



**Lambton**  
College

A Report on:

Collecting data from Yahoo Finance using Yahoo Finance API

# INDEX

## Contents

INTRODUCTION .....	3
Panasonic Holdings Corporation (PCRFY) .....	8
Arbor Metals Corp. (ABR.V) .....	13
Solid Power, Inc. (SLDP) .....	18
Boyd Gaming Corporation (BYD): .....	24
Microvast Holdings, Inc. (MVST).....	29
Conclusion.....	34

## INTRODUCTION

As per project Requirements 5 Selected Companies are:

- 1) Panasonic Holdings Corporation (PCRFY)
- 2) Arbor Metals Corp. (ABR.V)
- 3) Solid Power, Inc. (SLDP)
- 4) Boyd Gaming Corporation (BYD)
- 5) Microvast Holdings, Inc. (MVST)

The objective of a report is to describe how the stock market information was obtained from Yahoo Finance using the Yahoo Finance API. The report will also contain confirmation procedures, thorough justifications of the answer, including snippets of source code, graph analysis, and justification of conclusions through the use of patterns and examples.

### Installing necessary libraries:

```
In [1]: pip install yahoo_fin
```

```
Requirement already satisfied: yahoo_fin in c:\users\hp\anaconda3\lib\site-packages (0.8.9.1)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: requests-html in c:\users\hp\anaconda3\lib\site-packages (from yahoo_fin) (0.10.0)
Requirement already satisfied: feedparser in c:\users\hp\anaconda3\lib\site-packages (from yahoo_fin) (6.0.10)
Requirement already satisfied: requests in c:\users\hp\anaconda3\lib\site-packages (from yahoo_fin) (2.31.0)
Requirement already satisfied: pandas in c:\users\hp\anaconda3\lib\site-packages (from yahoo_fin) (1.5.3)
Requirement already satisfied: sgmlib3k in c:\users\hp\anaconda3\lib\site-packages (from feedparser->yahoo_fin) (1.0.0)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas->yahoo_fin) (2022.7)
Requirement already satisfied: numpy>=1.21.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas->yahoo_fin) (1.23.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas->yahoo_fin) (2.8.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests->yahoo_fin) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests->yahoo_fin) (2022.12.7)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests->yahoo_fin) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests->yahoo_fin) (3.4)
Requirement already satisfied: w3lib in c:\users\hp\anaconda3\lib\site-packages (from requests-html->yahoo_fin) (1.21.0)
Requirement already satisfied: parse in c:\users\hp\anaconda3\lib\site-packages (from requests-html->yahoo_fin) (1.19.1)
Requirement already satisfied: fake-useragent in c:\users\hp\anaconda3\lib\site-packages (from requests-html->yahoo_fin) (1.2.1)
Requirement already satisfied: pypeteer>=0.0.14 in c:\users\hp\anaconda3\lib\site-packages (from requests-html->yahoo_fin) (1.0.2)
```

```
In [2]: pip install yfinance
```

```
Requirement already satisfied: yfinance in c:\users\hp\anaconda3\lib\site-packages (0.2.28)
Requirement already satisfied: lxml>=4.9.1 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (4.9.1)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (2.3.8)
Requirement already satisfied: numpy>=1.16.5 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (1.23.5)
Requirement already satisfied: requests>=2.31 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (2.31.0)
Requirement already satisfied: pytz>=2022.5 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (2022.7)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (4.11.1)
Requirement already satisfied: appdirs>=1.4.4 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (1.4.4)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: pandas>=1.3.0 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (1.5.3)
Requirement already satisfied: html5lib>=1.1 in c:\users\hp\anaconda3\lib\site-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\hp\anaconda3\lib\site-packages (from beautifulsoup4->4.11.1->yfinance) (2.3.2.post1)
Requirement already satisfied: six>=1.9 in c:\users\hp\anaconda3\lib\site-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in c:\users\hp\anaconda3\lib\site-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (1.26.14)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from requests>=2.31->yfinance) (2022.12.7)
```

First, I configured the Python modules yfinance and yahoo\_fin, whose configurations make it easier to get information about finances from Yahoo Finance.

```
In [3]: #importing the get_data module|
        from yahoo_fin.stock_info import get_data
```

I deployed the get data method from the yahoo\_fin.stock\_info module for getting past stock price information for an individual stock or a group of stocks from Yahoo Finance.

```
In [28]: import yfinance as fy          #import yfinance for taking data from yahoo finance
        import matplotlib.pyplot as plot  #Import matplotlib for importing visulizations tools
        import numpy as np              #Import numpy for calculating functions
```

Import the yfinance library with alias fy for getting data from yahoo finance. Secondly, import matplotlib.pyplot for getting visualization and relations between each feature. Last, I Import numpy as np for calculating function in data.

### Highest Price Function:

```
#For getting high values from data we create visulizedDataHigh function which take company name as input
def visulizedDataHigh(company,code):
    fig, ax = plot.subplots(figsize= (12,6))          #use this for size of graph
    code['high'].plot(kind='bar',label="Highest price", color='blue')  #use plot function to draw bar chart

    #Now importing title, xlabel and ylabel.
    ax.set_title(f'Monthly Transaction Count of {company} in last 2 years')
    ax.set_xlabel('Month')
    ax.set_ylabel('Market price')
    ax.legend()

    #Use for Loop for separating highest company's stock price in last 2 years.

    for i, val in enumerate(code['high']):
        if val == max(code['high']):
            ax.bar(i, val,label="highest Stock Price", color='red')    #Separate color for highest price
            ax.legend()
        else:
            ax.bar(i, val, color='blue')          #Separate color except highest price
            ax.legend()
```

The given code is a Python function that visualizes the highest stock price of a company in the last two years. The function takes two arguments, the name of the company and the stock data of the company. The function uses the matplotlib library to plot a bar graph of the highest stock price of the company in the last two years. The highest stock price is plotted in red color, and the other stock prices are plotted in blue color. The function also adds a title, x-label, y-label, and legend to the graph. The for loop in the function separates the highest stock price from the other stock prices and plots them in different colors. The code uses the 'high' column of the stock data to plot the graph.

## Lowest Price Function:

```
#Now create function for the lowest stock price of company among all years.

def visulizedDataLow(company,code):
    fig, ax = plot.subplots(figsize= (12,6))
    code['low'].plot(kind='bar',label="Stock Price",color='blue')
    ax.set_title(f'Monthly Transaction Count of {company} in last 2 years')
    ax.set_xlabel('Month')
    ax.set_ylabel('Market price')
    ax.legend()

#Use for Loop for separating lowest panasonic stock price in last 2 years.
    for i, val in enumerate(code['low']):
        if val == min(code['low']):
            ax.bar(i, val,label="lowest Stock Price", color='#E3CF57')    #Separate color for highest
            ax.legend()
        else:
            ax.bar(i, val, color='blue')    #Separate color except high
            ax.legend()
```

The given code is a Python function that visualizes the lowest stock price of a company among the last two years. It uses the matplotlib library to plot a bar graph of the company's stock prices, with the x-axis representing the months and the y-axis representing the market price. The function takes two arguments: company and code, where company is the name of the company and code is a Pandas DataFrame containing the stock prices of the company. The function then uses a for loop to iterate over the code['low'] column and checks if the current value is the minimum value in the column. If it is, the function plots a bar with a different color to represent the lowest stock price. Otherwise, it plots a bar with the default color. Finally, the function sets the title, x-label, y-label, and legend of the plot.

## Correlation Matrix (Volume and Close):

```
#For getting correlation between volume and close price

import seaborn as sns
|
def findCorr(company_name,code):
    corr_matrix = code.corr()
    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
    plt.title(f"Correlation between volume and close price of {company_name}")
    corr = code['volume'].corr(code['close'])

    print(f'The correlation coefficient between volume and close is {corr:.2f}.')
```

First and for most I used seaborn python library for plotting correlation matrix.

The code is a Python function that takes in two parameters, the name of a company and a code that represents the company's stock data. The function calculates the correlation coefficient between the volume and close price of the stock data and displays it along with a heatmap that

shows the correlation matrix of all the variables in the data. The function uses the `corr()` method to calculate the correlation matrix and the `heatmap()` function from the Seaborn library to display the matrix. The title of the heatmap includes the name of the company. The correlation coefficient is calculated using the `corr()` method on the volume and close columns of the data. The function prints the correlation coefficient with two decimal places.

### Most/least transactions:

```
#Transactions history
def transactionsHistory(company_name,code):

    #Set graph size here
    fig, ax = plot.subplots(figsize= (8,4))
    #plot line graph with marker with red color.
    ax.plot(code['open'], color='red', linewidth = 2)
    plt.title(f'Monthly Transaction Count of {company_name}')
    plt.xlabel('Month')
    plt.ylabel('Transaction Count')
    plt.show()

    # Find the month with the most and least transactions
    most_transactions_month = code['open'].idxmax().strftime('%B, %Y')
    least_transactions_month = code['open'].idxmin().strftime('%B, %Y')
```

The code is a Python function called `transactionsHistory` that takes two arguments: `company_name` and `code`. The function creates a line graph with markers using the `plot` library, where the x-axis represents the month and the y-axis represents the transaction count. The function then finds the month with the most and least transactions using the `idxmax()` and `idxmin()` functions, respectively, and formats the output to display the month and year in a string. Set the graph size using `fig, ax = plot.subplots(figsize= (8,4))`. Plot the line graph with markers using `ax.plot(code['open'], color='red', linewidth = 2)`. Add a title, x-label, and y-label to the graph using `plt.title()`, `plt.xlabel()`, and `plt.ylabel()`, respectively. Display the graph using `plt.show()`. Find the month with the most transactions using `code['open'].idxmax().strftime('%B, %Y')`. Find the month with the least transactions using `code['open'].idxmin().strftime('%B, %Y')`. The search results do not provide information on the specific data structure of the code argument, so it is unclear what the open column represents.

### Relation between opening and closing:

```
#Plot correlation between close price and open price

def openClose(company_name,code):
    # Plotting a line plot of opening and closing prices
    plt.figure(figsize=(8,4))
    plt.plot(code['open'], label='Open',color = 'blue')
    plt.plot(code['close'], label='Close', color = 'red')
    plt.title(f'Opening and Closing Prices for {company_name}')
    plt.xlabel('Date')
    plt.ylabel('Price')
    plt.legend()
    plt.show()

    # Calculate the correlation coefficient between opening and closing prices
    corr = code['open'].corr(code['close'])

    print(f'The correlation coefficient between opening and closing prices is {corr:.2f}.')
```

The given code defines a function `openClose` that takes in two parameters, `company_name` and `code`. The function plots a line plot of opening and closing prices of the given code data, with the opening prices plotted in blue and the closing prices plotted in red. It also calculates the correlation coefficient between the opening and closing prices of the given code data and prints it out. The purpose of this code is to visualize and analyze the correlation between the opening and closing prices of a given stock.

## Panasonic Holdings Corporation (PCRFY)

- Explanation of data extraction process

```
#Input Panasonic Holdings Corporation Data set from yahoo_fin.stock_info
```

```
PCRFY = get_data("PCRFY",start_date = "20/09/2021",end_date="20/09/2023",  
                index_as_date = True, interval="1mo")
```

```
C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: Use  
rWarning: Parsing '20/09/2023' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=  
True for consistent parsing.
```

```
exec(code_obj, self.user_global_ns, self.user_ns)  
C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: Use  
rWarning: Parsing '20/09/2021' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=  
True for consistent parsing.  
exec(code_obj, self.user_global_ns, self.user_ns)
```

The given code is using the `get_data` function to retrieve data for the stock "PCRFY" between the dates "20/09/2021" and "20/09/2023" with a monthly interval. The retrieved data will be indexed by date [no relevant search result found].

- Validation Steps

```
In [274]: # For getting top 5 row we use head function. |  
PCRFY.head()
```

```
Out[274]:
```

	open	high	low	close	adjclose	volume	ticker
2021-10-01	12.37	13.66	11.20	12.41	12.41	8106100	PCRFY
2021-11-01	12.36	12.88	10.76	10.90	10.90	8551500	PCRFY
2021-12-01	10.89	11.59	10.50	10.99	10.99	8005700	PCRFY
2022-01-01	10.99	11.80	10.55	11.02	11.02	6105400	PCRFY
2022-02-01	11.37	11.37	9.91	10.51	10.51	5856900	PCRFY

Display the data that I have fetch using the API.



Time Period:
Sep 26, 2022 - Sep 26, 2023

Show:
Historical Prices

Frequency:
Monthly

Apply

Currency in USD

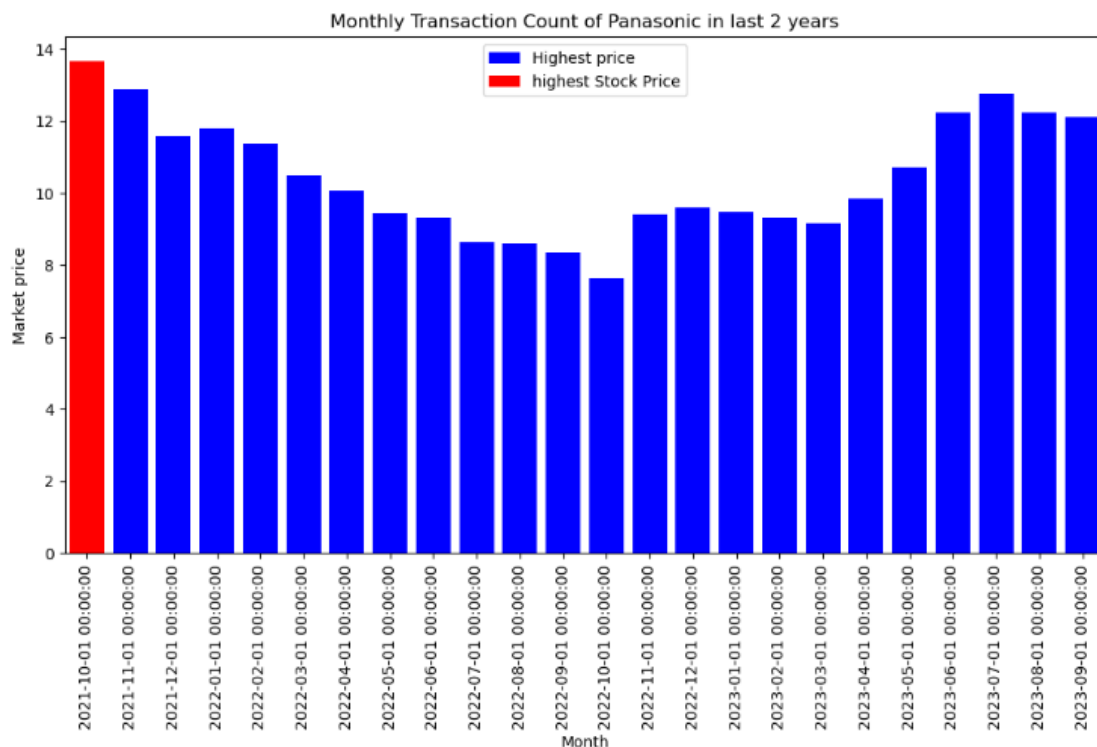
Download

Date	Open	High	Low	Close*	Adj Close**	Volume
Sep 01, 2023	11.76	12.12	11.23	11.60	11.60	2,169,800
Aug 01, 2023	12.25	12.25	10.49	11.54	11.54	3,670,400
Jul 01, 2023	12.30	12.75	11.69	12.33	12.33	3,170,700
Jun 01, 2023	10.45	12.24	10.43	12.24	12.24	5,191,800
May 01, 2023	9.45	10.72	9.45	10.40	10.40	3,487,900
Apr 01, 2023	9.02	9.85	8.95	9.38	9.38	2,758,900
Mar 01, 2023	8.53	9.15	8.51	8.90	8.90	3,016,200
Feb 01, 2023	9.23	9.31	8.51	8.71	8.71	2,951,000
Jan 01, 2023	8.32	9.46	8.05	9.22	9.22	3,362,000
Dec 01, 2022	9.44	9.59	8.16	8.35	8.35	3,815,600

The above data was taken from the yahoo finance website, and by comparing both of the images, I can say that the data I fetched using the ticker “Panasonic Holding Corporation” is accurate.

- In which year the company has the highest / lowest prices

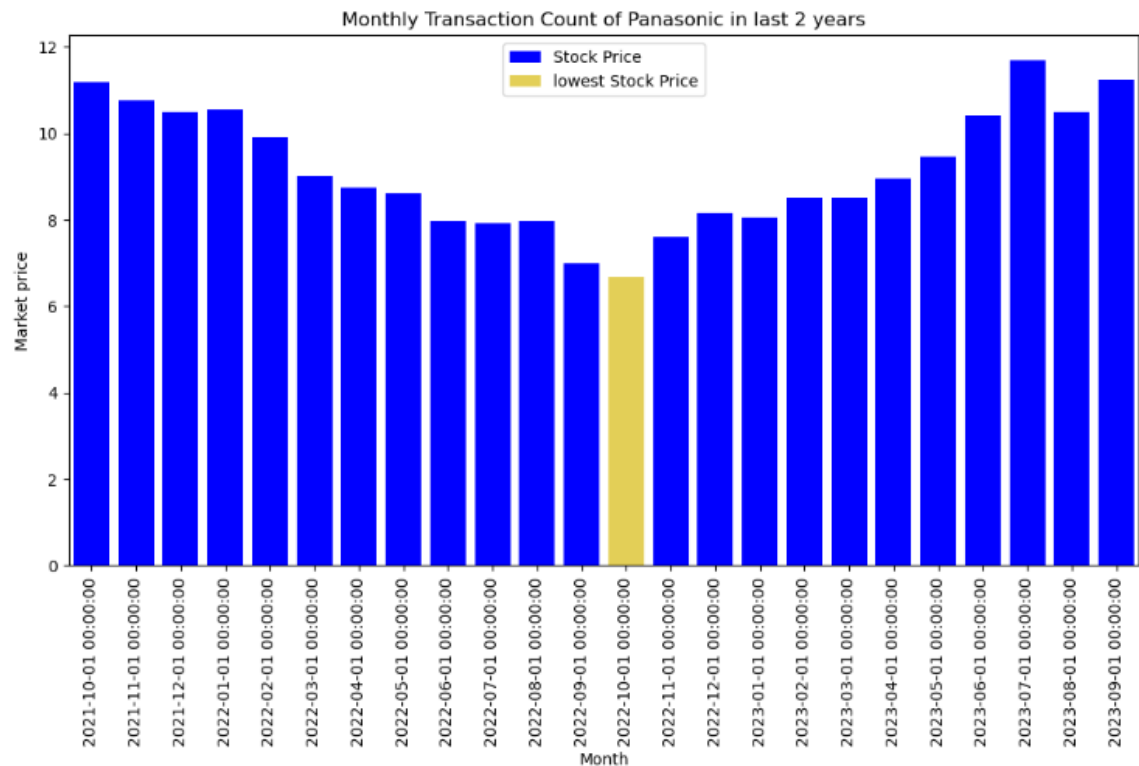
In [271]: `#Visualization of highest price in Panasonic company in Last 2 years  
visualizedDataHigh('Panasonic',PCRFY)`



The above code represents the function which we made in Introduction part, Here we can see that highest price of Panasonic stock was noted on 2021-10-01 and it's almost 13.5. Following years are falling until October 2022 and then again it has in rising state. Hopefully we will see highest price in couple of months.

- **Lowest price of Panasonic:**

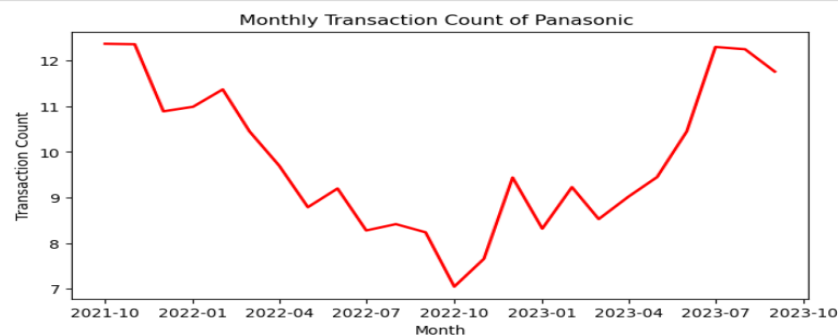
```
In [161]: #Visualization of Lowest price in Panasonic company in last 2 years
visulizedDataLow('Panasonic',PCRFY)
```



The following bar graph which is imported by matplotlib illustrated Panasonic's price in past 2 years. Where yellow bin represent the lowest price of stock other hand blue bins tells about normal price range of stocks.

- **What time of the year do your most/least transactions?**

```
In [191]: #Transaction History of Panasonic
transactionsHistory('Panasonic',PCRFY)
```

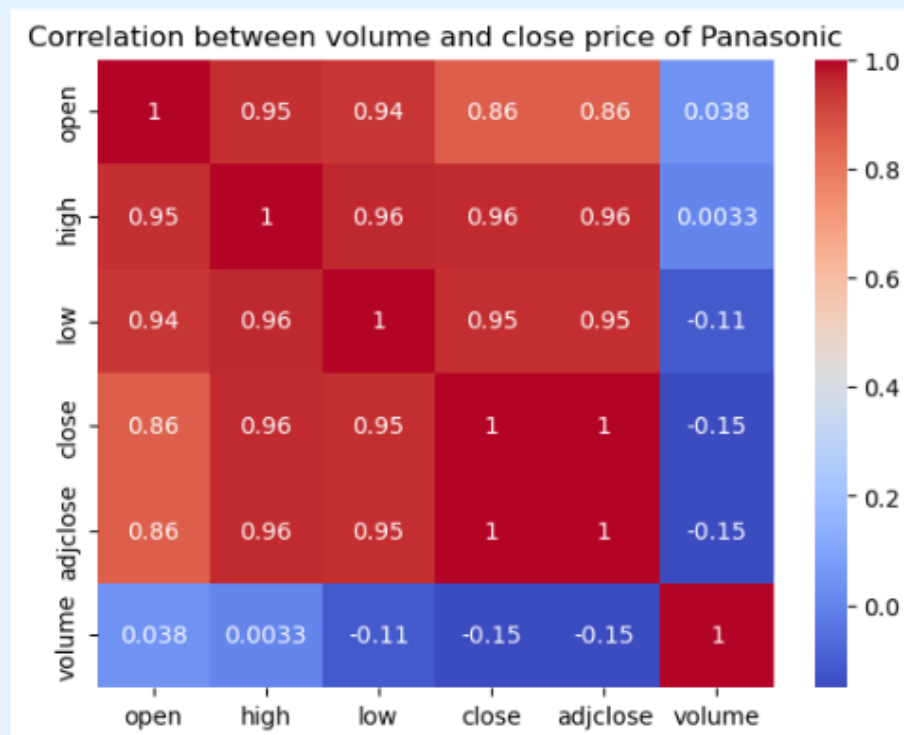


Above line graph represent most/least transaction in organization, where we can see that lowest stock price and lease amount of transaction both happened in same month, and most transaction noticed in October 2021 and July 2023.

- A relationship between Volume and close

```
In [239]: #To find correaltion between volume and close price, we used findcorr function.
findCorr('Panasonic',PCRFY)
```

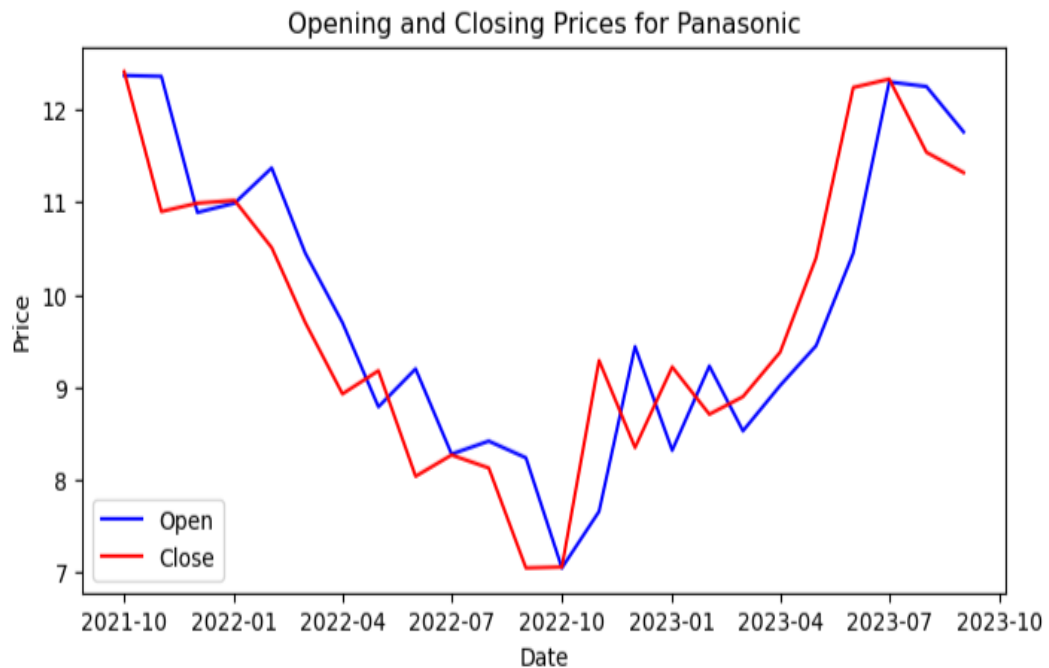
The correlation coefficient between volume and close is -0.15.



A correlation matrix which imported by use of one of the created function findCorr which takes Company\_name as Panasonic and code of company as PCRFY. Correlations between volume and close price of Panasonic is noticed negative 0.15 which highly suggest that both are not correlated with each other.

- A relationship between Opening and Closing

```
In [193]: #plot  
openClose('Panasonic',PCRFY)
```



The correlation coefficient between opening and closing prices is 0.86.

The correlation coefficient between starting and closing prices is 0.86, which is quite favorable. It suggests that, there is a significant propensity to cause an asset's closing and opening prices to fluctuate identically. It appears to be a strong prognostic correlation among these features in the collection whenever the initial cost is substantial and while the ending pricing is high.

## Arbor Metals Corp. (ABR.V)

- Explanation of data extraction process:

```
In [195]: #Input Arbor Metals Corporation Data set from yahoo_fin.stock_info

ABR = get_data("ABR.V",start_date = "20/09/2021",end_date="20/09/2023",index_as_date = True, interval="1mo")

C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2023' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2021' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Fetches the API for the Arbor Metal Corporation. Company using the ticker “ABR.V” for two years starting from Sep -22-2021 to Sep-22-2023 with the time interval of 1 month.

- Validation Steps

```
In [196]: #Get head of Arbor Metals Corp. stock data.
ABR.head()
```

```
Out[196]:
```

	open	high	low	close	adjclose	volume	ticker
2021-10-01	2.05	2.32	1.99	2.31	2.31	538600	ABR.V
2021-11-01	2.31	2.78	2.22	2.30	2.30	1844000	ABR.V
2021-12-01	2.41	2.75	2.41	2.61	2.61	479900	ABR.V
2022-01-01	2.62	2.79	2.10	2.51	2.51	1069900	ABR.V
2022-02-01	2.51	2.85	2.17	2.44	2.44	306900	ABR.V

Display the data that I have fetch using the API.

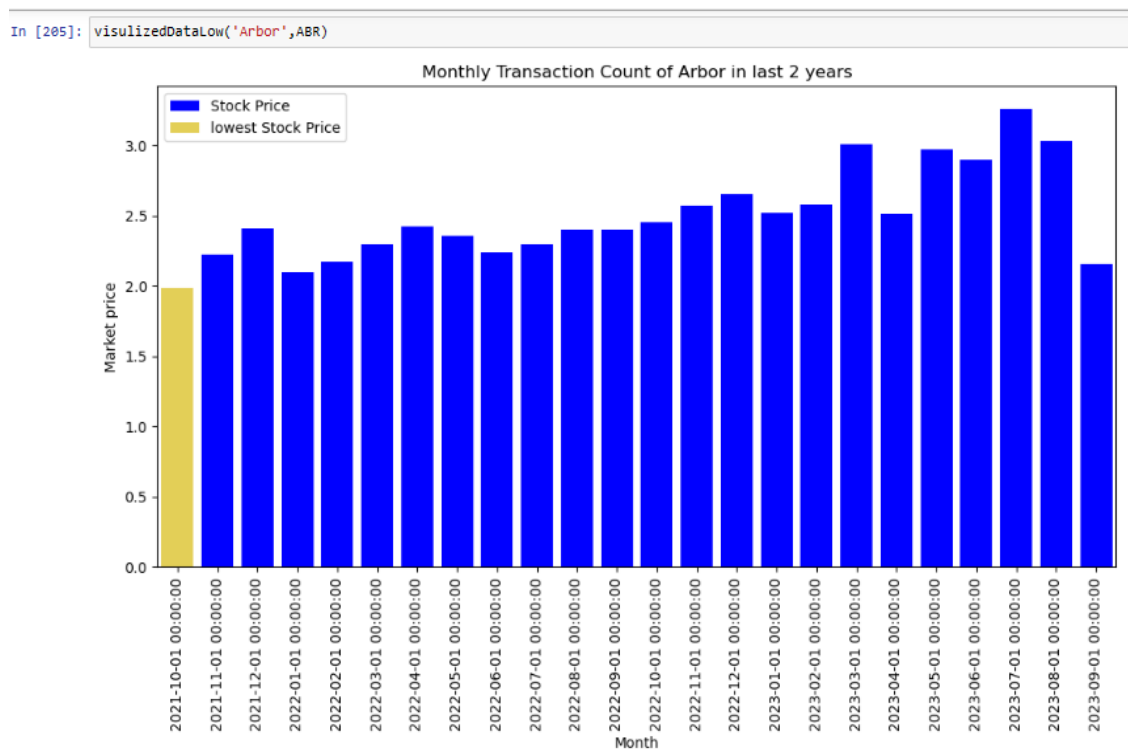


A stock price of the corporation is, nevertheless, gradually rising after 2023, and about to break their own high in near time.

- **Lowest price:**

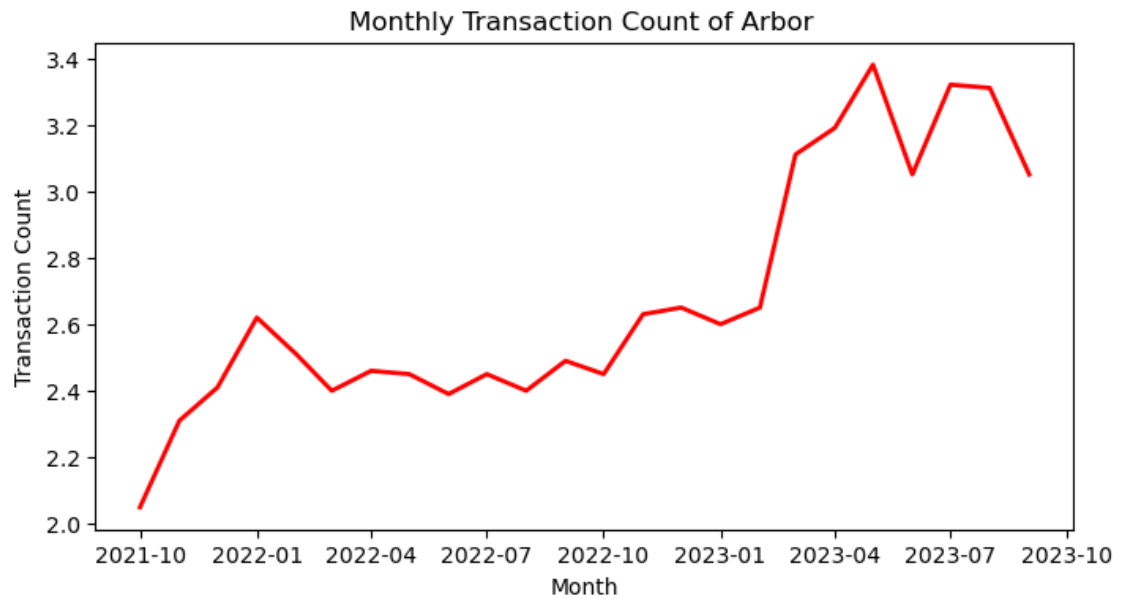
To below display the low price of the Arbor's graph, a bar graph was plotted using the Matplotlib program along with that we used sorted function and print tail of data which clarify that our graph is perfectly true .The graph fluctuates, as you can see, In starting periods price is decreasing continuously and after reached lowest on month of October 2021 it's started rising again.

This leads me to the conclusion that the company's price is highest in August 2021 and lowest in October 2021.



- What time of the year do your most/least transactions?

```
In [241]: transactionsHistory('Arbor',ABR)
```



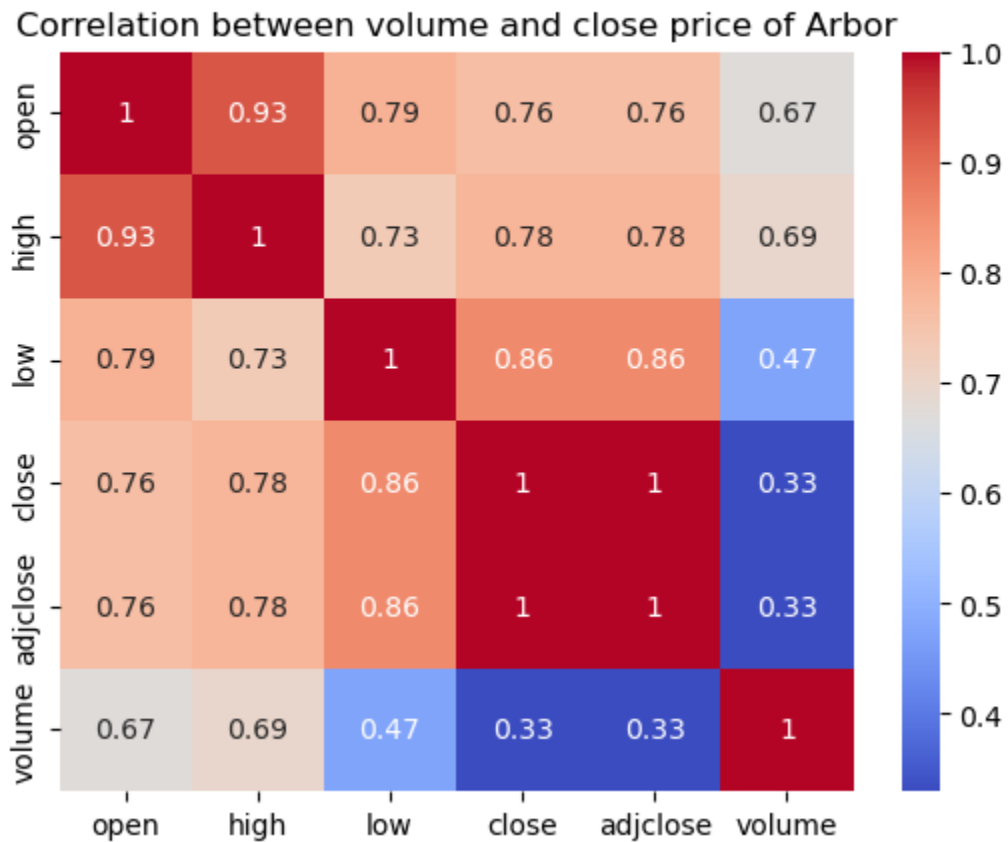
Above line graph represent most/least transaction in organization, where we can see that lowest stock price and lease amount of transaction both happened in same month, and most transaction noticed in May 2023.

- A relationship between Volume and close.



```
In [240]: findCorr('Arbor',ABR)
```

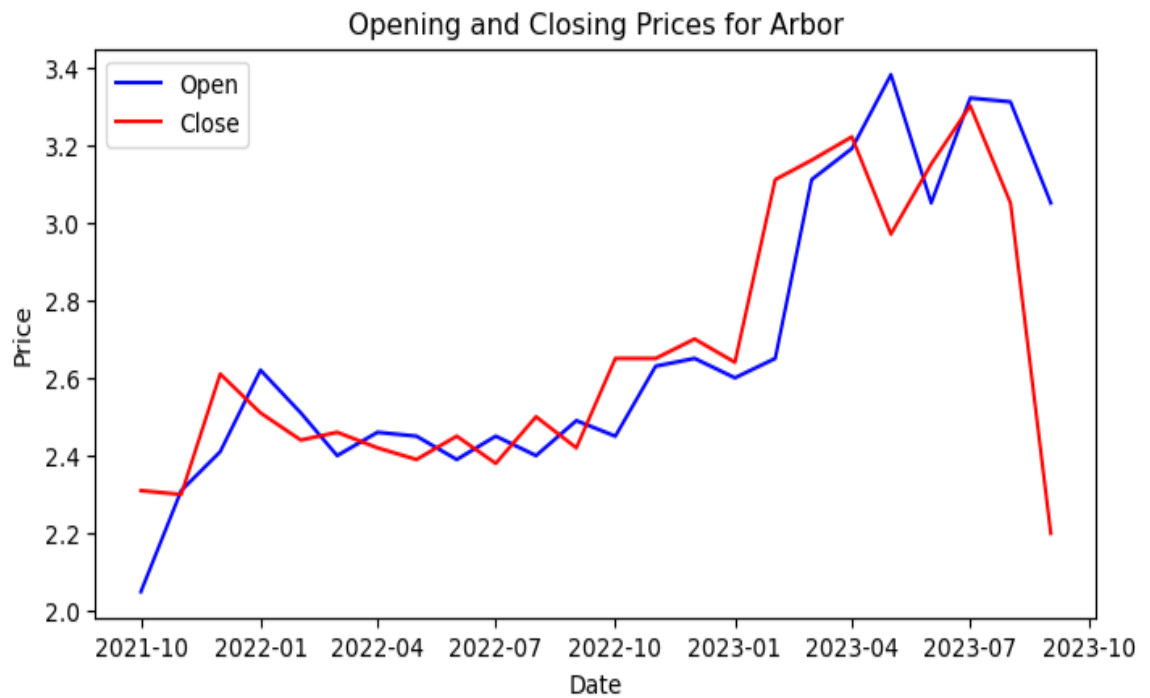
The correlation coefficient between volume and close is 0.33.



A slender positive correlation of 0.33 may be seen in the connection between volume and closing prices. It implies that, there is a some short of good tendency for larger trading volumes to be correlated with somewhat higher closing prices, but the connection is weak, indicating that other variables may also affect price changes. Hence, we can see that volume is more co relate with open as compare to close price.

- A relationship between Opening and Closing:

```
In [242]: openClose('Arbor',ABR)
```



The correlation coefficient between opening and closing prices is 0.76.

The correlation between starting and closing prices is 0.89, which is quite favorable. It suggests that, there is a significant propensity to cause an asset's closing and opening prices to fluctuate identically. It appears to be a strong prognostic correlation among these features in the collection whenever the initial cost is substantial and while the ending pricing is high.

## Solid Power, Inc. (SLDP)

- Explanation of data extraction process:

```
In [214]: #Input SPI Data set from yahoo_fin.stock_info

SLDP = get_data("SLDP",start_date = "20/09/2021",end_date="20/09/2023",index_as_date = True, interval="1mo")

C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2023' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2021' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Fetches the API for the Solid Power, Inc. Company using the ticker “SLDP” for two years starting from Sep -22-2021 to Sep-22-2023 with the time interval of 1 month.

- Validation Steps.

```
In [215]: SLDP.head()
```

```
Out[215]:
```

	open	high	low	close	adjclose	volume	ticker
2021-10-01	10.11	12.40	9.980	12.21	12.21	29918400	SLDP
2021-11-01	13.07	14.48	10.950	11.89	11.89	64703900	SLDP
2021-12-01	12.09	14.85	8.640	8.74	8.74	69308400	SLDP
2022-01-01	8.93	9.20	5.610	6.67	6.67	47721500	SLDP
2022-02-01	6.75	8.57	5.927	7.95	7.95	24315500	SLDP

Display the data that I have fetch using the API.

Time Period:
Sep 20, 2021 - Sep 20, 2023

Show:
Historical Prices

Frequency:
Monthly

Apply

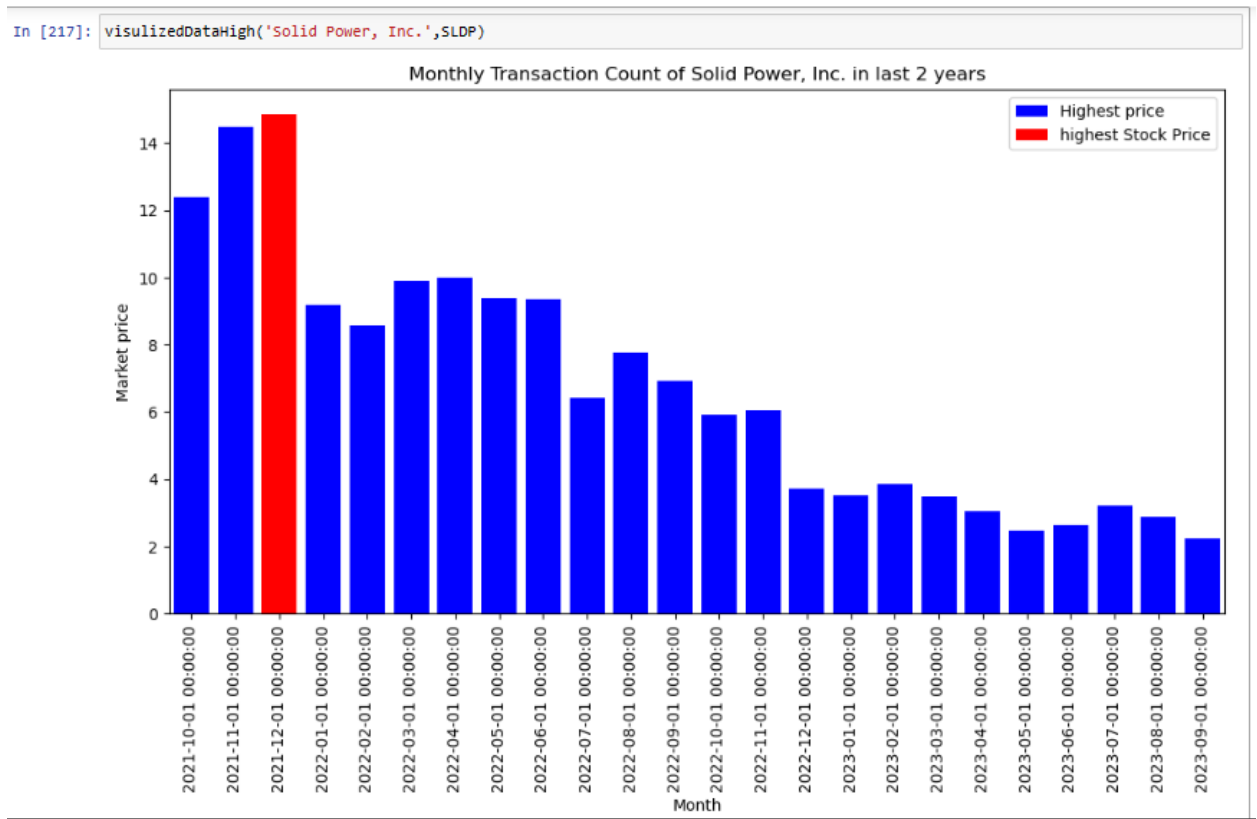
Currency in USD

Download

Date	Open	High	Low	Close*	Adj Close**	Volume
Sep 01, 2023	1.0800	1.1000	0.8100	0.9050	0.9050	1,400,900
Aug 01, 2023	1.3400	1.3500	1.0000	1.0900	1.0900	1,752,800
Jul 01, 2023	1.4000	1.5700	1.2400	1.3500	1.3500	1,783,000
Jun 01, 2023	1.2100	1.6100	1.1600	1.4000	1.4000	1,507,700
May 01, 2023	1.3500	1.3900	1.1200	1.1900	1.1900	1,316,300
Apr 01, 2023	1.2500	1.5000	1.0600	1.3100	1.3100	1,537,400
Mar 01, 2023	1.2900	1.3790	1.0000	1.1990	1.1990	1,354,600
Feb 01, 2023	1.7100	1.7800	1.1900	1.3100	1.3100	1,684,000
Jan 01, 2023	0.8800	1.8600	0.8600	1.6600	1.6600	4,151,900
Dec 01, 2022	1.4500	1.5100	0.7300	0.8600	0.8600	3,020,100

The above data was taken from the yahoo finance website, and by comparing both of the images, I can say that the data I fetched using the ticker “Solid Power, Inc.” is accurate.

- In which year the company has the highest / lowest prices:



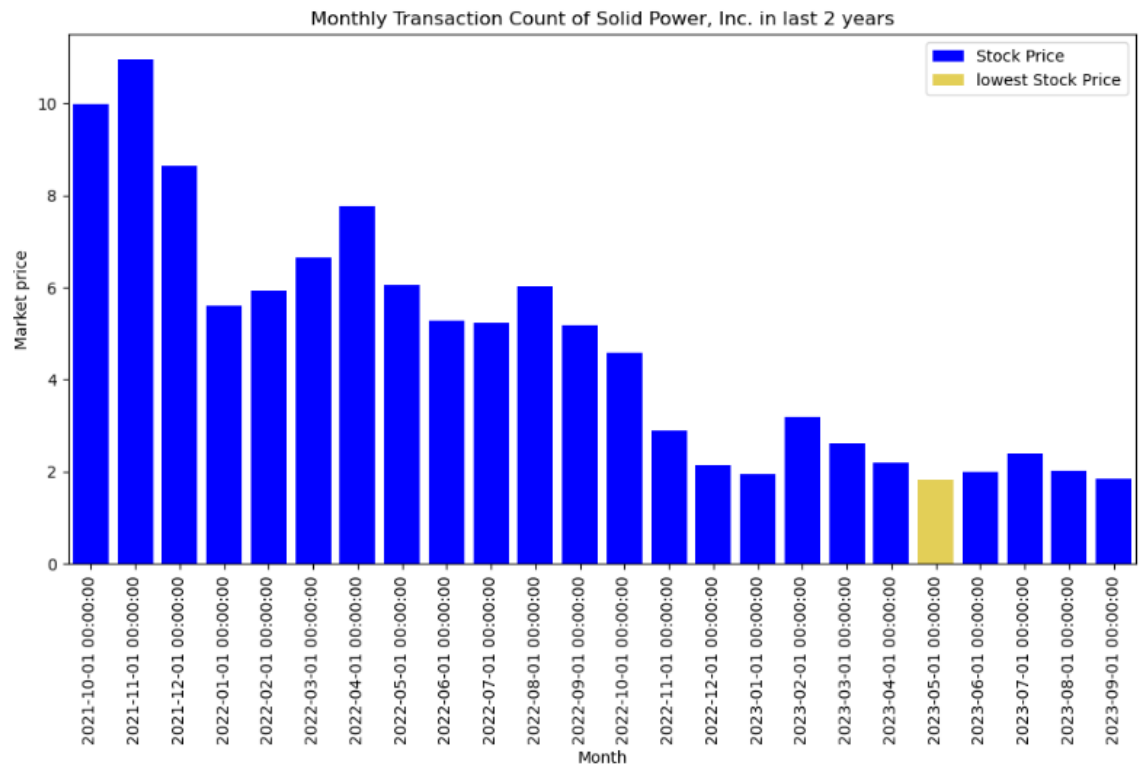
To display the highest price of the company, a bar graph was plotted using the Matplotlib program. Here, you can see that the stock price has been rising initially, while falling in following months. A stock price of the corporation is, nevertheless, gradually drop after 2022, and about to break their own low in near time.

- Lowest price:

To below display the low price of the Solid Power, INC graph, a bar graph was plotted using the Matplotlib program along with that we used sorted function and print tail of data which clarify that our graph is perfectly true .The graph fluctuates, as you can see, In starting periods price is decreasing continuously and after reached lowest on month of May 2023 it's started rising again.

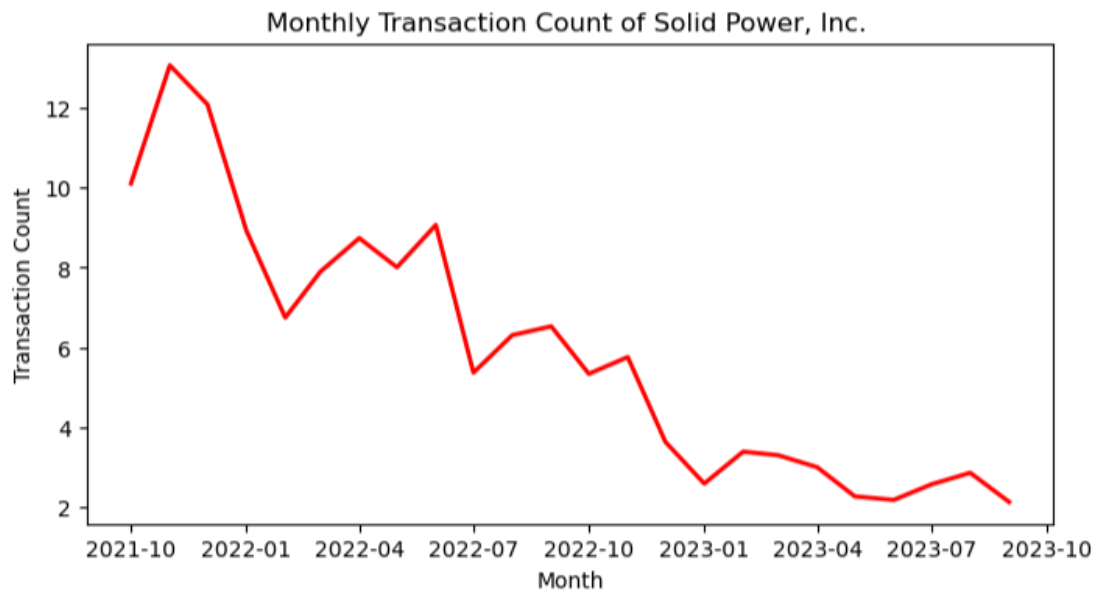
This leads me to the conclusion that the company's price is highest in November 2021 and lowest in May 2023.

```
In [218]: visualizedDataLow('Solid Power, Inc.',SLDP)
```



- What time of the year do your most/least transactions?

```
In [244]: transactionsHistory('Solid Power, Inc.',SLDP)
```

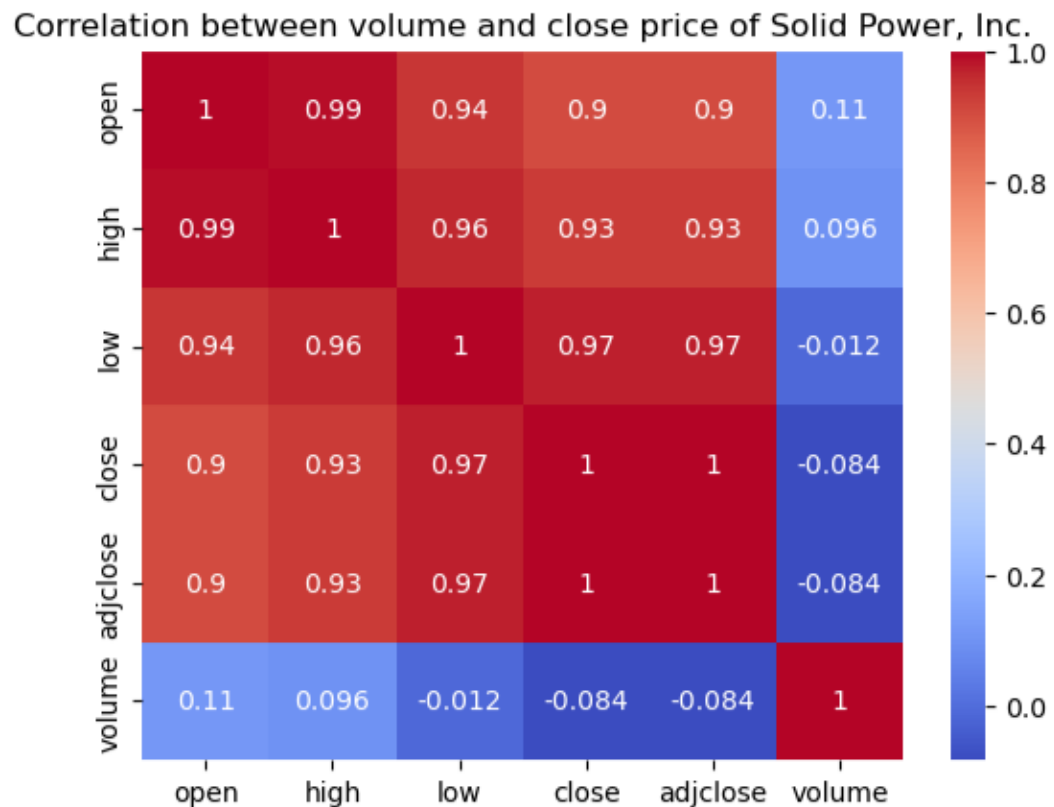


Above line graph represent most/least transaction in organization, where we can see that lowest stock price and lease amount of transaction both happened in same month, and most transaction noted at November, 2021.

- A relationship between Volume and close.

```
In [243]: findCorr('Solid Power, Inc.',SLDP)
```

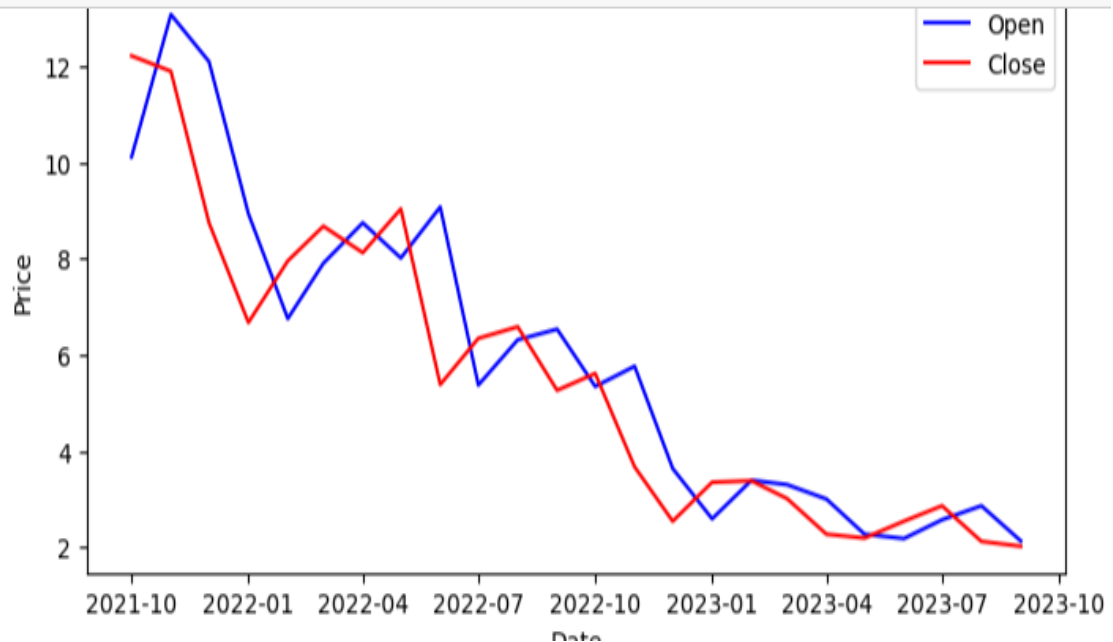
The correlation coefficient between volume and close is -0.08.



Here we can see that co relation between volume and close price is -0.026 which is too low. Hence, we can say that volume and close price are not depends on each others.

- A relationship between Opening and Closing:

```
In [245]: openClose('Solid Power, Inc.',SLDP)
```



The correlation coefficient between opening and closing prices is 0.90.

The correlation between starting and closing prices is 0.90, which is quite favorable. It suggests that, there is a significant propensity to cause an asset's closing and opening prices to fluctuate identically. It appears to be a strong prognostic correlation among these features in the collection whenever the initial cost is substantial and while the ending pricing is high.

## Boyd Gaming Corporation (BYD):

- Explanation of data extraction process.

```
In [246]: #Input Boyd Gaming Corporation Data set from yahoo_fin.stock_info

BYD = get_data("BYD", start_date = "20/09/2021", end_date="20/09/2023", index_as_date = True, interval="1mo")

C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2023' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2021' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Fetches the API for the Boyd Gaming Corporation. Company using the ticker “BYD” for two years starting from Sep -22-2021 to Sep-22-2023 with the time interval of 1 month.

- Validation Steps.

```
In [272]: BYD.head(4)
```

Out[272]:

	open	high	low	close	adjclose	volume	ticker
2021-10-01	64.099998	69.230003	63.200001	63.779999	62.610310	23070100	BYD
2021-11-01	64.120003	68.279999	57.860001	58.610001	57.535130	25045000	BYD
2021-12-01	59.700001	66.059998	56.450001	65.570000	64.367485	23251800	BYD
2022-01-01	65.959999	66.989998	54.599998	59.459999	58.369537	19970600	BYD

Display the data that I have fetched using the API.

[Finance Home](#)
[News](#)
[Taxes](#)
[Watchlists](#)
[My Portfolio](#)
[Real Estate](#)
[My Screeners](#)
[Tech](#)
[Market Data](#)
[Industry News](#)

Currency in USD

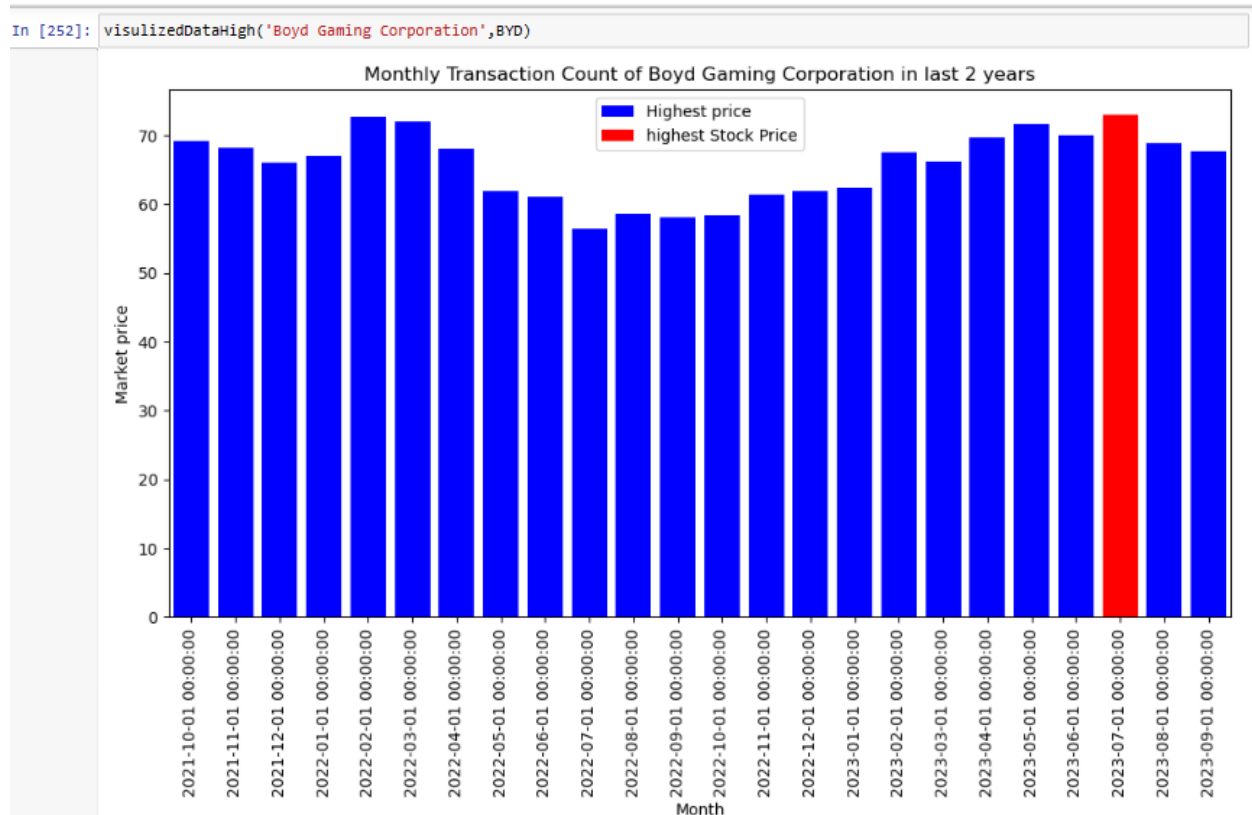
Download

Date	Open	High	Low	Close*	Adj Close**	Volume
Sep 14, 2023	0.16 Dividend					
Sep 01, 2023	67.25	67.75	61.66	61.68	61.52	14,016,200
Aug 01, 2023	68.05	68.80	63.97	66.87	66.70	14,937,700
Jul 01, 2023	69.47	73.00	65.53	68.32	68.15	15,806,100
Jun 14, 2023	0.16 Dividend					
Jun 01, 2023	63.64	70.04	63.30	69.37	69.04	16,477,000
May 01, 2023	69.95	71.69	63.22	63.73	63.42	20,442,200
Apr 01, 2023	64.22	69.78	61.85	69.40	69.07	17,552,600
Mar 14, 2023	0.16 Dividend					
Mar 01, 2023	64.95	66.17	57.74	64.12	63.64	20,859,700
Feb 01, 2023	62.22	67.50	61.02	65.13	64.64	20,717,100
Jan 01, 2023	55.28	62.38	54.04	62.31	61.84	13,733,400



The above data was taken from the yahoo finance website, and by comparing both of the images, I can say that the data I fetched using the ticker “Boyd Gaming Corporation.” is accurate.

- In which year the company has the highest / lowest prices.



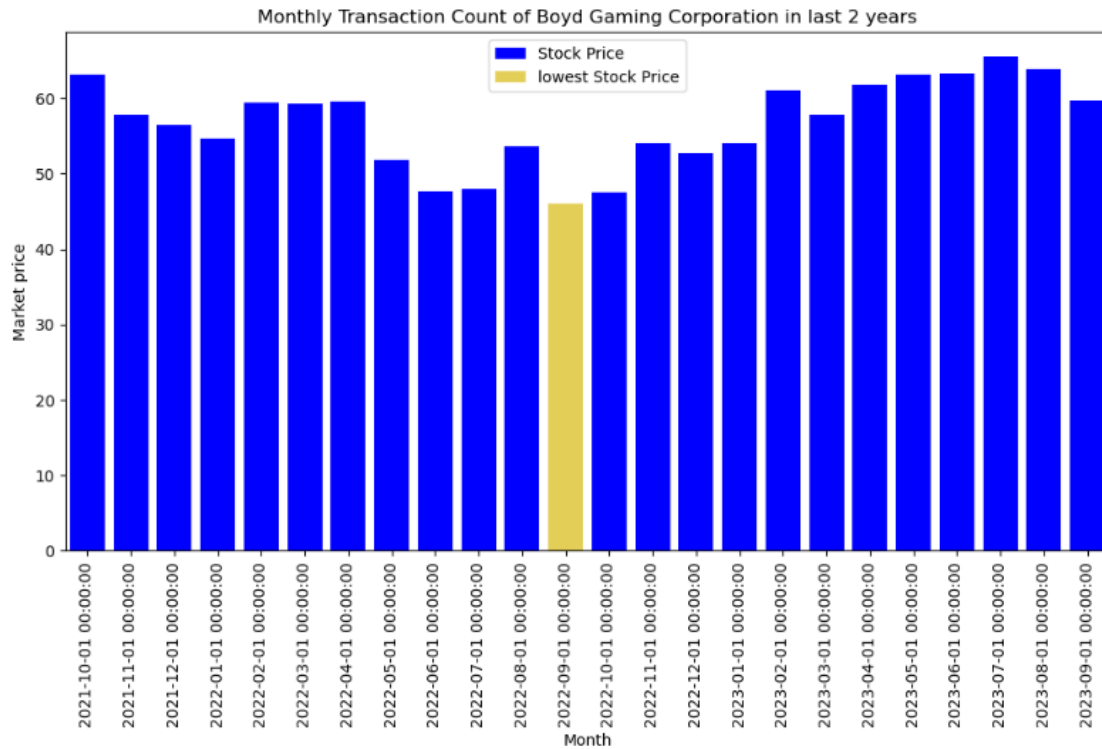
To display the highest price of the company, a bar graph was plotted using the Matplotlib program. Here, you can see that the stock price has been rising initially, while steady in following months. A stock price of the corporation is, nevertheless, gradually rise after 2023 and noted the highest price on July 2023, and about to break their own high in near time.

- Lowest price:

To below display the low price of the boyd Gaming Corporation graph, a bar graph was plotted using the Matplotlib program along with that we used sorted function and print tail of data which clarify that our graph is perfectly true .The graph fluctuates, as you can see, In starting periods price is decreasing continuously and after reached lowest on month of October 2022 it's started rising again.

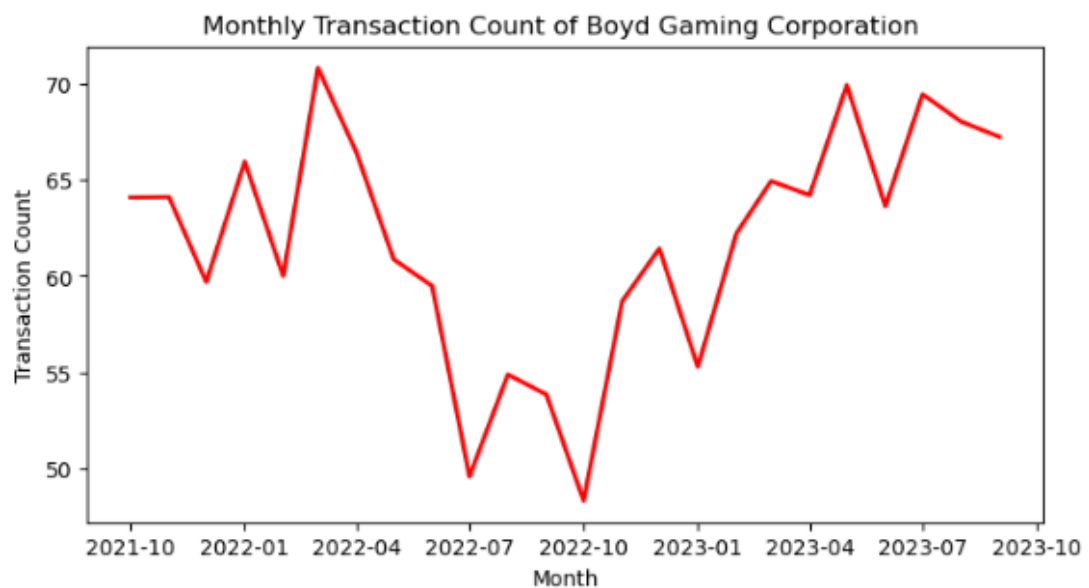
This leads me to the conclusion that the company's price is highest in July 2021 and lowest in October 2023.

```
In [253]: visulizedDataLow('Boyd Gaming Corporation',BYD)
```



- What time of the year do your most/least transactions?

```
In [255]: transactionsHistory('Boyd Gaming Corporation',BYD)
```



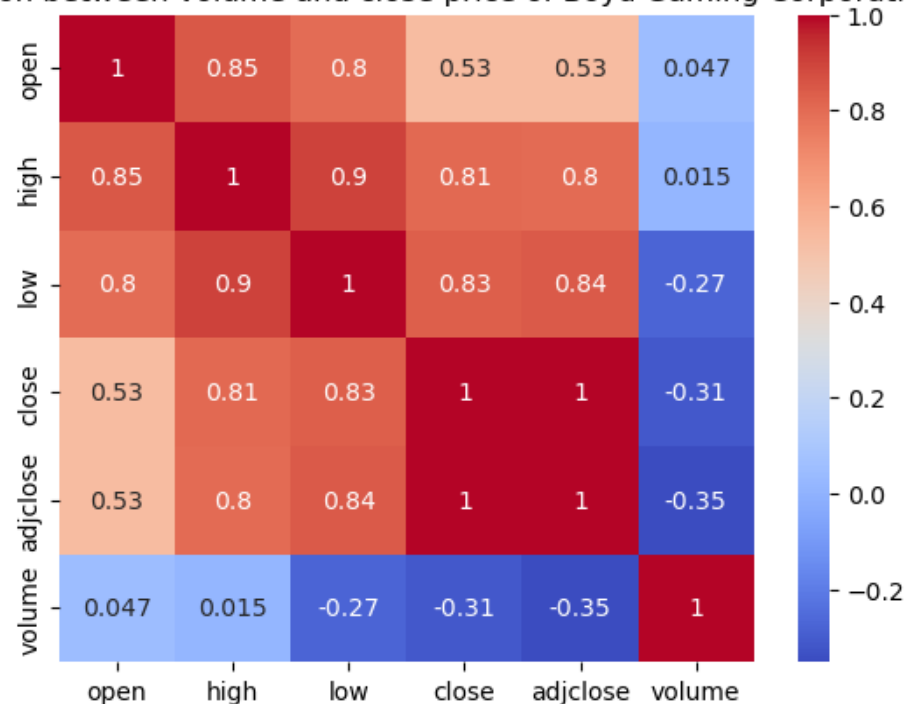
Above dash line graph represent most/least transaction in organization, where we can see that lowest stock price and lease amount of transaction both happened in same month, and least transaction noted at October, 2022.

- A relationship between Volume and close.

```
In [254]: findCorr('Boyd Gaming Corporation',BYD)
```

The correlation coefficient between volume and close is -0.31.

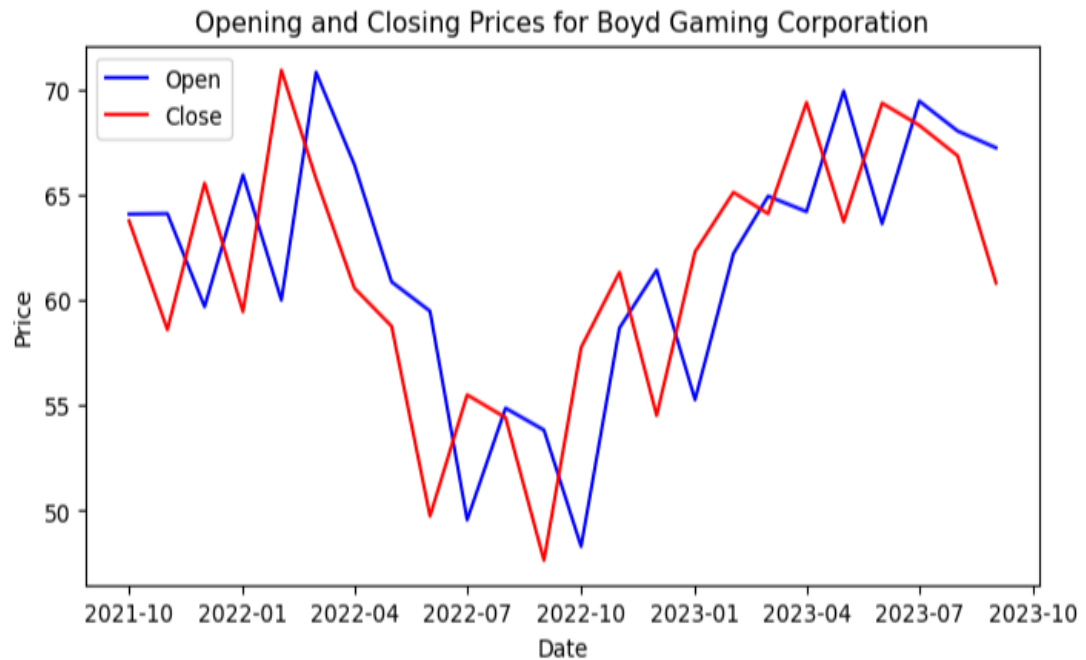
Correlation between volume and close price of Boyd Gaming Corporation



Here we can see that co relation between volume and close price is -0.31. This is too low. Hence, we can say that volume and close price are not depends on each others.

- A relationship between Opening and Closing.

```
In [256]: openClose('Boyd Gaming Corporation',BYD)
```



The correlation coefficient between opening and closing prices is 0.53.

The correlation between starting and closing prices is 0.54, which is quite favorable. It suggests that, there is a significant propensity to cause an asset's closing and opening prices to fluctuate identically. It appears to be a strong prognostic correlation among these features in the collection whenever the initial cost is substantial and while the ending pricing is high.

## Microvast Holdings, Inc. (MVST)

- Explanation of data extraction process.

```
In [257]: #Input Microvast Holdings, Inc Data set from yahoo fin.stock_info

MVST = get_data("MVST",start_date = "20/09/2021",end_date="20/09/2023",index_as_date = True, interval="1mo")

C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2023' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
C:\Users\win\AppData\Roaming\Python\Python39\site-packages\IPython\core\interactiveshell.py:3460: UserWarning: Parsing '20/09/2021' in DD/MM/YYYY format. Provide format or specify infer_datetime_format=True for consistent parsing.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Fetches the API for the Microvast Holdings, Inc Data set from yahoo fin.stock\_info. Company using the ticker “MVST” for two years starting from Sep -22-2021 to Sep-22-2023 with the time interval of 1 month.

- Validation Steps.

```
In [273]: MVST.head()
```

Out[273]:

	open	high	low	close	adjclose	volume	ticker
2021-10-01	8.270	8.620	7.380	8.46	8.46	33419700	MVST
2021-11-01	9.226	10.250	8.000	8.49	8.49	30442000	MVST
2021-12-01	8.550	8.850	5.505	5.66	5.66	29581800	MVST
2022-01-01	5.740	6.290	4.770	5.54	5.54	19122200	MVST
2022-02-01	5.600	7.425	5.430	7.14	7.14	14822300	MVST

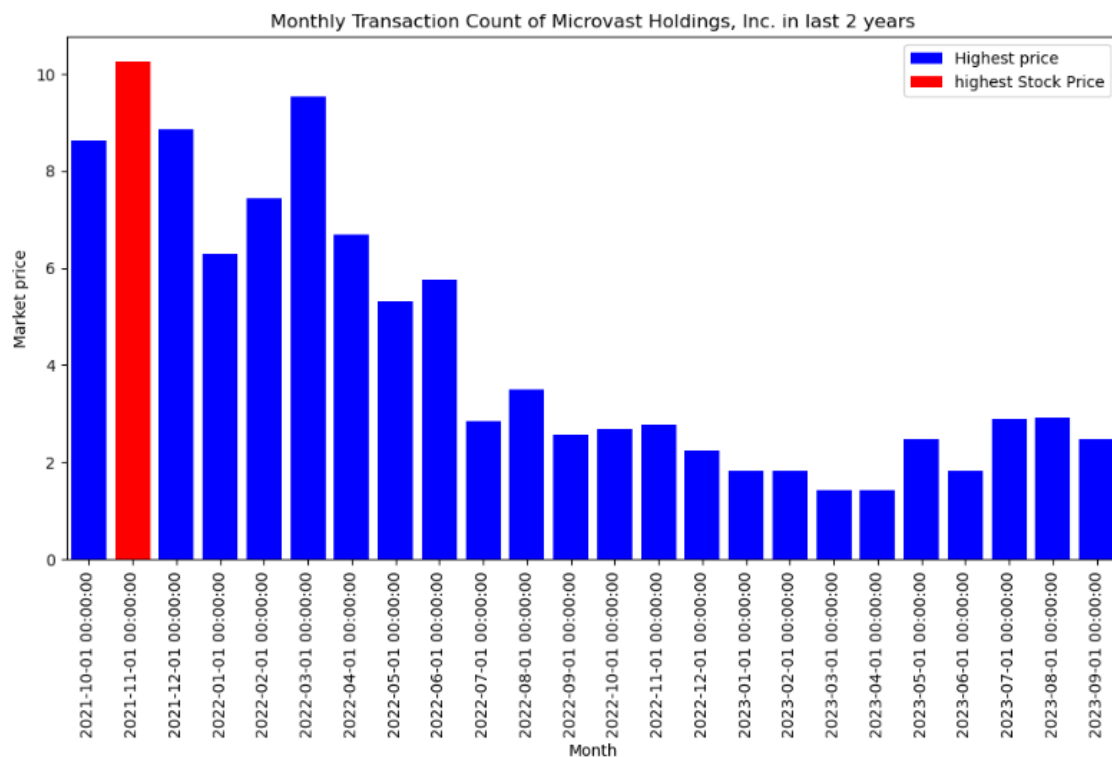
Display the data that I have fetched using the API.

The below data was taken from the yahoo finance website, and by comparing both of the images, I can say that the data I fetched using the ticker “Microvast Holdings, Inc.” is accurate.

Date	Open	High	Low	Close*	Adj Close**	Volume
Sep 01, 2023	2.2700	2.4650	1.7600	1.7900	1.7900	43,301,500
Aug 01, 2023	2.8700	2.9100	1.5330	2.2300	2.2300	97,011,000
Jul 01, 2023	1.6300	2.9000	1.6250	2.8700	2.8700	113,894,500
Jun 01, 2023	1.2400	1.8290	1.2100	1.6000	1.6000	70,257,300
May 01, 2023	1.0700	2.4600	0.9400	1.2400	1.2400	161,451,100
Apr 01, 2023	1.3100	1.4300	0.8800	1.0500	1.0500	29,920,600
Mar 01, 2023	1.2900	1.4300	1.0400	1.2400	1.2400	38,543,000
Feb 01, 2023	1.6500	1.8300	1.2800	1.2900	1.2900	31,263,200
Jan 01, 2023	1.5900	1.8100	1.4100	1.6300	1.6300	31,767,200
Dec 01, 2022	2.1800	2.2400	1.3800	1.5300	1.5300	27,440,100
Nov 01, 2022	2.5000	2.7700	1.9700	2.1900	2.1900	42,971,800
Oct 01, 2022	1.8100	2.6820	1.4900	2.4500	2.4500	128,659,500

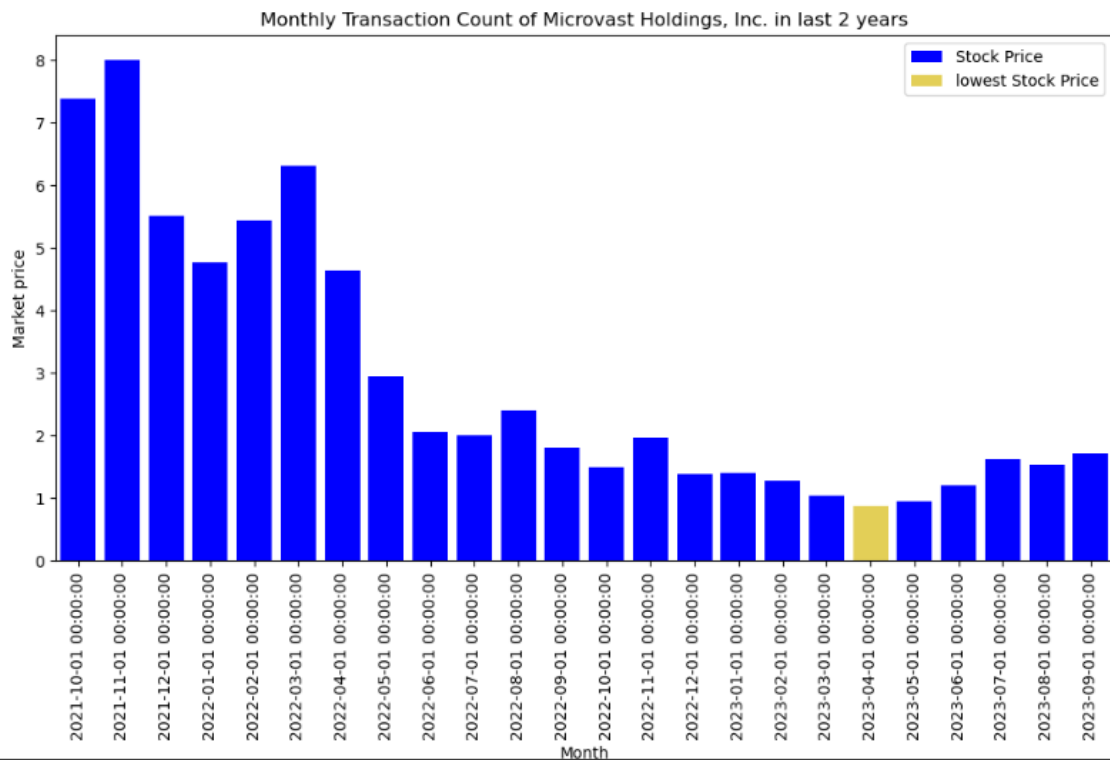
- In which year the company has the highest / lowest prices:

In [265]: `visualizedDataHigh('Microvast Holdings, Inc.',MVST)`



To display the highest price of the company, a bar graph was plotted using the Matplotlib program. Here, you can see that the stock price has been rising initially, while steady in following months. A stock price of the corporation is, nevertheless, gradually drop after July 2022 and noted the highest price on November 2021, and about to break their own high in near time.

```
In [266]: visualizedDataLow('Microvast Holdings, Inc.',MVST)
```

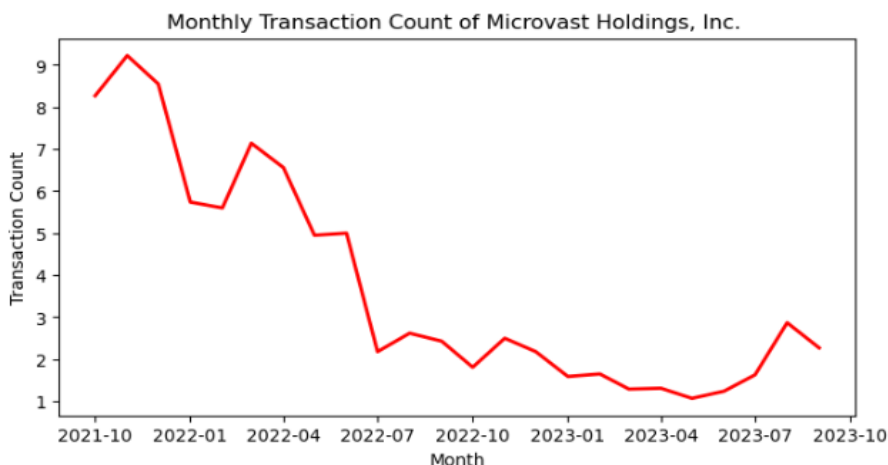


To below display the low price of the Microvast Holding INC graph, a bar graph was plotted using the Matplotlib program along with that we used sorted function and print tail of data which clarify that our graph is perfectly true .The graph fluctuates, as you can see, In starting periods price is decreasing continuously and after reached lowest on month of April 2023 it's started rising again.

To display the high and low price of the Industry, a line graphs were plotted using the Matplotlib program. Where we can show that highest price is 10.25 which one recorded on 2021-11-01. And lowest price is 0.8799 which one noted on 2022-06-01.

#### • What time of the year do your most/least transactions?

```
In [269]: transactionsHistory('Microvast Holdings, Inc.',MVST)
```



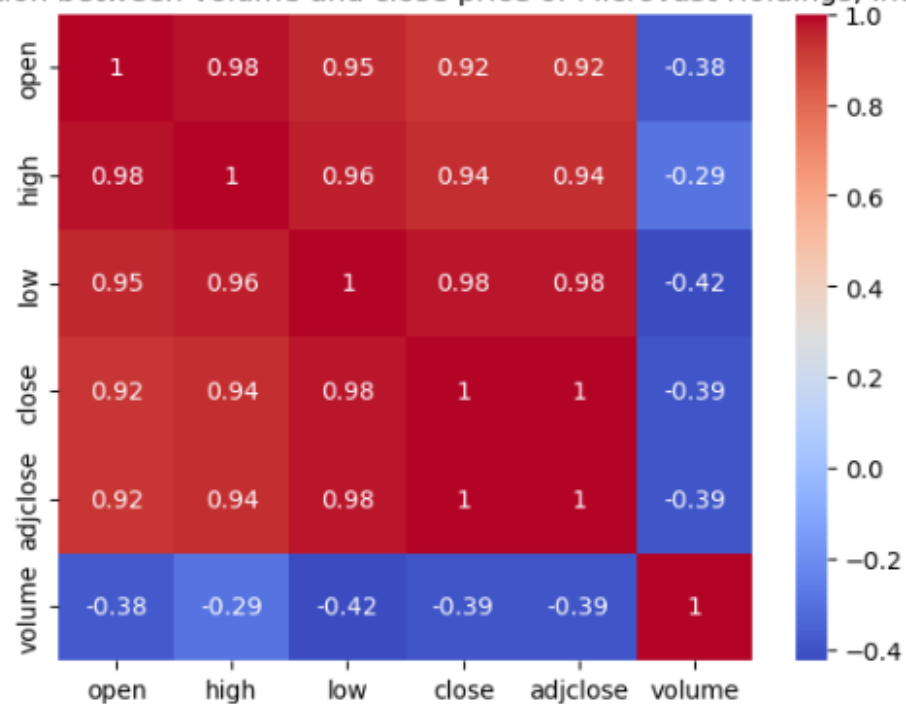
Above dash line graph represent most/least transaction in organization, where we can see that lowest stock price and lease amount of transaction both happened in same month, and least transaction noted at May, 2023.

- A relationship between Volume and close:

```
In [268]: findCorr('Microvast Holdings, Inc.',MVST)
```

The correlation coefficient between volume and close is -0.39.

Correlation between volume and close price of Microvast Holdings, Inc.



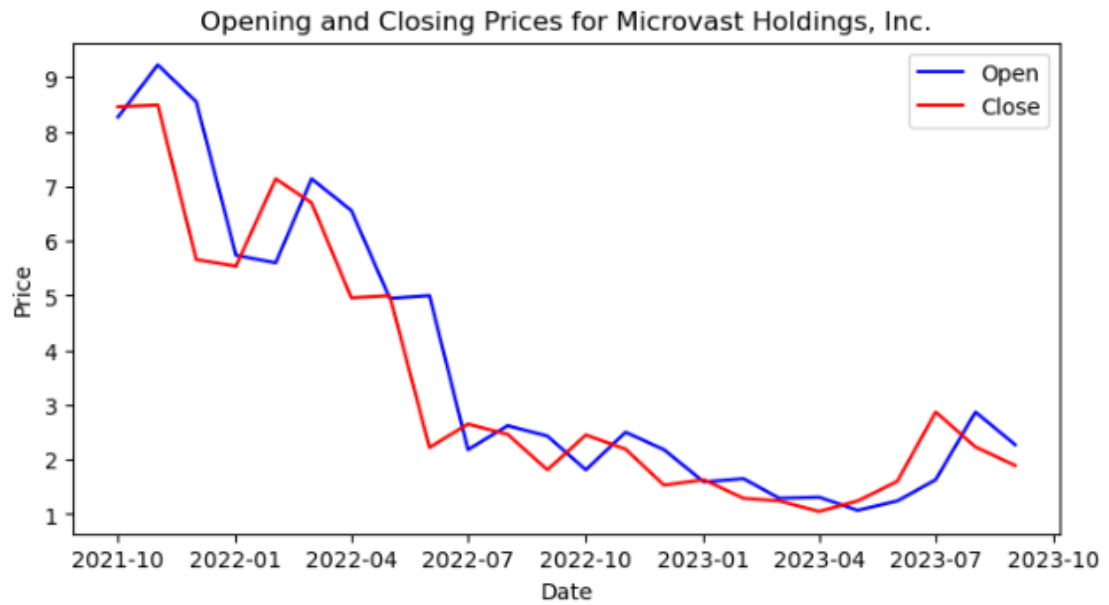
Here we can see that correlation between volume and close price is -0.39. This is too low. Hence, we can say that volume and close price are not dependent on each other.



- A relationship between Opening and Closing

The correlation between starting and closing prices is 0.92. This is quite favorable. It suggests that, there is a significant propensity to cause an asset's closing and opening prices to fluctuate identically. It appears to be a strong prognostic correlation among these features in the collection whenever the initial cost is substantial and while the ending pricing is high.

```
In [270]: openClose('Microvast Holdings, Inc.',MVST)
```



The correlation coefficient between opening and closing prices is 0.92.

## Conclusion:

In conclusion, Yahoo Finance is a rich resource of financial market data and tools to find compelling investments. By downloading historical data from Yahoo Finance, one can view historical price, dividend, and split data for most quotes to forecast the future of a company or gain market insight. Using Python with time-series data, one can perform data analysis on stock price data for technical analysis. The Yahoo Finance API provides access to information about finance summaries, stocks historical prices, and stock actions. By analyzing the data, one can get insights into the highest and lowest stock prices, correlation between volume and close price, correlation between open and close price, and transaction history.

.