**DataVista: Sales Data Analysis and Visualization**

**Objective**

The objective of this project is to analyze and visualize bakery sales data to uncover key trends, identify top-performing products, and explore the impact of time-based patterns on revenue. The goal is to provide actionable insights that will help optimize inventory, staffing, and promotional strategies.

**Data Sources**

1. **Bakery Sales Data**: Contains sales transactions, including details on the date, time, products sold, and transaction totals.
2. **Bakery Price Data**: Contains product names and their respective prices.

**Libraries Required**

* **Pandas**: For data manipulation and cleaning.
* **NumPy**: For numerical operations and array manipulations.
* **Matplotlib**: For creating basic visualizations.
* **Seaborn**: For advanced visualizations, especially for statistical graphics.

**Project Outline**

**1. Data Loading and Initial Exploration**

* Load the sales and price datasets.
* Display basic information and preview the datasets to understand their structure.

**2. Data Cleaning**

* Convert price in the price dataset to a numeric type.
* Convert datetime in the sales dataset to a proper datetime format.
* Drop any rows in the sales dataset where key data points (e.g., datetime or total) are missing.

**3. Data Transformation and Merging**

* **Reshape Sales Data**: Reshape the sales dataset to a long format where each row represents a single product sale within a transaction.
* **Merge Datasets**: Merge the reshaped sales data with the price dataset to associate each product with its price.
* **Calculate Revenue**: Create a new column for revenue by multiplying the quantity of each product sold by its price.

**4. Data Visualization**

**4.1 Monthly Sales Trend**

* **Objective**: To visualize the revenue trend over time.
* **Method**: Aggregate daily revenue by month and plot the monthly trend to observe seasonal patterns or growth.
* **Code**:

# Aggregate revenue by month

monthly\_sales = merged\_data['revenue'].resample('M').sum()

# Plot monthly sales trend

plt.figure(figsize=(10, 6))

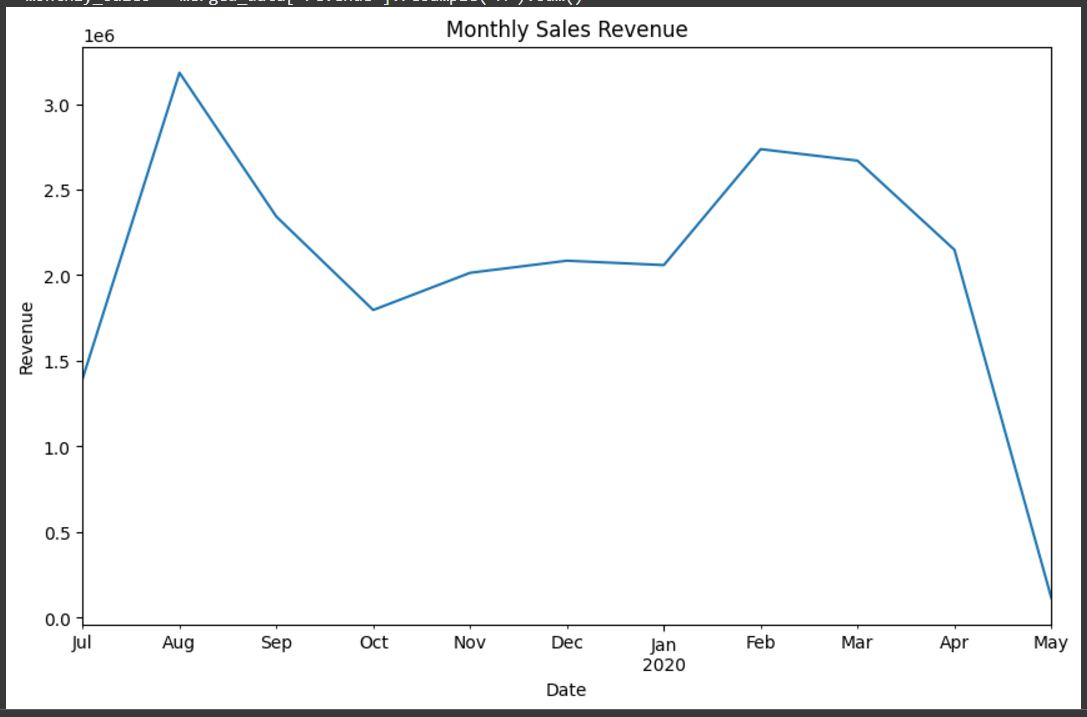
monthly\_sales.plot()

plt.title('Monthly Sales Revenue')

plt.xlabel('Date')

plt.ylabel('Revenue')

plt.show()



**4.2 Top Products by Quantity and Revenue**

* **Objective**: Identify top-selling products in terms of both quantity and revenue.
* **Method**: Group data by product, calculate the total quantity sold and total revenue for each product, and display the top 10 products.
* **Code**:

# Total quantity and revenue per product

product\_sales = merged\_data.groupby('product')['quantity'].sum()

product\_revenue = merged\_data.groupby('product')['revenue'].sum()

# Plot the top 10 products by quantity and revenue

fig, ax = plt.subplots(1, 2, figsize=(15, 6))

product\_sales.sort\_values(ascending=False).head(10).plot(kind='bar', ax=ax[0], color='skyblue')

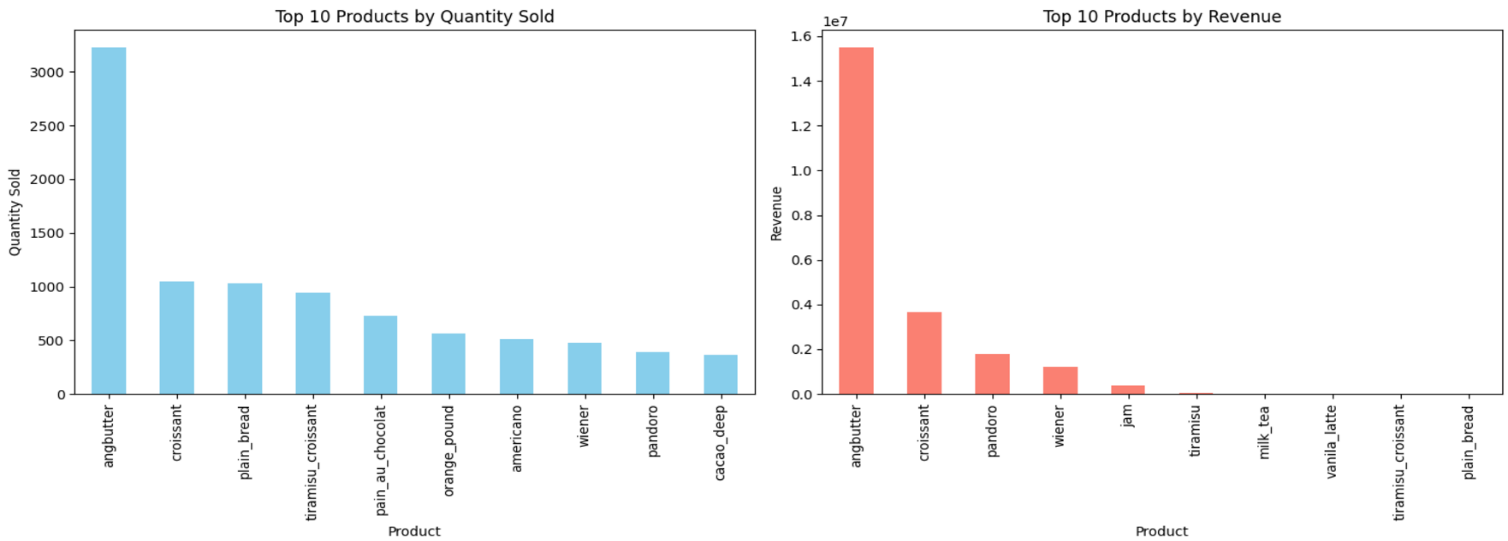
ax[0].set\_title('Top 10 Products by Quantity Sold')

product\_revenue.sort\_values(ascending=False).head(10).plot(kind='bar', ax=ax[1], color='salmon')

ax[1].set\_title('Top 10 Products by Revenue')

plt.tight\_layout()

plt.show()



**4.3 Revenue Contribution by Product**

* **Objective**: Visualize each product's contribution to total revenue.
* **Method**: Calculate the percentage contribution of each product’s revenue and display it as a pie chart.
* **Code**:

product\_revenue\_contrib = product\_revenue / product\_revenue.sum() \* 100

# Plot the revenue contribution by product

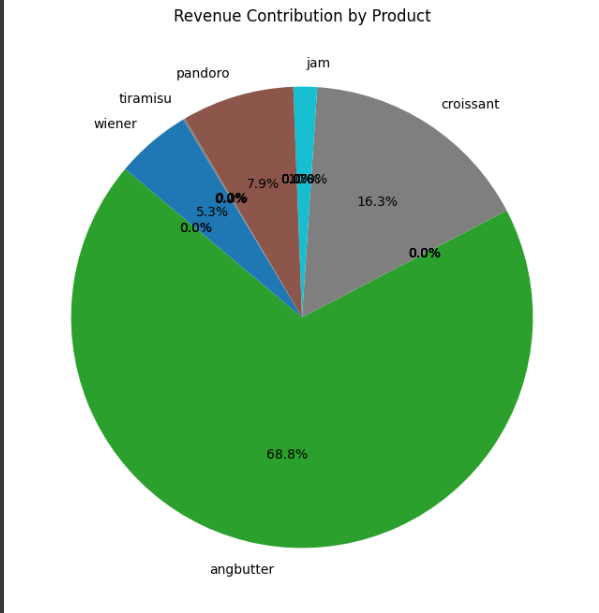
plt.figure(figsize=(10, 8))

product\_revenue\_contrib.plot(kind='pie', autopct='%1.1f%%', startangle=140)

plt.title('Revenue Contribution by Product')

plt.ylabel('')

plt.show()



**4.4 Sales by Day of the Week**

* **Objective**: Observe trends in sales based on the day of the week to understand customer behavior.
* **Method**: Group data by day of the week and calculate total revenue for each day.
* **Code**:

day\_of\_week\_sales = merged\_data.groupby('day\_of\_week')['revenue'].sum()

# Plot total revenue by day of the week

plt.figure(figsize=(10, 6))

day\_of\_week\_sales.plot(kind='bar', color='lightgreen')

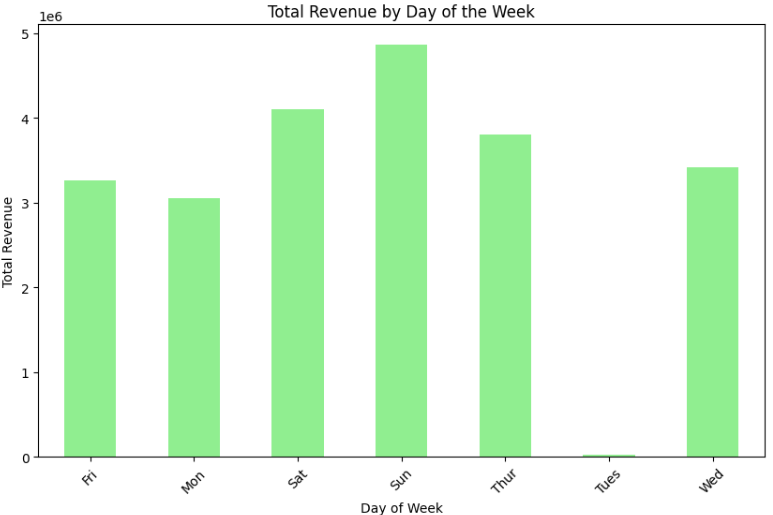
plt.title('Total Revenue by Day of the Week')

plt.xlabel('Day of Week')

plt.ylabel('Total Revenue')

plt.xticks(rotation=45)

plt.show()



**4.5 Product Sales Correlation Heatmap**

* **Objective**: Explore correlations between product sales to see if certain products are frequently purchased together.
* **Method**: Create a heatmap of product correlations based on sales quantities.
* **Code**:

product\_data = merged\_data.pivot\_table(index='datetime', columns='product', values='quantity', aggfunc='sum').fillna(0)

product\_corr = product\_data.corr()

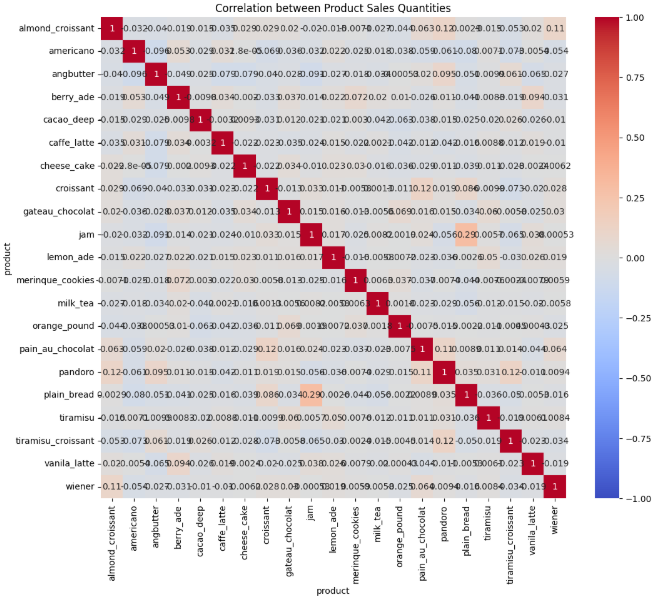
# Plot the heatmap

plt.figure(figsize=(12, 10))

sns.heatmap(product\_corr, annot=True, cmap='coolwarm', vmin=-1, vmax=1)

plt.title('Correlation between Product Sales Quantities')

plt.show()



**Summary and Insights**

Each visualization provides unique insights:

* **Monthly Sales Trend**: Shows growth patterns and seasonality in sales.
* **Top Products by Quantity and Revenue**: Identifies best-selling and high-revenue products.
* **Revenue Contribution by Product**: Highlights which products contribute most to total revenue.
* **Sales by Day of the Week**: Reveals patterns in sales based on the day to inform staffing and promotions.
* **Product Sales Correlation Heatmap**: Shows associations between products, useful for bundling and promotions.