Import all Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
import os
import scipy
```

Step 1: Data Upload

Check file Directory (path) and change path accordingly

In [2]:
```
os.getcwd()
```

Out[2]:
```
'C:\\Users\\HP PC'
```

In [3]:
```
os.chdir("C:/Users/HP PC/Data Analyst")
```

Read CSV file

In [4]:
```
df = pd.read_csv("Retail.csv")
```

In [5]:
```
df
```

Out[5]:

	Order_ID	Order_Date	Customer_Name	Customer_ID	Region	State	Produ
0	1	13-04-2022	Customer_286	C4657	North	Delhi	
1	2	12-03-2023	Customer_477	C4582	West	Maharashtra	
2	3	28-09-2022	Customer_1658	C5557	West	Rajasthan	
3	4	17-04-2022	Customer_190	C2674	East	Odisha	
4	5	13-03-2022	Customer_1131	C2291	North	Uttar Pradesh	
...	...	...	...	...	...	...	
4995	4996	04-12-2022	Customer_422	C1173	West	Goa	
4996	4997	15-06-2022	Customer_1581	C5399	West	Rajasthan	
4997	4998	18-01-2023	Customer_1867	C1513	East	West Bengal	
4998	4999	11-01-2023	Customer_1558	C2167	South	Andhra Pradesh	
4999	5000	16-04-2023	Customer_1632	C3925	South	Kerala	

5000 rows × 11 columns

Step 2: Data Cleaning

Total rows and columns

In [6]:
```
df.shape
```

Out[6]:
```
(5000, 11)
```

In [7]:
```
df.info
```

Out[7]:
```
<bound method DataFrame.info of       Order_ID  Order_Date  Customer_Name Custo
mer_ID Region           State  \
0            1  13-04-2022   Customer_286       C4657  North           Delhi
1            2  12-03-2023   Customer_477       C4582   West     Maharashtra
2            3  28-09-2022  Customer_1658       C5557   West       Rajasthan
3            4  17-04-2022   Customer_190       C2674   East          Odisha
4            5  13-03-2022  Customer_1131       C2291  North   Uttar Pradesh
...        ...         ...           ...         ...    ...             ...
4995      4996  04-12-2022   Customer_422       C1173   West             Goa
4996      4997  15-06-2022  Customer_1581       C5399   West       Rajasthan
4997      4998  18-01-2023  Customer_1867       C1513   East     West Bengal
4998      4999  11-01-2023  Customer_1558       C2167  South  Andhra Pradesh
4999      5000  16-04-2023  Customer_1632       C3925  South          Kerala

     Product_Category Product_Name  Quantity  Sales_Amount  Profit
0            Clothing        Jeans         2       3415.83  230.19
1         Electronics       Laptop         9       3049.89  589.25
2          Stationery       Folder         1       3818.16  513.63
3           Furniture         Sofa         7        573.91   97.99
4         Electronics   Headphones         5       4164.08  925.85
...               ...          ...       ...           ...     ...
4995      Electronics       Tablet         6       2126.41  177.93
4996         Clothing       Jacket         6       4912.59  255.07
4997      Electronics       Camera         7       1669.11  162.01
4998       Stationery       Folder         2        476.99   55.98
4999        Furniture          Bed        10        478.17  114.20

[5000 rows x 11 columns]>
```

Rename of column, check unique ID, Null, Duplicate

In [8]:
```
df = df.rename(columns = {"Sales_Amount" : "Sales"})
```

In [9]:
```
df["Customer_ID"].nunique()
```

Out[9]:
```
3843
```

In [10]:
```
df.isnull()
```

Out[10]:

	Order_ID	Order_Date	Customer_Name	Customer_ID	Region	State	Product_Cat
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
...	...	...	...	...	...	...	
4995	False	False	False	False	False	False	
4996	False	False	False	False	False	False	
4997	False	False	False	False	False	False	
4998	False	False	False	False	False	False	
4999	False	False	False	False	False	False	

5000 rows × 11 columns

In [11]:
```
df.duplicated()
```

Out[11]:
```
0       False
1       False
2       False
3       False
4       False
        ...
4995    False
4996    False
4997    False
4998    False
4999    False
Length: 5000, dtype: bool
```

Check summary statistics of data

In [12]:
```
df.describe()
```

Out[12]:

	Order_ID	Quantity	Sales	Profit
count	5000.000000	5000.000000	5000.000000	5000.000000
mean	2500.500000	5.519400	2543.584210	387.212584
std	1443.520003	2.852937	1423.844885	278.039807
min	1.000000	1.000000	102.630000	5.990000
25%	1250.750000	3.000000	1310.945000	162.642500
50%	2500.500000	6.000000	2525.815000	324.395000
75%	3750.250000	8.000000	3798.030000	565.280000
max	5000.000000	10.000000	4998.290000	1235.040000

Step 3: Exploratory Data Analysis (EDA)

Total Sales according to region

In [13]:
```
df.groupby("Region")["Sales"].sum().sort_values(ascending=False)
```

Out[13]:
```
Region
South    3315243.07
East     3180953.25
North    3111654.70
West     3110070.03
Name: Sales, dtype: float64
```

Top 5 Product by Sales

In [14]:
```
top = df.groupby("Product_Name")["Profit"].sum().sort_values(ascending= False).h
print(top)
```

```
Product_Name
Smartphone    111768.73
Marker        109896.34
Jacket        107260.63
Pencil        107144.14
Shoes         101799.88
Name: Profit, dtype: float64
```

Monthly Sales

In [15]:
```
df["Order_Date"] = pd.to_datetime(df["Order_Date"])
```

In [16]:
```
df["Order_Date"] = pd.to_datetime(df["Order_Date"])
df["Year_month"] = df["Order_Date"].dt.to_period("M")
mom = df.groupby("Year_month")["Sales"].sum().reset_index()
print(mom)
```

	Year_month	Sales
0	2022-01	579966.35
1	2022-02	440697.64
2	2022-03	529072.82
3	2022-04	599488.67
4	2022-05	524541.12
5	2022-06	526449.39
6	2022-07	517995.80
7	2022-08	544249.92
8	2022-09	478806.21
9	2022-10	511653.35
10	2022-11	489869.95
11	2022-12	554153.61
12	2023-01	539832.74
13	2023-02	486416.99
14	2023-03	520909.23
15	2023-04	480217.60
16	2023-05	550174.26
17	2023-06	493725.68
18	2023-07	614933.31
19	2023-08	489003.58
20	2023-09	558708.60
21	2023-10	633125.73
22	2023-11	563936.36
23	2023-12	489992.14

Monthly Sales Growth Percentage

In [17]:
```
df["Order_Date"] = pd.to_datetime(df["Order_Date"])
df["Year_month"] = df["Order_Date"].dt.to_period("M")
mom = df.groupby("Year_month")["Sales"].sum().reset_index()
mom["Growth"] = round(mom["Sales"].pct_change() * 100,2)
print(mom)
```

	Year_month	Sales	Growth
0	2022-01	579966.35	NaN
1	2022-02	440697.64	-24.01
2	2022-03	529072.82	20.05
3	2022-04	599488.67	13.31
4	2022-05	524541.12	-12.50
5	2022-06	526449.39	0.36
6	2022-07	517995.80	-1.61
7	2022-08	544249.92	5.07
8	2022-09	478806.21	-12.02
9	2022-10	511653.35	6.86
10	2022-11	489869.95	-4.26
11	2022-12	554153.61	13.12
12	2023-01	539832.74	-2.58
13	2023-02	486416.99	-9.89
14	2023-03	520909.23	7.09
15	2023-04	480217.60	-7.81
16	2023-05	550174.26	14.57
17	2023-06	493725.68	-10.26
18	2023-07	614933.31	24.55
19	2023-08	489003.58	-20.48
20	2023-09	558708.60	14.25
21	2023-10	633125.73	13.32
22	2023-11	563936.36	-10.93
23	2023-12	489992.14	-13.11

Total Sales for 2023 year

In [18]:
```
sales_2023 = df[df["Order_Date"].dt.year == 2023]
sales_2023.groupby(sales_2023["Order_Date"].dt.month)["Sales"].sum().sort_values
```

Out[18]:
```
Order_Date
10    633125.73
Name: Sales, dtype: float64
```

Top 3 customer by profit

In [19]:
```
top_Customer = df.groupby("Customer_Name")["Profit"].sum().sort_values(ascending
print(top_Customer)
```

```
Customer_Name
Customer_1167    4953.98
Customer_497     4408.94
Customer_1649    3862.51
Name: Profit, dtype: float64
```

Customer who purchased from more than 2 region

In [20]:
```
df.groupby("Customer_ID")["Region"].nunique().reset_index().query("Region > 2")
```

Out[20]:

	Customer_ID	Region
8	C1015	3
38	C1076	3
51	C1107	3
128	C1276	3
148	C1315	3
...	...	...
3598	C9399	3
3607	C9420	3
3616	C9437	3
3644	C9501	3
3652	C9524	3

65 rows × 2 columns

Average profit by each product

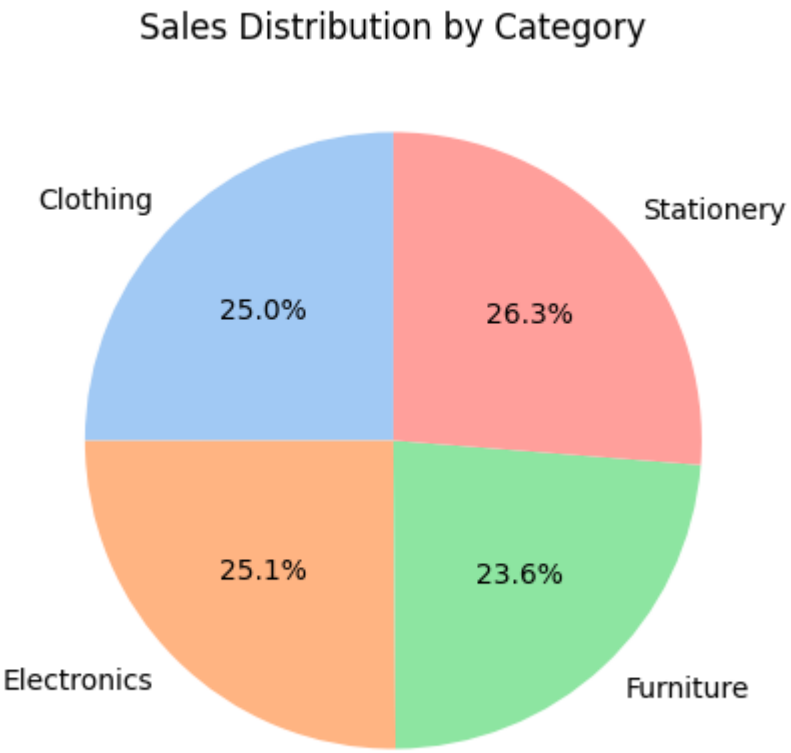
In [21]:
```
Average_profit = df.groupby("Product_Name")["Profit"].mean().sort_values(ascendi
print(Average_profit)
```

```
Product_Name
Dress         411.661627
Marker        407.023481
Headphones    403.951747
Jacket        401.725206
Pencil        399.791567
Notebook      398.146200
Smartphone    397.753488
Camera        395.773676
Laptop        390.396947
Cupboard      389.379683
Table         387.494549
Jeans         383.738765
Shoes         382.706316
Sofa          380.798710
Chair         379.463605
Shirt         374.036855
Pen           373.219084
Folder        368.890526
Bed           368.353991
Tablet        352.888046
Name: Profit, dtype: float64
```

Step 4: Data Visualization

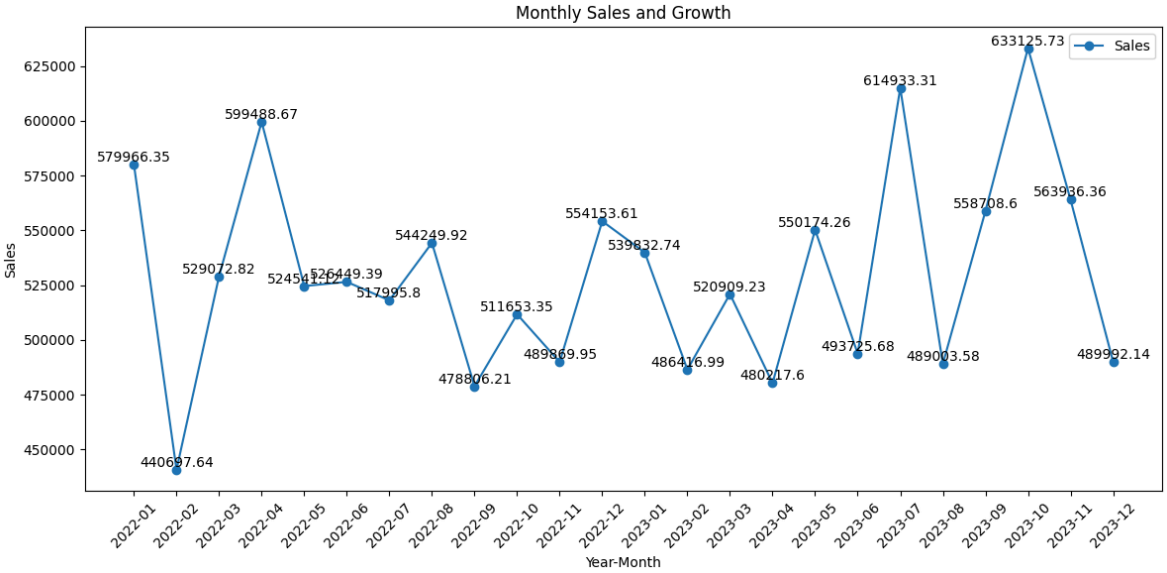
Pie chart: Product_Category by sales

In [22]:
```
category_sales = df.groupby("Product_Category")["Sales"].sum()
plt.figure(figsize=(10,5))
plt.pie(category_sales, labels=category_sales.index, autopct="%1.1f%%", startang
plt.title("Sales Distribution by Category")
plt.show()
```



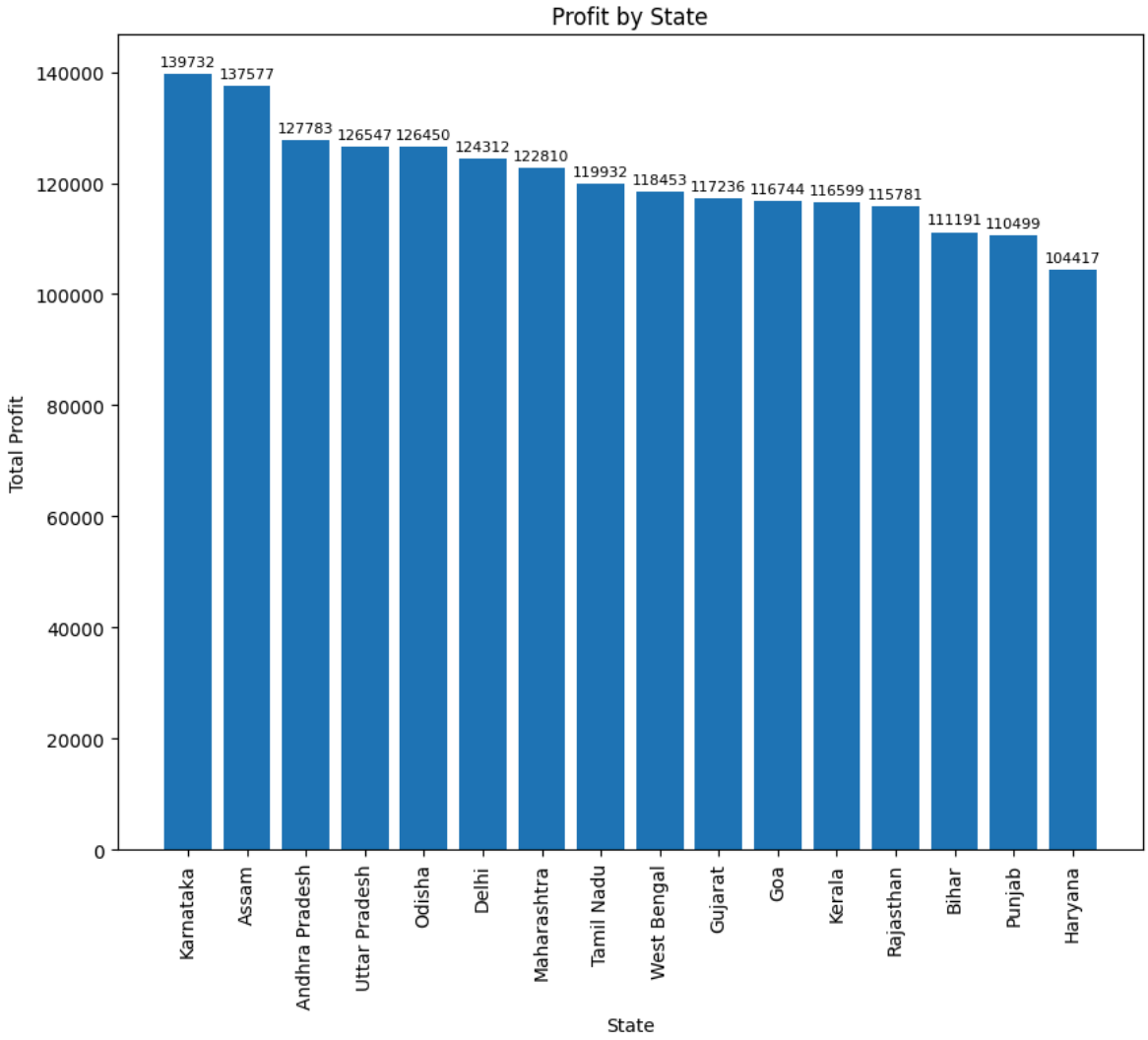
Line chart: Month by Sales

In [23]:
```
plt.figure(figsize=(12,6))
plt.plot(mom["Year_month"].astype(str), mom["Sales"], marker="o", label="Sales")

# Annotate values
for i, val in enumerate(mom["Sales"]):
    plt.text(i, val, str(val), ha="center", va="bottom")

plt.title("Monthly Sales and Growth")
plt.xlabel("Year-Month")
plt.ylabel("Sales")
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



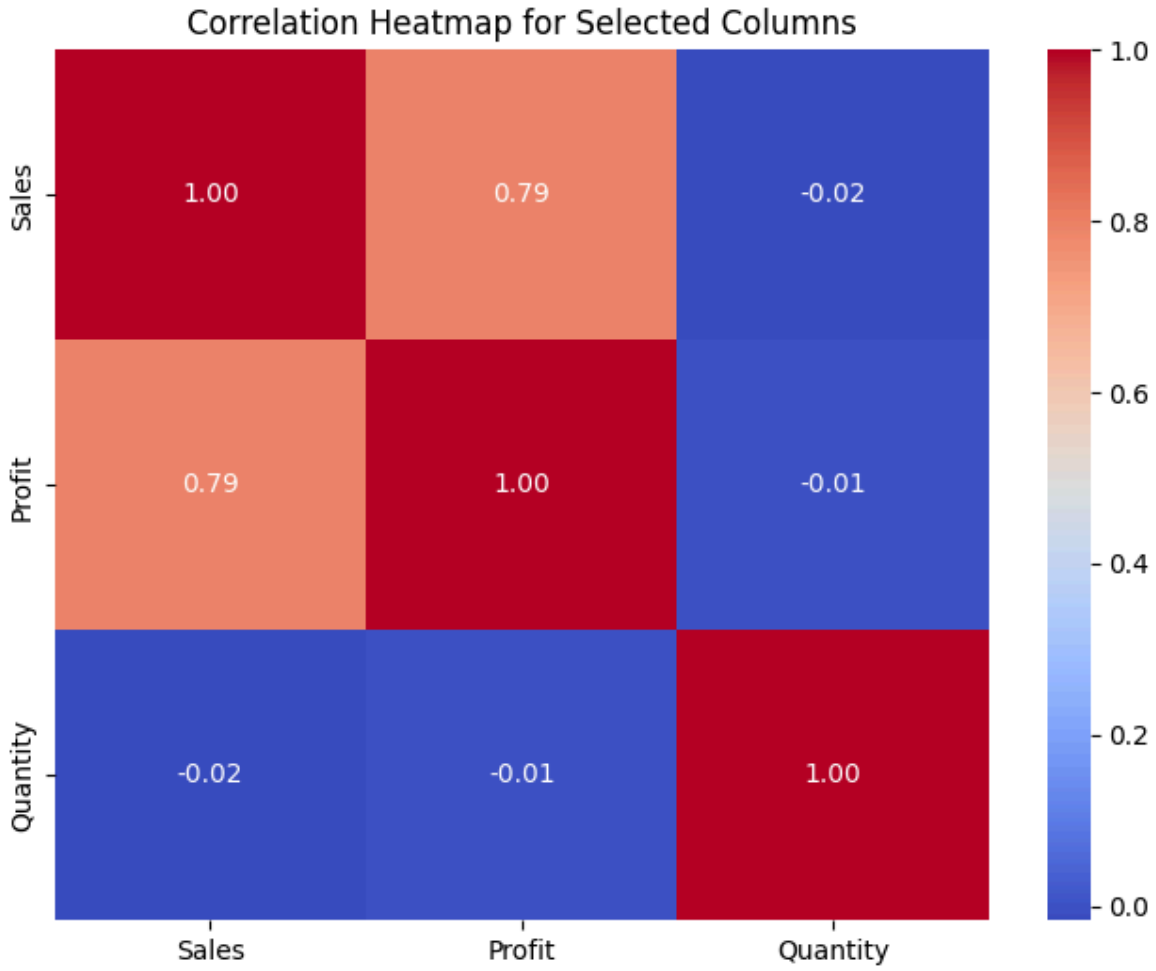
Bar chart: Profit by Sales

In [24]:
```
State_Profit = df.groupby("State")["Profit"].sum().sort_values(ascending=False)
plt.figure(figsize=(10,8))
bars = plt.bar(State_Profit.index, State_Profit.values)
plt.xticks(rotation=90)
plt.xlabel("State")
plt.ylabel("Total Profit")
plt.title("Profit by State")
plt.bar_label(bars, fmt="%.0f", fontsize=8, padding=3)
plt.show()
```



Heat Map: Coorelation of numeric Sales, Profit, Quantity

In [25]:
```
plt.figure(figsize=(8,6))
sns.heatmap(df[["Sales", "Profit", "Quantity"]].corr(), annot=True, cmap="coolwa
plt.title("Correlation Heatmap for Selected Columns")
plt.show()
```



Step 5: Statistical Data

Find the Outlier

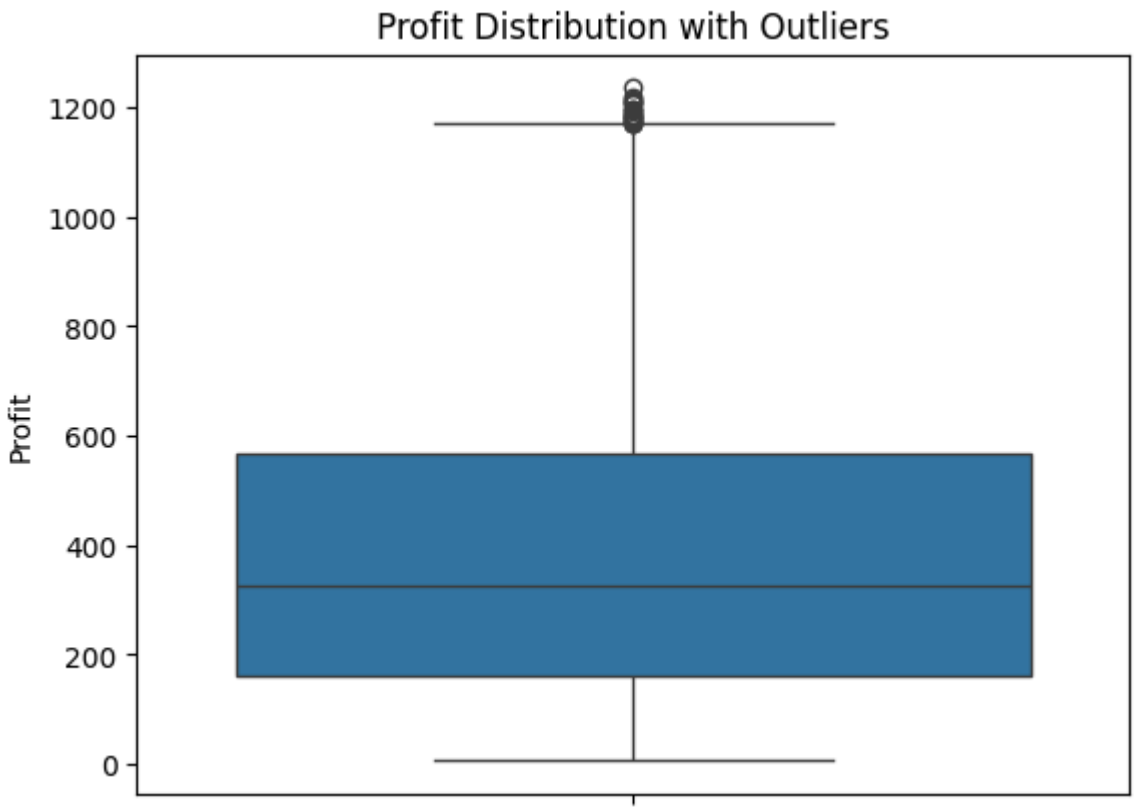
In [28]:
```
Q3 = df["Profit"].quantile(0.75)
Q1 = df["Profit"].quantile(0.25)
IQR = Q3-Q1

Outlier = df[(df["Profit"]< Q1 - 1.5*IQR) | (df['Profit'] > Q3 + 1.5*IQR)]
print(Outlier)
```

	Order_ID	Order_Date	Customer_Name	Customer_ID	Region	State
49	50	2022-10-01	Customer_1368	C8612	South	Karnataka
99	100	2023-06-05	Customer_1936	C8237	West	Gujarat
307	308	2022-10-04	Customer_421	C2703	East	West Bengal
649	650	2023-01-22	Customer_1835	C8764	South	Karnataka
726	727	2022-08-12	Customer_642	C2914	East	Assam
903	904	2023-06-09	Customer_633	C6245	East	Odisha
924	925	2023-11-23	Customer_201	C1161	East	Odisha
1860	1861	2023-08-17	Customer_185	C4938	South	Tamil Nadu
1951	1952	2023-05-28	Customer_298	C3146	South	Kerala
2526	2527	2023-01-01	Customer_1038	C5625	East	Assam
3089	3090	2022-06-03	Customer_1261	C5662	North	Uttar Pradesh
3210	3211	2022-05-22	Customer_1579	C6168	West	Maharashtra
3382	3383	2023-01-11	Customer_1699	C5806	East	Assam
3474	3475	2023-06-01	Customer_654	C1044	West	Goa
3580	3581	2022-05-03	Customer_18	C8472	West	Rajasthan
3835	3836	2022-04-14	Customer_850	C7903	East	Assam
3951	3952	2022-12-28	Customer_1559	C5513	South	Andhra Pradesh
3977	3978	2022-11-18	Customer_787	C5632	South	Kerala
4086	4087	2023-11-12	Customer_1594	C3396	West	Gujarat

	Product_Category	Product_Name	Quantity	Sales	Profit	Year_month
49	Stationery	Pen	9	4940.67	1185.80	2022-10
99	Furniture	Chair	6	4939.49	1197.36	2023-06
307	Stationery	Pencil	9	4917.27	1191.14	2022-10
649	Clothing	Dress	5	4851.07	1207.01	2023-01
726	Furniture	Sofa	1	4894.73	1197.61	2022-08
903	Electronics	Camera	9	4952.64	1235.04	2023-06
924	Furniture	Sofa	7	4899.48	1177.62	2023-11
1860	Clothing	Jacket	3	4969.06	1184.22	2023-08
1951	Stationery	Pencil	4	4900.42	1207.61	2023-05
2526	Stationery	Folder	10	4911.68	1209.41	2023-01
3089	Furniture	Bed	10	4903.07	1169.52	2022-06
3210	Electronics	Laptop	8	4885.11	1176.44	2022-05
3382	Stationery	Folder	4	4749.57	1180.61	2023-01
3474	Electronics	Headphones	8	4788.97	1169.92	2023-06
3580	Clothing	Jacket	3	4887.89	1186.32	2022-05
3835	Stationery	Notebook	9	4930.24	1213.19	2022-04
3951	Stationery	Notebook	10	4768.55	1175.86	2022-12
3977	Stationery	Notebook	8	4945.26	1217.60	2022-11
4086	Stationery	Folder	8	4922.74	1171.07	2023-11

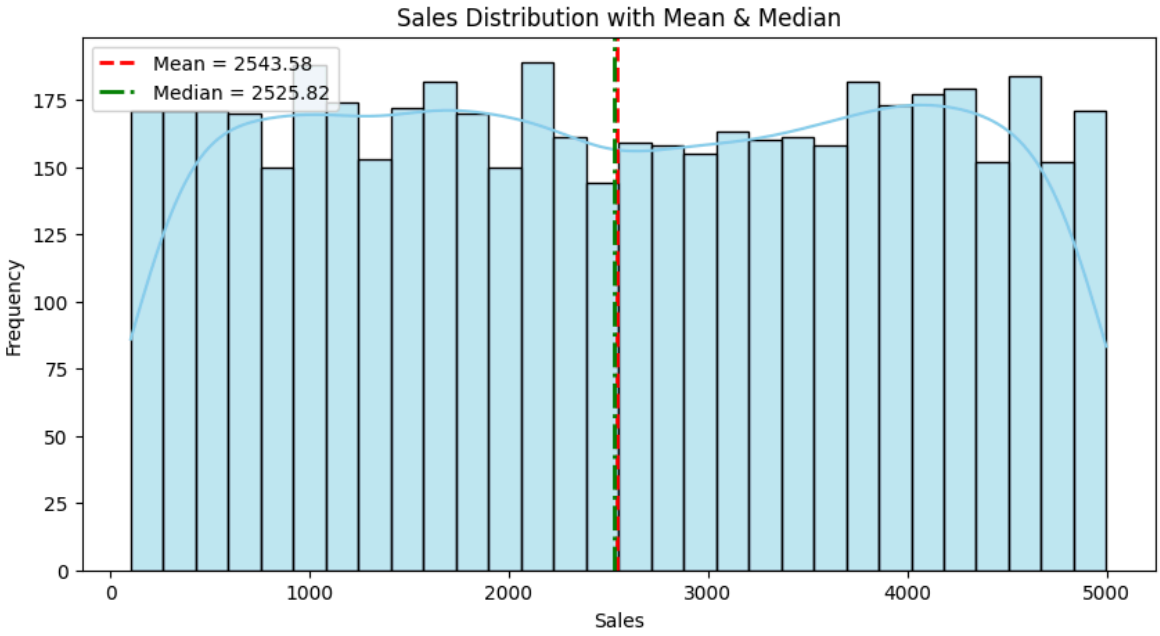
Boxplot chart: Profit Distribution with Outlier

In [38]:
```
sns.boxplot(y=df["Profit"])
plt.title("Profit Distribution with Outliers")
plt.show()
```



Histogram chart: Mean and Median for Sales column

In [34]:
```
plt.figure(figsize=(10,5))
sns.histplot(df["Sales"], bins=30, kde=True, color="skyblue")
mean_sales = df["Sales"].mean()
median_sales = df["Sales"].median()
plt.axvline(mean_sales, color="red", linestyle="--", linewidth=2, label=f"Mean =
plt.axvline(median_sales, color="green", linestyle="-.", linewidth=2, label=f"Me

plt.title("Sales Distribution with Mean & Median")
plt.xlabel("Sales")
plt.ylabel("Frequency")
plt.legend()
plt.show()
```



Step 6: Story

Smartphone leads sales with a total of 111,768.73.

South region is the biggest sales contributor with total sales of 3,315,243.07.

Month-on-Month (MoM) growth is inconsistent, as shown in the line chart.

Karnataka is the highest profit-earning state.

Stationery is the top-selling category, contributing 26.3% of overall sales.

There are no outliers in Sales, indicating that total revenue from transactions is fairly stable with no extreme values.

Profit shows outliers, meaning that while sales are stable, profit margins vary widely across transactions.

The mean and median are almost equal, indicating that the data distribution is roughly symmetric.