**11. Sales Over Time (Line, Scatter, Bar Plot)**

**1. Line Plot**

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('1. Monthly Sales (Line Plot)')

plt.xlabel('Month')

plt.ylabel('Sales')

plt.grid(True)

plt.show()

**2. Scatter Plot**

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()

**3. Bar Plot**

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. Temperature & Rainfall Visualization**

**1. Line Plot – Temperature**

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()

**2. Scatter Plot – Rainfall**

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. Word Frequency from Text File**

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}")

**14. Frequency of Customer Ages**

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)

**Output:**

CopyEdit

22 2

25 3

30 2

40 1

**15. Frequency of Likes on Posts**

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)

**16. Word Frequency in Customer Reviews**

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)

from collections import Counter

import string

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)

**17. Word Frequency with Stopword Removal & Plotting**

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') # file must have a 'feedback' column

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}")

# Plotting

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()

**18. Age and Body Fat Analysis**

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)

# Summary

stats = df.agg(['mean', 'median', 'std'])

plt.figure(figsize=(15, 10))

plt.subplots\_adjust(hspace=0.4, wspace=0.3)

# Boxplots

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')

# Scatter

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)

# Q-Q Plots

plt.subplot(2, 2, 4)

sm.qqplot(df['age'], line='s')

plt.title('Q-Q Plot: Age')

plt.tight\_layout()

plt.show()

print("Statistical Summary:\n", stats.round(2))

**19. Sales and Profit Analysis**

df = pd.read\_csv('sales\_data.csv') # must have Quantity Sold, Unit Price, Product

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))

**20. Customer Segmentation**

df = pd.read\_csv('customer\_data.csv') # must contain Age, Gender, Total Spending

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']))

**21. Data Cleaning and Transformation**

**Code**

import pandas as pd

# Load the data

employee\_df = pd.read\_csv("employee\_data.csv")

# Convert Salary to numeric (remove $ and commas)

employee\_df['Salary'] = pd.to\_numeric(employee\_df['Salary'].replace('[\$,]', '', regex=True), errors='coerce')

# Remove rows with missing Department

employee\_df = employee\_df.dropna(subset=["Department"])

# Extract First Name

employee\_df['First Name'] = employee\_df['Full Name'].str.split().str[0]

# Show the result

print(employee\_df.head())

**Output**

Employee ID Full Name Department Salary First Name

0 101 John Carter Sales 58000 John

1 102 Emily Johnson Marketing 62000 Emily

2 103 Raj Kapoor IT Support 55000 Raj

3 104 Alice Brown Human Resources 59000 Alice

4 105 Mark Turner Finance 67000 Mark

**22. Time Series Analysis**

**Code**

import pandas as pd

import matplotlib.pyplot as plt

# Load data

temp\_df = pd.read\_csv("temperature\_data.csv")

# Convert Date to datetime

temp\_df['Date'] = pd.to\_datetime(temp\_df['Date'])

# Extract month

temp\_df['Month'] = temp\_df['Date'].dt.to\_period('M')

# Monthly average

monthly\_avg\_temp = temp\_df.groupby('Month')['Temperature (Celsius)'].mean().reset\_index()

monthly\_avg\_temp['Month'] = monthly\_avg\_temp['Month'].dt.to\_timestamp()

# Plot

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

plt.plot(temp\_df['Date'], temp\_df['Temperature (Celsius)'], label='Daily Temp')

plt.plot(monthly\_avg\_temp['Month'], monthly\_avg\_temp['Temperature (Celsius)'], color='red', label='Monthly Avg')

plt.title('Temperature Trend Over Time')

plt.xlabel('Date')

plt.ylabel('Temperature (Celsius)')

plt.legend()

plt.grid(True)

plt.tight\_layout()

plt.show()

**23. Joining DataFrames & Order Time Analysis**

**Code**

import pandas as pd

# Load both files

orders\_df = pd.read\_csv("order\_data.csv")

customers\_df = pd.read\_csv("customer\_info.csv")

# Merge on Customer ID

merged\_df = pd.merge(orders\_df, customers\_df, on="Customer ID", how="inner")

print("Merged Data:")

print(merged\_df.head())

# Convert Order Date to datetime

merged\_df['Order Date'] = pd.to\_datetime(merged\_df['Order Date'])

# Sort by Customer and Order Date

merged\_df = merged\_df.sort\_values(by=['Customer ID', 'Order Date'])

# Time difference between consecutive orders

merged\_df['Time Diff'] = merged\_df.groupby('Customer ID')['Order Date'].diff()

# Average time between orders

avg\_time\_diff = merged\_df['Time Diff'].dropna().mean()

print("\nAverage time between consecutive orders:")

print(avg\_time\_diff)

**Output**

Merged Data:

Order ID Customer ID Order Date Name Email Phone Number

0 501 100 2023-01-10 John Smith john@email.com 123-456-7890

1 502 100 2023-01-20 John Smith john@email.com 123-456-7890