

## DAY 3 LAB EXPERIMENTS

NAME: LARAVIND

Reg No : 192424080

Date : 18/12/2025

**EXP\_11** You have been tasked with creating a simple plot to show the sales of a product over time.

```
DAY 3 LAB EXPERIMENTS
File Edit View Insert Runtime Tools Help
Commands + Code + Text + Run all

Files
- sample_data
- Tweets.csv
- ecommerce customers.csv

EXP_11 A Data Scientist working for a Company that sells products online.

1
import pandas as pd
import matplotlib.pyplot as plt

2
df=pd.read_csv("monthly_sales.csv",encoding="latin1")
print(df.head())

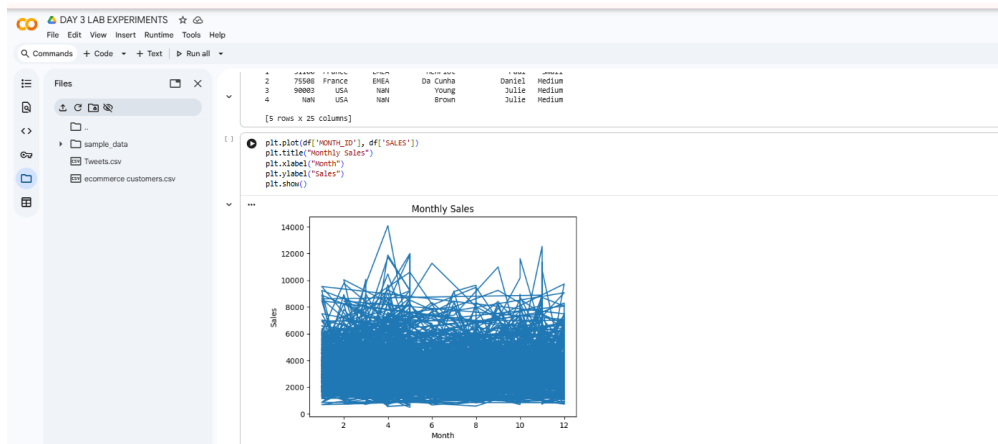
3
ORDERNUMBER QUANTITYORDERED PRICEEACH ORDERLINENUMBER SALES \
0 38587 30 95.78 2 2871.00
1 38121 34 81.35 5 2765.90
2 38134 41 94.74 2 3884.34
3 38145 45 83.26 6 3748.70
4 38159 49 398.00 14 15851.27

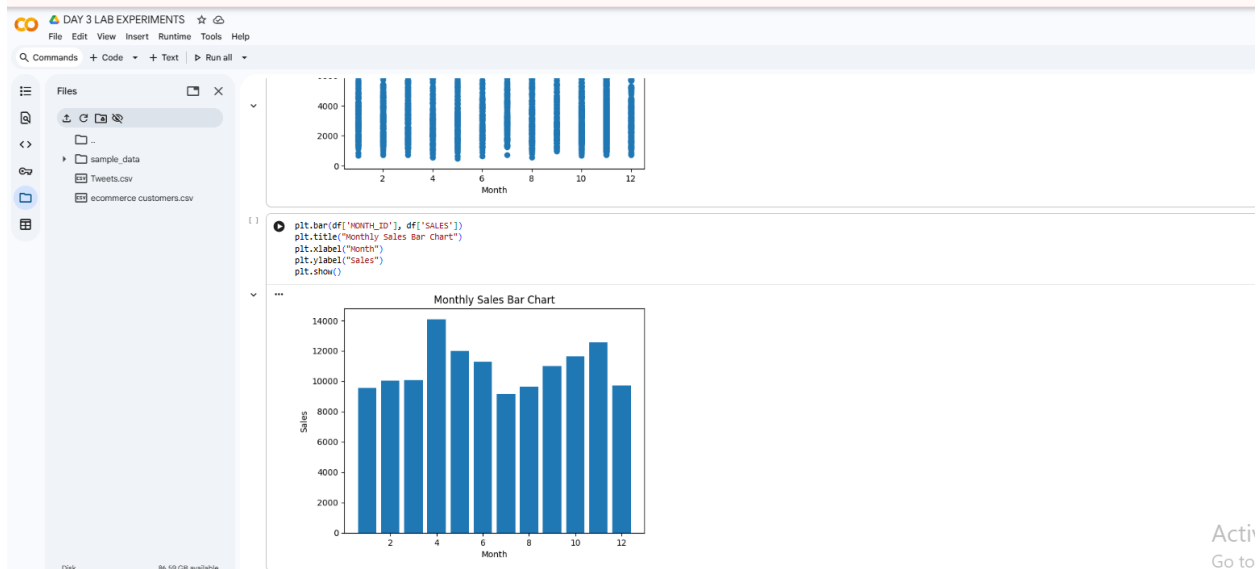
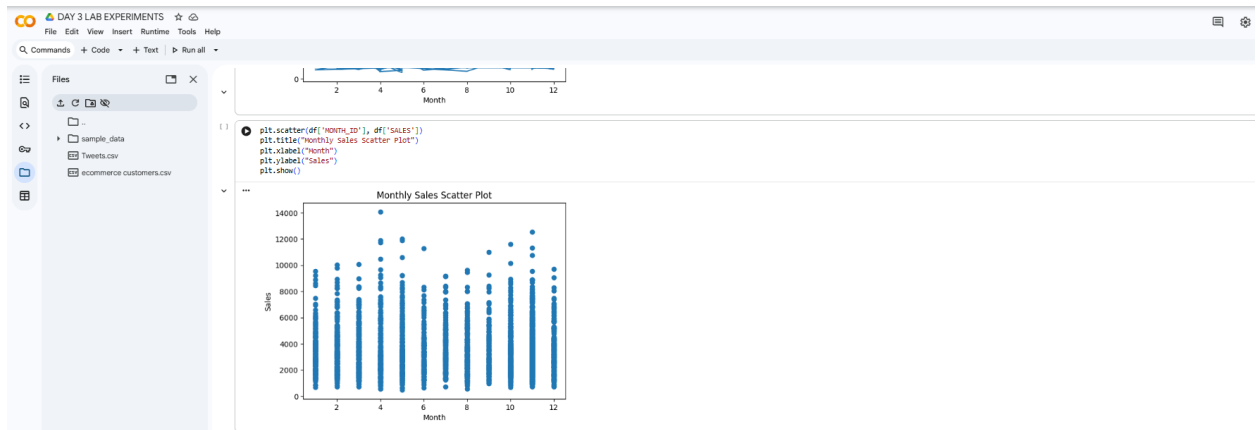
ORDERDATE STATUS QTR_ID MONTH_ID YEAR_ID ... \
0 2/24/2003 0:00 Shipped 1 2 2003 ...
1 5/17/2003 0:00 Shipped 2 5 2003 ...
2 7/11/2003 0:00 Shipped 3 7 2003 ...
3 8/15/2003 0:00 Shipped 3 8 2003 ...
4 10/10/2003 0:00 Shipped 4 10 2003 ...

ADDRESSLINE1 ADDRESSLINE2 CITY STATE \
0 887 Long Airport Avenue NaN NYC NY
1 59 rue de l'Abbaye NaN Paris NaN
2 27 rue du Colonel Pierre Aude NaN Paris NaN
3 78334 Hillside Dr. NaN Pasadena CA
4 7734 Strong St. NaN San Francisco CA

POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEASIZE
0 98021 USA NaN Kari Small
1 51188 France EMEA Henriot Paul Small
2 75585 France EMEA Da Cunha Daniel Medium
3 98083 USA NaN Young Julie Medium
4 NaN USA NaN Brown Julie Medium

[5 rows x 25 columns]
```





Activ  
Go to

## EXP\_12 To Develop a Python program that generates line plots and scatter plots to visualize the temperature and rainfall data.

DAY 3 LAB EXPERIMENTS

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- sample\_data
- Tweets.csv
- ecommerce customers.csv

EXP\_12 A Data Analysis project that involves analyzing the monthly temperature and rainfall data for a city.

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

import pandas as pd

df = pd.read_csv(
    "weather_data.csv",
    encoding="latin1",
    on_bad_lines="skip"
)

print(df.head())
```

	Location	Date_Time	Temperature_C	Humidity_pct
0	San Diego	2024-01-14 21:12:46	10.683001	41.195754
1	San Diego	2024-05-17 15:12:10	8.73414	58.319107
2	San Diego	2024-05-11 09:30:59	11.632436	38.820175
3	Philadelphia	2024-02-26 17:32:39	-8.628976	54.074474
4	San Antonio	2024-04-29 13:23:51	39.886213	72.899908

```
Precipitation_mm wind_Speed_kmh
0      4.400119      8.233540
1      9.113623     27.715161
2      4.607511     28.732951
3      3.18372      26.367305
4      9.590282     29.899622

/tmp/ipython-input-1235356151.py:3: DtypeWarning: Columns (2,3,4) have mixed types. Specify dtype option on import or set low_memory=False.
df = pd.read_csv(

df["Date_Time"] = pd.to_datetime(df["Date_Time"], errors="coerce")
df["Month"] = df["Date_Time"].dt.month
```

DAY 3 LAB EXPERIMENTS

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- sample\_data
- Tweets.csv
- ecommerce customers.csv

```
df["Date_Time"] = pd.to_datetime(df["Date_Time"], errors="coerce")
df["Month"] = df["Date_Time"].dt.month

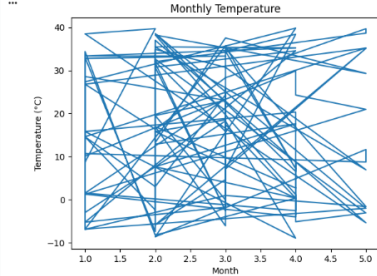
df["Date_Time"] = pd.to_datetime(df["Date_Time"], errors="coerce")
df["Month"] = df["Date_Time"].dt.month

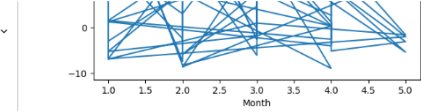
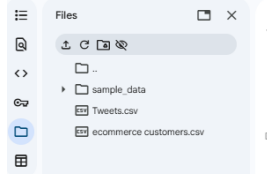
df["Temperature_C"] = pd.to_numeric(df["Temperature_C"], errors="coerce")

df_small = df.head(100)

plt.plot(df_small["Month"], df_small["Temperature_C"])
plt.xlabel("Month")
plt.ylabel("Temperature (°C)")
plt.title("Monthly Temperature")
plt.show()
```

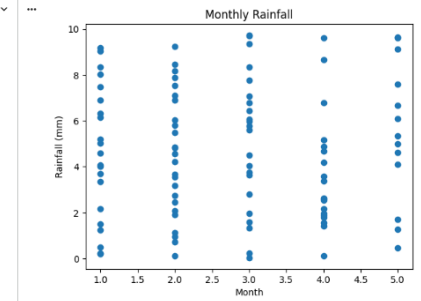
Monthly Temperature



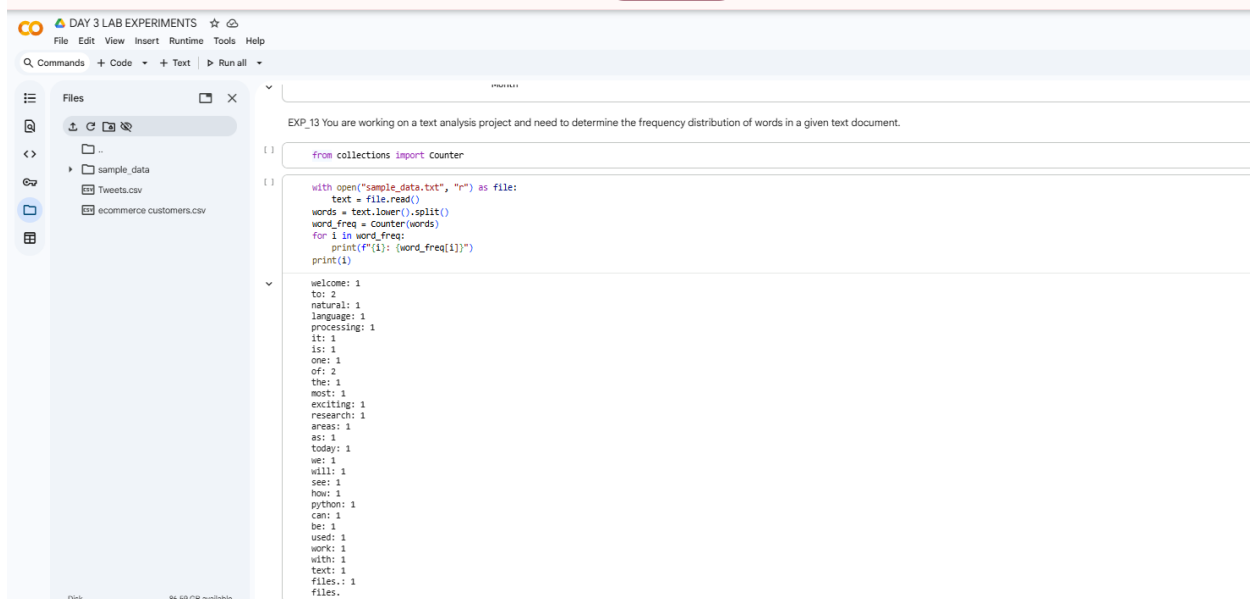


```
df["Precipitation_mm"] = pd.to_numeric(df["Precipitation_mm"], errors="coerce")

plt.scatter(df_small["Month"], df_small["Precipitation_mm"])
plt.xlabel("Month")
plt.ylabel("Rainfall (mm)")
plt.title("Monthly Rainfall")
plt.show()
```



## EXP\_13 To Develop a Python program that reads the text document, processes the text, and generates a frequency distribution of the words.



The screenshot shows a Jupyter Notebook environment with a file explorer on the left and a code editor on the right. The file explorer shows a directory with files: sample\_data, Tweets.csv, and ecommerce customers.csv. The code editor contains the following Python code:

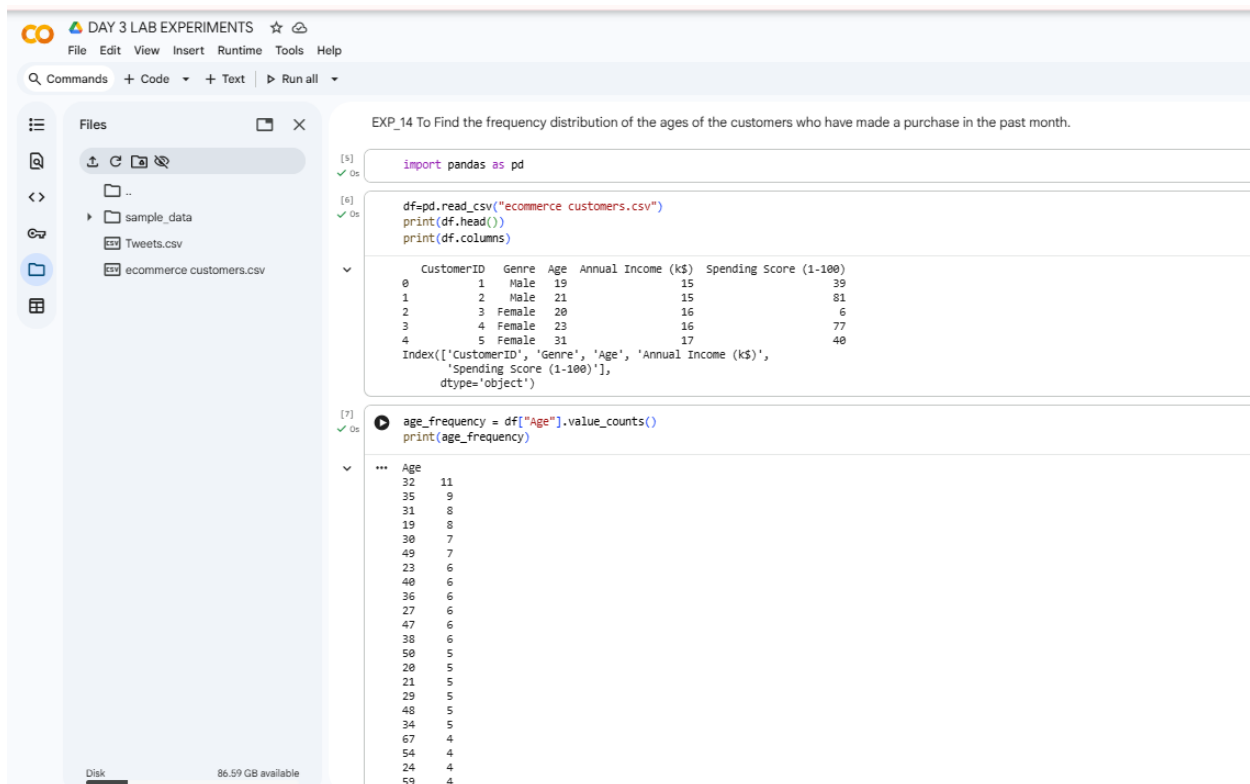
```
from collections import Counter

with open("sample_data.txt", "r") as file:
    text = file.read()
    words = text.lower().split()
    word_freq = Counter(words)
    for i in word_freq:
        print(f"{i}: {word_freq[i]}")
    print()
```

The output of the code is a list of words and their frequencies:

```
welcome: 1
to: 2
natural: 1
language: 1
processing: 1
it: 1
is: 1
one: 1
of: 2
the: 1
most: 1
exciting: 1
research: 1
areas: 1
as: 1
today: 1
we: 1
will: 1
see: 1
how: 1
python: 1
can: 1
be: 1
used: 1
work: 1
with: 1
text: 1
files: 1
files:
```

## EXP\_14 To 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.



The screenshot shows a Jupyter Notebook environment with a file explorer on the left and a code editor on the right. The file explorer shows a directory with files: sample\_data, Tweets.csv, and ecommerce customers.csv. The code editor contains the following Python code:

```
import pandas as pd

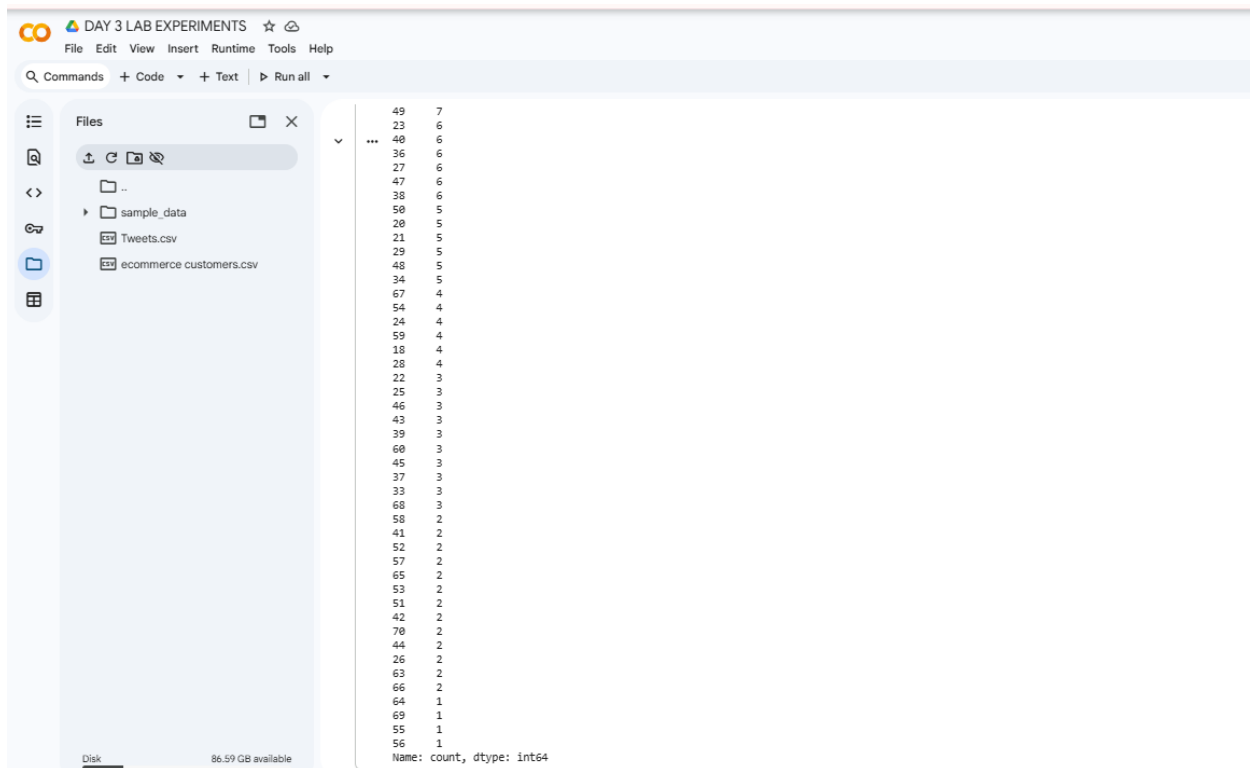
df = pd.read_csv("ecommerce customers.csv")
print(df.head())
print(df.columns)
```

The output of the code is a table showing the first 5 rows of the 'ecommerce customers.csv' file:

	CustomerID	Genre	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

The output of the code is a table showing the frequency distribution of the ages of the customers who have made a purchase in the past month:

```
Age
32    11
35     9
31     8
19     8
30     7
49     7
23     6
40     6
36     6
27     6
47     6
38     6
50     5
20     5
21     5
29     5
48     5
34     5
67     4
54     4
24     4
59     4
```



**EXP\_15 To Develop a Python program that calculates the frequency distribution of likes among the posts.**

DAY 3 LAB EXPERIMENTS

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- sample\_data
- Tweets.csv
- ecommerce customers.csv

EXP\_15 To Develop a Python program that calculates the frequency distribution of likes among the posts.

```
[ ]
import pandas as pd

[9]
df=pd.read_csv("Tweets.csv",encoding="latin1")
print(df.head())

...
tweet_id airline_sentiment airline_sentiment_confidence \
0 570306133677760513 neutral 1.0000
1 570301130808122368 positive 0.3406
2 570301003672813571 neutral 0.6837
3 570301031407624196 negative 1.0000
4 570300817074462722 negative 1.0000

negative_reason negative_reason_confidence airline \
0 NaN NaN Virgin America
1 NaN 0.0000 Virgin America
2 NaN NaN Virgin America
3 Bad Flight 0.7033 Virgin America
4 Can't Tell 1.0000 Virgin America

airline_sentiment_gold name negative_reason_gold retweet_count \
0 NaN cairdin NaN 0
1 NaN jnardino NaN 0
2 NaN yvonnalynn NaN 0
3 NaN jnardino NaN 0
4 NaN jnardino NaN 0

text tweet_coord \
0 @virginAmerica what @dhepburn said. NaN
1 @virginAmerica plus you've added commercials t... NaN
2 @virginAmerica I didn't today... Must mean I n... NaN
3 @virginAmerica it's really aggressive to blast... NaN
4 @virginAmerica and it's a really big bad thing... NaN

tweet_created tweet_location user_timezone
0 2015-02-24 11:35:52 -0800 NaN Eastern Time (US & Canada)
1 2015-02-24 11:15:59 -0800 NaN Pacific Time (US & Canada)
2 2015-02-24 11:15:48 -0800 Lets Play Central Time (US & Canada)
3 2015-02-24 11:15:36 -0800 NaN Pacific Time (US & Canada)
4 2015-02-24 11:14:45 -0800 NaN Pacific Time (US & Canada)

[11]
likes_frequency = df["retweet_count"].value_counts()
```

DAY 3 LAB EXPERIMENTS

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- sample\_data
- Tweets.csv
- ecommerce customers.csv

```
[12]
print("The Frequency Distribution of Likes are:")
print(likes_frequency)

... The Frequency Distribution of Likes are:
retweet_count
0    13873
1     640
2      66
3      22
4       17
5        5
7         3
6         3
22        2
18         1
15         1
31         1
11         1
8          1
9          1
28         1
32         1
44         1
Name: count, dtype: int64
```