# Data Analyst Portfolio Project: Online Retail Analysis

## Executive Summary

This project analyzes real transactional data from a UK-based online retail store (Dec 2010 – Dec 2011). The goal is to uncover product performance trends, customer behavior, and churn risk, while recommending actionable strategies for marketing, sales, and inventory management.

**Key Findings:**

- Core products drive steady revenue year-round, while select products experience seasonal or event-driven spikes.
- UK leads in sales, but international markets (Germany, Netherlands) show distinct product preferences.

**Business Impact:**

Optimizing stock planning, tailoring promotions by product seasonality, and customizing strategies by customer and country can increase profitability and reduce inefficiencies.

## 1. Dataset & Business Objective

**Dataset Overview:**

- Source: [UCI Machine Learning Repository: Online Retail Dataset]
- Transactions: Dec 2010 – Dec 2011
- Size: 54,910 rows
- Key Columns: InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country

**Business Problem:** How can the company forecast demand, identify top-performing products, and optimize marketing and supply chain planning?

**Approach:**

- SQL: Customer segmentation, product popularity, regional sales analysis
- Python: Time series forecasting, clustering, churn analysis

# 2. Data Cleaning & Preparation

## 2.1. Clean dataset in Python for SQL schema creation

**Actions Taken:**

| Column | Action |
|---|---|
| CustomerID | Drop rows where missing ID |
| CustomerID | Convert to Integer |
| InvoiceDate | Convert to datetime format |
| Quantity | Remove rows with Quantity <=0 |
| UnitPrice | Remove rows with Unitprice <=0 |
| Duplicates | Drop exact duplicates |
| Description | Strip whitespace, handle nulls |
| Description | Normalize description column with upper()<br>Remove non-product rows<br>non_products = ['MANUAL', 'AMAZON FEE', 'ADJUST BAD DEBT', 'POSTAGE', 'DOTCOM POSTAGE']<br>df = df[~df['Description'].isin(non_products)] |
| InvoiceNo | Remove canceled invoices |
| InvoiceNo | Ensure InvoiceNo is treated as String |
| Country | Strip whitespace |
| UnitPrice & Quantity | Check for Outliers in UnitPrice / Quantity with describe()<br>Remove UnitPrice < 0.01 |
| Add TotalPrice column | TotalPrice = Quantity × UnitPrice |

**Final Cleaning Review:**

| Action | Code |
|---|---|
| Reset Index After Cleaning | df = df.reset_index(drop=True) |
| Check for Remaining Nulls | print(df.isnull().sum()) |
| Check Data Types One Last Time | |

## 2.2. Set up a relational schema in MySQL

**Actions Taken:**

1. Create new Database and 4 tables in MySQL
2. Import cleaned data(CSV) from Python into SQL using Table Data Import Wizard
3. Created a reusable base for SQL analysis or joining with Python

**SQL Table Design:**

| Table | Row Count | Column | Notes |
|---|---|---|---|
| customers | 4335 | CustomerID, Country | Dropped rows with missing Customer ID |
| products | 3661 | StockCode, Description | Cleaned product descriptions |
| invoices | 18416 | InvoiceNo, InvoiceDate, CustomerID | Excluded canceled invoices |
| orders | 391294 | InvoiceNo, StockCode, Quantity, UnitPrice, TotalPrice | Removed rows with negative quantity and unitprice |

## 2.3. Data Analysis & Visualization: SQL Joins + Python Charts

*Note: All revenue values in this report are in **GBP (£).** For simplicity, charts are presented without currency symbols, but all monetary amounts reflect GBP.*

### 2.3.1. Total Revenue by Product by Month

**SQL Query Used:**

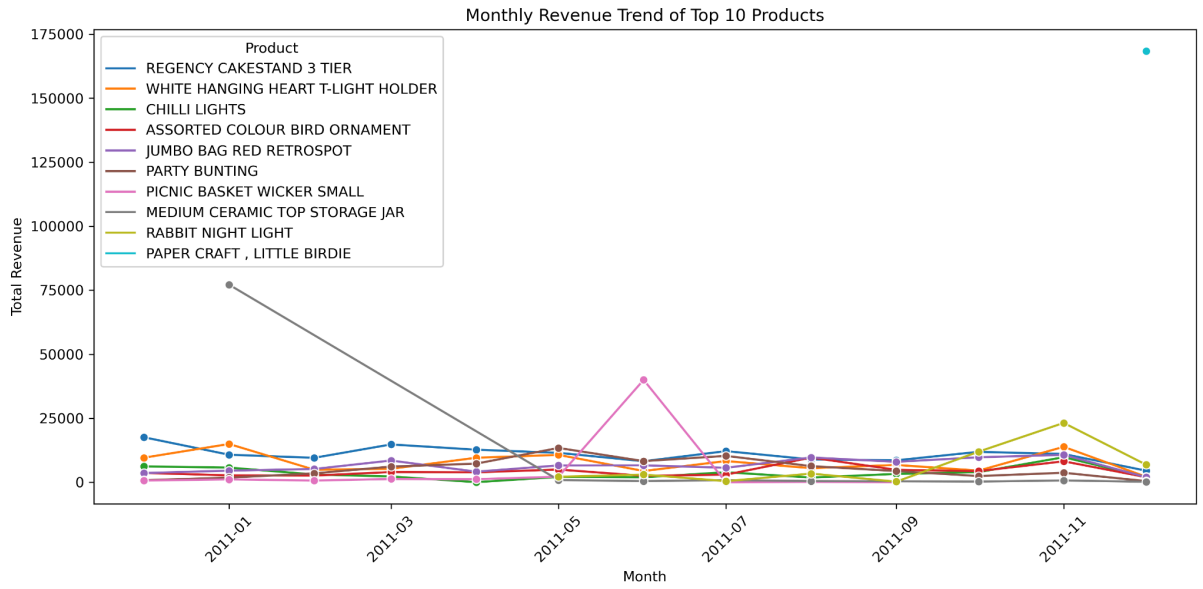
```
SELECT
   DATE_FORMAT(i.InvoiceDate, '%Y-%m') AS Month,
   p.Description AS Product,
   SUM(o.Quantity) AS TotalQuantity,
   SUM(o.TotalPrice) AS TotalRevenue
FROM orders o
JOIN products p ON o.StockCode = p.StockCode
JOIN invoices i ON o.InvoiceNo = i.InvoiceNo
GROUP BY Month, Product
ORDER BY Month, TotalRevenue DESC;
```

**SQL Explanation:**
Goal: Understand how product revenue trends vary over time, monthly.

**Logic:** Grouped by product and month using DATE_FORMAT, aggregated with SUM(Quantity) and SUM(TotalPrice).

**Purpose:** Identify top-performing products by month to support demand planning and promotional timing.ch product to identify trends over time.

**Python Chart:**




**Python Code Used:**

```
top_products = df.groupby('Product')['TotalRevenue'].sum().nlargest(10).index
filtered_df = df[df['Product'].isin(top_products)]

plt.figure(figsize=(12, 6))
sns.lineplot(data=filtered_df, x='Month', y='TotalRevenue', hue='Product', marker='o')

plt.title('Monthly Revenue Trend of Top 10 Products')
plt.xlabel('Month')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

**Python Explanation:**
Data Prep: Filtered for top 10 products by total revenue to reduce noise.
Chart Type: Line plot to visualize revenue trends over time per product.
Design Choice: Used sns.lineplot with distinct colors and markers for clarity; exported for documentation.

## 2.3.2. Top Products by Quarter (Quantity + Revenue)

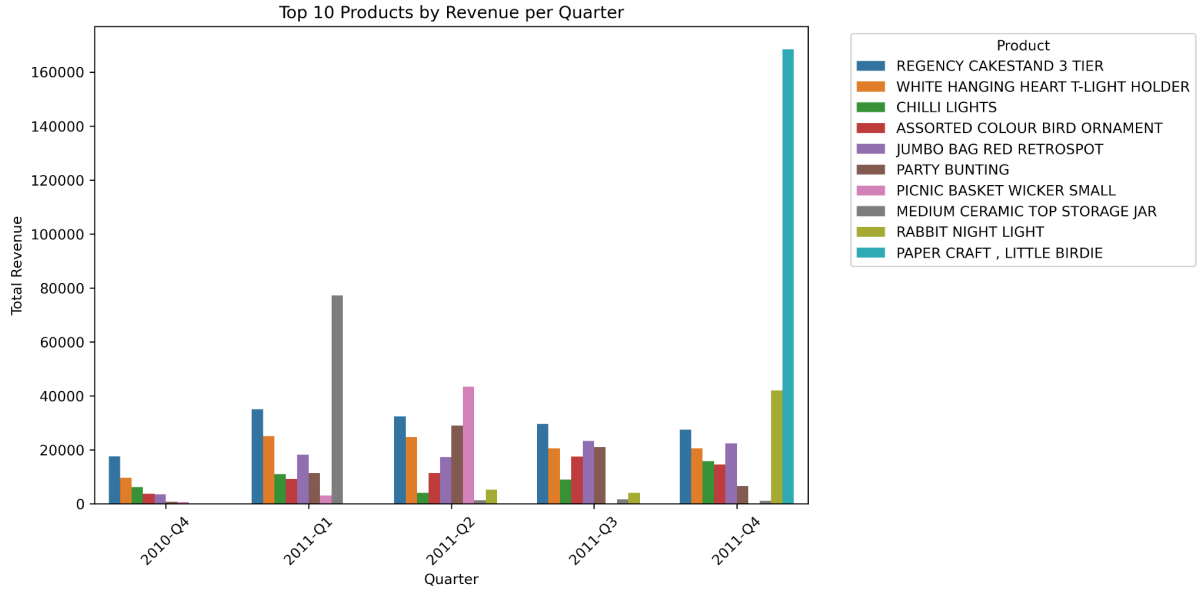
**SQL Query Used:**

```
SELECT
    CONCAT(YEAR(i.InvoiceDate), '-Q', QUARTER(i.InvoiceDate)) AS Quarter,
    p.Description AS Product,
    SUM(o.Quantity) AS TotalQuantity,
    SUM(o.TotalPrice) AS TotalRevenue
FROM orders o
JOIN products p ON o.StockCode = p.StockCode
JOIN invoices i ON o.InvoiceNo = i.InvoiceNo
GROUP BY Quarter, Product
ORDER BY Quarter, TotalRevenue DESC;
```

**SQL Explanation:**

Goal: Find top-selling products per quarter to identify seasonal patterns.
Logic: Grouped by CONCAT(YEAR, QUARTER) and product; aggregated quantity and revenue.
Purpose: Help the business spot seasonal demand spikes and allocate resources effectively.

**Python Chart:**



**Python Code Used:**

```
top_products = df_quarter.groupby('Product')['TotalRevenue'].sum().nlargest(10).index
filtered_df = df_quarter[df_quarter['Product'].isin(top_products)]

plt.figure(figsize=(12, 6))
sns.barplot(
    data=filtered_df,
    x='Quarter',
    y='TotalRevenue',
    hue='Product'
)

plt.title('Top 10 Products by Revenue per Quarter')
plt.xlabel('Quarter')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.legend(title='Product', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
```

**Python Explanation:**

Data Prep: Selected top 10 products overall, then plotted revenue per quarter.
Chart Type: Bar plot grouped by quarter and product to compare sales performance.
Design Choice: Used clear color palette and saved chart for reporting use.

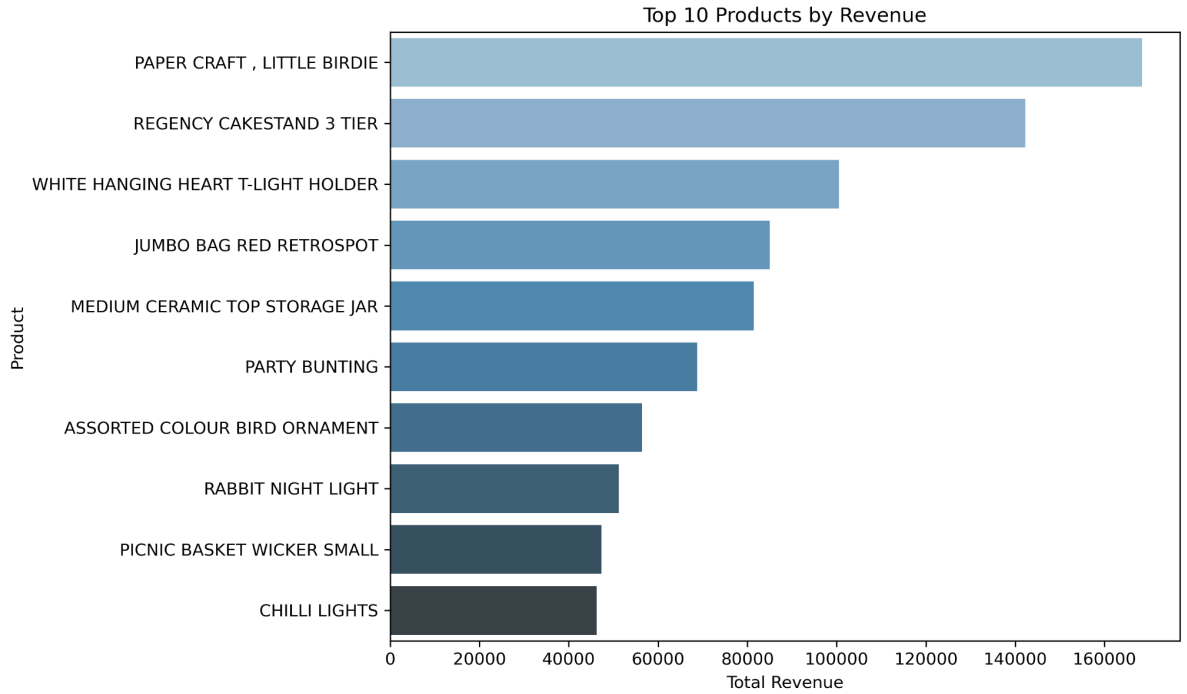
## 2.3.3. Top Products Overall (by Quantity and Revenue)

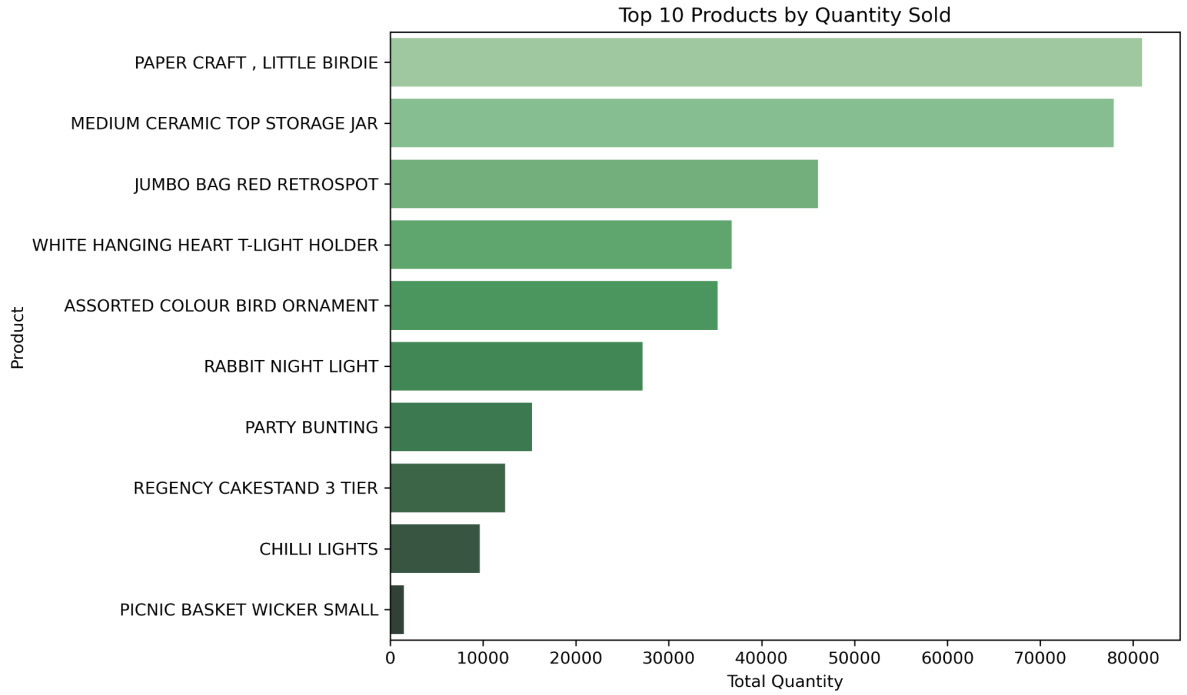
**SQL Query Used:**

```
SELECT
    p.Description AS Product,
    SUM(o.Quantity) AS TotalQuantity,
    SUM(o.TotalPrice) AS TotalRevenue
FROM orders o
JOIN products p ON o.StockCode = p.StockCode
GROUP BY Product
ORDER BY TotalRevenue DESC
LIMIT 10;
```

**SQL Explanation:**
Goal: Identify the top 10 best-selling products overall.
Logic: Aggregated total quantity and revenue by product; ordered by revenue.
Purpose: Recognize consistently high-performing products for inventory prioritization.

## Python Chart:



**Python Code Used:**

```
Top Products by Revenue
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='TotalRevenue', y='Product', palette='Blues_d')

plt.title('Top 10 Products by Revenue')
plt.xlabel('Total Revenue')
plt.ylabel('Product')
plt.tight_layout()
plt.savefig('top_10_products_by_revenue.png', dpi=300)
plt.show()

Top Products by Quantity Sold
df_quantity = df.sort_values(by='TotalQuantity', ascending=False)

plt.figure(figsize=(10, 6))
sns.barplot(data=df_quantity, x='TotalQuantity', y='Product', palette='Greens_d')

plt.title('Top 10 Products by Quantity Sold')
plt.xlabel('Total Quantity')
plt.ylabel('Product')
plt.tight_layout()
plt.savefig('top_10_products_by_quantity.png', dpi=300)
plt.show()
```

**Python Explanation:**

Data Prep: Sorted data by revenue and quantity separately.
Chart Type: Two horizontal bar charts (one for revenue, one for quantity).
Design Choice: Color-coded bars, simple layout, and exported high-resolution images.

## 2.3.4. Top Products by Country (by Revenue)

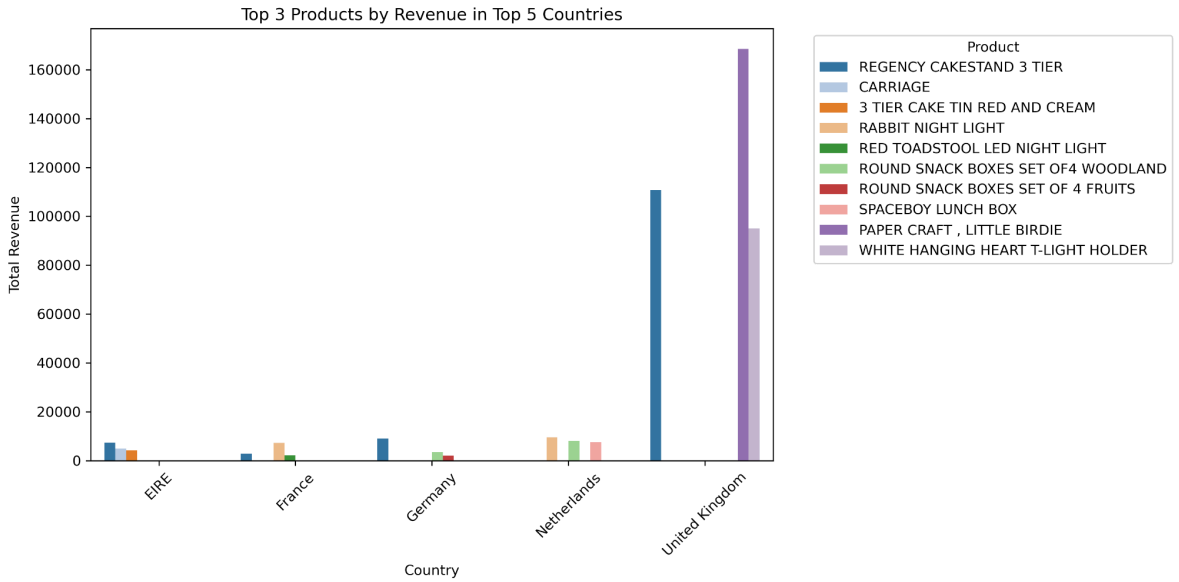
**SQL Query Used:**

```
SELECT
   c.Country,
   p.Description AS Product,
   SUM(o.TotalPrice) AS TotalRevenue,
   SUM(o.Quantity) AS TotalQuantity
FROM orders o
JOIN products p ON o.StockCode = p.StockCode
JOIN invoices i ON o.InvoiceNo = i.InvoiceNo
JOIN customers c ON i.CustomerID = c.CustomerID
GROUP BY c.Country, Product
ORDER BY c.Country, TotalRevenue DESC;
```

**SQL Explanation:**

Goal: Determine which products perform best in each country.
Logic: Joined customers, invoices, products, and orders; grouped by country and product; aggregated revenue and quantity.
Purpose: Reveal regional product preferences to inform localized strategy.

**Python Chart:**



**Python Code Used:**

```
top_countries = df.groupby('Country')['TotalRevenue'].sum().nlargest(5).index
filtered_df = df[df['Country'].isin(top_countries)]

top_products_per_country = (
    filtered_df
    .sort_values(['Country', 'TotalRevenue'], ascending=[True, False])
    .groupby('Country')
    .head(3)
)

num_products = top_products_per_country['Product'].nunique()
custom_palette = sns.color_palette("tab20", num_products)

unique_products = top_products_per_country['Product'].unique()
product_color_map = dict(zip(unique_products, custom_palette))

plt.figure(figsize=(12, 6))
sns.barplot(
    data=top_products_per_country,
    x='Country',
    y='TotalRevenue',
    hue='Product',
    palette=product_color_map
)

plt.title('Top 3 Products by Revenue in Top 5 Countries')
plt.xlabel('Country')
plt.ylabel('Total Revenue')
plt.xticks(rotation=45)
plt.legend(title='Product', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.savefig('top_products_by_country_distinct_colors.png', dpi=300)
plt.show()
```

**Python Explanation:**

Data Prep: Selected top 5 countries by total revenue, then top 3 products within each.
Chart Type: Grouped bar chart with products on hue and countries on x-axis.
Design Choice: Applied a custom color palette to avoid confusion; exported for use in final report.

# 3. Sales & Product Performance Analysis

**Business Problem**
What are the top-performing products and how do their sales vary over time and across countries?

## 3.1. Total Revenue by Product by Month

**What the chart shows:**

- Monthly sales patterns of top 10 products over time.
- Most top 10 products had steady monthly revenue.
- Four products showed unusual one-month sales spikes. This suggests an extraordinary one-off purchase or event-related demand, possibly tied to seasonal promotions, holidays, or bulk client orders. Such spikes should be reviewed for repeatability or forecasting value.

    1. In January, the Medium Ceramic Top Storage Jar generated £77,000 in revenue — over 145 times its average monthly revenue of £530.
    2. In June, Picnic Basket Wicker Small generated £40,000 — nearly 49 times its average of £816.
    3. Rabbit Night Light saw two consecutive spikes: £11,955 in October (4.5x average) and £23,190 in November (8.6x average), compared to its monthly average of £2,684. This suggests building demand leading into the holiday season
    4. In December, Paper Craft, Little Birdie generated £168,470 — nearly 40 times the average revenue (£4,254) of the other top products that month, strongly indicating a one-time bulk purchase or exceptional holiday-driven demand.

- Revenue in June 2011 dropped by 28% compared to the monthly average. This could reflect seasonal low demand or supply issues. Investigating inventory levels or campaign activity during that time could provide context.

**Business Insight:**

- Top 10 products show consistent demand and are critical to inventory planning and revenue stability
- Several products experience strong sales which could reflect large one-time bulk orders, promotions or seasonality likely due to holidays.

**Recommendation:**

- For all products, maintain stock year-round and consider bundling these with slower-moving items

- For four products with sales spikes, investigate these orders for context. Factor similar events into future demand forecasting. Plan promotions and increase stock levels ahead of the holiday season for these specific items.

## 3.2. Top Products by Quarter (Revenue & Quantity)

**What the chart shows:**

- A consistent group of best-selling products appears across all quarters.
- There is no significant sales uplift in Q4 (Oct–Dec), suggesting a lack of strong holiday-driven demand or ineffective seasonal promotions.

**Business Insight:**

- Core products maintain steady sales throughout the year, indicating stable demand independent of seasonality.
- The absence of a Q4 revenue spike, often expected during the holiday period, points to missed promotional opportunities or product misalignment with seasonal trends.

**Recommendation:**

- Ensure continuous inventory availability for consistently performing products throughout the year.
- Reevaluate and optimize Q4 promotional strategies to better capture seasonal demand.
- Identify and reposition products with strong gifting or seasonal appeal to boost Q4 performance.

## 3.3. Top Products Overall (by Revenue and Quantity)

**What the chart shows:**

- Top 10 products by revenue and by quantity.
- Lower-priced products, like the Medium Ceramic Top Storage Jar, sold in high volumes (79,000 units) but generated relatively modest revenue (£81,000), indicating a low unit price and thin margins. In contrast, premium items such as the Regency Cake Stand sold fewer units (14,000) yet generated significantly higher revenue (£140,000).

**Business Insight:**

- There's a mix of high-volume and high-margin products.
- This highlights the importance of balancing high-volume, lower-margin products that drive cash flow with high-margin products that maximize profitability per unit.

**Recommendation:**

- Promote premium products for profitability.
- Bundle or discount high-volume products to increase upselling potential.

## 3.4. Top Products by Country (by Revenue)

**What the chart shows:**

- Each country has its own top-selling products.
- The UK market overwhelmingly dominates sales, generating £374,143 in revenue from the top 3 products, while the highest revenue from any other country does not exceed £25,000. This highlights a significant concentration of demand in the UK, with international markets currently representing a small fraction of total sales.

**Business Insight:**

- There are clear regional differences in product demand.
- The concentration of sales in UK suggests a strong home market presence but also indicates significant growth opportunities in other regions.
- Tailored marketing and localization efforts could help capture more revenue internationally, especially focusing on the top-performing products.

**Recommendation:**

- Customize product marketing by country.
- Focus on top 3 products per market to drive regional sales.

# 4. Insights & Recommendations

**Stock & Inventory Planning**

- Keep core products consistently available.
- Anticipate seasonal/event-driven spikes and adjust stock accordingly.

**Product Strategy**

- Promote premium products (higher margins).
- Bundle or discount high-volume products to increase cart value.

**Marketing Strategy**

- Tailor campaigns to top products per region.
- Strengthen Q4 promotions with product bundles and premium positioning.

# 5. Final Summary

This analysis revealed distinct seasonal patterns and regional product preferences. Certain products drive revenue consistently throughout the year, while others are strongly seasonal. Additionally, top-selling products vary by country, indicating an opportunity for localized strategies. By focusing on timing, top performers, and regional trends, the business can improve inventory efficiency and increase revenue through targeted marketing.

**Skills & Tools Demonstrated**

SQL: Joins, aggregations, segmentation queries
Python: Data cleaning (Pandas), visualization (Matplotlib, Seaborn), clustering (Scikit-learn)
Business Analysis: Insights into product trends
Data Storytelling: Translating analysis into actionable business recommendations