Module 7: Data Wrangling using pandas

- Submitted by: Prince Wally G. Esteban
- Performed on: 04/07/25
- Submitted on: 04/07/25
- Submited to: Engr. Roman Richard

Excercise 1

```
# 1. read each file in
import pandas as pd
apple = pd.read_csv('aapl.csv')
amazon = pd.read_csv('amzn.csv')
facebook = pd.read_csv('fb.csv')
google = pd.read_csv('goog.csv')
netflix = pd.read_csv('nflx.csv')
```

```
# 2. Add ticker column
apple['ticker'] = 'AAPL'
amazon['ticker'] = 'AMZN'
facebook['ticker'] = 'FB'
google['ticker'] = 'GOOG'
netflix['ticker'] = 'NFLX'
```

```
# 3. Combine all DataFrames
faang = pd.concat([apple, amazon, facebook, google, netflix], ignore_index=True)
```

```
# 4. Save to CSV
faang.to_csv('faang.csv', index=False)
```

```
faang
```

	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
...	...	...	...	...	...	...	...
1250	2018-12-24	242.0000	250.6500	233.6800	233.8800	9547616	NFLX
1251	2018-12-26	233.9200	254.5000	231.2300	253.6700	14402735	NFLX
1252	2018-12-27	250.1100	255.5900	240.1000	255.5650	12235217	NFLX
1253	2018-12-28	257.9400	261.9144	249.8000	256.0800	10987286	NFLX
1254	2018-12-31	260.1600	270.1001	260.0000	267.6600	13508920	NFLX

1255 rows × 7 columns

Next steps: View recommended plots New interactive sheet

Excercise 2

```
# Convert 'date' to datetime and 'volume' to integer
faang['date'] = faang['date'].apply(pd.to_datetime)
faang['volume'] = faang['volume'].apply(pd.to_numeric)

# Sort by date and ticker
faang_sorted = faang.sort_values(by=['date', 'ticker'])

# Get the 7 rows with the highest volume
top_volume = faang_sorted.nlargest(7, 'volume')

# Melt the data into long format (date and ticker are ID variables)
faang_long = pd.melt(
    faang_sorted,
    id_vars=['date', 'ticker'],
    value_vars=['open', 'high', 'low', 'close', 'volume'],
    var_name='attribute',
    value_name='value'
)

# Print results for verification
top_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

Next steps: View recommended plots New interactive sheet

```
faang_long.head()
```

	date	ticker	attribute	value
0	2018-01-02	AAPL	open	166.9271
1	2018-01-02	AMZN	open	1172.0000
2	2018-01-02	FB	open	177.6800
3	2018-01-02	GOOG	open	1048.3400
4	2018-01-02	NFLX	open	196.1000

Next steps: View recommended plots New interactive sheet

Excercise 3

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# URL of the NHFR page listing hospitals
url = 'https://nhfr.doh.gov.ph/rfacilities2list.php'

# Send a GET request to the URL
response = requests.get(url)
response.raise_for_status()  # Raise an error for bad status codes

# Parse the HTML content
soup = BeautifulSoup(response.content, 'html.parser')

# Find the table containing hospital data
table = soup.find('table', {'id': 'example'})

# Extract table headers
headers = [header.text.strip() for header in table.find_all('th')]

# Extract table rows
rows = []
for row in table.find_all('tr')[1:]:
    cells = row.find_all('td')
    row_data = [cell.text.strip() for cell in cells]
    rows.append(row_data)

# Create a DataFrame
df = pd.DataFrame(rows, columns=headers)

# Save to CSV
df.to_csv('hospitals.csv', index=False)

print("Data has been successfully scraped and saved to 'hospitals.csv'.")



# Save to CSV
df = pd.read_csv('v_activefacilities.csv')
df.to_csv('hospitals.csv', index=False)


# convert to dataframe
newdf = pd.read_csv('hospitals.csv')
newdf
```

	Health Facility Code	Health Facility Code Short	Facility Name	Old Health Facility Name 1	Old Health Facility Name 2	Old Health Facility Name 3	Facility Major Type	Health Facility Type	Owi Classif
0	DOH000000000000467	467	A. DE LA CRUZ MATERNITY HOSPITAL	DE LA CRUZ MATERNITY HOSPITAL	NaN	NaN	Health Facility	Hospital	
1	DOH000000000005026	5026	A. ZARATE GENERAL HOSPITAL	NaN	NaN	NaN	Health Facility	Hospital	
2	DOH000000000006720	6720	A.M. YUMENA GENERAL HOSPITAL INC.	YUMENA SURGICAL AND MEDICAL CLINIC	NaN	NaN	Health Facility	Hospital	
3	DOH000000000003315	3315	ABELLA MIDWAY HOSPITAL	NaN	NaN	NaN	Health Facility	Hospital	
4	DOH000000000003409	3409	ABORLAN MEDICARE HOSPITAL	NaN	NaN	NaN	Health Facility	Hospital	Gov
...	...	...	...	...	...	...	...	...	
1334	DOH000000000007036	7036	ZAMBOANGA DOCTORS HOSPITAL, INC.	NaN	NaN	NaN	Health Facility	Hospital	
1335	DOH000000000004975	4975	ZAMBOANGA PENINSULA MEDICAL CENTER, INC.	ZAMBOANGA CHILDREN'S HOSPITAL, INC.	NaN	NaN	Health Facility	Hospital	
1336	DOH000000000004592	4592	ZAMBOANGA PUERICULTURE LYING-IN MATERNITY HOS...	NaN	NaN	NaN	Health Facility	Hospital	
1337	DOH000000000002910	2910	ZAMBOANGA SIBUGAY PROVINCIAL HOSPITAL	NaN	NaN	NaN	Health Facility	Hospital	Gov
1338	DOH000000000034464	34464	ZONE MEDICAL AND INTERVENTION HOSPITAL, INC.	NaN	NaN	NaN	Health Facility	Hospital	

1339 rows × 32 columns

Conclusion

I found it exciting wrangling data using pandas, especially now that I can actually understand what I am doing. This leads me closer to being better at coding and data science. In this activity, we practiced working with real-world datasets by reading, combining, cleaning, and transforming data using Python and pandas. We handled multiple CSV files, added relevant identifiers, and organized the data into a more useful format for analysis. Through these tasks, we strengthened our understanding of data