advanced technological analysis in python

1)time series analysis

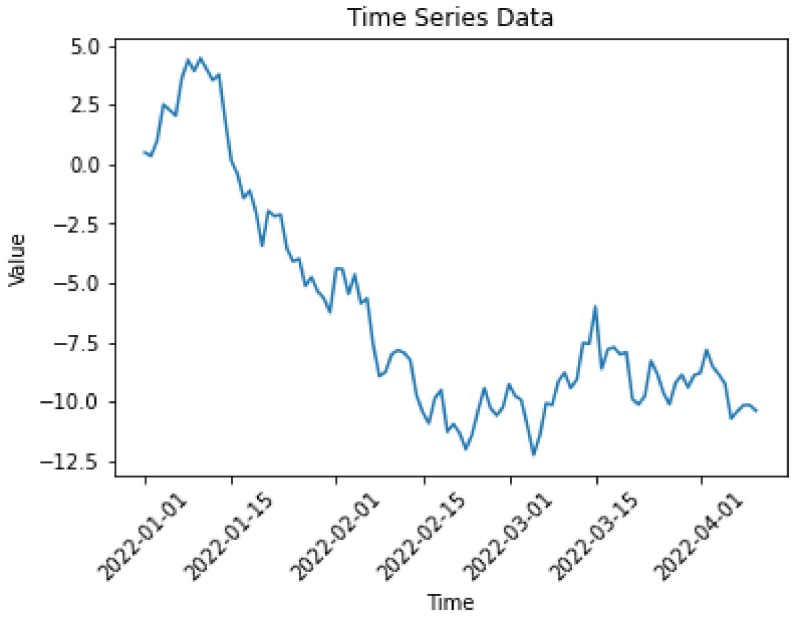
In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Generate random time-series data
np.random.seed(42)
dates = pd.date_range(start='2022-01-01', periods=100, freq='D')
values = np.random.randn(100).cumsum()

# Create a DataFrame from the generated data
data = pd.DataFrame({'date': dates, 'value': values})

# Set the 'date' column as the index
data.set_index('date', inplace=True)

# Plot the time-series data
plt.plot(data.index, data['value'])
plt.xlabel('Time')
plt.ylabel('Value')
plt.xticks(rotation = 45)
plt.title('Time Series Data')
plt.show()
```



2)sentimental analysis

In [4]:
```
pip install textblob
```

```
Requirement already satisfied: textblob in c:\users\acer\anaconda3\lib\site-p
ackages (0.18.0.post0)
Requirement already satisfied: nltk>=3.8 in c:\users\acer\anaconda3\lib\site-
packages (from textblob) (3.8.1)
Requirement already satisfied: tqdm in c:\users\acer\anaconda3\lib\site-packa
ges (from nltk>=3.8->textblob) (4.47.0)
Requirement already satisfied: joblib in c:\users\acer\anaconda3\lib\site-pac
kages (from nltk>=3.8->textblob) (0.16.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\acer\anaconda3\lib
\site-packages (from nltk>=3.8->textblob) (2024.5.15)
Requirement already satisfied: click in c:\users\acer\anaconda3\lib\site-pack
ages (from nltk>=3.8->textblob) (7.1.2)
Note: you may need to restart the kernel to use updated packages.
```

In [6]:
```
from textblob import TextBlob

text_1 = "The movie was so awesome."
text_2 = "The food here tastes terrible."

#Determining the Polarity
p_1 = TextBlob(text_1).sentiment.polarity
p_2 = TextBlob(text_2).sentiment.polarity

#Determining the Subjectivity
s_1 = TextBlob(text_1).sentiment.subjectivity
s_2 = TextBlob(text_2).sentiment.subjectivity

print("Polarity of Text 1 is", p_1)
print("Polarity of Text 2 is", p_2)
print("Subjectivity of Text 1 is", s_1)
print("Subjectivity of Text 2 is", s_2)
```

```
Polarity of Text 1 is 1.0
Polarity of Text 2 is -1.0
Subjectivity of Text 1 is 1.0
Subjectivity of Text 2 is 1.0
```

dictionary

In [8]:
```
pip install collections
```

```
Note: you may need to restart the kernel to use updated packages.

ERROR: Could not find a version that satisfies the requirement collections (f
rom versions: none)
ERROR: No matching distribution found for collections
```

In [9]:

```python
from collections import Counter

# counts word frequency
def count_words(text):
    skips = [".", ", ", ":", ";", "'", '"']
    for ch in skips:
        text = text.replace(ch, "")
    word_counts = {}
    for word in text.split(" "):
        if word in word_counts:
            word_counts[word]+= 1
        else:
            word_counts[word]= 1
    return word_counts

    # >>>count_words(text) You can check the function

# counts word frequency using
# Counter from collections
def count_words_fast(text):
    text = text.lower()
    skips = [".", ", ", ":", ";", "'", '"']
    for ch in skips:
        text = text.replace(ch, "")
    word_counts = Counter(text.split(" "))
    return word_counts
```

3)harmonical analysis pattern

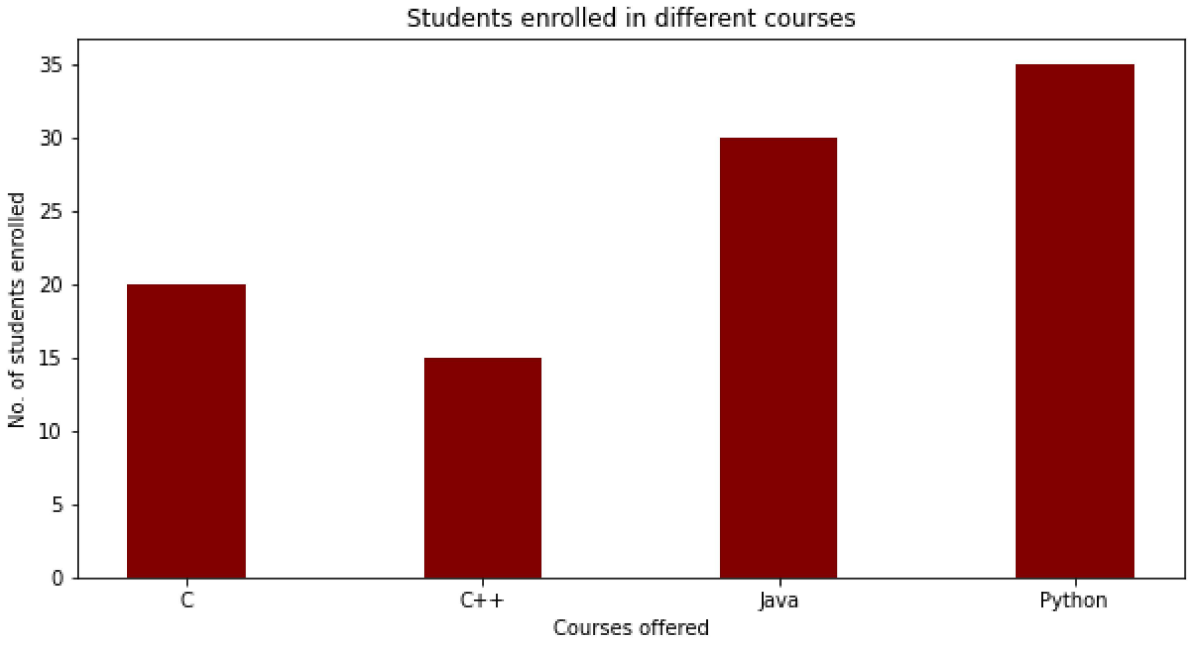
In [11]:
```
import numpy as np
import matplotlib.pyplot as plt


# creating the dataset
data = {'C':20, 'C++':15, 'Java':30,
        'Python':35}
courses = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(courses, values, color ='maroon',
        width = 0.4)

plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses")
plt.show()
```



4)simple harmonic

In [19]:
```
import pandas as pd
import datetime
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
!pip install yfinance
import yfinance as yf
%matplotlib inline
```

```
Collecting yfinance
  Using cached yfinance-0.2.40-py2.py3-none-any.whl (73 kB)
Requirement already satisfied: lxml>=4.9.1 in c:\users\acer\anaconda3\lib\sit
e-packages (from yfinance) (5.2.2)
Requirement already satisfied: pandas>=1.3.0 in c:\users\acer\anaconda3\lib\s
ite-packages (from yfinance) (2.0.3)
Requirement already satisfied: numpy>=1.16.5 in c:\users\acer\anaconda3\lib\s
ite-packages (from yfinance) (1.18.5)
Requirement already satisfied: requests>=2.31 in c:\users\acer\anaconda3\lib
\site-packages (from yfinance) (2.32.2)
Requirement already satisfied: html5lib>=1.1 in c:\users\acer\anaconda3\lib\s
ite-packages (from yfinance) (1.1)
Requirement already satisfied: peewee>=3.16.2 in c:\users\acer\anaconda3\lib
\site-packages (from yfinance) (3.17.5)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\acer\anacon
da3\lib\site-packages (from yfinance) (4.12.3)
Requirement already satisfied: pytz>=2022.5 in c:\users\acer\anaconda3\lib\si
te-packages (from yfinance) (2024.1)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\acer\anaconda3\l
ib\site-packages (from yfinance) (2.4.4)
Requirement already satisfied: platformdirs>=2.0.0 in c:\users\acer\anaconda3
\lib\site-packages (from yfinance) (4.2.2)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\acer\anaconda3
\lib\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\acer\anacon
da3\lib\site-packages (from pandas>=1.3.0->yfinance) (2.9.0.post0)
Requirement already satisfied: tzdata>=2022.1 in c:\users\acer\anaconda3\lib
\site-packages (from pandas>=1.3.0->yfinance) (2024.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\acer\anaconda3\lib\si
te-packages (from requests>=2.31->yfinance) (2.10)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\acer\anac
onda3\lib\site-packages (from requests>=2.31->yfinance) (3.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\acer\anaconda3
\lib\site-packages (from requests>=2.31->yfinance) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\acer\anaconda3
\lib\site-packages (from requests>=2.31->yfinance) (2020.6.20)
Requirement already satisfied: six>=1.9 in c:\users\acer\anaconda3\lib\site-p
ackages (from html5lib>=1.1->yfinance) (1.15.0)
Requirement already satisfied: webencodings in c:\users\acer\anaconda3\lib\si
te-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\acer\anaconda3\lib\s
ite-packages (from beautifulsoup4>=4.11.1->yfinance) (2.0.1)
Installing collected packages: yfinance
Successfully installed yfinance-0.2.40
```

In [33]:
```
!pip install yfinance
```

```
Requirement already satisfied: yfinance in c:\users\acer\anaconda3\lib\site-p
ackages (0.2.40)
Requirement already satisfied: numpy>=1.16.5 in c:\users\acer\anaconda3\lib\s
ite-packages (from yfinance) (1.18.5)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\acer\anaconda3
\lib\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: beautifulsoup4>=4.11.1 in c:\users\acer\anacon
da3\lib\site-packages (from yfinance) (4.12.3)
Requirement already satisfied: pytz>=2022.5 in c:\users\acer\anaconda3\lib\si
te-packages (from yfinance) (2024.1)
Requirement already satisfied: peewee>=3.16.2 in c:\users\acer\anaconda3\lib
\site-packages (from yfinance) (3.17.5)
Requirement already satisfied: html5lib>=1.1 in c:\users\acer\anaconda3\lib\s
ite-packages (from yfinance) (1.1)
Requirement already satisfied: frozendict>=2.3.4 in c:\users\acer\anaconda3\l
ib\site-packages (from yfinance) (2.4.4)
Requirement already satisfied: pandas>=1.3.0 in c:\users\acer\anaconda3\lib\s
ite-packages (from yfinance) (2.0.3)
Requirement already satisfied: requests>=2.31 in c:\users\acer\anaconda3\lib
\site-packages (from yfinance) (2.32.2)
Requirement already satisfied: platformdirs>=2.0.0 in c:\users\acer\anaconda3
\lib\site-packages (from yfinance) (4.2.2)
Requirement already satisfied: lxml>=4.9.1 in c:\users\acer\anaconda3\lib\sit
e-packages (from yfinance) (5.2.2)
Requirement already satisfied: soupsieve>1.2 in c:\users\acer\anaconda3\lib\s
ite-packages (from beautifulsoup4>=4.11.1->yfinance) (2.0.1)
Requirement already satisfied: webencodings in c:\users\acer\anaconda3\lib\si
te-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: six>=1.9 in c:\users\acer\anaconda3\lib\site-p
ackages (from html5lib>=1.1->yfinance) (1.15.0)
Requirement already satisfied: tzdata>=2022.1 in c:\users\acer\anaconda3\lib
\site-packages (from pandas>=1.3.0->yfinance) (2024.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\acer\anacon
da3\lib\site-packages (from pandas>=1.3.0->yfinance) (2.9.0.post0)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\acer\anaconda3
\lib\site-packages (from requests>=2.31->yfinance) (1.25.9)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\acer\anac
onda3\lib\site-packages (from requests>=2.31->yfinance) (3.3.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\acer\anaconda3
\lib\site-packages (from requests>=2.31->yfinance) (2020.6.20)
Requirement already satisfied: idna<4,>=2.5 in c:\users\acer\anaconda3\lib\si
te-packages (from requests>=2.31->yfinance) (2.10)
```

In [36]:
```
import pandas as pd
import yfinance as yf
import datetime
from datetime import date, timedelta
today = date.today()

d1 = today.strftime("%Y-%m-%d")
end_date = d1
d2 = date.today() - timedelta(days=365)
d2 = d2.strftime("%Y-%m-%d")
start_date = d2

data = yf.download('ADANIENT.NS',
                   start=start_date,
                   end=end_date,
                   progress=False)
data["Date"] = data.index
data = data[["Date", "Open", "High", "Low",
             "Close", "Adj Close", "Volume"]]
data.reset_index(drop=True, inplace=True)
print(data.head())
```

```
1 Failed download:
['ADANIENT.NS']: ImportError("cannot import name 'FuncType' from 'pandas._typ
ing' (C:\\Users\\ACER\\anaconda3\\lib\\site-packages\\pandas\\_typing.py)")

Empty DataFrame
Columns: [Date, Open, High, Low, Close, Adj Close, Volume]
Index: []
```

5)*clustering - K*

In [*]:
```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = [7.00, 3.50]
plt.rcParams["figure.autolayout"] = True

x = np.random.randn(10)
y = np.random.randn(10)
Cluster = np.array([0, 1, 1, 1, 3, 2, 2, 3, 0, 2])
centers = np.random.randn(4, 2)

fig = plt.figure()
ax = fig.add_subplot(111)

scatter = ax.scatter(x, y, c=Cluster, s=50)
for i, j in centers:
   ax.scatter(i, j, s=50, c='red', marker='+')

plt.show()
```

In [ ]: