), colour="blue", shape=4)
  xlab("Date") +
  ylab("Price in USD") +
  ggtitle("Price at Which Liquidation Occurred")
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

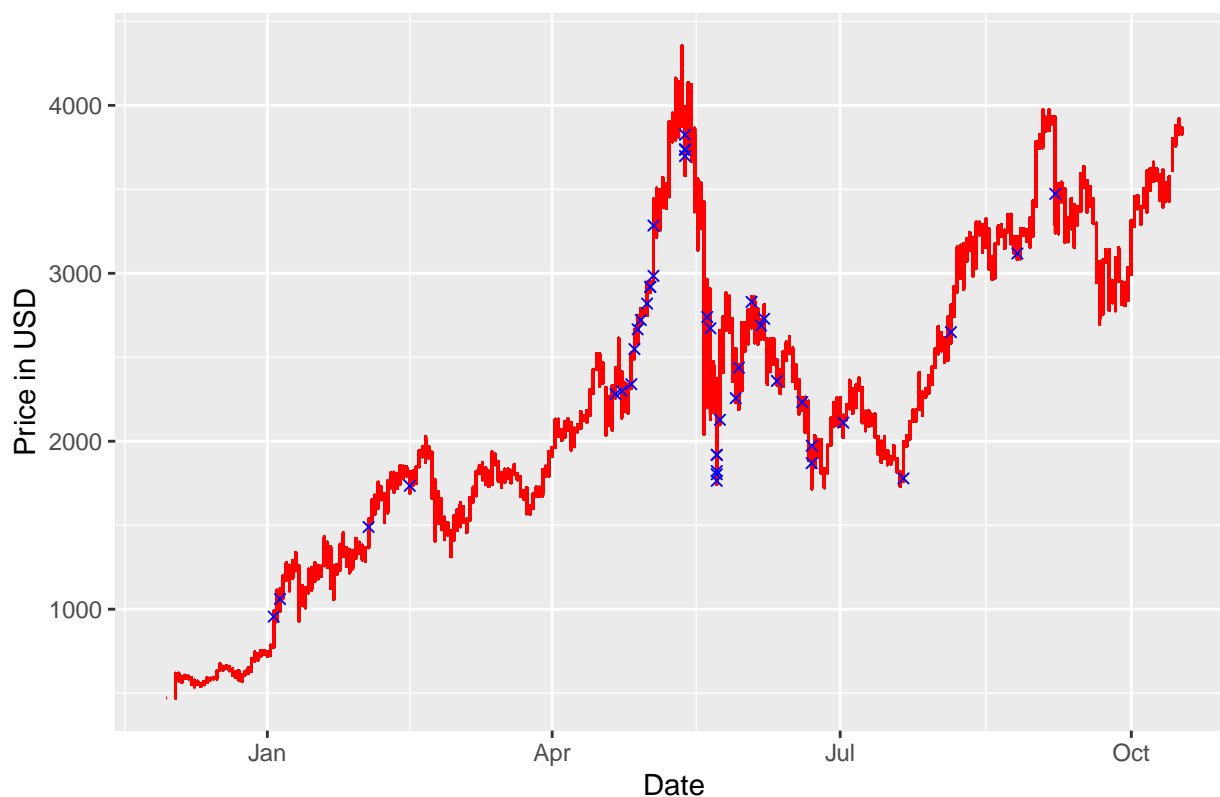
### Price at Which Liquidation Occurred



```
ggplot() +  
  geom_line(data=wethPrice, aes(x=as.Date(date),y=reservePriceUSD), colour="red") +  
  geom_point(data=wethLiquids.sig, aes(x=as.Date(date),y=reservePriceUSDPrincipal), colour="blue", shape="x") +  
  xlab("Date") +  
  ylab("Price in USD") +  
  ggtitle("Price at Which Statistically Significant Liquidation Occurred")
```

```
## Warning: Removed 13 row(s) containing missing values (geom_path).
```

## Price at Which Statistically Significant Liquidation Occurred



As can be seen above, the first graph shows all liquidations that had WETH as the principal reserve borrowed, and the second graph showed all statistically significant liquidations, which was calculated as a liquidation that occurred to a user that had a number of liquidations that was within the top 95% of liquidated users. We can see that these statistically significant liquidations occur at the low points of the price curve, meaning that as the price drops, your probability of liquidating is much higher. This can be further modeled with a survival curve.

## Next Steps

My next steps are to dive deeper into the story of WETH and how it is transacted on the AAVE platform, as well as perform more survival analysis so that users who utilize the app can have the best chance of performing a safe borrow and not have to worry about liquidation.