Name: Mohan Krishna Balaji

Reg No:192324221

Course Code: CSA0406

Course Name: Fundamental Data Science

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?

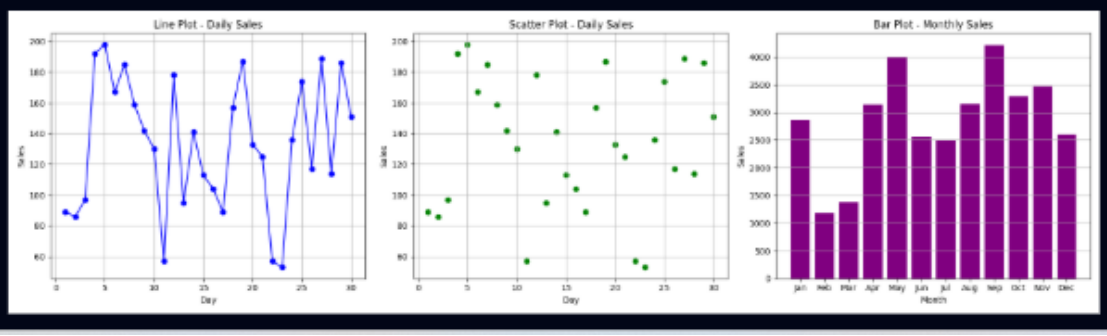
3. Develop a Python program to create a bar plot of the monthly sales data.

You have unsaved changes. To share your code, click the "Fork" button to generate a permanent link.

```python
import matplotlib.pyplot as plt
import numpy as np

# Example data for 30 days (daily sales)
days = np.arange(1, 31)
daily_sales = np.random.randint(50, 200, size=30)  # Replace with real daily data

# Example data for 12 months (monthly sales)
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
monthly_sales = np.random.randint(1000, 5000, size=12)  # Replace with real monthly data

# Create a figure with subplots
plt.figure(figsize=(18, 5))

# Line Plot
plt.subplot(1, 3, 1)
plt.plot(days, daily_sales, marker='o', linestyle='-', color='blue')
plt.title("Line Plot - Daily Sales")
plt.xlabel("Day")
plt.ylabel("Sales")
plt.grid(True)

# Scatter Plot
plt.subplot(1, 3, 2)
plt.scatter(days, daily_sales, color='green')
plt.title("Scatter Plot - Daily Sales")
plt.xlabel("Day")
plt.ylabel("Sales")
plt.grid(True)

# Bar Plot
plt.subplot(1, 3, 3)
plt.bar(months, monthly_sales, color='purple')
plt.title("Bar Plot - Monthly Sales")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.grid(axis='y')

# Layout and show all plots
plt.tight_layout()
plt.show()
```

Click Run or press shift + ENTER to run code

Enable code completions



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:

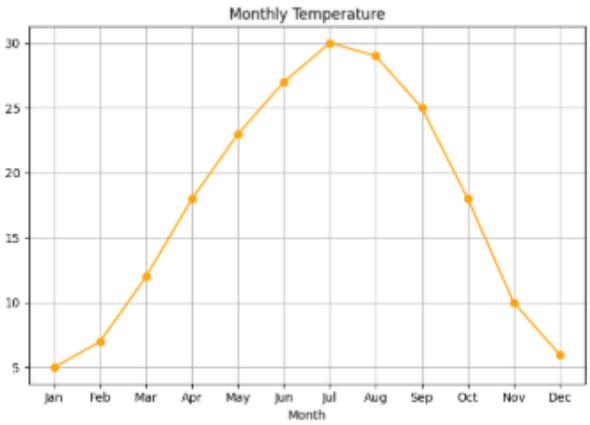
1. Develop a Python program to create a line plot of the monthly temperature data.

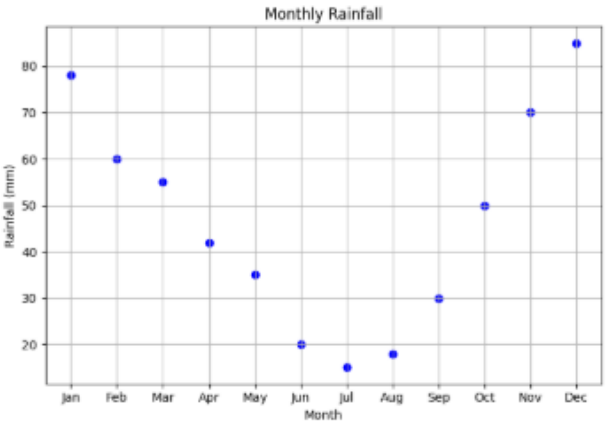
2: Develop a Python program to create a scatter plot of the monthly rainfall data.

```python
import matplotlib.pyplot as plt
import numpy as np

# Sample data for 12 months
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

# Example temperature (°C) and rainfall (mm) data
temperature = [5, 7, 12, 18, 23, 27, 30, 29, 25, 18, 10, 6]     # Replace with actual ter
rainfall = [78, 60, 55, 42, 35, 20, 15, 18, 30, 50, 70, 85]     # Replace with actual rai

# Create figure and subplots
plt.figure(figsize=(14, 5))

# 1. Line Plot - Monthly Temperature
plt.subplot(1, 2, 1)
plt.plot(months, temperature, marker='o', color='orange', linestyle='-')
plt.title("Monthly Temperature")
plt.xlabel("Month")
plt.ylabel("Temperature (°C)")
plt.grid(True)

# 2. Scatter Plot - Monthly Rainfall
plt.subplot(1, 2, 2)
plt.scatter(months, rainfall, color='blue')
plt.title("Monthly Rainfall")
plt.xlabel("Month")
plt.ylabel("Rainfall (mm)")
plt.grid(True)

# Layout adjustment and display
plt.tight_layout()
plt.show()
```

ck Run or press shift + ENTER to run code

Enable code completions

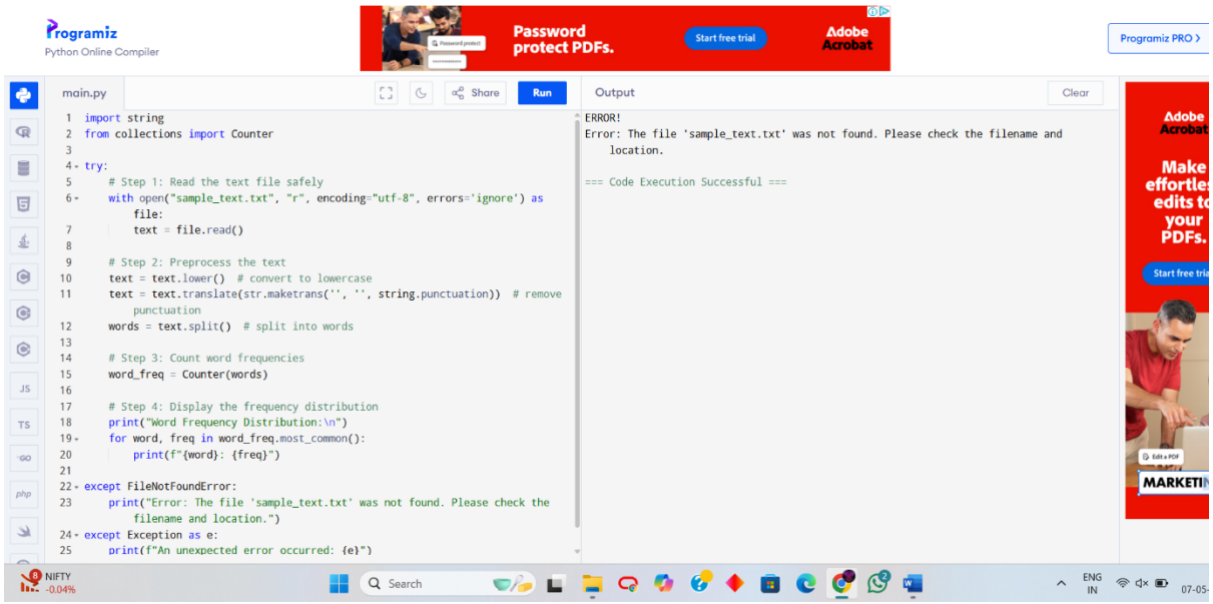




13. **Scenario:** You are working on a text analysis project and need to determine the frequency distribution of words in a given text document. You have a text document named "sample_text.txt"

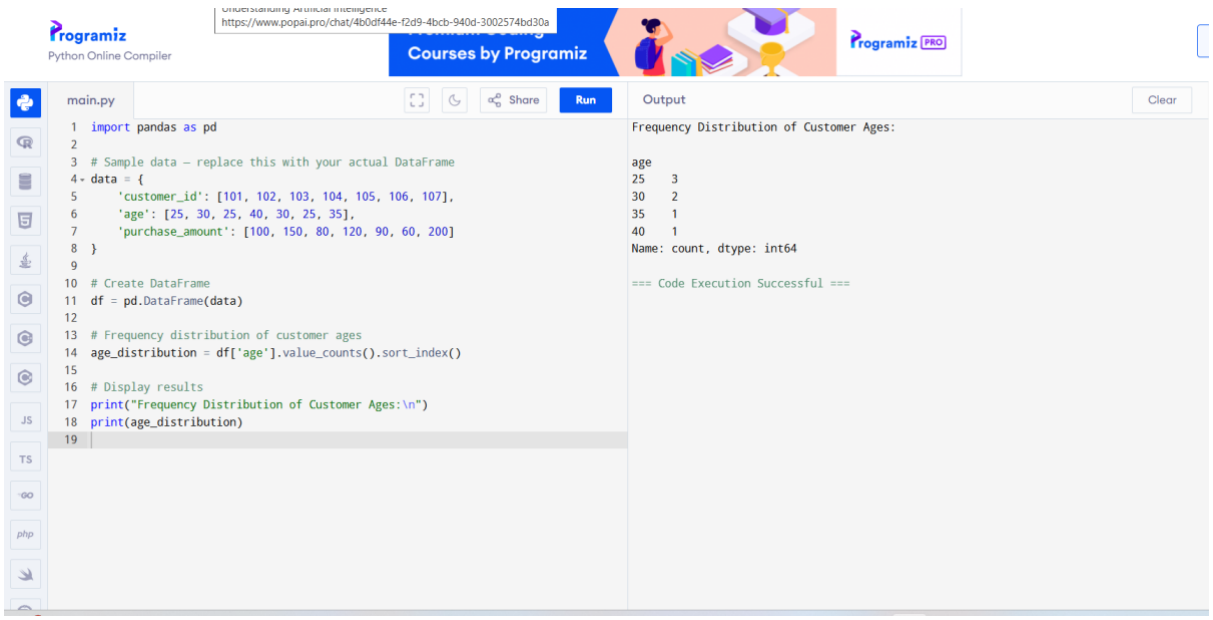
containing a paragraph of text. Your task is to develop a Python program that reads the text document, processes the text, and generates a frequency distribution of the words.

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



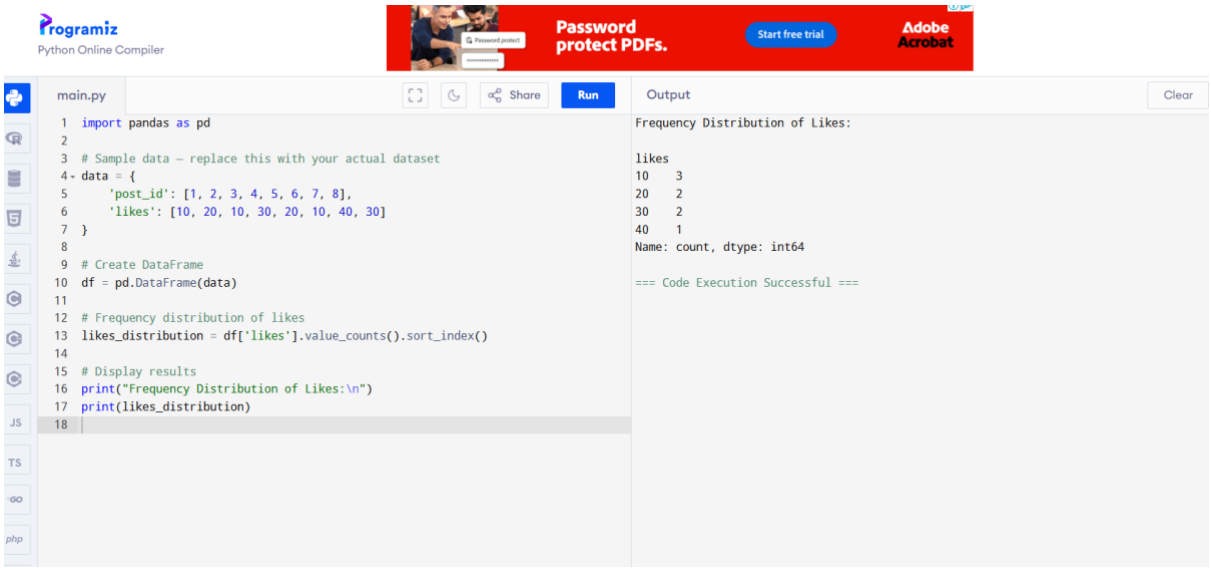
14. **Scenario**: You are a data analyst working for a company that sells products online. You have been tasked with analyzing the sales data for the past month. The data is stored in a Pandas data frame.

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.



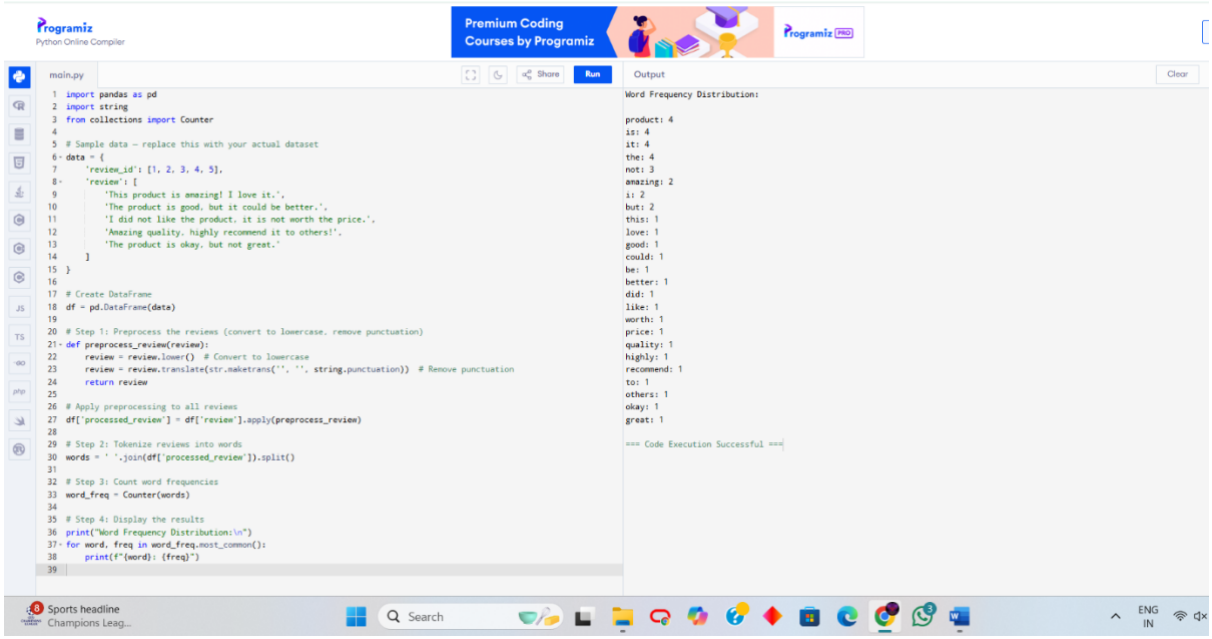
15. **Scenario:** You are a data analyst working for a social media platform. As part of your analysis, you have a dataset containing user interaction data, including the number of likes received by each post. Your task is to develop a Python program that calculates the frequency distribution of likes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

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



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?

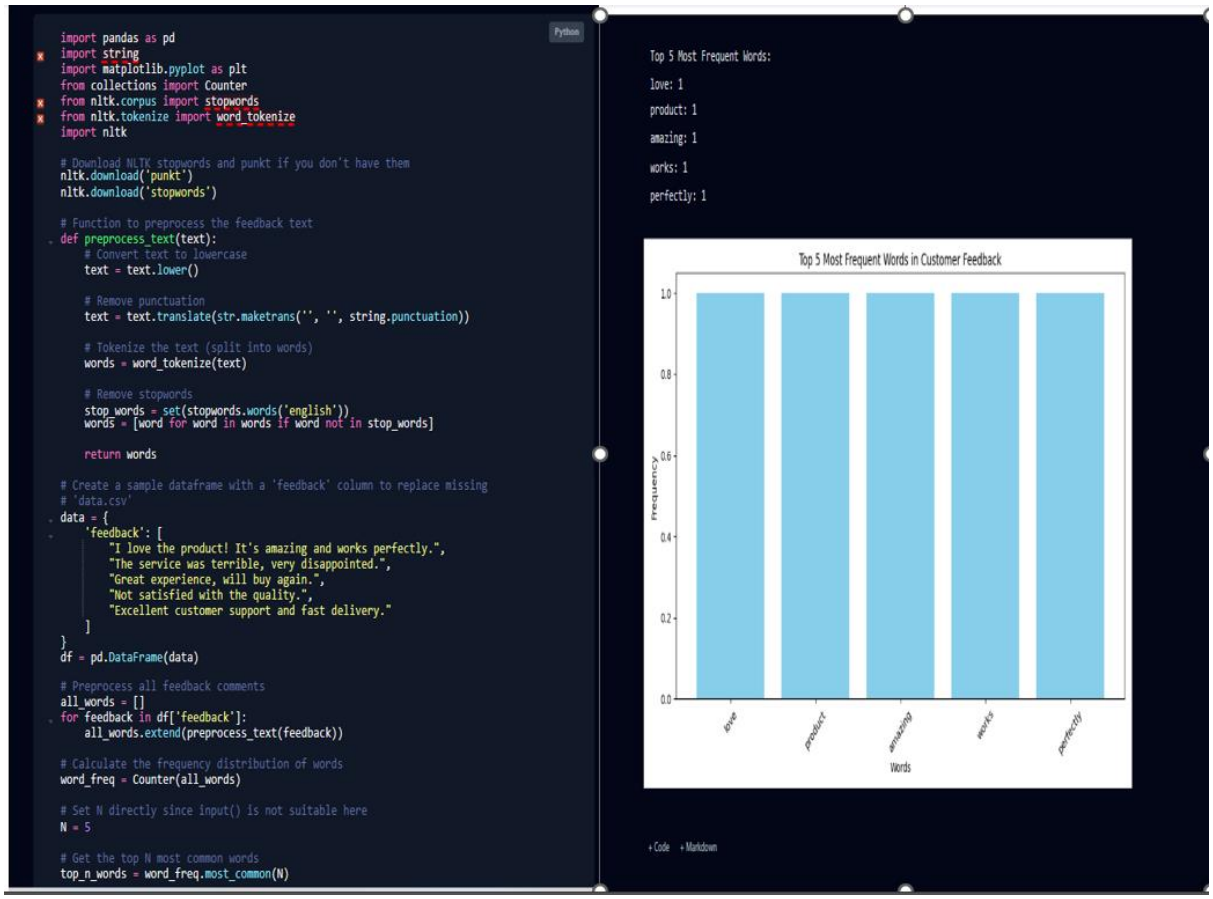


17. **Scenario:** You are a data analyst working for a marketing research company. Your team has collected a large dataset containing customer feedback from various social media platforms. The dataset consists of thousands of text entries, and your task is to develop a Python program to

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.



```python
import pandas as pd
import string
import matplotlib.pyplot as plt
from collections import Counter
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import nltk

# Download NLTK stopwords and punkt if you don't have them
nltk.download('punkt')
nltk.download('stopwords')

# Function to preprocess the feedback text
def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()

    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    # Tokenize the text (split into words)
    words = word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    words = [word for word in words if word not in stop_words]

    return words

# Create a sample dataframe with a 'feedback' column to replace missing
# 'data.csv'
data = {
    'feedback': [
        "I love the product! It's amazing and works perfectly.",
        "The service was terrible, very disappointed.",
        "Great experience, will buy again.",
        "Not satisfied with the quality.",
        "Excellent customer support and fast delivery."
    ]
}
df = pd.DataFrame(data)

# Preprocess all feedback comments
all_words = []
for feedback in df['feedback']:
    all_words.extend(preprocess_text(feedback))

# Calculate the frequency distribution of words
word_freq = Counter(all_words)

# Set N directly since input() is not suitable here
N = 5

# Get the top N most common words
top_n_words = word_freq.most_common(N)
```

```
Top 5 Most Frequent Words:
love: 1
product: 1
amazing: 1
works: 1
perfectly: 1
```


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

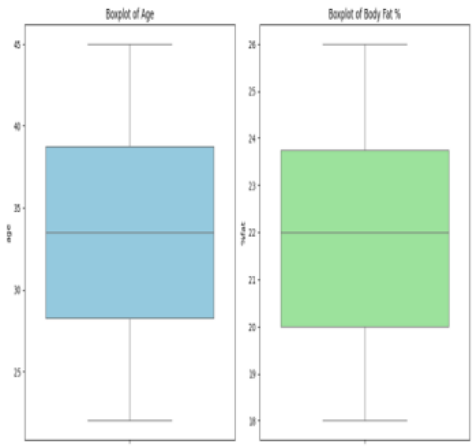
age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

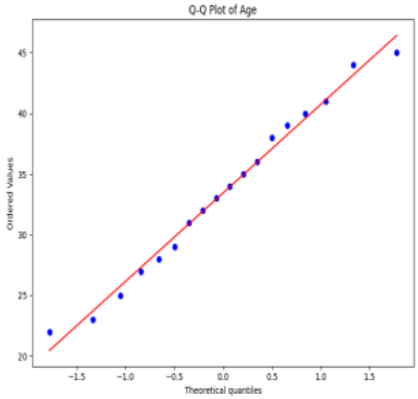
Question:

- 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.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats

# Sample data (age and body fat % of 18 randomly selected adults)
data = {
    'age': [23, 25, 31, 40, 22, 35, 41, 28, 39, 32, 27, 34, 44, 33, 38, 29, 45, 36],
    '%fat': [18, 20, 22, 25, 19, 24, 21, 23, 26, 22, 19, 21, 26, 22, 20, 24, 25, 23]
}

# Create DataFrame
df = pd.DataFrame(data)

# Calculate the mean, median, and standard deviation
age_mean = df['age'].mean()
age_median = df['age'].median()
age_std = df['age'].std()

fat_mean = df['%fat'].mean()
fat_median = df['%fat'].median()
fat_std = df['%fat'].std()

# Display the results
print(f"Age - Mean: {age_mean}, Median: {age_median}, Std: {age_std}")
print(f"Body Fat % - Mean: {fat_mean}, Median: {fat_median}, Std: {fat_std}")

# Draw boxplots for Age and %Fat
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.boxplot(data=df['age'], color='skyblue')
plt.title('Boxplot of Age')

plt.subplot(1, 2, 2)
sns.boxplot(data=df['%fat'], color='lightgreen')
plt.title('Boxplot of Body Fat %')

plt.tight_layout()
plt.show()

# Scatter plot of Age vs %Fat
plt.figure(figsize=(8, 6))
plt.scatter(df['age'], df['%fat'], color='purple')
plt.title('Scatter Plot of Age vs Body Fat %')
plt.xlabel('Age')
plt.ylabel('Body Fat %')
plt.grid(True)
plt.show()

# Q-Q plot for Age
plt.figure(figsize=(8, 6))
stats.probplot(df['age'], dist="norm", plot=plt)
plt.title('Q-Q Plot of Age')
plt.show()

# Q-Q plot for Body Fat %
plt.figure(figsize=(8, 6))
stats.probplot(df['%fat'], dist="norm", plot=plt)
plt.title('Q-Q Plot of Body Fat %')
plt.show()
```

OUT PUT:





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.

main.py

```python
import pandas as pd

# Load the "sales_data.csv" into a Pandas DataFrame
try:
    df = pd.read_csv('sales_data.csv')
except FileNotFoundError:
    print("Error: The file 'sales_data.csv' was not found. Please check the file path.")
    exit()

# Check the first few rows of the data
print(df.head())

# Step 1: Create a new column "Total Sales"
df['Total Sales'] = df['Quantity Sold'] * df['Unit Price']

# Step 2: Calculate the profit by applying a 20% profit margin on the "Total Sales"
df['Profit'] = df['Total Sales'] * 0.20

# Step 3: Calculate the total sales and profit for each product
product_sales_profit = df.groupby('Product').agg(
    total_sales=('Total Sales', 'sum'),
    total_profit=('Profit', 'sum')
).reset_index()

# Step 4: Sort the products by total profit and display the top 5 most profitable products
top_5_profitable_products = product_sales_profit.sort_values(by='total_profit', ascending
    =False).head(5)

# Display the top 5 most profitable products
print("\nTop 5 Most Profitable Products:")
print(top_5_profitable_products)
```

Output

```
ERROR!
Error: The file 'sales_data.csv' was not found. Please check the file path.

=== Code Execution Successful ===
```

https://www.programiz.com

20. Customer Segmentation: a) Load the "customer_data.csv" file into a Pandas data frame, which contains columns "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.

main.py

```python
import pandas as pd

# Load the "customer_data.csv" into a Pandas DataFrame
try:
    df = pd.read_csv('customer_data.csv')
except FileNotFoundError:
    print("Error: The file 'customer_data.csv' was not found. Please check the file path.")
    exit()

# Check the first few rows of the data
print(df.head())

# Step 1: Segment customers based on Total Spending
# Let's define the thresholds for segmentation: Low, Medium, High spenders
spending_thresholds = df['Total Spending'].quantile([0.33, 0.66])

# Function to segment based on Total Spending
def categorize_spending(row):
    if row['Total Spending'] <= spending_thresholds[0.33]:
        return 'Low Spender'
    elif row['Total Spending'] <= spending_thresholds[0.66]:
        return 'Medium Spender'
    else:
        return 'High Spender'

# Apply the function to create a new column 'Spending Segment'
df['Spending Segment'] = df.apply(categorize_spending, axis=1)

# Step 2: Calculate the average age of customers in each spending segment
average_age_per_segment = df.groupby('Spending Segment')['Age'].mean().reset_index()

# Display the average age for each spending segment
print("\nAverage Age of Customers in Each Spending Segment:")
print(average_age_per_segment)
```

Output

```
ERROR!
Error: The file 'customer_data.csv' was not found. Please check the file path.

=== Code Execution Successful ===
```