

**TASK 1**

Start coding or [generate](#) with AI.

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Set visualization styles
sns.set(style="whitegrid")
```

```
# Load datasets
customers = pd.read_csv("Customers.csv")
products = pd.read_csv("Products.csv")
transactions = pd.read_csv("Transactions.csv")
```

```
# Check for missing values
print("Missing values in Customers dataset:")
print(customers.isnull().sum())
print("\nMissing values in Products dataset:")
print(products.isnull().sum())
print("\nMissing values in Transactions dataset:")
print(transactions.isnull().sum())
```

→ Missing values in Customers dataset:

```
CustomerID      0
CustomerName    0
Region          0
SignupDate      0
dtype: int64
```

Missing values in Products dataset:

```
ProductID      0
ProductName     0
Category       0
Price          0
dtype: int64
```

Missing values in Transactions dataset:

```
TransactionID   0
CustomerID      0
ProductID       0
TransactionDate 0
Quantity        0
TotalValue      0
Price           0
dtype: int64
```

```
# Basic dataset information
print("\nCustomers Dataset Info:")
print(customers.info())
print("\nProducts Dataset Info:")
print(products.info())
print("\nTransactions Dataset Info:")
print(transactions.info())
```

→

Customers Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype
---  -
 0   CustomerID    200 non-null   object
 1   CustomerName  200 non-null   object
 2   Region        200 non-null   object
 3   SignupDate    200 non-null   object
dtypes: object(4)
memory usage: 6.4+ KB
```

None

Products Dataset Info:

&lt;class 'pandas.core.frame.DataFrame'&gt;

RangeIndex: 100 entries, 0 to 99

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	ProductID	100 non-null	object
1	ProductName	100 non-null	object
2	Category	100 non-null	object
3	Price	100 non-null	float64

dtypes: float64(1), object(3)

memory usage: 3.3+ KB

None

Transactions Dataset Info:

&lt;class 'pandas.core.frame.DataFrame'&gt;

RangeIndex: 1000 entries, 0 to 999

Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	TransactionID	1000 non-null	object
1	CustomerID	1000 non-null	object
2	ProductID	1000 non-null	object
3	TransactionDate	1000 non-null	object
4	Quantity	1000 non-null	int64
5	TotalValue	1000 non-null	float64
6	Price	1000 non-null	float64

dtypes: float64(2), int64(1), object(4)

memory usage: 54.8+ KB

None

# Quick overview of datasets

print("\nFirst few rows of Customers dataset:")

print(customers.head())

print("\nFirst few rows of Products dataset:")

print(products.head())

print("\nFirst few rows of Transactions dataset:")

print(transactions.head())



First few rows of Customers dataset:

	CustomerID	CustomerName	Region	SignupDate
0	C0001	Lawrence Carroll	South America	2022-07-10
1	C0002	Elizabeth Lutz	Asia	2022-02-13
2	C0003	Michael Rivera	South America	2024-03-07
3	C0004	Kathleen Rodriguez	South America	2022-10-09
4	C0005	Laura Weber	Asia	2022-08-15

First few rows of Products dataset:

	ProductID	ProductName	Category	Price
0	P001	ActiveWear Biography	Books	169.30
1	P002	ActiveWear Smartwatch	Electronics	346.30
2	P003	ComfortLiving Biography	Books	44.12
3	P004	BookWorld Rug	Home Decor	95.69
4	P005	TechPro T-Shirt	Clothing	429.31

First few rows of Transactions dataset:

	TransactionID	CustomerID	ProductID	TransactionDate	Quantity	\
0	T00001	C0199	P067	2024-08-25 12:38:23	1	
1	T00112	C0146	P067	2024-05-27 22:23:54	1	
2	T00166	C0127	P067	2024-04-25 07:38:55	1	
3	T00272	C0087	P067	2024-03-26 22:55:37	2	
4	T00363	C0070	P067	2024-03-21 15:10:10	3	

	TotalValue	Price
0	300.68	300.68
1	300.68	300.68
2	300.68	300.68
3	601.36	300.68
4	902.04	300.68

# Merge datasets for comprehensive analysis

merged\_data = transactions.merge(customers, on="CustomerID").merge(products, on="ProductID")

# Statistical summaries

print("\nStatistical summary of Transactions dataset:")

print(transactions.describe())



Statistical summary of Transactions dataset:

	Quantity	TotalValue	Price
count	1000.000000	1000.000000	1000.000000
mean	2.537000	689.995560	272.55407
std	1.117981	493.144478	140.73639
min	1.000000	16.080000	16.080000
25%	2.000000	295.295000	147.95000
50%	3.000000	588.880000	299.93000
75%	4.000000	1011.660000	404.40000
max	4.000000	1991.040000	497.76000

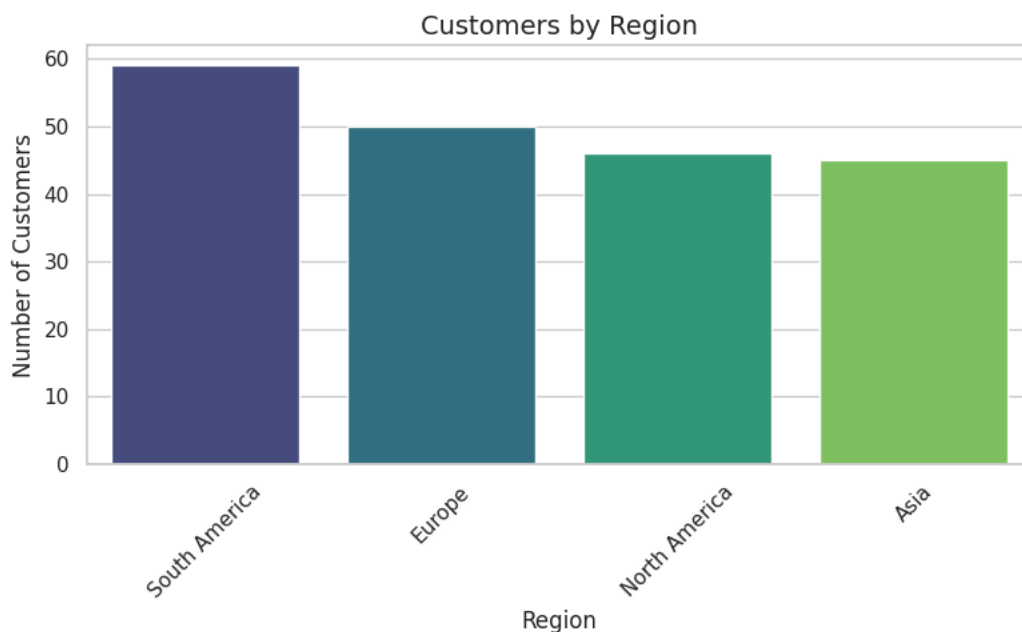
```
# Customers by Region
plt.figure(figsize=(8, 5))
region_counts = customers['Region'].value_counts()
sns.barplot(x=region_counts.index, y=region_counts.values, palette="viridis")
plt.title("Customers by Region", fontsize=14)
plt.xlabel("Region")
plt.ylabel("Number of Customers")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



<ipython-input-28-371504665d29>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le`

```
sns.barplot(x=region_counts.index, y=region_counts.values, palette="viridis")
```

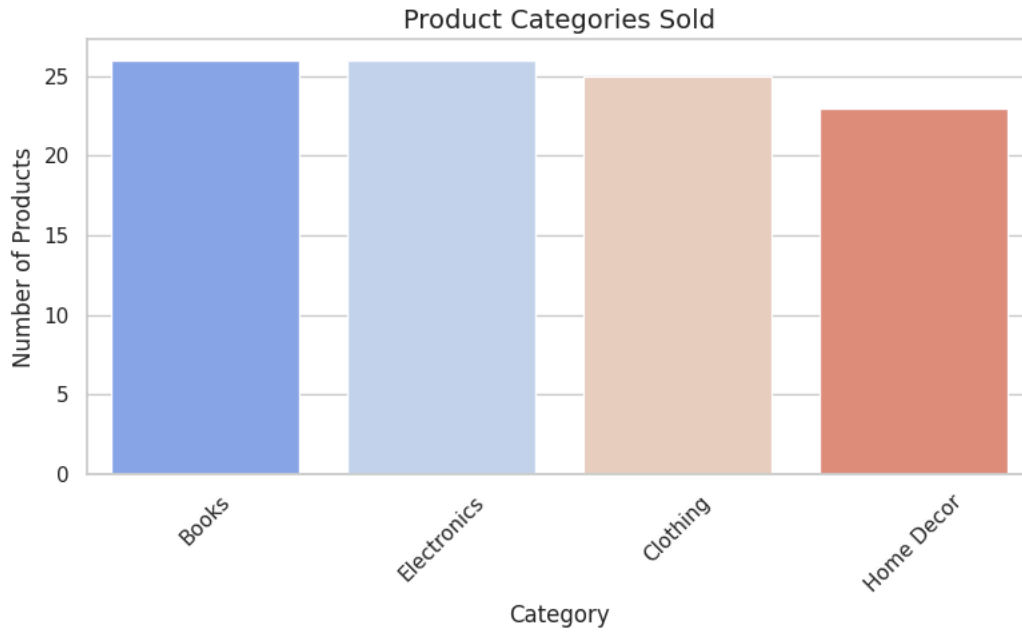


```
# Product Categories Sold
plt.figure(figsize=(8, 5))
category_counts = products['Category'].value_counts()
sns.barplot(x=category_counts.index, y=category_counts.values, palette="coolwarm")
plt.title("Product Categories Sold", fontsize=14)
plt.xlabel("Category")
plt.ylabel("Number of Products")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<ipython-input-29-b9bbb0e9cea5>:4: FutureWarning:
```

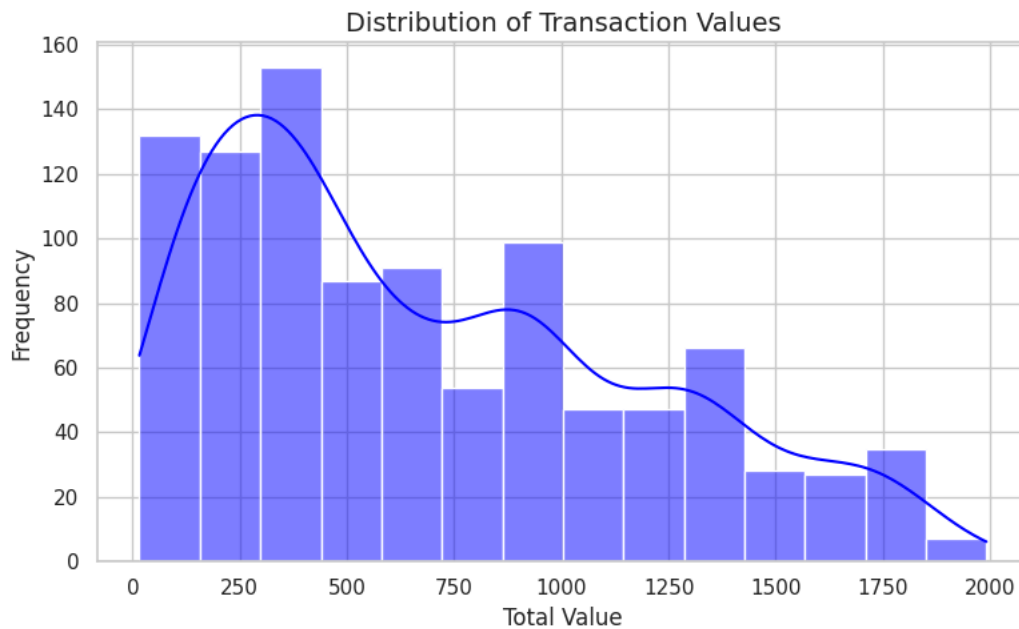
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `l

```
sns.barplot(x=category_counts.index, y=category_counts.values, palette="coolwarm")
```



```
# Distribution of Transaction Values
plt.figure(figsize=(8, 5))
sns.histplot(transactions['TotalValue'], kde=True, color="blue")
plt.title("Distribution of Transaction Values", fontsize=14)
plt.xlabel("Total Value")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```

```
<ipython-input-30-b9bbb0e9cea5>:4: FutureWarning:
```

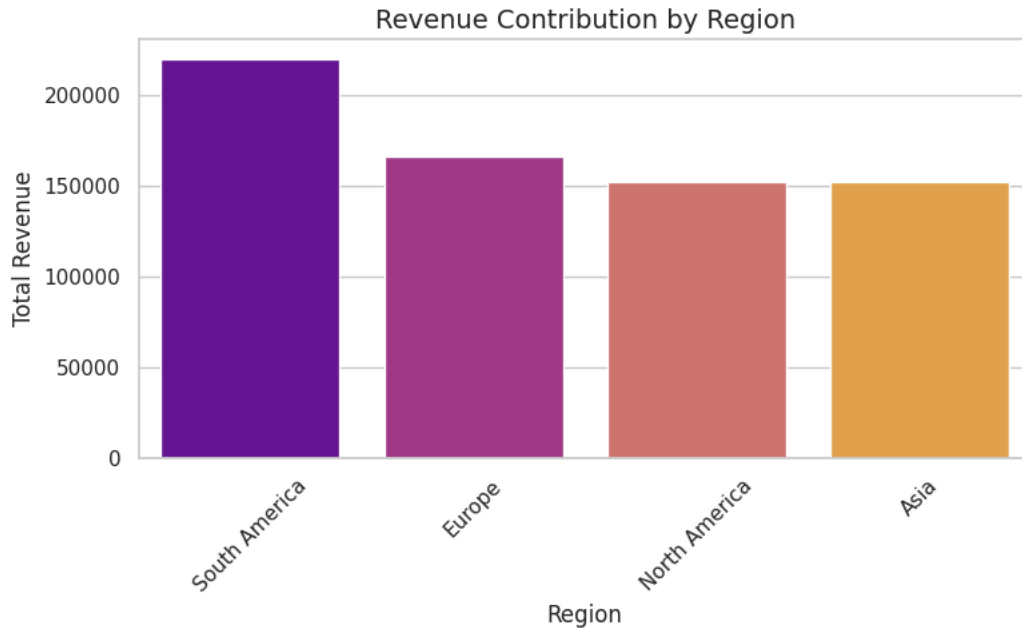


```
# Revenue Contribution by Region
plt.figure(figsize=(8, 5))
region_revenue = merged_data.groupby('Region')['TotalValue'].sum().sort_values(ascending=False)
sns.barplot(x=region_revenue.index, y=region_revenue.values, palette="plasma")
plt.title("Revenue Contribution by Region", fontsize=14)
plt.xlabel("Region")
plt.ylabel("Total Revenue")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
<ipython-input-31-ad681b0333b2>:4: FutureWarning:
```

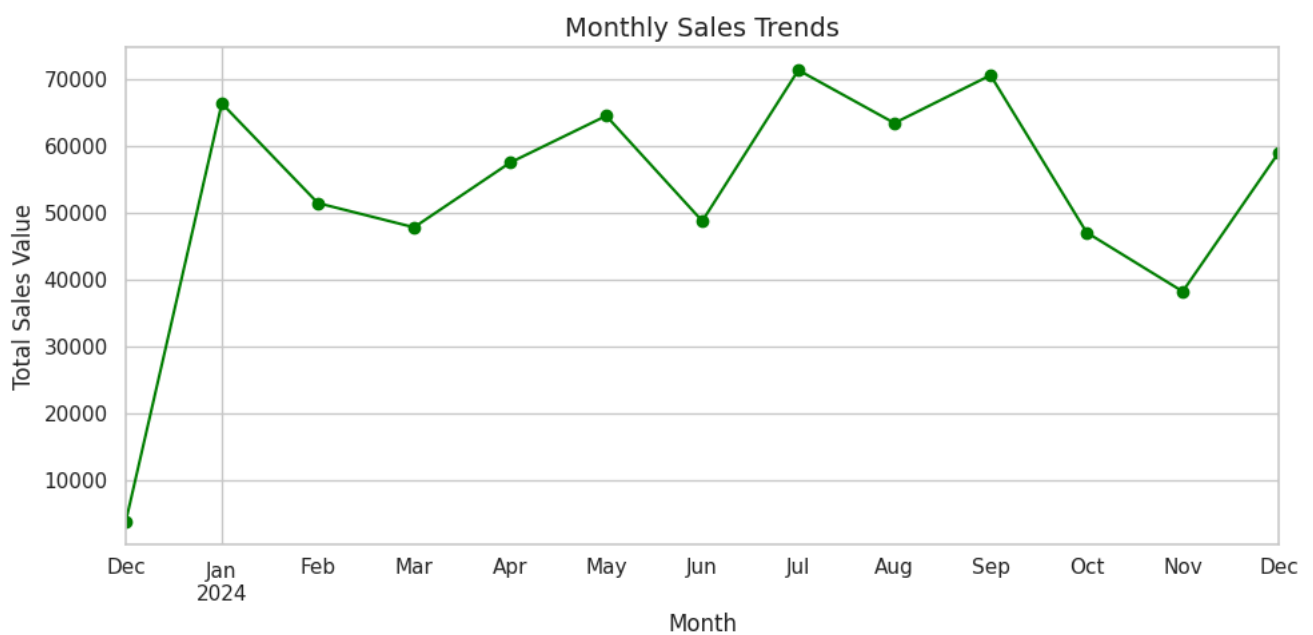
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

```
sns.barplot(x=region_revenue.index, y=region_revenue.values, palette="plasma")
```



```
# Monthly Transaction Trends
merged_data['TransactionDate'] = pd.to_datetime(merged_data['TransactionDate'])
merged_data['Month'] = merged_data['TransactionDate'].dt.to_period('M')
monthly_sales = merged_data.groupby('Month')['TotalValue'].sum()
```

```
plt.figure(figsize=(10, 5))
monthly_sales.plot(kind='line', marker='o', color='green')
plt.title("Monthly Sales Trends", fontsize=14)
plt.xlabel("Month")
plt.ylabel("Total Sales Value")
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
print("\nBusiness Insights:")
print("1. Region with the highest number of customers is:", region_counts.idxmax())
print("2. The product category with the highest sales volume is:", category_counts.idxmax())
print("3. The average transaction value is ${:.2f}".format(transactions['TotalValue'].mean()))
print("4. Region contributing the highest revenue is:", region_revenue.idxmax())
print("5. Monthly sales trend shows highest revenue in:", monthly_sales.idxmax())
```



Business Insights:

1. Region with the highest number of customers is: South America
2. The product category with the highest sales volume is: Books
3. The average transaction value is \$690.00
4. Region contributing the highest revenue is: South America
5. Monthly sales trend shows highest revenue in: 2024-07