Module 7: Data Wrangling with Pandas

CPE 311 Computational Thinking with Python

Submitted by: Corpuz, Micki Laurren B.

Performed on: 04/07/2025

Submitted on: 04/07/2025

Submitted to: Engr. Roman M. Richard

7.1 Supplementary Activity

Using the dataset provided, perform the following exercises:

Exercise 1

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

> 1. Read Each file in

```
import pandas as pd

amzn = pd.read_csv('/content/amzn.csv')
aapl = pd.read_csv('/content/aapl.csv')
fb = pd.read_csv('/content/fb.csv')
goog = pd.read_csv('/content/goog.csv')
nflx = pd.read_csv('/content/nflx.csv')
```

> 2. Add a column to each dataframe, called ticker, indication 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.

```
amzn['ticker'] = 'AMZN'
aapl['ticker'] = 'AAPL'
fb['ticker']   = 'FB'
goog['ticker'] = 'GOOG'
nflx['ticker'] = 'NFLX'

print('Preview Ticker: ')
aapl.head(3)
```

```
Preview Ticker:
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	AAPL
1	2018-01-03	169.2521	171.2337	168.6929	168.9578	29517899	AAPL
2	2018-01-04	169.2619	170.1742	168.8106	169.7426	22434597	AAPL

```
amzn.head(3)
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	1172.0	1190.00	1170.51	1189.01	2694494	AMZN
1	2018-01-03	1188.3	1205.49	1188.30	1204.20	3108793	AMZN
2	2018-01-04	1205.0	1215.87	1204.66	1209.59	3022089	AMZN

```
fb.head(3)
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	177.68	181.58	177.5500	181.42	18151903	FB
1	2018-01-03	181.88	184.78	181.3300	184.67	16886563	FB
2	2018-01-04	184.90	186.21	184.0996	184.33	13880896	FB

```
goog.head(3)
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	1048.34	1066.94	1045.23	1065.00	1237564	GOOG
1	2018-01-03	1064.31	1086.29	1063.21	1082.48	1430170	GOOG
2	2018-01-04	1088.00	1093.57	1084.00	1086.40	1004605	GOOG

```
nflx.head(3)
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	196.10	201.65	195.4200	201.07	10966889	NFLX
1	2018-01-03	202.05	206.21	201.5000	205.05	8591369	NFLX
2	2018-01-04	206.20	207.05	204.0006	205.63	6029616	NFLX

> 3. Append them together into a single dataframe.

```
faang = pd.concat([aapl, amzn, fb, goog, nflx], ignore_index=True)
```

```
# Check first 5 entries
faang.head()
```

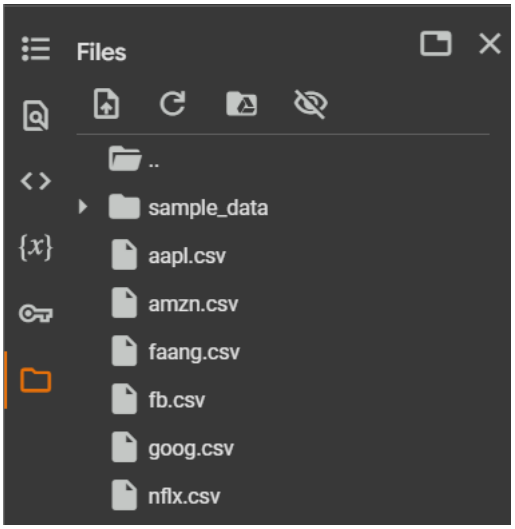
	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	AAPL
1	2018-01-03	169.2521	171.2337	168.6929	168.9578	29517899	AAPL
2	2018-01-04	169.2619	170.1742	168.8106	169.7426	22434597	AAPL
3	2018-01-05	170.1448	172.0381	169.7622	171.6751	23660018	AAPL
4	2018-01-08	171.0375	172.2736	170.6255	171.0375	20567766	AAPL

```
# Check last five entries
faang.tail()
```

	date	open	high	low	close	volume	ticker
1250	2018-12-24	242.00	250.6500	233.68	233.880	9547616	NFLX
1251	2018-12-26	233.92	254.5000	231.23	253.670	14402735	NFLX
1252	2018-12-27	250.11	255.5900	240.10	255.565	12235217	NFLX
1253	2018-12-28	257.94	261.9144	249.80	256.080	10987286	NFLX
1254	2018-12-31	260.16	270.1001	260.00	267.660	13508920	NFLX

> 4. Save the result in a CSV filed called faang csv.

```
faang.to_csv('/content/faang.csv')
```



Excecise 2

> 1. 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.

```
faang.dtypes
```

	0
date	object
open	float64
high	float64
low	float64
close	float64
volume	int64
ticker	object

dtype: object

```
faang['date'] = pd.to_datetime(faang['date'])

faang['volume'] = pd.to_numeric(faang['volume'])


# Check if recent changes were already committed
faang.dtypes
```

	0
date	datetime64[ns]
open	float64
high	float64
low	float64
close	float64
volume	int64
ticker	object

dtype: object

```
sorted_df = faang.sort_values(by = ['date', 'ticker'])
```

```
sorted_df.head()
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	AAPL
251	2018-01-02	1172.0000	1190.0000	1170.5100	1189.0100	2694494	AMZN
502	2018-01-02	177.6800	181.5800	177.5500	181.4200	18151903	FB
753	2018-01-02	1048.3400	1066.9400	1045.2300	1065.0000	1237564	GOOG
1004	2018-01-02	196.1000	201.6500	195.4200	201.0700	10966889	NFLX

> 2. Find the seven rows with the hihest value for volume.

```
faang.nlargest(n = 7, columns = 'volume')
```

	date	open	high	low	close	volume	ticker
644	2018-07-26	174.8900	180.1300	173.7500	176.2600	169803668	FB
555	2018-03-20	167.4700	170.2000	161.9500	168.1500	129851768	FB
559	2018-03-26	160.8200	161.1000	149.0200	160.0600	126116634	FB
556	2018-03-21	164.8000	173.4000	163.3000	169.3900	106598834	FB
182	2018-09-21	219.0727	219.6482	215.6097	215.9768	96246748	AAPL
245	2018-12-21	156.1901	157.4845	148.9909	150.0862	95744384	AAPL
212	2018-11-02	207.9295	211.9978	203.8414	205.8755	91328654	AAPL

> 3. 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 seperate coolumns for open, high, low, close, and volume.

```
melted_df = sorted_df.melt(
    id_vars = ['date', 'ticker'],
    value_vars = ['open', 'high', 'low', 'close', 'volume']
)
melted_df
```

	date	ticker	variable	value
0	2018-01-02	AAPL	open	1.669271e+02
1	2018-01-02	AMZN	open	1.172000e+03
2	2018-01-02	FB	open	1.776800e+02
3	2018-01-02	GOOG	open	1.048340e+03
4	2018-01-02	NFLX	open	1.961000e+02
...	...	...	...	...
6270	2018-12-31	AAPL	volume	3.500347e+07
6271	2018-12-31	AMZN	volume	6.954507e+06
6272	2018-12-31	FB	volume	2.462531e+07
6273	2018-12-31	GOOG	volume	1.493722e+06
6274	2018-12-31	NFLX	volume	1.350892e+07

6275 rows × 4 columns

Excecise 3

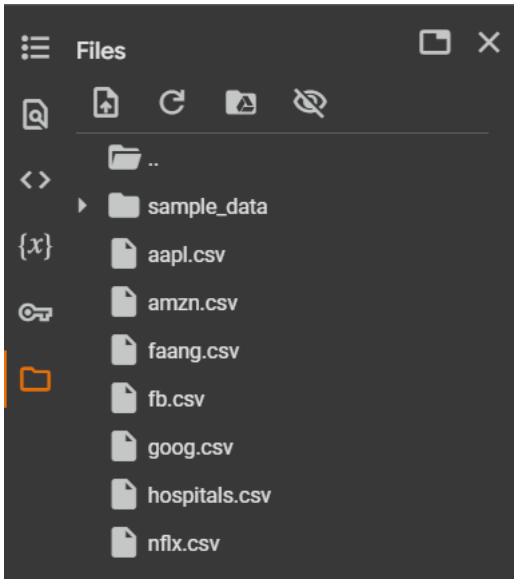
> 1. Using web scraping, search for the list of hospitals, their address and contact information. Save the list in a new csv file, hospitals.csv.

```
import requests as rq
from bs4 import BeautifulSoup

url = 'https://sulit.ph/list-of-hospitals-in-metro-manila-with-contact-details-website-and-social-media-accounts/'
request = rq.get(url)
```

```
request
```

```
<Response [200]>
```

```
# Save DataFrame to CSV
df.to_csv('hospitals.csv')
```




```
hospitals = pd.read_csv('hospitals.csv')
hospitals.head()
```

	CITY	NAME OF HOSPITAL	CONTACT NUMBER	WEBSITE / EMAIL	FACEBOOK LINK
0	LIST UPDATE	NaN	NaN	NaN	NaN
1	15 SEPT 2021	NaN	NaN	NaN	NaN
2	Caloocan	Caloocan City Medical Center	South 5310 7925, North 8282 3397, 0943 216 6963	NaN	https://www.facebook.com/Caloocan-City-Medical...
3	Caloocan	Dr. Jose N. Rodriguez Memorial Hospital and Sa...	0966 549 2697, 8294 2571 to 73	http://djnrmh.doh.gov.ph/	https://www.facebook.com/officialDJNRMHS
4	[illegible]	MCU – FDT Medical	[illegible]	[illegible]	[illegible]

Next steps: View recommended plots New interactive sheet

> 2. Using the generated hospitals.csv, convert the csv file into pandas dataframe. Prepare the data using the necessary preprocessing techniques.

```
# Since the first two entries are not actual cities but --- by judgement --- were mere update notice
# It can be dropped

new_hospitals = hospitals.drop([0, 1]).reset_index(drop=True)
new_hospitals.head()

new_hospitals.to_csv('new_hospitals.csv')
new_hospitals.head()
```

	CITY	NAME OF HOSPITAL	CONTACT NUMBER	WEBSITE / EMAIL	FACEBOOK LINK
0	Caloocan	Caloocan City Medical Center	South 5310 7925, North 8282 3397, 0943 216 6963	NaN	https://www.facebook.com/Caloocan-City-Medical...
1	Caloocan	Dr. Jose N. Rodriguez Memorial Hospital and Sa...	0966 549 2697, 8294 2571 to 73	http://djnrmh.doh.gov.ph/	https://www.facebook.com/officialDJNRMHS
2	Caloocan	MCU – FDT Medical Foundations Hospital	8367 2031	https://www.mcuhospital.org/	NaN
		Metro Balayan Medical			

Next steps: View recommended plots New interactive sheet

```
new_hospitals.dtypes
```

	0
CITY	object
NAME OF HOSPITAL	object
CONTACT NUMBER	object
WEBSITE / EMAIL	object
FACEBOOK LINK	object

dtype: object

```
# Preprocessing steps (e.g., cleaning and formatting)
process = new_hospitals.copy()

process = process.rename(
    columns = {
        'CITY': 'City',
        'NAME OF HOSPITAL': 'Hospital Name',
        'CONTACT NUMBER': 'Contact',
        'WEBSITE/EMAIL': 'Website/ Email',
        'FACEBOOK LINK': 'Facebook Link'
    }
)

process.head()
```

	City	Hospital Name	Contact	WEBSITE / EMAIL	Facebook Link
0	Caloocan	Caloocan City Medical Center	South 5310 7925, North 8282 3397, 0943 216 6963	NaN	https://www.facebook.com/Caloocan-City-Medical...
1	Caloocan	Dr. Jose N. Rodriguez Memorial Hospital and Sa...	0966 549 2697, 8294 2571 to 73	http://djnrmh.doh.gov.ph/	https://www.facebook.com/officialDJNRMHS
2	Caloocan	MCU – FDT Medical Foundations Hospital	8367 2031	https://www.mcuhospital.org/	NaN
		Metro Balayan Medical			

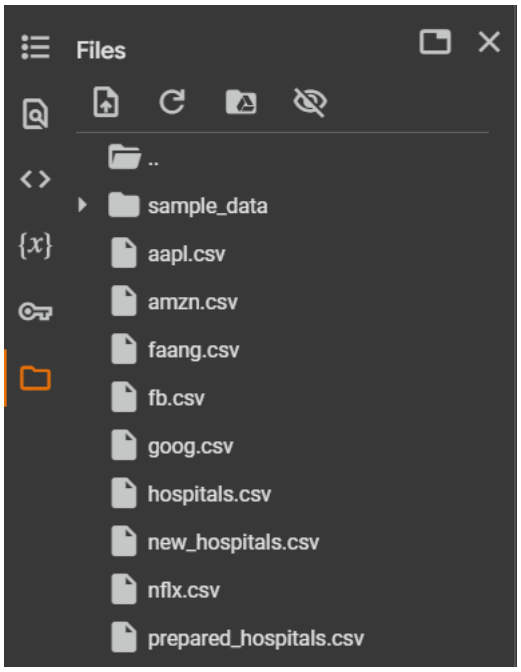
Next steps: View recommended plots New interactive sheet

```
# a. Check for missing values
process.isnull().sum()

# b. Fill missing values or drop rows/columns as needed
process.fillna('Unknown', inplace=True)

# c. Example of ensuring that phone numbers are strings (if necessary)
process['Contact'] = process['Contact'].astype(str)

# d. Add column
process['Contact Format'] = process['Contact'].apply(lambda x: 'Valid' if len(str(x)) >= 7 else 'Invalid')

process
```

	City	Hospital Name	Contact	WEBSITE / EMAIL	Facebook Link	Contact Format
0	Caloocan	Caloocan City Medical Center	South 5310 7925, North 8282 3397, 0943 216 6963	Unknown	https://www.facebook.com/Caloocan-City-Medical...	Valid
1	Caloocan	Dr. Jose N. Rodriguez Memorial Hospital and Sa...	0966 549 2697, 8294 2571 to 73	http://djnrmh.doh.gov.ph/	https://www.facebook.com/officialDJNRMHS	Valid
2	Caloocan	MCU – FDT Medical Foundations Hospital	8367 2031	https://www.mcuhospital.org/	Unknown	Valid
3	Caloocan	Metro Balayan Medical Center	(043) 740 1350	http://www.metrobalayanmc.com.ph/	https://www.facebook.com/metrobalayan/	Valid
4	Las Pinas	Alabang Medical Center	8807 8189, 8850 8719	Unknown	https://www.facebook.com/alabangmedicalcenter	Valid
...	...	...	...	...	...	...
91	Taguig	Medical Center of Taguig	8888 6284	Unknown	https://www.facebook.com/mctadminofficial/	Valid
92	Valenzuela	Allied Care Experts (ACE)	direct line to Admission 0917	Unknown	https://www.facebook.com/ACEMC-Valenzuela-	Valid

Next steps: View recommended plots New interactive sheet

```
# Save the processed DataFrame to a new CSV
process.to_csv('prepared_hospitals.csv', sep='|')
```



7.2 Conclusion

> In this activity, I worked on several tasks related to data processing in Python, such as web scraping, cleaning, and saving data using Pandas. I learned how to rename columns, reset indexes, change data types, and save DataFrames to CSV files with custom separators. I also practiced converting columns to strings. Throughout the process, I ran into some unfamiliar syntax, so I had to check online from time to time to ensure I was on the right track. It was a good reminder that it's okay to seek help when learning new things, and that small adjustments in code can significantly impact how data is handled and saved.