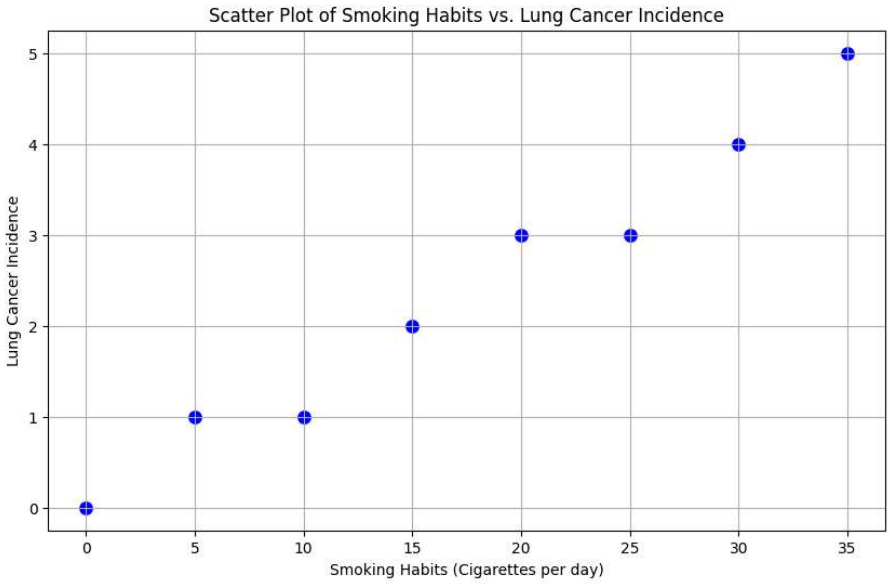
```
import pandas as pd
import pandas as pd

# Sample data to create the DataFrame
data = {
    'OrderID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'CustomerID': [101, 102, 103, 104, 101, 102, 105, 106, 107, 108],
    'ProductID': [1001, 1002, 1003, 1001, 1002, 1003, 1001, 1002, 1003, 1001],
    'Quantity': [2, 1, 5, 3, 2, 1, 4, 1, 2, 5],
    'TotalPrice': [20.0, 15.0, 50.0, 30.0, 20.0, 15.0, 40.0, 15.0, 30.0, 50.0]
}

# Create DataFrame
df = pd.DataFrame(data)
print(df)


# Step 2: Basic Exploration
print("Basic Info:")
print(df.info())
print("\nBasic Statistics:")
print(df.describe())
print("\nFirst 5 Rows:")
print(df.head())

# Step 3: Data Cleaning
# Check for missing values
print("\nMissing Values:")
print(df.isnull().sum())

# Drop duplicates
df.drop_duplicates(inplace=True)

# Step 4: Data Analysis
# Total Revenue
total_revenue = df['TotalPrice'].sum()
print("\nTotal Revenue: $", total_revenue)

# Top 5 Customers by Total Spend
top_customers = df.groupby('CustomerID')['TotalPrice'].sum().sort_values(ascending=False).head(5)
print("\nTop 5 Customers by Total Spend:")
print(top_customers)

# Top 5 Products by Quantity Sold
top_products = df.groupby('ProductID')['Quantity'].sum().sort_values(ascending=False).head(5)
print("\nTop 5 Products by Quantity Sold:")
print(top_products)

# Number of Orders per Customer
orders_per_customer = df['CustomerID'].value_counts()
print("\nNumber of Orders per Customer:")
print(orders_per_customer.head(5))

# Average Order Value
average_order_value = df['TotalPrice'].mean()
print("\nAverage Order Value: $", average_order_value)
```

```
   OrderID  CustomerID  ProductID  Quantity  TotalPrice
0        1         101       1001         2        20.0
1        2         102       1002         1        15.0
2        3         103       1003         5        50.0
3        4         104       1001         3        30.0
4        5         101       1002         2        20.0
5        6         102       1003         1        15.0
6        7         105       1001         4        40.0
7        8         106       1002         1        15.0
8        9         107       1003         2        30.0
9       10         108       1001         5        50.0
Basic Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   OrderID     10 non-null     int64
 1   CustomerID  10 non-null     int64
 2   ProductID   10 non-null     int64
```

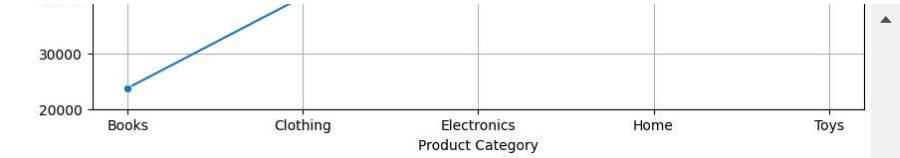
```
 3   Quantity    10 non-null     int64
 4   TotalPrice  10 non-null     float64
dtypes: float64(1), int64(4)
memory usage: 528.0 bytes
None

Basic Statistics:
        OrderID  CustomerID    ProductID   Quantity  TotalPrice
count  10.00000   10.000000    10.000000  10.000000    10.00000
mean    5.50000  103.900000  1001.900000   2.600000    28.50000
std     3.02765    2.514403     0.875595   1.577621    13.95429
min     1.00000  101.000000  1001.000000   1.000000    15.00000
25%     3.25000  102.000000  1001.000000   1.250000    16.25000
50%     5.50000  103.500000  1002.000000   2.000000    25.00000
75%     7.75000  105.750000  1002.750000   3.750000    37.50000
max    10.00000  108.000000  1003.000000   5.000000    50.00000

First 5 Rows:
   OrderID  CustomerID  ProductID  Quantity  TotalPrice
0        1         101       1001         2        20.0
1        2         102       1002         1        15.0
2        3         103       1003         5        50.0
3        4         104       1001         3        30.0
4        5         101       1002         2        20.0

Missing Values:
OrderID       0
CustomerID    0
ProductID     0
Quantity      0
TotalPrice    0
dtype: int64

Total Revenue: $ 285.0

Top 5 Customers by Total Spend:
CustomerID
```

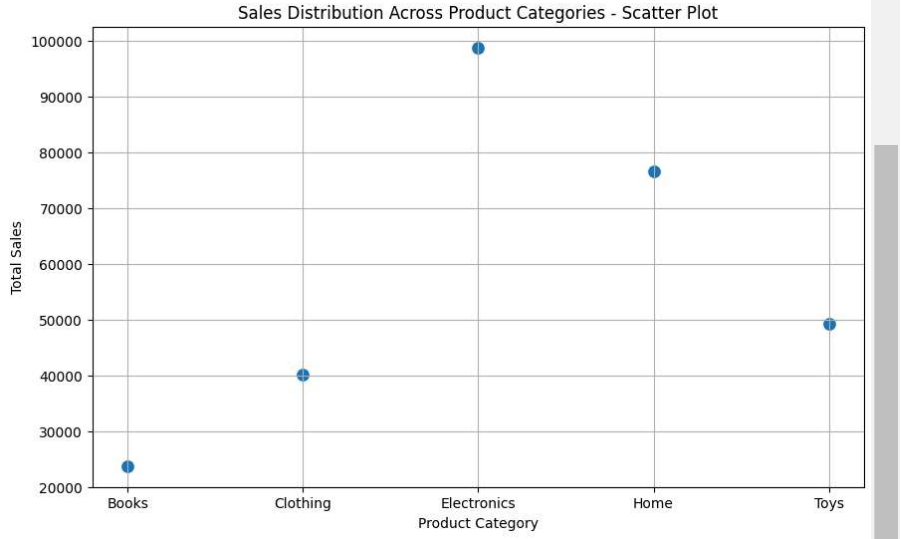
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = {
    'IndividualID': range(1, 21),
    'SmokingHabits': [5, 20, 15, 0, 10, 30, 25, 0, 0, 35, 5, 15, 10, 20, 25, 5, 10, 0, 15, 20],
    'LungCancerIncidence': [1, 3, 2, 0, 1, 4, 3, 0, 0, 5, 1, 2, 1, 3, 3, 1, 1, 0, 2, 3]
}
df = pd.DataFrame(data)
correlation = df['SmokingHabits'].corr(df['LungCancerIncidence'])
print(f"Correlation Coefficient: {correlation:.3f}")
plt.figure(figsize=(10, 6))
sns.scatterplot(x='SmokingHabits', y='LungCancerIncidence', data=df, s=100, color='blue', edgecolor='w')
plt.title('Scatter Plot of Smoking Habits vs. Lung Cancer Incidence')
plt.xlabel('Smoking Habits (Cigarettes per day)')
plt.ylabel('Lung Cancer Incidence')
plt.grid(True)
plt.show()
```

```
Correlation Coefficient: 0.983
```

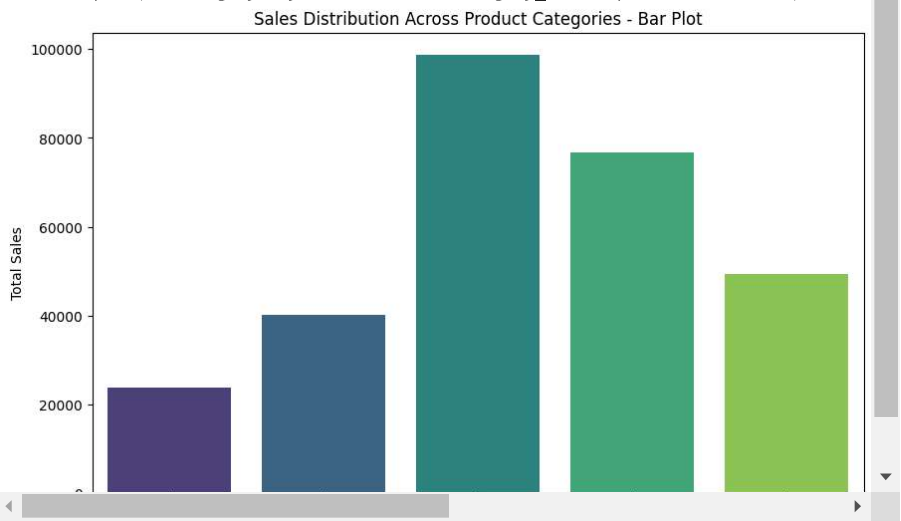



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Step 1: Load the Dataset
# Sample data for the sales data
data = {
    'Date': pd.date_range(start='2023-01-01', periods=365, freq='D'),
    'Category': ['Electronics', 'Clothing', 'Home', 'Books', 'Toys', 'Electronics', 'Clothing', 'Home', 'Books', 'Toys'] * 36 + ['Electronic
    'Sales': [1500, 500, 1000, 300, 700, 1200, 600, 1100, 350, 650] * 36 + [1600, 550, 1050, 330, 720]
}

# Create DataFrame
df = pd.DataFrame(data)

# Step 2: Aggregate Sales by Category
category_sales = df.groupby('Category')['Sales'].sum().reset_index()

# Step 3: Create Visualizations

# Line Plot
plt.figure(figsize=(10, 6))
sns.lineplot(x='Category', y='Sales', data=category_sales, marker='o')
plt.title('Sales Distribution Across Product Categories - Line Plot')
plt.xlabel('Product Category')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()

# Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Category', y='Sales', data=category_sales, s=100)
plt.title('Sales Distribution Across Product Categories - Scatter Plot')
plt.xlabel('Product Category')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()

# Bar Plot
plt.figure(figsize=(10, 6))
sns.barplot(x='Category', y='Sales', data=category_sales, palette='viridis')
plt.title('Sales Distribution Across Product Categories - Bar Plot')
plt.xlabel('Product Category')
plt.ylabel('Total Sales')
plt.show()
```

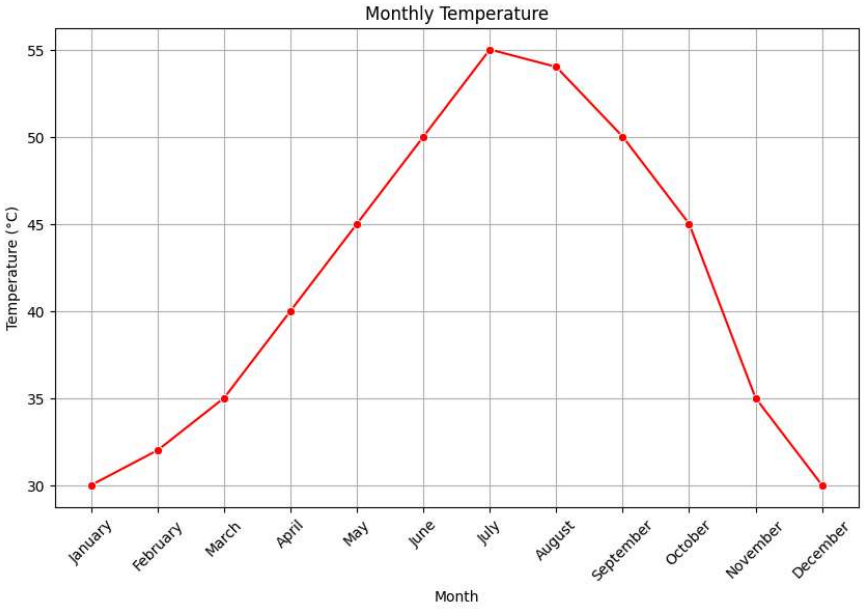


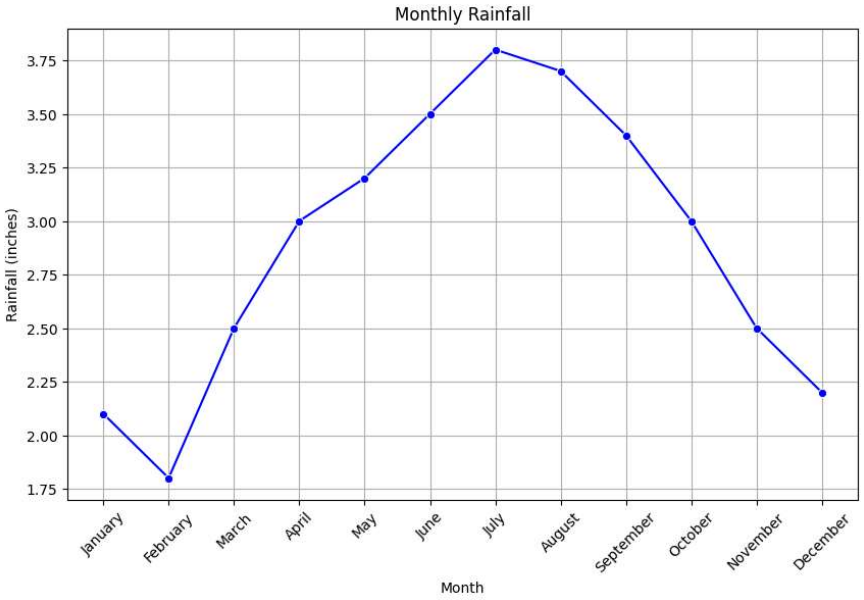


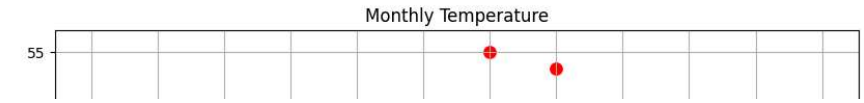

```
<ipython-input-3-8905a68766d9>:41: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(x='Category', y='Sales', data=category_sales, palette='viridis')
```



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

# Step 1: Load the Dataset
# Sample data for monthly temperature and rainfall
data = {
    'Month': ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'],
    'Temperature': [30, 32, 35, 40, 45, 50, 55, 54, 50, 45, 35, 30],
    'Rainfall': [2.1, 1.8, 2.5, 3.0, 3.2, 3.5, 3.8, 3.7, 3.4, 3.0, 2.5, 2.2]
}

# Create DataFrame
df = pd.DataFrame(data)

# Step 2: Create Visualizations

# Line Plot for Temperature
plt.figure(figsize=(10, 6))
sns.lineplot(x='Month', y='Temperature', data=df, marker='o', color='red')
plt.title('Monthly Temperature')
plt.xlabel('Month')
plt.ylabel('Temperature (°C)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

# Line Plot for Rainfall
plt.figure(figsize=(10, 6))
sns.lineplot(x='Month', y='Rainfall', data=df, marker='o', color='blue')
plt.title('Monthly Rainfall')
plt.xlabel('Month')
plt.ylabel('Rainfall (inches)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

# Scatter Plot for Temperature
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Month', y='Temperature', data=df, s=100, color='red')
plt.title('Monthly Temperature')
plt.xlabel('Month')
plt.ylabel('Temperature (°C)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

# Scatter Plot for Rainfall
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Month', y='Rainfall', data=df, s=100, color='blue')
plt.title('Monthly Rainfall')
plt.xlabel('Month')
plt.ylabel('Rainfall (inches)')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```


Monthly Temperature
Temperature (°C)
55
50
45
40
35
30
January
February
March
April
May
June
July
August
September
October
November
December
Month


Monthly Rainfall
Rainfall (inches)
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
January
February
March
April
May
June
July
August
September
October
November
December
Month


Monthly Temperature
55