stock-data

September 20, 2023

[ ]:
```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

[ ]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True,␣
 ↪subplot_titles=("Historical Share Price", "Historical Revenue"),␣
 ↪vertical_spacing = .3)
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date,␣
 ↪infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Share␣
 ↪Price"), row=1, col=1)
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date,␣
 ↪infer_datetime_format=True), y=revenue_data.Revenue.astype("float"),␣
 ↪name="Revenue"), row=2, col=1)
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeslider_visible=True)
    fig.show()
```

[ ]:
```
tesla = yf.Ticker('TSLA')
```

[ ]:
```
tesla_data = tesla.history(period="max")
```

[ ]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head(5)
```

[ ]:
```
                       Date      Open      High       Low     Close  \
0 2010-06-29 00:00:00-04:00  1.266667  1.666667  1.169333  1.592667
1 2010-06-30 00:00:00-04:00  1.719333  2.028000  1.553333  1.588667
```

```
2 2010-07-01 00:00:00-04:00  1.666667  1.728000  1.351333  1.464000
3 2010-07-02 00:00:00-04:00  1.533333  1.540000  1.247333  1.280000
4 2010-07-06 00:00:00-04:00  1.333333  1.333333  1.055333  1.074000

      Volume  Dividends  Stock Splits
0  281494500        0.0           0.0
1  257806500        0.0           0.0
2  123282000        0.0           0.0
3   77097000        0.0           0.0
4  103003500        0.0           0.0
```

[ ]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data = requests.get(url).text
```

[ ]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<!--?xml version="1.0" encoding="utf-8"?-->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
 <head>
  <title>
   403 Forbidden
  </title>
 </head>
 <body>
  <h1>
   Error 403 Forbidden
  </h1>
  <p>
   Forbidden
  </p>
  <h3>
   Error 54113
  </h3>
  <p>
   Details: cache-iad-kcgs7200037-IAD 1693637072 2924664591
  </p>
  <hr/>
  <p>
   Varnish cache server
  </p>
 </body>
</html>
```

[ ]:
```
tesla_revenue = pd.DataFrame(columns = ["Date","Revenue"])

for table in soup.find_all('table'):
    if table.find('th').getText().startswith("Tesla Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            if len(col) != 2: continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

            tesla_revenue = tesla_revenue.append({"Date":Date, "Revenue":
 ↪Revenue}, ignore_index=True)
```

[ ]:
```
tesla_revenue.dropna(axis=0, how='all', subset=['Revenue']) #drop NaN values
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""] #drop empty␣
 ↪string values
```

[ ]:
```
tesla_revenue.tail(5)
```

[ ]:
```
Empty DataFrame
Columns: [Date, Revenue]
Index: []
```

[ ]:
```
gme = yf.Ticker('GME')
```

[ ]:
```
gme_data = gme.history(period = "max")
```

[ ]:
```
gme_data.reset_index(inplace=True)
gme_data.head(5)
```

[ ]:
```
                       Date      Open      High       Low     Close    Volume  \
0 2002-02-13 00:00:00-05:00  1.620128  1.693350  1.603296  1.691666  76216000
1 2002-02-14 00:00:00-05:00  1.712708  1.716074  1.670626  1.683251  11021600
2 2002-02-15 00:00:00-05:00  1.683251  1.687459  1.658002  1.674835   8389600
3 2002-02-19 00:00:00-05:00  1.666417  1.666417  1.578047  1.607504   7410400
4 2002-02-20 00:00:00-05:00  1.615920  1.662210  1.603296  1.662210   6892800

   Dividends  Stock Splits
0        0.0           0.0
1        0.0           0.0
2        0.0           0.0
3        0.0           0.0
4        0.0           0.0
```

[ ]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url).text
```

[ ]:
```
soup = BeautifulSoup(html_data, "html5lib")
print(soup.prettify())
```

```
<!--?xml version="1.0" encoding="utf-8"?-->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
 <head>
  <title>
   403 Forbidden
  </title>
 </head>
 <body>
  <h1>
   Error 403 Forbidden
  </h1>
  <p>
   Forbidden
  </p>
  <h3>
   Error 54113
  </h3>
  <p>
   Details: cache-iad-kcgs7200154-IAD 1693637233 547992884
  </p>
  <hr/>
  <p>
   Varnish cache server
  </p>
 </body>
</html>
```

[ ]:
```
gme_revenue = pd.DataFrame(columns = ["Date","Revenue"])

for table in soup.find_all('table'):
    if table.find('th').getText().startswith("GameStop Quarterly Revenue"):
        for row in table.find("tbody").find_all("tr"):
            col = row.find_all("td")
            if len(col) != 2: continue
            Date = col[0].text
            Revenue = col[1].text.replace("$","").replace(",","")

            gme_revenue = gme_revenue.append({"Date":Date, "Revenue":Revenue},␣
 ↪ignore_index=True)
```

[ ]:
```
gme_revenue.tail(5)
```

[ ]:
```
Empty DataFrame
Columns: [Date, Revenue]
Index: []
```

[ ]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla')
```

[ ]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

[ ]:
```

```