# FINAL ESSAY

## Application software packages in finance

Topic: **VALUING A COMPANY – PRICE TO SALES WITH PYTHON ( A CASE STUDY OF TESLA TESLA'S ASTRONOMICAL VALUATION)**


**Lecturer** ***M. Fin. Ngo Phu Thanh***

**Student** ***Nguyen Tuan Hung***

***K194141723***

**Class** ***211CN08***


*Ho Chi Minh City, 24th January 2022*

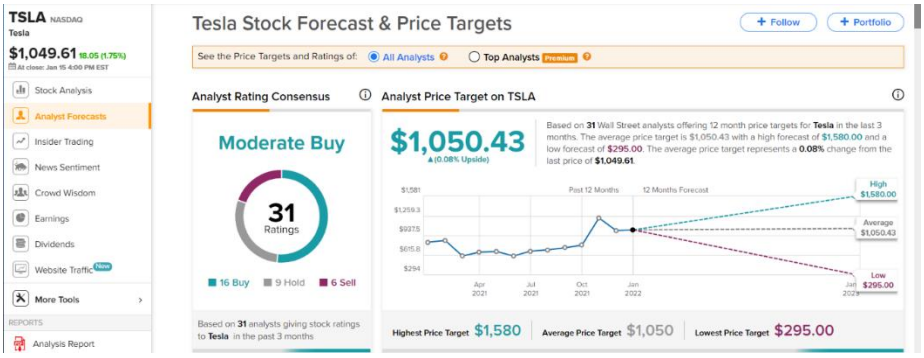
# Contents

## 1. Introduction

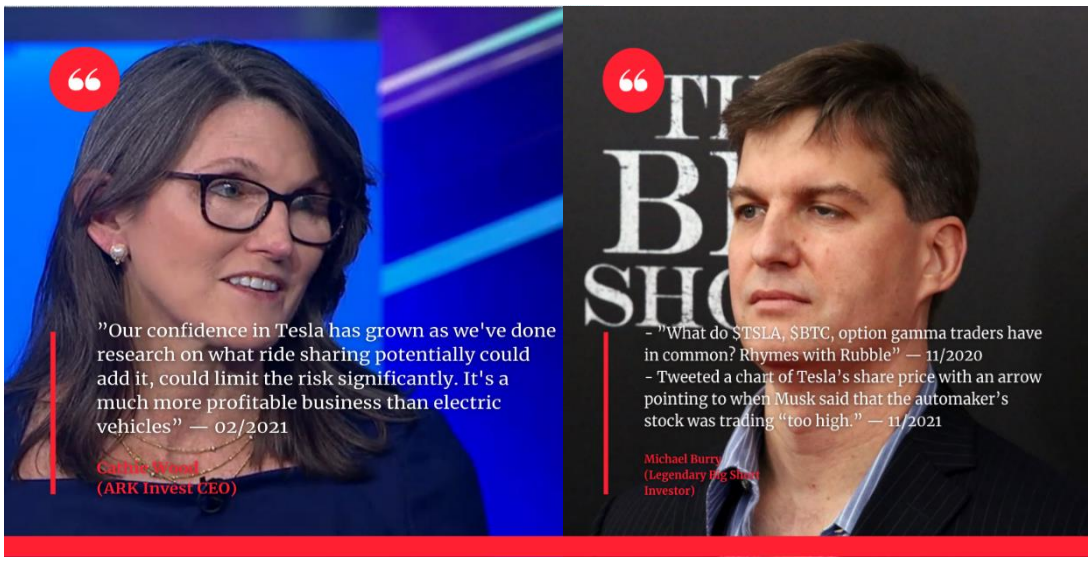
Tesla is an electric car company that went public on June 29, 2010 and a unique company in so many ways. Ever since their IPO Tesla's stock price has been at the center of many interesting moments in the financial world. From Elon Musk (Tesla's CEO) jokingly selling "shorts" to mock short sellers on wallstreet to again Elon Musk himself saying that Tesla's stock is overpriced in his opinion.

Forget about the product, the policy effect, and the leadership for a moment. Tesla Inc (TSLA) itself is unquestionably one of **the most polarizing among investors and traders**. Almost everyone has a strong opinion regarding TSLA and where the price is targeting.

On TipRanks, the 1-year prediction of this stock price ranges from a low of $295 to a high of $1,580. That is a mind-boggling **5.4x** difference.




Even when we seek advice from the most famous experts of our time, we receive strongly polarized views.



In 2021, Tesla stock hasn't been completely straight up. After reaching all-time highs in early November 2021, CEO Elon Musk stated that he intended to sell a bunch of the stock to pay taxes on.... all of his TSLA stock option gains! That caused a tumble for a week or so. However, beginning on December 21st, the stock began to rally even more dramatically. In less than 10 trading days, it was up 300 points, clocking in a gain of over $300 billion dollars in market cap!


Tesla has always been a company whose moves are unpredictable since last year: shocking acquisition of $1.5 billion in Bitcoin, Elon Musk's controversial tweets, and news about their blowout earnings in November 2021. In the 2022 New Year Greeting, TSLA continues to be a company that is difficult to grasp. How does a stock jump 13% on the first trading day of the new year? Given that everyone seems to be on the waiting list for a Model 3, X, or Y (Tesla's models of electric cars), is it any surprise they beat all Wall Street expectations?

This motivated me to make insights from the actual key metrics of Tesla's valuations and how they compare to other stocks. My goal is to know more about how exactly outrageous Tesla's stock price is in comparison to other big cap companies.

## 2. Methodology (Scraping with Python Finance API and Valuing a company based on Price to Sales Ratio)

2.1. *Filtering and putting together a basket of companies to compare Tesla against*

I've chosen a list of 60 companies from the Technology, Healthcare and Automotive industry. What all of these companies have in common is their market capitalizations are mostly greater than $100 billion, classifying them as large caps. The reason I wanted to add both the Technology and Automotive companies is that Tesla is frequently considered to contain both. By the way, I also wanted to include a completely different area (Healthcare) to give it some more room for comparison. The following is a complete list of the ticker symbols utilized in the analysis:

```
GOOGL, AAPL, FB, MSFT, AMZN, TSLA, NVDA, ASML, NFLX, TCEHY, BABA, TXN, SONY, INTC, SHOP, QCOM, INTU,
MRK, ISRG, MDT, AMGN, AZN, JNJ, CRM, AVGO, UNH, IBM, TSM, NVS, ABBV, ADBE, CSCO, CVS, AMAT, BMY, SNY,
NOW, LRCX, ORCL, SAP, MU, ACN, TMO, SYK, PFE, ANTM, JD, ABNB, F, BYDDF, SSUN.F, AMD, INFY, LLY, ABT,
NVO, DHR, GSK, ADI, TM
```

2.2**.** *Process of calculating Price to Sales and comparing key valuation metrics of Tesla with other companies with Python* 2.1 Why use Price to Sales ratio?Investors are always seeking ways to compare the value of stocks and Price to sales is a common tool used by them. The **price to sales ratio(P/S Ratio or PSR)** is computed by taking the market capitalisation of a company and divide by annual revenues. utilizes a company's market capitalization and revenue to determine whether the stock is valued properly. It also tells us how many years of revenues are required to cover the current market value of a company.

I've decided to use **market capitalization** and **PSR** (Price-to-sales ratio) to better comprehend Tesla's valuation. The PSR of a company is computed by taking the market capitalisation of a company and divide by annual revenues.

Differently than the **P/E** (price to earnings ratio or **P/E ratio**) can be used for companies with no earnings at all (In a highly cyclical industry such as semiconductors, there are years when only a few companies produce any earnings). For this reason, I chose to use price-to-sales instead of the P/E ratio to determine how much they are paying for a dollar of the company's sales rather than a dollar of its earnings. If a company's earnings are negative, the P/E ratio is not optimal since it will not be able to value the stock because the denominator is less than zero. PSR is very useful for growth companies which may not get profits during the growth years.

The PSR can be used for spotting recovery situations or for double-checking that a company's growth has not become overvalued. If the P/S ratio is lower than comparable companies in the same industry that is profitable, investors might consider following the stock due to the low valuation. Besides, it is also important to compute the industry average price to sales ratio of similar industry companies to use it as a reference.

However, sales by its own may not be sufficient to make any investment decisions. The P/S ratio needs to be used with other financial ratios and metrics when determining whether a stock is valued properly. I decided to compute each company's gross profit ratio and use it in conjunction with the price to sales ratio. **The gross profit ratio** informs us how much profit a company generates on its cost of sales. A business with higher gross profit margins might devote more resources to research and development in order to accelerate the company's growth. Alternatively, it might pay out greater profits to shareholders.

Gross Profit Margin = (Sales – Cost of Goods Sold)/Sales

**3. Process (Step-by-step)**

3.1. *Scraping, Cleaning, Integrating, Selecting and Transforming Data.*

After reading theories about PSR, I calculated the price to sales ratio and gross profit ratio for a bunch of companies in the above list. As previously mentioned, I limited my analysis to companies with more than 100 billion dollars market capitalisation.

For my analysis with Python, I used **fmpcloud**, a fantastic financial API, to extract the data. By opening an account with them, we just get a free API with a daily call limit of 250. After I've made a list of the stocks I want to analyze, I could iterate over each stock to get the essential metrics. I have used Jose Manu's code as a reference. Here is my script to step by step:

- First, I got all the technological, healthcare and automative companies and add them to a list. I used below endpoint passing as parameters for the sector and market capitalisation.

```
import requests
import pandas as pd

import requests

demo= '4460dee503ef783e805ec4e77143183d'# A free API key for a daily limit of 250 calls

companies = requests.get(f'https://fmpcloud.io/api/v3/stock-screener?\
sector=technology&marketCapMoreThan=100000000000&limit=100&apikey={demo}')
companies2 = requests.get(f'https://fmpcloud.io/api/v3/stock-screener?\
industry=internet&marketCapMoreThan=1000000000&limit=100&apikey=37aa64183b9f75e5e7d721fee0595c16')
companies3 = requests.get(f'https://fmpcloud.io/api/v3/stock-screener?\
industry=auto&marketCapMoreThan=100000000000&limit=100&apikey={demo}')
companies = companies.json()
companies2 = companies2.json()
companies3 = companies3.json()


technological_companies = []
internet_companies = []
automotive_companies = []

for item in companies:
  technological_companies.append(item['symbol'])
for item2 in companies2:
  internet_companies.append(item2['symbol'])
for item3 in companies3:
  automotive_companies.append(item3['symbol'])

print(technological_companies)
print(internet_companies)
print(automotive_companies)
# ['AAPL', 'MSFT', 'TSM', 'NVDA',...
# ['GOOG', 'GOOGL', 'AMZN', 'FB',...
# ['TSLA', 'TM', 'VWAGY', 'VWAP.BR',...
```

- I didn't combine all the items from all the above lists because many companies of internet_companies list are not tech companies. After considering all resources, I decided to filter and select companies what I really expect to get (Technology, Healthcare and Automotive industries).

```
# Finally basket of companies
large_caps_companies = ['GOOGL','AAPL','FB','MSFT','AMZN','TSLA','NVDA','ASML','NFLX','TCEHY','BABA','TXN','SONY','INTC','SHOP',
                        'QCOM','INTU','MRK','ISRG','MDT','AMGN','AZN','JNJ','CRM','AVGO','UNH','IBM','TSM','NVS','ABBV','ADBE',
                        'CSCO','CVS','AMAT','BMY','SNY','NOW','LRCX','ORCL','SAP','MU','ACN','TMO','SYK','PFE','ANTM','JD','ABNB'
                        'F','BYDDF','SSUN.F', 'AMD', 'INFY','LLY','ABT','NVO','DHR','GSK','ADI','TM']

print(*large_caps_companies, sep=", ")
```

```
GOOGL, AAPL, FB, MSFT, AMZN, TSLA, NVDA, ASML, NFLX, TCEHY, BABA, TXN, SONY, INTC, SHOP, QCOM, INTU, MRK, ISRG, MDT, AMGN, AZN,
JNJ, CRM, AVGO, UNH, IBM, TSM, NVS, ABBV, ADBE, CSCO, CVS, AMAT, BMY, SNY, NOW, LRCX, ORCL, SAP, MU, ACN, TMO, SYK, PFE, ANTM,
JD, ABNB, F, BYDDF, SSUN.F, AMD, INFY, LLY, ABT, NVO, DHR, GSK, ADI, TM
```

- Then I looped through the list of stocks to make an HTTP request to the API and retrieve Income Statement data. The revenue and gross profit ratios are calculated by

parsing the response. The data of financial statements for a basket of stocks I used is in the last fiscal year (the most recent 12-month accounting period).

- Next, I retrieved the most current market capitalisation.

- After that, I calculated the price to sales ratio and add them to an empty dictionary.

```
# Import finance API
demo = "69a98313d6b4e57bcc94bdac584e016c"
# Finally basket of companies
large_caps_companies=['GOOGL','AAPL','FB','MSFT','AMZN','TSLA','NVDA','ASML','NFLX','TCEHY','BABA','TXN','SONY','INTC','SHOP',
                      'QCOM','INTU','MRK','ISRG','MDT','AMGN','AZN','JNJ','CRM','AVGO','UNH','IBM','TSM','NVS','ABBV','ADBE',
                      'CSCO','CVS','AMAT','BMY','SNY','NOW','LRCX','ORCL','SAP','MU','ACN','TMO','SYK','PFE','ANTM','JD','ABNB',
                      'F','BYDDF','SSUN.F', 'AMD', 'INFY','LLY','ABT','NVO','DHR','GSK','ADI','TM']
# Get historical stock data
# Loop through the list of stocks(large_caps_companies), making an HTTP call to the API to get Income Statement data
pricetosales = {}
for item in large_caps_companies:
    try:
      #annual income statement
      IS = requests.get(f'https://fmpcloud.io/api/v3/income-statement/{item}?apikey={demo}')
      IS = IS.json()
      Revenue = IS[0]['revenue']
      grossprofitratio = IS[0]['grossProfitRatio']
      operatingprofitratio = IS[0]['operatingIncomeRatio']
      netincomeratio = IS[0]['netIncomeRatio']
      #most recent market capitliazation
      MarketCapit = requests.get(f'https://fmpcloud.io/api/v3/market-capitalization/{item}?apikey={demo}')
      MarketCapit = MarketCapit.json()
      MarketCapit = MarketCapit[0]['marketCap']
      #company sector
      Sector = requests.get(f"https://fmpcloud.io/api/v3/profile/{item}?apikey={demo}")
      Sector = Sector.json()
      Sector_Name = Sector[0]["sector"]
      Industry = Sector[0]["industry"]
      Beta = Sector[0]["beta"]
      #growth rate
      Growth = requests.get(f"https://fmpcloud.io/api/v3/income-statement-growth/{item}?limit=40&apikey={demo}")
      Growth = Growth.json()
      Rev_growth = Growth[0]["growthRevenue"]
      EBITDA_growth = Growth[0]["growthEBITDA"]

      #Price to sales
      p_to_sales = MarketCapit/Revenue

      pricetosales[item] = {}
      pricetosales[item]["Beta"] = Beta
      pricetosales[item]['revenue'] = Revenue
      pricetosales[item]['Gross_Profit_ratio'] = grossprofitratio
      pricetosales[item]['price_to_sales'] = p_to_sales
      pricetosales[item]['Market_Capit'] = MarketCapit
      pricetosales[item]['sector'] = Sector_Name
      pricetosales[item]['industry'] = Industry
      pricetosales[item]["Operating_Profit_ratio"] = operatingprofitratio
      pricetosales[item]["Annual_Revenue_Growth"] = Rev_growth
      pricetosales[item]["Annual_EBITDA_Growth"] = EBITDA_growth
    except:
      pass
```

- Finally, I had my various key metrics for each of the companies. I think it would be better if I convert these data to a Pandas Dataframe with the aim of exploring some further analysis.

- After using the Pandas DataFrame method from_dict and passing the dictionary as argument, I had a Dataframe that fit my needs.

```
price_to_sales_df = pd.DataFrame.from_dict(pricetosales, orient='index')
```

- I have got various key metrics by scrapping with the above code:
    - Operational metrics (all annual): Gross Profit Ratio, Operating Income Ratio, Net Income Ratio, Revenue, Revenue Growth, EBITDA Growth.
    - Financial metrics: Market Capitalization, Price-To-Sales Ratio.

We can see a sample of our resulting Pandas DataFrame in below image.

| | Beta | revenue | Gross_Profit_ratio | price_to_sales | Market_Capit | sector | industry | Operating_Profit_ratio | Annual_Revenue_Grow |
|---|---|---|---|---|---|---|---|---|---|
| **TSM** | 1.015044 | 1339238500000 | 0.530996 | 0.482224 | 6.458126e+11 | Technology | Semiconductors | 0.436626 | 0.2516 |
| **MU** | 1.149518 | 27705000000 | 0.376214 | 3.311445 | 9.174357e+10 | Technology | Semiconductors | 0.224436 | 0.2925 |
| **NVDA** | 1.313735 | 16675000000 | 0.623448 | 34.931339 | 5.824801e+11 | Technology | Semiconductors | 0.264408 | 0.5272 |
| **BYDDF** | 0.584099 | 156597691000 | 0.193785 | 0.606032 | 9.490317e+10 | Consumer Cyclical | Auto Manufacturers | 0.043951 | 0.2259 |
| **NFLX** | 0.813482 | 29697844000 | 0.416366 | 5.928828 | 1.760734e+11 | Communication Services | Entertainment | 0.208584 | 0.1881 |
| **ADBE** | 1.003041 | 15785000000 | 0.881850 | 15.068557 | 2.378572e+11 | Technology | Software—Infrastructure | 0.361419 | 0.2266 |

- As we can see, I had one row of data for each stock code was listed. Here are the columns:

- Beta - a way of measuring a stock's volatility compared with the overall market's volatility
- revenue - the total amount of income generated by the sale of goods related to the company's primary operations
- Gross_Profit_ratio - a financial ratio that measures the performance and efficiency of a business
- price_to_sales - a valuation ratio that compares a company’s stock price to its revenues
- Market_Capit - the total value of all a company's shares of stock
- sector - an area of the economy in which businesses share the same or related business activity, product, or service
- industry - a group of companies with similarities in the business activities
- Operating_Profit_ratio - a performance ratio that reflects the percentage of profit a company produces from its operations

- Annual_Revenue_Growth - the change in the consolidated revenue for such year as compared to the prior year
- Annual_EBITDA_Growth - the change in the consolidated EBITDA for such year as compared to the prior year

Finally, let's do some steps of data cleaning, data integration, data selection and data transformation before analysis and drawing insights:

- First step is checking types and nan values of columns and rows if they are not what I really want

```
price_to_sales_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 60 entries, GOOGL to TM
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Beta                    60 non-null     float64
 1   revenue                 60 non-null     int64  
 2   Gross_Profit_ratio      60 non-null     float64
 3   price_to_sales          60 non-null     float64
 4   Market_Capit            60 non-null     float64
 5   sector                  60 non-null     object 
 6   industry                60 non-null     object 
 7   Operating_Profit_ratio  60 non-null     float64
 8   Annual_Revenue_Growth   60 non-null     float64
 9   Annual_EBITDA_Growth    60 non-null     float64
```

It seemed to have the correct type I want.

- Next, I will create a new function that return the formatted dataframe. It contains:

    - Ordering the dtframe by "marketcap_in_b" column for comparison
    - Converting market cap that’s being in Scientific Format  to integer to view more easily
    - Create a new column that contains symbols of index to reveal more information about the data points of graphs

```
def preparing_data(df):
    df.columns = df.columns.str.lower()
    df = df.rename(columns={"price_to_sales": "PSR", #Ordering the dtframe by "marketcap_in_b" column for comparison
                            "market_capit": "marketcap_in_b"}).sort_values(by="marketcap_in_b", ascending=False)
    # Because market cap is being in Scientific Format => convert it to integer to view more easily
    df["marketcap_in_b"] = df["marketcap_in_b"].astype('int64')
    # To reveal more information about the data points of graphs => create a new column that contains symbols of index
    df['Symbol'] = df.index.to_list()
    temp_cols=df.columns.tolist()
    new_cols=temp_cols[-1:] + temp_cols[:-1]
    df=df[new_cols]
    return df
```

- I applied that function on df dataframe to avoid the case where changing df also changes price_to_sales_df

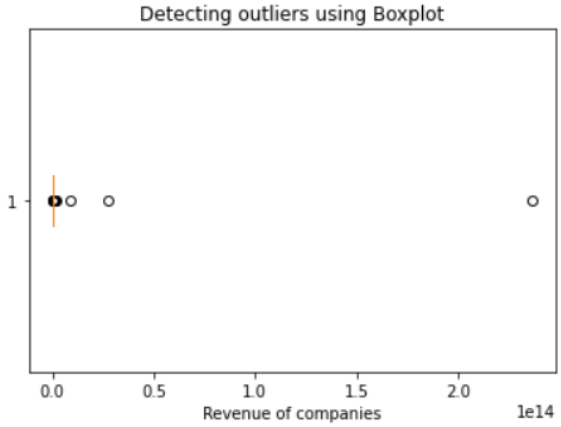
```
df = preparing_data(price_to_sales_df) #To avoid the case where changing df also changes price_to_sales_df
df.head()
```

| | Symbol | beta | revenue | gross_profit_ratio | PSR | marketcap_in_b | sector | industry | operating_profit_ratio | annual_revenue_gr |
|---|---|---|---|---|---|---|---|---|---|---|
| **AAPL** | AAPL | 1.202736 | 365817000000 | 0.417794 | 7.216629 | 2639965645250 | Technology | Consumer Electronics | 0.298529 | 0.33 |
| **MSFT** | MSFT | 0.862138 | 168088000000 | 0.689258 | 13.237947 | 2225139966213 | Technology | Software—Infrastructure | 0.423005 | 0.17 |
| **GOOGL** | GOOGL | 1.058286 | 182527000000 | 0.535784 | 9.513440 | 1736459707781 | Communication Services | Internet Content & Information | 0.263424 | 0.12 |
| **AMZN** | AMZN | 1.095793 | 386064000000 | 0.244105 | 3.797567 | 1466103959525 | Consumer Cyclical | Internet Retail | 0.062627 | 0.37 |
| **TSLA** | TSLA | 1.977192 | 31536000000 | 0.210236 | 29.615733 | 933961770240 | Consumer Cyclical | Auto Manufacturers | 0.063229 | 0.28 |

- In my plan with graph for insight 3, I will make in the Visualize Insights. I set up bubble size = revenue. But I find some outliers by using visualization. Outliers are data points that don't belong to a certain population. It is an abnormal observation that lies far away from other values. Using boxplot, I detect 3 companies are outliers because their revenue are too high compared to other companies. So, I decide to remove them. We can see more details below:

```
#Detecting outliers using Boxplot: insight3
df_3 = df.copy()
df_3 = df["revenue"].sort_values(ascending=False)
plt.boxplot(df_3, vert=False)
plt.title("Detecting outliers using Boxplot")
plt.xlabel('Revenue of companies')
```

```
Text(0.5, 0, 'Revenue of companies')
```



```
# Data for insight 3
df_3 = df.copy().sort_values(by="revenue", ascending=False)
df_3 = df_3.iloc[3:] #Delete the first 3 companies
df_3.head()
```

| | Symbol | beta | revenue | gross_profit_ratio | PSR | marketcap_in_b | sector | industry | operating_profit_ratio | annual_revenue_g |
|---|---|---|---|---|---|---|---|---|---|---|
| **TSM** | TSM | 1.015044 | 1339238500000 | 0.530996 | 0.487761 | 653228543877 | Technology | Semiconductors | 0.436626 | 0.2 |
| **JD** | JD | 0.742017 | 745801886000 | 0.146297 | 0.149412 | 111431590947 | Consumer Cyclical | Internet Retail | 0.068140 | 0.2 |
| **BABA** | BABA | 0.890015 | 717289000000 | 0.412782 | 0.454924 | 326312230922 | Consumer Cyclical | Internet Retail | 0.230839 | 0.4 |
| **TCEHY** | TCEHY | 0.513791 | 482064000000 | 0.459549 | 1.188674 | 573017022464 | Communication Services | Internet Content & Information | 0.373440 | 0.2 |
| **AMZN** | AMZN | 1.095793 | 386064000000 | 0.244105 | 3.797567 | 1466103959525 | Consumer Cyclical | Internet Retail | 0.062627 | 0.3 |

3.2. *Make sense of Tesla's high valuations in relation to its core business.*

I intended to use operational measures (profitability, scale, and growth) to make sense of Tesla's valuations after gathering the essential information regarding Tesla's valuation in comparison to other firms.

## 4. Analysis and data visualization to find insights

*Insight #1: Tesla stock's valuation is certainly too high. The semiconductor*. industry, on the other hand, is in a similar situation.

| | sector | marketcap_in_b | PSR | revenue |
|---|---|---|---|---|
| **AAPL** | Technology | 2652869820846 | 7.251904 | 365817000000 |
| **MSFT** | Technology | 2222587253089 | 13.222760 | 168088000000 |
| **GOOGL** | Communication Services | 1730452577495 | 9.480529 | 182527000000 |
| **AMZN** | Consumer Cyclical | 1446822293994 | 3.747623 | 386064000000 |
| **TSLA** | Consumer Cyclical | 947920983795 | 30.058377 | 31536000000 |
| **FB** | Communication Services | 843346179200 | 9.810344 | 85965000000 |
| **TSM** | Technology | 645812564060 | 0.482224 | 1339238500000 |
| **NVDA** | Technology | 582480080000 | 34.931339 | 16675000000 |
| **TCEHY** | Communication Services | 573043230863 | 1.188729 | 482064000000 |
| **UNH** | Healthcare | 434353429359 | 1.699089 | 255639000000 |

**Graph 1: Analyzing Top 10 Market Cap stocks**

The biggest companies of SaaS (Software as a service) provider has a revenue multiple of 10, which is where AAPL, GOOGL, and FB all lying around.

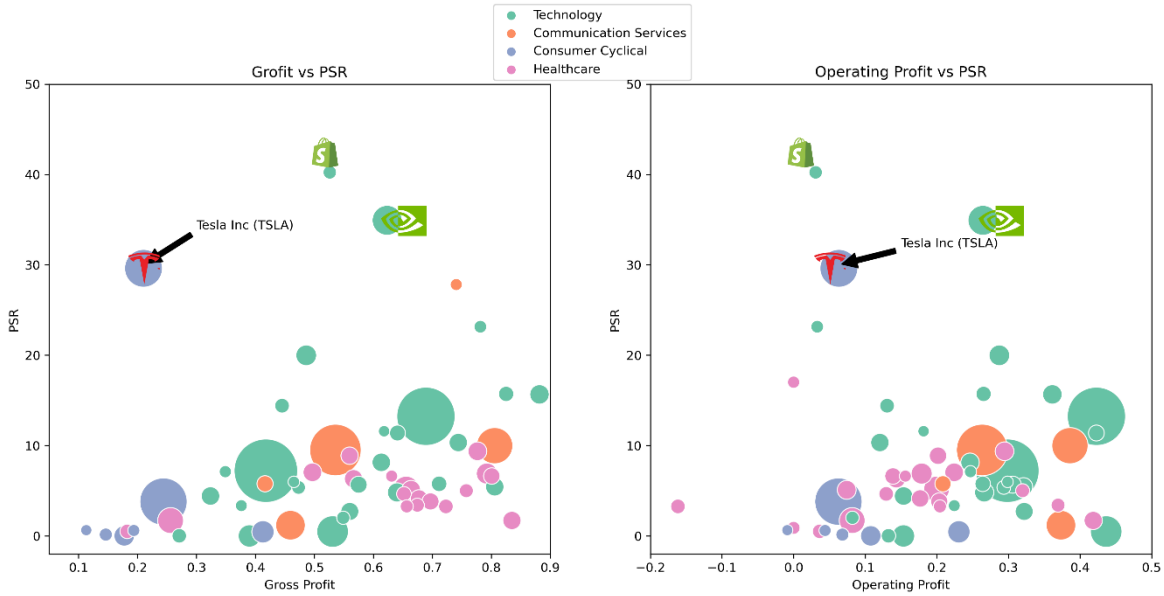
In comparison, Tesla's PSR is greater than 30 (**PSR of 30.05**), **which is thrice that of GAFAM** (4 most powerful tech companies, namely Google, Apple, Facebook, Amazon and Microsoft).

By the way, we can find that NVDA and ASML both have PSRs over 20s. This indicates that the semiconductor industry is likewise highly valuated. Healthcare, on the other hand, is undervalued.

*Insight #2: Definitely Tesla's profitability is not a reasonable justification of the high valuation.*

The below charts compare a company's PSR with its Gross Profit and Operating Profit Margin. Each bubble symbolizes a different firm.

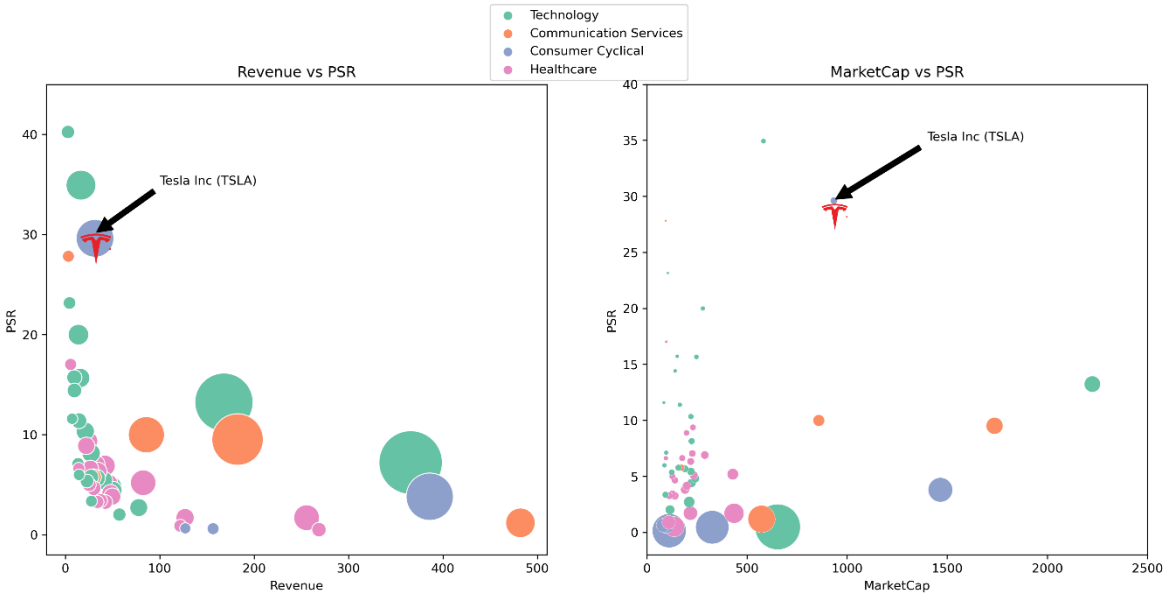
Profitability and PSR have a positive correlation, implying that the profitability is higher, the higher the valuations. **Tesla**, however, **goes against this trend and stands out as a clear outlier**. Other automotive industry businesses (blue) have comparable profitability, but their PSRs are 1/30 of Tesla's. By the way, there is one tech company with a PSR of more than 37. It's Shopify, which is really incredible.




**Graph 2: PSR vs Profitability, Bubble Size = Market Cap**

*Insight #3: Tesla's business size (market cap and revenue) is not also* considered *a justification for its high valuation.*

The below charts compare a company's PSR with its Revenue and Operating Profit. When analyzing at Revenue (left), we can see that Tesla's PSR is high when compared to other firms with similar revenue (~$30billion). Furthermore, Tesla has a high PSR given its market capitalization of more than ~$500bn.



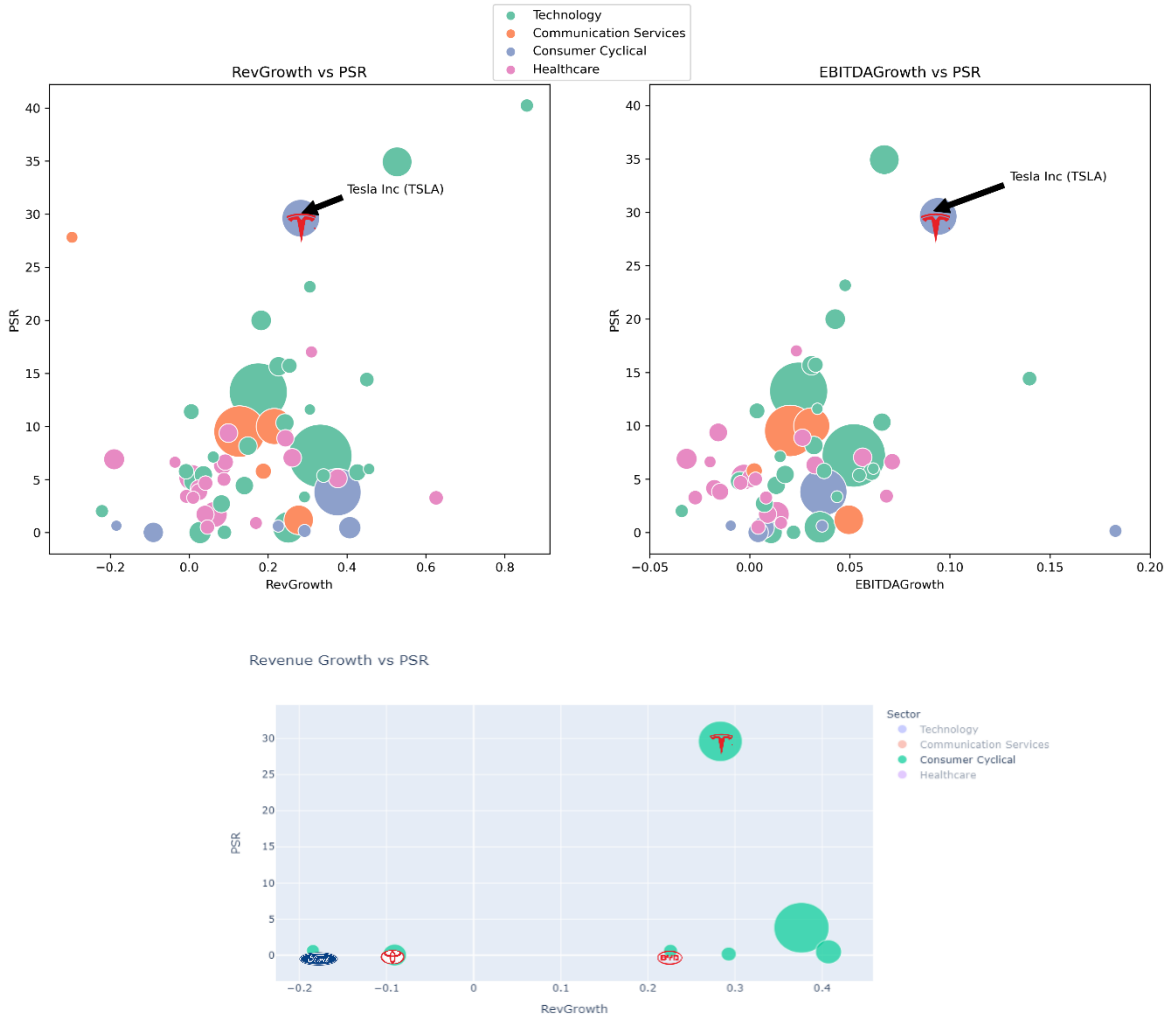

**Graph 3: PSR vs Revenue/Market Cap, Bubble Size = Market Cap (Left), Revenue (Right)**

*Insight #4: Tesla's high growth rate (in profits and revenue) could be a good justification for the company's high valuation.*

The charts below compare a firm's PSR to its revenue and EBITDA growth. Looking at Revenue Growth (left), it is clear that Tesla's revenue growth is relatively high when compared to other firms. In the automotive industry, most of companies experienced negative revenue growth, but Tesla is growing at a rate of ~30%. It's amazing to see Tesla's growth despite the epidemic and semiconductor scarcity.

When looking to EBITDA growth, **Tesla has been one of the companies with the highest**. I believe Tesla's valuation has risen significantly as a result of the company's

development of a profitable business model (I understand there is debate about this too due to the revenues in regulatory credits).




**Graph 4: PSR vs Revenue/EBITDA Growth, Bubble Size = Market Cap**

## 5. Final Conclusions

In conclusion, I believe the charts assist to demonstrate the extent to which Tesla's value differs from that of other firms. Tesla's valuation is definitely high with its current profitability and growth. As a result, it is a warning sign for investors to buy, especially for institutions.

However, there is no denying that Tesla's growth has been nothing short of remarkable. Looking at the big picture, Tesla is still in its early stages of development,

with most of its growth and technology yet to reveal. Vehicle ramp-up, autonomous technology, a solar energy business, and the creation of an AI platform are all still in the making.

The stock did just fine in 2021 after rising 743% in 2020. I believe upcoming earnings will continue to blow out expectations. In 2022, Wall Street is currently projecting deliveries of about 1.42 million vehicles, including 296,000 in the first quarter. Tesla has two new manufacturing facilities, in Texas and Germany, that will be ramping up production early in 2022. That would amount to another year of big growth with Tesla.

Yet, customer value lies in the eye of the beholder. We may either ignore the high valuations or invest in Elon's revolutionary growth. Tesla is always remain a company with strong polarizing.

**6. Reference**


- Valuing a company – Price to Sales Ratio with Python:
https://codingandfun.com/valuing-a-company-price-to-sales-ratio-with-python/

- How to Use Price-To-Sales Ratios to Value Stocks:
**https://www.investopedia.com/articles/fundamental/03/032603.asp**

- What Is a Good Price-to-Sales (P/S) Ratio?:

https://www.investopedia.com/ask/answers/032515/what-considered-favorable-price-sales-ratio.asp

- Bubble Charts in Python:

https://plotly.com/python/bubble-charts/

- Plotly Python Open Source Graphing Library Financial Charts:

https://plotly.com/python/financial-charts/