Data Processing and Feature engineering:

1. Data Cleaning

a) Loading the Dataset

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv('Superstore_Sales_Dataset.csv')
df
```

Out[3]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Ar
0	ORD1000	2023-03-28 00:00:00	2023-03-30 00:00:00	Customer 86	Home Office	Office Supplies	Binders	Binders 96	
1	ORD1001	2023-10-15 00:00:00	2023-10-17 00:00:00	Customer 449	Home Office	Technology	Phones	Phones 54	
2	ORD1002	2023-01-24 00:00:00	2023-01-30 00:00:00	Customer 90	Corporate	Technology	Printers	Printers 47	
3	ORD1003	2023-04-04 00:00:00	2023-04-09 00:00:00	Customer 383	Home Office	Technology	Printers	Printers 92	
4	ORD1004	2023-06-11 00:00:00	2023-06-15 00:00:00	Customer 84	Small Business	Office Supplies	Paper	Paper 43	
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17 00:00:00	2023-02-23 00:00:00	Customer 122	Corporate	Technology	Monitors	Monitors 66	
996	ORD1996	2023-03-27 00:00:00	2023-04-01 00:00:00	Customer 326	Consumer	Technology	Laptops	Laptops 49	
997	ORD1997	2023-09-12 00:00:00	2023-09-15 00:00:00	Customer 197	Small Business	Office Supplies	Paper	Paper 14	
998	ORD1998	2023-04-13 00:00:00	2023-04-15 00:00:00	Customer 350	Small Business	Office Supplies	Binders	Binders 90	
999	ORD1999	2023-10-12 00:00:00	2023-10-15 00:00:00	Customer 499	Home Office	Technology	Monitors	Monitors 97	

1000 rows × 14 columns

In [6]:
```
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Order ID          1000 non-null   object
 1   Order Date        1000 non-null   object
 2   Ship Date         1000 non-null   object
 3   Customer Name     1000 non-null   object
 4   Customer Segment  1000 non-null   object
 5   Category          1000 non-null   object
 6   Sub-Category      1000 non-null   object
 7   Product Name      1000 non-null   object
 8   Sales Amount      1000 non-null   float64
 9   Profit            1000 non-null   float64
 10  Discount          1000 non-null   float64
 11  Quantity          1000 non-null   int64
 12  Region            1000 non-null   object
 13  State             1000 non-null   object
dtypes: float64(3), int64(1), object(10)
memory usage: 109.5+ KB
```

Out[6]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sa Amo
0	ORD1000	2023-03-28 00:00:00	2023-03-30 00:00:00	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611
1	ORD1001	2023-10-15 00:00:00	2023-10-17 00:00:00	Customer 449	Home Office	Technology	Phones	Phones 54	62
2	ORD1002	2023-01-24 00:00:00	2023-01-30 00:00:00	Customer 90	Corporate	Technology	Printers	Printers 47	454
3	ORD1003	2023-04-04 00:00:00	2023-04-09 00:00:00	Customer 383	Home Office	Technology	Printers	Printers 92	404
4	ORD1004	2023-06-11 00:00:00	2023-06-15 00:00:00	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295

b) Handle Missing Values & Date Formatting

In [8]:
```
df.interpolate(method='linear', inplace=True)
```

```
C:\Users\Manisha\AppData\Local\Temp\ipykernel_1324\2868764835.py:1: FutureWarning: D
ataFrame.interpolate with object dtype is deprecated and will raise in a future vers
ion. Call obj.infer_objects(copy=False) before interpolating instead.
  df.interpolate(method='linear', inplace=True)
```

In [10]:
```
df['Order Date'] = pd.to_datetime(df['Order Date'], errors='coerce')
```

```
df['Ship Date'] = pd.to_datetime(df['Ship Date'], errors='coerce')
```

In [11]:
```
df
```

Out[11]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sal Amou
0	ORD1000	2023-03-28	2023-03-30	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611.3
1	ORD1001	2023-10-15	2023-10-17	Customer 449	Home Office	Technology	Phones	Phones 54	62.1
2	ORD1002	2023-01-24	2023-01-30	Customer 90	Corporate	Technology	Printers	Printers 47	454.6
3	ORD1003	2023-04-04	2023-04-09	Customer 383	Home Office	Technology	Printers	Printers 92	404.4
4	ORD1004	2023-06-11	2023-06-15	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295.6
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17	2023-02-23	Customer 122	Corporate	Technology	Monitors	Monitors 66	10.6
996	ORD1996	2023-03-27	2023-04-01	Customer 326	Consumer	Technology	Laptops	Laptops 49	847.5
997	ORD1997	2023-09-12	2023-09-15	Customer 197	Small Business	Office Supplies	Paper	Paper 14	367.6
998	ORD1998	2023-04-13	2023-04-15	Customer 350	Small Business	Office Supplies	Binders	Binders 90	749.4
999	ORD1999	2023-10-12	2023-10-15	Customer 499	Home Office	Technology	Monitors	Monitors 97	549.4

1000 rows × 14 columns

In [12]:
```
df['Year'] = df['Order Date'].dt.year
df['Month'] = df['Order Date'].dt.month
df['Week'] = df['Order Date'].dt.isocalendar().week
df['Day'] = df['Order Date'].dt.day
df['DayOfWeek'] = df['Order Date'].dt.dayofweek
```

In [13]:
```
df
```

Out[13]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sal Amou
0	ORD1000	2023-03-28	2023-03-30	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611.3
1	ORD1001	2023-10-15	2023-10-17	Customer 449	Home Office	Technology	Phones	Phones 54	62.7
2	ORD1002	2023-01-24	2023-01-30	Customer 90	Corporate	Technology	Printers	Printers 47	454.6
3	ORD1003	2023-04-04	2023-04-09	Customer 383	Home Office	Technology	Printers	Printers 92	404.4
4	ORD1004	2023-06-11	2023-06-15	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295.6
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17	2023-02-23	Customer 122	Corporate	Technology	Monitors	Monitors 66	10.6
996	ORD1996	2023-03-27	2023-04-01	Customer 326	Consumer	Technology	Laptops	Laptops 49	847.5
997	ORD1997	2023-09-12	2023-09-15	Customer 197	Small Business	Office Supplies	Paper	Paper 14	367.6
998	ORD1998	2023-04-13	2023-04-15	Customer 350	Small Business	Office Supplies	Binders	Binders 90	749.4
999	ORD1999	2023-10-12	2023-10-15	Customer 499	Home Office	Technology	Monitors	Monitors 97	549.4

1000 rows × 19 columns

c) Government Payday (15th & last day of month)

In [15]:
```
df['Is_Payday'] = df['Day'].isin([15, df['Order Date'].dt.daysinmonth])
```

In [16]:
```
df
```

Out[16]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sal Amou
0	ORD1000	2023-03-28	2023-03-30	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611.3
1	ORD1001	2023-10-15	2023-10-17	Customer 449	Home Office	Technology	Phones	Phones 54	62.7
2	ORD1002	2023-01-24	2023-01-30	Customer 90	Corporate	Technology	Printers	Printers 47	454.6
3	ORD1003	2023-04-04	2023-04-09	Customer 383	Home Office	Technology	Printers	Printers 92	404.4
4	ORD1004	2023-06-11	2023-06-15	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295.6
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17	2023-02-23	Customer 122	Corporate	Technology	Monitors	Monitors 66	10.6
996	ORD1996	2023-03-27	2023-04-01	Customer 326	Consumer	Technology	Laptops	Laptops 49	847.5
997	ORD1997	2023-09-12	2023-09-15	Customer 197	Small Business	Office Supplies	Paper	Paper 14	367.6
998	ORD1998	2023-04-13	2023-04-15	Customer 350	Small Business	Office Supplies	Binders	Binders 90	749.4
999	ORD1999	2023-10-12	2023-10-15	Customer 499	Home Office	Technology	Monitors	Monitors 97	549.4

1000 rows × 20 columns

d) Earthquake (April 16, 2016) impact flag

In [17]:
```
df['Is_Earthquake'] = df['Order Date'] == pd.Timestamp('2016-04-16')
```

In [18]:
```
df
```

Out[18]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sal Amou
0	ORD1000	2023-03-28	2023-03-30	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611.3
1	ORD1001	2023-10-15	2023-10-17	Customer 449	Home Office	Technology	Phones	Phones 54	62.7
2	ORD1002	2023-01-24	2023-01-30	Customer 90	Corporate	Technology	Printers	Printers 47	454.6
3	ORD1003	2023-04-04	2023-04-09	Customer 383	Home Office	Technology	Printers	Printers 92	404.4
4	ORD1004	2023-06-11	2023-06-15	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295.6
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17	2023-02-23	Customer 122	Corporate	Technology	Monitors	Monitors 66	10.6
996	ORD1996	2023-03-27	2023-04-01	Customer 326	Consumer	Technology	Laptops	Laptops 49	847.5
997	ORD1997	2023-09-12	2023-09-15	Customer 197	Small Business	Office Supplies	Paper	Paper 14	367.6
998	ORD1998	2023-04-13	2023-04-15	Customer 350	Small Business	Office Supplies	Binders	Binders 90	749.4
999	ORD1999	2023-10-12	2023-10-15	Customer 499	Home Office	Technology	Monitors	Monitors 97	549.4

1000 rows × 21 columns

2. Feature Engineering

Rolling Statistics

In [20]:
```
df['Sales_Rolling_7'] = df['Sales Amount'].rolling(window=7).mean()
df['Sales_Rolling_30'] = df['Sales Amount'].rolling(window=30).mean()
df['Sales_Lag_7'] = df['Sales Amount'].shift(7)
df['Sales_Lag_30'] = df['Sales Amount'].shift(30)
```

In [21]:
```
df
```

Out[21]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sal Amou
0	ORD1000	2023-03-28	2023-03-30	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611.3
1	ORD1001	2023-10-15	2023-10-17	Customer 449	Home Office	Technology	Phones	Phones 54	62.7
2	ORD1002	2023-01-24	2023-01-30	Customer 90	Corporate	Technology	Printers	Printers 47	454.6
3	ORD1003	2023-04-04	2023-04-09	Customer 383	Home Office	Technology	Printers	Printers 92	404.4
4	ORD1004	2023-06-11	2023-06-15	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295.6
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17	2023-02-23	Customer 122	Corporate	Technology	Monitors	Monitors 66	10.0
996	ORD1996	2023-03-27	2023-04-01	Customer 326	Consumer	Technology	Laptops	Laptops 49	847.5
997	ORD1997	2023-09-12	2023-09-15	Customer 197	Small Business	Office Supplies	Paper	Paper 14	367.0
998	ORD1998	2023-04-13	2023-04-15	Customer 350	Small Business	Office Supplies	Binders	Binders 90	749.4
999	ORD1999	2023-10-12	2023-10-15	Customer 499	Home Office	Technology	Monitors	Monitors 97	549.4

1000 rows × 25 columns

Average sales per category

In [32]:
```
df['Avg_Sales_Category'] = df.groupby('Category')['Sales Amount'].transform('mean')
df['Avg_Sales_Sub-Category'] = df.groupby('Sub-Category')['Sales Amount'].transform
```

In [33]:
```
df
```

Out[33]:

	Order ID	Order Date	Ship Date	Customer Name	Customer Segment	Category	Sub-Category	Product Name	Sal Amou
0	ORD1000	2023-03-28	2023-03-30	Customer 86	Home Office	Office Supplies	Binders	Binders 96	611.3
1	ORD1001	2023-10-15	2023-10-17	Customer 449	Home Office	Technology	Phones	Phones 54	62.7
2	ORD1002	2023-01-24	2023-01-30	Customer 90	Corporate	Technology	Printers	Printers 47	454.6
3	ORD1003	2023-04-04	2023-04-09	Customer 383	Home Office	Technology	Printers	Printers 92	404.4
4	ORD1004	2023-06-11	2023-06-15	Customer 84	Small Business	Office Supplies	Paper	Paper 43	295.6
...	...	...	...	...	...	...	...	...	
995	ORD1995	2023-02-17	2023-02-23	Customer 122	Corporate	Technology	Monitors	Monitors 66	10.6
996	ORD1996	2023-03-27	2023-04-01	Customer 326	Consumer	Technology	Laptops	Laptops 49	847.5
997	ORD1997	2023-09-12	2023-09-15	Customer 197	Small Business	Office Supplies	Paper	Paper 14	367.6
998	ORD1998	2023-04-13	2023-04-15	Customer 350	Small Business	Office Supplies	Binders	Binders 90	749.4
999	ORD1999	2023-10-12	2023-10-15	Customer 499	Home Office	Technology	Monitors	Monitors 97	549.4

1000 rows × 27 columns

Top-selling product per category

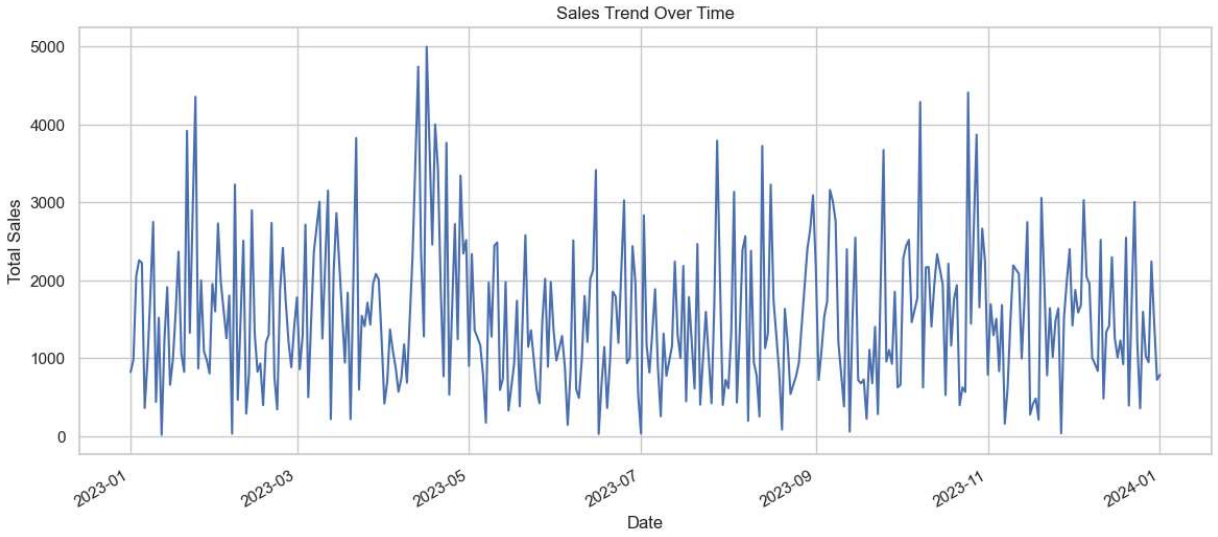
In [34]:
```
top_products = df.groupby('Sub-Category')['Sales Amount'].sum().nlargest(5)
print("Top-Selling Products:", top_products)
```

```
Top-Selling Products: Sub-Category
Sofas        58750.99
Paper        46329.49
Binders      46039.48
Monitors     44821.69
Bookcases    44817.72
Name: Sales Amount, dtype: float64
```

3 Exploratory Data analysis

1. Sales trend

In [39]:
```
plt.figure(figsize=(14,6))
df.groupby('Order Date')['Sales Amount'].sum().plot()
plt.title("Sales Trend Over Time")
plt.xlabel("Date")
plt.ylabel("Total Sales")
plt.show()
```



2. Heatmap

In [41]:
```
corr = df[['Sales Amount', 'Profit', 'Quantity']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```


Correlation Matrix
Sales Amount
Profit
Quantity
1
0.77
-0.0034
Sales Amount
Profit
Quantity
1.0
0.8
0.6
0.4
0.2
0.0

In [ ]:

In [ ]: