Assignment No-1

CODE:-

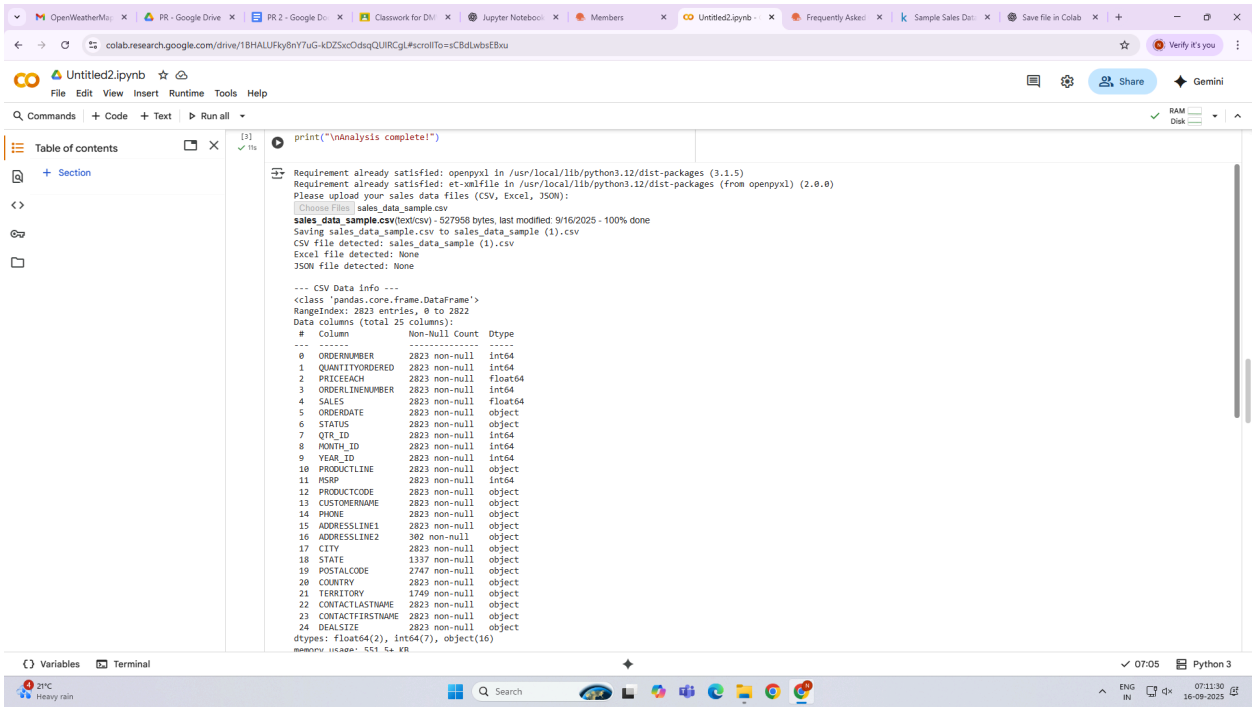
```
# Install necessary package
!pip install openpyxl

# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files

# Upload files
print("Please upload your sales data files (CSV, Excel, JSON):")
uploaded = files.upload()

# Automatically detect uploaded files by extension
csv_file = None
excel_file = None
json_file = None

for filename in uploaded.keys():
    if filename.endswith('.csv'):
        csv_file = filename
    elif filename.endswith(('.xls', '.xlsx')):
        excel_file = filename
    elif filename.endswith('.json'):
        json_file = filename

print(f"CSV file detected: {csv_file}")
print(f"Excel file detected: {excel_file}")
print(f"JSON file detected: {json_file}")

# Load files into DataFrames
df_csv = pd.read_csv(csv_file, encoding='latin1') if csv_file else
pd.DataFrame()
df_excel = pd.read_excel(excel_file) if excel_file else pd.DataFrame()
df_json = pd.read_json(json_file) if json_file else pd.DataFrame()
```

```python
# Function to explore data
def explore_data(df, name='DataFrame'):
    print(f"\n--- {name} info ---")
    print(df.info())
    print(f"\n--- {name} head ---")
    print(df.head())
    print(f"\n--- {name} missing values ---")
    print(df.isnull().sum())
    print(f"\n--- {name} duplicates --- {df.duplicated().sum()}")

# Explore all datasets
if not df_csv.empty: explore_data(df_csv, 'CSV Data')
if not df_excel.empty: explore_data(df_excel, 'Excel Data')
if not df_json.empty: explore_data(df_json, 'JSON Data')

# Function to clean data
def clean_data(df):
    if df.empty:
        return df
    # Drop duplicates
    df = df.drop_duplicates().copy()

    # Fill missing numeric with 0, categorical with mode
    for col in df.columns:
        if pd.api.types.is_numeric_dtype(df[col]):
            df[col] = df[col].fillna(0)
        else:
            if not df[col].mode().empty:
                df[col] = df[col].fillna(df[col].mode()[0])
            else:
                df[col] = df[col].fillna('Unknown')
    return df

# Clean all datasets
df_csv_clean = clean_data(df_csv)
df_excel_clean = clean_data(df_excel)
df_json_clean = clean_data(df_json)
```

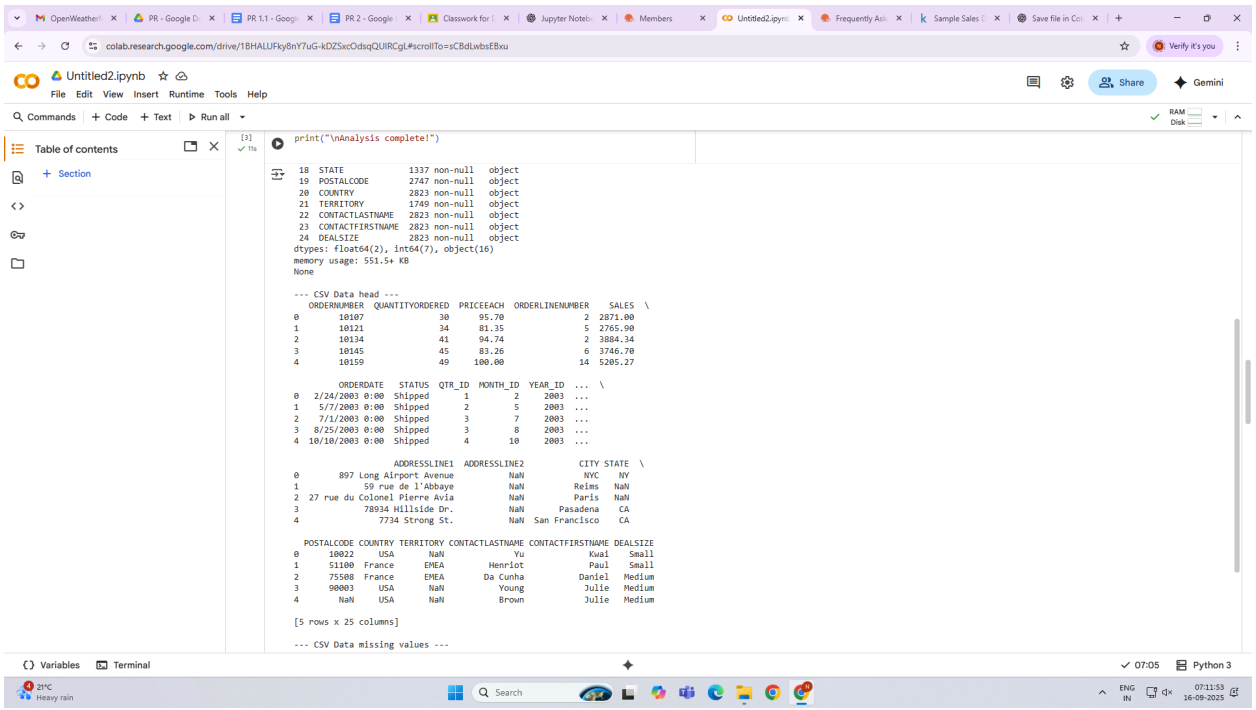
```
# Unify columns before concatenation
# Find all columns used across datasets
all_cols =
set(df_csv_clean.columns).union(set(df_excel_clean.columns)).union(set(df_
json_clean.columns))

# Make sure all DataFrames have all columns
def align_columns(df, all_cols):
    for col in all_cols:
        if col not in df.columns:
            df[col] = np.nan  # fill missing columns with NaN
    return df[sorted(all_cols)]

df_csv_aligned = align_columns(df_csv_clean, all_cols)
df_excel_aligned = align_columns(df_excel_clean, all_cols)
df_json_aligned = align_columns(df_json_clean, all_cols)

# Concatenate all data
df_all = pd.concat([df_csv_aligned, df_excel_aligned, df_json_aligned],
ignore_index=True)

print("\n--- Combined DataFrame info ---")
print(df_all.info())

# Example data transformation: create 'Total_Sales' if 'Quantity' and
'Price' exist
if 'Quantity' in df_all.columns and 'Price' in df_all.columns:
    df_all['Total_Sales'] = df_all['Quantity'].astype(float) *
df_all['Price'].astype(float)
else:
    print("Columns 'Quantity' and/or 'Price' missing, skipping
'Total_Sales' calculation.")

# Example descriptive statistics
print("\n--- Descriptive Statistics ---")
if 'Total_Sales' in df_all.columns:
    total_sales = df_all['Total_Sales'].sum()
    print(f"Total Sales: {total_sales}")
```

```
if 'OrderID' in df_all.columns and 'Total_Sales' in df_all.columns:
    avg_order_value =
df_all.groupby('OrderID')['Total_Sales'].sum().mean()
    print(f"Average Order Value: {avg_order_value}")

if 'Product_Category' in df_all.columns:
    print("\nProduct Category Distribution:")
    print(df_all['Product_Category'].value_counts())

# Visualizations
sns.set(style="whitegrid")

# Total Sales by Product Category (bar plot)
if 'Product_Category' in df_all.columns and 'Total_Sales' in
df_all.columns:
    plt.figure(figsize=(10,6))
    sns.barplot(data=df_all, x='Product_Category', y='Total_Sales',
estimator=sum, ci=None)
    plt.title('Total Sales by Product Category')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# Pie chart of Product Category distribution
if 'Product_Category' in df_all.columns:
    plt.figure(figsize=(6,6))
    df_all['Product_Category'].value_counts().plot.pie(autopct='%1.1f%%')
    plt.title('Product Category Distribution')
    plt.ylabel('')
    plt.show()

# Boxplot of Total Sales
if 'Total_Sales' in df_all.columns:
    plt.figure(figsize=(8,5))
    sns.boxplot(x=df_all['Total_Sales'])
    plt.title('Distribution of Total Sales')
    plt.show()

print("\nAnalysis complete!")
```

OUTPUT : –



```
print("\nAnalysis complete!")

Requirement already satisfied: openpyxl in /usr/local/lib/python3.12/dist-packages (3.1.5)
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.12/dist-packages (from openpyxl) (2.0.0)
Please upload your sales data files (CSV, Excel, JSON):
Choose Files sales_data_sample.csv
sales_data_sample.csv(text/csv) - 527958 bytes, last modified: 9/16/2025 - 100% done
Saving sales_data_sample.csv to sales_data_sample (1).csv
CSV file detected: sales_data_sample (1).csv
Excel file detected: None
JSON file detected: None

--- CSV Data Info ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ORDERNUMBER       2823 non-null   int64
 1   QUANTITYORDERED   2823 non-null   int64
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object
 6   STATUS            2823 non-null   object
 7   QTR_ID            2823 non-null   int64
 8   MONTH_ID          2823 non-null   int64
 9   YEAR_ID           2823 non-null   int64
 10  PRODUCTLINE       2823 non-null   object
 11  MSRP              2823 non-null   int64
 12  PRODUCTCODE       2823 non-null   object
 13  CUSTOMERNAME      2823 non-null   object
 14  PHONE             2823 non-null   object
 15  ADDRESSLINE1      2823 non-null   object
 16  ADDRESSLINE2      302 non-null    object
 17  CITY              2823 non-null   object
 18  STATE             1337 non-null   object
 19  POSTALCODE        2747 non-null   object
 20  COUNTRY           2823 non-null   object
 21  TERRITORY         1749 non-null   object
 22  CONTACTLASTNAME   2823 non-null   object
 23  CONTACTFIRSTNAME  2823 non-null   object
 24  DEALSIZE          2823 non-null   object
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```



```
print("\nAnalysis complete!")

 18  STATE             1337 non-null   object
 19  POSTALCODE        2747 non-null   object
 20  COUNTRY           2823 non-null   object
 21  TERRITORY         1749 non-null   object
 22  CONTACTLASTNAME   2823 non-null   object
 23  CONTACTFIRSTNAME  2823 non-null   object
 24  DEALSIZE          2823 non-null   object
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
None

--- CSV Data head ---
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                    ADDRESSLINE1  ADDRESSLINE2          CITY STATE  \
0          897 Long Airport Avenue           NaN           NYC    NY
1               59 rue de l'Abbaye           NaN         Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN         Paris   NaN
3                78934 Hillside Dr.           NaN      Pasadena    CA
4                   7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN             Yu             Kwai    Small
1      51100  France      EMEA        Henriot             Paul    Small
2      75508  France      EMEA       Da Cunha           Daniel   Medium
3      90003     USA       NaN          Young            Julie   Medium
4        NaN     USA       NaN          Brown            Julie   Medium

[5 rows x 25 columns]

--- CSV Data missing values ---
```