Module 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python

Name: Bulambao, Adrian Justin

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 exercises:

In [ ]:

```
```

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 [1]:

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

```
In [2]: facebook.head(1)
```

Out[2]:

	date	open	high	low	close	volume
0	2018-01-02	177.68	181.58	177.55	181.42	18151903

```
In [3]: apple.head(1)
```

Out[3]:

	date	open	high	low	close	volume
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934

```
In [4]: amazon.head(1)
```

Out[4]:

	date	open	high	low	close	volume
0	2018-01-02	1172.0	1190.0	1170.51	1189.01	2694494

```
In [5]: netflix.head(1)
```

Out[5]:

	date	open	high	low	close	volume
0	2018-01-02	196.1	201.65	195.42	201.07	10966889

```
In [6]: google.head(1)
```

Out[6]:

	date	open	high	low	close	volume
0	2018-01-02	1048.34	1066.94	1045.23	1065.0	1237564

```
In [7]: # 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

        facebook['ticker'] = 'FB'
        apple['ticker'] = 'AAPL'
        amazon['ticker'] = 'AMZN'
        netflix['ticker'] = 'NFLX'
        google['ticker'] = 'GOOG'
```

```
In [8]: # 3. Append them together into a single dataframe.
        faang = pd.merge(facebook, apple, how = 'outer')
        faang = pd.merge(faang, amazon, how = 'outer')
        faang = pd.merge(faang, netflix, how = 'outer')
        faang = pd.merge(faang, google, how = 'outer')

        faang.head(1)
```

Out[8]:

	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	AAPL

In [9]:
```
#4.     Save the result in a CSV file called faang.csv.
faang.to_csv("faang.csv")
```

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.

In [10]:
```
# 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['date'] = pd.to_datetime(faang['date']) #pd.to_datetime is a command to set t
faang.volume = faang.volume.astype('int64') #astype is also a command to change a d

faang.dtypes
```

Out[10]:
```
date      datetime64[ns]
open             float64
high             float64
low              float64
close            float64
volume             int64
ticker            object
dtype: object
```

In [11]:
```
#Find the seven rows with the highest value for volume.
faang.nlargest(7,'volume')
#nlargest is used to find the largest n rows
# it takes two entries (number of rows, the column)
```

Out[11]:

	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 [12]:
```
#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 = faang.melt(id_vars = ['date', 'ticker'], var_name = 'variable', value_name
#this melts the dataframe so that it won't be a wide and long format
faang.head()
```

Out[12]:

	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 [13]:
```
#this modules are needed for this exercise
from bs4 import BeautifulSoup # this modules is used for web scraping and extract d
import requests # used to request from a url
import pandas as pd
```

In [14]:
```
url = 'https://shop.philcare.com.ph/accredited-hospitals' # the url is set to a var
```

In [15]:
```
page = requests.get(url) #used the get() command from requests to see if it will ge
#then it is placed on a variable
soup = BeautifulSoup(page.text, 'html')
#used BeautifulSoup to take the page, and 'html' to see it in html form
```

In [16]:
```
#soup.find('table')
#this reads the whole html and outputs the "table" class
#it is commented to prevent a long page of the html code
```

In [17]:
```
table = soup.find('table') #the parts of the html with the table class is stored
```

In [18]:
```
titles = table.find_all('th')#the 'th' class in the table is stored in another vari
```

In [19]:
```
titles
#the result below is the th class in the table class
#these are the column names
```

Out[19]:
```
[<th>Provider Name</th>,
 <th>Complete Address</th>,
 <th>City</th>,
 <th>Province</th>,
 <th>Region</th>,
 <th>Area</th>,
 <th>Contact No.</th>]
```

In [20]:
```
table_titles = [title.text for title in titles]
print(table_titles)
#the th class placed into a list
```

```
['Provider Name', 'Complete Address', 'City', 'Province', 'Region', 'Area', 'Contact
No.']
```

In [21]:
```
df = pd.DataFrame(columns = table_titles)

df
#the th class placed on a list is converted to dataframe columns
```

Out[21]:

Provider Name	Complete Address	City	Province	Region	Area	Contact No.

In [22]:
```
column_data = table.find_all('tr')
#from the table class the tr class is taken
#in the website, the tr class are the rows
```

In [23]:
```
for row in column_data[1:]: #[1:] is used to take the second one since the first on
    row_data = row.find_all('td')# all the tr class contain a td class which is the
    row_data = [data.text for data in row_data]

    length = len(df)
    df.loc[length] = row_data
```

In [24]:
```
df.head() #output of the dataframe
```

Out[24]:

	Provider Name	Complete Address	City	Province	Region	Area	C
0	CLINICA LAGUNA MULTISPECIALTY CENTER AND DIAGN...	UNIT 207 PARIAN COMMERCE CENTER PARIAN CALAMBA...	CALAMBA CITY	LAGUNA	Region IV-A (CALABARZON)	SOUTH LUZON	
1	ABELLA MIDWAY HOSPITAL	125 P. VALERO ST. BRGY. POBLACION VALENCIA CIT...	VALENCIA CITY	BUKIDNON	Region X	MINDANAO	(088
2	ABESAMIS EYE CARE AND CONTACT LENS CENTER (MAK...	SUITE 904 MEDICAL PLAZA MAKATI, DELA ROSA CORN...	MAKATI CITY	METRO MANILA	NCR	METRO MANILA	(02)
3	ACCURATE MEDICAL DIAGNOSTICS (MABALACAT BRANCH)	LOT 15 BLOCK 10 MC ARTHUR HI-WAY, MABIGA BRGY....	MABALACAT	PAMPANGA	Region III	NORTH LUZON	(045 8706 893
4	ACCURATE MEDICAL DIAGNOSTICS (ANGELES CITY BRA...	2442 STO. ENTIERRO ST. BRGY. STO. CRISTO ANGEL...	ANGELES CITY	PAMPANGA	Region III	NORTH LUZON	(045

In [25]:
```python
df.to_csv('Hospitals.csv') #code to convert the dataframe into a csv file
```

7.2 Conclusion:

The first and second exercise was easy, since there are instructions to follow, it was instructions to be followed, reading the csv, merging many csv/s into one, changing datatypes, making new columns, and finding specific rows, in the third one it was very hard, because I need to find a website that containthe data I need, I was lucky enough to find one that is in table form, I have no knowledge on what to do and how to properly take the data from an html, and how to list what I need, there was no instruction on gathering data, and some websites are not available for data scraping, I want to learn more about this.

```
In [ ]:
```