MODULE 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python

Submitted by: Christan Ray R. Sanchez

Performed on: 04/07/25

Submitted on: 04/07/25

Submitted to: Engr.Roman M. 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 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 in a single dataframe.
4. Save the result in a CSV file called faang.csv

In [24]:
```
import pandas as pd

#1 Reading the files in...
FB = pd.read_csv('fb.csv')
AAPL = pd.read_csv('aapl.csv')
AMZN = pd.read_csv('amzn.csv')
NFLX= pd.read_csv('nflx.csv')
GOOG = pd.read_csv('goog.csv')

#2 creating a new column for each datafram called ticker
FB['ticker'] = 'FB'
AAPL['ticker'] = 'AAPL'
AMZN['ticker'] = 'AMZN'
NFLX['ticker'] = 'NFLX'
GOOG['ticker'] = 'GOOG'
```

```
# 3. Append files into a single DataFrame

faang = pd.concat([FB, AAPL, AMZN, NFLX, GOOG], ignore_index=True)  # Combining all

#4. Saving the result in a CSV file called faang.csv

faang.to_csv('faang.csv', index=False)  #Writing the final DataFrame to a CSV file

faang.head(4)
```

Out[24]:

	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
3	2018-01-05	185.59	186.90	184.9300	186.85	13574535	FB

Exercise 2:

- With faang, use type conversion to change the date column in 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 have separate columns for open, high, low, close, and volume.

In [56]:

```
#using type conversion to make the date column in date time
faang['date'] = pd.to_datetime(faang['date'])
faang['volume'] = pd.to_numeric(faang['volume'])

#finding the seven rows
top_7_volume = faang.nlargest(7, 'volume')

top_7_volume
```

Out[56]:

	date	open	high	low	close	volume	ticker
142	2018-07-26	174.8900	180.1300	173.7500	176.2600	169803668	FB
53	2018-03-20	167.4700	170.2000	161.9500	168.1500	129851768	FB
57	2018-03-26	160.8200	161.1000	149.0200	160.0600	126116634	FB
54	2018-03-21	164.8000	173.4000	163.3000	169.3900	106598834	FB
433	2018-09-21	219.0727	219.6482	215.6097	215.9768	96246748	AAPL
496	2018-12-21	156.1901	157.4845	148.9909	150.0862	95744384	AAPL
463	2018-11-02	207.9295	211.9978	203.8414	205.8755	91328654	AAPL

In [57]:
```
# Melting the DataFrame to convert it to long format
long_faang = faang.melt(id_vars=['date', 'ticker'], value_vars=['open', 'high', 'lo
long_faang
```

Out[57]:

	date	ticker	variable	value
0	2018-01-02	FB	open	177.68
1	2018-01-03	FB	open	181.88
2	2018-01-04	FB	open	184.90
3	2018-01-05	FB	open	185.59
4	2018-01-08	FB	open	187.20
...	...	...	...	...
6270	2018-12-24	GOOG	volume	1590328.00
6271	2018-12-26	GOOG	volume	2373270.00
6272	2018-12-27	GOOG	volume	2109777.00
6273	2018-12-28	GOOG	volume	1413772.00
6274	2018-12-31	GOOG	volume	1493722.00

6275 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.

In [39]:
```
import requests

# New humdata JSON metadata URL
```

```
url = 'https://data.nsw.gov.au/data/datastore/dump/e17840df-ecfc-4e38-b51b-9f49af5d

# Request
response = requests.get(url)

# Check if it is successful
if response.ok:
    # if the response is CSV, you might convert the data as CSV
    with open('hospitals.csv', 'wb') as file:
        file.write(response.content)
    print("File downloaded successfully.")
else:
    print(f'Request was not successful and returned code: {response.status_code}.')
```

```
File downloaded successfully.
```

In [58]:

```
# Fill missing values in each column with appropriate default values
hospitals.fillna({
    'Name': 'Unknown',
    'Address': 'Not Available',
    'Suburb': 'Not Available',
    'Postcode': 'Not Available',
    'Phone': 'N/A',
    'Email Address': 'Not Available',
    'Fax': 'N/A',
    'LHD': 'Not Available',
    'Hospital Website': 'Not Available',
    'ED': 'Not Available'
}, inplace=True)

hospitals.head(5)
```

Out[58]:

	Name	Address	Suburb	Postcode	Phone	Email Address	Fax	LHD	Hospital Website
0	Albury Wodonga Health	201 Borella Road	Albury	2640	02 6058 4444	Not Available	N/A	Albury Wodonga Health	Not Available
1	Armidale Rural Referral Hospital	Rusden Street	Armidale	2350	02 6776 9500	Not Available	02 6776 4774	Hunter New England Local Health District	Not Available
2	Auburn Hospital & Community Health Services	Hargrave Road	Auburn	2144	02 8759 3000	Not Available	02 9563 9666	Western Sydney Local Health District	Not Available
3	Ballina District Hospital	Cherry Street	Ballina	2478	02 6686 2111	Not Available	02 6686 6731	Northern NSW Local Health District	Not Available
4	Balmain Hospital	29 Booth Street	Balmain	2041	02 9395 2111	Not Available	02 9395 2020	Sydney Local Health District	Not Available

7.2 Conclusion:

I worked with multiple datasets and cleaned them using pandas. First, I combined stock data for Facebook, Apple, Amazon, Netflix, and Google, added a column for the ticker symbol, and saved it as a CSV. I then formatted the date and volume columns, sorted the data, and used melt() to convert it into a long format for easier analysis.

For the hospital data, I used web scraping to collect names, addresses, and contact details, then saved it in a CSV file. After loading the data into a pandas dataframe, I handled missing values by replacing NaNs with appropriate terms like "Not Available" or "N/A."

This activity was quite fun, and I learned a lot about data manipulation and how important it is to ensure data quality. I hope to improve more in this area and continue developing my skills.

In [ ]: