Pandas Mid Course Consolidation

1. Import libraries in the first cell

In [1]:
```
import pandas as pd
```

2. Load data from SPX file to df (url in notebook #3)

Make sure to use Date in the index

In [2]:
```
df = pd.read_csv(
    'https://s3.eu-west-1.amazonaws.com/neueda.conygre.com/pydata/SPX.csv'
    ,index_col='Date'
    ,parse_dates=True
)
df
```

Out[2]:

Date	Price	Open	High	Low	Change %
2017-12-29	2,673.61	2,689.15	2,692.12	2,673.61	-0.52%
2017-12-28	2,687.54	2,686.10	2,687.66	2,682.69	0.18%
2017-12-27	2,682.62	2,682.10	2,685.64	2,678.91	0.08%
2017-12-26	2,680.50	2,679.09	2,682.74	2,677.96	-0.11%
2017-12-22	2,683.34	2,684.22	2,685.35	2,678.13	-0.05%
...	...	...	...	...	...
2006-01-10	1,289.69	1,290.15	1,290.15	1,283.76	-0.04%
2006-01-09	1,290.15	1,285.45	1,290.78	1,284.82	0.37%
2006-01-06	1,285.45	1,273.48	1,286.09	1,273.48	0.94%
2006-01-05	1,273.48	1,273.46	1,276.91	1,270.30	0.00%
2006-01-04	1,273.46	1,268.80	1,275.37	1,267.74	0.37%

3017 rows × 5 columns

3. Convert all columns to numerical

- Clean the columns with `.str.replace()`
- Convert to numerical with `pd.to_numeric()`

In [3]:
```
df['Price'] = pd.to_numeric(df['Price'].str.replace(',',''))
df['Open'] = pd.to_numeric(df['Open'].str.replace(',',''))
df['High'] = pd.to_numeric(df['High'].str.replace(',',''))
df['Low'] = pd.to_numeric(df['Low'].str.replace(',',''))
df['Change %'] = pd.to_numeric(df['Change %'].str.replace('%',''))
df.dtypes
```

Out[3]:
```
Price       float64
Open        float64
High        float64
Low         float64
Change %    float64
dtype: object
```

4. Sort the dataframe in chronological order

In [4]:
```
df.sort_index(inplace=True)
df
```

Out[4]:

Date	Price	Open	High	Low	Change %
2006-01-04	1273.46	1268.80	1275.37	1267.74	0.37
2006-01-05	1273.48	1273.46	1276.91	1270.30	0.00
2006-01-06	1285.45	1273.48	1286.09	1273.48	0.94
2006-01-09	1290.15	1285.45	1290.78	1284.82	0.37
2006-01-10	1289.69	1290.15	1290.15	1283.76	-0.04
...	...	...	...	...	...
2017-12-22	2683.34	2684.22	2685.35	2678.13	-0.05
2017-12-26	2680.50	2679.09	2682.74	2677.96	-0.11
2017-12-27	2682.62	2682.10	2685.64	2678.91	0.08
2017-12-28	2687.54	2686.10	2687.66	2682.69	0.18
2017-12-29	2673.61	2689.15	2692.12	2673.61	-0.52

3017 rows × 5 columns

5. Plot a line chart: June 2015 to June 2016, High + Low

In [5]:
```
rows = "'2015-06-01' <= Date and Date < '2016-06-01'"
cols = ['Low','High']
df.query(rows)[cols]
```

Out[5]:

Date	Low	High
2015-06-01	2102.54	2119.15
2015-06-02	2099.14	2117.59
2015-06-03	2109.61	2121.92
2015-06-04	2093.23	2112.89
2015-06-05	2085.67	2100.99
...	...	...
2016-05-24	2052.65	2079.67
2016-05-25	2078.93	2094.73
2016-05-26	2087.08	2094.30
2016-05-27	2090.06	2099.06
2016-05-31	2088.66	2103.48

253 rows × 2 columns

6. Find the min, max, average, of all columns

In [6]:
```
funcs = ['min','mean','max']
df.agg(funcs)
```

Out[6]:

	Price	Open	High	Low	Change %
min	676.530000	679.280000	695.270000	666.790000	-9.030000
mean	1576.930597	1576.562337	1585.094839	1567.404445	0.032105
max	2690.160000	2692.710000	2694.970000	2685.920000	11.580000

In [ ]:
```

```