Million Lines Project

Project Data Source and Project Purpose

The original excel file was a near 1 million lines of one-column excel data, including information of four sets of data: company, fund, person, and asset of investors (as shown in figure 1). But all the information was mixed, and there was no order within the data. In addition, each set of data included different information, and some lines included useless information. However, there was one pattern that could be utilized – the font of the beginning of each set of data were the same across all data.

The purpose: Was to brief out the industry sector distribution, and evaluate the data quality to the CEO. was to put all the data into relational databases, with varied columns indicating the attributes of the specific data sets.

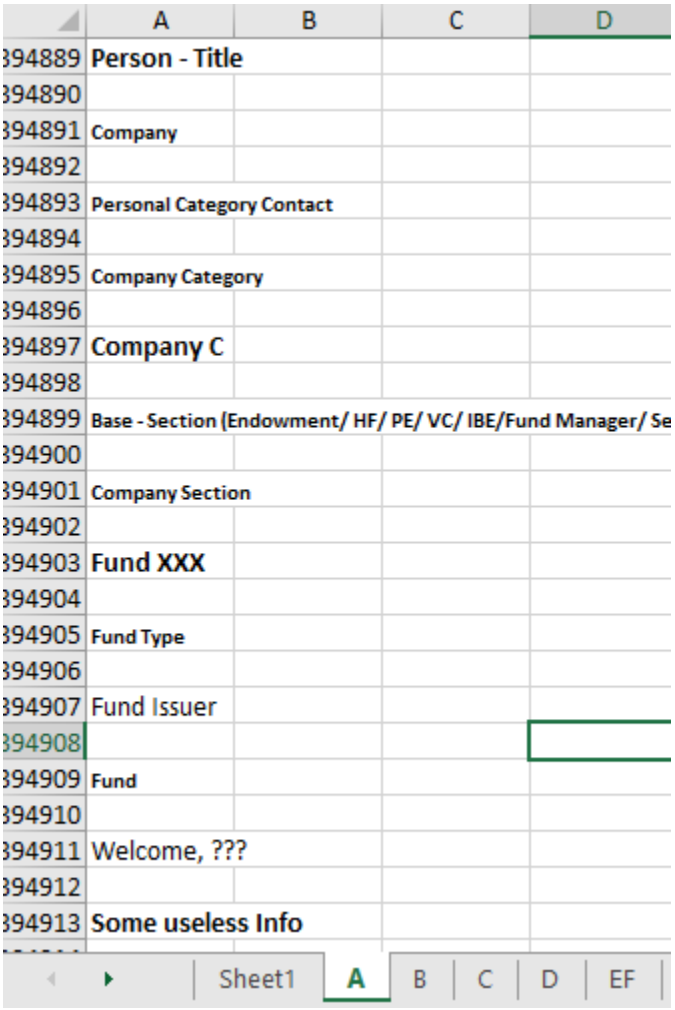

Figure 1 A Sample of the Original Data

Project Methodology

The first step: Remove all unnecessary information. All the unnecessary included the same words, so I used wildcards with excel replace function to remove the unnecessary information.

The second step: Chop the data into small pieces. The original data was too big synthetically, would take life long to open it in all each time. So I used VBA Sub to chop the data within each sheet into separate worksheets. The code was a bit long, so I did not include here.

The third step: Since each new set of data began with a specific font (bold and font size = 12), I used a second VBA to define a function to find out if a cell contains that certain font style. By applying cell.Font.Bold and cell.Font property with IF-Else structure.

The fourth step: Then I used the function described above in the next column to identify out the lines with the specific format, and added a special symbol (1####, which would almost impossible to appear in names) to the front of the cell. This step is to assist OpenRefine to recognize patterns.

The fifth step: I used OpenRefine and Record to turn each line into separate records. And save as separate excel (Se a showcase as in figure 2).

A	B	C	D
first	Column3 2	Column3 3	Column3 4
Company Name	Base -Type	Type	
Company Name	Base -Type	Type	
People - Title	Company	Person Category	Company Category
Asset Name	Base -Type	Asset	
Company Name	Base -Type	Type	
People - Title	Company	Person Category	Company Category
Fund Name	Fund type	Issuer	Fund
Asset Name	Base -Type	Asset	
Asset Name	Base -Type	Asset	
Fund Name	Fund type	Issuer	Fund
Asset Name	Base -Type	Asset	

Figure 2 Showcase of Initial Manipulated Table

The sixth step: I then used Python to find patterns, and put all Asset data into one excel table, all fund data into one excel table, all personal data into people excel table, and the others I assume them to be companies.

The seventh step: Afterwards, I validated the data. And fit them into pre-set columns by using excel split column, if-else function, etc. I also made some validation, to ensure only the right data fit into the right table. A showcase of this step for company table could be find in figure 3.

A	B	C	D	E	F
Name	City	State	Country	Category	General_Cate
Company Name	City 1	State 1	Country 1	Category	General Category
Company Name	City 2	State 2	Country 2	Category	General Category
Company Name	City 3		Country 3	Category	General Category
Company Name	City 4	State 4	Country 4	Category	General Category
Company Name	City 5	State 5	Country 5	Category	General Category
Company Name	City 6	State 6	Country 6	Category	General Category
Company Name	City 7	State 7	Country 7	Category	General Category
Company Name	City 8	State 8		Category	General Category
Company Name		State 9	Country 9	Category	General Category
Company Name	City 10	State 10	Country 10	Category	General Category
Company Name	City 11	State 11		Category	General Category

Figure 3 Showcase of data validation in General

The eighth step: I then used Python to group the data based on categories, locations (if any), and generated data reports by Python. At first there were some un-correspondence within the table, as the sum and the data within each category did not match. But after recheck and re-validation, the errors were corrected, and the report was neat and clean.

The ninth step: The company CEO prefer to see Excel reports, so, I used excel to made a visualized reports and pivot-table report. As shown in figure below.

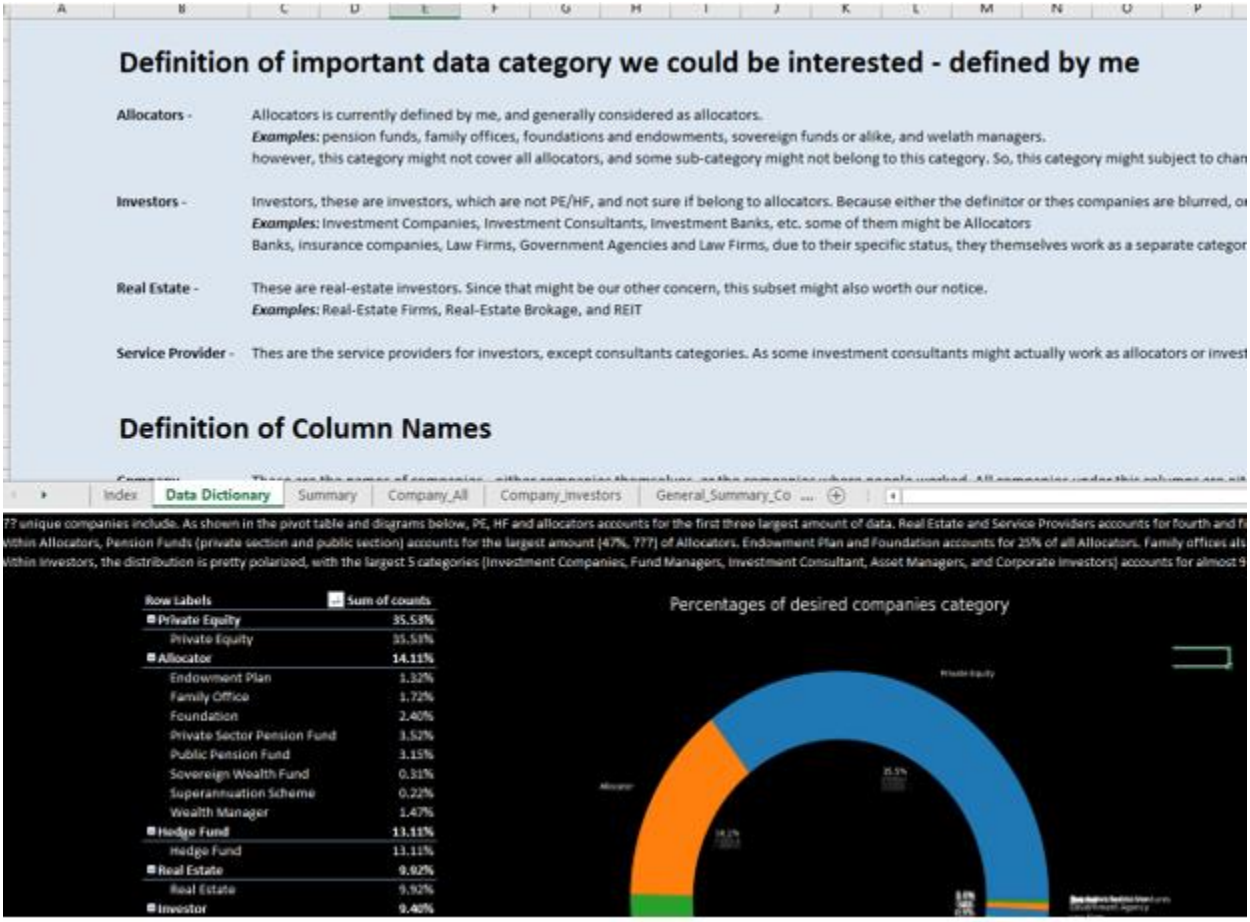

Figure 4 A screenshot of modified report