Plotting Candlestick Chart - OHLC

Takao Oba

Assisted by quantpy

In [1]:

```python
# import dependencies
import datetime as dt
import pandas as pd
from pandas_datareader import data as pdr
import plotly.offline as pyo
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [2]:

```python
# initiating jupyter notebook so that the graphs can be seen
pyo.init_notebook_mode(connected = True)
```

Getting the stock market data

In [5]:

```python
end = dt.datetime.now()
start = dt.datetime(2015,1,1)

import yfinance as yfin
yfin.pdr_override()
df = pdr.get_data_yahoo('AAPL', start, end)
df.head()
```

```
[*********************100%***********************]  1 of 1 completed
```

Out[5]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2015-01-02	27.847500	27.860001	26.837500	27.332500	24.603205	212818400
2015-01-05	27.072500	27.162500	26.352501	26.562500	23.910095	257142000
2015-01-06	26.635000	26.857500	26.157499	26.565001	23.912344	263188400
2015-01-07	26.799999	27.049999	26.674999	26.937500	24.247650	160423600
2015-01-08	27.307501	28.037500	27.174999	27.972500	25.179304	237458000

Generate moving average items

In [8]:

```
# moving average 50
df['MA50'] = df['Close'].rolling(window = 10).mean()
df['MA50'].head(20)
# the first 10 values would be NaN because of the windows
```

Out[8]:

```
Date
2015-01-02         NaN
2015-01-05         NaN
2015-01-06         NaN
2015-01-07         NaN
2015-01-08         NaN
2015-01-09         NaN
2015-01-12         NaN
2015-01-13         NaN
2015-01-14         NaN
2015-01-15    27.23950
2015-01-16    27.15600
2015-01-20    27.21775
2015-01-21    27.30000
2015-01-22    27.41625
2015-01-23    27.44350
2015-01-26    27.47075
2015-01-27    27.46800
2015-01-28    27.59525
2015-01-29    27.82275
2015-01-30    28.08125
Name: MA50, dtype: float64
```

In [10]:

```
# to not introduce NaN, we will use minimum periods parameter
df['MA50'] = df['Close'].rolling(window = 50, min_periods = 0).mean()
df['MA200'] = df['Close'].rolling(window = 200, min_periods = 0).mean()
print(df['MA50'].head())
print(df['MA200'].head())
```

```
Date
2015-01-02    27.332500
2015-01-05    26.947500
2015-01-06    26.820000
2015-01-07    26.849375
2015-01-08    27.074000
Name: MA50, dtype: float64
Date
2015-01-02    27.332500
2015-01-05    26.947500
2015-01-06    26.820000
2015-01-07    26.849375
```

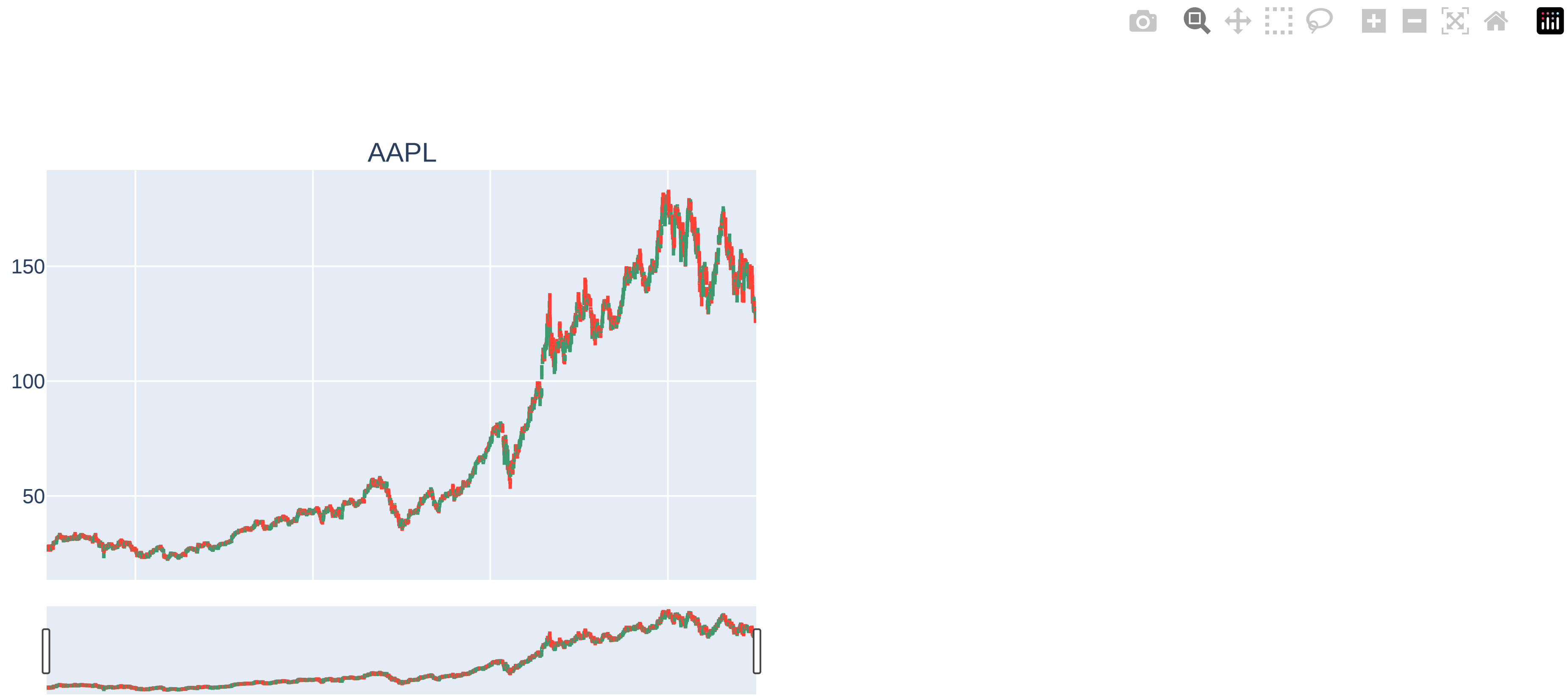
```
2015-01-08    27.074000
Name: MA200, dtype: float64
```

Generating a subplots

```
In [30]: fig = make_subplots(rows = 2, cols = 1, shared_xaxes = True,
                             vertical_spacing = 0.1, subplot_titles = ('AAPL', 'Volume'),
                             row_width = [0.2,0.7])
```

Look into adding candle stick graph (OHLC)

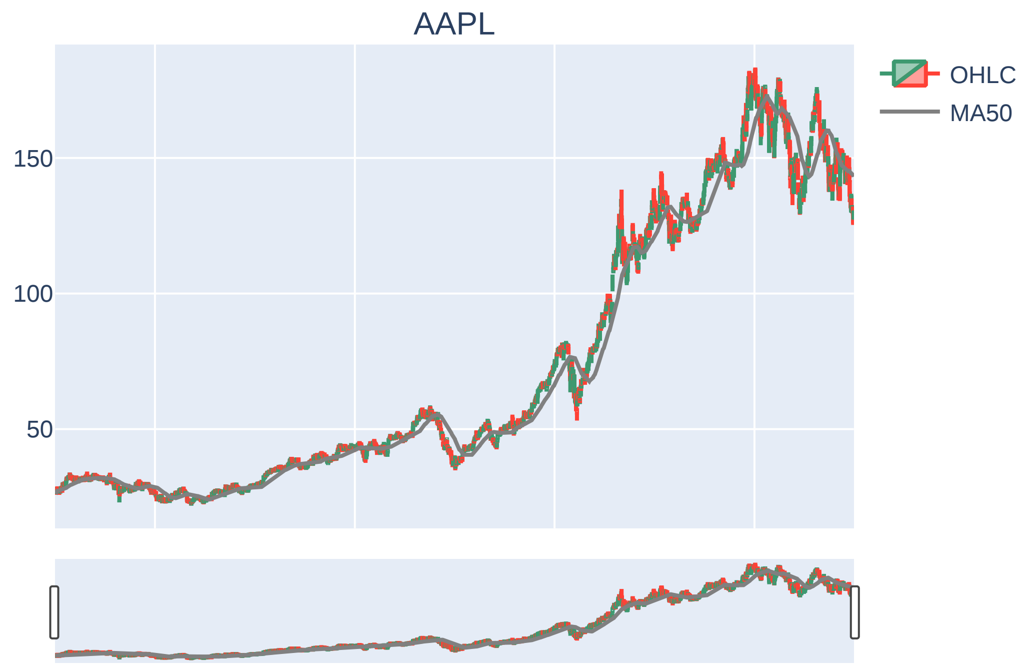
```
In [31]: fig.add_trace(go.Candlestick(x = df.index, open = df['Open'], high = df['High'], low = df['Low'], close = df['Close'], name = "C
                       row = 1, col = 1
         )

         # We can see that the plot below shows two plots: one for AAPL and one for volume
```



Adding moving average items

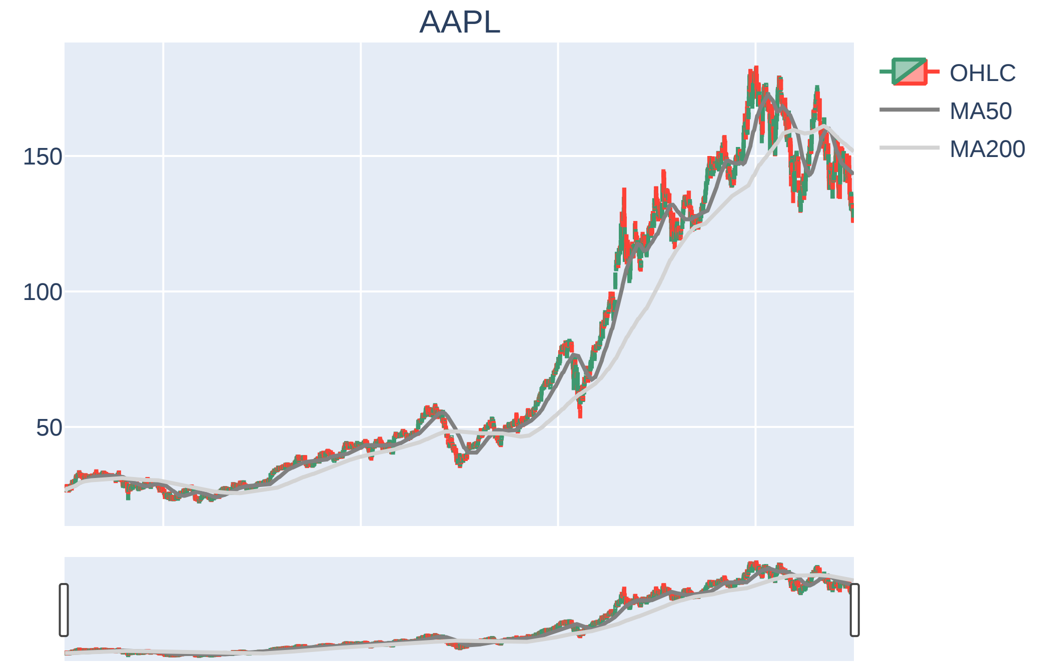
In [32]:

```
fig.add_trace(go.Scatter(x = df.index, y = df['MA50'], marker_color = 'grey', name = 'MA50'), row = 1, col = 1)
```



In [33]:

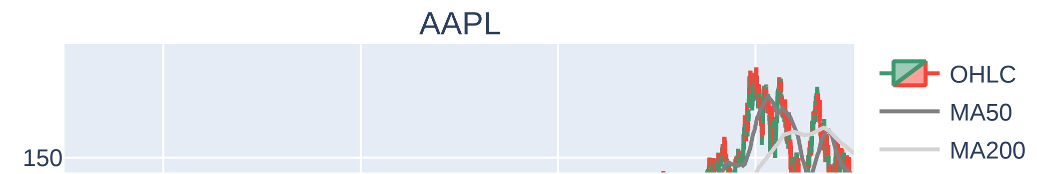
```
fig.add_trace(go.Scatter(x = df.index, y = df['MA200'], marker_color = 'lightgrey', name = 'MA200'), row = 1, col = 1)
```

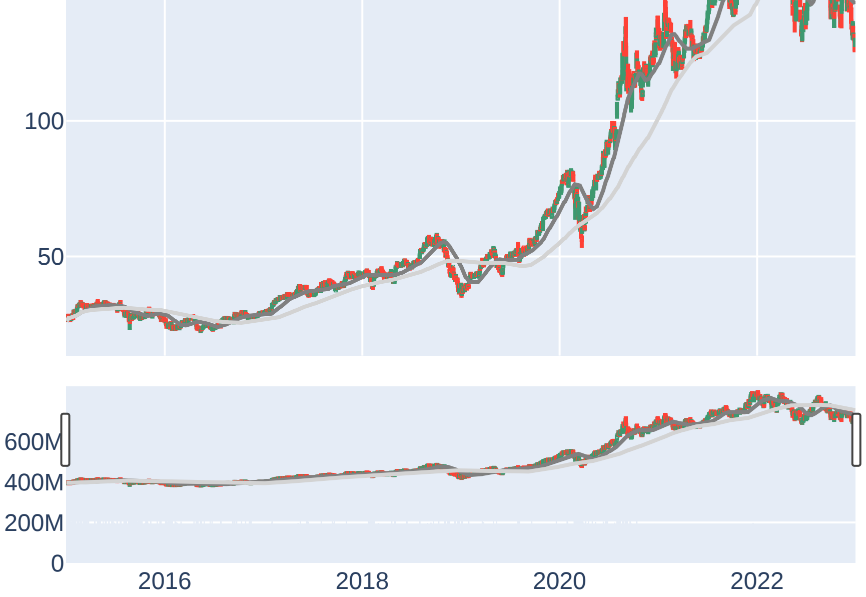


```
In [34]: # From above, we have an Open High Low Close graph with the moving average of 50 and 200
```

Adding volume bar chart in subplot

```
In [35]: fig.add_trace(go.Bar(x = df.index, y = df['Volume'], marker_color = 'red', showlegend = False), row = 2, col = 1)

         # This allows the volume graph to be only shown in the bottom graph and also not shown in the legend
```

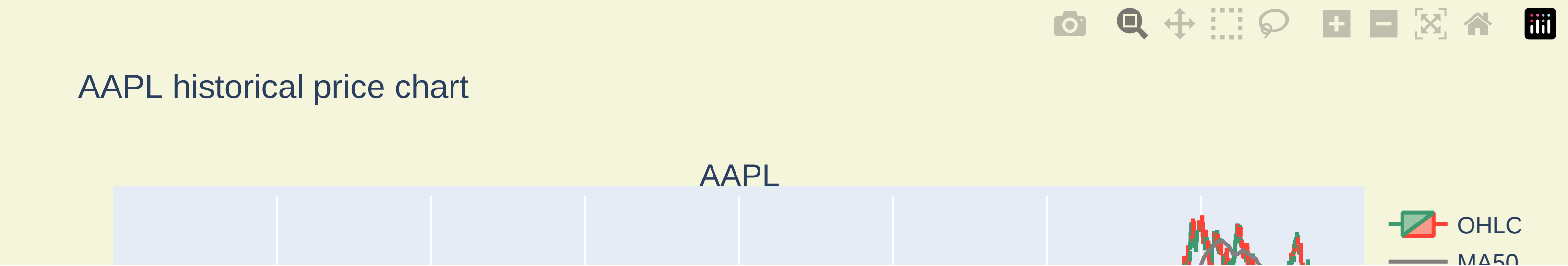


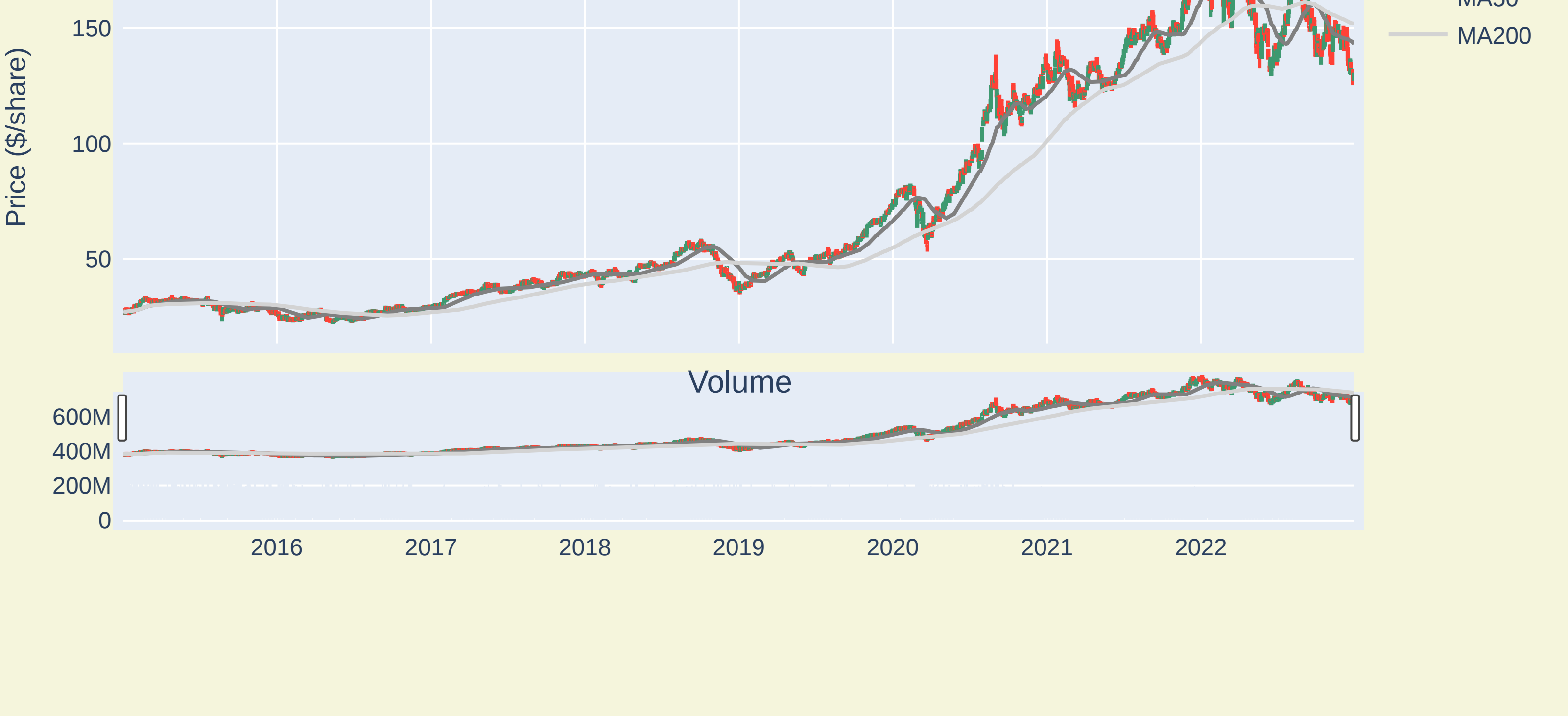


Updating layout with appropriate label, colors, and sizes

In [39]:

```
fig.update_layout(title = 'AAPL historical price chart',
                 xaxis_tickfont_size = 12,
                 yaxis = dict(
                     title = 'Price ($/share)',
                     titlefont_size = 14,
                     tickfont_size = 12
                 ),
                  autosize = False,
                  width = 800,
                  height = 500,
                  margin = dict(l = 50,r = 50,b = 100,t = 100, pad = 5),
                  paper_bgcolor = 'beige'
                 )
```

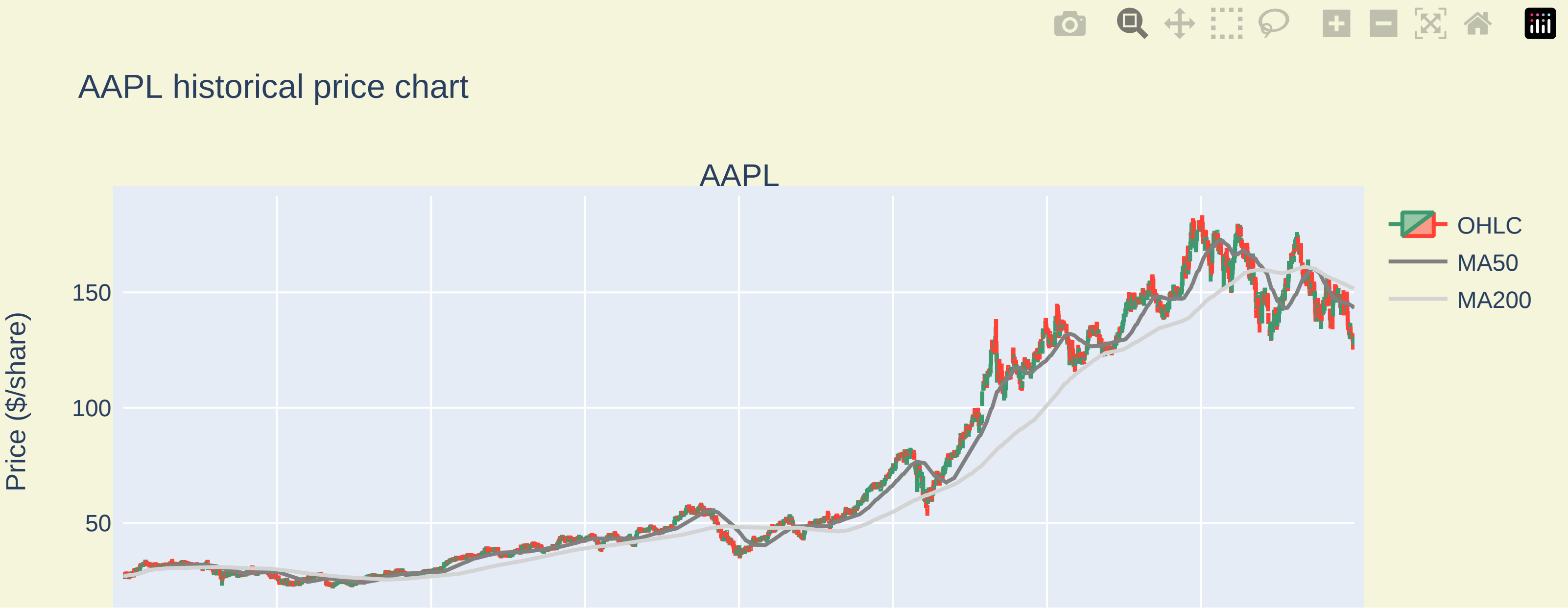


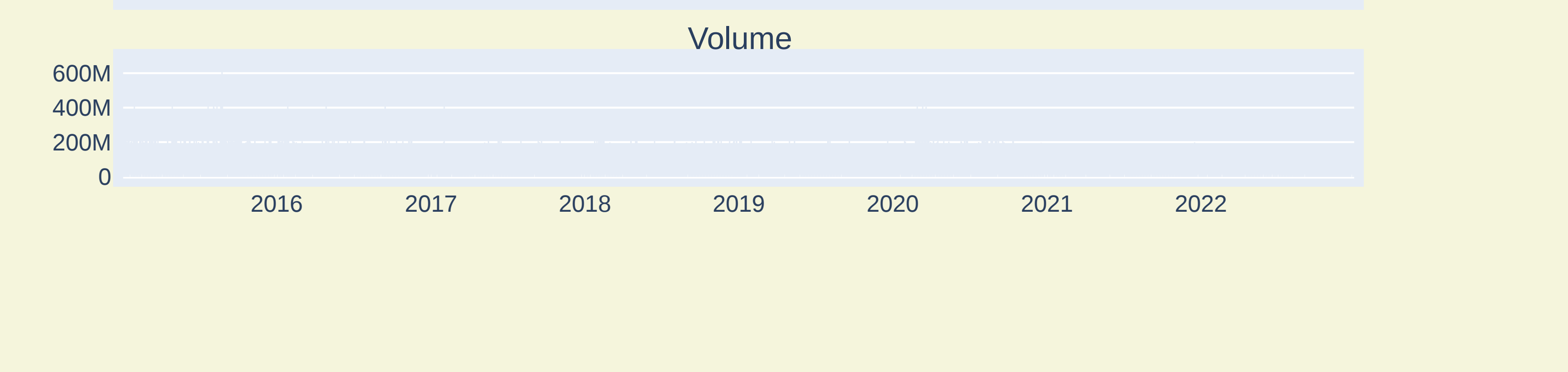


Removing the rangeslider from subplot

In [41]:

```
fig.update(layout_xaxis_rangeslider_visible = False)
fig.show()
```





In [ ]:

```
# Zoom in through drag and draw rectangle
# The volume plot would also change time frame accordingly.
```

In conclusion, I was successfully able to generate OHLC candlestick graphs utilizing stock data through plotly. Furthermore, I have incorporated the volume bar chart as well to see the volume for each corresponding time. During the process, I ensured the user's accessibility and readablity.