1_TimeSeriesData

A time series is the series of data points listed in time order.

A time series is a sequence of successive equal interval points in time.

A time-series analysis consists of methods for analyzing time series data in order to extract meaningful insights and other useful characteristics of data.

For performing time series analysis download stock_data.csv

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
# reading the dataset using read_csv
df = pd.read_csv("/content/stock_data.csv",
                 parse_dates=True,
                 index_col="Date")

# displaying the first five rows of dataset
df.head()
```

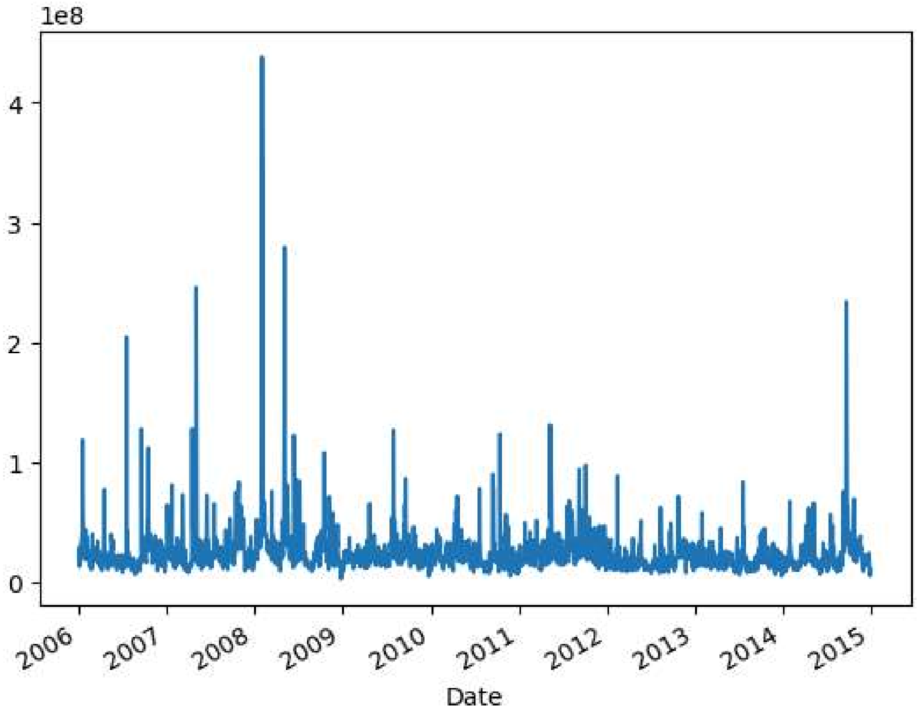
```
             Open   High    Low  Close    Volume  Name
Date
2006-01-03  39.69  41.22  38.79  40.91  24232729  AABA
2006-01-04  41.22  41.90  40.77  40.97  20553479  AABA
2006-01-05  40.93  41.73  40.85  41.53  12829610  AABA
2006-01-06  42.88  43.57  42.80  43.21  29422828  AABA
2006-01-09  43.10  43.66  42.82  43.42  16268338  AABA
```

Next steps: Generate code with df View recommended plots

```
# deleting column
df=df.drop(columns='Name',axis=1)
```

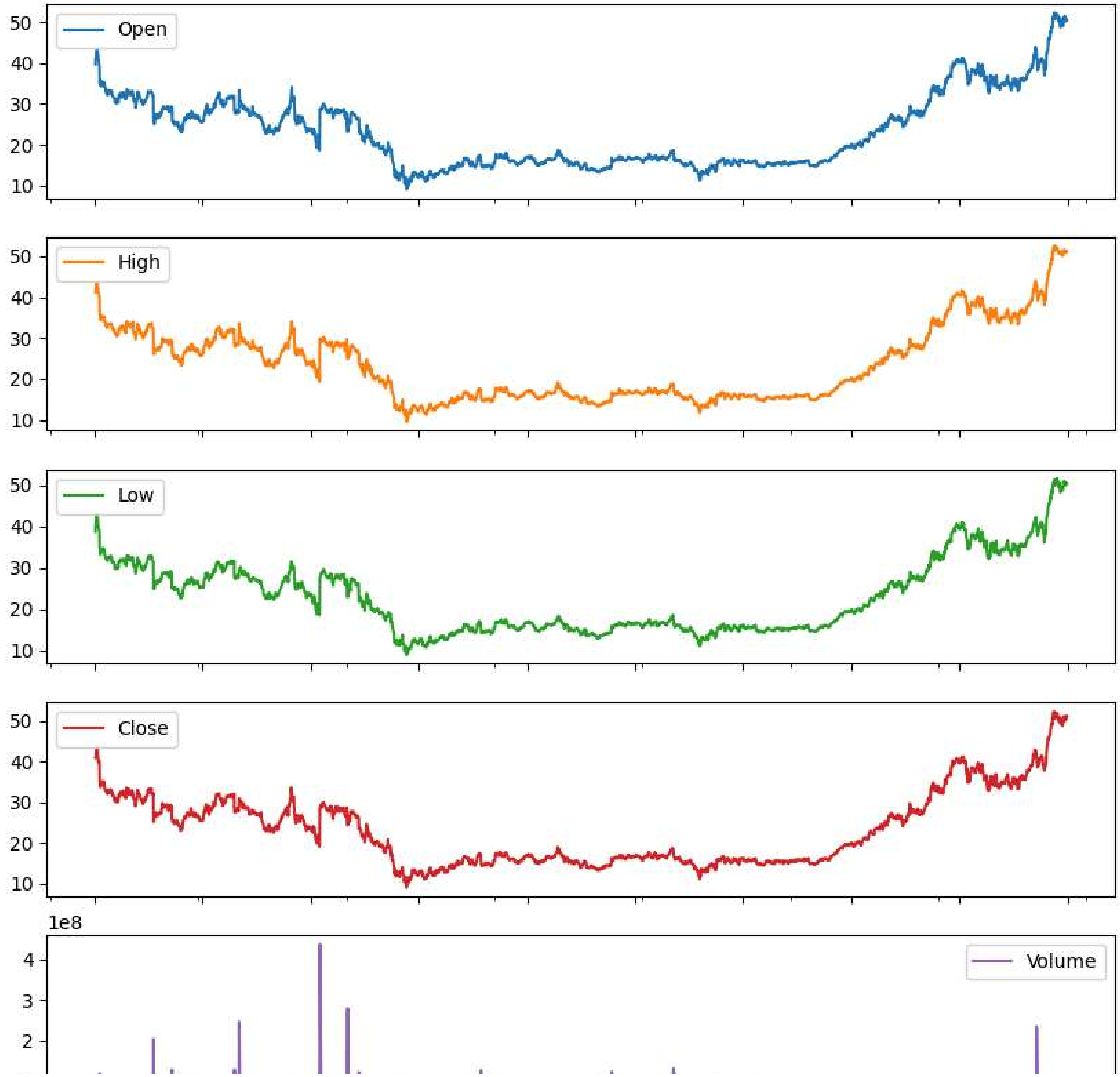
```
df['Volume'].plot()
```

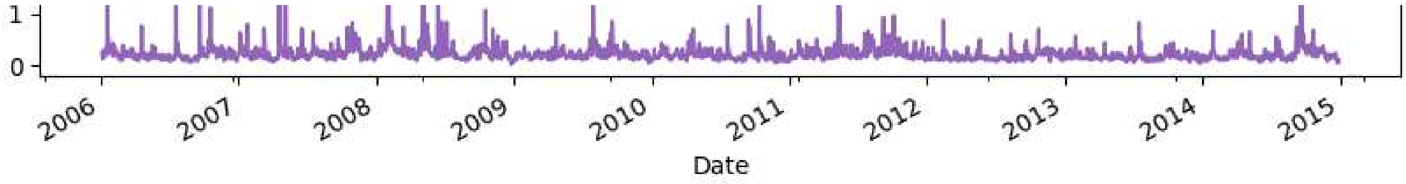
```
<Axes: xlabel='Date'>
```



```
df.plot(subplots=True, figsize=(10, 12))
```

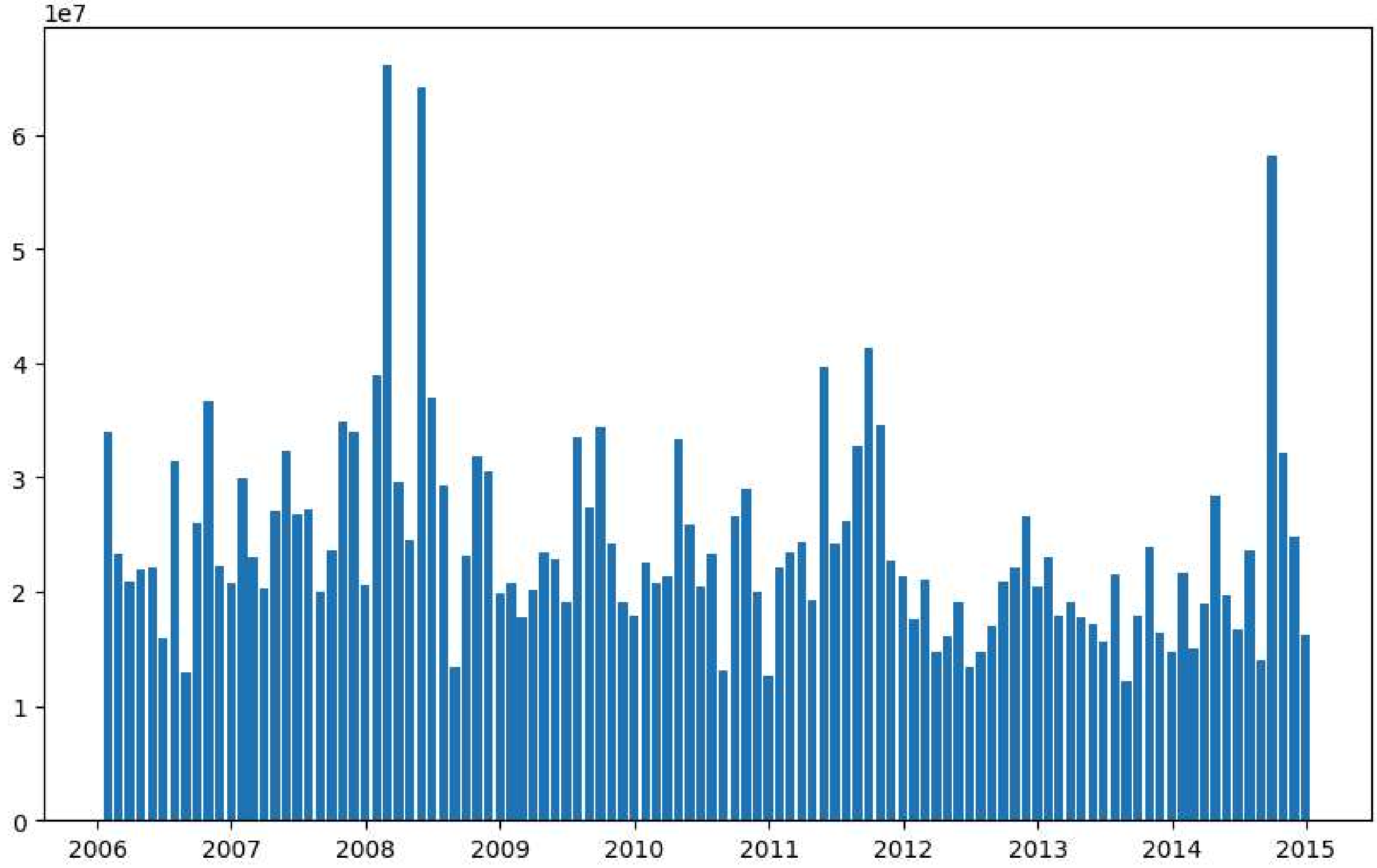
```
array([<Axes: xlabel='Date'>, <Axes: xlabel='Date'>,
       <Axes: xlabel='Date'>, <Axes: xlabel='Date'>,
       <Axes: xlabel='Date'>], dtype=object)
```





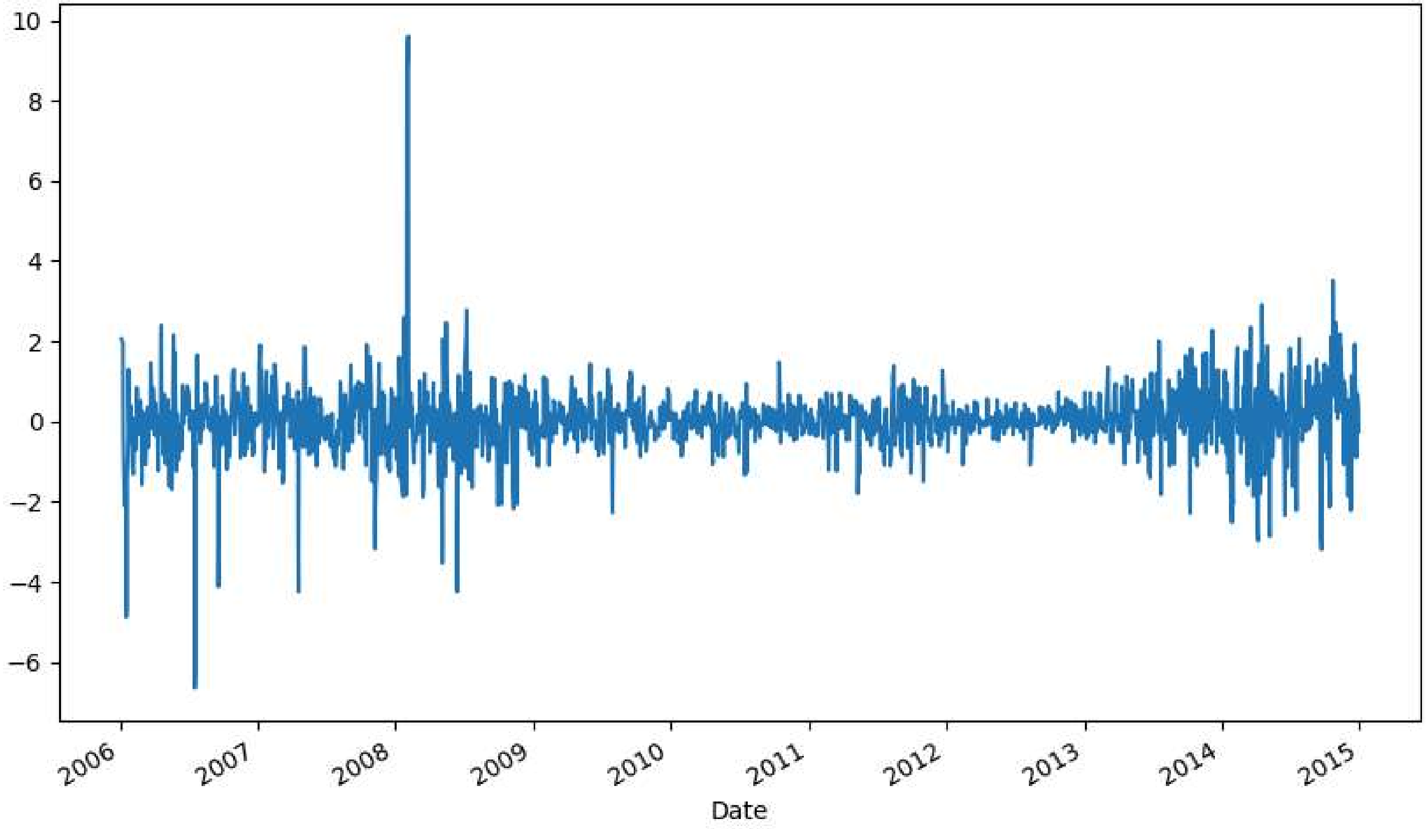
```
# Resampling the time series data based on monthly 'M' frequency
df_month = df.resample("M").mean()

# using subplot
fig, ax = plt.subplots(figsize=(10, 6))

# plotting bar graph
ax.bar(df_month['2006':].index,
       df_month.loc['2006':, "Volume"],
       width=25, align='center')
```

```
<BarContainer object of 108 artists>
```



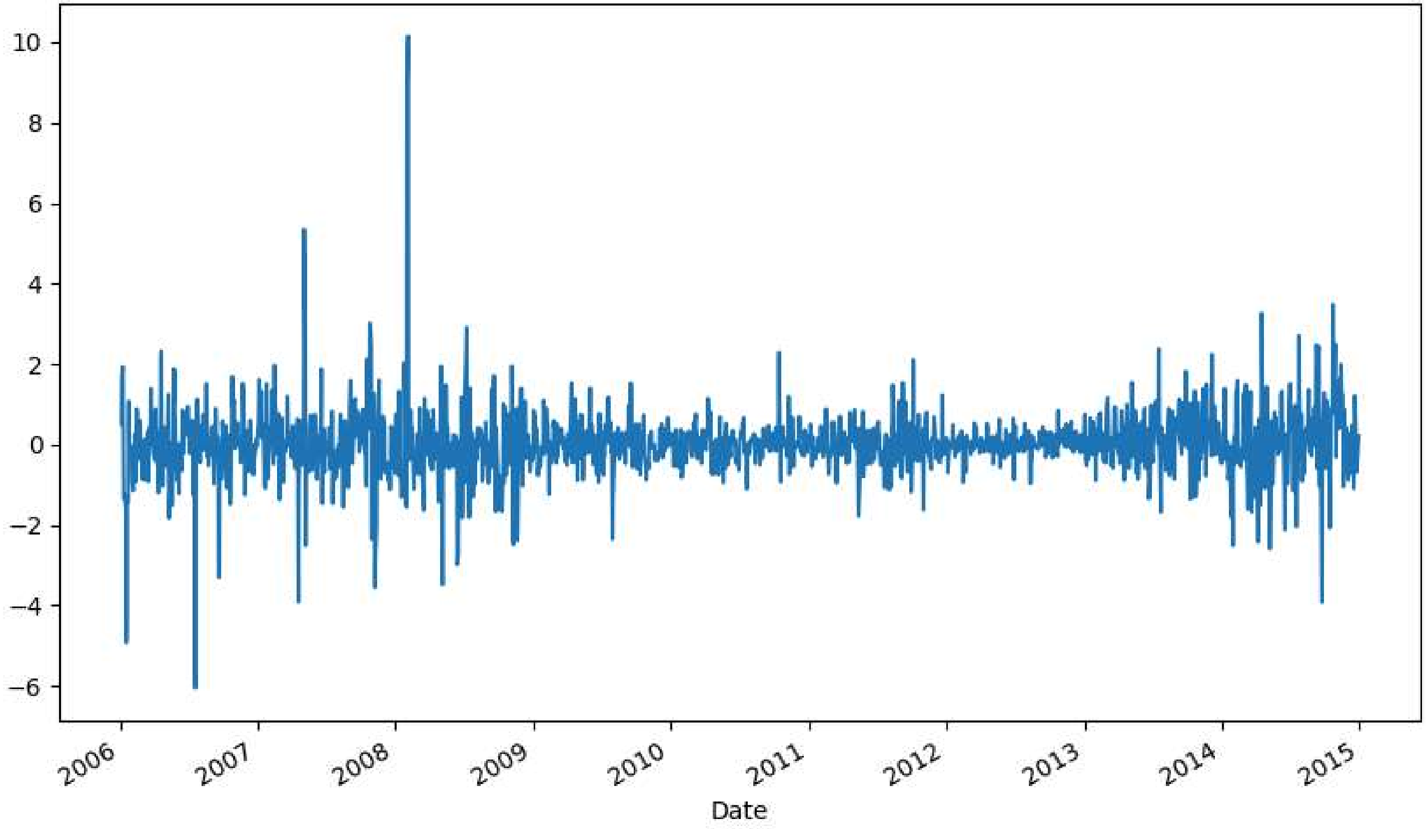
```
df.Low.diff(2).plot(figsize=(10, 6))
```

```
<Axes: xlabel='Date'>
```



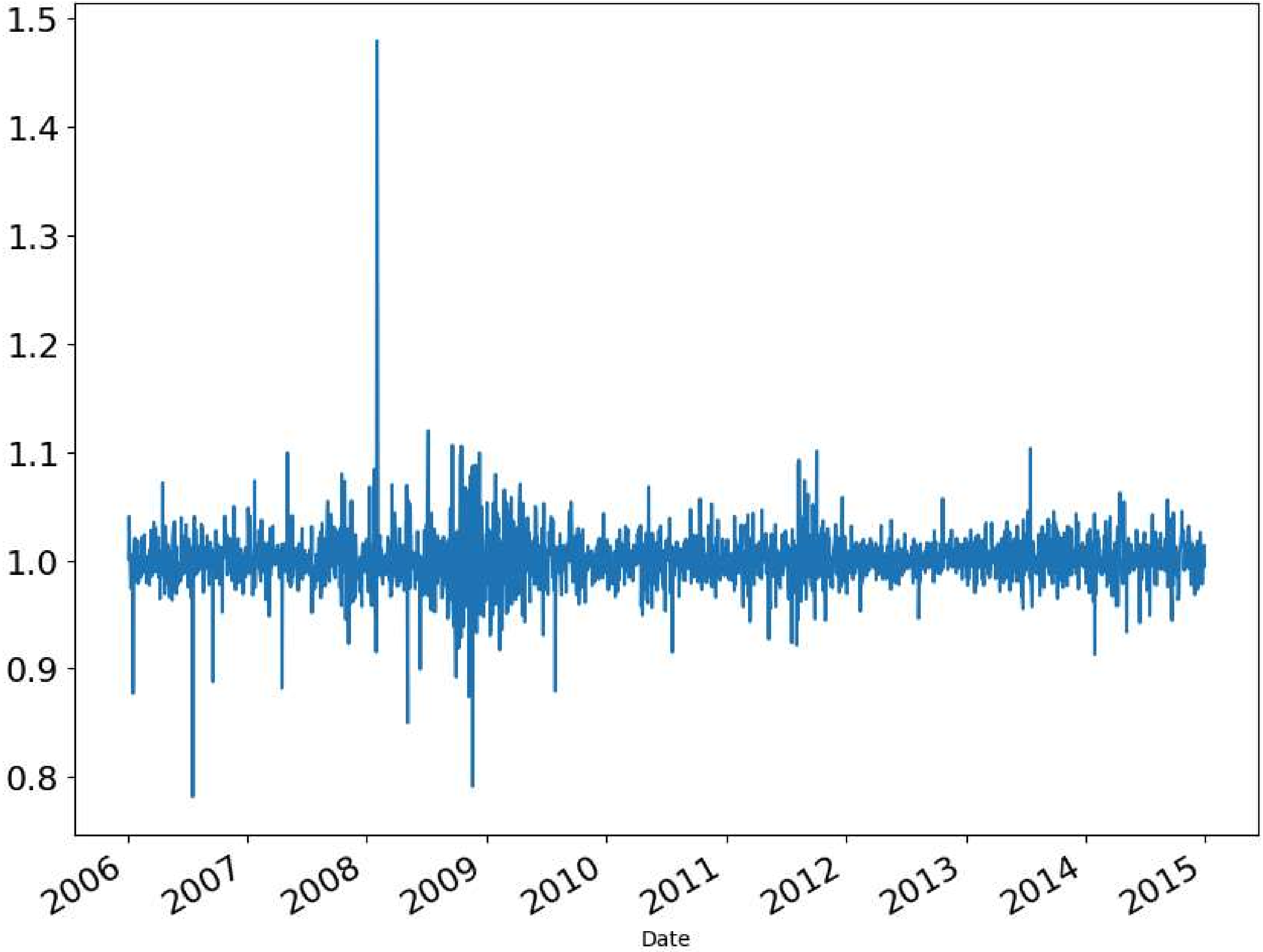
```
df.High.diff(2).plot(figsize=(10, 6))
```

```
<Axes: xlabel='Date'>
```



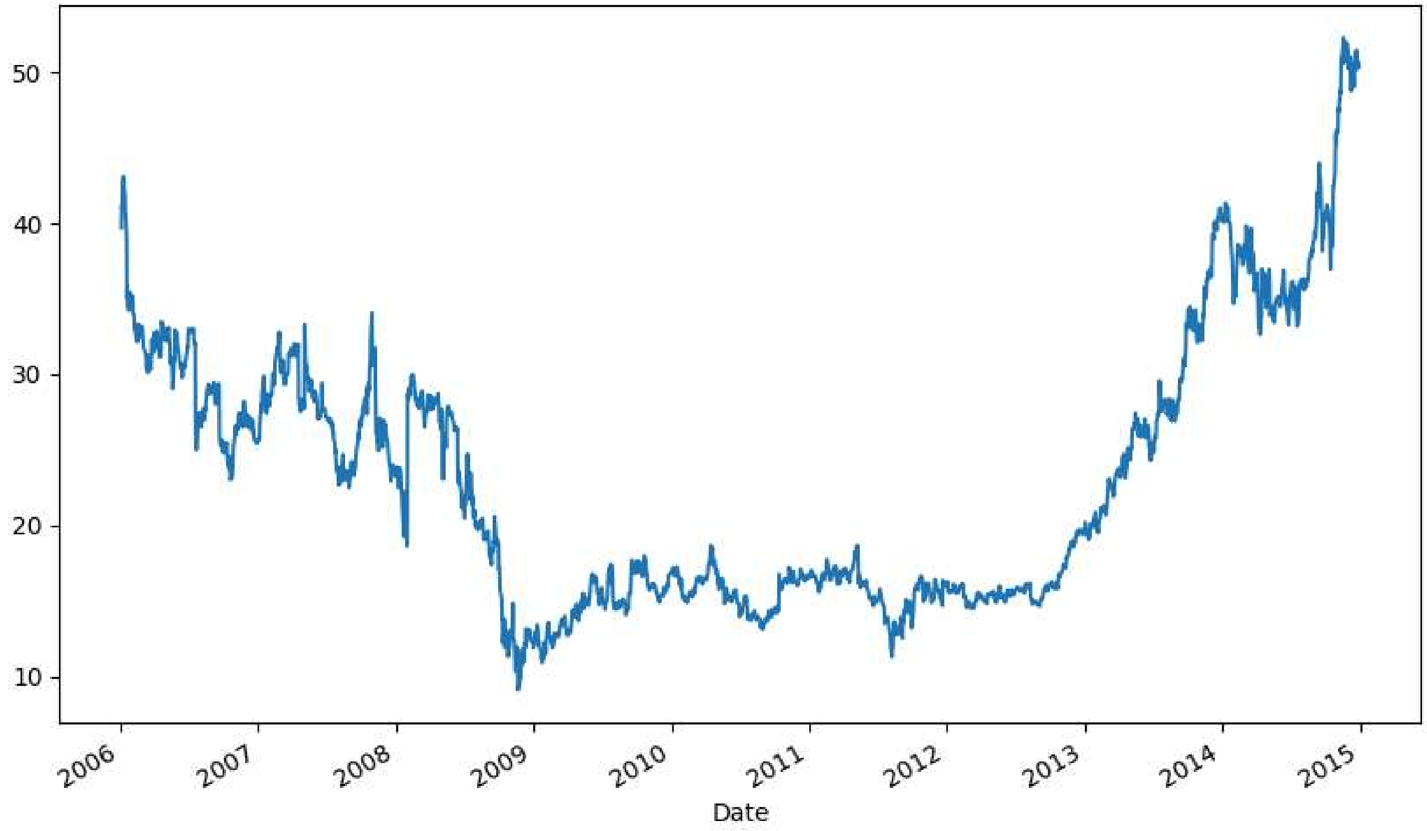
```
df['Change'] = df.Close.div(df.Close.shift())
df['Change'].plot(figsize=(10, 8), fontsize=16)
```

```
<Axes: xlabel='Date'>
```



```
df['Open'].plot(figsize=(10, 6))
```

```
<Axes: xlabel='Date'>
```



Market_Basket

Market basket analysis is used by companies to identify items that are frequently purchased together.

How Does Market Basket Analysis Work?

Market basket analysis is frequently used by restaurants, retail stores, and online shopping platforms to encourage customers to make more purchases in a single visit. This is a use-case of data science in marketing that increases company sales and drives business growth and commonly utilizes the Apriori algorithm.

What is the Apriori Algorithm?

The Apriori algorithm is the most common technique for performing market basket analysis.

It is used for association rule mining, which is a rule-based process used to identify correlations between items purchased by users.

What Are the Components of the Apriori Algorithm?

The Apriori algorithm has three main components:

- Support
- Lift
- Confidence

Here is a tabular representation of this purchase data:

```
      Milk      Beer      Eggs   Bread  Bananas   Apples
```

Basket1 1 1 1 1 0 0

Basket2 1 0 0 1 0 0

Basket3 1 0 0 1 0 1

Basket4 0 0 0 1 1 1

Let’s calculate the support, confidence, and lift.

Support

The first component of the Apriori algorithm is support – we use it to assess the overall popularity of a given product with the following formula:

Support(item) = Transactions comprising the item / Total transactions

A high support value indicates that the item is present in most purchases, therefore marketers should focus on it more.

Confidence

Confidence tells us the likelihood of different purchase combinations. We calculate that using the following formula:

Confidence (Bread -> Milk) = Transactions comprising bread and milk / Transactions comprising bread

Lift

Finally, lift refers to the increase in the ratio of the sale of milk when you sell bread:

Lift = Confidence (Bread -> Milk) / Support(Bread) = 0.75/1 = 1.3.

This means that customers are 1.3 times more likely to buy milk if you also sell bread.

Step 1: Pre-Requisites for Performing Market Basket Analysis

Download the dataset "groceries_dataset.csv"

Step 2: Reading the Dataset

```
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')
df = pd.read_csv('content/drive/My Drive/Data/Groceries_dataset.csv')
df.head()
```

Step 3: Data Preparation for Market Basket Analysis

Before we perform market basket analysis, we need to convert this data into a format that can easily be ingested into the Apriori algorithm. In other words, we need to turn it into a tabular structure comprising ones and zeros, as displayed in the bread and milk example above.

To achieve this, the first group items that have the same member number and date:

```
df[‘single_transaction’] = df[‘Member_number’].astype(str)+’_’+df[‘Date’].astype(str)

df.head()
```

```
df2 = pd.crosstab(df['single_transaction'], df['itemDescription'])
df2.head()
```

```
def encode(item_freq):
    res = 0
    if item_freq > 0:
        res = 1
    return res

basket_input = df2.applymap(encode)
```

Step 4: Build the Apriori Algorithm for Market Basket Analysis

Now, let’s import the Apriori algorithm from the MLXtend Python package and use it to discover frequently-bought-together item combinations:

```
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

frequent_itemsets = apriori(basket_input, min_support=0.001, use_colnames=True)

rules = association_rules(frequent_itemsets, metric="lift")

rules.head()
```

```
rules.sort_values(["support", "confidence","lift"],axis = 0, ascending = False).head(8)
```

3_TextVisualization

Load the Pacakges

To get started, open a Colab notebook and load the Pandas, Matplotlib, and Wordcloud packages.

```
import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from wordcloud import STOPWORDS
```

```
from google.colab import drive
```

```
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

```
df=pd.read_csv('/content/netflix_titles.csv', usecols=['cast'])
df.head()
```

	cast
0	NaN
1	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
2	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
3	NaN
4	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...

Next steps: Generate code with `df` View recommended plots

```
ndf=df.dropna()
ndf.head()
```

	cast
1	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...
2	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...
4	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...
5	Kate Siegel, Zach Gilford, Hamish Linklater, H...
6	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...

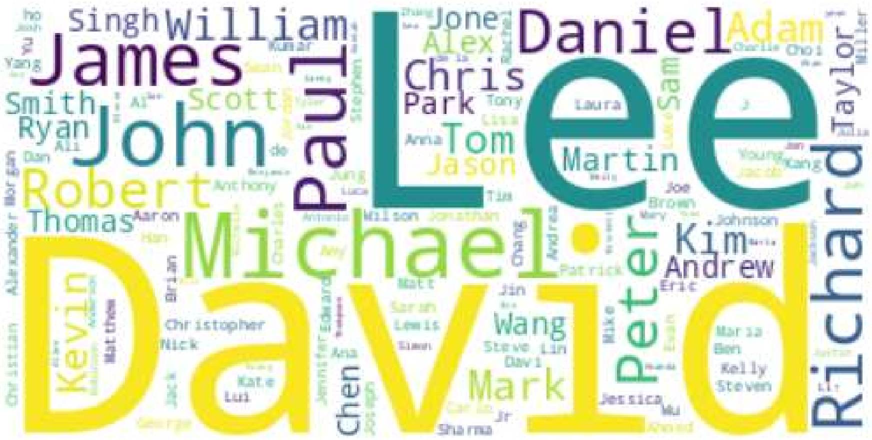
Next steps: Generate code with `ndf` View recommended plots

```
text = " ".join(item for item in ndf['cast'])
print(text)
```

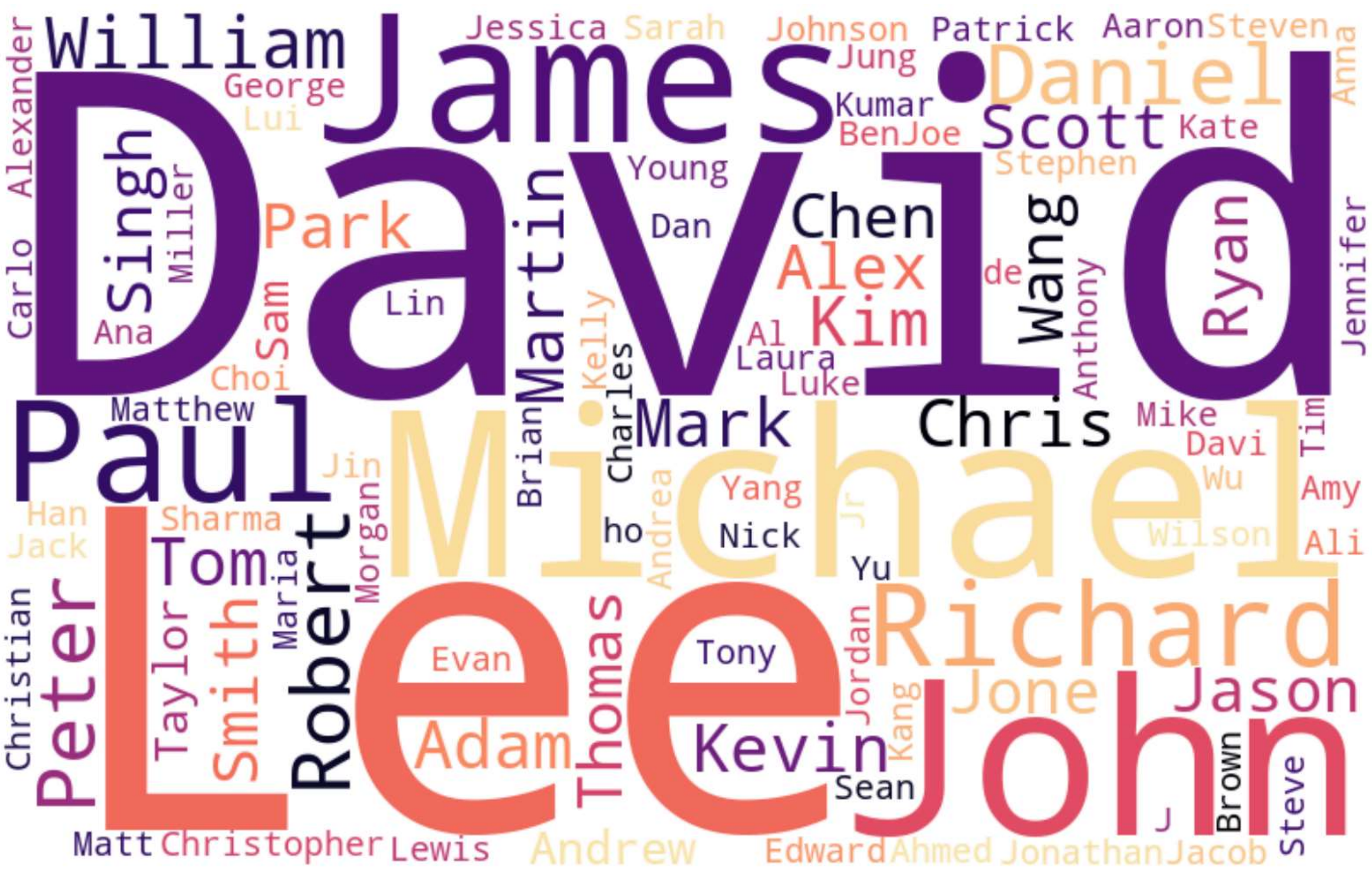
```
Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Dillon Windvogel, Natasha Thahane, Arno Greeff, Xolile Tshabalala, Getmore Sitho
```

```
stopwords = set(STOPWORDS)
```

```
wordcloud = WordCloud(background_color="white").generate(text)
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0, y=0)
plt.show()
```



```
wordcloud = WordCloud(background_color="white",
                      max_words=100,
                      max_font_size=300,
                      width=800,
                      height=500,
                      colormap="magma"
                     ).generate(text)

plt.figure(figsize=(20,20))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.margins(x=0, y=0)
plt.savefig("cloud.jpg", format="jpg")
plt.show()
```



1_Clustering_Hiatogram_HeatMap

What is Clustering?

Clustering is the process of separating different parts of data based on common characteristics. Disparate industries including retail, finance and healthcare use clustering techniques for various analytical tasks.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
df = pd.read_csv('/content/Mall_Customers.csv')
print(df.head(15))
```

```
    CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0            1    Male   19                  15                      39
1            2    Male   21                  15                      81
2            3  Female   20                  16                       6
3            4  Female   23                  16                      77
4            5  Female   31                  17                      40
5            6  Female   22                  17                      76
6            7  Female   35                  18                       6
7            8  Female   23                  18                      94
8            9    Male   64                  19                       3
9           10  Female   30                  19                      72
10          11    Male   67                  19                      14
11          12  Female   35                  19                      99
12          13  Female   58                  20                      15
13          14  Female   24                  20                      77
14          15    Male   37                  20                      13
```

```
from sklearn.cluster import KMeans
```