

```
In [3]: # Importing Libraries
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [4]: sheets = pd.read_excel("C:/Users/Admin/3D Objects/Sale_Region.xlsx", sheet_name=None)
```

```
In [5]: sheets
```

```

Out[5]: {'sales':   Order No Order Date Customer Index    Channel  Quantity  Unit Pri
ce \
  0  ORD1179 2022-01-02          896 Wholesale      61  1334.63
  1  ORD1302 2022-01-05          353 Wholesale      76  3128.87
  2  ORD1162 2022-01-07          616 Export        58  2275.95
  3  ORD1420 2022-01-08          463 Distributor   42  7519.58
  4  ORD1031 2022-01-14          854 Export        5  2344.04
  ..
  ...
  ...
  ...
  495 ORD1385 2025-12-11          766 Wholesale      66  3000.24
  496 ORD1061 2025-12-13          866 Distributor   17  7327.82
  497 ORD1428 2025-12-16          669 Distributor   81  4428.93
  498 ORD1232 2025-12-20          651 Wholesale      83  4107.58
  499 ORD1210 2025-12-31          163 Wholesale      74  4044.58

      Line Total  Total Unit Cost
  0     81412.43      5866.40
  1     237794.12      4617.59
  2     132005.10      7257.34
  3     315822.36      6049.77
  4     11720.20       5158.61
  ..
  ...
  495  198015.84      6991.87
  496  124572.94      6758.04
  497  358743.33      5973.52
  498  340929.14      4313.90
  499  299298.92      2368.88

[500 rows x 8 columns],
'company':   Company No  Company Name
  0        101  Maruti Suzuki
  1        103  Axis Bank
  2        103  Dabur India
  3        105  Asian Paints
  4        107  Bajaj Auto
  ..
  ...
  495      988  Dabur India
  496      988  Infosys
  497      994  JSW Steel
  498      997  ITC Limited
  499      999  Bharti Airtel

[500 rows x 2 columns],
'state':      id         name    State State Code  County  Type \
  0      1  Sai Verma      Delhi      DL  New Delhi  Town
  1      2  Vivaan Kumar    Punjab      PB  Ludhiana  City
  2      3  Vivaan Nair     Kerala      KL  Kochi     Town
  3      4  Sai Reddy     Uttar Pradesh  UP  Lucknow   Town
  4      5  Aarav Kumar      Delhi      DL  New Delhi  Town
  ..
  ...
  ...
  495  496  Vihaan Chopra  Uttar Pradesh  UP  Lucknow   City
  496  497  Dhruv Singh    Karnataka    KA  Bangalore Other
  497  498  Aryan Reddy    Rajasthan    RJ  Jaipur    City
  498  499  Krishna Chopra  Delhi      DL  New Delhi  Other
  499  500  Dhruv Verma    Karnataka    KA  Bangalore Town

          Latitude  Area Code Population Median Income Households  bvgLongitude

```

```

0    13.9850      629    436693      91786     1836    -96.6433
1    28.0420      986    680392    118492     2742   -91786.0000
2    22.1634      133    432172      58629     8733  -118492.0000
3    18.0214      555    644820      44686      220  -58629.0000
4    27.5065      745    227755      29399     2420  -44686.0000
..    ...
495   22.0294     181    238577      39661     9969 -134190.0000
496   15.4157     722    726217    118441     6560 -39661.0000
497   13.3922     366    544179      65473     2941 -118441.0000
498   8.8449      234    899203    140295     4300 -65473.0000
499   14.3225     157    332872    122624     3842 -140295.0000

```

[500 rows x 12 columns],  
'product': Index Product

```

0      1 Product E
1      2 Product A
2      3 Product C
3      4 Product E
4      5 Product A
..    ...
495   496 Product E
496   497 Product D
497   498 Product D
498   499 Product B
499   500 Product A

```

[500 rows x 2 columns],  
'region': State State Code Region

```

0      Delhi      DL North-West
1      Punjab     PB   East
2      Kerala     KL North-East
3  Uttar Pradesh UP   South
4      Delhi      DL   East
..    ...
495  Uttar Pradesh UP North-East
496      Karnataka KA North-West
497      Rajasthan RJ   Central
498      Delhi      DL   East
499      Karnataka KA   Central

```

[500 rows x 3 columns],  
'Sheet1': Empty DataFrame  
Columns: []  
Index: []}

In [6]: df\_sales=sheets['sales']  
df\_company=sheets['company']  
df\_product=sheets['product']  
df\_state=sheets['state']  
df\_region=sheets['region']

In [7]: df\_sales.head(5)

Out[7]:

	Order No	Order Date	Customer Index	Channel	Quantity	Unit Price	Line Total	Total Unit Cost
0	ORD1179	2022-01-02	896	Wholesale	61	1334.63	81412.43	5866.40
1	ORD1302	2022-01-05	353	Wholesale	76	3128.87	237794.12	4617.59
2	ORD1162	2022-01-07	616	Export	58	2275.95	132005.10	7257.34
3	ORD1420	2022-01-08	463	Distributor	42	7519.58	315822.36	6049.77
4	ORD1031	2022-01-14	854	Export	5	2344.04	11720.20	5158.61

In [8]:

```
# Assign dataframes to each sheet

print("df_sales shape:",(df_sales.shape))
print("df_company shape:",(df_company.shape))
print("df_product shape:",(df_product.shape))
print("df_state shape:",(df_state.shape))
print("df_region shape:",(df_region.shape))
```

df\_sales shape: (500, 8)  
df\_company shape: (500, 2)  
df\_product shape: (500, 2)  
df\_state shape: (500, 12)  
df\_region shape: (500, 3)

In [9]:

```
df_company.head(5)
```

Out[9]:

	Company No	Company Name
0	101	Maruti Suzuki
1	103	Axis Bank
2	103	Dabur India
3	105	Asian Paints
4	107	Bajaj Auto

In [10]:

```
df_product.head(5)
```

```
Out[10]:   Index  Product
```

0	1	Product E
1	2	Product A
2	3	Product C
3	4	Product E
4	5	Product A

```
In [11]: df_state.head(5)
```

```
Out[11]:
```

	<b>id</b>	<b>name</b>	<b>State</b>	<b>State Code</b>	<b>County</b>	<b>Type</b>	<b>Latitude</b>	<b>Area Code</b>	<b>Population</b>	<b>Median Income</b>	<b>House</b>
0	1	Sai Verma	Delhi	DL	New Delhi	Town	13.9850	629	436693	91786	
1	2	Vivaan Kumar	Punjab	PB	Ludhiana	City	28.0420	986	680392	118492	
2	3	Vivaan Nair	Kerala	KL	Kochi	Town	22.1634	133	432172	58629	
3	4	Sai Reddy	Uttar Pradesh	UP	Lucknow	Town	18.0214	555	644820	44686	
4	5	Aarav Kumar	Delhi	DL	New Delhi	Town	27.5065	745	227755	29399	



```
In [12]: df_region.head(5)
```

```
Out[12]:
```

	<b>State</b>	<b>State Code</b>	<b>Region</b>
0	Delhi	DL	North-West
1	Punjab	PB	East
2	Kerala	KL	North-East
3	Uttar Pradesh	UP	South
4	Delhi	DL	East

```
In [13]: df_sales.isnull().sum()
```

```
Out[13]: Order No      0  
Order Date      0  
Customer Index   0  
Channel          0  
Quantity         0  
Unit Price       0  
Line Total        0  
Total Unit Cost   0  
dtype: int64
```

```
In [14]: df_company.isnull().sum()
```

```
Out[14]: Company No      0  
Company Name      0  
dtype: int64
```

```
In [15]: df_product.isnull().sum()
```

```
Out[15]: Index      0  
Product      0  
dtype: int64
```

```
In [16]: df_state.isnull().sum()
```

```
Out[16]: id      0  
name      0  
State      0  
State Code  0  
County      0  
Type       0  
Latitude    0  
Area Code   0  
Population  0  
Median Income 0  
Households  0  
bvgLongtitude 0  
dtype: int64
```

```
In [17]: df_region.isnull().sum()
```

```
Out[17]: State      0  
State Code    0  
Region       0  
dtype: int64
```

```
In [20]: #Data Cleaning
```

```
df=df_sales.merge(  
    df_company,  
    how='left',  
    left_on='Customer Index',  
    right_on='Company No'  
)
```

```
In [21]: df.head(5)
```

Out[21]:

	Order No	Order Date	Customer Index	Channel	Quantity	Unit Price	Line Total	Total Unit Cost	Company Nc
0	ORD1179	2022-01-02	896	Wholesale	61	1334.63	81412.43	5866.40	896.0
1	ORD1302	2022-01-05	353	Wholesale	76	3128.87	237794.12	4617.59	NaN
2	ORD1162	2022-01-07	616	Export	58	2275.95	132005.10	7257.34	NaN
3	ORD1420	2022-01-08	463	Distributor	42	7519.58	315822.36	6049.77	463.0
4	ORD1031	2022-01-14	854	Export	5	2344.04	11720.20	5158.61	NaN



In [23]:

```
df=df.merge(  
    df_product,  
    how='left',  
    left_on='Customer Index',  
    right_on='Index'  
)
```

In [26]:

```
df.head(5)
```

Out[26]:

	Order No	Order Date	Customer Index	Channel	Quantity	Unit Price	Line Total	Total Unit Cost	Company Nc
0	ORD1179	2022-01-02	896	Wholesale	61	1334.63	81412.43	5866.40	896.0
1	ORD1302	2022-01-05	353	Wholesale	76	3128.87	237794.12	4617.59	NaN
2	ORD1162	2022-01-07	616	Export	58	2275.95	132005.10	7257.34	NaN
3	ORD1420	2022-01-08	463	Distributor	42	7519.58	315822.36	6049.77	463.0
4	ORD1031	2022-01-14	854	Export	5	2344.04	11720.20	5158.61	NaN



In [ ]:

```
# EDA
```

In [36]:

```
sheets = pd.read_excel("C:/Users/Admin/3D Objects/Sales Datas.xlsx", sheet_name=None)
```

In [37]:

```
sheets
```

```
Out[37]: {'Sales EDA':      Ord No    Ord Date Ord Month    Company Name     Channel   Pro
duct \
 0  ORD1179 2022-01-02      Jan  Maruti Suzuki    Wholesale  Product E
 1  ORD1302 2022-01-05      Jan  Axis Bank    Wholesale  Product A
 2  ORD1162 2022-01-07      Jan  Dabur India    Export    Product C
 3  ORD1420 2022-01-08      Jan  Asian Paints  Distributor Product E
 4  ORD1031 2022-01-14      Jan  Bajaj Auto    Export    Product A
 ..
 495 ORD1385 2025-12-11      Dec  Dabur India    Wholesale  Product E
 496 ORD1061 2025-12-13      Dec  Infosys    Distributor Product D
 497 ORD1428 2025-12-16      Dec  JSW Steel    Distributor Product D
 498 ORD1232 2025-12-20      Dec  ITC Limited  Wholesale  Product B
 499 ORD1210 2025-12-31      Dec  Bharti Airtel Wholesale  Product A

      Quantity  Unit Price    Revenue    Cost    Region    State \
 0       61    1334.63  81412.43  5866.40 North-West  Delhi
 1       76    3128.87 237794.12  4617.59    East  Punjab
 2       58    2275.95 132005.10  7257.34 North-East  Kerala
 3       42    7519.58 315822.36  6049.77    South Uttar Pradesh
 4        5    2344.04  11720.20  5158.61    East  Delhi
 ..
 495      66    3000.24 198015.84  6991.87 North-East Uttar Pradesh
 496      17    7327.82 124572.94  6758.04 North-West Karnataka
 497      81    4428.93 358743.33  5973.52    Central Rajasthan
 498      83    4107.58 340929.14  4313.90    East  Delhi
 499      74    4044.58 299298.92  2368.88    Central Karnataka

      County  Latitude Longtitude Total cost    Profit Payment Mode
 0  New Delhi  13.9850 -96.6433 357850.40 276437.97      EMI
 1  Ludhiana  28.0420 -91786.0000 350936.84 113142.72 Net Banking
 2   Kochi  22.1634 -118492.0000 420925.72 288920.62 Wire Transfer
 3  Lucknow  18.0214 -58629.0000 254090.34 -61732.02      EMI
 4  New Delhi  27.5065 -44686.0000 25793.05 14072.85      EMI
 ..
 495  Lucknow  22.0294 -134190.0000 461463.42 263447.58      EMI
 496  Bangalore  15.4157 -39661.0000 114886.68 -9686.26      UPI
 497   Jaipur  13.3922 -118441.0000 483855.12 125111.79 Wire Transfer
 498  New Delhi  8.8449 -65473.0000 358053.70 17124.56 Credit Card
 499  Bangalore  14.3225 -140295.0000 175297.12 -124001.80      UPI

 [500 rows x 18 columns]}
```

```
In [4]: import pandas as pd

df_Sales_EDA = pd.read_excel("C:/Users/Admin/3D Objects/Sales Data.xlsx")

print("df_Sales_EDA shape:", df_Sales_EDA.shape)

df_Sales_EDA shape: (500, 18)
```

```
In [5]: df_Sales_EDA.head(10)
```

Out[5]:

	Ord No	Ord Date	Ord Month	Company Name	Channel	Product	Quantity	Unit Price	Revenue
0	ORD1179	2022-01-02	Jan	Maruti Suzuki	Wholesale	Product E	61	1334.63	81412.43
1	ORD1302	2022-01-05	Jan	Axis Bank	Wholesale	Product A	76	3128.87	237794.12
2	ORD1162	2022-01-07	Jan	Dabur India	Export	Product C	58	2275.95	132005.10
3	ORD1420	2022-01-08	Jan	Asian Paints	Distributor	Product E	42	7519.58	315822.36
4	ORD1031	2022-01-14	Jan	Bajaj Auto	Export	Product A	5	2344.04	11720.20
5	ORD1230	2022-01-15	Jan	Sun Pharma	Wholesale	Product A	79	4149.46	327807.34
6	ORD1390	2022-01-17	Jan	Tata Consultancy Services	Export	Product A	94	3560.44	334681.36
7	ORD1366	2022-01-20	Jan	Maruti Suzuki	Wholesale	Product A	32	6694.34	214218.88
8	ORD1431	2022-01-20	Jan	Mahindra & Mahindra	Wholesale	Product E	76	1116.46	84850.96
9	ORD1149	2022-01-24	Jan	Adani Enterprises	Wholesale	Product B	65	2687.88	174712.20

In [6]:

```
import pandas as pd

# Load the file
df = pd.read_excel("C:/Users/Admin/3D Objects/Sales Datas.xlsx")

# Clean column names
df.columns = df.columns.str.strip().str.lower()

# Show cleaned column names
print("Cleaned column names:", df.columns.tolist())
```

Cleaned column names: ['ord no', 'ord date', 'ord month', 'company name', 'channel', 'product', 'quantity', 'unit price', 'revenue', 'cost', 'region', 'state', 'county', 'latitude', 'longitude', 'total cost', 'profit', 'payment mode']

In [73]:

```
# Clean column names
df.columns = df.columns.str.strip().str.lower()

# Convert 'ord date' to datetime
df['ord date'] = pd.to_datetime(df['ord date'])

# Extract order month as period (e.g., 2022-01)
```

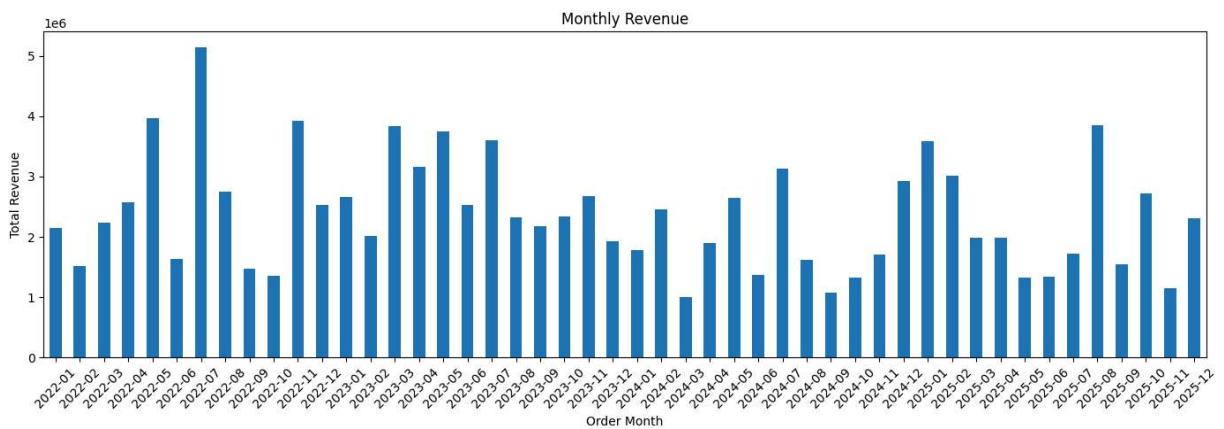
```

df['order_month'] = df['ord date'].dt.to_period('M')

# Group by order month and calculate total revenue
monthly_sales = df.groupby('order_month')['revenue'].sum()

# Plot
plt.figure(figsize=(14, 5))
monthly_sales.plot(kind='bar')
plt.title("Monthly Revenue")
plt.xlabel("Order Month")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

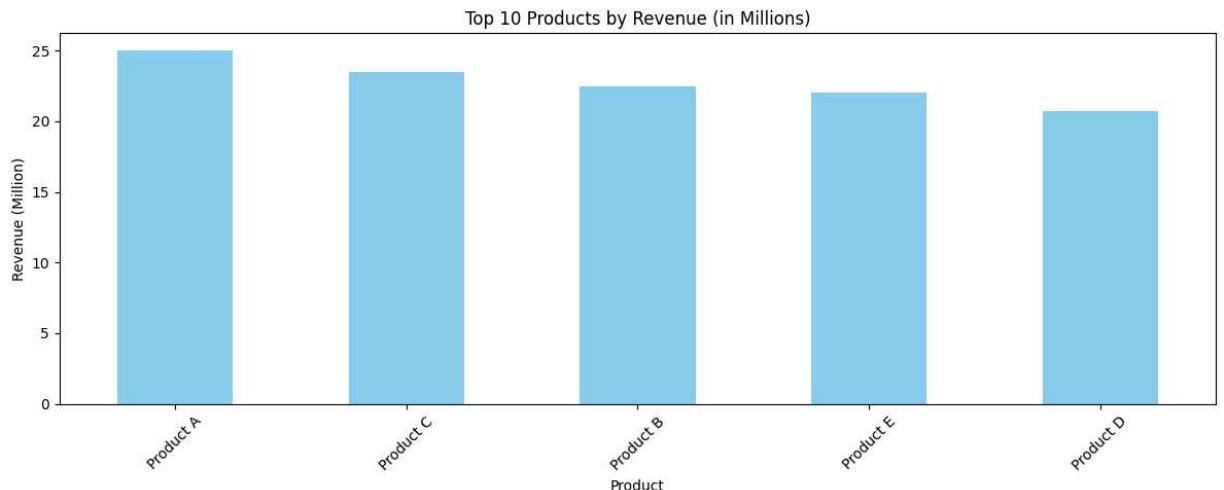
```



```

In [76]: top_prod = df.groupby('product')['revenue'].sum() / 1_000_000
top_prod = top_prod.nlargest(10)
import matplotlib.pyplot as plt
plt.figure(figsize=(12, 5))
top_prod.plot(kind='bar', color='skyblue')
plt.title("Top 10 Products by Revenue (in Millions)")
plt.xlabel("Product")
plt.ylabel("Revenue (Million)")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

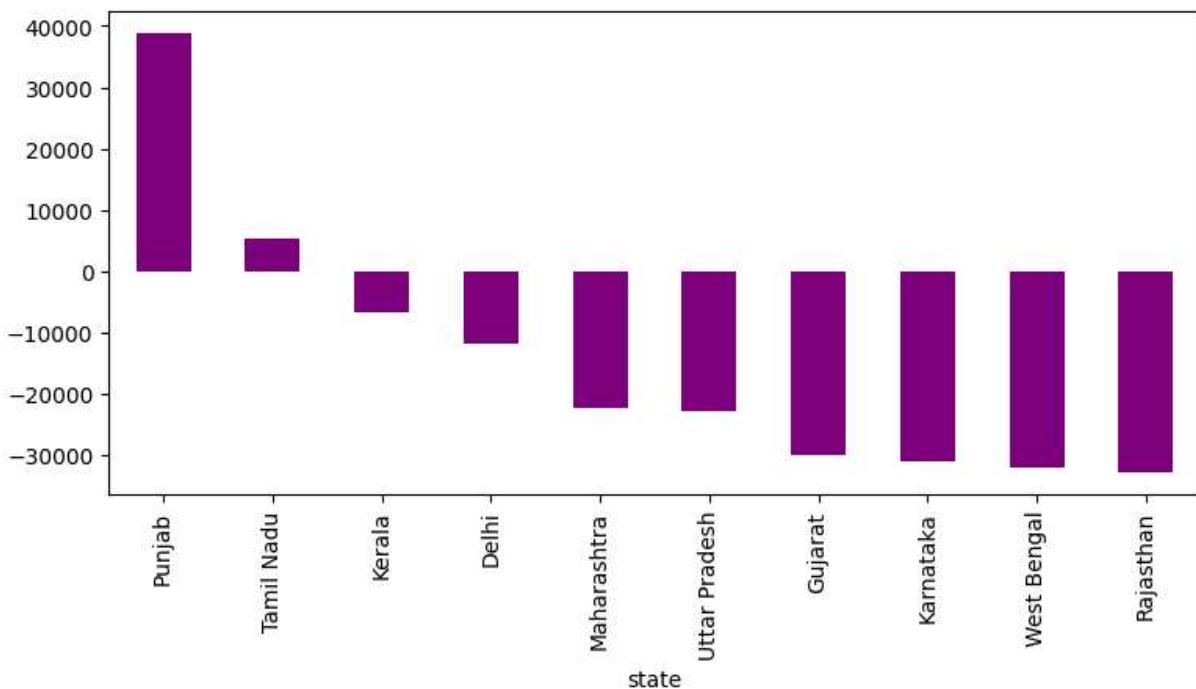
```



```
In [80]: top_states_by_profit = (
    df.groupby('state')['profit']
    .mean()
    .sort_values(ascending=False)
    .head(10)
)
```

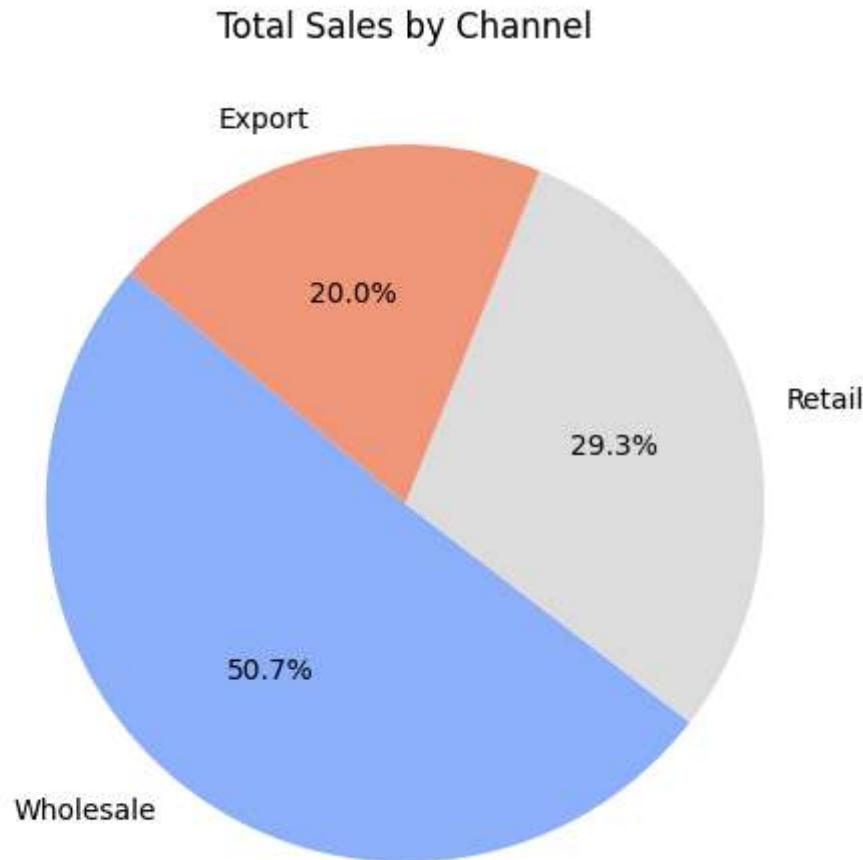
```
plt.figure(figsize=(9, 4))
top_states_by_profit.plot(kind='bar', color='purple')
plt.title
```

```
Out[80]: <function matplotlib.pyplot.title(label: 'str', fontdict: 'dict[str, Any] | None' = None, loc: "Literal['left', 'center', 'right'] | None" = None, pad: 'float | None' = None, *, y: 'float | None' = None, **kwargs) -> 'Text'>
```



```
In [81]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = {
    'channel': ['Retail', 'Wholesale', 'Export', 'Retail', 'Wholesale'],
    'revenue': [1000, 2000, 1500, 1200, 1800]
}
df = pd.DataFrame(data)
chan_sales = df.groupby('channel')['revenue'].sum().sort_values(ascending=False)
plt.figure(figsize=(5, 5))
plt.pie(
    chan_sales.values,
    labels=chan_sales.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=sns.color_palette('coolwarm', len(chan_sales)) # Match number of colors
)
plt.title('Total Sales by Channel')
```

```
plt.tight_layout()  
plt.show()
```



```
In [48]: top_rev = (  
    df.groupby('Company Name')['Revenue']  
        .sum()  
        .sort_values(ascending=False)  
        .head(10)  
)  
bottom_rev = (  
    df.groupby('Company Name')['Revenue']  
        .sum()  
        .sort_values(ascending=True)  
        .head(10)  
)  
fig, axes = plt.subplots(1, 2, figsize=(16, 5))  
  
# Plot 1: Top 10 customers by revenue (converted to millions)  
sns.barplot(  
    x=top_rev.values / 1e6,      # X-axis: revenue in millions  
    y=top_rev.index,            # Y-axis: customer names  
    palette='Blues_r',          # Color palette (reversed blues)  
    ax=axes[0]                  # Draw on the left subplot  
)  
axes[0].set_title('Top 10 Customers by Revenue', fontsize=14)  
axes[0].set_xlabel('Revenue (Million)', fontsize=12)
```

```

axes[0].set_ylabel('Company Name', fontsize=12)

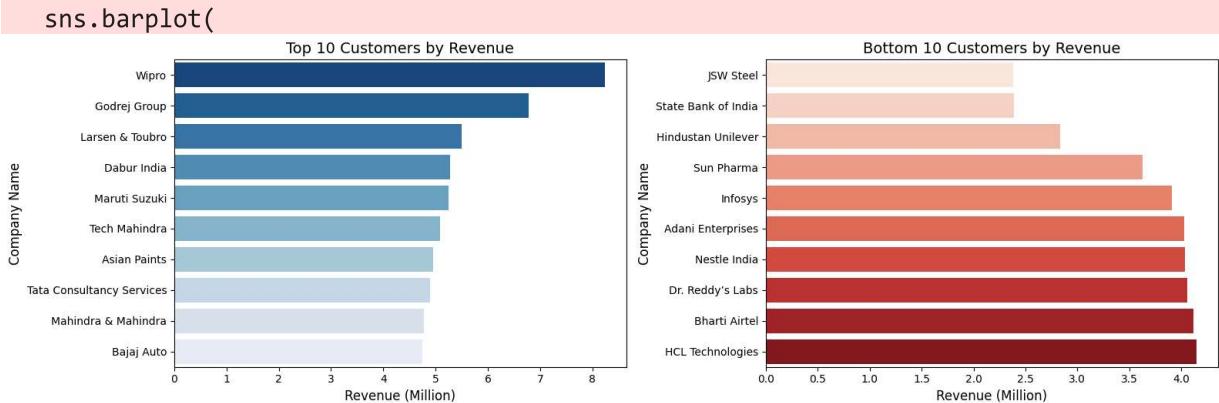
# Plot 2: Bottom 10 customers by revenue (converted to millions)
sns.barplot(
    x=bottom_rev.values / 1e6,
    y=bottom_rev.index,
    palette='Reds',
    ax=axes[1]
)
axes[1].set_title('Bottom 10 Customers by Revenue', fontsize=14)
axes[1].set_xlabel('Revenue (Million)', fontsize=12)
axes[1].set_ylabel('Company Name', fontsize=12)

# Adjust layout and display both charts
plt.tight_layout()
plt.show()

```

C:\Users\Admin\AppData\Local\Temp\ipykernel\_13332\2591065423.py:16: FutureWarning:  
 Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1  
 4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

C:\Users\Admin\AppData\Local\Temp\ipykernel\_13332\2591065423.py:27: FutureWarning:  
 Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1  
 4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.



In [49]:

```

state_rev = df.groupby('State').agg(
    revenue=('Revenue', 'sum'), # Sum revenue for each state
    orders=('Ord No', 'nunique') # Count unique order numbers
).sort_values('revenue', ascending=False).head(10) # Top 10 states by revenue
plt.figure(figsize=(15, 5))
sns.barplot(
    x=state_rev.index, # State names
    y=state_rev['revenue'] / 1e6, # Revenue in millions
    palette='Greens'
)
plt.title("Top 10 States by Revenue", fontsize=14)
plt.xlabel("State", fontsize=12)
plt.ylabel("Revenue (Million)", fontsize=12)
plt.xticks(rotation=45)

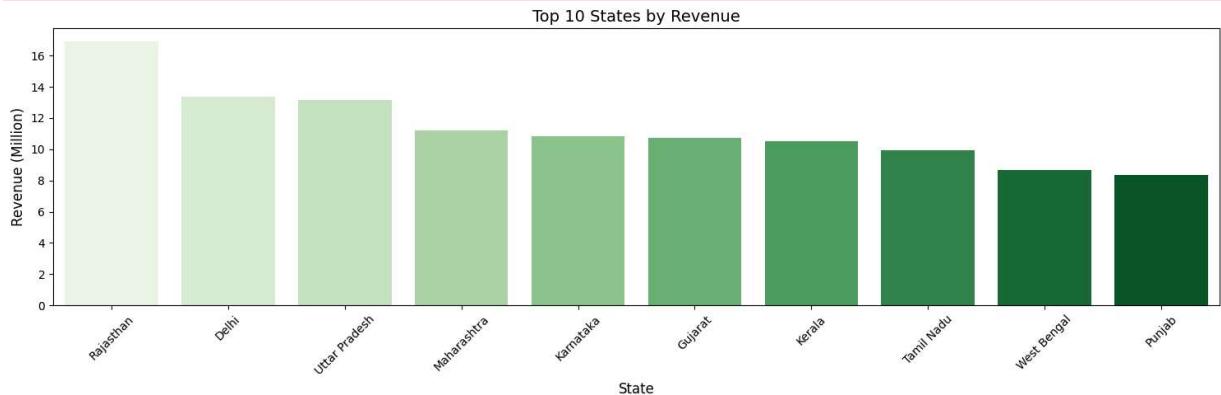
```

```
plt.tight_layout()  
plt.show()
```

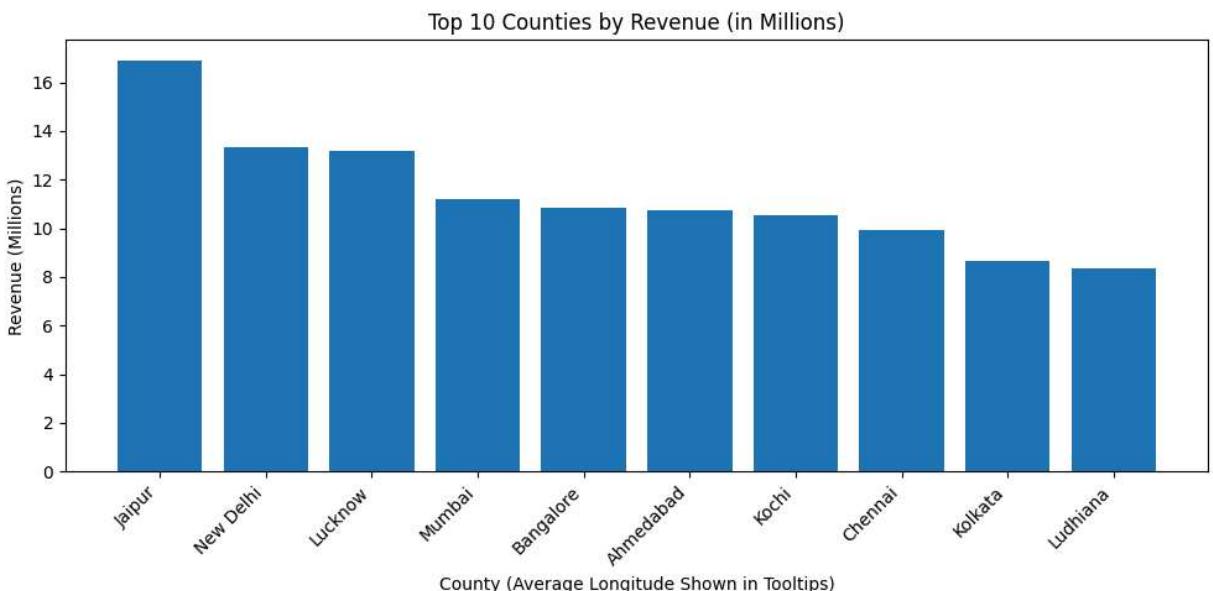
C:\Users\Admin\AppData\Local\Temp\ipykernel\_13332\1838061161.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1  
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

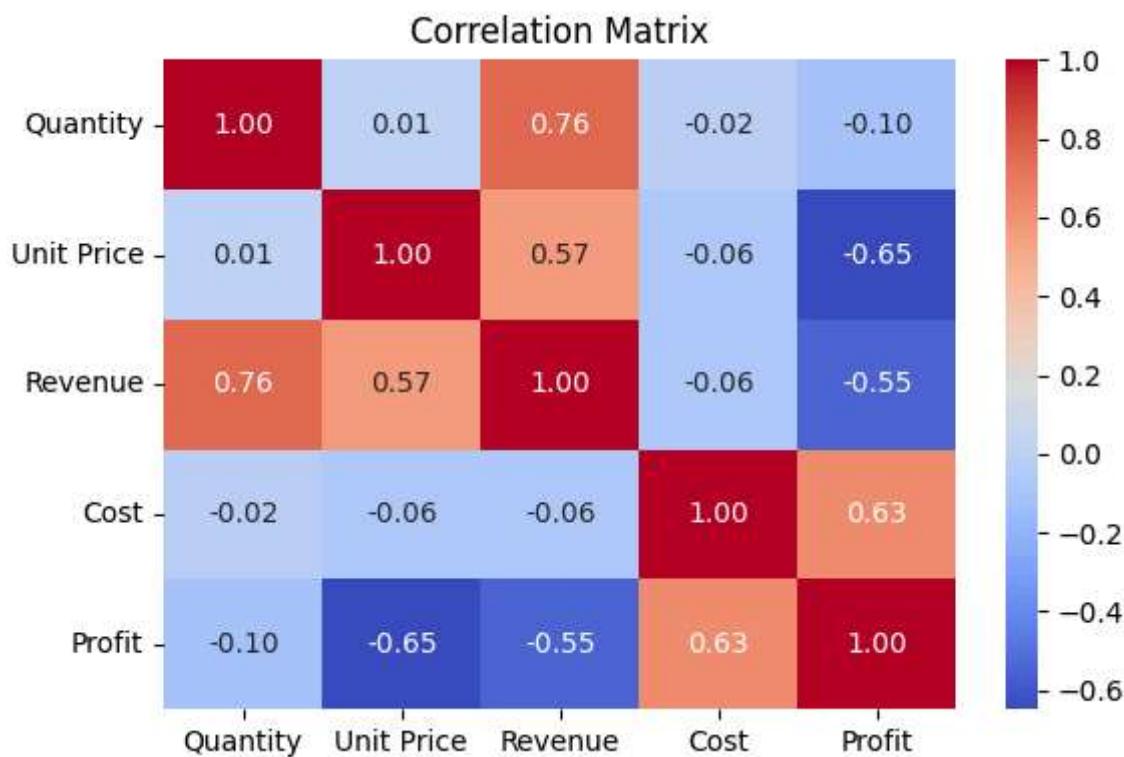
```
sns.barplot(
```



```
In [50]: county_summary = df.groupby('County').agg(  
    total_revenue=('Revenue', 'sum'),  
    avg_longitude=('Longitude', 'mean'))  
)  
county_summary['total_revenue_m'] = county_summary['total_revenue'] / 1e6  
top_counties = county_summary.sort_values(by='total_revenue', ascending=False).head(10)  
plt.figure(figsize=(10, 5))  
plt.bar(top_counties.index, top_counties['total_revenue_m'])  
plt.title('Top 10 Counties by Revenue (in Millions)')  
plt.xlabel('County (Average Longitude Shown in Tooltips)')  
plt.ylabel('Revenue (Millions)')  
plt.xticks(rotation=45, ha='right')  
plt.tight_layout()
```



```
In [51]: num_cols = ['Quantity', 'Unit Price', 'Revenue', 'Cost', 'Profit'] # Corrected col  
corr = df[num_cols].corr()  
import seaborn as sns  
import matplotlib.pyplot as plt  
plt.figure(figsize=(6, 4))  
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")  
plt.title("Correlation Matrix")  
plt.tight_layout()  
plt.show()
```



```
In [ ]:
```