

# Large\_Holder\_Activity

April 12, 2021

## 0.1 Alchemix Analytics Question #11

Author Discord Username: dtradke

Date: April 12, 2021

### 0.1.1 Introduction

This notebook is authored for the intention of answering question 13 for the Alchemix Analytics Questions, found at the following link (<https://www.notion.so/Analytics-Bounties-for-Alchemix-ffd6c25bdef3486c9b8dfa9476ac6a39>). This question is geared at understanding what Alchemix (ALCX) functions that the top 25 volume holders of the currency actually use. In this study, I focus on transactions made to three functions: Transmuter, Alchemist/Vault, and 4 Farm Pools. I also include an analysis of the current wealth disparity of the landscape of Alchemix by computing the Gini index of coin volume distribution amongst the community.

I used the Numpy, Pandas, and Matplotlib Python libraries to format and display the data. I used several SQL queries to collect data from the Flipside Crypto database for currency holder addresses, balance, and transaction data. I saved the various data into JSON files for easy offline loading in the future without internet access. My analysis corresponds with the time these JSON files were saved.

The substance of my insights can be organized into four main points:

1. The wealth distribution and Gini index of the ALCX community
2. How the top 25 volume holders use Transmuter compared to the community
3. How the top 25 volume holders use Alchemist/Vault compared to the community
4. How the top 25 volume holders use 4 Farm Pools compared to the community.

The first part of my code is necessary to import the libraries that I will use throughout the notebook. These libraries must be installed on your computer through the command line interface, typically using Pip (i.e. pip3 install numpy).

```
[125]: import numpy as np
import urllib.request as rq
import json
from collections import Counter as c
import time
import matplotlib.pyplot as plt
import os
```

```
import pandas as pd
import operator
import datetime
```

The below code loads the dataset either from a saved JSON file located at 'path', or calls the passed url to load the data from the Flipside database.

```
[126]: def loadData(url, path):
        if os.path.exists(path):
            with open(path, 'r') as fp:
                dataset = json.load(fp)
        else:
            try:
                dataset = rq.urlopen(url)
                dataset = dataset.read()
                dataset = json.loads(dataset)
            except Exception as e:
                print('Unable to get data from flipsidecrypto API. Check the URL_
→below: \n{}'.format(url))

            with open(path, 'w') as fp:
                json.dump(dataset, fp)
        return dataset
```

The below function is designed to take in a list of dictionary (or JSON) elements, and clean the data according to how I will use it later in the notebook. The following functions are all used in the analysis of the research questions, however are all used to clean, calculate, and visualize data later in the notebook.

```
[127]: # Load transactions
def getTransactions(dataset):
    from_addresses = {}
    for i, val in enumerate(dataset):
        if val['FROM_ADDRESS'] not in list(from_addresses.keys()):
            from_addresses[val['FROM_ADDRESS']] = []

        timestamp = val['BLOCK_TIMESTAMP']
        year = timestamp[:4]
        hour = timestamp[11:13]
        day = timestamp[8:10]
        month = timestamp[5:7]
        tweet_timestamp = int(year + month + day)

        from_addresses[val['FROM_ADDRESS']].append(tweet_timestamp)

    df = pd.DataFrame(columns = ['address', 'transactions'])
    days = []
    for key in from_addresses.keys():
```

```

        df.loc[len(df)] = [key, len(from_addresses[key])]
        days = days + from_addresses[key]
    days = np.sort(np.unique(days))
    df = df.sort_values('transactions')
    return df, from_addresses, days

```

The below function loads the user addresses and corresponding balance (volume of ALCX coins) into a Pandas dataframe. The dataframe has the headers 'USR\_ADD' and 'BALANCE' for future reference.

```

[128]: # Format user addresses and balance into Pandas DataFrame
def getHoldersDF(dataset):
    df = pd.DataFrame(columns = ['address', 'balance'])
    for i, val in enumerate(dataset):
        df.loc[len(df)] = [val['USR_ADD'], val['BALANCE']]
    return df

```

**Plots Code** The below functions are used to visualize the data. First, the function plotTopOverTime is used to display the transaction activity over time shown as lines in a line graph. Each line corresponds with a user, and the legend shows users who have made at least one transaction and are in the top 10 volume holders.

```

[129]: # Plots the activity of the highest 25 volume ALCX holders over time
def plotTopOverTime(top_amt, balances, transaction_amt, from_addresses, days,
    ↪target_add_name="Transmuter"):
    fig = plt.figure(figsize=(10, 6))
    top_adds = list(balances.head(top_amt)["address"])

    for user, key in enumerate(top_adds):
        count_arr = np.zeros((days.shape))
        if key in list(from_addresses.keys()): # accounts for the addresses who
    ↪do not make any transactions
            (unique, counts) = np.unique(np.array(from_addresses[key]),
    ↪return_counts=True)
            for i, u in enumerate(unique):
                count_arr[days == u] = counts[i]

        if key in top_adds[:10] and np.amax(count_arr) > 1:
            plt.plot(np.arange(days.shape[0]), count_arr, label="..." + key[-4:])
        else:
            plt.plot(np.arange(days.shape[0]), count_arr)

    day_labels = []
    for i in range(days.shape[0]):
        day_labels.append(str(i)[:4] + "-" + str(i)[4:6] + "-" + str(i)[6:])
    plt.xticks(np.arange(days.shape[0])[:2], day_labels[:2], rotation=90)
    plt.yticks(fontsize=15)

```

```

plt.ylabel("Transactions", fontsize=18)
plt.xlabel("Dates", fontsize=18)
plt.legend(loc = 'upper left', ncol=2)
plt.title("Transactions Made to "+target_add_name+" by Top "+str(top_amt)+"
→Over Time", fontsize=18)
plt.show()

```

The below code calculated the Gini index (explained further below), and helps visualize the wealth disparity in the top 25 volumous coin holders. The Gini index is calculated as a function of the entire population despite the wealthDistribution bar chart only showing the top 25 volume holders.

```

[130]: # Calculates/displays the ALCX volume distirbution and GINI index
def gini(x):
    # Mean absolute difference
    mad = np.abs(np.subtract.outer(x, x)).mean()
    # Relative mean absolute difference
    rmad = mad/np.mean(x)
    # Gini coefficient
    g = 0.5 * rmad
    return g

def wealthDistribution(top_amt, balances):
    top_holders = list(balances.head(top_amt)["address"])
    top_balances = np.array(list(balances.head(top_amt)["balance"]))

    fig = plt.figure(figsize=(10, 6))
    user_ids = []
    for i, val in enumerate(top_holders):
        user_ids.append("..." + val[-4:])

    plt.bar(user_ids, top_balances)
    plt.ylabel("ALCX Volume", fontsize=18)
    plt.xlabel("User Addresses", fontsize=18)
    plt.xticks(rotation=90)
    plt.yticks(fontsize=15)

    gini_idx = gini(np.array(list(balances["balance"])))

    plt.title("ALCX Volume Distribution of top 25 holders ; GINI index:
→"+str(round(gini_idx,3)), fontsize=18)
    plt.show()

```

The below function creates a bar chart showing the number of transactions made with each function by the accounts with the top 25 volume of ALCX coins.

```

[131]: # Creates a bar chart for the top n most volumous owners, and "other" who is
→everyone else

```

```

def topAmtBar(top_amt, balances, transaction_amt, target_add_name="Transmuter"):
    top_holders = list(balances.head(top_amt)["address"])
    top_balances = np.array(list(balances.head(top_amt)["balance"]))

    fig = plt.figure(figsize=(10, 6))
    user_ids, transactions_made = [], []
    for i, val in enumerate(top_holders): # loop through top holders
        user_ids.append("..." + val[-4:])
        row = transaction_amt.loc[transaction_amt['address'] == val]
        if val in list(transaction_amt['address']):
            transactions_made.append(int(row["transactions"]))
        else:
            transactions_made.append(0) #no transactions made

    plt.bar(user_ids, transactions_made)
    plt.ylabel("Transactions", fontsize=18)
    plt.xlabel("User Addresses", fontsize=18)
    plt.xticks(rotation=90)
    plt.yticks(fontsize=15)
    plt.title("Transactions Made to " + target_add_name + " by Top " + str(top_amt) + "
→ALCX Volume Holders, Mean: " + str(np.mean(np.array(transactions_made))),
→fontsize=18)
    plt.show()

```

The below function creates a pie chart showing the number of transactions made with each function by the accounts with the top 25 volume of ALCX coins. We also calculate the percentage of total transactions with each function which belong to the top 25 volume holders and display it in the title of each pie chart in our analysis.

```

[132]: # Creates a pie chart for the top n most volumous owners, and "other" who is
→everyone else
def topAmtPie(top_amt, balances, transaction_amt, target_add_name="Transmuter",
→rot_angle=90):
    top_holders = list(balances.head(top_amt)["address"])
    top_balances = np.array(list(balances.head(top_amt)["balance"]))

    fig = plt.figure(figsize=(10, 6))
    user_ids, transactions_made = [], []
    for i, val in enumerate(top_holders): # loop through top holders
        user_ids.append("..." + val[-4:])
        row = transaction_amt.loc[transaction_amt['address'] == val]
        if val in list(transaction_amt['address']):
            transactions_made.append(int(row["transactions"]))
        else:
            transactions_made.append(0) #no transactions made

```

```

    perc_transactions = np.sum(np.array(transactions_made)) / np.sum(np.
    ↳ array(transaction_amt["transactions"]))

    user_ids.append("Other")
    transactions_made.append(np.sum(np.array(transaction_amt["transactions"])))
    ↳ np.sum(np.array(transactions_made)))

    plt.pie(transactions_made, labels=user_ids, startangle=rot_angle,
    ↳ autopct='%1.1f%%', rotatelabels=True)
    plt.axis('equal')
    plt.title("Percent of Transactions Made to "+target_add_name+" by Largest
    ↳ 25 Volume Holders: "+str(round(100*perc_transactions,2))+"%", fontsize=18)
    plt.show()

```

### 0.1.2 Experiment 1: Wealth Distribution

Our first experiment involves collecting every wallet which contains ALCX and their corresponding amount of currency. We either load a pre-saved JSON file, or load the results from a SQL query located at the below url. The data comes from the Flipside Crypto transaction data from ALCX.

I also calculate the Gini index of the entire ALCX community. In economics, the Gini index is a measure of statistical dispersion intended to represent the income inequality or wealth inequality within a nation or any other group of people. Values closer to 0 represent more equality, and values closer to 1 represent more inequality amongst the entire population.

```

[133]: # Load holder balances
path = 'data/Max_Holder_data.json'
max_holders_url = 'https://api.flipsidecrypto.com/api/v2/queries/
    ↳ 0be6ad53-2eea-4684-85e6-23af172e61fc/data/latest'
dataset = loadData(max_holders_url, path)
balances = getHoldersDF(dataset)

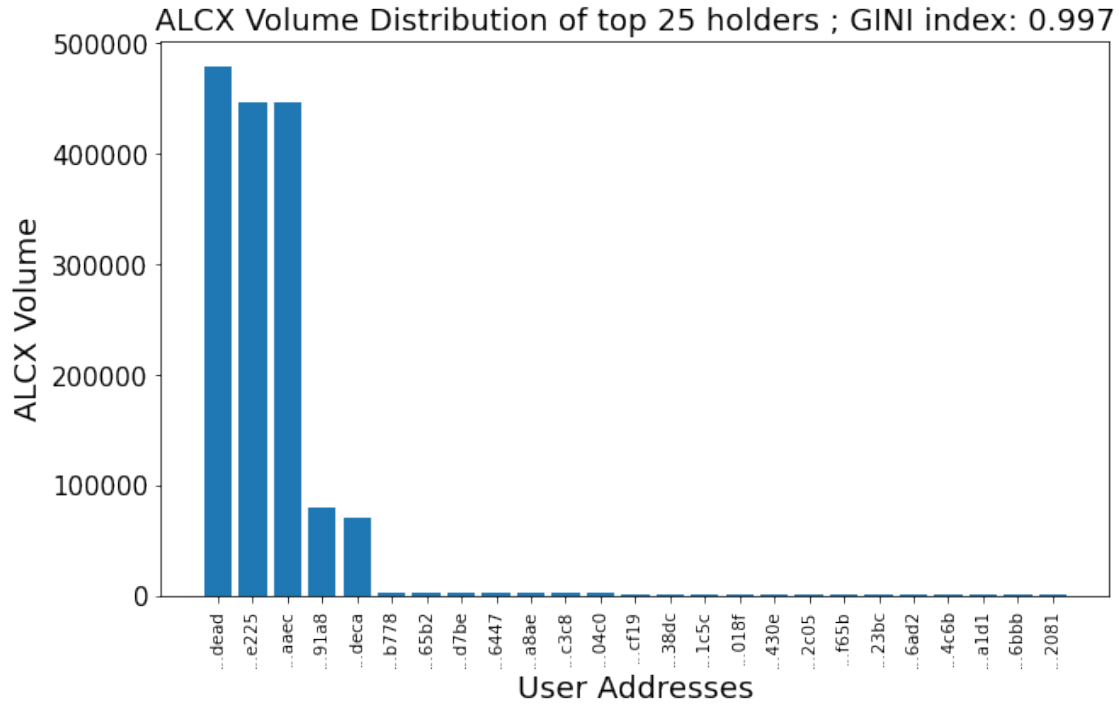
```

**Results** The below chart shows the top 25 holders of ALCX according to volume. It is obvious that the top 5 ALCX holders hold a vast majority of the currency, and among them, the top 3 hold far more than the next two. This represents incredible inequality among the ALCX community, which is also represented with the calculated Gini index of 0.997, shown in the title of the figure. While ALCX is still in infance, this distiribution could change with time to become more equal among the entire community.

```

[134]: wealthDistribution(25, balances)

```



### 0.1.3 Experiment 2: Transactions with Transmuter

Our second experiment analyzes how the top 25 volume holders of ALCX interact with the Transmuter function. I either load transaction data from a saved JSON file or directly from the Flipside Crypto dataset using the url below. I perform the same analysis for each of the remaining experiments.

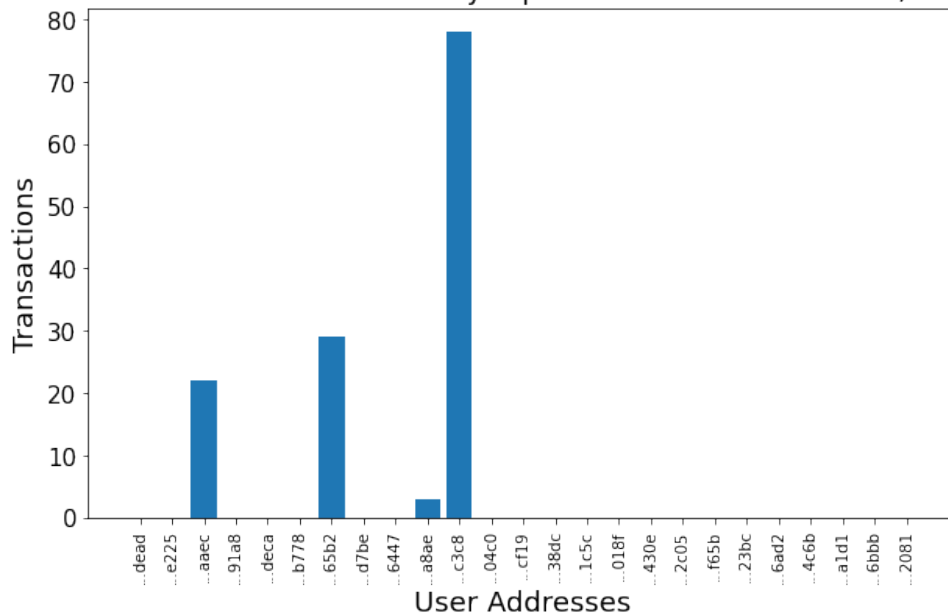
```
[135]: # Load transactions with Transmuter
path = 'data/transmuter_data.json'
url = 'https://api.flipsidecrypto.com/api/v2/queries/
↳b9aca1e1-7a2c-4cdf-bc99-57613baa3ee8/data/latest'
dataset = loadData(url, path)
transaction_amt, from_addresses, days = getTransactions(dataset)
```

### 0.1.4 Results

The bar chart below shows the interactions with Transmuter among the top 25 ALCX volume holders. The first interesting finding is that only 4 of the top 25 (16%) of the top 25 volume holders interact with Transmuter. Among the top 5 holders, who own the vast majority of wealth, only one holder interacts with Transmuter. This accounts for a mean of 5.28 interactions among the top 25.

```
[136]: topAmtBar(25, balances, transaction_amt, target_add_name='Transmuter')
```

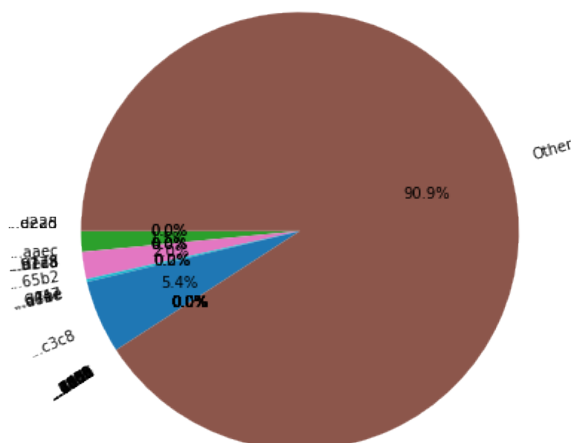
Transactions Made to Transmuter by Top 25 ALCX Volume Holders, Mean: 5.28



The below pie chart shows the transactions with Transmuter from the top 25 compared to the remainder of the population. The top 25 holders only account for 9.15% of the transactions with Transmuter, leaving the remainder of the community to account for over 90% of transactions. Among the top 25, a single holder accounts for more than 50% of their transactions (5.4% of total population's transactions). This result shows that the wealthiest holders tend not to use the Transmuter function as much as the rest of the population.

```
[137]: topAmtPie(25, balances, transaction_amt, target_add_name='Transmuter',
↳ rot_angle=180)
```

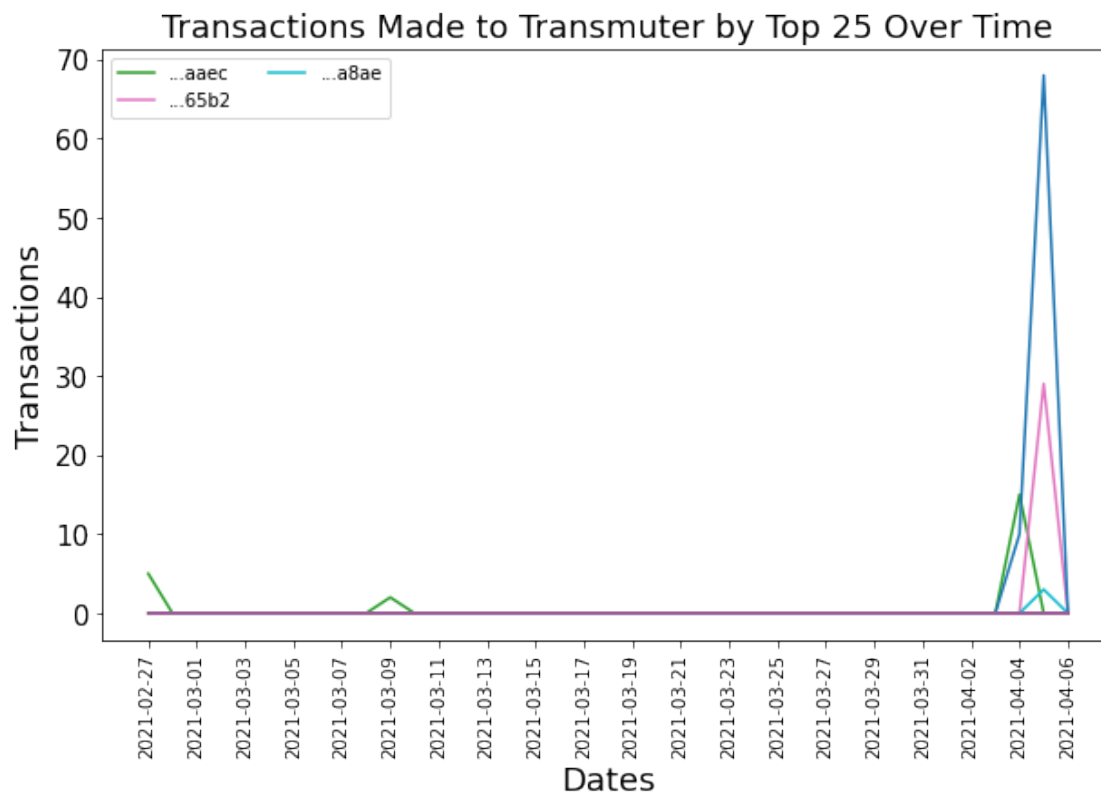
Percent of Transactions Made to Transmuter by Largest 25 Volume Holders: 9.15%





The chart below is different than the two above in that it has a temporal aspect to it. I show the number of Transmuter transactions at daily resolution since February 27, 2021. The legend shows holders that are in the top 10 volume holders and have made at least one transaction. It is clear that the transactions with Transmuter have picked up substantially within the wealthiest community in the last week.

```
[138]: plotTopOverTime(25, balances, transaction_amt, from_addresses, days,
    ↪target_add_name="Transmuter")
```



### 0.1.5 Experiment 3: Transactions with Alchemist/Vault

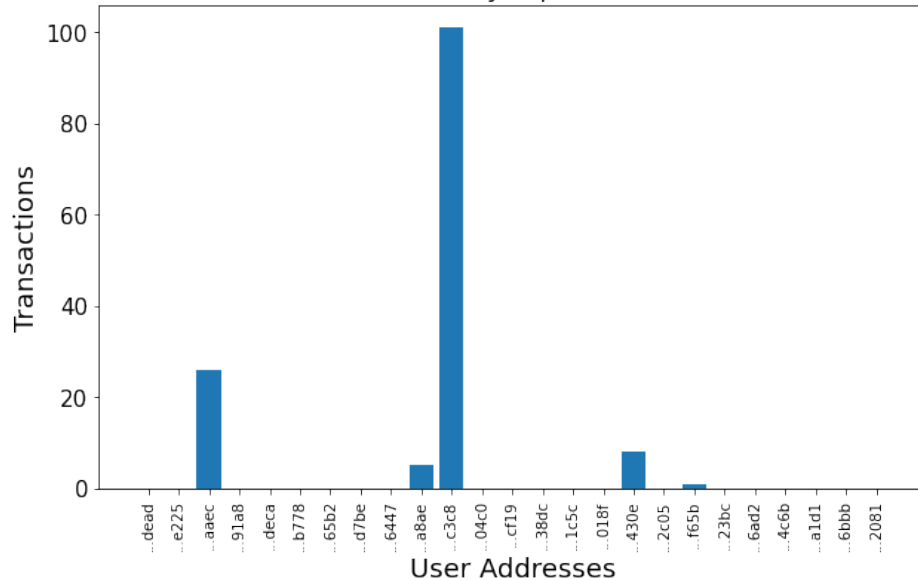
The below results correspond with the top 25 volume ALCX holders transactions with Alchemist/Vault. The data load and graph configurations are all the same as Experiment 2.

```
[139]: # Load transactions with Alchemist/Vault
path = 'data/Alchemist-Vault_data.json'
url = 'https://api.flipsidecrypto.com/api/v2/queries/
    ↪28d73cd0-ed5e-4baa-9845-c0e124e14c0a/data/latest'
dataset = loadData(url, path)
transaction_amt, from_addresses, days = getTransactions(dataset)
```

**Results** The below chart shows the number of transactions made with Alchemist/Vault by the users within the top 25 volume holders. It is interesting to note that the user with address ‘...c3c8’ makes the most transactions with Alchemist/Vault, as they were also the most active with Transmuter in Experiment 2. There is still not many transactions among the top 25 volume holders, with a mean of 5.64 transactions.

```
[140]: topAmtBar(25, balances, transaction_amt, target_add_name='Alchemist/Vault')
```

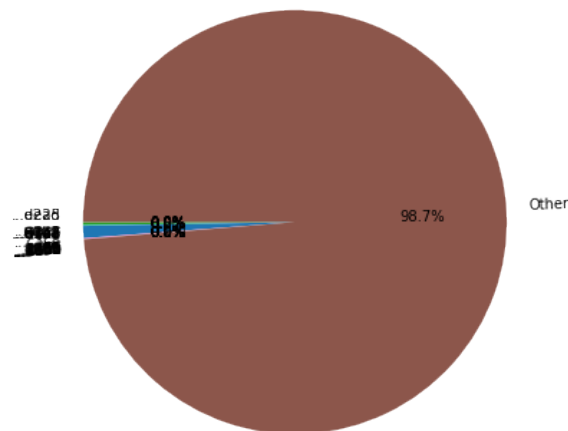
Transactions Made to Alchemist/Vault by Top 25 ALCX Volume Holders, Mean: 5.64



The below pie chart shows the percent of transactions to Alchemist/Vault which came from the top 25 volume holders. The disparity with the rest of the population is even more present in this experiment, with only 3.3% of the total transactions.

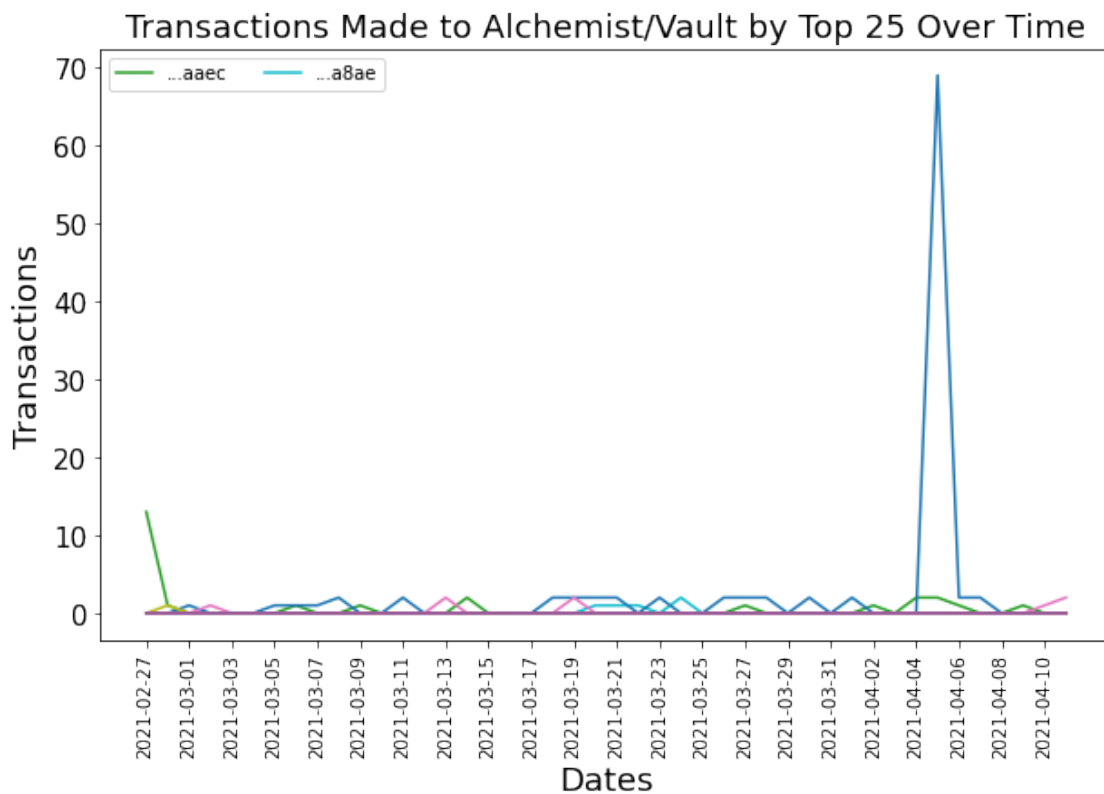
```
[141]: topAmtPie(25, balances, transaction_amt, target_add_name='Alchemist/Vault',
    ↪rot_angle=180)
```

Percent of Transactions Made to Alchemist/Vault by Largest 25 Volume Holders: 1.3%



The below chart shows the interactions with Alchemist/Vault over time. Like previously, the legend only shows users in the top 10 volume holders who make at least one transaction. It is clear that only a single user in the top 25 makes any significant number of transactions with Alchemist/Vault.

```
[142]: plotTopOverTime(25, balances, transaction_amt, from_addresses, days,
    ↳target_add_name="Alchemist/Vault")
```



### 0.1.6 Experiment 4: Transactions with 4 Farm Pools

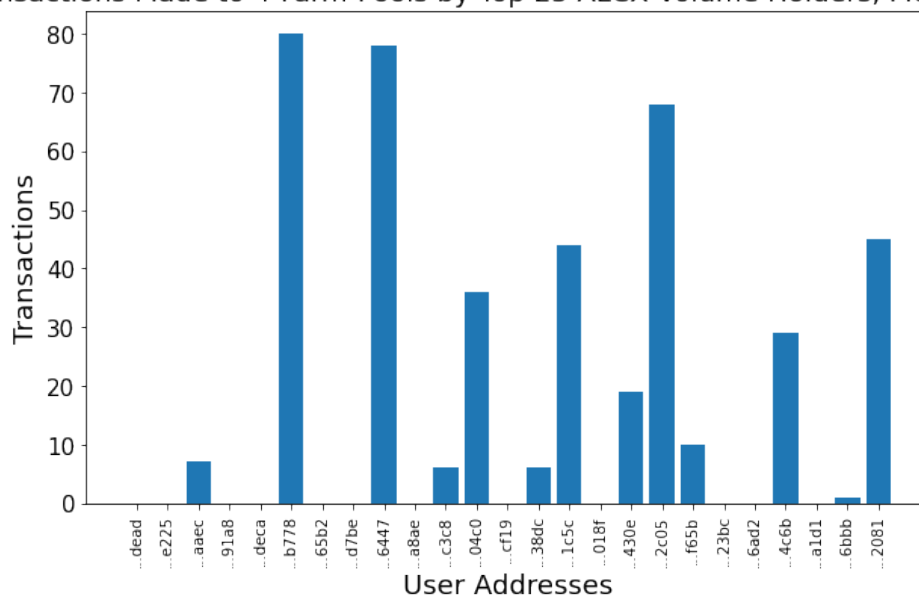
The below experiment is similar to Experiment 2 and Experiment 3, however these transactions are from the top 25 volume holders to 4 Farm Pools. The data is loaded in according to the same methods as before, either an existing JSON file or from the Flipside Crypto dataset using the below url.

```
[143]: # 4-Farm-Pools Data
url = 'https://api.flipsidecrypto.com/api/v2/queries/
      ↪2b017995-8bd1-4688-bb1b-2f6f767b2494/data/latest'
path = 'data/4-Farm-Pools_data.json'
dataset = loadData(url, path)
transaction_amt, from_addresses, days = getTransactions(dataset)
```

The below chart shows the transactions between the top 25 volume holders and 4 Farm Pools. It is clear that these transactions are much more common within the top 25 community, with a mean of 17.16 transactions. Interestingly enough the user with address ‘...c3c8’ who had the majority of transactions in the first two experiments does not have many with 4 Farm Pools.

```
[144]: topAmtBar(25, balances, transaction_amt, target_add_name='4-Farm-Pools')
```

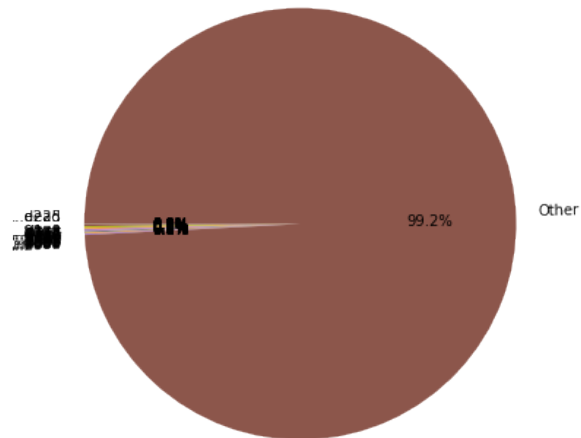
Transactions Made to 4-Farm-Pools by Top 25 ALCX Volume Holders, Mean: 17.16



The below pie chart shows the disparity of function calls to 4 Farm Pools from the top 25 compared to the rest of the population. The top 25 volume holders only account for a slim amount of overall calls, only 0.8% compared to 99.2% being made by those outside of the top 25.

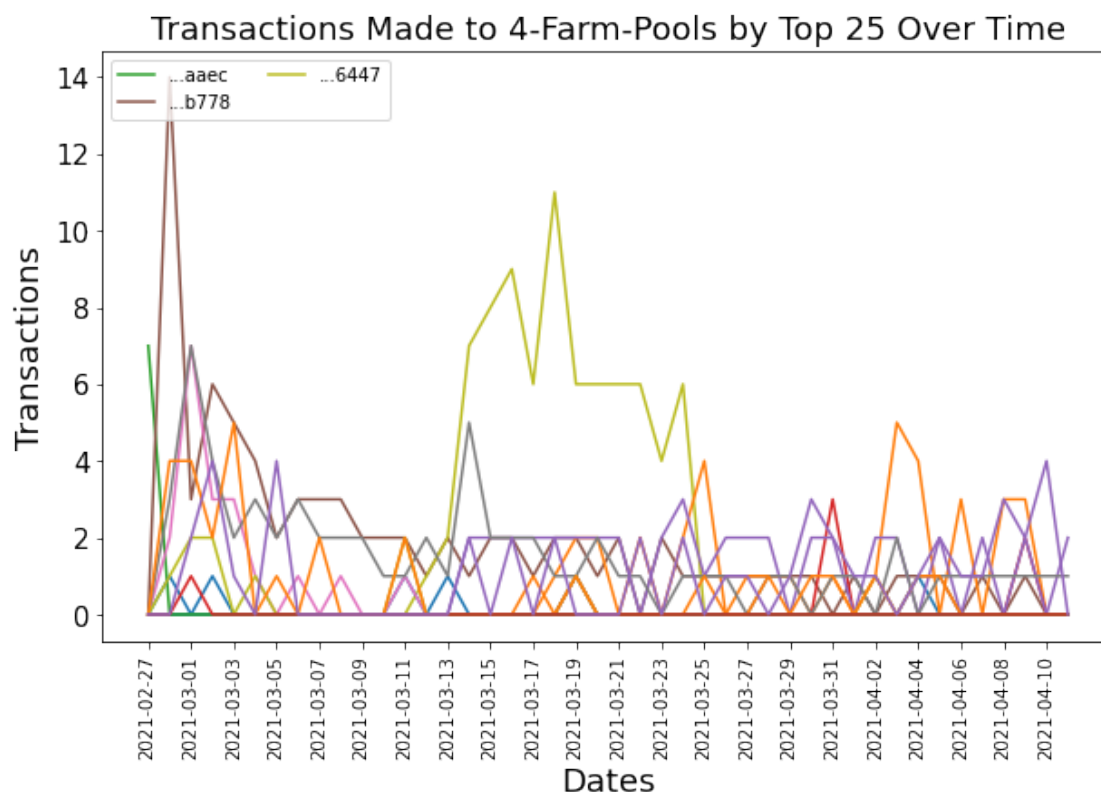
```
[145]: topAmtPie(25, balances, transaction_amt, target_add_name='4-Farm-Pools',
    ↪rot_angle=180)
```

Percent of Transactions Made to 4-Farm-Pools by Largest 25 Volume Holders: 0.82%



The below graph shows transactions over time made by the top 25 volume holders to 4 Farm Pools. The top 25 are clearly more active over time, seen below, with a top 10 holder (yellow) making a large amount of transactions over the last two weeks of March 2021.

```
[146]: plotTopOverTime(25, balances, transaction_amt, from_addresses, days,
    ↪target_add_name="4-Farm-Pools")
```



### 0.1.7 Conclusion

This notebook has analyzed the wealth disparity among the ALCX community, and the transactions from the top 25 volume holders with Transmuter, Alchemist/Vault, and 4 Farm Pools. The first finding is that the ALCX community has extreme inequality, with a Gini index of 0.997. The top 5 holders account for the vast majority of the population's wealth. While this could be a notion of infancy, more data will be needed to track the inequality in the community over time. The second main finding is that the top 25 volume holders seldom transact with Transmuter, Alchemist/Vault, and 4 Farm Pools compared to the rest of the community, despite having the majority of the wealth. Further analysis using artificial intelligence could reveal how the transactions of the top 25 volume holders impact the price of ALCX and the activity of those outside of the top 25.

[ ]: