Technological Institute of the Philippines
Computer Engineering Department
Quezon City Campus

Hands-on Activity 7.1 Data Collection and Wrangling

Course: CPE 311	Program: BSCpE
Course Title: Computational Thinking with Python	**Date Performed:**
Section: CPE22S3	**Date Submitted:**
Student Name: Jhillian M. Cabos	**Instructor's Name:** Roman Richard

7.1 Supplementary Activity

Using the datasets provided, perform the following exercises:

Exercise 1

We want to look at data for the Facebook, Apple, Amazon, Netflix, and Google (FAANG) stocks, but we were given each as a separate CSV file. Combine them into a single file and store the dataframe of the FAANG data as faang for the rest of the exercises:

1. Read each file in.
2. Add a column to each dataframe, called ticker, indicating the ticker symbol it is for (Apple's is AAPL, for example). This is how you look up a stock. Each file's name is also the ticker symbol, so be sure to capitalize it.
3. Append them together into a single dataframe.
4. Save the result in a CSV file called faang.csv.

```
import pandas as pd


netflix_df = pd.read_csv('nflx.csv')
netflix_df.head()
```

	date	open	high	low	close	volume
0	2018-01-02	196.10	201.65	195.4200	201.07	10966889
1	2018-01-03	202.05	206.21	201.5000	205.05	8591369
2	2018-01-04	206.20	207.05	204.0006	205.63	6029616
3	2018-01-05	207.25	210.02	205.5900	209.99	7033240
4	2018-01-08	210.02	212.50	208.4400	212.05	5580178

Netflix Dataframe and renaming

```
netflix_df = pd.read_csv('nflx.csv', usecols=['date', 'open', 'high', 'low', 'close', 'volume'])
netflix_df = netflix_df.rename(columns={
    'date': 'nflxdate',
    'open': 'nflxopen',
    'high': 'nflxhigh',
    'low': 'nflxlow',
    'close': 'nflxclose',
    'volume': 'nflxvolume'})
netflix_df.to_csv('netflix.csv', index=False)
netflix_df.head()
```

	nflxdate	nflxopen	nflxhigh	nflxlow	nflxclose	nflxvolume
0	2018-01-02	196.10	201.65	195.4200	201.07	10966889
1	2018-01-03	202.05	206.21	201.5000	205.05	8591369
2	2018-01-04	206.20	207.05	204.0006	205.63	6029616
3	2018-01-05	207.25	210.02	205.5900	209.99	7033240
4	2018-01-08	210.02	212.50	208.4400	212.05	5580178

Facebook Dataframe and renaming

```
fb_df = pd.read_csv('fb.csv')
fb_df.head()
```

	date	open	high	low	close	volume
0	2018-01-02	177.68	181.58	177.5500	181.42	18151903
1	2018-01-03	181.88	184.78	181.3300	184.67	16886563
2	2018-01-04	184.90	186.21	184.0996	184.33	13880896
3	2018-01-05	185.59	186.90	184.9300	186.85	13574535
4	2018-01-08	187.20	188.90	186.3300	188.28	17994726

```
fb_df = pd.read_csv('fb.csv', usecols=['date', 'open', 'high', 'low', 'close', 'volume'])
fb_df = fb_df.rename(columns={
    'date': 'fbdate',
    'open': 'fbopen',
    'high': 'fbhigh',
    'low': 'fblow',
    'close': 'fbclose',
    'volume': 'fbvolume'})
fb_df.head()
```

	fbdate	fbopen	fbhigh	fblow	fbclose	fbvolume
0	2018-01-02	177.68	181.58	177.5500	181.42	18151903
1	2018-01-03	181.88	184.78	181.3300	184.67	16886563
2	2018-01-04	184.90	186.21	184.0996	184.33	13880896
3	2018-01-05	185.59	186.90	184.9300	186.85	13574535
4	2018-01-08	187.20	188.90	186.3300	188.28	17994726

```
fb_df.to_csv('facebook.csv', index=False)
```

Google

```
google_df = pd.read_csv('goog.csv', usecols=['date', 'open', 'high', 'low', 'close', 'volume'])
google_df = google_df.rename(columns={
    'date': 'googledate',
    'open': 'googleopen',
    'high': 'googlehigh',
    'low': 'googlelow',
    'close': 'googleclose',
    'volume': 'googlevolume'})
google_df.head()
```

	googledate	googleopen	googlehigh	googlelow	googleclose	googlevolume
0	2018-01-02	1048.34	1066.94	1045.23	1065.00	1237564
1	2018-01-03	1064.31	1086.29	1063.21	1082.48	1430170
2	2018-01-04	1088.00	1093.57	1084.00	1086.40	1004605
3	2018-01-05	1094.00	1104.25	1092.00	1102.23	1279123
4	2018-01-08	1102.23	1111.27	1101.62	1106.94	1047603

```
google_df.to_csv('google.csv', index=False)
```

Amazon

```
amazon_df = pd.read_csv('amzn.csv', usecols=['date', 'open', 'high', 'low', 'close', 'volume'])
amazon_df = amazon_df.rename(columns={
    'date': 'amazondate',
    'open': 'amazonopen',
    'high': 'amazonhigh',
    'low': 'amazonlow',
    'close': 'amazonclose',
    'volume': 'amazonvolume'})
amazon_df.head()
```

	amazondate	amazonopen	amazonhigh	amazonlow	amazonclose	amazonvolume
0	2018-01-02	1172.00	1190.00	1170.51	1189.01	2694494
1	2018-01-03	1188.30	1205.49	1188.30	1204.20	3108793
2	2018-01-04	1205.00	1215.87	1204.66	1209.59	3022089
3	2018-01-05	1217.51	1229.14	1210.00	1229.14	3544743
4	2018-01-08	1236.00	1253.08	1232.03	1246.87	4279475

```
amazon_df.to_csv('amazon.csv', index=False)
```

Apple

```
apple_df = pd.read_csv('aapl.csv', usecols=['date', 'open', 'high', 'low', 'close', 'volume'])
apple_df = apple_df.rename(columns={
    'date': 'appledate',
    'open': 'appleopen',
    'high': 'applehigh',
    'low': 'applelow',
    'close': 'appleclose',
    'volume': 'applevolume'})
apple_df.head()
```

	appledate	appleopen	applehigh	applelow	appleclose	applevolume
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934
1	2018-01-03	169.2521	171.2337	168.6929	168.9578	29517899
2	2018-01-04	169.2619	170.1742	168.8106	169.7426	22434597
3	2018-01-05	170.1448	172.0381	169.7622	171.6751	23660018
4	2018-01-08	171.0375	172.2736	170.6255	171.0375	20567766

```
apple_df.to_csv('apple.csv', index=False)
```

Combining them together

```
dfs = {}
csv_files = ['google.csv', 'amazon.csv', 'apple.csv', 'facebook.csv', 'netflix.csv']
for file in csv_files:
    df_name = file.split('.')[0]
    dfs[df_name] = pd.read_csv(file)
FAANG_df = pd.concat(dfs.values(), axis=1)
```

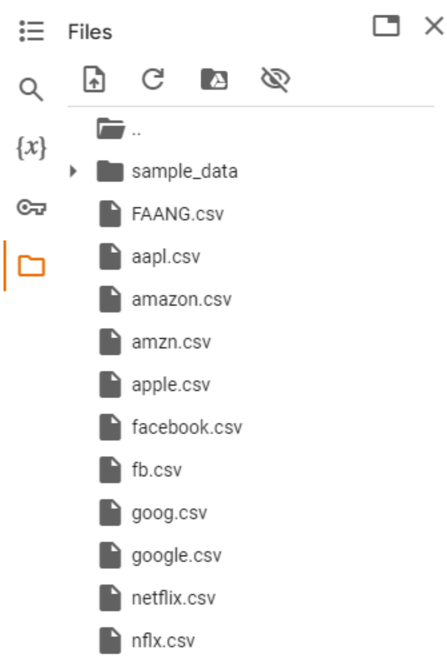
```
FAANG_df = pd.concat(dfs.values(), axis=1)
FAANG_df.to_csv('FAANG.csv', index=False)
```

```
pd.read_csv('FAANG.csv')
```

	googledate	googleopen	googlehigh	googlelow	googleclose	googlevolume	amazon
0	2018-01-02	1048.34	1066.94	1045.23	1065.00	1237564	2018-0
1	2018-01-03	1064.31	1086.29	1063.21	1082.48	1430170	2018-0
2	2018-01-04	1088.00	1093.57	1084.00	1086.40	1004605	2018-0
3	2018-01-05	1094.00	1104.25	1092.00	1102.23	1279123	2018-0
4	2018-01-08	1102.23	1111.27	1101.62	1106.94	1047603	2018-0
...	...	...	...	...	...	...	
246	2018-12-24	973.90	1003.54	970.11	976.22	1590328	2018-1
247	2018-12-26	989.01	1040.00	983.00	1039.46	2373270	2018-1
248	2018-12-27	1017.15	1043.89	997.00	1043.88	2109777	2018-1
249	2018-12-28	1049.62	1055.56	1033.10	1037.08	1413772	2018-1
250	2018-12-31	1050.96	1052.70	1023.59	1035.61	1493722	2018-1

251 rows × 30 columns

Files Output:



keyboard_arrow_down Corrected way on doing this activity

```
df = pd.read_csv('nflx.csv')
df['ticker'] = 'NFLX'
df.to_csv('nflx.csv', index=False)


df_fb = pd.read_csv('fb.csv')
df_fb['ticker'] = 'FB'
df_fb.to_csv('fb.csv', index=False)


df_goog = pd.read_csv('goog.csv')
df_goog['ticker'] = 'GOOG'
df_goog.to_csv('goog.csv', index=False)


df_amzn = pd.read_csv('amzn.csv')
df_amzn['ticker'] = 'AMZN'
df_amzn.to_csv('amzn.csv', index=False)


df_appl = pd.read_csv('/content/aapl.csv')
df_appl['ticker'] = 'APPL'
df_appl.to_csv('appl.csv', index=False)


dfs = {}
csv_files = ['goog.csv', 'amzn.csv', 'aapl.csv', 'fb.csv', 'nflx.csv']
for file in csv_files:
    df_name = file.split('.')[0]
    dfs[df_name] = pd.read_csv(file)
FAANG_df = pd.concat(dfs.values(), axis=1)
FAANG_df.to_csv('faang.csv', index=False)
```

Exercise 2

- With faang, use type conversion to change the date column into a datetime and the volume column into integers. Then, sort by date and ticker.
- Find the seven rows with the highest value for volume.
- Right now, the data is somewhere between long and wide format. Use melt() to make it completely long format. Hint: date and ticker are our ID variables (they uniquely identify each row). We need to melt the rest so that we don't have separate columns for open, high, low, close, and volume.

```
FAANG_df = pd.read_csv('faang.csv')
FAANG_df.head()
```

	date	open	high	low	close	volume	ticker	date.1	open.1	high.1
0	2018-01-02	1048.34	1066.94	1045.23	1065.00	1237564	GOOG	2018-01-02	1172.00	1190.00
1	2018-01-03	1064.31	1086.29	1063.21	1082.48	1430170	GOOG	2018-01-03	1188.30	1205.49
2	2018-01-04	1088.00	1093.57	1084.00	1086.40	1004605	GOOG	2018-01-04	1205.00	1215.87
3	2018-01-05	1094.00	1104.25	1092.00	1102.23	1279123	GOOG	2018-01-05	1217.51	1229.14

```
df_faang = pd.read_csv('faang.csv')
df_faang['date'] = pd.to_datetime(df_faang['date'])
df_faang['volume'] = df_faang['volume'].astype(int)
df_faang_sorted = df_faang.sort_values(by=['date', 'ticker'])
df_faang_sorted.to_csv('faang_sorted.csv', index=False)


df_sorted_csv = pd.read_csv('faang_sorted.csv')
(df_sorted_csv)
```

	date	open	high	low	close	volume	ticker	date.1	open.1	high.1
0	2018-01-02	1048.34	1066.94	1045.23	1065.00	1237564	GOOG	2018-01-02	1172.00	1190.00
1	2018-01-03	1064.31	1086.29	1063.21	1082.48	1430170	GOOG	2018-01-03	1188.30	1205.49
2	2018-01-04	1088.00	1093.57	1084.00	1086.40	1004605	GOOG	2018-01-04	1205.00	1215.87
3	2018-01-05	1094.00	1104.25	1092.00	1102.23	1279123	GOOG	2018-01-05	1217.51	1229.14
4	2018-01-08	1102.23	1111.27	1101.62	1106.94	1047603	GOOG	2018-01-08	1236.00	1253.08
...	...	...	...	...	...	...	...	...	...	...
246	2018-12-24	973.90	1003.54	970.11	976.22	1590328	GOOG	2018-12-24	1346.00	1396.03
247	2018-12-26	989.01	1040.00	983.00	1039.46	2373270	GOOG	2018-12-26	1368.89	1473.16

7 rows with highes values

```
df_faang = pd.read_csv('faang.csv')
top_volume = df_faang.nlargest(7, 'volume')
(top_volume)
```

	date	open	high	low	close	volume	ticker	date.1	open.1	high.1
22	2018-02-02	1122.00	1123.07	1107.28	1111.90	4857943	GOOG	2018-02-02	1477.39	1498.00
77	2018-04-24	1052.00	1057.00	1010.59	1019.98	4760260	GOOG	2018-04-24	1535.80	1539.50
245	2018-12-21	1015.30	1024.02	973.69	979.54	4595891	GOOG	2018-12-21	1464.99	1480.00
182	2018-09-21	1192.00	1192.21	1166.04	1166.09	4405584	GOOG	2018-09-21	1954.22	1957.31
207	2018-10-26	1037.03	1106.53	1034.09	1071.47	4187586	GOOG	2018-10-26	1649.59	1698.46

Using melt()

```
df_faang = pd.read_csv('faang.csv')
df_long_format = pd.melt(df_faang, id_vars=['date', 'ticker'], value_vars=['open', 'high', 'low', 'close', 'volume'], var_name='variable
(df_long_format)
```

	date	ticker	variable	value
0	2018-01-02	GOOG	open	1048.34
1	2018-01-03	GOOG	open	1064.31
2	2018-01-04	GOOG	open	1088.00
3	2018-01-05	GOOG	open	1094.00
4	2018-01-08	GOOG	open	1102.23
...	...	...	...	...
1250	2018-12-24	GOOG	volume	1590328.00
1251	2018-12-26	GOOG	volume	2373270.00
1252	2018-12-27	GOOG	volume	2109777.00
1253	2018-12-28	GOOG	volume	1413772.00
1254	2018-12-31	GOOG	volume	1493722.00

1255 rows × 4 columns

Exercise 3

- Using web scraping, search for the list of the hospitals, their address and contact information. Save the list in a new csv file, hospitals.csv.
- Using the generated hospitals.csv, convert the csv file into pandas dataframe. Prepare the data using the necessary preprocessing techniques.

```
import requests, csv
from bs4 import BeautifulSoup
url = "https://nhfr.doh.gov.ph/"
response = requests.get(url)
if response.status_code == 200:
    soup = BeautifulSoup(response.content, 'html.parser')
    hospitals = soup.find_all('div', class_='hospital')
    hospital_data = []
    for hospital in hospitals:
        name = hospital.find('h2').text.strip()
        address = hospital.find('p', class_='address').text.strip()
        contact = hospital.find('p', class_='contact').text.strip()
        hospital_data.append({'Name': name, 'Address': address, 'Contact': contact})
    with open('hospital.csv', 'w', newline='', encoding='utf-8') as file:
        writer = csv.DictWriter(file, fieldnames=['Name', 'Address', 'Contact'])
        writer.writeheader()
        writer.writerows(hospital_data)


pd.read_csv('hospital.csv')
```

ıtp

	index	Provider ID	Hospital Name	Address	City	State	ZIP Code	County Name	Phone Number	Hospital Type	...	Readmission national comparison footnote	P expe na comp
0	0	10005	MARSHALL MEDICAL CENTER SOUTH	2505 U S HIGHWAY 431 NORTH	BOAZ	AL	35957	MARSHALL	2565938310	Acute Care Hospitals	...	NaN	Sam

```
hospital_df = pd.read_csv('hospital.csv')
hospital_df['date'] = pd.to_datetime(df_faang['date'])
hospital_df_sorted = hospital_df.sort_values(by=['date'])
hospital_df_sorted.to_csv('hospital.csv', index=False)
```

keyboard_arrow_down

Conclusion

PROVIDENCE

Even though I had to check the libraries and get references online it was such a great experience to do this since I was able to get data through functions and know I see the sorting of our simple file managers in a new light still kinda confusing though

,,,,,

			HOSPITAL										
...	...	...	...	...	...	...	...	...	...	...	...	...	
4813	4813	501333	KITTITAS VALLEY COMMUNITY	603 SOUTH CHESTNUT	ELLENSBURG	WA	98926	KITTITAS	5099629841	Critical Access Hospitals	...	NaN	Ab