

Look at the raw data

In [3]: analyzer.data.head()

Out[3]:

	NVDA_Close_NVDA	NVDA_High_NVDA	NVDA_Low_NVDA	NVDA_Open_NVDA	NVDA_Volume_NVDA	^GSPC_Close_^GSPC	^C
Date							
2022-02-14	24.226711	24.833702	23.715560	23.894264	440424000	4401.669922	44
2022-02-15	26.451010	26.500926	24.742850	24.907575	699869000	4471.069824	44
2022-02-16	26.466984	26.537867	25.509576	26.215403	732676000	4475.009766	44
2022-02-17	24.466309	25.742186	24.124877	25.587443	810595000	4380.259766	44
2022-02-18	23.602745	24.944513	23.061645	24.627041	761255000	4348.870117	43

Plots

Price

Varies quite a bit over the time range

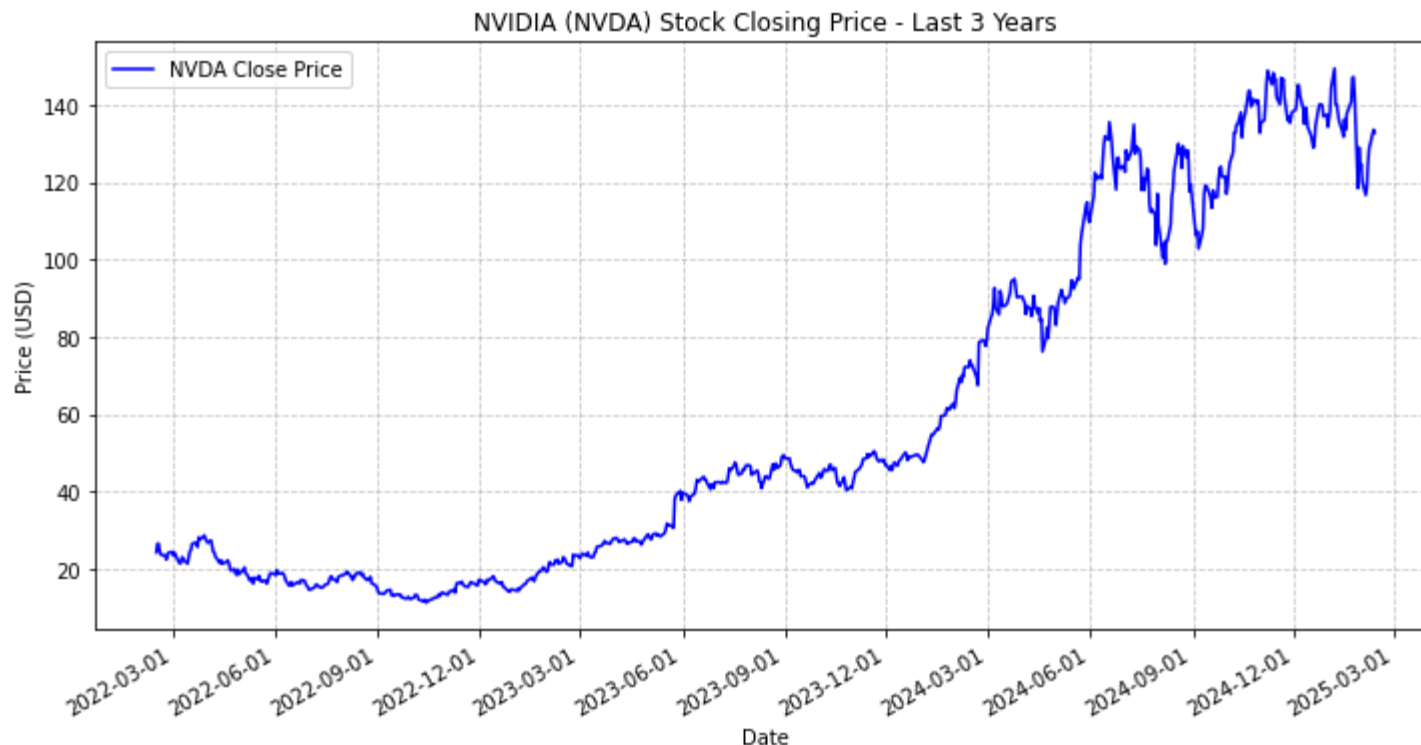
```
In [4]: fig, ax = analyzer.plot_close_price()  
fig
```

/home/kjp/anaconda3/lib/python3.7/site-packages/pandas/plotting/_matplotlib/converter.py:103: FutureWarning: Using an implicitly registered datetime converter for a matplotlib plotting method. The converter was registered by pandas on import. Future versions of pandas will require you to explicitly register matplotlib converters.

To register the converters:

```
>>> from pandas.plotting import register_matplotlib_converters  
>>> register_matplotlib_converters()  
warnings.warn(msg, FutureWarning)
```

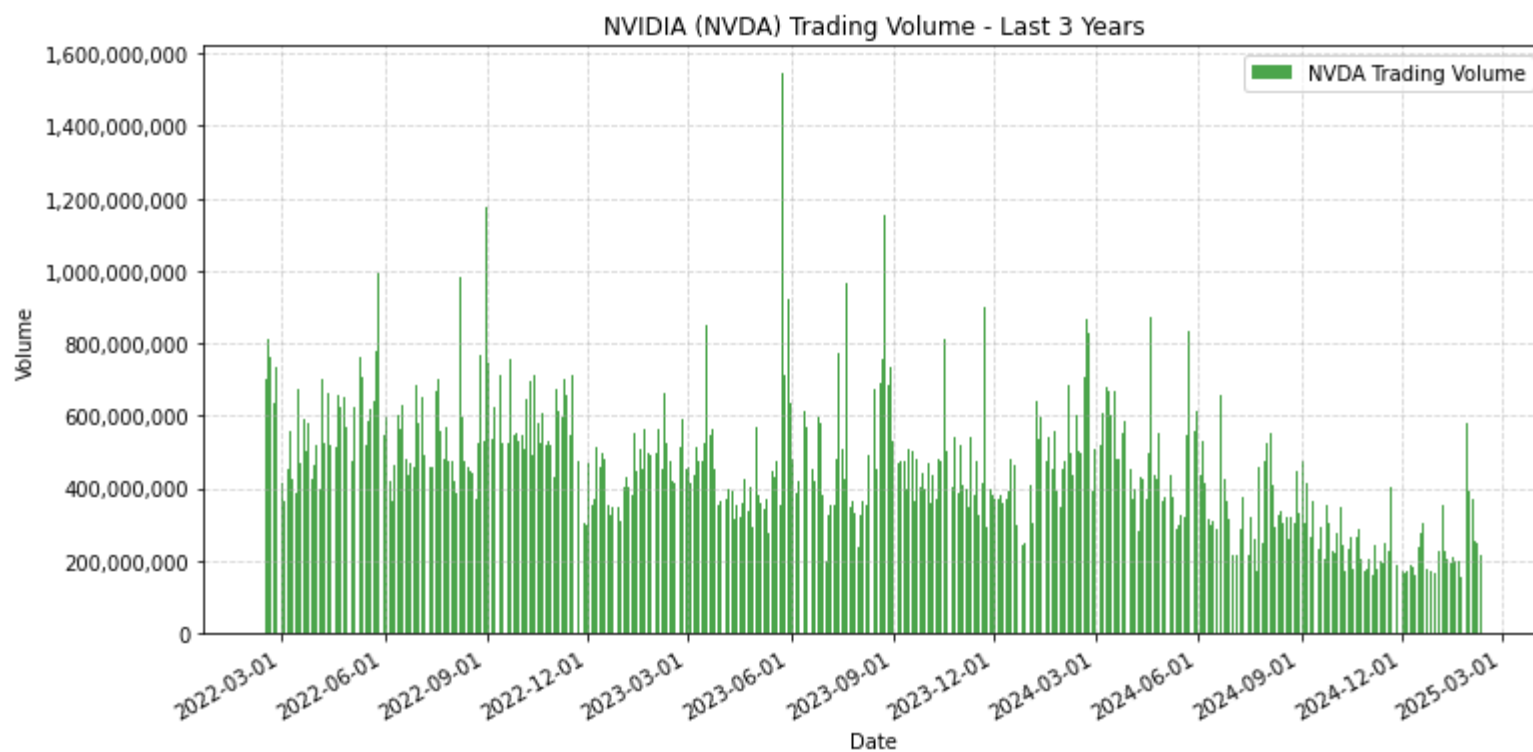
Out[4]:



Volume, in shares

```
In [5]: fig, ax = analyzer.plot_volume()  
fig
```

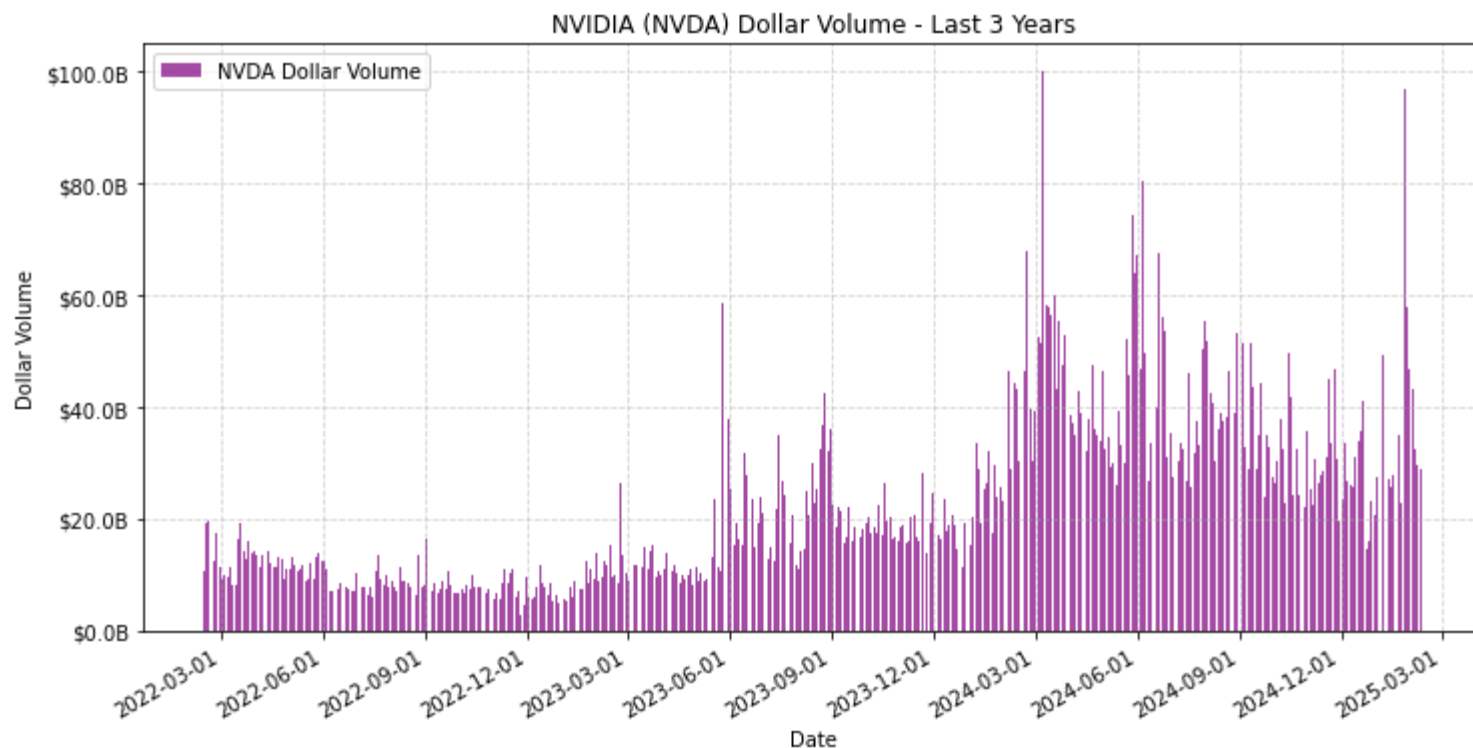
Out[5]:



Volume (in dollars)


```
In [6]: fig, ax = analyzer.plot_dollar_volume()  
fig
```

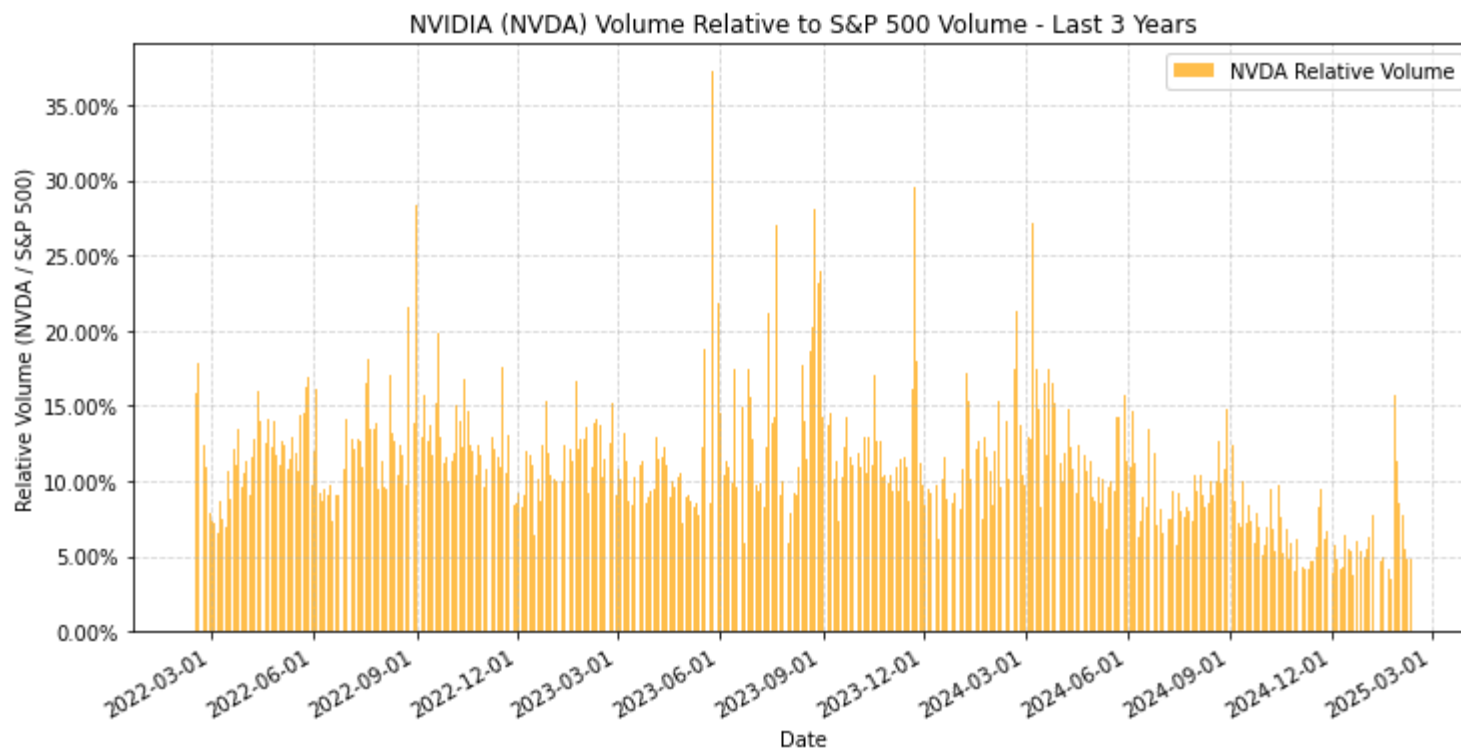
Out[6]:



Volume (relative to total market volume)

```
In [7]: fig, ax = analyzer.plot_relative_volume()  
fig
```

Out[7]:



Let's assume that the training data and test data are from the first/second half.

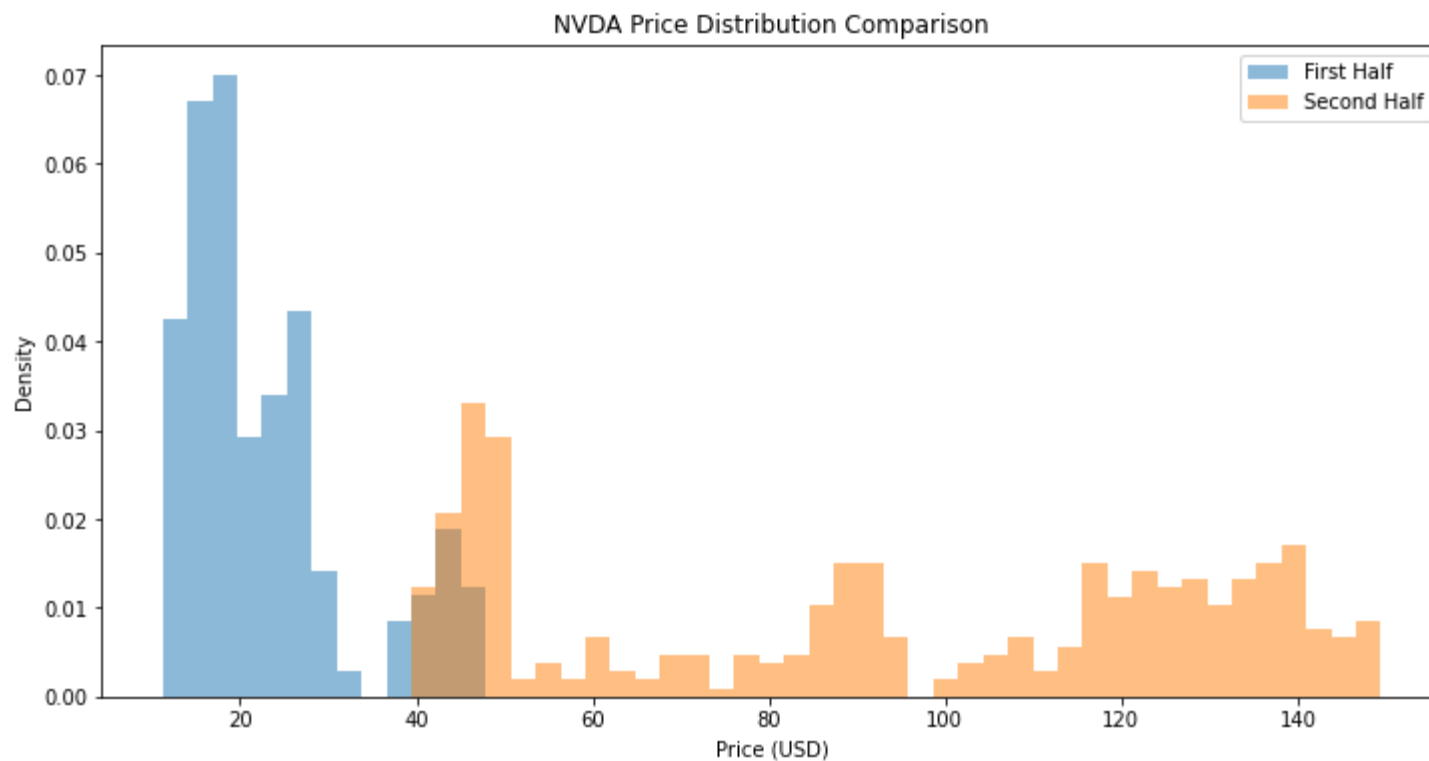
Are the distributions of training/test data similar ?

Note: we are assuming the second half is closer to the present, and therefore a more accurate proxy for unseen data that will be encountered in the near future.

Price distributions

```
In [8]: fig, ax = analyzer.plot_price_histogram()  
fig
```

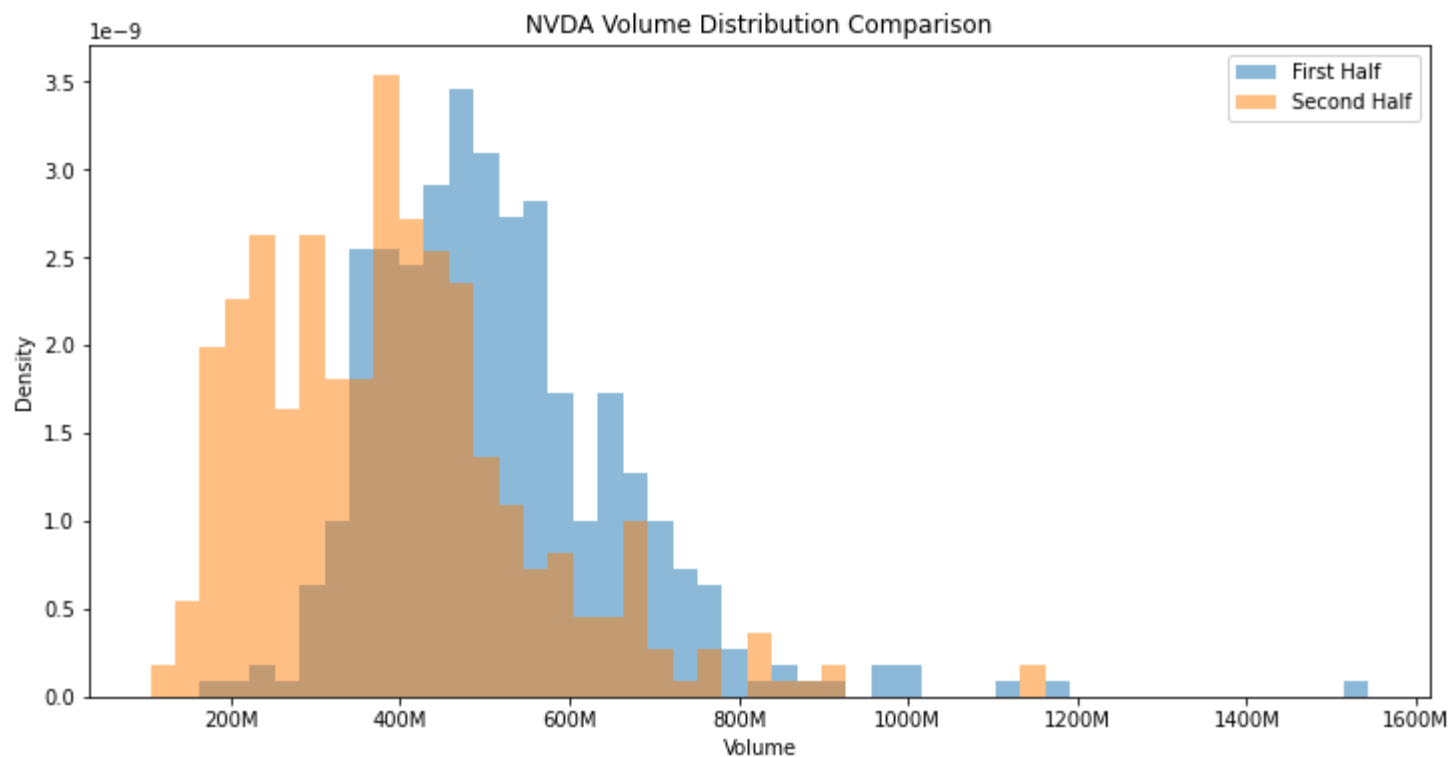
Out[8]:



Volume (shares) distributions

```
In [9]: fig, ax = analyzer.plot_volume_histogram()  
fig
```

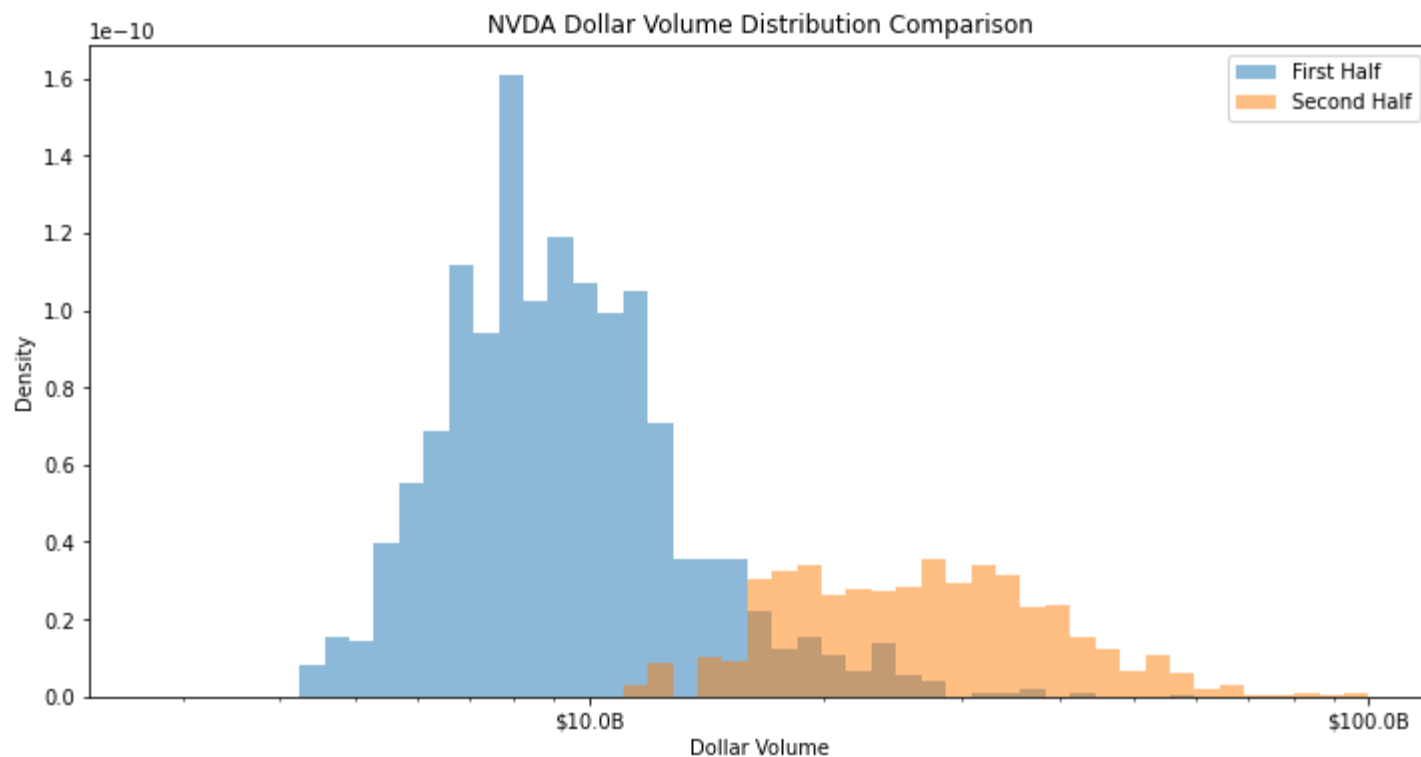
Out[9]:



Volume (dollars) distributions

```
In [10]: fig, ax = analyzer.plot_dollar_volume_histogram()  
fig
```

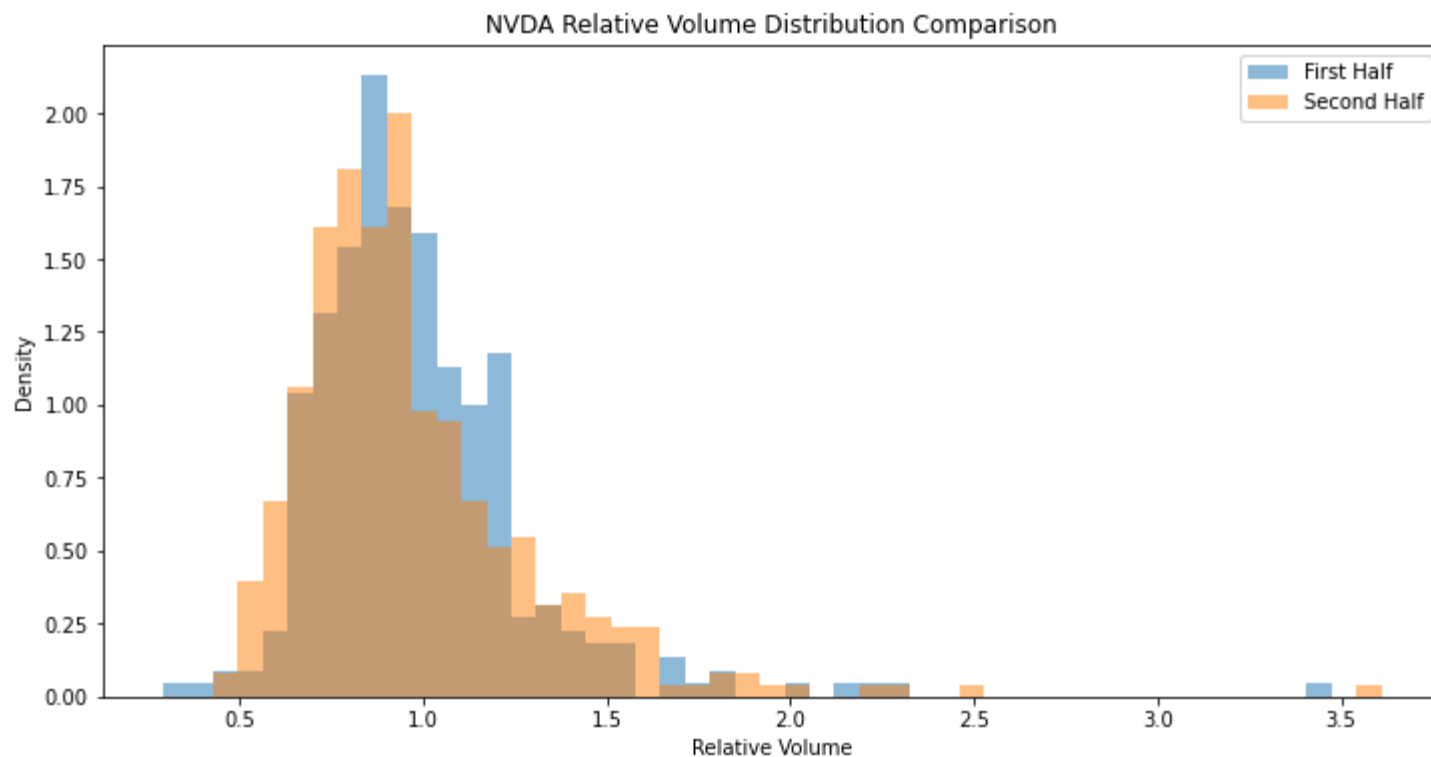
Out[10]:



Volume (relative to market) distributions

```
In [11]: fig, ax = analyzer.plot_relative_volume_histogram()  
fig
```

Out[11]:



```
In [12]: print("Done")
```

Done

