Module 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python

Name: Bautista, Jhon Hendricks
Section: CPE22S3

Performed on: 04/05/2025
Submitted on: 04/06/2025
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 the dataframe of the FAANG data as faang for the rest of the exercise:

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

In [27]:
```
# 1. read each file in

import pandas as pd
# Read each CSV file
facebook_df = pd.read_csv('fb.csv')
apple_df = pd.read_csv('aapl.csv')
amazon_df = pd.read_csv('amzn.csv')
netflix_df = pd.read_csv('nflx.csv')
google_df = pd.read_csv('goog.csv')
```

In [29]:
```
# 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 b

facebook_df['ticker'] = 'fb'
```

```
apple_df['ticker'] = 'aapl'
amazon_df['ticker'] = 'amzn'
netflix_df['ticker'] = 'nflx'
google_df['ticker'] = 'goog'

facebook_df.head()
apple_df.head()
amazon_df.head()
netflix_df.head()
google_df.head()
```

Out[29]:

	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
3	2018-01-05	1094.00	1104.25	1092.00	1102.23	1279123	goog
4	2018-01-08	1102.23	1111.27	1101.62	1106.94	1047603	goog

In [33]:
```
# 3. Append them together into a single dataframe.

# merge the dataframes on the common column
# I used outer since we need all the input from both dataframes
faang = pd.merge(facebook_df, apple_df, how = 'outer')
faang = pd.merge(faang, amazon_df, how = 'outer')
faang = pd.merge(faang, netflix_df, how = 'outer')
faang = pd.merge(faang, google_df, how = 'outer')
```

In [35]:
```
# 4. Save the result in a CSV file called faang.csv

faang.to_csv('faang.csv', index = False)
```

Exercise 2

- With faang, use type conversion to change the date column into a datetime and volume column into integers. Then, sort by date and ticker.
- Find the sevens rows with the highest values for volume.
- Right now, the data is somewhere between long and wide format. Use melt() to make it comepletely long format. Hint: date and ticker are our ID variable (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.

In [83]:
```
import pandas as pd

faang = pd.read_csv('faang.csv') # loading data
faang.head()
```

Out[83]:

	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	aapl
1	2018-01-02	177.6800	181.5800	177.5500	181.4200	18151903	fb
2	2018-01-02	196.1000	201.6500	195.4200	201.0700	10966889	nflx
3	2018-01-02	1048.3400	1066.9400	1045.2300	1065.0000	1237564	goog
4	2018-01-02	1172.0000	1190.0000	1170.5100	1189.0100	2694494	amzn

In [68]:
```
# 1. With faang, use type conversion to change the date column into a datetime and
# Then, sort by date and ticker.

faang['date'] = pd.to_datetime(faang['date']) # using the pd to datetime method for
faang['volume'] = faang['volume'].astype('int64') # using the astype method for cha
```

In [72]:
```
# 2. Find the sevens rows with the highest values for volume.

faang.nlargest(7,'volume') # use nlargest method, pass 7 to get 7 highest values an
```

Out[72]:

	date	open	high	low	close	volume	ticker
710	2018-07-26	174.8900	180.1300	173.7500	176.2600	169803668	fb
265	2018-03-20	167.4700	170.2000	161.9500	168.1500	129851768	fb
285	2018-03-26	160.8200	161.1000	149.0200	160.0600	126116634	fb
270	2018-03-21	164.8000	173.4000	163.3000	169.3900	106598834	fb
911	2018-09-21	219.0727	219.6482	215.6097	215.9768	96246748	aapl
1226	2018-12-21	156.1901	157.4845	148.9909	150.0862	95744384	aapl
1061	2018-11-02	207.9295	211.9978	203.8414	205.8755	91328654	aapl

In [85]:
```
# 3. Right now, the data is somewhere between long and wide format.
# Use metl() to make it comepletely long format. Hint: data and ticker are our ID v
# We need to melt the rest so that we don't have separate columns for open, high, l

faang = faang.melt(id_vars = ['date', 'ticker'], # sets what identifier to use
                   var_name = 'variable', # sets the new column name for the melted
                   value_name = 'value') # sets new column name for melted values
faang.head() # checking
```

Out[85]:

	date	ticker	variable	value
0	2018-01-02	aapl	open	166.9271
1	2018-01-02	fb	open	177.6800
2	2018-01-02	nflx	open	196.1000
3	2018-01-02	goog	open	1048.3400
4	2018-01-02	amzn	open	1172.0000

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 [72]:

```
# Using web scraping, search for the list of the hospitals, their address and conta
# Save the list in a new csv file, hospitals.csv.

'''In this cell I will first test/try to get all the needed elements from the websi

from bs4 import BeautifulSoup # import the needed modules
import requests


source = requests.get('https://top.org.ph/healthcare-equipment/hospitals/') # reque

soup = BeautifulSoup(source.text, 'lxml') # use the module and a parser

contentWrap = soup.find('div', class_= 'card-wrapper') # this pertains to the whole

hospitalName = contentWrap.find('div', class_= 'com-name-cc').h4.text # extracting

print(hospitalName) # checking

print()


hospitalAdd = contentWrap.find('div', class_= 'row') # using the container we find
address = hospitalAdd.find('p', class_= 'contact').text # extracting the address

print(address) # checking

print()

hospitalContact = contentWrap.find_all('div', class_= 'col-xl-6 col-lg-12')[2] # fi
contact_link = hospitalContact.find('a', href=True) # extracting contact number
if contact_link: # this is a block of code for getting only the list of contact num
    phone_numbers = contact_link.text.strip().split('|') # spliting using |
```

```
    phone_numbers = [num.strip() for num in phone_numbers]  # stroring in list
    print(phone_numbers) # checking
```

```
The Perpetual Help Medical Center - Las Pinas (PHMC-LP)

Alabang-Zapote Rd., Pamplona III  Las Pinas City

['(02) 8874 8515', '(02) 8880 7700', '(02) 8874 2582']
```

In [76]:
```
# Using web scraping, search for the list of the hospitals, their address and conta
# Save the list in a new csv file, hospitals.csv.

'''Now I will modify the previous
cell so that I can find all the data and store it in a csv'''

from bs4 import BeautifulSoup # import the needed modules
import requests
import csv

 # request from the website
source = requests.get('https://top.org.ph/healthcare-equipment/hospitals/')

# use the module and a parser
soup = BeautifulSoup(source.text, 'lxml')

# this pertains to the whole div containing the needed data
contentWrap = soup.find_all('div', class_= 'card-wrapper')

with open('hospitals.csv', 'w', newline='', encoding='utf-8') as csvfile:
    writer = csv.writer(csvfile) # creating writer

    writer.writerow(['Hospital Name', 'Address', 'Phone Numbers']) # create the hea

    for hospital in contentWrap:
        hospitalName = hospital.find('div', class_= 'com-name-cc').h4.text # extrac
        print(hospitalName) # checking

        hospitalAdd = hospital.find('div', class_= 'row') # using the container we
        address = hospitalAdd.find('p', class_= 'contact').text # extracting the ad
        print(address) # checking

        hospitalContact = hospital.find_all('div', class_= 'col-xl-6 col-lg-12')[2]
        contact = hospitalContact.find('a', href=True) # extracting contact number
        if contact: # this is a block of code for getting only the list of contact
            phone_numbers = contact.text.strip().split('|') # spliting using |
            phone_numbers = [num.strip() for num in phone_numbers]  # stroring in l
            print(phone_numbers) # checking

        print()
        # used to write every data scrapped per iteration of the loop
        writer.writerow([hospitalName, address, ', '.join(phone_numbers)])
```

```
The Perpetual Help Medical Center - Las Pinas (PHMC-LP)
Alabang-Zapote Rd., Pamplona III  Las Pinas City
['(02) 8874 8515', '(02) 8880 7700', '(02) 8874 2582']

St. Luke's Medical Center
279 E Rodriguez Sr. Ave. Quezon City
['(02) 8789 7700']

Asian Hospital and Medical Center
2205 Civic Dr, Alabang, Muntinlupa
['(02) 8771 9000']

Makati Medical Center
No. 2 Amorsolo Street, Legaspi Village, Makati City
['(02) 8888 8999']

Cardinal Santos Medical Center
10 Wilson, Greenhills West, San Juan, Metro Manila
['(02) 8727 0001']

Manila Doctors Hospital
667 United Nations Ave, Ermita, Manila
['(02) 8558 0888', '(02) 8558 0797', '(02) 8558 0798']

Westlake Medical Center
Pacita Complex, National Highway, San Pedro, Laguna
['(02) 553 8185']

ManilaMed - Medical Center Manila
850 UN. Ave, Paco Manila
['(02) 8522 3899']

Fatima University Medical Center
20 MacArthur Highway, Valenzuela City
['(02) 8291 6538']

TEBOW CURE Children's Hospital
Corner Banawe Street, J.P. Laurel Avenue, Brgy. W. Aquino, Davao City, 8000, Davao d
el Sur
['(08) 2224 6048']
```

In [49]:
```python
# Using the generated hospitals.csv, convert the csv file into pandas dataframe.
# Prepare the data using the necessary preprocessing techniques.

import pandas as pd

df = pd.read_csv('hospitals.csv') # load data
```

In [51]:
```python
# Using the generated hospitals.csv, convert the csv file into pandas dataframe.
# Prepare the data using the necessary preprocessing techniques.

df.isnull().sum() # checking for null values
```

Out[51]:
```
Hospital Name    0
Address          0
Phone Numbers    0
dtype: int64
```

In [53]:
```
# Using the generated hospitals.csv, convert the csv file into pandas dataframe.
# Prepare the data using the necessary preprocessing techniques.

df = df.sort_values(by = 'Hospital Name', ascending = True) # sorting the data
df.head()
```

Out[53]:

	Hospital Name	Address	Phone Numbers
2	Asian Hospital and Medical Center	2205 Civic Dr, Alabang, Muntinlupa	(02) 8771 9000
4	Cardinal Santos Medical Center	10 Wilson, Greenhills West, San Juan, Metro Ma...	(02) 8727 0001
8	Fatima University Medical Center	20 MacArthur Highway, Valenzuela City	(02) 8291 6538
3	Makati Medical Center	No. 2 Amorsolo Street, Legaspi Village, Makati...	(02) 8888 8999
5	Manila Doctors Hospital	667 United Nations Ave, Ermita, Manila	(02) 8558 0888, (02) 8558 0797, (02) 8558 0798

In [41]:
```
# Using the generated hospitals.csv, convert the csv file into pandas dataframe.
# Prepare the data using the necessary preprocessing techniques.

df.drop_duplicates(inplace=True) # removing duplicate entries
```

In [43]:
```
# Using the generated hospitals.csv, convert the csv file into pandas dataframe.
# Prepare the data using the necessary preprocessing techniques.

df.dtypes # check if there is need of change in datatype for ease of analysis
# no need to change data types since they are mostly strings
```

Out[43]:
```
Hospital Name    object
Address          object
Phone Numbers    object
dtype: object
```

7.2 Conclusion:

In this activity I was able to learn about the ways of data collection and merging of data. I learned to utilize unique identifiers for dataset so that when they merge we can still idenfity which data came from which. In the first 2 exercises, I felt that it was more on handling the merge of datasets and at first it was hard because I was not familiar with the melt() method. After acomplishing the exercise, I now know the use case of melt() which is to organize and

tidy the data. For the last exercise, it is very challenging because it is my first time doing webscrapping and I had a hard time reading html tags to find the correct field of div I can extract the data. I need to practice more webscrapping so that I can work faster and easily find ways to scrape data in websites.