James Cross - Assignment 03 Task 02

Refer to `cross-chap03-task02.ipynb`

Exercise 1

```
import pandas as pd

# Name of each file containing ticker information
files = ['aapl.csv', 'amzn.csv', 'fb.csv', 'goog.csv', 'nflx.csv']

faang = pd.DataFrame()

for file in files:
    # to create the ticker symbol, split filename at the '.' and uppercase it
    ticker = file.split('.')[0].upper()
    # path is relative to ch_03
    df = pd.read_csv(f'./exercises/{file}')
    # create ticker column in each csv/df
    df['ticker'] = ticker
    faang = faang.append(df)

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

Exercise 2

```
# change datatype of date to datetime
faang['date'] = pd.to_datetime(faang['date'])
# change datatype of volume to int
faang['volume'] = faang['volume'].astype(int)
# sort by date first, then by ticker
faang = faang.sort_values(by=['date', 'ticker'])
```

Running `faang.dtypes` before and after these changes shows the updates. `date` goes from 'object' to 'datatime64' and `volume` goes from 'float64' to 'int32'.

Exercise 3

```
faang.nsmallest(7, 'volume')
```

output:

```
    date        high        low         open        close       volume      ticker
126 2018-07-03  1135.819946 1100.020020 1135.819946 1102.890015 679000.0    GOOG
226 2018-11-23  1037.589966 1022.398987 1030.000000 1023.880005 691500.0    GOOG
99  2018-05-24  1080.469971 1066.150024 1079.000000 1079.239990 766800.0    GOOG
130 2018-07-10  1159.589966 1149.589966 1156.979980 1152.839966 798400.0    GOOG
152 2018-08-09  1255.541992 1246.010010 1249.900024 1249.099976 848600.0    GOOG
159 2018-08-20  1211.000000 1194.625977 1205.020020 1207.770020 870800.0    GOOG
161 2018-08-22  1211.839966 1199.000000 1200.000000 1207.329956 887400.0    GOOG
```

Exercise 4

```
faang_melted = faang.melt(id_vars=['date', 'ticker'],
                          value_vars=['open', 'high', 'low', 'close', 'volume'],
                          var_name='stock')
faang_melted.head(3)
```

```
    date        ticker  stock   value
0   2018-01-02  AAPL    open    42.540001
1   2018-01-02  AMZN    open    1172.000000
2   2018-01-02  FB      open    177.679993
```

```
# the other points in the 'melted' data can be view like this:
faang_melted[faang_melted['stock'] == 'high'].head(3)
```

```
        date        ticker  stock   value
1255    2018-01-02  AAPL    high    43.075001
1256    2018-01-02  AMZN    high    1190.000000
1257    2018-01-02  FB      high    181.580002
```

Exercise 5

We can try to confirm if there was a glitch by finding a different source of the data to see if it correlates with what we have. If we know that July 26th was the only day that the glitch occured on, we could possbily modify the data to be the average of the surrounding dates, so that it eliminates possible outliers.

```
faang[faang['date'] == '2018-07-25']
faang[faang['date'] == '2018-07-26']
faang[faang['date'] == '2018-07-27']
```

The data shows that Facebook stock price crashed on that day and stuck there for atleast the next day. Finding a matching reason for that crash (earnings report, etc) would be a helpful note to make.

Exercise 6

```
# read covid19_case.csv
covid = pd.read_csv('./exercises/covid19_cases.csv')
# create date column
covid['date'] = pd.to_datetime(covid['dateRep'])
# set index to date and sort by index
covid = covid.set_index('date')
covid = covid.sort_index()
# replace United States with USA and United Kingdom with UK
covid = covid.replace({"United_States_of_America": "USA", "United_Kingdom": "UK"})
# make a new df with list of countries (below)
country_list = ['Argentina', 'Brazil', 'China', 'Colombia', 'India', 'Italy', 'Mexico', 'Per
covid_short = covid[covid['countriesAndTerritories'].isin(country_list)]
```

```
In [54]: covid = covid.replace({"United_States_of_America": "USA", "United_Kingdom": "UK"})
country_list = ['Argentina', 'Brazil', 'China', 'Colombia', 'India', 'Italy', 'Mexico', 'Peru', 'Russia', 'Spain', 'Turkey', 'UK'
covid_short = covid[covid['countriesAndTerritories'].isin(country_list)]
covid_short.head(3)
```

Out[54]:

date	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterritoryCode	popData2019	continentExp	Cumulative_number_for_1 19_
2020-01-01	01/01/2020	1	1	2020	0	0	Russia	RU	RUS	145872260.0	Europe	
2020-01-01	01/01/2020	1	1	2020	0	0	UK	UK	GBR	66647112.0	Europe	
2020-01-01	01/01/2020	1	1	2020	0	0	Brazil	BR	BRA	211049519.0	America	

Figure 1: part 4 and 5

```
# pivot data around the list of countries
covid_pivot = covid_short.pivot(columns='countriesAndTerritories', values='cases').fillna(0)
```

```
In [55]: covid_pivot = covid_short.pivot(columns='countriesAndTerritories', values='cases').fillna(0)
covid_pivot.head(3)
```

Out[55]:

countriesAndTerritories	Argentina	Brazil	China	Colombia	India	Italy	Mexico	Peru	Russia	Spain	Turkey	UK	USA
date													
2020-01-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-01-02	0.0	0.0	2095.0	0.0	0.0	0.0	0.0	0.0	2.0	1.0	0.0	2.0	1.0
2020-01-03	0.0	1.0	574.0	0.0	0.0	240.0	2.0	0.0	0.0	28.0	0.0	12.0	3.0

Figure 2: part 6

Exercise 7

```
# the hint about passing in index_col='cases' was wrong (i think?). cases failed but 'index
total_cases = pd.read_csv('./exercises/covid19_total_cases.csv', index_col='index')
```

```
# "T" will transpose the data and then it gets sorted by cases
total_cases.T.sort_values(by='cases', ascending=False).head(20)
```