Time-series Analysis Examples

Importing Libraries & Dataset(s)

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set()
```

In [2]:

```
brent = pd.read_csv('data/brent.csv')
```

In [ ]:

Inspecting the Dataset

In [3]:

```
brent.head()
```

Out[3]:

	Date	Price
0	04-Jan-00	23.95
1	05-Jan-00	23.72
2	06-Jan-00	23.55
3	07-Jan-00	23.35
4	10-Jan-00	22.77

In [4]:

```
brent.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5016 entries, 0 to 5015
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    5016 non-null   object 
 1   Price   5016 non-null   float64
dtypes: float64(1), object(1)
memory usage: 78.5+ KB
```

In [ ]:

Preprocessing the Dataset for the Time-series Analysis

In [5]:

```
brent.Date = pd.to_datetime(brent['Date'])
```

In [6]:

```
brent.set_index(['Date'], inplace=True)
```

In [7]:

```
brent.head()
```

Out[7]:

	Price
Date	
2000-01-04	23.95
2000-01-05	23.72
2000-01-06	23.55
2000-01-07	23.35
2000-01-10	22.77

In [8]:

```
brent.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5016 entries, 2000-01-04 to 2019-09-30
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Price   5016 non-null   float64
dtypes: float64(1)
memory usage: 78.4 KB
```

In [ ]:

```

```

Q: Create a Dataframe that contains only the dates from 1 January 2017 to 30 April 2018, then calculate the average price of this period

In [9]:

```
brent.loc['2017-01-01':'2018/04/30']
```

Out[9]:

	Price
Date	
2017-01-03	55.05
2017-01-04	54.57
2017-01-05	54.99
2017-01-06	55.90
2017-01-09	54.39
...	...
2018-04-24	75.86
2018-04-25	73.79
2018-04-26	75.39
2018-04-27	75.33
2018-04-30	75.92

339 rows × 1 columns

In [10]:

```
brent.loc['2017-01-01':'2018/04/30'].mean().round(2)
```

Out[10]:

```
Price    57.55
dtype: float64
```

In [ ]:

```

```

Q: Compare the average prices from March 2015 and March 2016, and plot the results

In [11]:

```
mar15 = brent.loc['2015-05-01':'2015-05-31']
mar15.head()
```

Out[11]:

	Price
Date	
2015-05-01	64.13
2015-05-04	64.62
2015-05-05	65.44
2015-05-06	66.22
2015-05-07	64.93

In [12]:

```
mar16 = brent.loc['2016-05-01':'2016-05-31']
mar16.head()
```

Out[12]:

	Price
Date	
2016-05-02	45.82
2016-05-03	43.09
2016-05-04	43.08
2016-05-05	44.39
2016-05-06	44.60

In [13]:

```
mar15.mean()
```

Out[13]:

```
Price    64.075
dtype: float64
```

In [14]:

```
mar16.mean()
```

Out[14]:

```
Price    46.742381
dtype: float64
```

In [ ]:

```

```

Q: Add a new column called 'Quarter' that contains each date's respective quarter of the year 2016, then calculate the average price for each quarter. Repeat this process without using the created new 'Quarter' column

In [15]:

```
brent.head()
```

Out[15]:

	Price
Date	
2000-01-04	23.95
2000-01-05	23.72
2000-01-06	23.55
2000-01-07	23.35
2000-01-10	22.77

In [16]:

```
brent.index.quarter
```

Out[16]:

```
Int64Index([1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
            ...
            3, 3, 3, 3, 3, 3, 3, 3, 3, 3],
           dtype='int64', name='Date', length=5016)
```

In [17]:

```
brent['Quarter'] = brent.index.quarter
```

In [18]:

```
brent.sample(10)
```

Out[18]:

Date	Price	Quarter
2011-07-28	118.16	3
2003-04-30	23.60	2
2007-02-01	56.74	1
2008-09-19	93.46	3
2008-10-07	83.17	4
2019-07-24	63.83	3
2007-09-06	76.21	3
2001-06-13	29.13	2
2008-11-19	48.35	4
2010-03-26	77.98	1

In [19]:

```
brent['2016'].groupby('Quarter').agg(average_price=('Price', 'mean'))
```

Out[19]:

Quarter	average_price
1	33.842742
2	45.566875
3	45.801061
4	49.052222

In [25]:

```
# Using 'resample()' function
brent.loc['2016','Price'].resample('Q').mean()
```

Out[25]:

```
Date
2016-03-31    33.842742
2016-06-30    45.566875
2016-09-30    45.801061
2016-12-31    49.052222
Freq: Q-DEC, Name: Price, dtype: float64
```

==========

GOOD LUCK!