**NOBO setup:**

This was done in python 3.9.12 and anaconda 4.12.0. The libraries used were pandas, numpy, PrettyTable, and uszipcode. Used Jupyter Notebook as my notebook editor to write the code.



The data above is what was originally given to us for cleaning and analysis.



This is what the data looked like after cleaning the columns and structure of the data. Notice the new columns that have been added for further analysis.

**NOTE:** Will include GitHub links at the end if you want to see the code

**Cleanup Process:**

Since this was a relatively small data set, it was easy to go through the csv and manually change some of the format. This would not work with larger data sets but it was much easier to move the extra column names and entries than using Python to do it.

First, load the csv into Pandas and create a backup copy. Most of the changes I made were to the original df – (DataFrame) which typically is not recommended.

Secondly, the “SORT ZIPCODE,” “# OF NOBO’S,” and “7TH LINE OF ADDRESS” columns were all dropped as they did not provide any relevant information for our goal. I also changed the “1ST LINE OF ADDRESS” to be named “CLIENT.”

Third, created a function to create the “CLASSIFICATION” column. I achieved this by using a key word search function. This function would look at the address line and categorize based on their needs. For example, if IRA or BANK was mentioned, this would have been classified as “RETAIL.” The main issue with this is that there is an “OTHER” classification for clients that did not match any of the keywords and typically were first and last names. I would guess that most people with the other classification are most likely retail investors but could also be an investment company.

Fourth, was time to deal with the all of the different address line columns. To start, create a new column called “COMBINED ADDRESS” (will be renamed later) and then combine the 5th and 6th line of address columns. This was repeated with the extra address columns to make it easier.

Finally, rearranged some of the columns to make it more readable. Then using the *uszipcode* library you can have it go through the “ZIP CODE” column and create a new column with the corresponding states. If there was no zip code or was international, this would be known.

Now the data is clean and can be used for an analysis without worrying about accuracy.

**Data Analysis:**

So I did the data analysis in a separate Jupyter Notebook file to make it easier to understand.

First, I used PrettyTable to make the results usable and easily readable. So a table was created for the count of NOBO’s for each state and the count of each classification category.

Second, graphs were created to display the proportion of clients who had over 100,000, 1,000,000, and 5,000,000 shares. Then the proportion was calculated and displayed on the graph.

Third, was to calculate the proportion of known shares and total shares issued. So it calculated the proportion of total shares each client makes up for that CUSIP and the missing CUSIP proportion of data. The final goal was the table that contained CUSIP, CLIENT, proportion of known shares, and proportion of total shares. This table successfully conveys the original goal.

**Links:**

<https://github.com/rosborn40383/NOBO/blob/main/NOBO%20Cleanup.ipynb> **original cleanup**

[NOBO/NOBOO-add-states.ipynb at main · rosborn40383/NOBO (github.com)](https://github.com/rosborn40383/NOBO/blob/main/NOBOO-add-states.ipynb) **Adding the states column**

[NOBO/NOBO\_GRAPHICS2.ipynb at main · rosborn40383/NOBO (github.com)](https://github.com/rosborn40383/NOBO/blob/main/NOBO_GRAPHICS2.ipynb) **Clean graphics for viewing**