DeFi Example Notebook:

DAR Assignment 1 (Fall 2021)

Data Analytics Research Instructors

08/24/2021

Introductory Decentralized Finance (DeFi) Research Notebook

This notebook is broken into two main parts:

```
* Part 1 is a basic introduction to github and RStudio Server
* Part 2 is an introduction to the DeFi transaction dataset
```

This R Notebook and its related R scripts provide a very basic introduction to an interesting **Decentralized Finance (DeFi)** dataset. All data was obtained by querying an API on The Graph, an indexing protocol for querying networks like Ethereum, for transaction data based on
the AAVE protocol. For more information on AAVE see the AAVE developer notes. The AAVE protocol is based on Ethereum, an important cryptocurrency platform.

The RPI github repository for all the code required for this notebook, including a snapshots of AAVE transaction and user data, may be found at:

- https://github.rpi.edu/DataINCITE/IDEA-Blockchain

The IDEA-Blockchain github also contains notebooks used to harvest the AAVE dataset, which you are welcome to examine.

BEFORE YOU BEGIN

To contribute or submit to any RPI github repository you must validate your RPI github.com ID and send a confirmation email to John Erickson at `erickj4@rpi.edu`. Please do the following **now**:

- Browse to http://github.rpi.edu
- Login using your RPI credentials
- **PLEASE DO THIS IMMEDIATELY BEFORE READING ANY FURTHER!!**

DAR ASSIGNMENT 1: CLONING A NOTEBOOK AND UPDATING THE REPOSITORY

In this assignment we're asking you to...

- clone the `IDEA-Blockchain` github repository...
- create a personal branch using git, and...
- make additions to the repository by creating a new, customized notebook.

The instructions which follow explain how to accomplish this.

For DAR Fall 2021 you *must* be using RStudio Server on the IDEA Cluster. Instructions for accessing "The Cluster" appear at the end of this notebook. Don't forget to validate your RPI github ID as above and email `erickj4@rpi.edu`

Cloning an RPI github repository

The recommended procedure for cloning and using this repository is as follows:

- Access the RPI network via VPN
  - See https://itssc.rpi.edu/hc/en-us/articles/360008783172-VPN-Connection-and-Installation for information
- Access RStudio Server on the IDEA Cluster at http://lp01.idea.rpi.edu/rstudio-ose/
  - You must be on the RPI VPN!!
- Access the Linux shell on the IDEA Cluster by clicking the **Terminal** tab of RStudio Server (lower left panel).
  - You now see the Linux shell on the IDEA Cluster
  - `cd` (change directory) to enter your home directory using: `cd ~`
  - Type `pwd` to confirm
  - NOTE: Advanced users may use `ssh` to directly access the Linux shell from a macOS or Linux command line
- Type `git clone https://github.rpi.edu/DataINCITE/IDEA-Blockchain.git` from within your `home` directory
  - This will create a new directory `IDEA-Blockchain`
- In the Linux shell, `cd` to `IDEA-Blockchain/DefiResearch/StudentNotebooks`
  - Type `ls -al` to list the current contents
  - Don't be surprised if you see many files!
- In the Linux shell, type `git checkout -b dar-yourrcs` where `yourrcs` is your RCS id
  - For example, if your RCS is `erickj4`, your new branch should be `dar-erickj4`
  - It is *critical* that you include your RCS id in your branch id
- Now in the RStudio Server UI, navigate to the `IDEA-Blockchain/DefiResearch/StudentNotebooks` directory via the **Files** panel (lower right panel)
  - Under the **More** menu, set this to be your R working directory
- **REQUIRED FOR ASSIGMENT**:
  1. In RStudio, make a **copy* **of `blockchain-notebook-f21.Rmd` file using a *new, original, descriptive* filename that** includes your RCS ID!**
     - Open `blockchain-notebook-f21.Rmd`
     - Save As... using new filename
     - Example filename for user `erickj4`: `erickj4-assignment1-f21.Rmd`
  2. Edit your new notebook using RStudio and save
     - Change the `title:` and `subtitle:` headers (at the top of the file)
     - Change the `author:`
     - Change the `date:`
     - Save your changes
  3. Use the RStudio `Knit` command to create an HTML file; repeat as necessary
     - Use the down arrow next to the word `Knit` and select **Knit to HTML**
  4. In the Linux terminal, use `git add` to add each new file you want to add to the repository
     - Type: `git add yourfilename.Rmd`
     - Type: `git add yourfilename.html` (created when you knitted)
  5. When you're ready, in Linux commit your changes:
     - Type: `git commit -m "some comment"` where "some comment" is a useful comment describing your changes
  6. Finally, push your commits to the RPI github repo
     - Type: `git push origin dar-yourrcs` (where `dar-yourrcs` is the branch you've been working in)
  7. **REQUIRED:** On the RPI github, submit a pull request.
     - In a web browser, navigate to https://github.rpi.edu/DataINCITE/IDEA-Blockchain
     - In the branch selector drop-down (usually says "master"), select your branch
     - Submit a pull request for your branch

Confirm what you just did

For this assignment you will be asked to confirm the following in LMS:

```
* The location of the github: _Type repository URL here_
* Your github ID: _Type your RCS ID here_
* The name of your new branch: _Type your new branch name here_
* The name of your new (copied) notebook: _Type your new notebook name here_
```

Please also see this handy github "cheatsheet": https://education.github.com/git-cheat-sheet-education.pdf

Exploring a DeFi Transaction Dataset

Prepare Transaction Data

We begin by loading in the data to a csv file. The dataset has over 400,000 rows, and 27 columns.

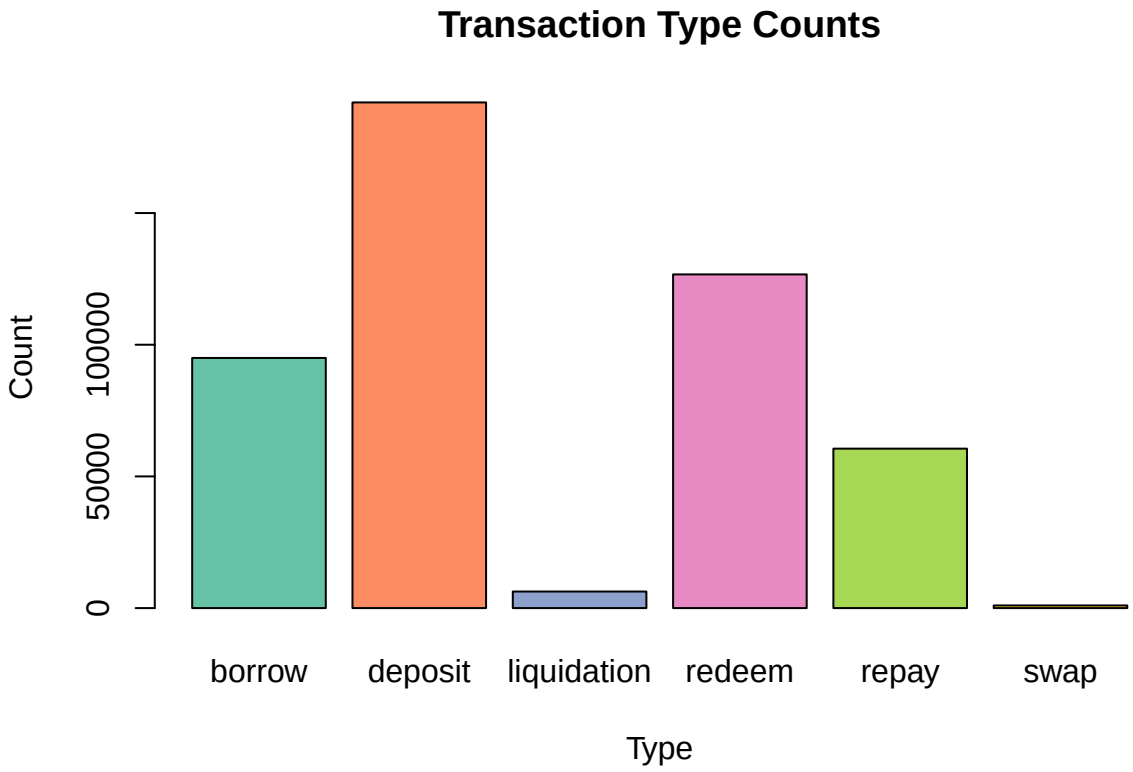
```
#load Rds (binary version of csv file) into dataframe
df<-read_rds('../Data/transactions.Rds')

# Let's take a look
head(df)
```

```
## # A tibble: 6 x 26
##   amount borrowRate borrowRateMode onBehalfOf pool  reserve timestamp user
##    <dbl>      <dbl> <chr>          <chr>      <chr> <chr>       <dbl> <chr>
## 1 4.15e4       6.27 Variable       0x94ee9c6~ 0xb5~ DAI        1.62e9 0x94~
## 2 7.00e6       2.59 Variable       0x51346d3~ 0xb5~ USDT       1.62e9 0x51~
## 3 1.50e4       8.80 Variable       0x416d7f3~ 0xb5~ USDC       1.62e9 0x41~
## 4 8.19e3      48.7  Stable         0x78cbc5e~ 0xb5~ USDC       1.62e9 0x78~
## 5 1.10e4       3.23 Variable       0xbed4dbd~ 0xb5~ USDT       1.63e9 0xbe~
## 6 4.00e4       5.74 Variable       0x2627ffc~ 0xb5~ USDT       1.62e9 0x26~
## # ... with 18 more variables: type <chr>, reservePriceETH <dbl>,
## #   reservePriceUSD <dbl>, amountUSD <dbl>, collateralAmount <lgl>,
## #   collateralReserve <lgl>, principalAmount <lgl>, principalReserve <lgl>,
## #   reservePriceETHPrincipal <lgl>, reservePriceUSDPrincipal <lgl>,
## #   reservePriceETHCollateral <lgl>, reservePriceUSDCollateral <lgl>,
## #   amountUSDPincipal <lgl>, amountUSDCollateral <lgl>,
## #   borrowRateModeFrom <lgl>, borrowRateModeTo <lgl>, stableBorrowRate <lgl>,
## #   variableBorrowRate <lgl>
```

Analyze Transaction Types

Next, we will examine the different types of transactions present in the data. We make a bar plot to visualize the number of each transaction types. Deposit is the most common type of transaction, whereas swaps are the most rare.

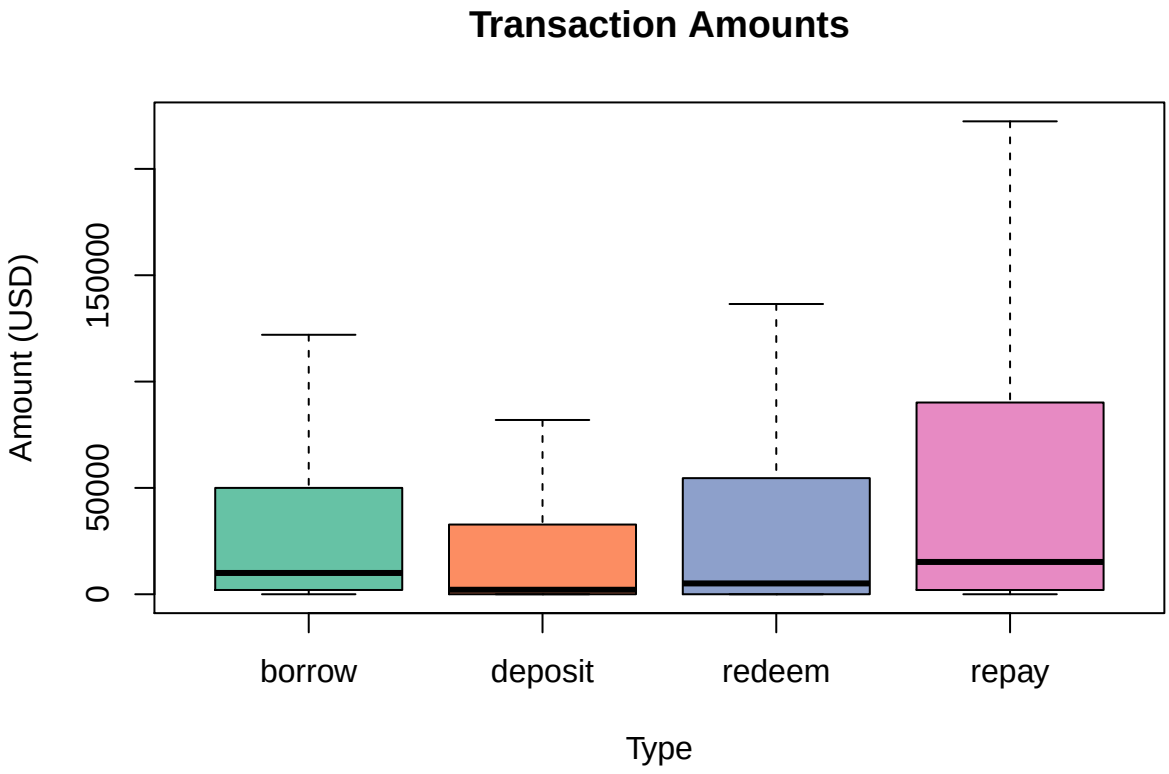
```
#set color palette
colors = brewer.pal(6,"Set2")

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



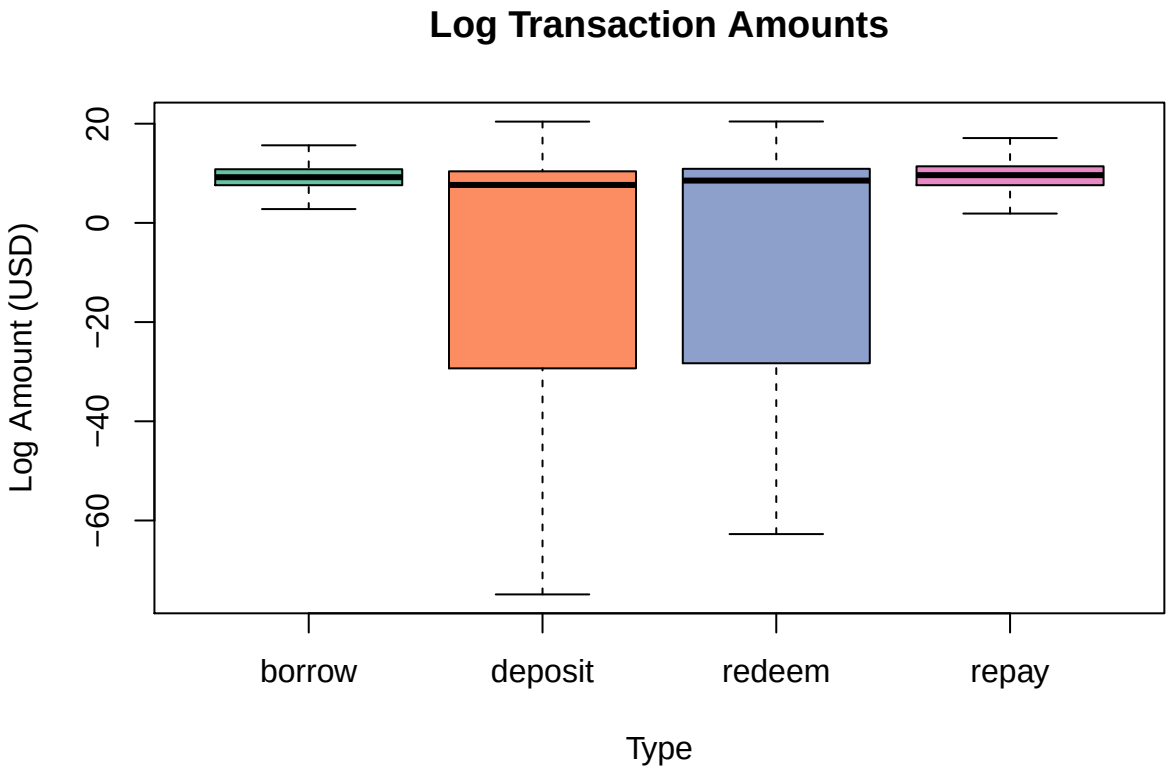
There are more deposits than borrows, because users often need to overcollateralize for loans.

Now, we will examine the amount of US dollars being used in the different types of transactions. We create box plots for the 4 types of transactions that have the amount feature associated with them, and 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="Am
```




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

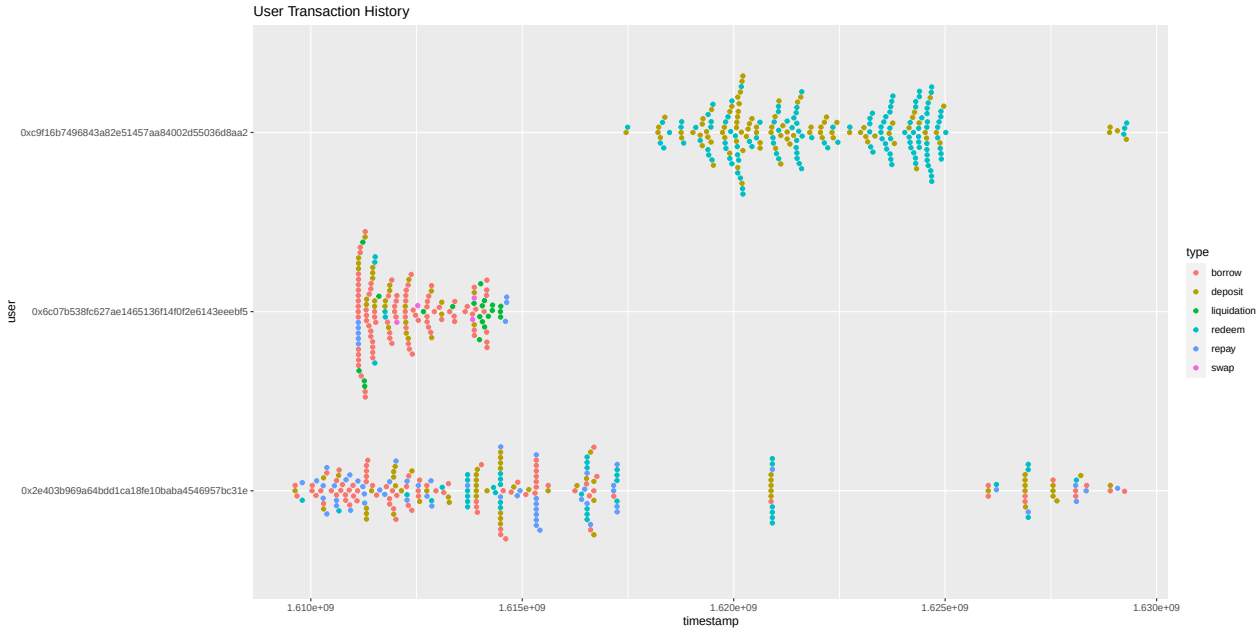


There are many borrows and repays with high transactions amounts, but deposits and redeems have much lower transactions amounts.

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)

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

#liquidations
#borrow
#deposit
#redeem
#repay
#swap

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

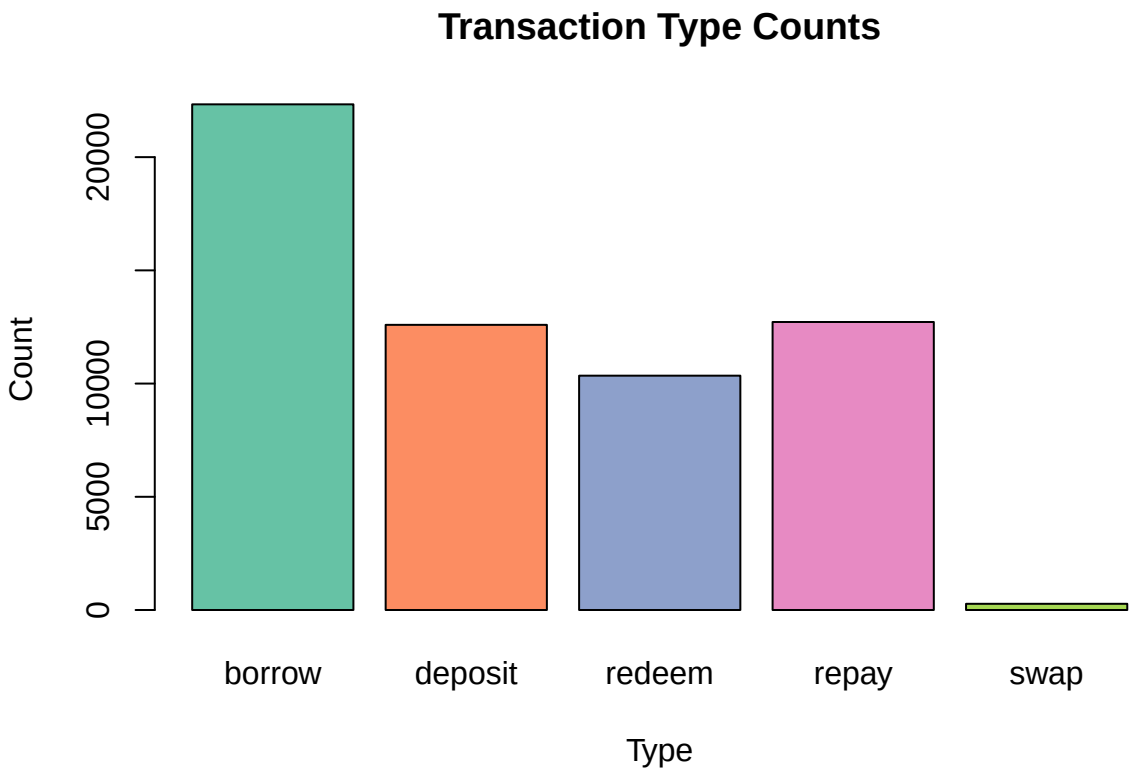


Users have very different transactions patterns, which we will try to better understand.

Analyze Individual Currencies (USDT)

USDT is interesting because it has more borrows than deposits. This may be because it is a stable coin.

```
df.usd<-filter(df,reserve=="USDT")
barplot(table(df.usd$type), main='Transaction Type Counts', xlab='Type',ylab='Count',col=colors)
```



APPENDIX: Accessing RStudio Server on the IDEA Cluster

The IDEA Cluster provides five compute nodes (4x 48 cores, 1x 80 cores, 1x storage server)

- The Cluster requires RCS credentials, enabled via registration in class
  - email John Erickson for problems `erickj4@rpi.edu`
- RStudio, Jupyter, MATLAB, GPUs (on two nodes); lots of storage and computes
- Access via RPI physical network or VPN only

RStudio GUI Access:

- http://lp01.idea.rpi.edu/rstudio-ose/ or http://lp01.idea.rpi.edu/rstudio-ose-3/ (RStudio)
- Linux terminal accessible from within RStudio "Terminal" or via ssh (below)

Shared Data on Cluster:

- Users enrolled in DAR have access to `/academics/MATP-4910-F21`
  - Usually DAR users will see a symbolic ("soft") link in their home directories
  - If you do not, type the following in the **Terminal** via RStudio: `ln -s /academics/MATP-4910-F21/ MATP-4910-F21`
- All idea_users have access to shared storage via `/data` ("data" in your home directories)
  - You might wish to use this for data sharing in team projects...
  - ...but we recommend using github for shared code development
- Shell access to nodes: You must access "landing pad" first, then compute node:
- `ssh your_rcs@lp01.idea.rpi.edu` For example: `ssh erickj4@lp01.idea.rpi.edu`
- Then, `ssh` to the desired compute node, e.g.: `ssh idea-node-02`