DAR F21 Project Status Notebook Assignment 3

DeFi

Jaimin Vyas

10-01-2021

Contents


```
#import libraries
if (!require("ggplot2")) {
  install.packages("ggplot2")
  library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

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

```
## Loading required package: knitr
```

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

```
## Loading required package: RColorBrewer
```

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

```
## Loading required package: beeswarm
```

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

```
## Loading required package: tidyverse
```

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

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

```
## Loading required package: ggbeeswarm
```

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

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
## Attaching package: 'xts'
## The following objects are masked from 'package:dplyr':
##
##     first, last
```

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

```
## Loading required package: devtools
## Loading required package: usethis
```

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

```
## Loading required package: ggfortify
```

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

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

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

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

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

```
## Loading required package: survival
```

```r
raw_df <- read_rds('../../Data/transactions.Rds')
```

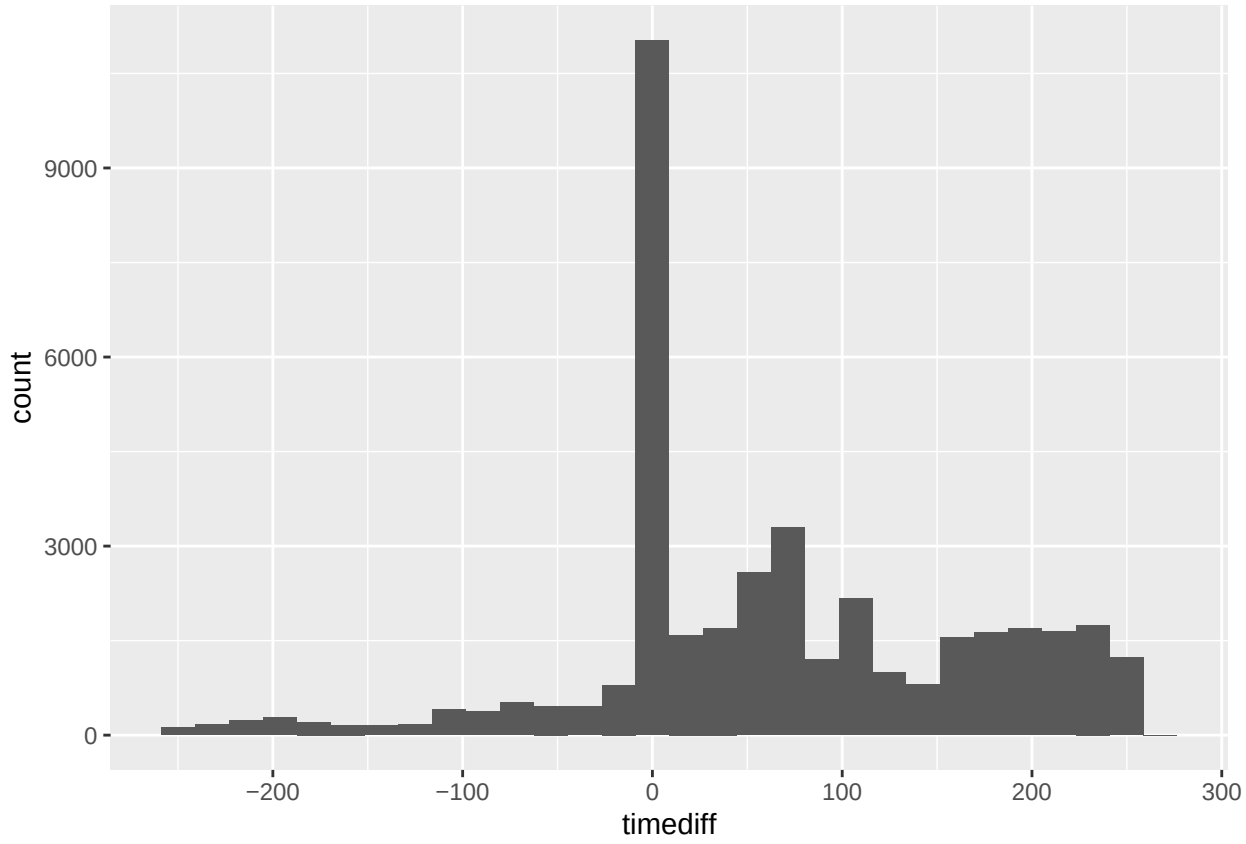
```r
diffs <- raw_df %>%
  filter(type=="borrow" | type=="deposit") %>%
  select(timestamp,user,type,amountUSD) %>%
  arrange(user,timestamp)
m <- max(diffs$timestamp)

final <- data.frame(user=numeric(0),timediff=numeric(0),status=numeric(0))

for (u in unique(diffs$user)) {
 x <- diffs[diffs$user==u,]
 minBorrow <- m
 minDeposit <- m
 for (row in 1:nrow(x)) {
   if (minBorrow > x[row,"timestamp"] & x[row,"type"] == "borrow") {
     minBorrow <- x[row,"timestamp"]
   }
   if (minDeposit > x[row,"timestamp"] & x[row,"type"] == "deposit") {
     minDeposit <- x[row,"timestamp"]
   }
 }
 if (minBorrow-minDeposit == 0) {
   final[nrow(final)+1,] <- c(u,(minBorrow-minDeposit)/86400,0)
 } else{
   final[nrow(final)+1,] <- c(u,(minBorrow-minDeposit)/86400,1)
 }
}
head(final,20)
```

```
##            user      timediff status
## 1  2.577533e+33 155.016284722      1
## 2  6.663597e+34   0.002604167      1
## 3  4.867107e+35 115.682881944      1
## 4  8.009427e+35 256.591469907      1
## 5  1.358443e+37 -24.209710648      1
```

```
## 6  8.092096e+37 144.050729167      1
## 7  1.976226e+38 186.344756944      1
## 8  1.537845e+39 141.453634259      1
## 9  3.732290e+40   0.243680556      1
## 10 5.816491e+40 105.139351852      1
## 11 2.154573e+42  61.445162037      1
## 12 4.516334e+43   0.035034722      1
## 13 6.257108e+43  39.494189815      1
## 14 6.293512e+43  73.425509259      1
## 15 8.891014e+43   0.000000000      0
## 16 1.293605e+44 179.339560185      1
## 17 1.320123e+44 183.227233796      1
## 18 1.325103e+44  83.602916667      1
## 19 1.417418e+44  53.337685185      1
## 20 1.464170e+44 177.928726852      1
```

```
ggplot(final, aes(x=timediff)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
km <- with(final, Surv(timediff, status))
head(km,80)
```

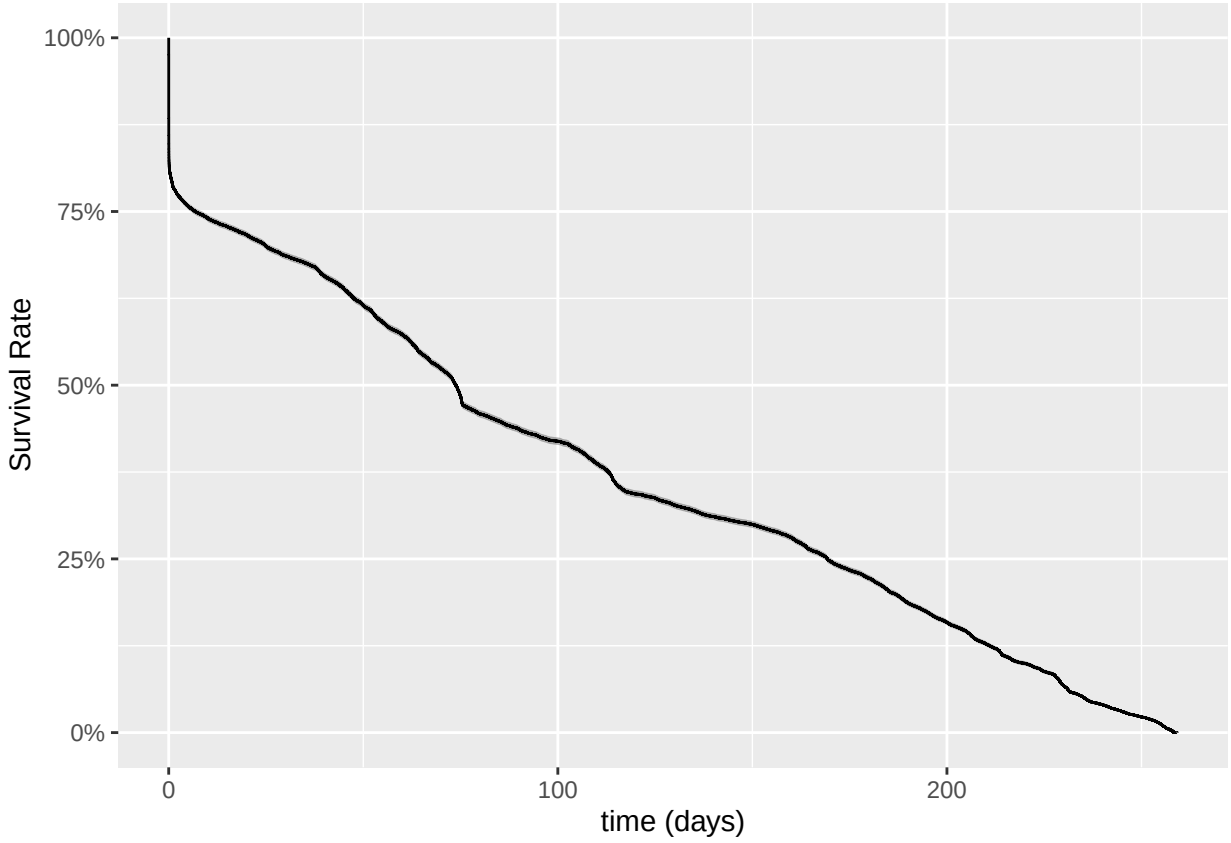
```
##  [1]  1.550163e+02   2.604167e-03   1.156829e+02   2.565915e+02  -2.420971e+01
##  [6]  1.440507e+02   1.863448e+02   1.414536e+02   2.436806e-01   1.051394e+02
## [11]  6.144516e+01   3.503472e-02   3.949419e+01   7.342551e+01   0.000000e+00+
## [16]  1.793396e+02   1.832272e+02   8.360292e+01   5.333769e+01   1.779287e+02
## [21]  2.335535e+02  -2.593657e+00  -1.048012e+02   7.664352e-01  -1.613086e+02
## [26]  4.650992e+01   9.375000e-04   1.541427e+02   6.759259e-03   6.680459e+01
```

```
## [31]  1.399023e+02   6.670035e+00  -1.696775e+01   1.599259e+02   6.389140e+01
## [36]  3.006274e+01   0.000000e+00+  1.141170e+02   1.118228e+02   5.081019e-03
## [41] -9.308958e+00   4.919877e+01   7.222222e-03   1.682500e+02   7.332847e+01
## [46]  2.234954e-02   2.074618e+02   4.371795e+01   6.395197e+01   1.207642e+02
## [51]  1.687727e+02   6.595439e+01   2.483977e+01   1.771014e+02  -5.080716e+01
## [56]  2.662037e-03   6.269024e+01   2.182824e+02   4.204666e+01   7.335794e+01
## [61]  6.025463e-02   6.301696e+01   2.077414e+01   2.410880e-02   8.272767e+01
## [66]  1.694374e+02   0.000000e+00+  1.870688e+02   1.626308e+02  -6.369443e+01
## [71]  1.157944e+02   2.637438e+01   1.626297e+02   1.844907e-02   2.285629e+02
## [76]  2.492310e+01  -7.657712e+01   2.418981e-02  -2.251931e+02   1.147275e+02
```

```
final <- final %>%
  filter(timediff > 0)
km_fit <- survfit(Surv(timediff, status) ~ 1, data=final)
summary(km_fit, times = c(1,30,60,90*(1:10)))
```

```
## Call: survfit(formula = Surv(timediff, status) ~ 1, data = final)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1  25240    6783    0.788 0.00228        0.784        0.793
##    30  21974    3266    0.686 0.00259        0.681        0.691
##    60  18347    3627    0.573 0.00276        0.568        0.578
##    90  13991    4356    0.437 0.00277        0.432        0.442
##   180   7111    6880    0.222 0.00232        0.218        0.227
```

```
autoplot(km_fit,xlab="time (days)",ylab="Survival Rate",title="Survival Rate of Deposit before the firs
```

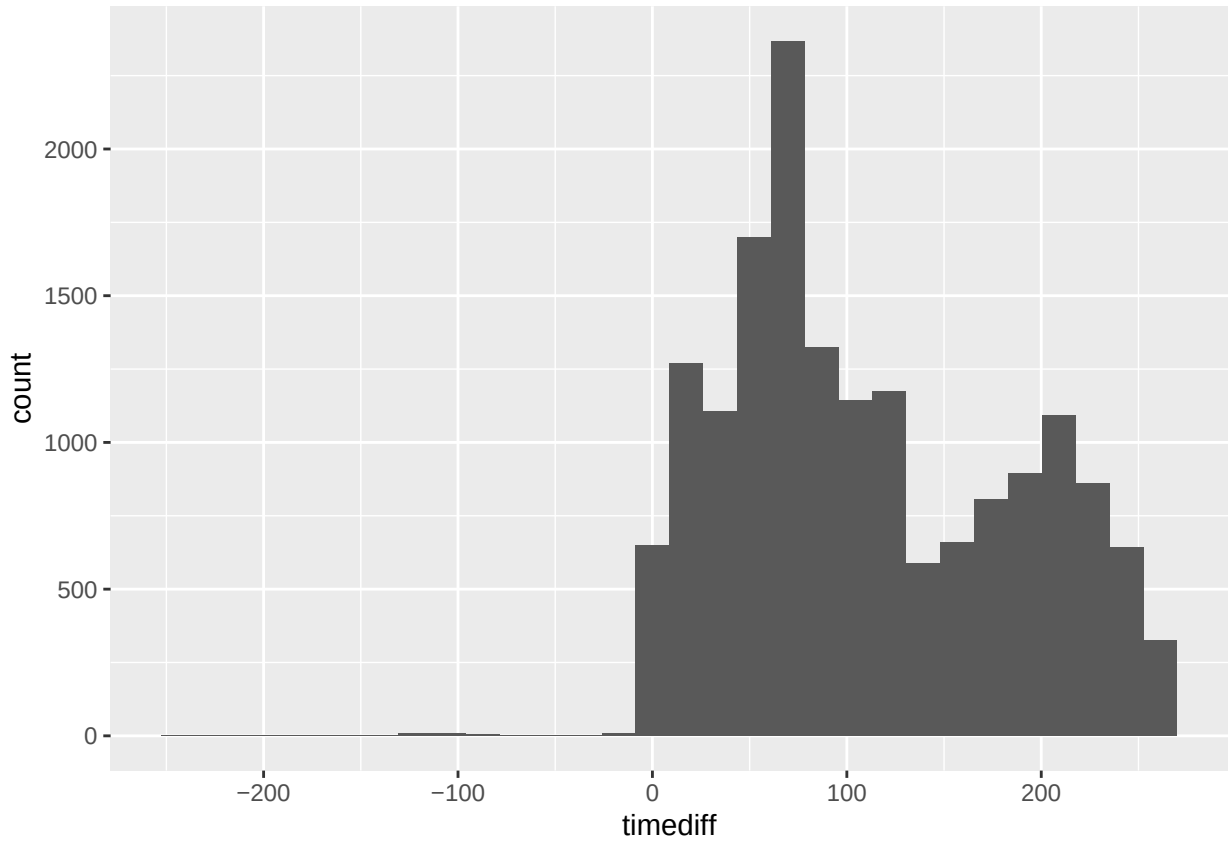


```
diffs <- raw_df %>%
  filter(type=="liquidation" | type=="borrow") %>%
  select(timestamp,user,type) %>%
  arrange(user,timestamp)
m <- max(diffs$timestamp)
final <- data.frame(user=numeric(0),timediff=numeric(0),status=numeric(0))

for (u in unique(diffs$user)) {
  x <- diffs[diffs$user==u,]
  minBorrow <- m
  minLiquidation <- m
  for (row in 1:nrow(x)) {
    if (minBorrow > x[row,"timestamp"] & x[row,"type"] == "borrow") {
      minBorrow <- x[row,"timestamp"]
    }
    if (minLiquidation > x[row,"timestamp"] & x[row,"type"] == "liquidation") {
      minLiquidation <- x[row,"timestamp"]
    }
  }
  if (minBorrow-minLiquidation == 0) {
   final[nrow(final)+1,] <- c(u,(minLiquidation-minBorrow)/86400,0)
 } else{
   final[nrow(final)+1,] <- c(u,(minLiquidation-minBorrow)/86400,1)
 }
}
head(final,20)
```

```
##            user  timediff status
## 1  6.663597e+34  81.55281      1
## 2  1.358443e+37  24.53483      1
## 3  3.732290e+40  96.49073      1
## 4  4.516334e+43 253.69304      1
## 5  8.891014e+43  60.64589      1
## 6  1.325103e+44  37.51581      1
## 7  1.620776e+44   9.28772      1
## 8  1.784060e+44 105.12630      1
## 9  1.971883e+44 177.18818      1
## 10 2.237597e+44 161.63376      1
## 11 2.377097e+44 212.36407      1
## 12 2.594664e+44 211.55412      1
## 13 3.447318e+44 157.87262      1
## 14 3.455863e+44 185.28301      1
## 15 4.342087e+44 110.88876      1
## 16 4.654995e+44  70.06772      1
## 17 4.811595e+44  55.91792      1
## 18 5.885460e+44  12.53829      1
## 19 7.417449e+44  61.02763      1
## 20 9.549468e+44 113.00664      1
```

```
ggplot(final, aes(x=timediff)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
km <- with(final, Surv(timediff, status))
head(km,80)
```

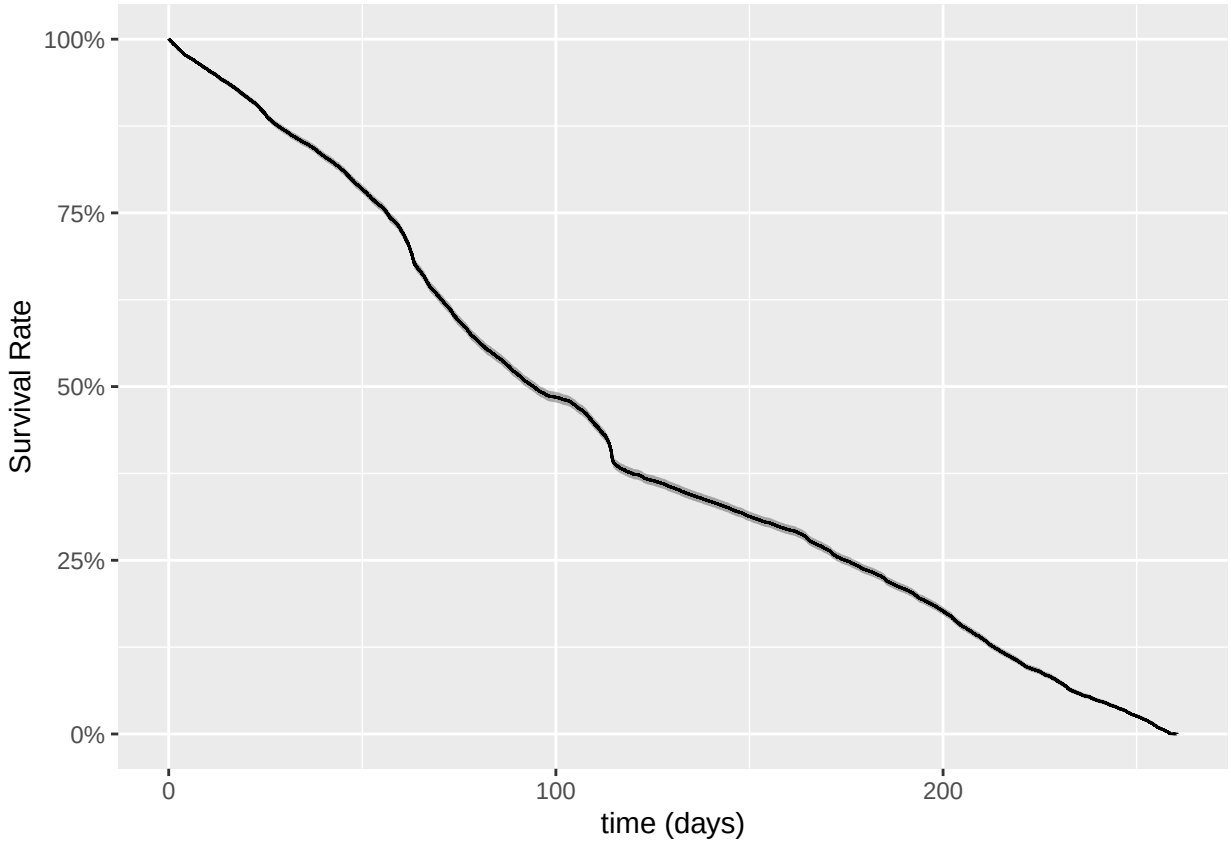
```
##  [1]  81.5528125  24.5348264  96.4907292 253.6930440  60.6458912  37.5158102
##  [7]   9.2877199 105.1262963 177.1881829 161.6337616 212.3640741 211.5541204
## [13] 157.8726157 185.2830093 110.8887616  70.0677199  55.9179167  12.5382870
## [19]  61.0276273 113.0066435  51.1322801 152.9995370 168.6531481 203.4557870
## [25]  61.4720949  64.0195486 231.1445139  76.9022338 153.9932755 225.5182407
## [31] 178.2950579  77.4698958 143.7440972  37.4541088 138.5470255 225.1589815
## [37]  78.6579282  63.3206019  50.3648727 172.0033912 253.0051157  14.0712384
## [43] 193.4203472 129.9742245 150.3072685  61.2018866  29.1796065  66.1415972
## [49]  45.8292245 257.1030556  -4.0091204  31.0889352  49.6852431 231.8342708
## [55]   0.7242708 107.3215162 129.6490162 162.2153819 107.5789236 177.0586343
## [61]  49.1623380  77.5523032 117.9858796  60.6741551 248.0293171 112.6926273
## [67]  63.8197338 155.6355903  22.2368171  35.8322454  33.7519329 214.2147454
## [73] 207.6211458  20.6350463  15.5826042 122.4982986 235.9735301 117.0007176
## [79] 202.4057176  89.9509491
```

```
final <- final %>%
  filter(timediff > 0)
km_fit <- survfit(Surv(timediff, status) ~ 1, data=final)
summary(km_fit, times = c(1,30,60,90*(1:10)))
```

```
## Call: survfit(formula = Surv(timediff, status) ~ 1, data = final)
##
##  time n.risk n.event survival  std.err lower 95% CI upper 95% CI
##     1  16509      89    0.995 0.000567        0.994        0.996
##    30  14414    2095    0.868 0.002624        0.863        0.874
```

```
##     60  12030     2384    0.725 0.003467        0.718        0.732
##     90   8607     3423    0.519 0.003878        0.511        0.526
##    180   3922     4685    0.236 0.003297        0.230        0.243
```

```
autoplot(km_fit,xlab="time (days)",ylab="Survival Rate",title="Survival Rate of Borrow before the first
```



Weekly Work Summary

NOTE: Follow an outline format; use bullets to express individual points.

- RCS ID: vyasj2
- Project Name: DeFi
- Summary of work since last week
  - Describe the important aspects of what you worked on and accomplished
- Summary of github commits
  - Branch name: dar-vyasj2
  - https://github.rpi.edu/DataINCITE/IDEA-Blockchain/tree/dar-vyasj2
- List of presentations, papers, or other outputs
  - https://docs.google.com/presentation/d/1cBPMaj-quV0y45F9t3aXbeywlW4GOT8miq7E3wY-7d8/edit#slide=id.gf5261890f6_1_0
- All work was done by me, with help from https://rviews.rstudio.com/2017/09/25/survival-analysis-with-r/ for suvival analysis code

Personal Contribution

- The professors proposed the idea to use Survival Analysis to find the "survival rate" of deposits over time.

Discussion of Primary Findings

- Primary Findings:
  - I wanted to know how much time went by between a deposit and the first borrow by the same user, and a similar statistic for borrows and liquidations.
  - As is seen in the notebook, I manipulated the original dataframe and reduced it down to only borrows and deposits, or only borrows and liquidations. I used base R to do this, because I could only get so far using dplyr, and at some point had to use a for loop to iterate over every user's transactions. I plan to see if I can rewrite the code using dplyer, but I expect that I will still have to utilize base R for part of it.
  - I found that for borrows/deposits, around 20% of people borrow almost immediately after they deposit tokens into their account. After that, the curve seems to follow a fairly steady slope downward, and shows that nobody holds on to their deposit for more than 260 days, meaning that people are almost guaranteed to borrow after 2/3 of a year. This long discrepancy could be due to the fact that some people would forget about the money they deposited, or could account for yield farmers, people who deposit money to lend them out to other people, and collect interest based on that. For borrows/liquidations, I found that there was a fairly steady curve downward that ended at 260 days, just like the borrows/deposits survival curve. They both end at the same time most likely due to the fact that the data is recent, and the last timestamp would be the most recent datapoint.

Future Steps

- Discussion of next steps
  - For the next steps, I would like to take a bit of a deeper dive into the transactions and calculate on average how many transactions or how much time it takes to empty a single deposit fully, after how many borrows or how much time people liquidate, as well as the net amount transacted every day. I think this would be useful information to know because it describes user interactions, and provides insight on how careful people are with their transactions.