COURSE: FUNDAMENTALS OF DATA SCIENCE

CODE:DSA0406

NAME: JUHAINA AFREEN A

REG: 192224158

11. Scenario : You are a data scientist working for a company that sells products online.

You have been tasked with creating a simple plot to show the sales of a product over time.

Question:

1. Write code to create a simple line plot in Python using Matplotlib to predict sales happened in a month?

2. Write code to create a scatter plot in Python using Matplotlib to predict sales happened in a month?

3. Develop a Python program to create a bar plot of the monthly sales data. import

matplotlib.pyplot as plt months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',

'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

sales = [150, 200, 250, 300, 280, 350, 400, 420, 390, 450, 470, 500]

plt.figure(figsize=(8, 5)) plt.plot(months, sales, marker='o',

linestyle='-', color='blue') plt.title('NEWS.txtMonthly Sales (Line

Plot)') plt.xlabel('Month') plt.ylabel('Sales') plt.grid(True)

plt.show() plt.figure(figsize=(8, 5)) plt.scatter(months, sales,

color='green') plt.title('2.Monthly Sales (Scatter Plot)')

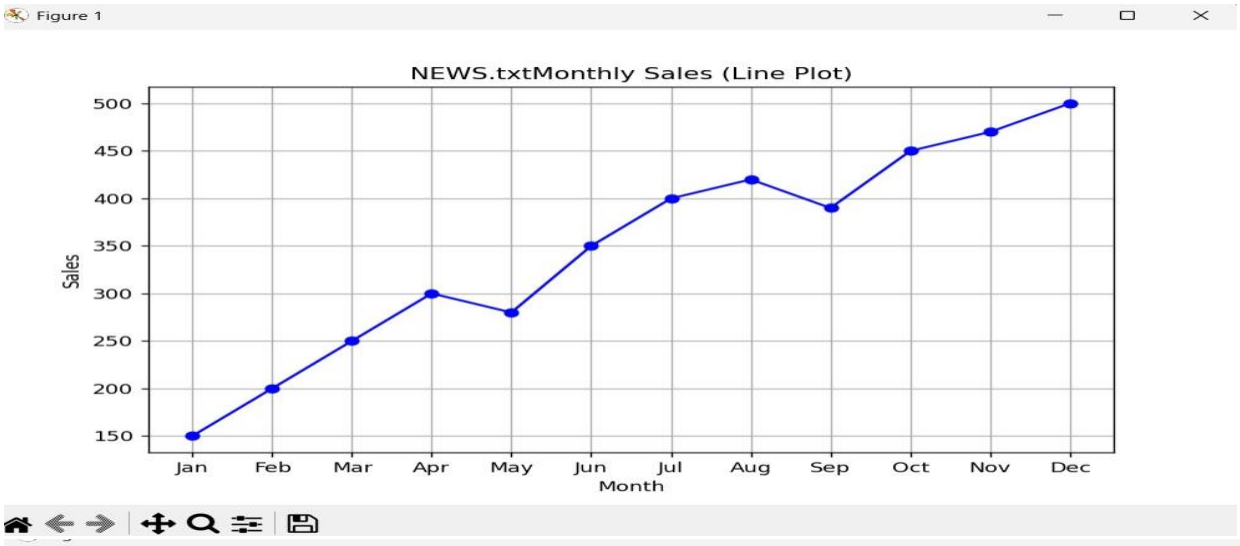
plt.xlabel('Month') plt.ylabel('Sales') plt.grid(True) plt.show()

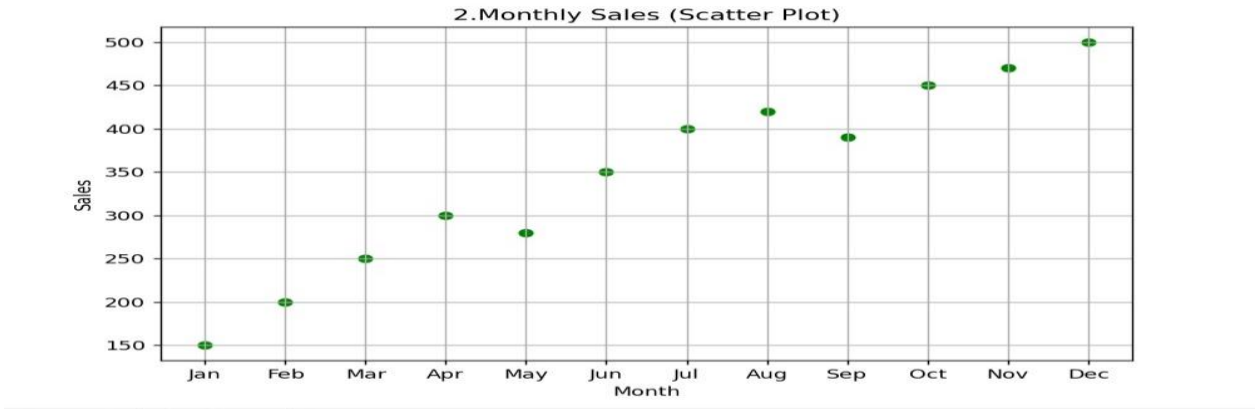
plt.figure(figsize=(8, 5)) plt.bar(months, sales, color='orange')

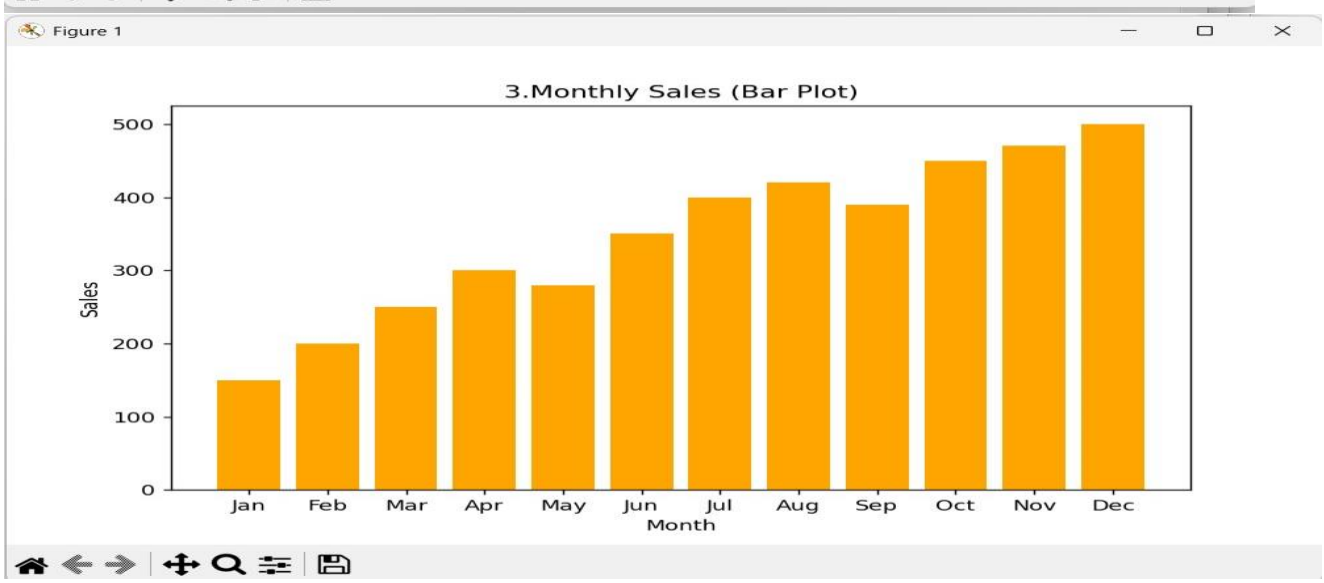
plt.title('3.Monthly Sales (Bar Plot)')

plt.xlabel('Month')

plt.ylabel('Sales') plt.show()







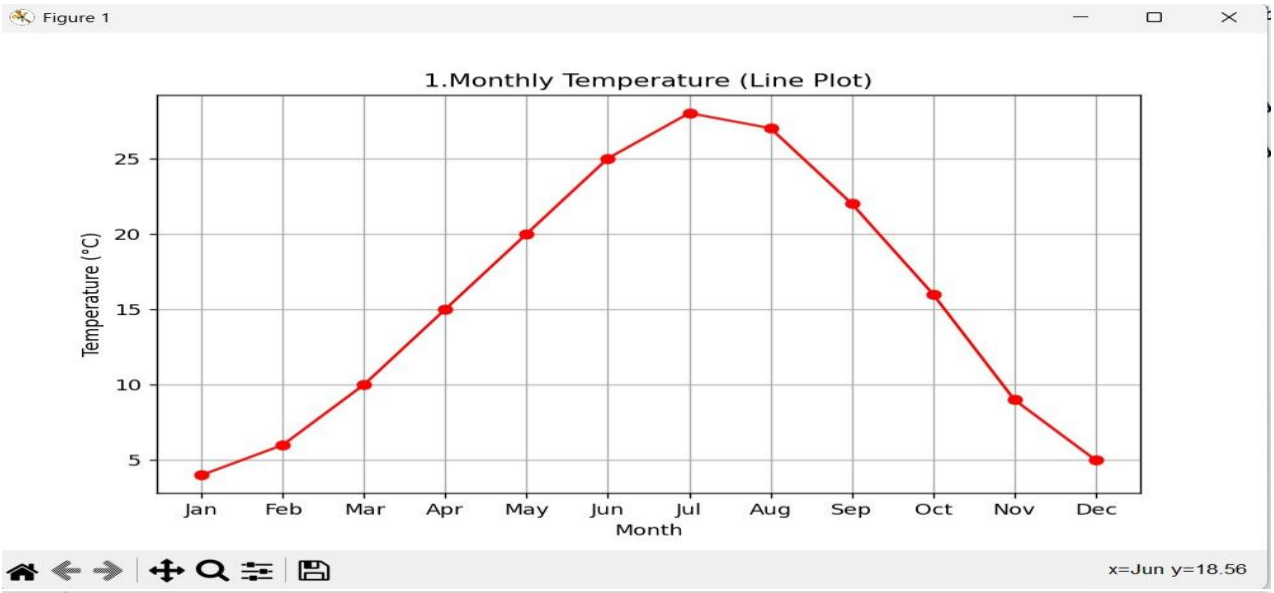
12. Scenario: You are working on a data analysis project that involves analyzing the monthly temperature and rainfall data for a city. You have a dataset containing the monthly temperature and rainfall values for each month of a year. Your task is to develop a Python program that generates line plots and scatter plots to visualize the temperature and rainfall data.

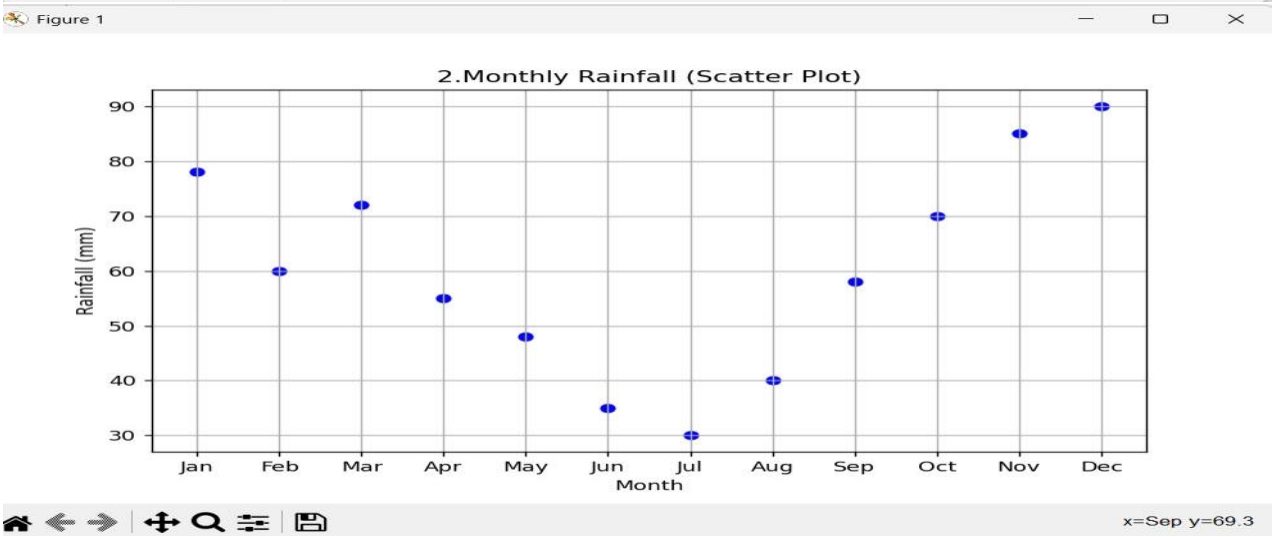
Question:

1. Develop a Python program to create a line plot of the monthly temperature data.

2: Develop a Python program to create a scatter plot of the monthly rainfall data.

```
import matplotlib.pyplot as plt months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
temperature = [4, 6, 10, 15, 20, 25, 28, 27, 22, 16, 9, 5]
rainfall = [78, 60, 72, 55, 48, 35, 30, 40, 58, 70, 85, 90]
plt.figure(figsize=(8, 5)) plt.plot(months, temperature,
marker='o', color='red') plt.title('1.Monthly
Temperature (Line Plot)') plt.xlabel('Month')
plt.ylabel('Temperature (°C)') plt.grid(True) plt.show()
plt.figure(figsize=(8, 5)) plt.scatter(months, rainfall,
color='blue') plt.title('2.Monthly Rainfall (Scatter
Plot)') plt.xlabel('Month') plt.ylabel('Rainfall (mm)')
plt.grid(True) plt.show()
```



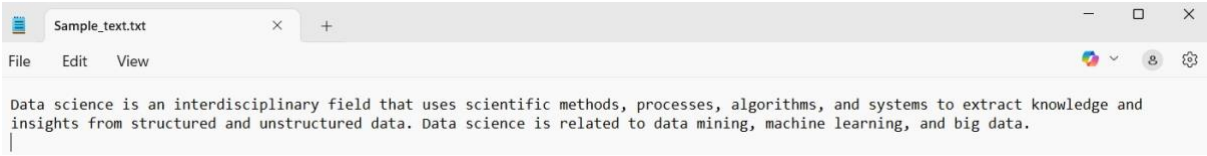


13. Scenario: You are working on a text analysis project and need to determine the frequency distribution of words in a given text document. You have a text document named "sample_text.txt" containing a paragraph of text. Your task is to develop a Python program that reads the text document, processes the text, and generates a frequency distribution of the words.

Question: How would you develop a Python program to calculate the frequency distribution of words in a text document?

```
import string from collections

import Counter

with open('sample_text.txt', 'r') as file:

   text = file.read() text = text.lower() text =

text.translate(str.maketrans('', '', string.punctuation))

words = text.split()
```

```
word_freq = Counter(words) for
word, freq in word_freq.items():
print(f"{word}: {freq}")
```




```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:/Users/Mayu/OneDrive/Documents/Desktop/FOD/Lab Question 13.py
data: 5
science: 2
is: 2
an: 1
interdisciplinary: 1
field: 1
that: 1
uses: 1
scientific: 1
methods: 1
processes: 1
algorithms: 1
and: 4
systems: 1
to: 2
extract: 1
knowledge: 1
insights: 1
from: 1
structured: 1
unstructured: 1
related: 1
mining: 1
machine: 1
learning: 1
big: 1
```

14. Scenario: You are a data analyst working for a company that sells products online. You have been tasked with analyzing the sales data for the past month. The data is stored in a Pandas data frame.

Question: Develop a code in python to find the frequency distribution of the ages of the customers who have made a purchase in the past month.

```
import pandas as pd

data = {

   'CustomerID': [101, 102, 103, 104, 105, 106, 107, 108],

   'Age': [25, 30, 22, 25, 30, 40, 22, 25],

   'PurchaseAmount': [200, 150, 180, 210, 160, 300, 190, 220]

}

df = pd.DataFrame(data)
```

age_frequency = df['Age'].value_counts().sort_index()

print("Frequency distribution of customer ages:")

print(age_frequency)

```
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [M
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in

= RESTART: C:/Users/Mayu/OneDrive/Documents/Desktop/FOD/Lab Qu
Frequency distribution of customer ages:
Age
22    2
25    3
30    2
40    1
Name: count, dtype: int64
```

15. Scenario: You are a data analyst working for a social media platform. As part of your analysis, you have a dataset containing user interaction data, including the number of likes received by each post. Your task is to develop a Python program that calculates the frequency distribution of likes among the posts.

Question: Develop a Python program to calculate the frequency distribution of likes among the posts? import pandas as pd

data = {

'PostID': [201, 202, 203, 204, 205, 206, 207, 208],

'Likes': [10, 15, 10, 20, 15, 10, 25, 20]

}

df = pd.DataFrame(data) like_frequency =

df['Likes'].value_counts().sort_index()

print("Frequency distribution of likes among posts:")

print(like_frequency)

```
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" fo

= RESTART: C:/Users/Mayu/OneDrive/Documents/Desktop/F
Frequency distribution of likes among posts:
Likes
10    3
15    2
20    2
25    1
Name: count, dtype: int64
```

16. Scenario: You are working on a project that involves analyzing customer reviews for a product. You have a dataset containing customer reviews, and your task is to develop a Python program that calculates the frequency distribution of words in the reviews.

Question: Develop a Python program to calculate the frequency distribution of words in the customer reviews dataset? import pandas as pd from collections import Counter

import string data = {

```
    'ReviewID': [1, 2, 3, 4],
    'ReviewText': [
        "Great product, really loved it!",
        "Good quality, but too expensive.",
        "Amazing product, worth the price.",
        "Not bad, but expected better quality."
    ]
}
```

df = pd.DataFrame(data)

all_reviews = ' '.join(df['ReviewText'].str.lower()) all_reviews =

all_reviews.translate(str.maketrans('', '', string.punctuation)) words =

all_reviews.split() word_freq = Counter(words) print("Frequency

distribution of words in customer reviews:")

print(word_freq)

```
Edit  Shell  Debug  Options  Window  Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun  7 2023, 05:45:37) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

= RESTART: C:/Users/Mayu/OneDrive/Documents/Desktop/FOD/Lab Question 16.py
Frequency distribution of words in customer reviews:
Counter({'product': 2, 'quality': 2, 'but': 2, 'great': 1, 'really': 1, 'loved':
 1, 'it': 1, 'good': 1, 'too': 1, 'expensive': 1, 'amazing': 1, 'worth': 1, 'the
': 1, 'price': 1, 'not': 1, 'bad': 1, 'expected': 1, 'better': 1})
```

17. Scenario: You are a data analyst working for a marketing research company. Your team has collected a large dataset containing customer feedback from various social media platforms. The dataset consists of thousands of text entries, and your task is to develop a Python program to analyze the frequency distribution of words in this dataset. Your program should be able to perform the following tasks:

Load the dataset from a CSV file (data.csv) containing a single column named “feedback” with each row representing a customer comment.

Preprocess the text data by removing punctuation, converting all text to lowercase, and eliminating any stop words (common words like “the,” “and,” “is” etc. that don’t carry significant meaning).

Calculate the frequency distribution of words in the preprocessed dataset.

Display the top N most frequent words and their corresponding frequencies, where N is provided as user input.

Plot a bar graph to visualize the top N most frequent words and their frequencies.

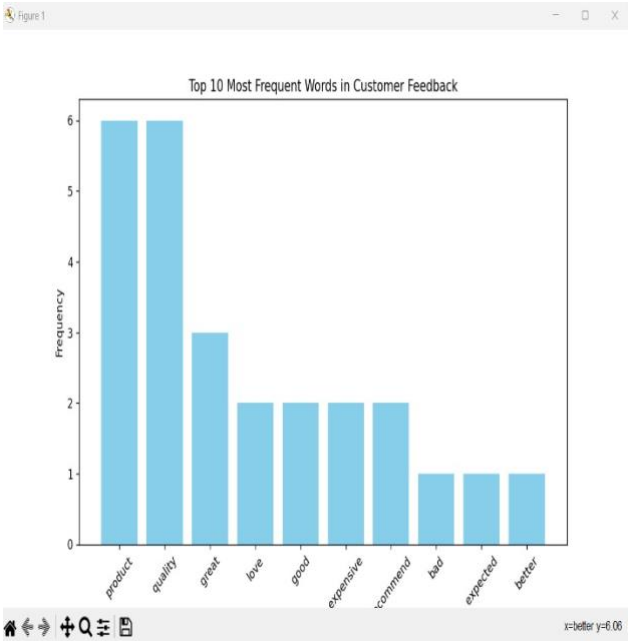
Question: Create a Python program that fulfills these requirements and gain insights from the customer feedback data.

```
import pandas as pd import string from
collections import Counter import
matplotlib.pyplot as plt from nltk.corpus
import stopwords import nltk
nltk.download('stopwords') df =
pd.read_csv('data.csv') stop_words =
set(stopwords.words('english'))
df['feedback'] = df['feedback'].str.lower()
df['feedback'] = df['feedback'].apply(lambda x: x.translate(str.maketrans('', ''),
string.punctuation))) def preprocess_text(text):
    words = text.split()
    return [word for word in words if word not in stop_words]

df['processed_feedback'] = df['feedback'].apply(preprocess_text) all_words =
[word for feedback in df['processed_feedback'] for word in feedback] word_freq
= Counter(all_words)
N = int(input("Enter the number of top frequent words to display: "))
top_n_words = word_freq.most_common(N)
```

```
print(f"\nTop {N} most frequent words:") for
word, freq in top_n_words:
    print(f"{word}: {freq}")
words, frequencies = zip(*top_n_words)
plt.figure(figsize=(10, 6)) plt.bar(words, frequencies,
color='skyblue') plt.title(f"Top {N} Most Frequent Words in
Customer Feedback") plt.xlabel('Words') plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```

```
feedback
"The product is great, I love it!"
"Good quality, but too expensive."
"Not bad, but I expected better quality."
"Excellent product, totally worth the price!"
"I love this product, it's fantastic."
"The quality of this product is amazing!"
"Not impressed with the quality, too expensive."
"Great product, would recommend it to others."
"Very good quality, highly recommend!"
"Product exceeded my expectations, great quality."
```

```
[nltk_data]   Unzipping corpora\stopwords.zip.
Enter the number of top frequent words to display: 10

Top 10 most frequent words:
product: 6
quality: 6
great: 3
love: 2
good: 2
expensive: 2
recommend: 2
bad: 1
expected: 1
better: 1
```



18. Suppose a hospital tested the age and body fat data for 18 randomly selected adults with the following result.

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

Question:

- **Calculate the mean, median and standard deviation of age and %fat using Pandas.**
- **Draw the boxplots for age and %fat.**
- **Draw a scatter plot and a q-q plot based on these two variables.**

CODE:

```
import pandas as pd import

matplotlib.pyplot as plt

import statsmodels.api as sm

data = {

   'age': [23, 23, 27, 27, 39, 41, 47, 49, 50,

          52, 54, 54, 56, 57, 58, 58, 60, 61],

   '%fat': [9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2,

           34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7]

}
```

```
df = pd.DataFrame(data)
stats = df.agg(['mean', 'median', 'std']) plt.figure(figsize=(15,
10))
plt.subplots_adjust(hspace=0.4, wspace=0.3)
plt.subplot(2, 2, 1) df.boxplot(column='age')
plt.title('Age Distribution') plt.subplot(2, 2, 2)
df.boxplot(column='%fat') plt.title('Body Fat
Percentage') plt.subplot(2, 2, 3)
plt.scatter(df['age'], df['%fat'], c='teal',
alpha=0.7) plt.xlabel('Age') plt.ylabel('% Fat')
plt.grid(True, linestyle='--', alpha=0.7)
plt.subplot(2, 2, 4) sm.qqplot(df['age'], line='s',
label='Age') sm.qqplot(df['%fat'], line='s',
label='% Fat') plt.legend() plt.title('Q-Q Plot
Comparison') plt.tight_layout() plt.show()
print("Statistical Summary:\n", stats.round(2))
```

OUTPUT:

```
= RESTART: C:/Users/Mayu/OneDrive/Docu
Statistical Summary:
            age    %fat
mean      46.44   28.78
median    51.00   30.70
std       13.22    9.25
>
```

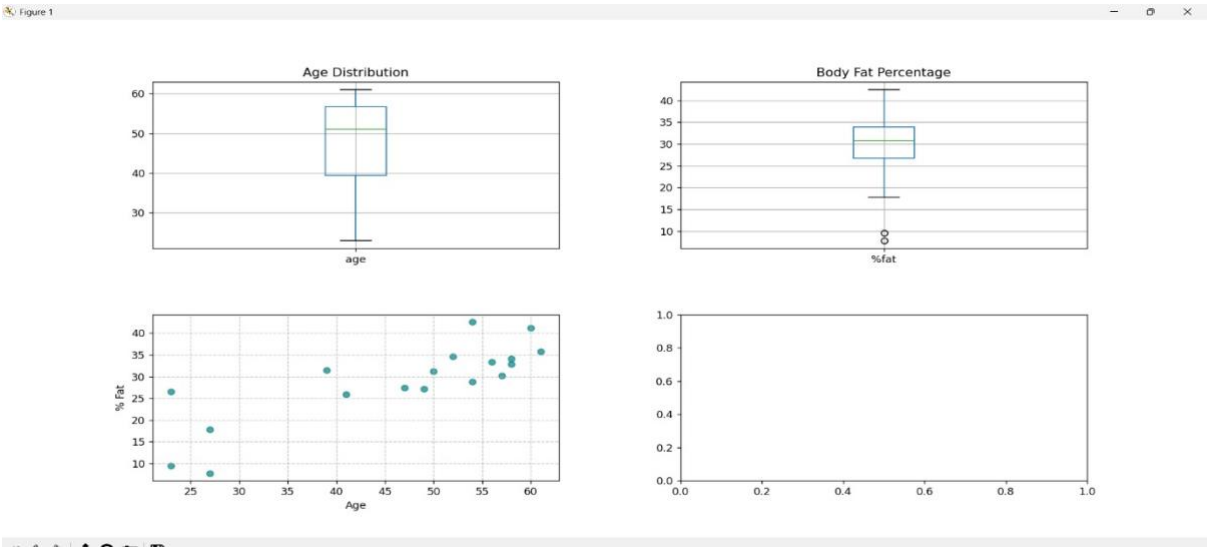


Figure 1
Age Distribution
age
Body Fat Percentage
%fat
% Fat
Age

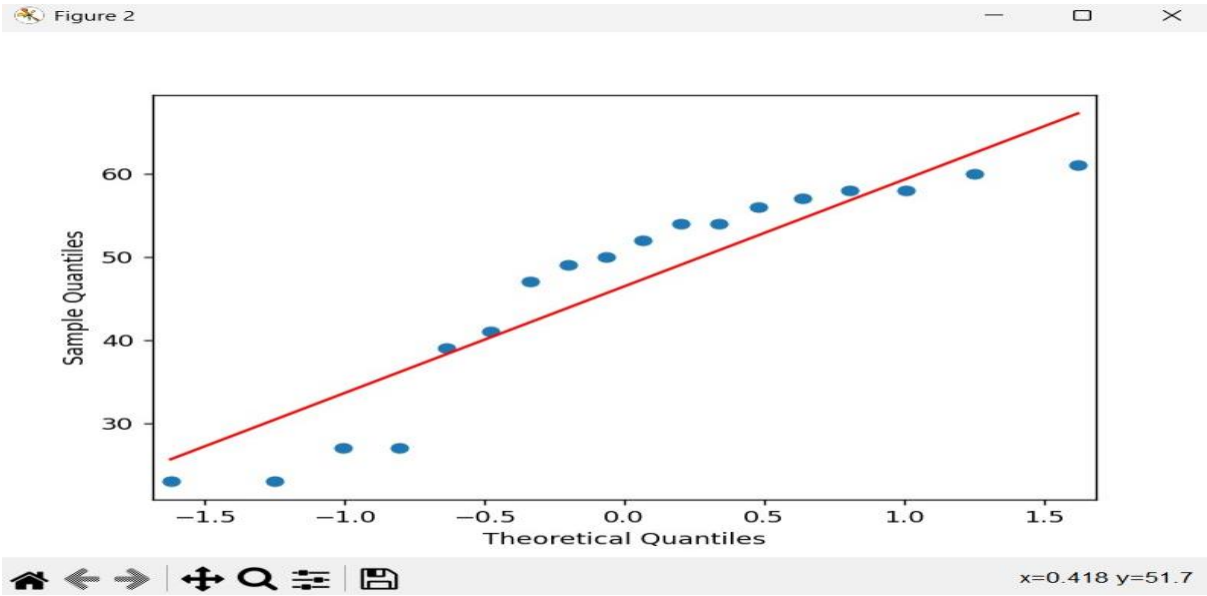


Figure 2
Sample Quantiles
Theoretical Quantiles
x=0.418 y=51.7

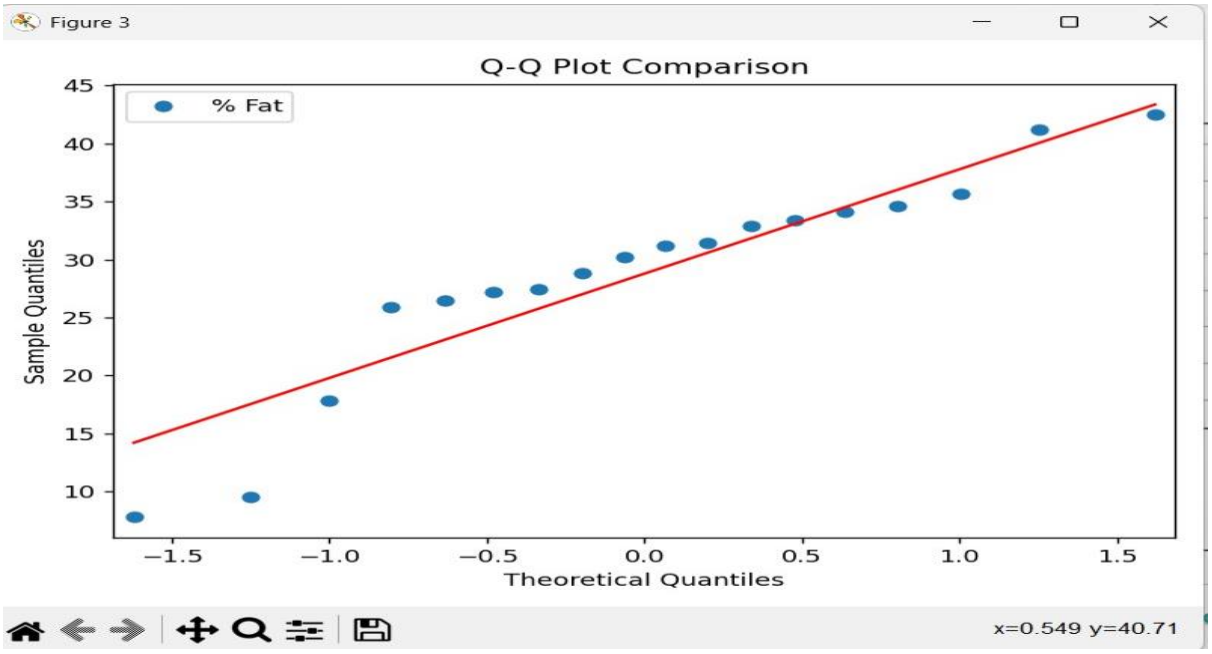


Figure 3
Q-Q Plot Comparison
% Fat
Sample Quantiles
Theoretical Quantiles
x=0.549 y=40.71

19. Sales and Profit Analysis: a) Load the "sales_data.csv" file into a Pandas data frame, which contains columns "Date," "Product," "Quantity Sold," and "Unit Price"

b) Create a new column named "Total Sales" that calculates the total sales for each transaction (Quantity Sold * Unit Price).

c) Calculate the total sales for each product and the overall profit, considering a 20% profit margin on each product. Display the top 5 most profitable products.

CODE:

```
import pandas as pd df = pd.read_csv('sales_data.csv') df['Total Sales'] = df['Quantity
Sold'] * df['Unit Price'] product_sales = df.groupby('Product').agg({'Total Sales':
'sum'}).reset_index() product_sales['Profit'] = product_sales['Total Sales'] * 0.20
overall_profit = product_sales['Profit'].sum() top_products =
product_sales.nlargest(5, 'Profit') print("Total Sales per Product:")
print(product_sales.sort_values('Total Sales', ascending=False).to_string(index=False))
print(f"\nOverall Company Profit: ${overall_profit:,.2f}") print("\nTop 5 Profitable
Products:") print(top_products[['Product', 'Profit']].to_string(index=False))
```

OUTPUT:

sales_data.csv

File Edit View

```
Date,Product,Quantity Sold,Unit Price
2023-01-01,Product A,100,20.00
2023-01-02,Product B,80,25.00
2023-01-03,Product C,50,30.00
2023-01-04,Product D,75,35.00
2023-01-05,Product A,120,20.00
2023-01-06,Product B,90,25.00
2023-01-07,Product C,60,30.00
2023-01-08,Product D,85,35.00
2023-01-09,Product A,110,20.00
2023-01-10,Product B,75,25.00
```

```
>
= RESTART: C:/Users/Mayu/OneDrive/Documents/Desktop/FC
Total Sales per Product:
  Product  Total Sales  Profit
Product A        6600.0  1320.0
Product B        6125.0  1225.0
Product D        5600.0  1120.0
Product C        3300.0   660.0

Overall Company Profit: $4,325.00

Top 5 Profitable Products:
  Product  Profit
Product A  1320.0
Product B  1225.0
Product D  1120.0
Product C   660.0
>
```

20. Customer Segmentation: a) Load "customer_data." file into a Pandas data frame, which contains "Customer ID," "Age," "Gender," and "Total Spending."

b) Segment customers into three groups based on their total spending: "High Spenders," "Medium Spenders," and "Low Spenders." Assign these segments to a new column in the data frame.

c) Calculate the average age of customers in each spending segment.

CODE:

```
import pandas as pd df =

pd.read_csv('customer_data.csv') quantiles =

df['Total Spending'].quantile([0.33, 0.67])

df['Spending Segment'] = pd.cut(df['Total Spending'], bins=[-1, quantiles[0.33],
quantiles[0.67], float('inf')],                    labels=['Low Spenders', 'Medium
Spenders', 'High Spenders'])

avg_age = df.groupby('Spending Segment')['Age'].mean()

print("Customer Segmentation:") print(df[['Customer

ID', 'Spending Segment']]) print("\nAverage Age per

Spending Segment:") print(avg_age) print("\nData

Quality Checks:")

df.info()

print("\nMissing                    Values:")

print(df.isna().sum())    print("\nGender
```

```
Distribution:")

print(df['Gender'].value_counts())

print("\nSpending Segment Statistics:")

print(df.groupby('Spending Segment', observed=True)['Total Spending'].agg(['mean', 'median', 'std']))
```

OUTPUT:

customer_data.csv

File Edit View

```
Customer ID,Age,Gender,Total Spending
1,35,Female,100
2,42,Male,200
3,50,Female,300
4,28,Male,75
5,55,Female,350
6,38,Female,180
7,45,Male,225
8,32,Female,90
9,51,Male,280
10,40,Female,160
```

```
Customer Segmentation:
   Customer ID Spending Segment
0            1     Low Spenders
1            2  Medium Spenders
2            3    High Spenders
3            4     Low Spenders
4            5    High Spenders
5            6  Medium Spenders
6            7  Medium Spenders
7            8     Low Spenders
8            9    High Spenders
9           10  Medium Spenders

Average Age per Spending Segment:
Spending Segment
Low Spenders       31.666667
Medium Spenders    41.250000
High Spenders      52.000000
Name: Age, dtype: float64

Data Quality Checks:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
```

```
Data columns (total 5 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Customer ID        10 non-null     int64
 1   Age                10 non-null     int64
 2   Gender             10 non-null     object
 3   Total Spending     10 non-null     int64
 4   Spending Segment   10 non-null     category
dtypes: category(1), int64(3), object(1)
memory usage: 594.0+ bytes

Missing Values:
Customer ID          0
Age                  0
Gender               0
Total Spending       0
Spending Segment     0
dtype: int64

Gender Distribution:
Gender
Female    6
Male      4
Name: count, dtype: int64
```

```
Spending Segment Statistics:
                          mean  median        std
Spending Segment
Low Spenders         88.333333    90.0  12.583057
Medium Spenders     191.250000   190.0  27.801379
High Spenders       310.000000   300.0  36.055513
```