Module 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python

Submitted by: Dela Cruz, Eugene

Performed on 03/20/2024

Submitted on 03/20/2024

Submitted to: Engr. Roman M. Richard

7.1 Supplementary Activity

using the data sets 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

apple = pd.read_csv('/content/aapl.csv'); amazon = pd.read_csv('/content/amzn.csv')
fb = pd.read_csv('/content/fb.csv'); google = pd.read_csv('/content/goog.csv')
netflix = pd.read_csv('/content/nflx.csv')

apple = apple.assign(ticker='APPL')
amazon = amazon.assign(ticker='AMZN')
fb = fb.assign(ticker='FB')
google = google.assign(ticker='GGLE')
netflix = netflix.assign(ticker='NTFX')

faang = pd.concat([apple, amazon, fb, google, netflix])
faang
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	APPL
1	2018-01-03	169.2521	171.2337	168.6929	168.9578	29517899	APPL
2	2018-01-04	169.2619	170.1742	168.8106	169.7426	22434597	APPL
3	2018-01-05	170.1448	172.0381	169.7622	171.6751	23660018	APPL
4	2018-01-08	171.0375	172.2736	170.6255	171.0375	20567766	APPL
...	...	...	...	...	...	...	...
246	2018-12-24	242.0000	250.6500	233.6800	233.8800	9547616	NTFX
247	2018-12-26	233.9200	254.5000	231.2300	253.6700	14402735	NTFX
248	2018-12-27	250.1100	255.5900	240.1000	255.5650	12235217	NTFX
249	2018-12-28	257.9400	261.9144	249.8000	256.0800	10987286	NTFX
250	2018-12-31	260.1600	270.1001	260.0000	267.6600	13508920	NTFX

1255 rows × 7 columns

Next steps: View recommended plots

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

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 vollume.
- 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

```
#conversion of date and volume
faang.rename(columns = {'date':'datetime'}, inplace = True)
faang.rename(columns = {'volume':'integers'}, inplace = True)
faang.sort_values(['datetime', 'ticker'])
```

	datetime	open	high	low	close	integers	ticker
0	2018-01-02	1172.0000	1190.0000	1170.5100	1189.0100	2694494	AMZN
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	APPL
0	2018-01-02	177.6800	181.5800	177.5500	181.4200	18151903	FB
0	2018-01-02	1048.3400	1066.9400	1045.2300	1065.0000	1237564	GGLE
0	2018-01-02	196.1000	201.6500	195.4200	201.0700	10966889	NTFX
...	...	...	...	...	...	...	...
250	2018-12-31	1510.8000	1520.7600	1487.0000	1501.9700	6954507	AMZN
250	2018-12-31	157.8529	158.6794	155.8117	157.0663	35003466	APPL
250	2018-12-31	134.4500	134.6400	129.9500	131.0900	24625308	FB
250	2018-12-31	1050.9600	1052.7000	1023.5900	1035.6100	1493722	GGLE
250	2018-12-31	260.1600	270.1001	260.0000	267.6600	13508920	NTFX

1255 rows × 7 columns

```
faang.nlargest(7,'integers')
```

	datetime	open	high	low	close	integers	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
182	2018-09-21	219.0727	219.6482	215.6097	215.9768	96246748	APPL
245	2018-12-21	156.1901	157.4845	148.9909	150.0862	95744384	APPL
212	2018-11-02	207.9295	211.9978	203.8414	205.8755	91328654	APPL

```
faang.melt(id_vars = ['datetime','ticker'],
         value_vars = ['open','high','low','close', 'integers']
)
```

	datetime	ticker	variable	value
0	2018-01-02	APPL	open	1.669271e+02
1	2018-01-03	APPL	open	1.692521e+02
2	2018-01-04	APPL	open	1.692619e+02
3	2018-01-05	APPL	open	1.701448e+02
4	2018-01-08	APPL	open	1.710375e+02
...	...	...	...	...
6270	2018-12-24	NTFX	integers	9.547616e+06
6271	2018-12-26	NTFX	integers	1.440274e+07
6272	2018-12-27	NTFX	integers	1.223522e+07
6273	2018-12-28	NTFX	integers	1.098729e+07
6274	2018-12-31	NTFX	integers	1.350892e+07

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.

7.2 Conclusion:

I therefore conclude that I put together different files with data about Facebook, Apple, Amazon, Netflix, and Google (FAANG) stocks into one big file called "faang". Then, I changed some parts of the data to make it easier to work with. After that, I sorted the data and found the seven rows with the most volume. Lastly, I searched online for information about hospitals, like their names, addresses, and contact details. I saved this information in a file called "hospitals.csv", and then I organized it into a table using a program called pandas.