EDS ASSIGNMENT

DATA SET GIVEN IS SALES DATASET

NAME : PRINCE BANSAL

DIV : CS6

ROLL NO. : 43

BATCH : C62

PRN : 202401100064

In [1]:
```
!pip install pandas
```
```
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in c:\programdata\anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2.9.0.post0) Requirement already satisfied:
pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

In [9]:
```
!pip install numpy
```
```
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (1.26.4)
```

In [17]:
```
import pandas as pd
df=pd.read_csv("C://Users//ASUS//Downloads//sales_data_sample.csv", encoding='latin1') df
```

Out[17]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDERDATE	STATUS	QTR_ID	MONTH_ID	YEAR_ID	...	ADDRESSLINE1	ADDRESSLINE2	CITY	STATE	POSTALCO
0	10107	30	95.70	2	2871.00	2/24/2003 0:00	Shipped	1	2	2003	...	897 Long Airport Avenue	NaN	NYC	NY	100
1	10121	34	81.35	5	2765.90	5/7/2003 0:00	Shipped	2	5	2003	...	59 rue de	NaN	Reims	NaN	511
2	10134	41	94.74	2	3884.34	7/1/2003 0:00	Shipped	3	7	2003	...	27 rue du Colonel Pierre Avia	NaN	Paris	NaN	755
3	10145	45	83.26	6	3746.70	8/25/2003 0:00	Shipped	3	8	2003	...	78934 Hillside	NaN	Pasadena	CA	900
4	10159	49	100.00	14	5205.27	10/10/2003 0:00	Shipped	4	10	2003	...	7734 Strong St.	NaN	San Francisco	CA	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2818	10350	20	100.00	15	2244.40	12/2/2004 0:00	Shipped	4	12	2004	...	C/ Moralzarzal,	NaN	Madrid	NaN	280
2819	10373	29	100.00	1	3978.51	1/31/2005 0:00	Shipped	1	1	2005	...	Torikatu 38	NaN	Oulu	NaN	901
2820	10386	43	100.00	4	5417.57	3/1/2005 0:00	Resolved	1	3	2005	...	C/ Moralzarzal,	NaN	Madrid	NaN	280
2821	10397	34	62.24	1	2116.16	3/28/2005 0:00	Shipped	1	3	2005	...	1 rue Alsace-	NaN	Toulouse	NaN	310

2823 rows × 25 columns

1. Find the total number of orders.

In [18]:
```
total_orders = df['ORDERNUMBER'].nunique() print(total_orders)
```
```
307
```

2. Calculate the total sales.

In [19]:
```
total_sales=df['SALES'].sum()
print(total_sales)
```
```
10032628.85
```

3. Find the average quantity ordered per order.

In [20]:
```
avg_quantity = df['QUANTITYORDERED'].mean() print(avg_quantity)
```
```
35.09280906836698
```

4. List all unique product lines.

In [21]:
```
unique_productlines = df['PRODUCTLINE'].unique()
print(unique_productlines)
```
```
['Motorcycles' 'Classic Cars' 'Trucks and Buses' 'Vintage Cars' 'Planes' 'Ships' 'Trains']
```

5. Find the maximum sales value.

In [22]:
```
max_sale=df['SALES'].max()
print(max_sale)
```
```
14082.8
```

6. Find the minimum sales value.

In [23]:
```
min_sale=df['SALES'].min()
print(min_sale)
```
```
482.13
```

7. Identify the most sold product (by quantity).

In [24]:
```
most_sold_product = df.groupby('PRODUCTCODE')['QUANTITYORDERED'].sum().idxmax()
print(most_sold_product)
```
```
S18_3232
```

8. Find the number of orders shipped.

In [25]:
```
orders_shipped=df[df['STATUS']=='Shipped'].shape[0]
print(orders_shipped)
```
```
2617
```

9. Find the top 5 customers based on total sales.

In [26]:
```
top_customers = df.groupby('CUSTOMERNAME')['SALES'].sum().sort_values(ascending=False).head(5)   print(top_customers)
```
```
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
```

10. Count the number of orders per country.

In [27]:
```
orders_per_country = df['COUNTRY'].value_counts()
print(orders_per_country)
```
```
COUNTRY
USA            1004
Spain           342
France          314
Australia       185
UK              144
Italy           113
Finland          92
Norway           85
Singapore        79
Canada           70
Denmark          63
Germany          62
Sweden           57
Austria          55
Japan            52
Belgium          33
Switzerland      31
Philippines      26
Ireland          16
Name: count, dtype: int64
```

11. Find out in which quarter maximum sales happened.

In [28]:
```
best_quarter = df.groupby('QTR_ID')['SALES'].sum().idxmax()
print(best_quarter)
```
```
4
```

12. Calculate total sales per year.

In [29]:
```
sales_per_year = df.groupby('YEAR_ID')['SALES'].sum() print(sales_per_year)
```
```
YEAR_ID
2003    3516979.54
2004    4724162.60
2005    1791486.71
Name: SALES, dtype: float64
```

13. Find the month with highest total sales.

In [30]:
```
best_month = df.groupby('MONTH_ID')['SALES'].sum().idxmax()
print(best_month)
```
```
11
```

14. Find the average sales price (PRICEEACH).

In [31]:
```
average_price = df['PRICEEACH'].mean() print(average_price)
```
```
83.65854410201914
```

15. Find all customers from USA.

In [32]:
```
customers_usa=df[df['COUNTRY']=='USA']['CUSTOMERNAME'].unique() print(customers_usa)
```
```
['Land of Toys Inc.' 'Toys4GrownUps.com' 'Corporate Gift Ideas Co.' 'Technics Stores Inc.'
 'Mini Wheels Co.' 'Vitachrome Inc.'
 'Tekni Collectables Inc.' 'Gift Depot Inc.' "Marta's Replicas Co." 'Diecast Classics Inc.'
 'FunGiftIdeas.com' 'Classic Legends Inc.' 'Classic Gift Ideas, Inc' 'West Coast Collectables
 Co.'
 'Cambridge Collectables Co.' 'Super Scale Inc.'
 'Mini Gifts Distributors Ltd.' 'Online Diecast Creations Co.' 'Collectables For Less Inc.' 'Motor
 Mint Distributors Inc.'
 'Mini Classics' 'Mini Creations Ltd.' "Men 'R' US Retailers, Ltd." 'Collectable Mini Designs Co.'
 'Gifts4AllAges.com'
 'The Sharp Gifts Warehouse' 'Diecast Collectables'
 'Online Mini Collectables' 'Muscle Machine Inc' 'Microscale Inc.'
 'Boards & Toys Co.' 'Signal Collectibles Ltd.' 'Signal Gift Stores' 'Gift Ideas Corp.' 'Auto-
 Moto Classics Inc.']
```

16. Calculate how many products have MSRP greater than 100.

In [33]:
```
products_high_msrp=(df['MSRP']>100).sum() print(products_high_msrp)
```
```
1268
```

17. Find the product line that generated the highest sales.

In [35]:
```
best_productline = df.groupby('PRODUCTLINE')['SALES'].sum().idxmax() print(best_productline)
```
```
Classic Cars
```

18. Find number of unique cities customers are from.

In [37]:
```
unique_cities=df['CITY'].nunique()
print(unique_cities)
```
```
73
```

19. Find correlation between Quantity Ordered and Sales.

In [38]:
```
correlation = df['QUANTITYORDERED'].corr(df['SALES']) print(correlation)
```
```
0.5514261919183567
```

20. Check how many deals are 'Large'.

In [40]:
```
large_deals=(df['DEALSIZE']=='Large').sum() print(large_deals)
```
```
157
```

In [ ]: