NAME: ADEYEMO OPEYEMI

ID :

ASSIGNMENT

In [2]:
```
import pandas as pd
```

In [3]:
```
data=pd.read_csv("sales_data_sample.csv",encoding="ISO-8859-1")
data
```

Out[3]:
```
      ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  OR
   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  1
 ...          ...              ...        ...              ...      ...
2818        10350               20     100.00               15  2244.40
2819        10373               29     100.00                1  3978.51
2820        10386               43     100.00                4  5417.57
2821        10397               34      62.24                1  2116.16
2822        10414               47      65.52                9  3079.44

2823 rows × 25 columns
```

In [4]:
```
data.head()
```

Out[4]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDEF
0	10107	30	95.70	2	2871.00	2/24
1	10121	34	81.35	5	2765.90	5/7
2	10134	41	94.74	2	3884.34	7/1
3	10145	45	83.26	6	3746.70	8/25
4	10159	49	100.00	14	5205.27	10/10

5 rows × 25 columns

In [5]:

```
data.shape
```

Out[5]:

```
(2823, 25)
```

In [6]:

```
data.describe()
```

Out[6]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SAL
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.0000
mean	10258.725115	35.092809	83.658544	6.466171	3553.8890
std	92.085478	9.741443	20.174277	4.225841	1841.8651
min	10100.000000	6.000000	26.880000	1.000000	482.1300
25%	10180.000000	27.000000	68.860000	3.000000	2203.4300
50%	10262.000000	35.000000	95.700000	6.000000	3184.8000
75%	10333.500000	43.000000	100.000000	9.000000	4508.0000
max	10425.000000	97.000000	100.000000	18.000000	14082.8000

SALE AND REVENUES ANALYSIS

1. What is the total revenue generated from sales

In [237...

```
total_revenue=data['SALES'].sum()
total_revenue
```

Out[237...

```
10032628.85
```

2. What are the top 5 best-selling products based on total sales?

In [17]:
```
top_5_products = data.groupby("PRODUCTCODE")["SALES"].sum().nlargest(5)
top_5_products
```

Out[17]:
```
PRODUCTCODE
S18_3232    288245.42
S10_1949    191073.03
S10_4698    170401.07
S12_1108    168585.32
S18_2238    154623.95
Name: SALES, dtype: float64
```

In [19]:
```
data['PRODUCTCODE']
```

Out[19]:
```
0       S10_1678
1       S10_1678
2       S10_1678
3       S10_1678
4       S10_1678
          ...
2818    S72_3212
2819    S72_3212
2820    S72_3212
2821    S72_3212
2822    S72_3212
Name: PRODUCTCODE, Length: 2823, dtype: object
```

In [20]:
```
data.columns
```

Out[20]:
```
Index(['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER',
       'SALES', 'ORDERDATE', 'STATUS', 'QTR_ID', 'MONTH_ID', 'YEAR_ID',
       'PRODUCTLINE', 'MSRP', 'PRODUCTCODE', 'CUSTOMERNAME', 'PHONE',
       'ADDRESSLINE1', 'ADDRESSLINE2', 'CITY', 'STATE', 'POSTALCODE',
       'COUNTRY', 'TERRITORY', 'CONTACTLASTNAME', 'CONTACTFIRSTNAME',
       'DEALSIZE'],
      dtype='object')
```

3. What is the average order value (total sales / number of orders)?

In [243…
```
num_orders = data["ORDERNUMBER"].nunique()
print(number_of_orders)
average_order_value = total_revenue / num_orders
print(average_order_value)
```

```
307
32679.57280130293
```

4. Which month has the highest sales?

In [245…
```
highest_sales_month=data.groupby('MONTH_ID')['SALES'].sum().idxmax()
highest_sales_month
```

Out[245…
```
11
```

5. How do quarterly sales trends vary over the years?

In [253…

```
quarterly_sales = data.groupby(["YEAR_ID", "QTR_ID"])['SALES'].sum().unstack()
quarterly_sales
```

Out[253…

QTR_ID	1	2	3	4
YEAR_ID				
2003	445094.69	562365.22	649514.54	1860005.09
2004	833730.68	766260.73	1109396.27	2014774.92
2005	1071992.36	719494.35	NaN	NaN

Customer & Market Insights

6. Who are the top 5 customers based on revenue?

In [256…

```
top_5_customers=data.groupby('CUSTOMERNAME')['SALES'].sum().nlargest(5)
top_5_customers
```

Out[256…

```
CUSTOMERNAME
Euro Shopping Channel           912294.11
Mini Gifts Distributors Ltd.    654858.06
Australian Collectors, Co.      200995.41
Muscle Machine Inc              197736.94
La Rochelle Gifts               180124.90
Name: SALES, dtype: float64
```

7. What is the distribution of orders by deal size (small, medium, large)?

In [38]:

```
order_dealsize=data['DEALSIZE'].value_counts()
order_dealsize
```

Out[38]:

```
DEALSIZE
Medium    1384
Small     1282
Large      157
Name: count, dtype: int64
```

8. Which country generates the most revenue?

In [40]:

```
country_with_highestRevenue=data.groupby('COUNTRY')['SALES'].sum().idxmax()
country_with_highestRevenue
```

Out[40]:

```
'USA'
```

9. How does sales performance vary by territory?

In [45]:

```
sales_acrossTerritory=data.groupby('TERRITORY')['SALES'].sum()
sales_acrossTerritory
```

Out[45]:

```
TERRITORY
APAC      746121.83
EMEA     4979272.41
Japan     455173.22
Name: SALES, dtype: float64
```

In [47]:

```
sales_by_territory = data.groupby("TERRITORY")["SALES"].sum().dropna()
sales_by_territory
```

Out[47]:

```
TERRITORY
APAC      746121.83
EMEA     4979272.41
Japan     455173.22
Name: SALES, dtype: float64
```

10.What is the most common order quantity?

In [90]:

```
mostcommon_orderquantity=data['QUANTITYORDERED'].mode()
mostcommon_orderquantity
```

Out[90]:

```
0    34
Name: QUANTITYORDERED, dtype: int64
```

In [92]:

```
mostcommon_orderquantity=data['QUANTITYORDERED'].mode()[0]
mostcommon_orderquantity
```

Out[92]:

```
34
```

PRODUCT PERFORMANCE

11. What is the profit margin per product (assuming MSRP - PRICEEACH)?

In [102…

```
PROFIT_MARGIN = data["MSRP"] - data["PRICEEACH"]
PROFIT_MARGIN
```

Out[102…

```
0       -0.70
1       13.65
2        0.26
3       11.74
4       -5.00
        ...
2818   -46.00
2819   -46.00
2820   -46.00
2821    -8.24
2822   -11.52
Length: 2823, dtype: float64
```

12. Which product line contributes the most to total sales?

In [267…

```
top_productline=data.groupby('PRODUCTLINE')['SALES'].sum().idxmax()
top_productline
```

Out[267…

```
'Classic Cars'
```

13. How does pricing affect order quantities?

In [113…

```
pricing_quantity_correlation = data["PRICEEACH"].corr(data["QUANTITYORDERED"])
pricing_quantity_correlation
```

Out[113…

```
0.005564033259239912
```

14. What is the relationship between MSRP and sales performance?

In [115…

```
corr_MSRP_Sales=data['MSRP'].corr(data['SALES'])
corr_MSRP_Sales
```

Out[115…

```
0.6352394340572505
```

15. How many unique products are sold in the dataset?

In [292…

```
unique_products_count = data["PRODUCTCODE"].nunique()
unique_products_count
```

Out[292…

```
109
```

16. What is the average delivery time based on order status and order date?

In [1]:

```
average_delivery_time = data.groupby("STATUS")["ORDERDATE"]).mean()
average_delivery_time
```

```
  Cell In[1], line 1
    average_delivery_time = data.groupby("STATUS")["ORDERDATE"]).mean()
                                                              ^

SyntaxError: unmatched ')'
```

17. What percentage of orders are pending, shipped, or canceled?

In [269…

```
order_status_distribution=data['STATUS'].value_counts(normalize=True)*100
order_status_distribution
```

Out[269…

```
STATUS
Shipped       92.702798
Cancelled      2.125399
Resolved       1.664896
On Hold        1.558626
In Process     1.452356
Disputed       0.495926
Name: proportion, dtype: float64
```

18. What is the average number of orders per customer?

```
In [205…  average_orders_per_customer = data["ORDERNUMBER"].nunique() / data["CUSTOMERNAME"].
          average_orders_per_customer
```

```
Out[205…  3.3369565217391304
```

19. How many orders are placed per month?

```
In [273…  orders_per_month = data.groupby("MONTH_ID")["ORDERNUMBER"].nunique()
          orders_per_month
```

```
Out[273…  MONTH_ID
          1     25
          2     25
          3     25
          4     26
          5     28
          6     16
          7     16
          8     17
          9     20
          10    30
          11    61
          12    18
          Name: ORDERNUMBER, dtype: int64
```

```
In [217…  data['QUANTITYORDERED']
```

```
Out[217…  0       30
          1       34
          2       41
          3       45
          4       49
                  ..
          2818    20
          2819    29
          2820    43
          2821    34
          2822    47
          Name: QUANTITYORDERED, Length: 2823, dtype: int64
```

20. Are there seasonal trends in sales performance?

```
In [225…  seasonal_trends = data.groupby("QTR_ID")["SALES"].mean()
          seasonal_trends
```

```
Out[225…  QTR_ID
          1    3535.064256
          2    3650.838324
          3    3496.840577
          4    3541.846444
          Name: SALES, dtype: float64
```