Module 7: Data Wrangling with Pandas

CPE311 Computational Thinking with Python

Submitted by: Marquez, Clard Jozrein M.

Performed on: 03-20-2024

Sumbitted on: 03-20-2024

Submitted to: Engr. Roman M. Richar

7.1 Supplementary Activity

Using the datasets provided, perform the following exercises:

Exercise 1

```
import pandas as pd

# Read in the csv files
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')


# create a list with corresponding file names
tickers = {
    'AAPL': '/content/aapl.csv',
    'AMZN': '/content/amzn.csv',
    'FB': '/content/fb.csv',
    'GOOG': '/content/goog.csv',
    'NFLX': '/content/nflx.csv'
}
```

```
# Create an empty dataframe to store the combined datas
faang = pd.DataFrame()

# for loop function through each ticker symbol and the file names
for ticker, file_name in tickers.items():

    df = pd.read_csv(file_name) # Read the CSV file in the dataframe

    df['ticker'] = ticker # this will add the new column 'ticker' to each data

    faang = pd.concat([faang, df], ignore_index=True) # appending the dataframes to the empt

faang.to_csv('/content/faang.csv', index=False) # created a csv file of the combined datafra


df = pd.read_csv('/content/faang.csv') # read the created combined data
df
```

	date	open	high	low	close	volume	ticker
0	2018-01-02	166.9271	169.0264	166.0442	168.9872	25555934	AAPL
1	2018-01-03	169.2521	171.2337	168.6929	168.9578	29517899	AAPL
2	2018-01-04	169.2619	170.1742	168.8106	169.7426	22434597	AAPL
3	2018-01-05	170.1448	172.0381	169.7622	171.6751	23660018	AAPL
4	2018-01-08	171.0375	172.2736	170.6255	171.0375	20567766	AAPL
...	...	...	...	...	...	...	...
1250	2018-12-24	242.0000	250.6500	233.6800	233.8800	9547616	NFLX
1251	2018-12-26	233.9200	254.5000	231.2300	253.6700	14402735	NFLX
1252	2018-12-27	250.1100	255.5900	240.1000	255.5650	12235217	NFLX
1253	2018-12-28	257.9400	261.9144	249.8000	256.0800	10987286	NFLX
1254	2018-12-31	260.1600	270.1001	260.0000	267.6600	13508920	NFLX

1255 rows × 7 columns

Next steps: View recommended plots

Exercise 2

```
faang['date'] = pd.to_datetime(faang['date']) # uses a type conversion to change the column
faang['volume'] = faang['volume'].astype(int) # and the volume into integers.

faang = faang.sort_values(['date', 'ticker']) # sorting the values by date abd ticker

top_volume = faang.nlargest(7, 'volume') # finding the highest value for volume

# using the melt() function to make the data completely long format
faang_long = faang.melt(
    id_vars=['date','ticker'],
    value_vars=['open', 'high', 'low', 'close', 'volume'],
    var_name='variable',
    value_name='value'
)
```

```
top_volume
```

	date	open	high	low	close	volume	ticker
644	2018-07-26	174.8900	180.1300	173.7500	176.2600	169803668	FB
555	2018-03-20	167.4700	170.2000	161.9500	168.1500	129851768	FB
559	2018-03-26	160.8200	161.1000	149.0200	160.0600	126116634	FB
556	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	AAPL
245	2018-12-21	156.1901	157.4845	148.9909	150.0862	95744384	AAPL
212	2018-11-02	207.9295	211.9978	203.8414	205.8755	91328654	AAPL

Next steps: View recommended plots

```
faang_long
```

	date	ticker	variable	value
0	2018-01-02	AAPL	open	1.669271e+02
1	2018-01-02	AMZN	open	1.172000e+03
2	2018-01-02	FB	open	1.776800e+02
3	2018-01-02	GOOG	open	1.048340e+03
4	2018-01-02	NFLX	open	1.961000e+02

Next steps: View recommended plots

Conclusion

In this activity, I was able to perform different techniques in data processing and how to merge all csv files into one dataframe. This activity allows us to maximize the use of the tool pandas and makes data manipulation easy for us. I also learned the melt() function and how to use it, as it used to change a dataframe from wide to long. It is used to create a specific format of the DataFrame object where one or more columns work as identifiers. All the remaining columns are treated as values and unpivoted to the row axis and only two columns, variable and value.