

```

# -*- coding: utf-8 -*-
"""Assignment_Matplotlib.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1w4512Japr0hAl0HfHkLs5Rp4c_SdRGoW

# In class assignment

- Get the daily stock prices for Google, IBM, and Apple from AlphaVantage using
their API
- Line chart: Compare the daily closing price of Google, IBM, and Apple stocks
for March 2025
- Some example codes are provided below. You should add your own code to produce
the a graph with similar characteristics as shown in the attached file.
- Show all the x-labels (dates on X Axis tick labels, where the dates are show
vertically)
- There are some missing dates and values (weekends/holidays)
- Consider how to fill those missing values
- Play with the y axis limit, usually it should start from 0.
"""

# import matlab and numpy
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

!pip install alpha_vantage

from alpha_vantage.timeseries import TimeSeries
ts = TimeSeries(key='TC54JI2GWZ0Z8I0J', output_format='pandas')
# Use the key
data_google, meta_data_google = ts.get_daily(symbol='GOOGL', outputsize='full')
data_ibm, meta_data_ibm = ts.get_daily(symbol='IBM', outputsize='full')
data_apple, meta_data_apple = ts.get_daily(symbol='AAPL', outputsize='full')
# use outputsize='full' for all data
# you may need to reset index for getting date into a column

print(meta_data_google)
print('---')
print(meta_data_ibm)
print('---')
print(meta_data_apple)
print('---')

```

```

print(data_google.info())
print('---')
print(data_ibm.info())
print('---')
print(data_apple.info())
print('---')

start_date = "2025-02-28" # for filling data purpose
end_date = "2025-03-31"

# add code to extract only the dates you needs
# use conditional row filtering based on start and end dates

data_google_filtered = data_google[(data_google.index >= start_date) &
(data_google.index <= end_date)]
data_ibm_filtered = data_ibm[(data_ibm.index >= start_date) & (data_ibm.index <=
end_date)]
data_apple_filtered = data_apple[(data_apple.index >= start_date) &
(data_apple.index <= end_date)]

print('#Google data')
print(data_google_filtered.head(1))
print('---')
print(data_google_filtered.tail(1))
print('')

print('#IBM data')
print(data_ibm_filtered.head(1))
print('---')
print(data_ibm_filtered.tail(1))
print('')

print('#Apple data')
print(data_apple_filtered.head(1))
print('---')
print(data_apple_filtered.tail(1))

data_google_filtered

new_columns = ["open", "high", "low", "close", "volume"] #name columns

for df in [data_google_filtered, data_ibm_filtered, data_apple_filtered]:
    df.columns = new_columns

```

```

# set date
full_date_range = pd.date_range(start='2025-02-28', end='2025-03-31', freq='D')

# Fill missing data
for name, df in zip(['google', 'ibm', 'apple'],
                    [data_google_filtered, data_ibm_filtered,
data_apple_filtered]):
    df = df.reindex(full_date_range) # reset index date
    df.fillna(method='ffill', inplace=True) # fill forward
    df.index.name = 'date'
    if name == 'google':
        data_google_filtered = df
    elif name == 'ibm':
        data_ibm_filtered = df
    else:
        data_apple_filtered = df

data_google_filtered

dfs = [data_google_filtered, data_ibm_filtered, data_apple_filtered]

# delete the first row (28 February 2025)
for i in range(len(dfs)):
    dfs[i] = dfs[i].iloc[1:]

data_google_filtered, data_ibm_filtered, data_apple_filtered = dfs

data_google_filtered

data_ibm_filtered

data_apple_filtered

#add index and turn index date to column
data_google_filtered = data_google_filtered.reset_index()
data_ibm_filtered = data_ibm_filtered.reset_index()
data_apple_filtered = data_apple_filtered.reset_index()

data_google_filtered

# styel & font
plt.rcParams['font.size'] = 16
plt.rcParams['lines.linewidth'] = 2
plt.style.use('tableau-colorblind10')

```

```
# figure
fig = plt.figure(figsize=(10,6))

# plot
plt.plot(data_google_filtered['date'], data_google_filtered['close'],
label='Google', marker='*')
plt.plot(data_ibm_filtered['date'], data_ibm_filtered['close'],
label='IBM',marker='o')
plt.plot(data_apple_filtered['date'], data_apple_filtered['close'],
label='Apple',marker='x')

# label
plt.gca().set_xticks(data_google_filtered['date'])
plt.xticks(rotation=90)

plt.xlabel('Date')
plt.ylabel('Closing Price (USD)')
plt.title('Daily Closing Prices - March 2025')
plt.ylim(0) # Y starts with 0
plt.legend()
plt.grid(True)
plt.tight_layout()

plt.show()
```