Importing Necessary Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express.express as px
%matplotlib inline
```

Importing the Dataset CSV into a Pandas Dataframe

```
df = pd.read_csv('/content/drive/MyDrive/Superstore.csv')
df.head(10)
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Ship Date	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	6/27/2010	8923	255.28	159.42	2277814.44	1422416.66	855397.78
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/12/2012	8/30/2012	844090	205.70	117.11	173579828.30	98846590.90	74733237.40
2	Europe	The Netherlands	Office Supplies	Online	L	5/2/2014	5/16/2014	179	651.21	524.96	116556.09	93946.84	22609.25
3	Sub-Saharan Africa	Sao Tome and Principe	Fruit	Online	M	6/20/2014	8/04/2014	9925	9.33	6.92	92585.25	68674.00	23911.25
4	Sub-Saharan Africa	Namibia	Office Supplies	Online	L	2/12/2012	3/18/2012	2321	651.21	524.96	1511384.41	1218172.16	293212.25
5	Sub-Saharan Africa	Mali	Clothes	Online	M	7/24/2011	7/29/2011	9243	109.28	35.84	1009979.04	331267.12	678711.92
6	Asia	Bangladesh	Baby Food	Offline	H	10/18/2015	11/07/2015	8267	255.28	159.42	2110335.76	1317911.14	792424.62
7	Middle East and North Africa	Algeria	Household	Online	H	10/26/2013	11/26/2013	6687	668.27	502.54	4468176.49	3360000.98	1108175.51
8	North America	Mexico	Personal Care	Offline	M	7/30/2010	8/28/2010	5783	81.73	56.67	472604.59	327697.61	144906.98
9	Sub-Saharan Africa	Mozambique	Household	Offline	M	10/17/2013	11/05/2013	7799	668.27	502.54	5211109.73	3919184.46	1291925.27

10 rows × 14 columns

```
df.shape
```

```
(100000, 14)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 14 columns):
 0   Region          100000 non-null  object
 1   Country         100000 non-null  object
 2   Item Type       100000 non-null  object
 3   Sales Channel   100000 non-null  object
 4   Order Priority  100000 non-null  object
 5   Order Date      100000 non-null  object
 6   Ship Date       100000 non-null  object
 7   Units Sold      100000 non-null  int64
 8   Unit Price      100000 non-null  float64
 9   Unit Cost       100000 non-null  float64
 10  Total Revenue   100000 non-null  float64
 11  Total Cost      100000 non-null  float64
 12  Total Profit    100000 non-null  float64
dtypes: int64, float64, object
memory usage: 10.7 MB
```

```
df.describe()
```

	Units Sold	Unit Price	Unit Cost	Total Revenue	Total Cost	Total Profit
count	100000.00000	100000.00000	100000.00000	1.000000e+06	1.000000e+06	1.000000e+06
mean	5000.50000	267.07320	188.07308	1.336537e+06	9.414674e+05	3.921863e+05
std	2886.89568	216.87974	164.46862	1.251162e+06	8.810084e+05	3.670788e+05
min	1.00000	9.33000	6.92000	9.258500e+03	6.867400e+03	2.391100e+03
25%	2501.00000	109.28000	35.84000	3.397134e+05	1.072671e+05	7.445426e+04
50%	5000.50000	205.70000	117.11000	8.783771e+05	5.823176e+05	2.844096e+05
75%	7500.00000	437.20000	351.21000	1.966824e+06	1.580226e+06	6.387035e+05
max	10000.00000	668.27000	524.96000	6.651099e+06	5.248023e+06	1.662886e+06

```
df.describe(include='object')
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Ship Date
count	100000	100000	100000	100000	100000	100000	100000
unique	7	185	12	2	4	2372	2401
top	Sub-Saharan Africa	Cyprus	Cosmetics	Online	C	2/1/2012	2/12/2012
freq	26671	1000	8445	50000	25000	100	100

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

```
Region            0
Country           0
Item Type         0
Sales Channel     0
Order Priority    0
Order Date        0
Ship Date         0
Units Sold        0
Unit Price        0
Unit Cost         0
Total Revenue     0
Total Cost        0
Total Profit      0
dtype: int64
```

```
df.isnull().sum()/100*100
```

```
Region            0.0
Country           0.0
Item Type         0.0
Sales Channel     0.0
Order Priority    0.0
Order Date        0.0
Ship Date         0.0
Units Sold        0.0
Unit Price        0.0
Unit Cost         0.0
Total Revenue     0.0
Total Cost        0.0
Total Profit      0.0
dtype: float64
```

```
df.drop_duplicates(inplace=True)
```

```
df.shape
```

```
(100000, 14)
```

```
df.rename(columns={'Units Sold':'Units_Sold', 'Unit Price':'Unit_Price', 'Unit Cost':'Unit_Cost', 'Total Revenue':'Total_Revenue', 'Total Cost':'Total_Cost', 'Total Profit':'Total_Profit'},inplace=True)
```

```
df.head(2)
```

	Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Ship Date	Units_Sold	Unit_Price	Unit_Cost	Total_Revenue	Total_Cost	Total_Profit
0	Australia and Oceania	Tuvalu	Baby Food	Offline	H	5/28/2010	6/27/2010	8923	255.28	159.42	2277814.44	1422416.66	855397.78
1	Central America and the Caribbean	Grenada	Cereal	Online	C	8/12/2012	8/30/2012	844090	205.70	117.11	173579828.30	98846590.90	74733237.40

```
df.columns
```

```
Index(['Region', 'Country', 'Item Type', 'Sales Channel', 'Order Priority',
       'Order Date', 'Ship Date', 'Units_Sold', 'Unit_Price', 'Unit_Cost',
       'Total_Revenue', 'Total_Cost', 'Total_Profit'],
      dtype='object')
```

```
df.dtypes
```

```
Region            object
Country           object
Item Type         object
Sales Channel     object
Order Priority    object
Order Date        object
Ship Date         object
Units_Sold        int64
Unit_Price        float64
Unit_Cost         float64
Total_Revenue     float64
Total_Cost        float64
Total_Profit      float64
dtype: object
```

```
df.nunique()
```

```
Region            7
Country         185
Item Type        12
Sales Channel     2
Order Priority    4
Order Date     2372
Ship Date      2401
Units_Sold     9990
Unit_Price     8064
Unit_Cost      7603
Total_Revenue  99456
Total_Cost     99456
Total_Profit   99456
dtype: int64
```

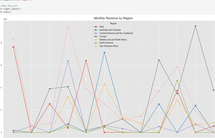
```
df.groupby('Item Type').count
```

```
```

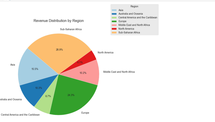
```
plt.title('Monthly Revenue and Profit')
plt.legend()
plt.xticks(rotation=45)
# Adjust layout to prevent clipping of messages
plt.tight_layout()

# Show the plot
plt.show()
```

```
# Group by item_type and calculate the sum of total_profit and total_revenue
grouped_data = df.groupby('item_type')[['total_profit', 'total_revenue']].sum()

# Sort the data by total_revenue in descending order
grouped_data = grouped_data.sort_values('total_revenue', ascending=False)

# Plot the data
plt.figure(figure=(12, 6))
ax = grouped_data.plot(kind='bar')

# Add labels and title
plt.xlabel('Item Type')
plt.ylabel('Amount')
plt.title('Total Revenue and Total Profit by Item Type')
plt.legend(['Total Revenue', 'Total Profit'])

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Adjust layout to prevent clipping of messages
plt.tight_layout()

# Show the plot
plt.show()
```

```
region_pivot.sort_values(ascending=False).plot(kind='barh', figsize=(12,6), cmap='viridis')
plt.xlabel('Revenue')
plt.ylabel('Region')
plt.title('Monthly Revenue by Region')
plt.legend(bbox_to_anchor=(1.05, 1), loc='upper left', title='Region')
plt.tight_layout()
plt.show()

# Convert the index to datetime
monthly_revenue.index = pd.to_datetime(monthly_revenue.index)

# Create the line plot
plt.figure(figure=(12, 6))
plt.plot(monthly_revenue.index, monthly_revenue.values)

# Add labels and title
plt.xlabel('Month')
plt.ylabel('Total Revenue')
plt.title('Monthly Revenue')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Adjust layout to prevent clipping of messages
plt.tight_layout()

# Show the plot
plt.show()
```




```
# Group by region and calculate the percentage
region_revenue = df.groupby('region')['total_revenue'].sum()
total_revenue = region_revenue.sum()
region_percentage = (region_revenue / total_revenue) * 100

# Create the pie chart
plt.figure(figure=(10, 8))
plt.pie(region_percentage, labels=region_percentage.index, autopct='%1.1f%%')

# Add a title
plt.title('Revenue Distribution by Region')

# Ensure the pie chart is a circle
plt.axis('equal')

# Show the plot
plt.show()

# Group by region and calculate the sum of units_sold
region_units = df.groupby('region')['units_sold'].sum()

# Create the bar chart
plt.figure(figure=(12, 6))
plt.bar(region_units.index, region_units.values)

# Add labels and title
plt.xlabel('Region')
plt.ylabel('Units Sold')
plt.title('Units Sold by Region')

# Rotate x-axis labels for better readability
plt.xticks(rotation=45)

# Adjust layout to prevent clipping of messages
plt.tight_layout()

# Show the plot
plt.show()
```



Regional Sales Performance Analysis

1. Overview

The dataset provides information about units sold, total revenue, and total profit across different regions. Here's a summary of the key findings:

Regions Covered:

- Asia
- Australia and Oceania
- Central America and the Caribbean
- Europe
- Middle East and North Africa
- North America
- Sub-Saharan Africa

2. Units Sold

- **Sub-Saharan Africa** has the highest number of units sold (100,075), followed by **Europe** (99,117) and **Australia and Oceania** (99,075).
- **North America** has the lowest number of units sold (50,170).

3. Total Revenue

- **Europe** leads with the highest total revenue (USD 10,060,570), followed by **Sub-Saharan Africa** (USD 10,037,037) and **Asia** (USD 10,007,081).
- **North America** has the lowest total revenue (USD 5,041,554).

4. Total Profit

- **Europe** also has the highest total profit (USD 7,538,038.65), with **Sub-Saharan Africa** in second place (USD 7,517,862.07) and **Asia** in third (USD 7,510,867.47).
- **North America** has the lowest total profit (USD 3,867,164.75).

5. Profitability Analysis

- **Sub-Saharan Africa** has the highest total profit and also the highest units sold, indicating a potentially high profit margin and successful sales strategy.
- **Europe** despite having the second-highest units sold, has the highest total revenue and total profit, suggesting strong market performance and high revenue generation.

6. Key Insights

- **Sub-Saharan Africa** and **Europe** are the top performers in terms of total profit. While **North America** shows relatively lower performance in both revenue and profit, there is an opportunity for growth, with sales and revenue data not linked to profitability, indicating that other factors may play a role.
- **Asia** has a substantial presence in units sold and revenue but does not lead in profitability, indicating that while it sells many units, the profit margin may vary or be low.

7. Recommendations

- **Focus on Europe and Sub-Saharan Africa:** These regions show strong performance in terms of revenue and profit. Strategic efforts should continue in these areas to maintain dominance.
- **Evaluate North American Market Strategy:** Consider reassessing the market strategy for North America to improve sales and profitability.
- **Investigate Profit Margins:** Analyze regions with high units sold but low total profit. May be investigated cost structures and pricing strategies to enhance profitability.

Conclusion

The data provides insights into the dynamics in sales performance across regions. By focusing on high-performing regions and addressing the challenges in lower-performing ones, there is an opportunity to optimize overall sales and profitability.

```
```