```
from google.colab import files
uploaded=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving sales_data_sample.csv to sales_data_sample.csv

```
import pandas as pd
df=pd.read_csv('/content/sales_data_sample.csv',encoding='latin1')
df.head()
```

Show hidden output

1. Find the number of rows and columns in the dataset.

```
rows, cols = df.shape
print(f"Rows: {rows}, Columns: {cols}")
```

```
Rows: 2823, Columns: 25
```

2. List all column names.

```
print(df.columns.tolist())
```

```
['ORDERNUMBER', 'QUANTITYORDERED', 'PRICEEACH', 'ORDERLINENUMBER', 'SALES', 'ORD
```

3. Check for missing/null values in the dataset.

```
print(df.isnull().sum())
```

```
ORDERNUMBER         0
QUANTITYORDERED     0
PRICEEACH           0
ORDERLINENUMBER     0
SALES               0
ORDERDATE           0
STATUS              0
QTR_ID              0
MONTH_ID            0
YEAR_ID             0
PRODUCTLINE         0
MSRP                0
PRODUCTCODE         0
CUSTOMERNAME        0
PHONE               0
```

```
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

4. Find the total number of unique products sold.

```
print(df['PRODUCTCODE'].nunique())
```

```
109
```

5. Find the total sales (SALES) made by each product.

```
product_sales = df.groupby('PRODUCTCODE')['SALES'].sum()
print(product_sales)
```

```
PRODUCTCODE
S10_1678      97107.00
S10_1949     191073.03
S10_2016     106017.46
S10_4698     170401.07
S10_4757     113093.73
                 ...
S700_3505     88565.41
S700_3962     80482.72
S700_4002     76175.63
S72_1253      51661.82
S72_3212      61064.10
Name: SALES, Length: 109, dtype: float64
```

6. Identify the top 5 products with the highest total sales.

```
top5_products = product_sales.sort_values(ascending=False).head(5)
print(top5_products)
```

```
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
```

7. Find the average quantity ordered across all transactions.

```
avg_quantity = df['QUANTITYORDERED'].mean()
print(avg_quantity)
```

```
35.09280906836698
```

8. Find the month with the highest total sales.

```
df['ORDERDATE'] = pd.to_datetime(df['ORDERDATE'])
df['MONTH'] = df['ORDERDATE'].dt.month
month_sales = df.groupby('MONTH')['SALES'].sum()
best_month = month_sales.idxmax()
print(f"Best Month: {best_month}")
```

```
Best Month: 11
```

9. Calculate the correlation between quantity ordered and sales.

```
correlation = df['QUANTITYORDERED'].corr(df['SALES'])
print(f"Correlation: {correlation}")
```

```
Correlation: 0.5514261919183568
```

10. Find the customer with the highest number of orders.

```
top_customer = df['CUSTOMERNAME'].value_counts().idxmax()
print(f"Top Customer: {top_customer}")
```

```
Top Customer: Euro Shopping Channel
```

11. Calculate the total sales by each country.

```
country_sales = df.groupby('COUNTRY')['SALES'].sum()
print(country_sales)
```

```
COUNTRY
Australia       630623.10
Austria         202062.53
Belgium         108412.62
Canada          224078.56
Denmark         245637.15
Finland         329581.91
France         1110916.52
Germany         220472.09
Ireland          57756.43
Italy           374674.31
```

```
Japan           188167.81
Norway          307463.70
Philippines      94015.73
Singapore       288488.41
Spain          1215686.92
Sweden          210014.21
Switzerland     117713.56
UK              478880.46
USA            3627982.83
Name: SALES, dtype: float64
```

12. Find the average price per unit across all transactions.

```
avg_price = df['PRICEEACH'].mean()
print(avg_price)
```

```
83.65854410201914
```

13. Find the number of orders for each status (e.g., Shipped, Cancelled, On Hold, etc.).

```
order_status_count = df['STATUS'].value_counts()
print(order_status_count)
```

```
STATUS
Shipped       2617
Cancelled       60
Resolved        47
On Hold         44
In Process      41
Disputed        14
Name: count, dtype: int64
```

14. Find the product with the highest average price per unit.

```
highest_avg_price = df.groupby('PRODUCTCODE')['PRICEEACH'].mean().idxmax()
print(f"Product with Highest Average Price: {highest_avg_price}")
```

```
Product with Highest Average Price: S10_1949
```

15. Find the total revenue generated from each deal size category.

```
deal_size_revenue = df.groupby('DEALSIZE')['SALES'].sum()
print(deal_size_revenue)
```

```
DEALSIZE
Large     1302119.26
Medium    6087432.24
Small     2643077.35
Name: SALES, dtype: float64
```

16. Calculate the monthly growth rate (percent) in sales

```
monthly_sales = df.groupby('MONTH')['SALES'].sum().sort_index()
growth_rate = monthly_sales.pct_change() * 100
print(growth_rate)
```

```
MONTH
1           NaN
2      3.126130
3     -6.902470
4    -11.280354
5     38.031825
6    -50.782437
7     13.220076
8     28.052309
9    -11.312772
10    91.751100
11    88.981173
12   -70.046561
Name: SALES, dtype: float64
```

17. Find the most popular product for each deal size category.

```
popular_product_dealsize = df.groupby(['DEALSIZE', 'PRODUCTCODE']).size().reset_index(name=
idx = popular_product_dealsize.groupby('DEALSIZE')['counts'].idxmax()
result = popular_product_dealsize.loc[idx]
print(result)
```

```
    DEALSIZE PRODUCTCODE  counts
1      Large    S10_1949      14
87    Medium    S18_3232      34
212    Small    S24_1444      26
```

18. Find the minimum, maximum, and average sales value.

```
min_sales = df['SALES'].min()
max_sales = df['SALES'].max()
avg_sales = df['SALES'].mean()
print(f"Min Sales: {min_sales}, Max Sales: {max_sales}, Average Sales: {avg_sales}")
```

```
Min Sales: 482.13, Max Sales: 14082.8, Average Sales: 3553.889071909316
```

19. Identify the top 5 customers who generated the most profit per unit sold (SALES/QUANTITYORDERED).

```
df['PROFIT_PER_UNIT'] = df['SALES'] / df['QUANTITYORDERED']
top5_profit_customers = df.groupby('CUSTOMERNAME')['PROFIT_PER_UNIT'].mean().sort_values(as
```

```
print(top5_profit_customers)
```

```
CUSTOMERNAME
Super Scale Inc.              128.452353
Volvo Model Replicas, Co      119.288947
Royale Belge                  115.195000
La Corne D'abondance, Co.     113.650435
Herkku Gifts                  113.558621
Name: PROFIT_PER_UNIT, dtype: float64
```

20. Find the city with the highest sales.

```
city_sales = df.groupby('CITY')['SALES'].sum()
top_city = city_sales.idxmax()
print(f"City with Highest Sales: {top_city}")
```

```
City with Highest Sales: Madrid
```