In [ ]:

```
Nurshanov Dias IT3-2208
6 lab
python
```

In [1]:

```
#1 Exercise

import pandas as pd
df_bronze = pd.read_csv('Summer Olympic medals/Bronze.csv')
df_silver = pd.read_csv('Summer Olympic medals/Silver.csv')
df_gold = pd.read_csv('Summer Olympic medals/Gold.csv')

df_gold.head(5)
```

Out[1]:

	NOC	Country	Total
0	USA	United States	2088.0
1	URS	Soviet Union	838.0
2	GBR	United Kingdom	498.0
3	FRA	France	378.0
4	GER	Germany	407.0

In [2]:

```
#2 exercise
weather_2 = pd.read_csv('Week_6/weather2.csv')
temps_f = weather_2[['Min TemperatureF', 'Mean TemperatureF', 'Max TemperatureF']]
print('temps_f\n',temps_f.head(5))

temps_c = (temps_f - 32) * 5 / 9
print('temps_c\n',temps_c.head(5))

temps_c.columns = temps_c.columns.str.replace('F','C')
print(temps_c.head(5))
```

```
temps_f
    Min TemperatureF  Mean TemperatureF  Max TemperatureF
0                 21                 28                32
1                 17                 21                25
2                 16                 24                32
3                 27                 28                30
4                 25                 30                34
temps_c
    Min TemperatureF  Mean TemperatureF  Max TemperatureF
0          -6.111111          -2.222222          0.000000
1          -8.333333          -6.111111         -3.888889
2          -8.888889          -4.444444          0.000000
3          -2.777778          -2.222222         -1.111111
4          -3.888889          -1.111111          1.111111
   Min TemperatureC  Mean TemperatureC  Max TemperatureC
0          -6.111111          -2.222222          0.000000
1          -8.333333          -6.111111         -3.888889
2          -8.888889          -4.444444          0.000000
3          -2.777778          -2.222222         -1.111111
4          -3.888889          -1.111111          1.111111
```

In [3]:

```
#Read the files 'sales-jan-2015.csv', 'sales-feb-2015.csv' and 'sales-mar-2015.csv' into the DataFrames jan, fe
#Use parse_dates=True and index_col='Date'.
#Extract the 'Units' column of jan, feb, and mar to create the Series jan_units, feb_units, and mar_units respe
#Construct the Series quarter1 by appending feb_units to jan_units and then appending mar_units to the result.
#Verify that quarter1 has the individual Series stacked vertically. To do this:
#Print the slice containing rows from jan 27, 2015 to feb 2, 2015.
#Print the slice containing rows from feb 26, 2015 to mar 7, 2015.
#Compute and print the total number of units sold from the Series quarter1.

feb = pd.read_csv('Sales/sales-feb-2015.csv', parse_dates=['Date'], index_col='Date')
jan = pd.read_csv('Sales/sales-jan-2015.csv', parse_dates=['Date'], index_col='Date')
mar = pd.read_csv('Sales/sales-mar-2015.csv', parse_dates=['Date'], index_col='Date')

jan_units = jan['Units']
feb_units = feb['Units']
mar_units = mar['Units']

quarter1= jan_units._append(feb_units)._append(mar_units)
quarter1.head(5)

quarter1.info()

quarter1 = quarter1.sort_index()

print(quarter1.loc['2015-01-27':'2015-02-02'].head(5))
print()
print(quarter1.loc['2015-02-26':'2015-03-07'].head(5))
print()
print(quarter1.sum())
```

```
<class 'pandas.core.series.Series'>
DatetimeIndex: 60 entries, 2015-01-21 19:13:21 to 2015-03-13 16:25:24
Series name: Units
Non-Null Count  Dtype
--------------  -----
60 non-null     int64
dtypes: int64(1)
memory usage: 960.0 bytes
Date
2015-01-27 07:11:55    18
2015-02-02 08:33:01     3
2015-02-02 20:54:49     9
Name: Units, dtype: int64

Date
2015-02-26 08:57:45     4
2015-02-26 08:58:51     1
2015-03-06 02:03:56    17
2015-03-06 10:11:45    17
Name: Units, dtype: int64

642
```

In [4]:

```
df_bronze = pd.read_csv('Summer Olympic medals/Bronze.csv')
df_silver = pd.read_csv('Summer Olympic medals/Silver.csv')
df_gold = pd.read_csv('Summer Olympic medals/Gold.csv')

medal_list = [df_bronze, df_silver, df_gold]

medals = pd.concat(medal_list, keys=['Bronze', 'Silver', 'Gold'], axis=1, join='inner')
medals
```

Out[4]:

	Bronze			Silver			Gold		
	NOC	Country	Total	NOC	Country	Total	NOC	Country	Total
0	USA	United States	1052.0	USA	United States	1195.0	USA	United States	2088.0
1	URS	Soviet Union	584.0	URS	Soviet Union	627.0	URS	Soviet Union	838.0
2	GBR	United Kingdom	505.0	GBR	United Kingdom	591.0	GBR	United Kingdom	498.0
3	FRA	France	475.0	FRA	France	461.0	FRA	France	378.0
4	GER	Germany	454.0	GER	Germany	350.0	GER	Germany	407.0
...	...	...	...	...	...	...	...	...	...
133	SEN	Senegal	NaN	SEN	Senegal	1.0	SEN	Senegal	NaN
134	SUD	Sudan	NaN	SUD	Sudan	1.0	SUD	Sudan	NaN
135	TGA	Tonga	NaN	TGA	Tonga	1.0	TGA	Tonga	NaN
136	BDI	Burundi	NaN	BDI	Burundi	NaN	BDI	Burundi	1.0
137	UAE	United Arab Emirates	NaN	UAE	United Arab Emirates	NaN	UAE	United Arab Emirates	1.0

138 rows × 9 columns

In [ ]: