In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset (make sure it's in the same folder)
df = pd.read_csv("data.csv", encoding='ISO-8859-1')
df.head()
```

Out[1]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

In [2]:
```
# Convert InvoiceDate to datetime
df['InvoiceDate'] = pd.to_datetime(df['InvoiceDate'])

# Drop rows with missing CustomerID
df = df.dropna(subset=['CustomerID'])

# Remove negative quantities (canceled/returned orders)
df = df[df['Quantity'] > 0]

# Create TotalPrice column
df['TotalPrice'] = df['Quantity'] * df['UnitPrice']

# Check cleaned data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 397924 entries, 0 to 541908
Data columns (total 9 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   InvoiceNo    397924 non-null  object
 1   StockCode    397924 non-null  object
 2   Description  397924 non-null  object
 3   Quantity     397924 non-null  int64
 4   InvoiceDate  397924 non-null  datetime64[ns]
 5   UnitPrice    397924 non-null  float64
 6   CustomerID   397924 non-null  float64
 7   Country      397924 non-null  object
 8   TotalPrice   397924 non-null  float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(4)
memory usage: 30.4+ MB
```

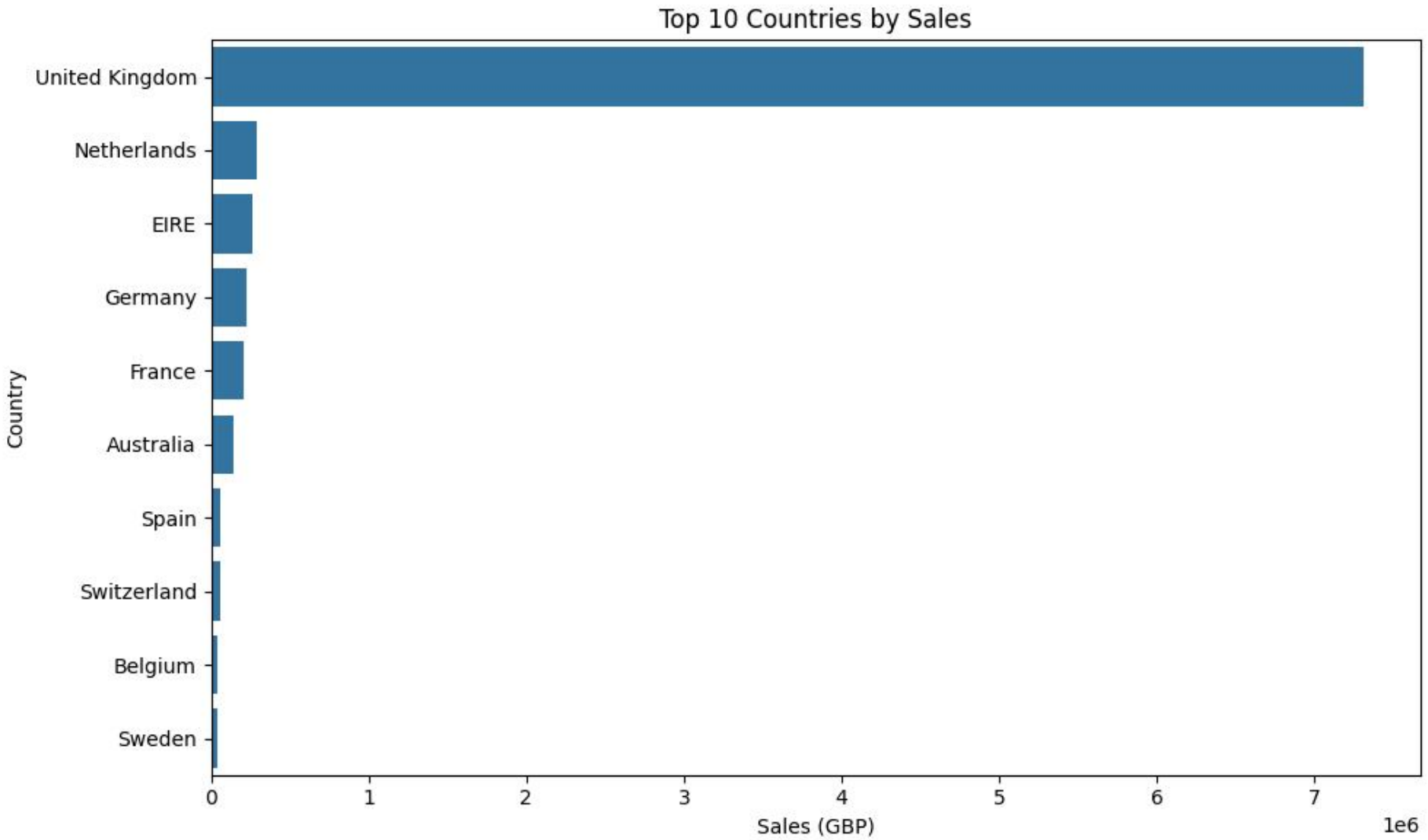
In [3]:

```python
# Total Sales
total_sales = df['TotalPrice'].sum()

# Total Unique Customers
unique_customers = df['CustomerID'].nunique()

# Returning Customers
orders_per_customer = df.groupby('CustomerID')['InvoiceNo'].nunique()
returning_customers = orders_per_customer[orders_per_customer > 1].count()
return_rate = (returning_customers / unique_customers) * 100

# Top 10 Selling Products
top_products = df.groupby('Description')['Quantity'].sum().sort_values(ascending=False).head(10)

# Display
print(f"💰 Total Sales: £{total_sales:.2f}")
print(f"👥 Unique Customers: {unique_customers}")
print(f"🔁 Returning Customer Rate: {return_rate:.2f}%")
print("🏆 Top 10 Selling Products:")
print(top_products)
```

```
💰 Total Sales: £8911407.90
👥 Unique Customers: 4339
🔁 Returning Customer Rate: 65.57%
🏆 Top 10 Selling Products:
Description
PAPER CRAFT , LITTLE BIRDIE           80995
MEDIUM CERAMIC TOP STORAGE JAR        77916
WORLD WAR 2 GLIDERS ASSTD DESIGNS     54415
JUMBO BAG RED RETROSPOT               46181
WHITE HANGING HEART T-LIGHT HOLDER    36725
ASSORTED COLOUR BIRD ORNAMENT         35362
PACK OF 72 RETROSPOT CAKE CASES       33693
POPCORN HOLDER                        30931
RABBIT NIGHT LIGHT                    27202
MINI PAINT SET VINTAGE                26076
Name: Quantity, dtype: int64
```

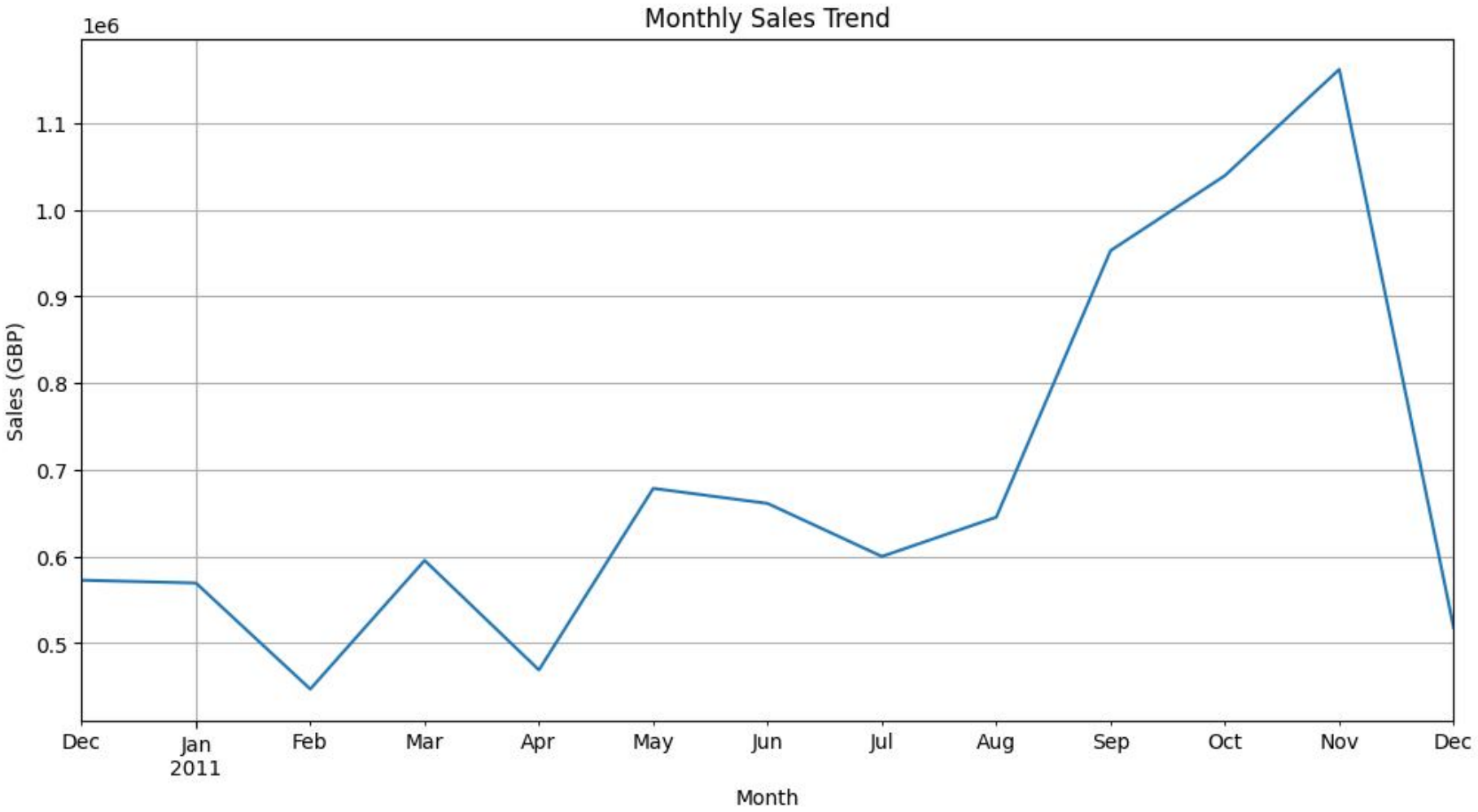
In [4]:

```python
top_countries = df.groupby('Country')['TotalPrice'].sum().sort_values(ascending=False).head(10)

plt.figure(figsize=(10,6))
sns.barplot(x=top_countries.values, y=top_countries.index)
plt.title("Top 10 Countries by Sales")
plt.xlabel("Sales (GBP)")
plt.ylabel("Country")
plt.tight_layout()
plt.show()
```



In [5]:
```
df['Month'] = df['InvoiceDate'].dt.to_period('M')
monthly_sales = df.groupby('Month')['TotalPrice'].sum()

monthly_sales.plot(kind='line', figsize=(12,6), title="Monthly Sales Trend")
plt.ylabel("Sales (GBP)")
plt.xlabel("Month")
```

```
plt.grid(True)
plt.show()
```



In [6]:
```
## 📌 Business Insights

- 🔼 Total sales over the period: **£XXX**
- 🔁 Returning customer rate: **XX%**
- 🛒 Top product: **“Product Name” with X units sold**
- 🌍 United Kingdom is the biggest market
```

```
- 🗓️ Most sales happened during Q4 (Oct–Dec)
- ⚠️ Returns were removed from the data – consider analyzing cancellations separately.
```

```
  Cell In[6], line 3
    - 🔼 Total sales over the period: **£XXX**
      ^
SyntaxError: invalid character '🔼' (U+1F53C)
```

In [ ]: