

Market

Tags

Besides the main field of Data, I am also a young investor. Recently, I want to observe the market overview not with available support tools but with what I can create.

So I've decided to try creating a Dashboard using PowerBI with the data source being crawled from the web for free using Python.

Let's see what I will do.

▼ Get & Transform Data with Python

I access the TCBS website to retrieve data. (<https://tcinvest.tcbs.com.vn/tc-price/tc-analysis/financial?ticker=TCB>).

Quý - Tỷ lệ	2021Q1	2021Q2	2021Q3	2021Q4	2022Q1	2022Q2	2022Q3	2022Q4	2023Q1	2023Q2
I. Kết quả kinh doanh										
Thu nhập lãi thuần	6,124	6,588	6,742	7,245	8,111	7,794	7,565	6,819	6,527	6,295
Lãi thuần từ HĐ dịch vụ	1,325	1,457	1,497	2,103	1,687	1,987	2,123	2,535	1,944	2,019
Lãi thuần từ HĐ đầu tư	812	717	384	279	280	411	209	-422	-228	72
Lãi thuần từ HĐ khác	671	444	156	532	488	743	441	495	1,057	939
Tổng thu nhập HĐ (TOI)	8,932	9,206	8,779	10,159	10,006	10,934	10,338	9,427	9,300	9,325
YoY%	48.1	56.2	17.1	30.9	12.0	18.8	17.8	-7.2	-7.1	-14.7
Chi phí dự phòng	-851	-598	-589	-627	-218	-417	-609	-691	-535	-807
Lợi nhuận trước dự phòng	6,369	6,615	6,151	6,767	7,004	7,739	7,324	5,437	6,158	6,456
Chi phí hoạt động	-2,563	-2,591	-2,628	-3,392	-3,003	-3,196	-3,014	-3,990	-3,142	-2,869
LN trước thuế	5,518	6,018	5,562	6,140	6,785	7,321	6,715	4,746	5,623	5,649
LN sau thuế	4,476	4,807	4,432	4,684	5,615	5,882	5,368	3,572	4,537	4,503
LNST & CD thiếu số	4,397	4,711	4,338	4,592	5,505	5,804	5,298	3,544	4,497	4,455
YoY%	79.0	67.2	40.0	16.2	25.2	23.2	22.1	-22.8	-18.3	-23.2
II. Bảng cân đối kế toán										
Tiền & tương đương	4,273	3,554	3,303	3,579	3,277	3,204	3,026	4,216	2,852	3,113
Tiền gửi tại NHNN	1,764	4,405	3,395	4,909	8,999	4,814	4,709	11,476	15,834	7,860
Tiền gửi TCTD khác	22,454	31,310	51,058	39,520	58,126	57,270	65,183	69,925	56,675	63,215
Cho vay TCTD khác	15,907	20,775	26,570	31,065	14,405	14,962	7,215	13,050	10,265	8,079
Chứng khoán đầu tư	91,447	95,560	99,347	102,670	124,013	98,072	104,673	104,626	99,162	112,613
Cho vay khách hàng	296,290	313,514	321,042	347,341	365,743	391,824	410,546	420,524	465,425	466,546
Nợ xấu	1,135	1,118	1,829	2,294	2,442	2,359	2,665	3,818	3,946	5,012
Dự phòng cho vay KH	-2,491	-2,896	-3,373	-3,736	-3,927	-4,049	-4,397	-4,771	-5,280	-5,793
Cho vay KH ròng	293,799	310,618	317,669	343,606	361,815	387,775	406,148	415,752	460,145	460,753
Tài sản cố định	4,726	6,739	7,106	7,224	7,265	7,220	7,282	8,411	8,275	8,742
Lãi, phí phải thu	5,664	5,736	6,224	5,808	7,253	8,072	8,681	8,029	8,365	8,875

Here I am using TCB for testing data retrieval.


```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import json
```

✓ 0.0s

The first step is to import the necessary libraries for this task.

```
url = "https://apipubaws.tcbs.com.vn/tcanalysis/v1/finance/FPT/incomestatement?yearly=0&isAll=true"
response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    financial_data = json.loads(soup.text)

    latest_data = financial_data[0]
    ticker = latest_data['ticker']
    revenue = latest_data['revenue']
    postTaxProfit = latest_data['postTaxProfit']

    df = pd.DataFrame({'Ticker': [ticker], 'Revenue': [revenue], 'PostTaxProfit': [postTaxProfit]})
    df
```

✓ 0.4s

	Ticker	Revenue	PostTaxProfit
0	FPT	12484	1856

In the income statement, I will extract Revenue and PostTaxProfit (Profit after tax).

```
url = "https://apipubaws.tcbs.com.vn/tcanalysis/v1/finance/FPT/balancesheet?yearly=0&isAll=true"
response = requests.get(url)

if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    financial_data = json.loads(soup.text)

    latest_data = financial_data[0]
    df['Debt'] = latest_data['debt']
    df['Asset'] = latest_data['asset']
    df['Equity'] = latest_data['equity']
    df
```

✓ 0.3s

	Ticker	Revenue	PostTaxProfit	Debt	Asset	Equity
0	FPT	12484	1856	31929	60524	28595

In the balancesheet, I will also retrieve Debt, Equity, and Asset. Now, I will proceed to gather data for each group of stocks belonging to different industries to have the best overview of the market.

First, I will create a dataframe containing the names and IDs of each industry.

IndustryId		IndustryName
0	1	Financial Services
1	2	Real Estate
2	3	Insurance
3	4	Electricity, Water, and Gas
4	5	Banking
5	6	Basic Resources
6	7	Chemicals
7	8	Food and Beverage
8	9	Tourism and Entertainment
9	10	Retail
10	11	Oil and Gas
11	12	Information Technology
12	13	Telecommunications
13	14	Automobiles and Parts
14	15	Personal Care and Household Goods
15	16	Media
16	17	Health
17	18	Construction and Materials
18	19	Industrial Goods and Services

I use <https://simplize.vn/> to select stocks within each industry and extract a list of stocks for each separate industry.

```

in_1 = ['SSI', 'VND', 'VCI', 'SHS', 'HCM', 'VIX', 'FTS', 'MBS', 'BSI']
in_2 = ['VHM', 'VIC', 'BCM', 'VRE', 'NVL', 'KBC', 'KDH', 'SSH', 'DIG',
        'NLG', 'DXG']
in_3 = ['BIC', 'PVI', 'BMI', 'MIG', 'BVH']
in_4 = ['POW', 'PGV', 'DNH', 'VSH', 'NT2', 'DTK', 'HND', 'QTP', 'SBH', 'GAS', 'GEG']
in_5 = ['MBB', 'VCB', 'BID', 'CTG', 'VPB', 'TCB', 'ACB', 'SSB', 'STB', 'VIB',
        'HDB', 'SHB', 'TPB', 'LPB', 'EIB']
in_6 = ['HPG', 'HSG', 'NKG', 'TVN', 'DHC', 'TNA']
in_7 = ['GVR', 'AAA', 'DPR', 'BRR', 'DGC', 'DCM', 'DPM']
in_8 = ['MSN', 'VNM', 'MCH', 'VSF', 'QNS', 'KDC', 'IDP', 'SBT']
in_9 = ['VJC', 'HVN', 'SCS', 'ACV']
in_10 = ['MWG', 'FRT', 'DGW', 'SAS', 'PET', 'AST']
in_11 = ['PLX', 'BSR', 'OIL', 'PVS']
in_12 = ['FPT']
in_13 = ['FOX']
in_14 = ['CTF', 'SVC', 'HAX']
in_15 = ['PNJ', 'LIX', 'NET']
in_16 = ['HTP']
in_17 = ['IMP', 'DHG', 'DVN', 'DBD', 'TRA', 'DMC']
in_18 = ['C4G', 'HUT', 'VCG', 'SNZ', 'LGC', 'CTR', 'PC1', 'CC1', 'CII', 'SJG',
        'HHV']
in_19 = ['GEX', 'GEE', 'TBD']

```

Here are the stocks with significant weights for each industry.

I create a dictionary containing stock codes and industry codes.

```

~ lists_stock_of_industry = [
    in_1, in_2, in_3, in_4, in_5, in_6, in_7, in_8, in_9, in_10, in_11, in_12, in_13, in_14, in_15, in_16, in_17, in_18, in_19
]
stock_to_id = {}

~ for i, code_list in enumerate(lists_stock_of_industry, start=1):
~     for code in code_list:
~         stock_to_id[code] = i
~     print(stock_to_id)
✓ 0.0s
{'SSI': 1, 'VND': 1, 'VCI': 1, 'SHS': 1, 'HCM': 1, 'VIX': 1, 'FTS': 1, 'MBS': 1, 'BSI': 1, 'VHM': 2, 'VIC': 2, 'BCM': 2, 'VRE': 2, 'NVL': 2, 'KBC': 2, 'KDH': 2, 'SSH': 2, 'DIG': 2, 'NLG': 2, 'DXG': 2, 'BIC': 3, 'PVI': 3, 'BMI': 3, 'MIG': 3, 'BVH': 3, 'POW': 4, 'PGV': 4, 'DNH': 4, 'VSH': 4, 'NT2': 4, 'DTK': 4, 'HND': 4, 'QTP': 4, 'SBH': 4, 'GAS': 4, 'GEG': 4, 'MBB': 5, 'VCB': 5, 'BID': 5, 'CTG': 5, 'VPB': 5, 'TCB': 5, 'ACB': 5, 'SSB': 5, 'STB': 5, 'VIB': 5, 'HDB': 5, 'SHB': 5, 'TPB': 5, 'LPB': 5, 'EIB': 5, 'HPG': 6, 'HSG': 6, 'NKG': 6, 'TVN': 6, 'DHC': 6, 'TNA': 6, 'GVR': 7, 'AAA': 7, 'DPR': 7, 'BRR': 7, 'DGC': 7, 'DCM': 7, 'DPM': 7, 'MSN': 8, 'VNM': 8, 'MCH': 8, 'VSF': 8, 'QNS': 8, 'KDC': 8, 'IDP': 8, 'SBT': 8, 'VJC': 9, 'HVN': 9, 'SCS': 9, 'ACV': 9, 'MWG': 10, 'FRT': 10, 'DGW': 10, 'SAS': 10, 'PET': 10, 'AST': 10, 'PLX': 11, 'BSR': 11, 'OIL': 11, 'PVS': 11, 'FPT': 12, 'FOX': 13, 'CTF': 14, 'SVC': 14, 'HAX': 14, 'PNJ': 15, 'LIX': 15, 'NET': 15, 'HTP': 16, 'IMP': 17, 'DHG': 17, 'DVN': 17, 'DBD': 17, 'TRA': 17, 'DMC': 17, 'C4G': 18, 'HUT': 18, 'VCG': 18, 'SNZ': 18, 'LGC': 18, 'CTR': 18, 'PC1': 18, 'CC1': 18, 'CII': 18, 'SJG': 18, 'HHV': 18, 'GEX': 19, 'GEE': 19, 'TBD': 19}

```



```

df = pd.DataFrame(columns=['Ticker', 'Industry', 'Revenue', 'PostTaxProfit', 'Debt', 'Asset', 'Equity'])
url = "https://apipubaws.tcbs.com.vn/tcanalysis/v1/finance/"
for stock_list in lists_stock_of_industry:
    for stock in stock_list:
        response = requests.get(url + stock + "/incomestatement?yearly=0&isAll=true")
        if response.status_code == 200:
            soup = BeautifulSoup(response.content, 'html.parser')
            income_data = json.loads(soup.text)
            latest_data = income_data[0]
            ticker = latest_data['ticker']
            revenue = latest_data['revenue']
            postTaxProfit = latest_data['postTaxProfit']

            response = requests.get(url + stock + "/balancesheet?yearly=0&isAll=true")
            if response.status_code == 200:
                soup = BeautifulSoup(response.content, 'html.parser')
                balancesheet_data = json.loads(soup.text)
                latest_data = balancesheet_data[0]
                debt = latest_data['debt']
                asset = latest_data['asset']
                equity = latest_data['equity']

            row = {'Ticker': ticker, 'Industry': stock_to_id[ticker], 'Revenue': revenue, 'PostTaxProfit': postTaxProfit, 'Debt': debt, 'Asset': asset, 'Equity': equity}
            df = pd.concat([df, pd.DataFrame([row])], ignore_index = True)

```

I use a loop to perform the same operation as before with FPT, and additionally, I add a column for Industry with the corresponding code.

	Ticker	Industry	Revenue	PostTaxProfit	Debt	Asset	Equity
0	SSI	1	1680	567	28143	50136	21994
1	VND	1	1604	429	26970	42050	15080
2	VCI	1	501	117	7150	14520	7370
3	SHS	1	308	141	944	10771	9828
4	HCM	1	588	157	6381	14300	7919
...
110	SJG	18	1275	230	14547	23560	9012
111	HHV	18	612	109	27517	36079	8562
112	GEX	19	7996	652	31035	52438	21403
113	GEE	19	3854	87	10320	16199	5878
114	TBD	19	358	3	690	1281	591

115 rows × 7 columns

Here are the results.

```

industry_df.to_csv('industry_df.csv', index = False, encoding='utf-8-sig')
df.to_csv('stocks.csv', index = False, encoding='utf-8-sig')

```

✓ 0.0s

I export two CSV files and continue the work with PowerBI.

▼ Visualization with PowerBI

A ^B _C Ticker	1 ² ₃ Industry	1 ² ₃ Revenue	1 ² ₃ PostTaxProfit	1 ² ₃ Debt	1 ² ₃ Asset	1 ² ₃ Equity
Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
SSI	1	1680	567	28143	50136	21994
VND	1	1604	429	26970	42050	15080
VCI	1	501	117	7150	14520	7370
SHS	1	308	141	944	10771	9828
HCM	1	588	157	6381	14300	7919
VIX	1	688	566	113	8552	8439
FTS	1	260	140	2775	6193	3418
MBS	1	401	124	6062	10761	4699
BSI	1	316	124	3512	8080	4568
VHM	2	32614	9749	221148	391331	170183
VIC	2	47295	400	461475	599177	137703
BCM	2	1094	-26	30882	48683	17801
VRE	2	2173	1001	8751	44194	35443
NVL	2	1054	-684	213097	256820	43723
KBC	2	2328	1012	13720	33765	20044
KDH	2	582	256	9504	22970	13466
SSH	2	1005	454	8206	13472	5265
DIG	2	162	9	6172	14047	7875
NLG	2	953	231	13698	26849	13151
DXG	2	714	157	16372	30498	14126
BIC	3	895	129	4987	7523	2537
PVI	3	1546	326	19992	28391	8400
BMI	3	1191	74	4410	6896	2486
MIG	3	934	60	6869	8798	1929
BVH	3	9891	422	198621	220768	22146
POW	4	8429	181	28065	61896	33831
PGV	4	15353	1103	45895	65053	19159
DNH	4	472	235	1920	7562	5642
VSH	4	657	262	4895	9943	5048
NT2	4	2183	144	4493	9166	4673
DTK	4	2965	175	8594	16760	8166
HND	4	3366	331	1833	8195	6363
QTP	4	3708	248	2440	7850	5410
SBH	4	126	29	82	2259	2176
GAS	4	24043	3196	20796	88247	67451
GFG	4	480	7	10618	16329	5712

1 ² ₃ IndustryId	A ^B _C IndustryName
Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
1	Financial Services
2	Real Estate
3	Insurance
4	Electricity, Water, and Gas
5	Banking
6	Basic Resources
7	Chemicals
8	Food and Beverage
9	Tourism and Entertainment
10	Retail
11	Oil and Gas
12	Information Technology
13	Telecommunications
14	Automobiles and Parts
15	Personal Care and Household Goods
16	Media
17	Health
18	Construction and Materials
19	Industrial Goods and Services

Since the data is sourced from a reputable website like TCBS (Techcom Securities), it's highly reliable and clean.

Custom Column

Add a column that is computed from the other columns.

New column name
NetProfitMargin

Custom column formula ⓘ
= [PostTaxProfit]/[Revenue]

Available columns
 Ticker
 IndustryId
 Revenue
 PostTaxProfit
 Debt
 Asset

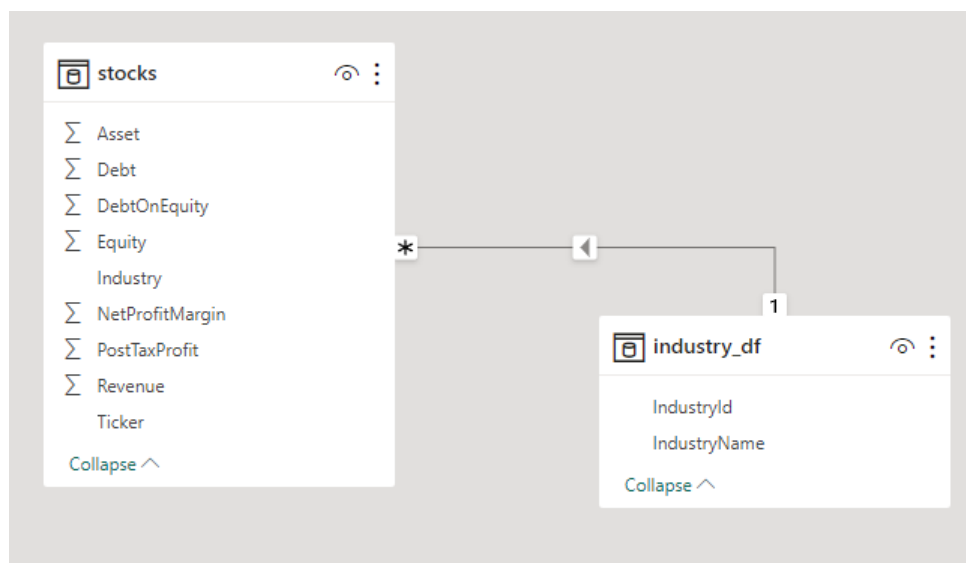
<< Insert

[Learn about Power Query formulas](#)

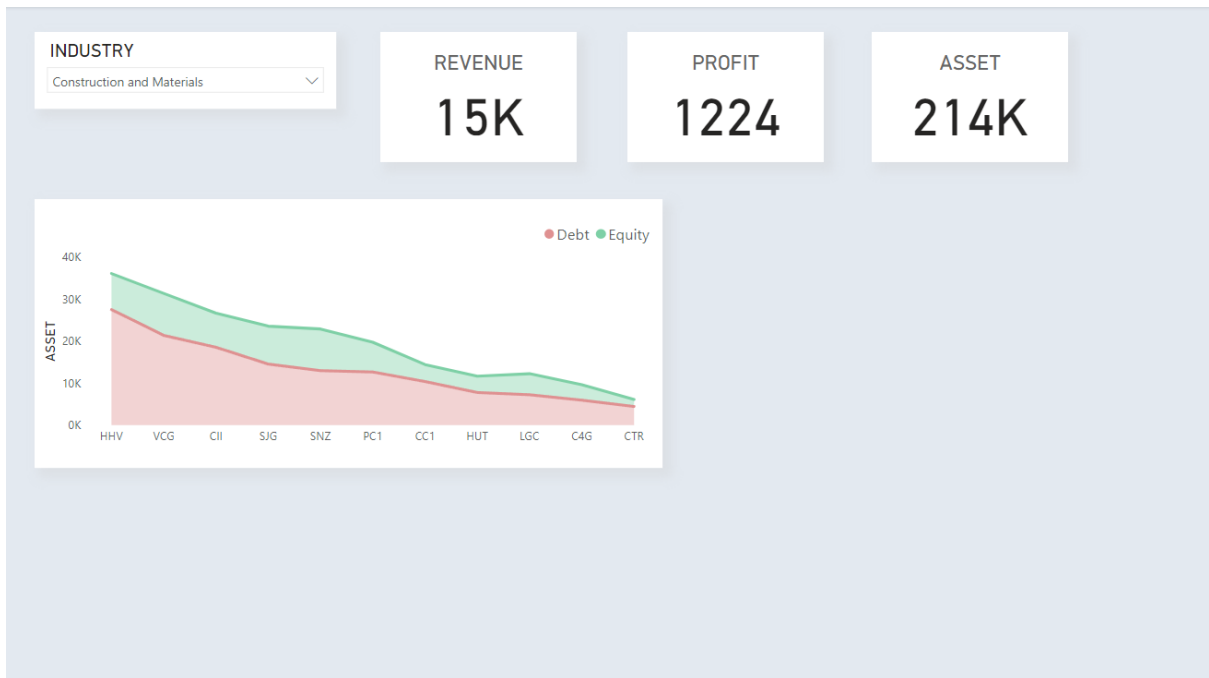
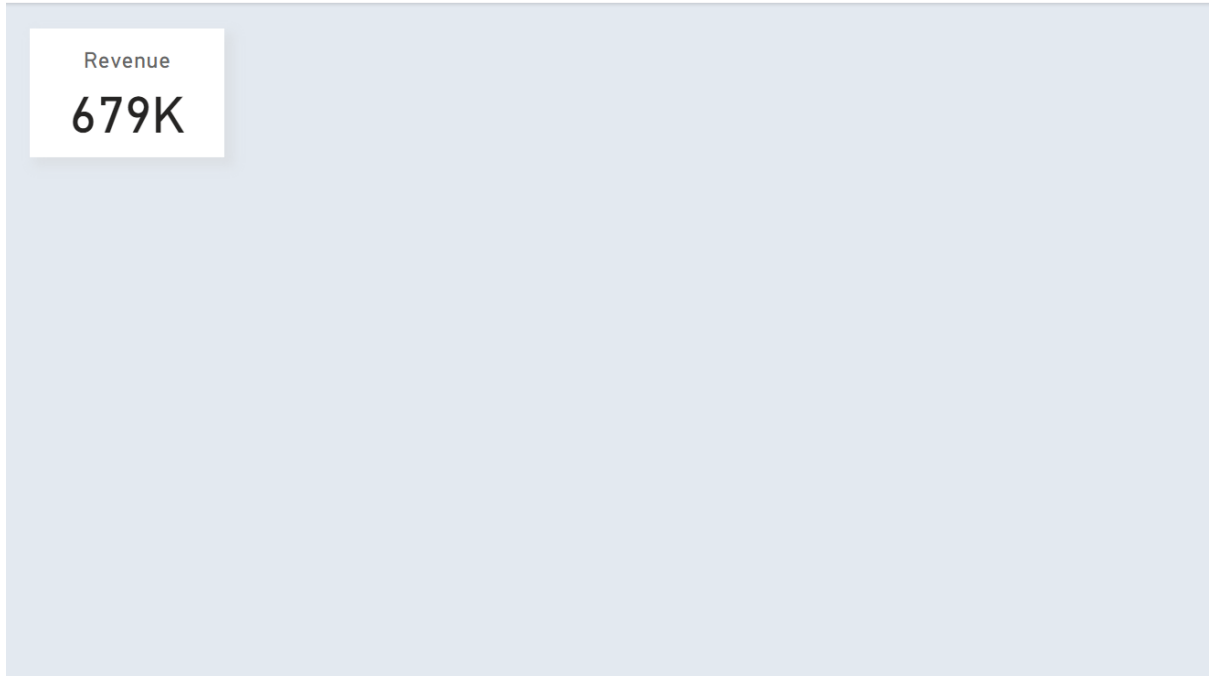
✓ No syntax errors have been detected.

OK Cancel

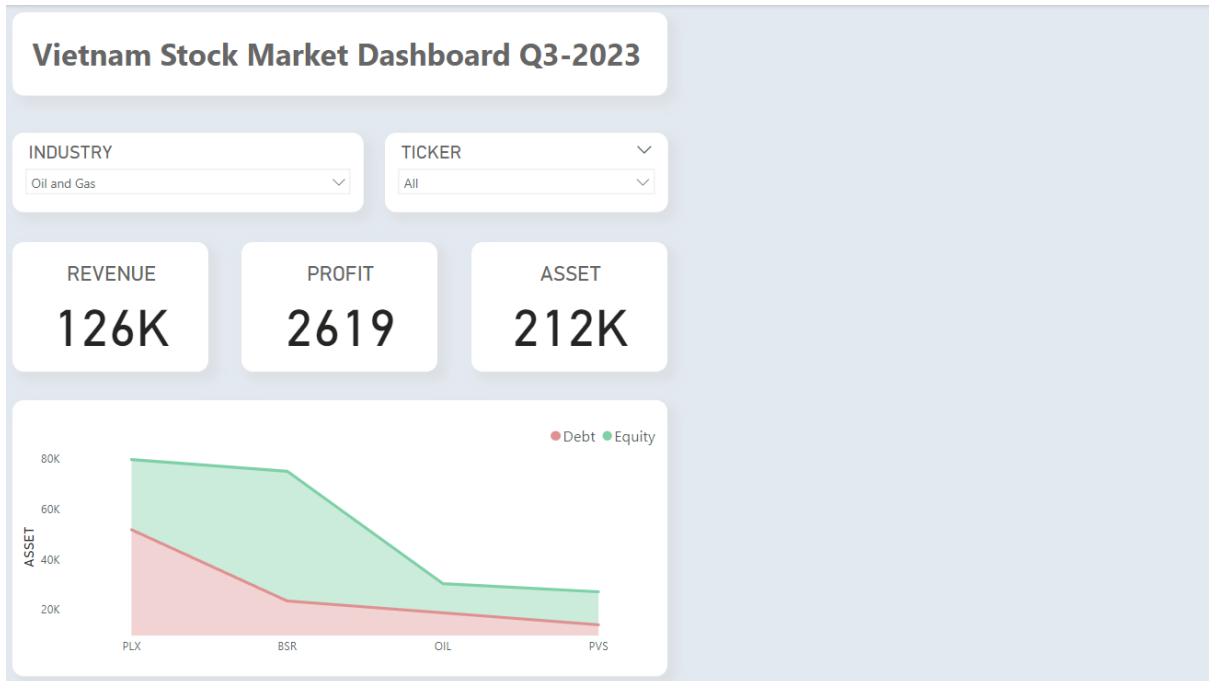
Here, I've added a column for Net Profit Margin, which represents the efficiency of the business process in generating profit after tax.



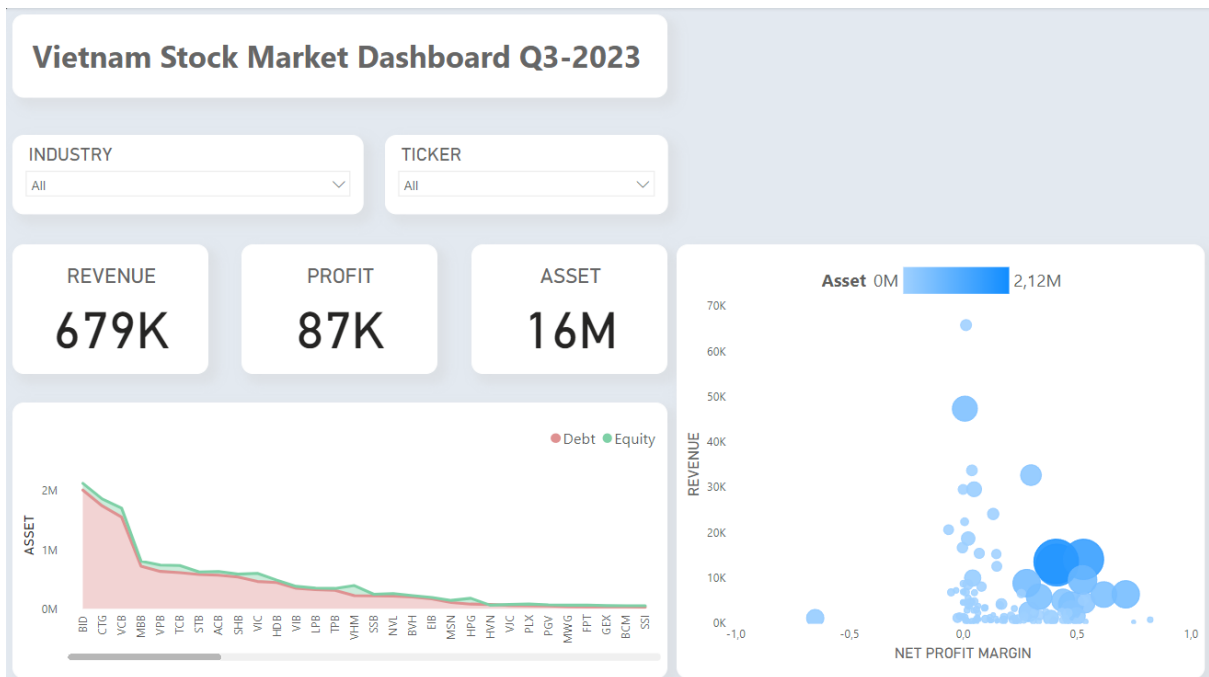
Finally, I establish a relationship between the two tables using the IndustryId.



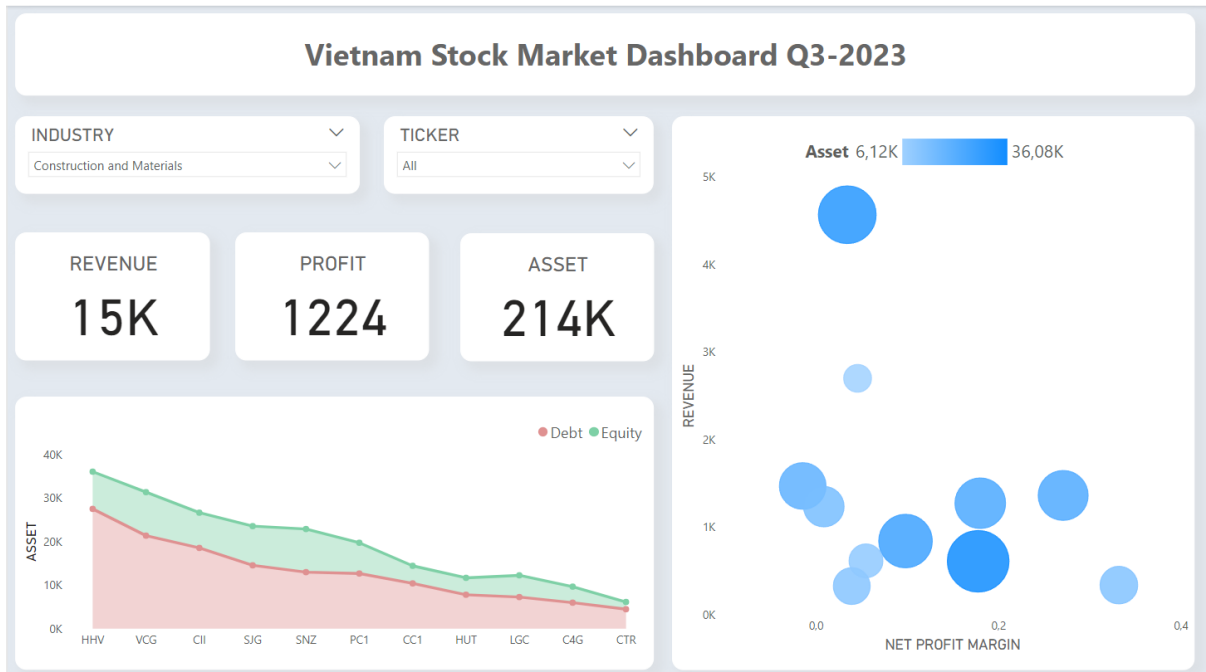
Here, I create a Stacked Area Chart representing a company's Asset. According to the formula, $\text{Asset} = \text{Debt} + \text{Equity}$ (owner's equity). This index partially reflects the company's strong financial position when Debt is much lower than Equity.



Adjust the layout.



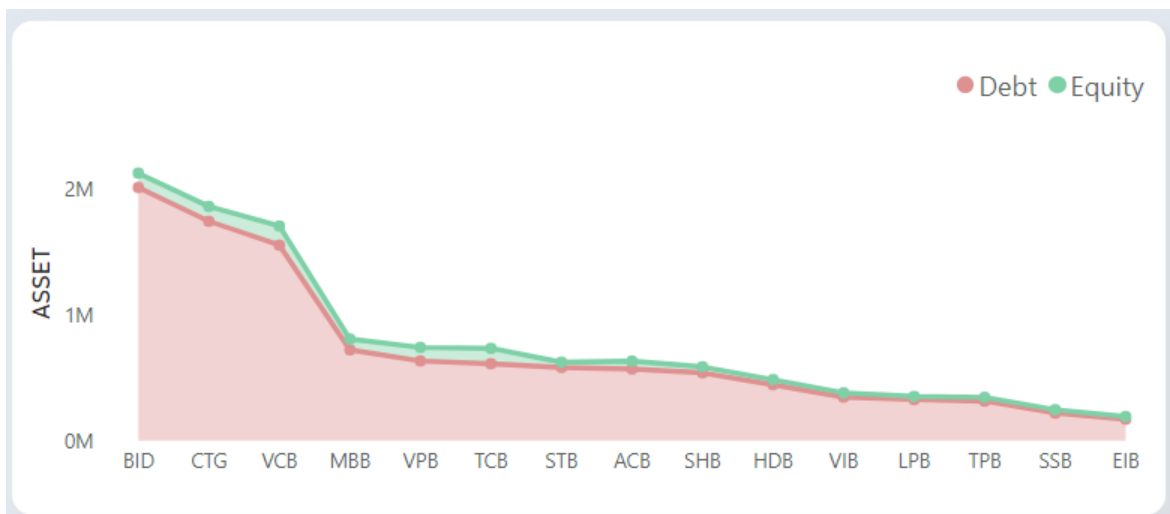
And add a scatter plot with NetProfitMargin on the X-axis, Revenue on the Y-axis, and Marker Size as Asset. This chart helps me identify stocks that operate well (located in the upper-right part with high X values), have a large scale of business operations (located in the upper part with high Y values), and possess significant assets (large Marker Size - indicating large Assets).

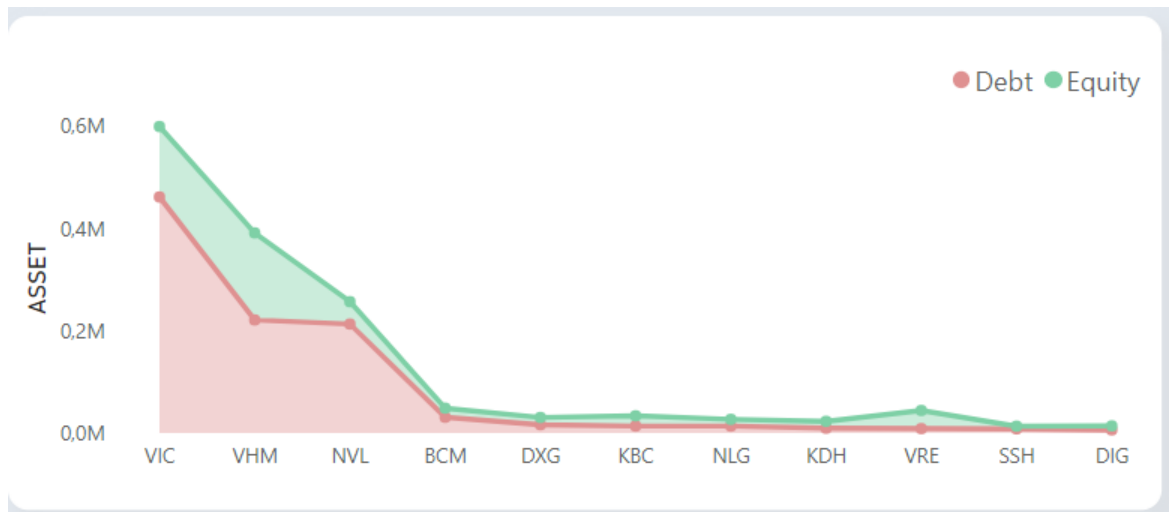


The final result.

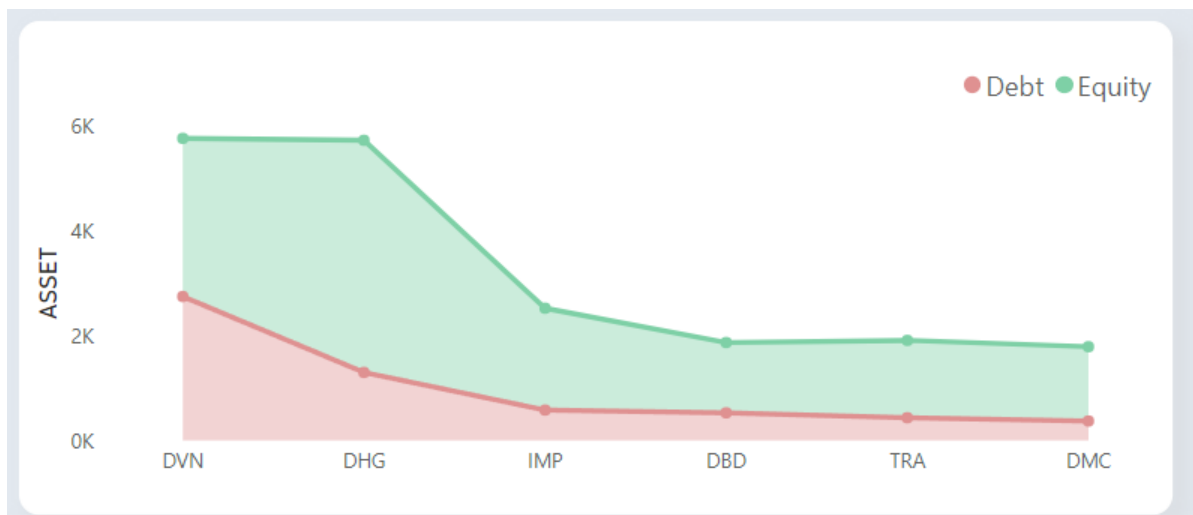
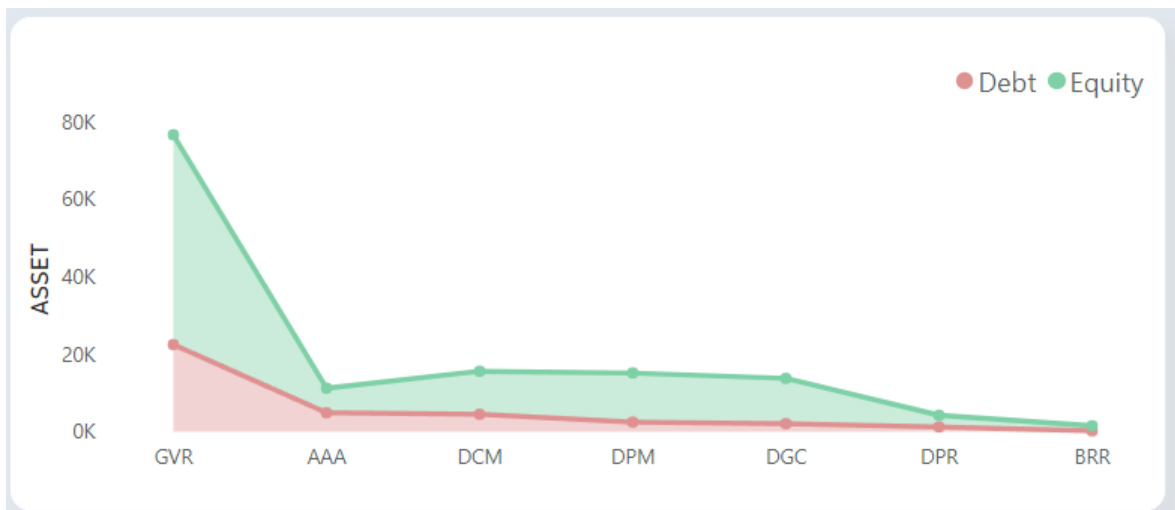
▼ Analyzing Visualization

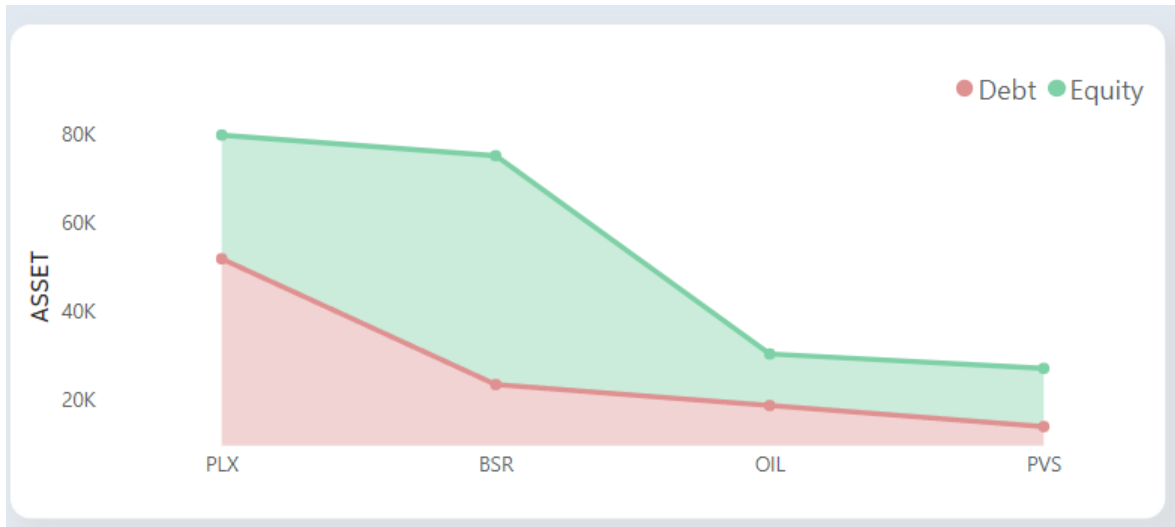
It can be observed that the Banking and Real Estate industries currently have high Debt ratios.



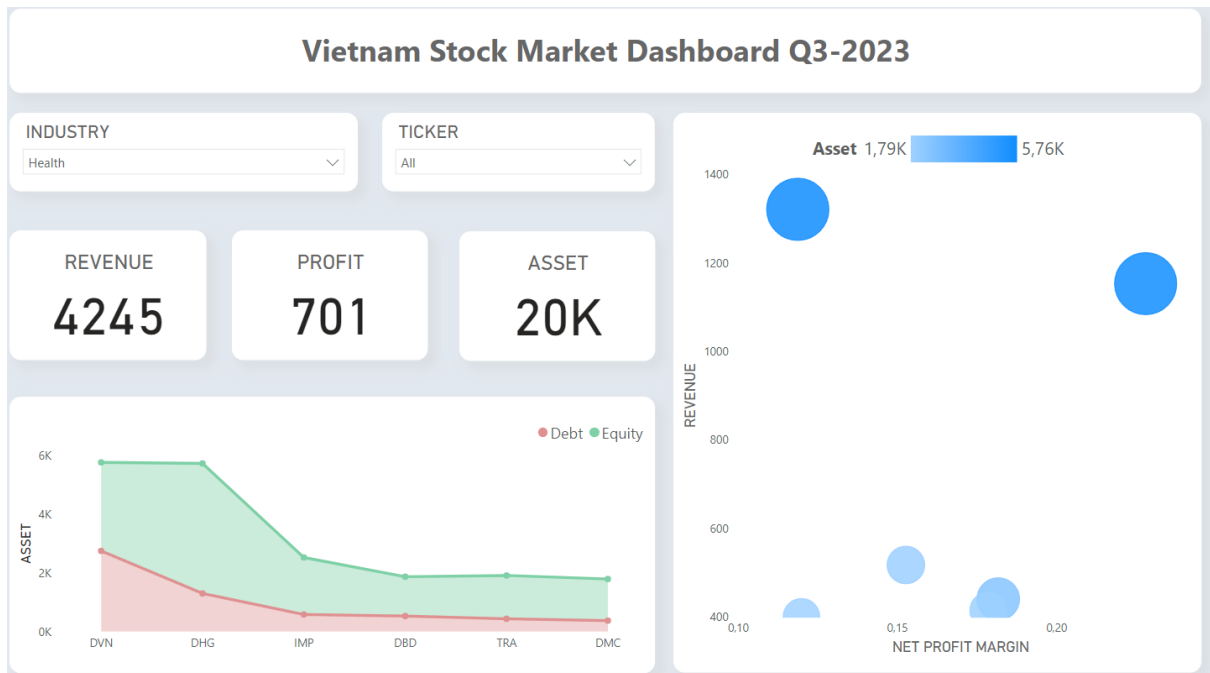


On the contrary, the Chemicals, Health, and Oil & Gas industries have low Debt ratios.

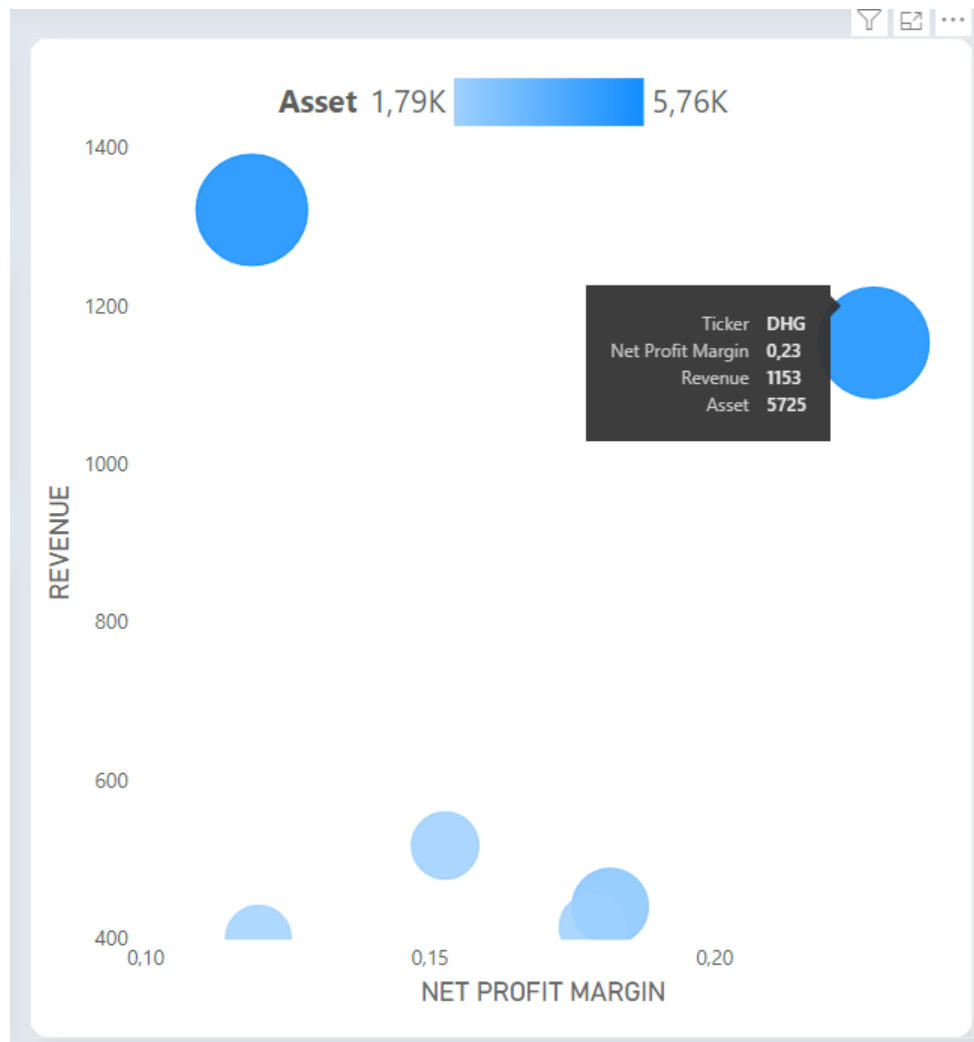




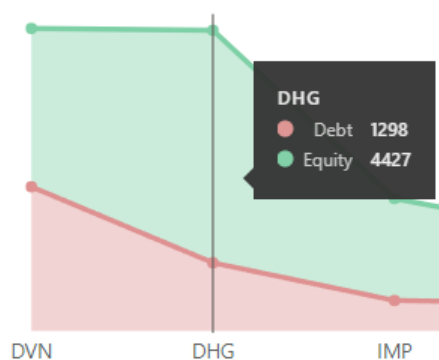
When examining each industry, and pausing at the Health industry.



I discovered that DHG (Hau Giang Pharmaceutical Joint Stock Company) has the highest Revenue, Net Profit Margin, and Asset among the stocks in the same industry group.



Furthermore, DHG also has a relatively low Debt/Equity ratio. This indicates that the company does not have an excessive amount of Debt compared to its Equity.



I will add DHG to my list of potential stocks to monitor further.

▼ Concluding

The strength as well as the weakness of this project lies in its data quantity.

As can be seen from the beginning, I have gathered quite a limited set of attributes, making it challenging for in-depth analysis.

However, this Dashboard has the ability to provide a quick and focused summary of some crucial information, suitable for an overall overview.

Another point for improvement is that I should obtain data for Q2 or Q3 of the previous year for easier comparison.