Module 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python
Submitted by: Silang, Elijah Yohance O.
Performed on: 04/07/2025
Submitted on: 04/DD/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:

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.

```
import pandas as pd

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

```
fb = fb.assign(ticker = lambda x: 'FB')
ap = ap.assign(ticker = lambda x: 'AAPL')
am = am.assign(ticker = lambda x: 'AMZN')
ne = ne.assign(ticker = lambda x: 'NFLX')
go = go.assign(ticker = lambda x: 'GOOG')
```

```
faang = pd.concat([fb, ap, am, ne, go], ignore_index=True)
faang.to_csv('faang.csv')
faang.head()
```

	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
4	2018-01-08	187.20	188.90	186.3300	188.28	17994726	FB

Next steps: View recommended plots | New interactive sheet

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.

```
# checking the type first because it is a good practice as I observe from my mentor.
print(type(faang.date[0]))
print(type(faang.volume[0]))
```

```
<class 'str'>
<class 'numpy.int64'>
```

```
from datetime import datetime

for date in faang.date:
  new_date = datetime.strptime(date, '%Y-%m-%d')

for integer in faang.volume:
  new_volume = int(integer)


#checking
print(type(new_date))
print(type(new_volume))
```

```
<class 'datetime.datetime'>
<class 'int'>
```

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.

```
import pandas as pd

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

```
---------------------------------------------------------------------------
HTTPError                                 Traceback (most recent call last)
<ipython-input-49-25a57fbbaf16> in <cell line: 0>()
      2
      3 url = 'https://sulit.ph/list-of-hospitals-in-metro-manila-with-contact-details-website-and-social-media-accounts/'
----> 4 hospital = pd.read_html(url)

                                  13 frames
/usr/lib/python3.11/urllib/request.py in http_error_default(self, req, fp, code, msg, hdrs)
    641 class HTTPDefaultErrorHandler(BaseHandler):
    642     def http_error_default(self, req, fp, code, msg, hdrs):
--> 643         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    644
    645 class HTTPRedirectHandler(BaseHandler):

HTTPError: HTTP Error 403: Forbidden
```

7.2 Conclusion:

- Write your conclusion here.

This is the first time I concatenate dataframes and before this activity, I have always wondered how collected datas were combined, so now I knew.