```
from google.colab import files
files1 = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving online_retail_II.xlsx to online_retail_II.xlsx
```

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

```
df = pd.read_excel("online_retail_II.xlsx")
```

```
df.head()
```

```
{"type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Description  522533 non-null  object
 3   Quantity     525461 non-null  int64
 4   InvoiceDate  525461 non-null  datetime64[ns]
 5   Price        525461 non-null  float64
 6   Customer ID  417534 non-null  float64
 7   Country      525461 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 32.1+ MB
```

```
df.isnull().values.any()
```

```
np.True_
```

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

```
df.tail()
```

```
{"repr_error":"0","type":"dataframe"}
```

```
df = df[~df['Invoice'].astype(str).str.startswith('C')]
```

```
df['TotalAmount'] = df['Quantity'] * df['Price']
```

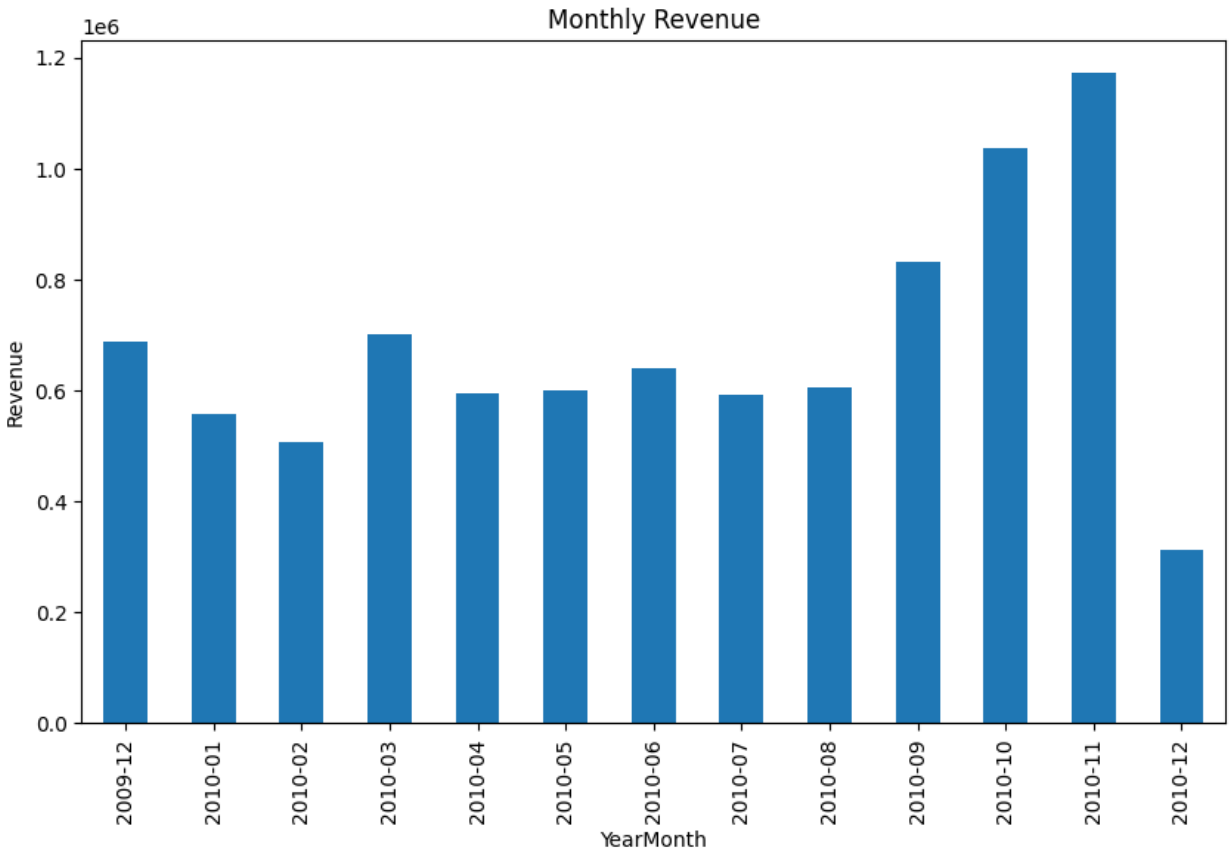
```
df.head()
```

```
{"type":"dataframe","variable_name":"df"}
```

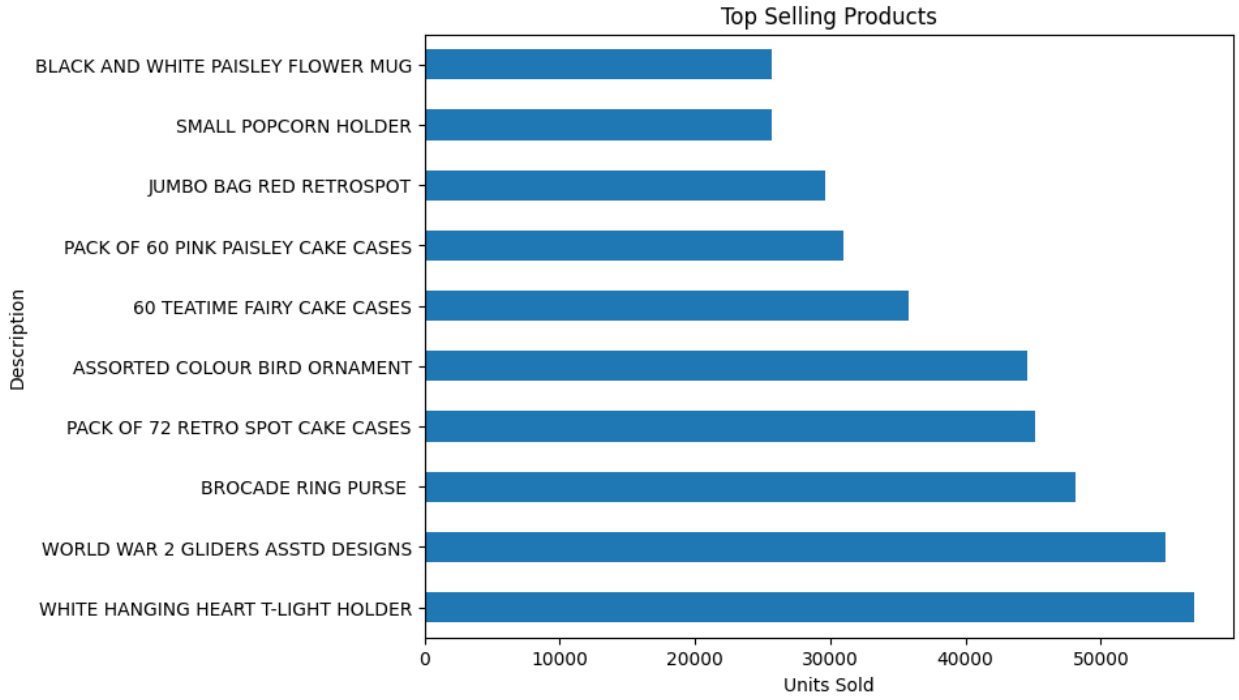
```
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])
df['YearMonth'] = df['InvoiceDate'].dt.to_period('M')
df['Date'] = df['InvoiceDate'].dt.date
df['Hour'] = df['InvoiceDate'].dt.hour

df.head()
```

```
{"type":"dataframe","variable_name":"df"}
```

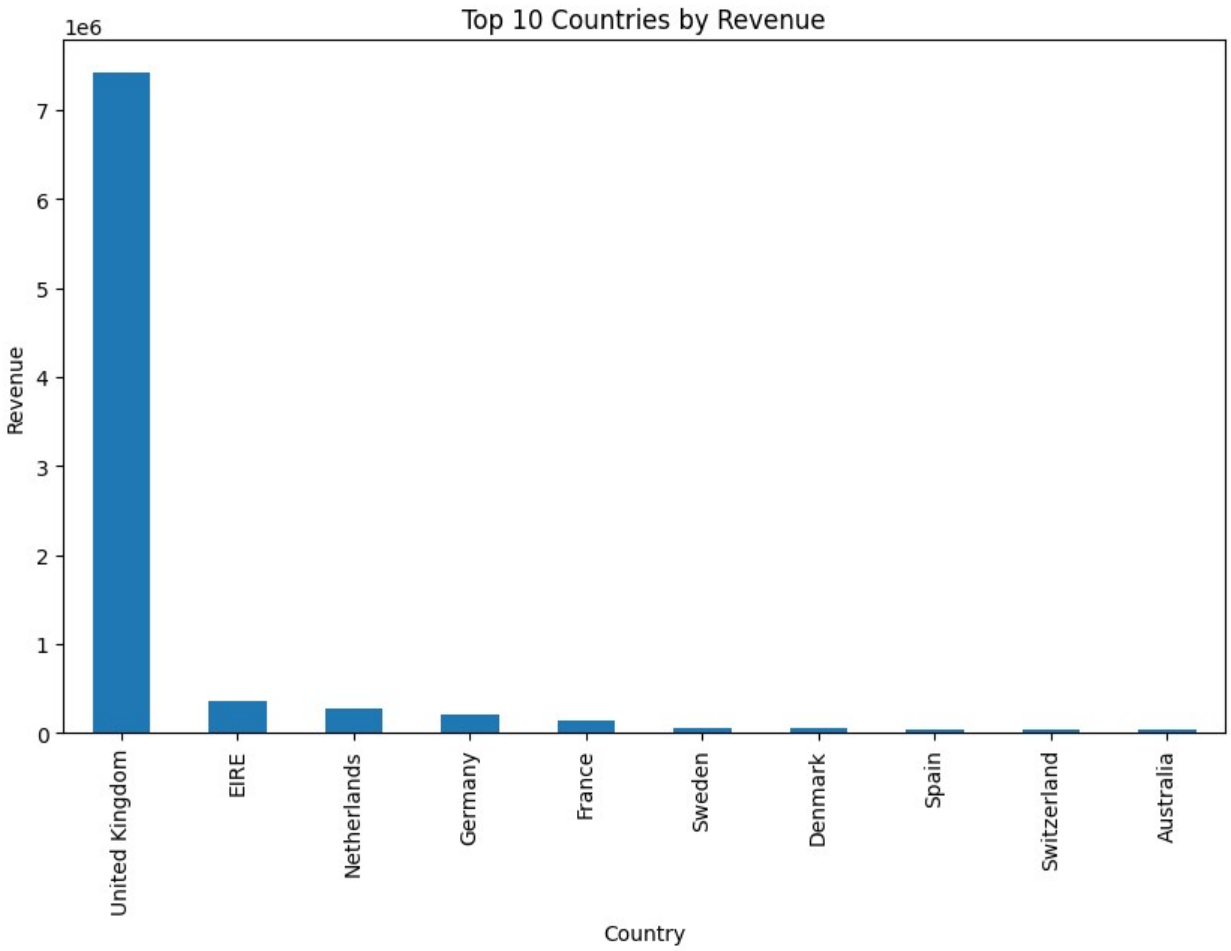
```
monthly_revenue = df.groupby('YearMonth')['TotalAmount'].sum()
top_products = df.groupby('Description')
['Quantity'].sum().sort_values(ascending=False).head(10)
revenue_by_country = df.groupby('Country')
['TotalAmount'].sum().sort_values(ascending=False)
daily_orders = df.groupby('Date')['Invoice'].nunique()

total_customers = df['Customer ID'].nunique()

order_value = df.groupby('Invoice')['TotalAmount'].sum()
avg_order_value = order_value.mean()

orders_per_customer = df.groupby('Customer ID')['Invoice'].nunique()

revenue_per_customer = df.groupby('Customer ID')['TotalAmount'].sum()

hourly_sales = df.groupby('Hour')['TotalAmount'].sum()
df['Weekday'] = df['InvoiceDate'].dt.day_name()
weekday_sales = df.groupby('Weekday')['TotalAmount'].sum()

returns = df[df['Quantity'] < 0]
num_returns = returns['Invoice'].nunique()
return_rate = num_returns / df['Invoice'].nunique()

monthly_revenue.plot(kind='bar', figsize=(10, 6), title='Monthly
Revenue')
plt.ylabel('Revenue')
plt.show()
```



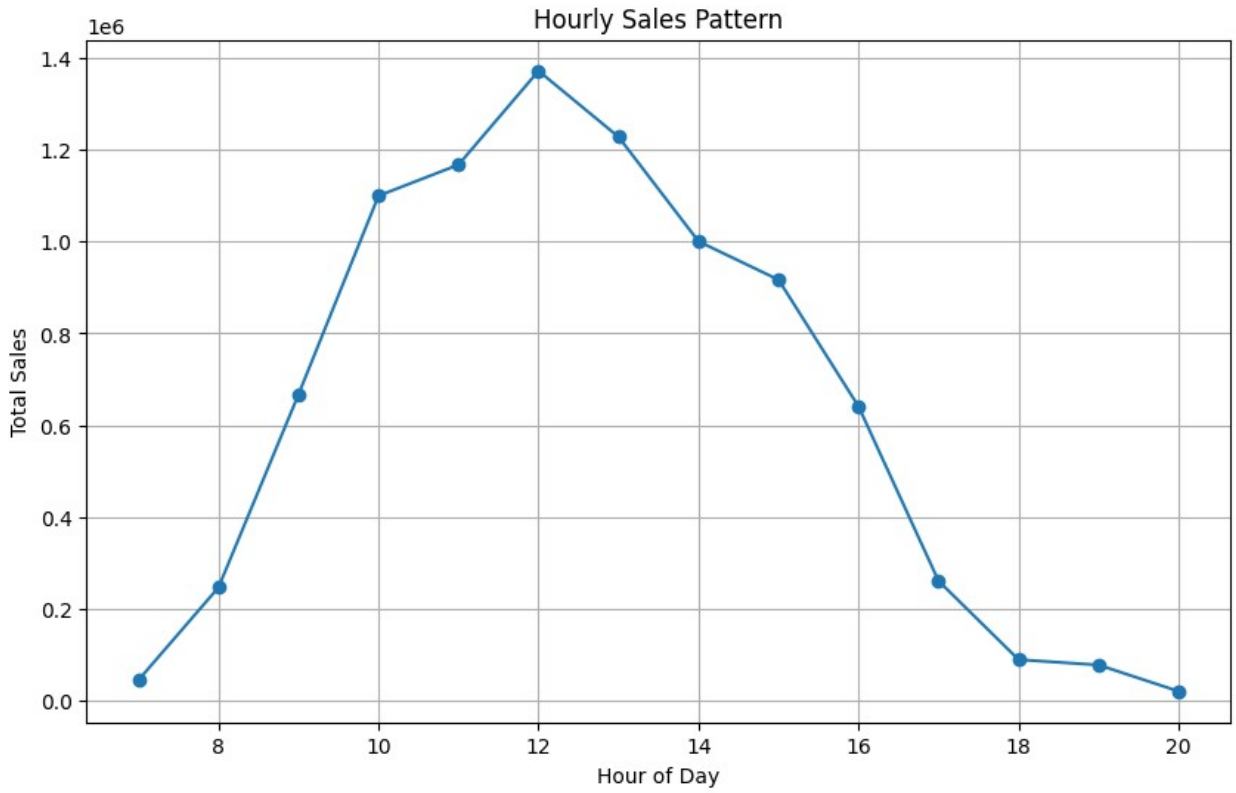

```
top_products.plot(kind='barh', figsize=(8, 6), title='Top Selling
Products')
plt.xlabel('Units Sold')
plt.show()
```




```
revenue_by_country.head(10).plot(kind='bar', figsize=(10, 6),
title='Top 10 Countries by Revenue')
plt.ylabel('Revenue')
plt.show()
```



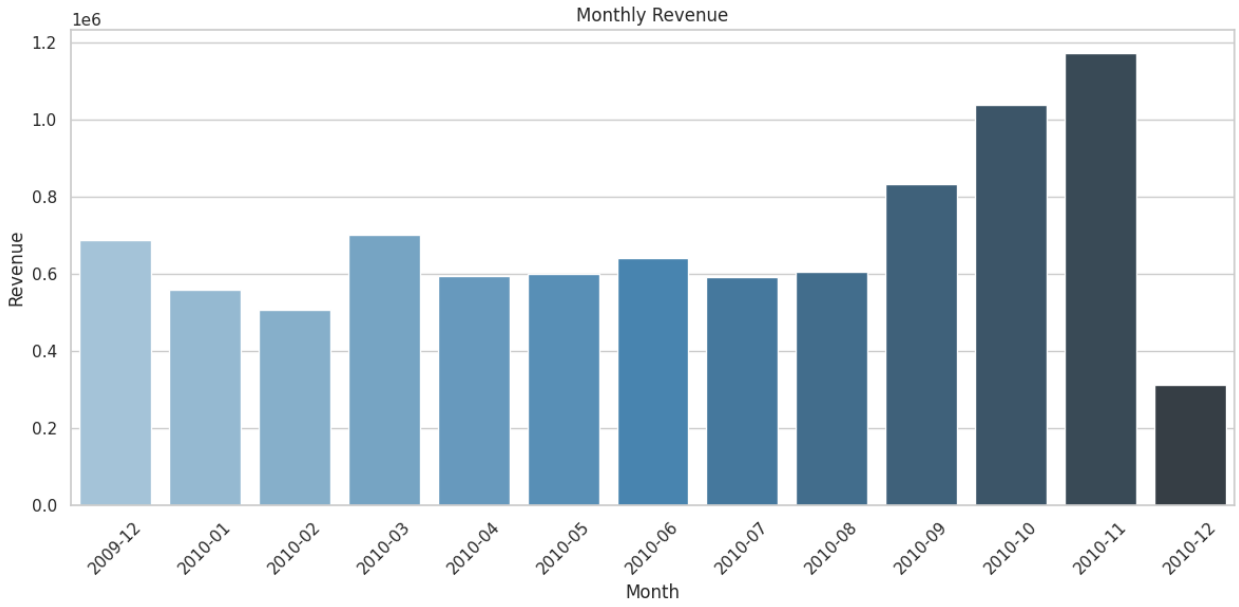

```
hourly_sales.plot(kind='line', marker='o', figsize=(10, 6),
title='Hourly Sales Pattern')
plt.xlabel('Hour of Day')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()
```



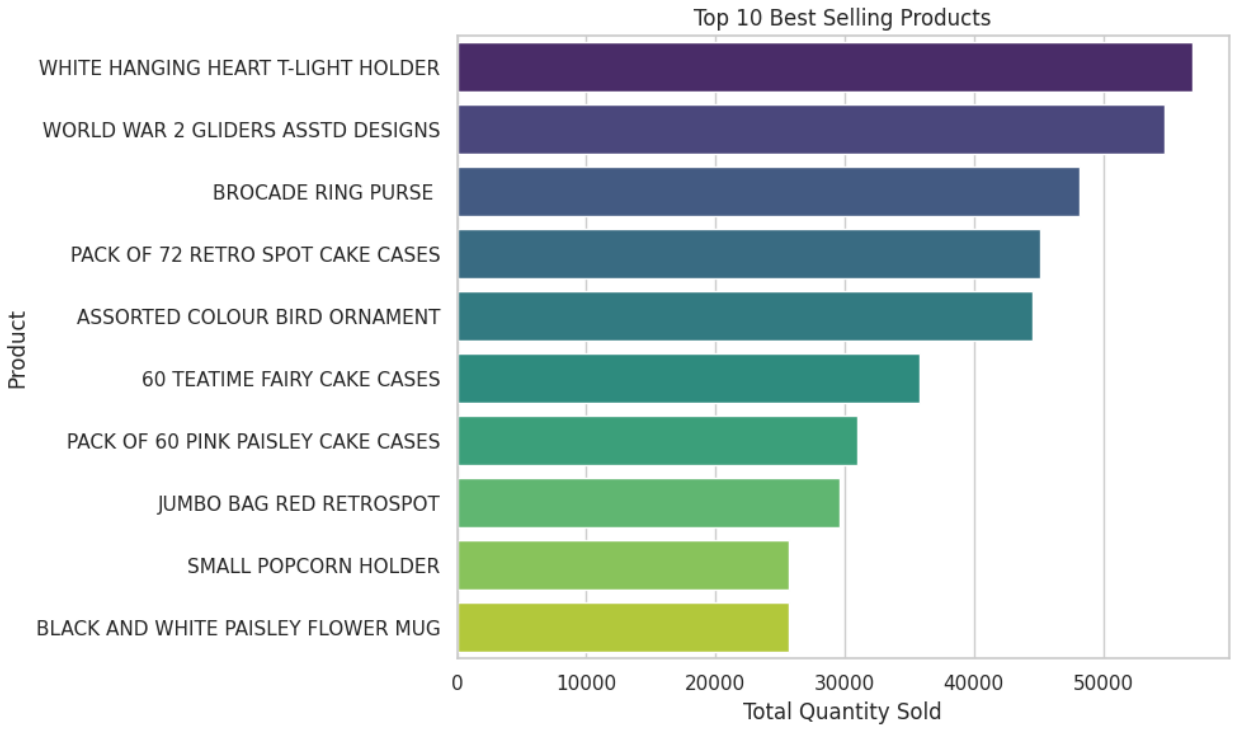

```
sns.set(style='whitegrid')
monthly_revenue = df.groupby('YearMonth')
['TotalAmount'].sum().reset_index()
top_products = df.groupby('Description')
['Quantity'].sum().sort_values(ascending=False).head(10).reset_index()
country_revenue = df.groupby('Country')
['TotalAmount'].sum().sort_values(ascending=False).head(10).reset_inde
x()
ourly_orders = df.groupby('Hour')['Invoice'].nunique().reset_index()

plt.figure(figsize=(12, 6))
sns.barplot(data=monthly_revenue, x='YearMonth', y='TotalAmount',
palette='Blues_d')
plt.xticks(rotation=45)
plt.title('Monthly Revenue')
plt.ylabel('Revenue')
plt.xlabel('Month')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-1617719207.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.
```

```
  sns.barplot(data=monthly_revenue, x='YearMonth', y='TotalAmount',
palette='Blues_d')
```

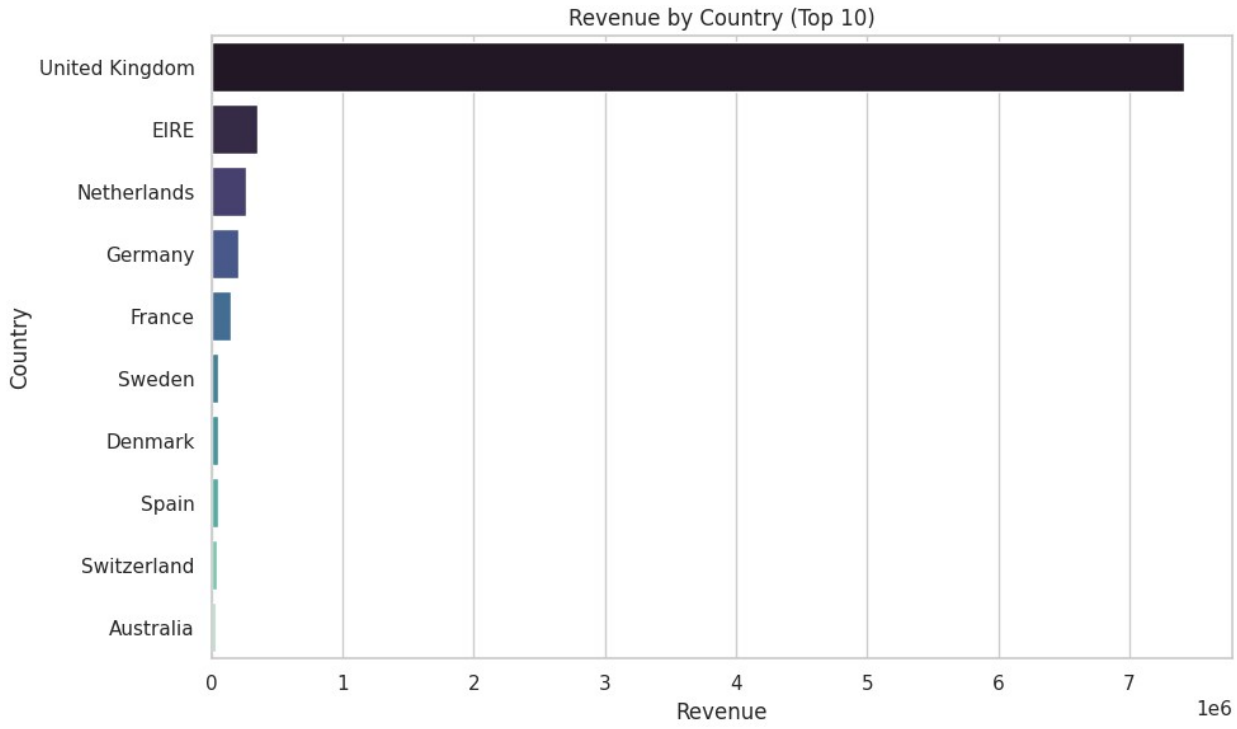



```
plt.figure(figsize=(10, 6))
sns.barplot(data=top_products, x='Quantity', y='Description',
palette='viridis')
plt.title('Top 10 Best Selling Products')
plt.xlabel('Total Quantity Sold')
plt.ylabel('Product')
plt.tight_layout()
plt.show()

/tmp/ipython-input-1383054224.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=top_products, x='Quantity', y='Description',
palette='viridis')
```

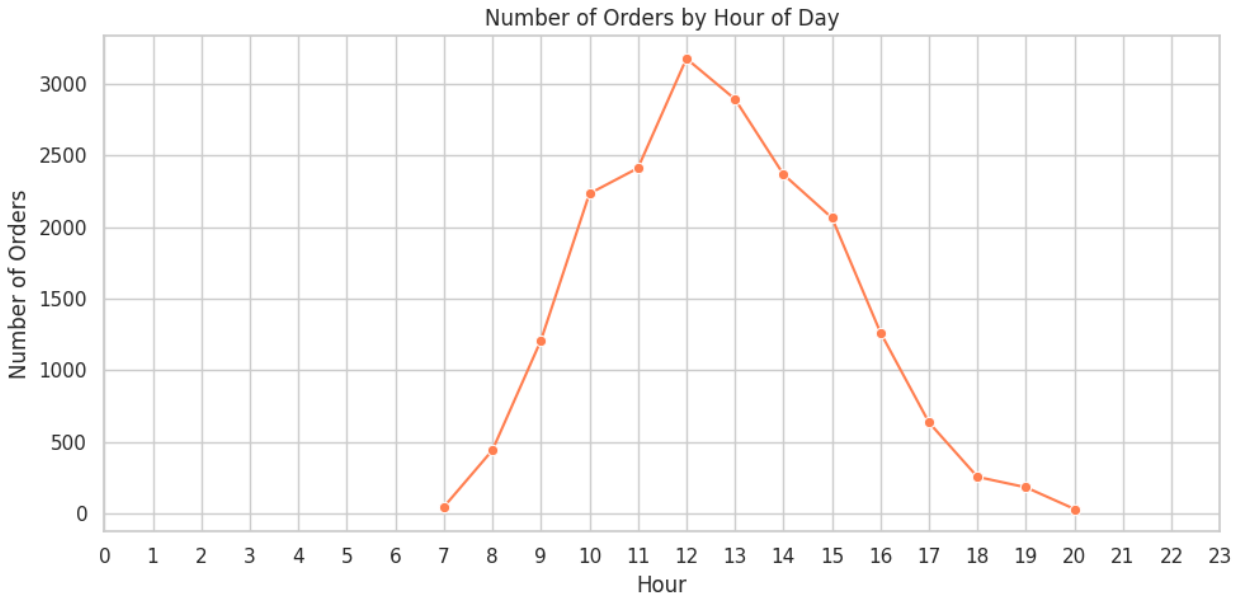



```
plt.figure(figsize=(10, 6))
sns.barplot(data=country_revenue, y='Country', x='TotalAmount',
palette='mako')
plt.title('Revenue by Country (Top 10)')
plt.xlabel('Revenue')
plt.ylabel('Country')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-574974669.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=country_revenue, y='Country', x='TotalAmount',
palette='mako')
```

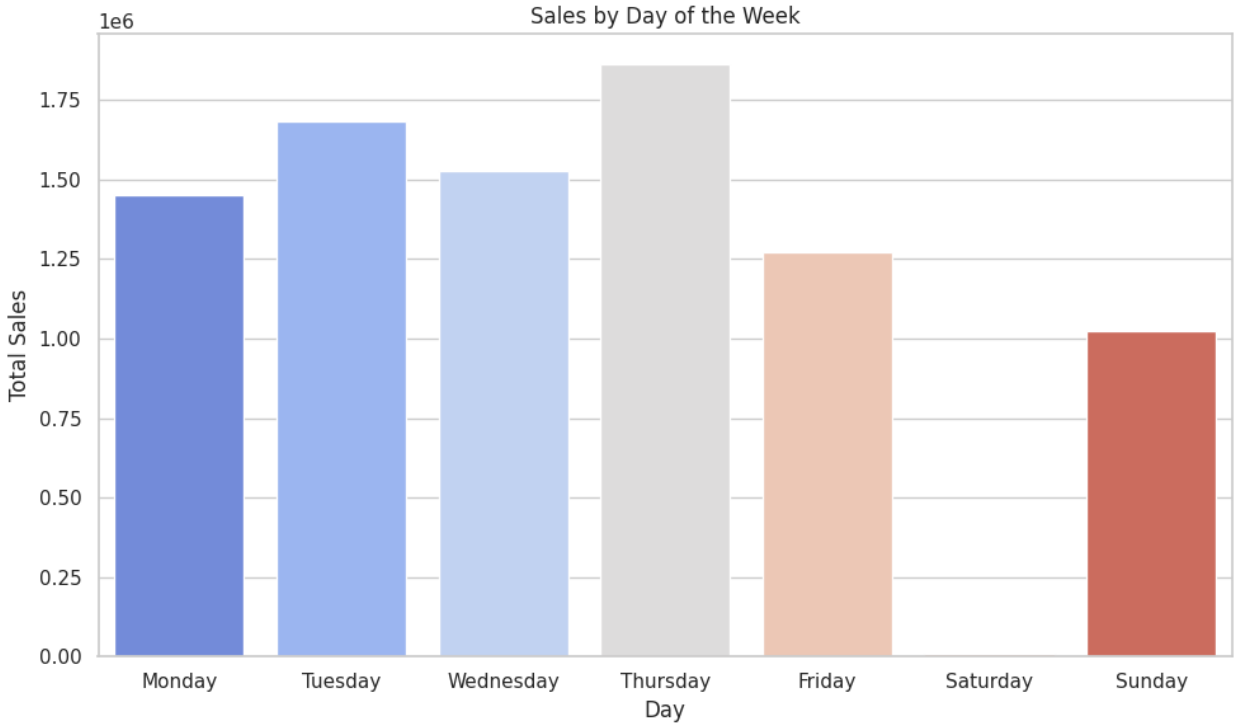



```
plt.figure(figsize=(10, 5))
sns.lineplot(data=ourly_orders, x='Hour', y='Invoice', marker='o',
color='coral')
plt.title('Number of Orders by Hour of Day')
plt.xlabel('Hour')
plt.ylabel('Number of Orders')
plt.xticks(range(0, 24))
plt.tight_layout()
plt.show()
```




```
df['DayOfWeek'] = df['InvoiceDate'].dt.day_name()
day_sales = df.groupby('DayOfWeek')['TotalAmount'].sum().reset_index()

days_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday', 'Sunday']
day_sales['DayOfWeek'] = pd.Categorical(day_sales['DayOfWeek'],
categories=days_order, ordered=True)
day_sales = day_sales.sort_values('DayOfWeek')

plt.figure(figsize=(10, 6))
sns.barplot(data=day_sales, x='DayOfWeek', y='TotalAmount',
palette='coolwarm')
plt.title('Sales by Day of the Week')
plt.xlabel('Day')
plt.ylabel('Total Sales')
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-124804112.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data=day_sales, x='DayOfWeek', y='TotalAmount',
palette='coolwarm')
```


Sales by Day of the Week
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Total Sales
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day