# Product Sales Analysis Using Python

To perform a product sales analysis using Python, we'll walk through a basic example of how to analyze sales data, calculate key metrics, and visualize the results. We'll use libraries such as Pandas for data manipulation, Matplotlib for visualization, and NumPy for numerical operations.

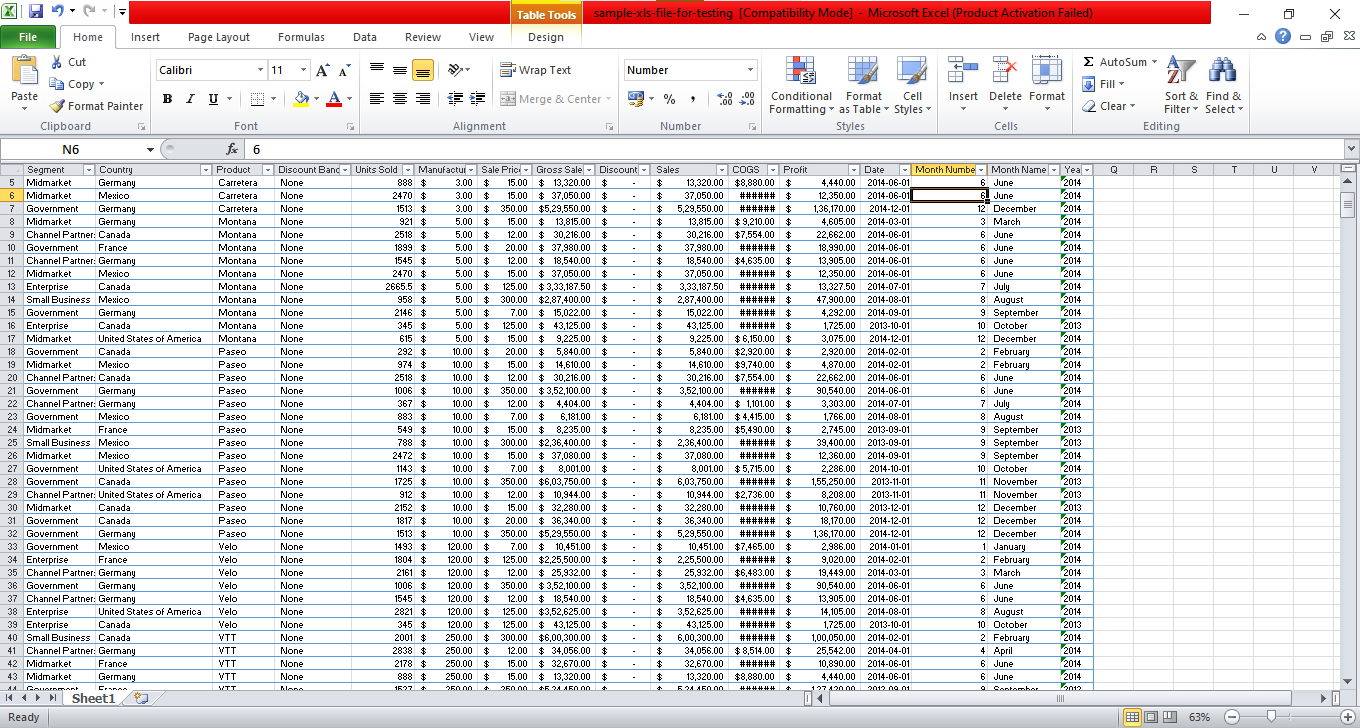
Assuming you have sales data in a CSV file named "sales\_data.csv" with columns like 'Product', 'Date', 'Revenue', and 'Quantity', here's a step-by-step approach

**OVERVIEW**

In this post, I use Python Pandas & Python Matplotlib to analyze and answer business questions about 12 months worth of sales data. The data contains hundreds of thousands of electronics store purchases broken down by month, product type, cost, purchase address, etc. The dataset can be downloaded [here](https://github.com/alfifutuhi13/miniproject1/blob/main/Pandas-Data-Science-Tasks-master.zip). In this analysis, I’m using jupyter notebook.



**Sample Data Base:**

****

**To conduct a more comprehensive product sales analysis in Python, we'll cover various aspects such as data preprocessing, exploratory data analysis (EDA), key metrics calculation, and visualization. We'll use sample sales data for demonstration purposes**.

**1. Import Necessary Libraries:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

**2. Load and Explore the Data:**

**Assuming you have a CSV file named "sales\_data.csv" containing relevant sales data.**

**# Load the sales data into a DataFrame**

**sales\_data = pd.read\_csv('sales\_data.csv')**

**# Display basic information about the data**

**print(sales\_data.info())**

**# Display the first few rows of the DataFrame**

**print(sales\_data.head())**

**3. Data Preprocessing:**

**Ensure the data is in the appropriate format and handle any missing or incorrect values.**

**# Convert the 'Date' column to datetime format**

**sales\_data['Date'] = pd.to\_datetime(sales\_data['Date'])**

**# Check for missing values**

**print('Missing values:\n', sales\_data.isnull().sum())**

**# Drop rows with missing values**

**sales\_data.dropna(inplace=True)**

**4. Key Metrics Calculation:**

**Calculate key metrics such as total revenue, total quantity sold, and average selling price.**

**# Total revenue**

**total\_revenue = sales\_data['Revenue'].sum()**

**# Total quantity sold**

**total\_quantity\_sold = sales\_data['Quantity'].sum()**

**# Average selling price**

**average\_selling\_price = total\_revenue / total\_quantity\_sold**

**print('Total Revenue:', total\_revenue)**

**print('Total Quantity Sold:', total\_quantity\_sold)**

**print('Average Selling Price:', average\_selling\_price)**

**5. Exploratory Data Analysis (EDA):**

**Explore the data to understand the distribution and relationships between variables.**

**# Summary statistics**

**print(sales\_data.describe())**

**# Visualize the distribution of revenue and quantity sold**

**plt.figure(figsize=(12, 6))**

**sns.histplot(sales\_data['Revenue'], bins=30, kde=True)**

**plt.title('Distribution of Revenue')**

**plt.xlabel('Revenue')**

**plt.ylabel('Frequency')**

**plt.show()**

**plt.figure(figsize=(12, 6))**

**sns.histplot(sales\_data['Quantity'], bins=30, kde=True)**

**plt.title('Distribution of Quantity Sold')**

**plt.xlabel('Quantity Sold')**

**plt.ylabel('Frequency')**

**plt.show()**

**6. Product Performance Analysis:**

**Analyze the performance of products based on revenue and quantity sold.**

**# Group data by product and calculate total revenue and total quantity sold for each product**

**product\_performance = sales\_data.groupby('Product').agg({'Revenue': 'sum', 'Quantity': 'sum'}).reset\_index()**

**# Sort products by revenue in descending order**

**product\_performance = product\_performance.sort\_values(by='Revenue', ascending=False)**

**# Display the top-performing products**

**print('Top Performing Products:')**

**print(product\_performance.head())**

**# Visualize top performing products**

**plt.figure(figsize=(12, 6))**

**sns.barplot(x='Product', y='Revenue', data=product\_performance.head(10))**

**plt.xticks(rotation=45)**

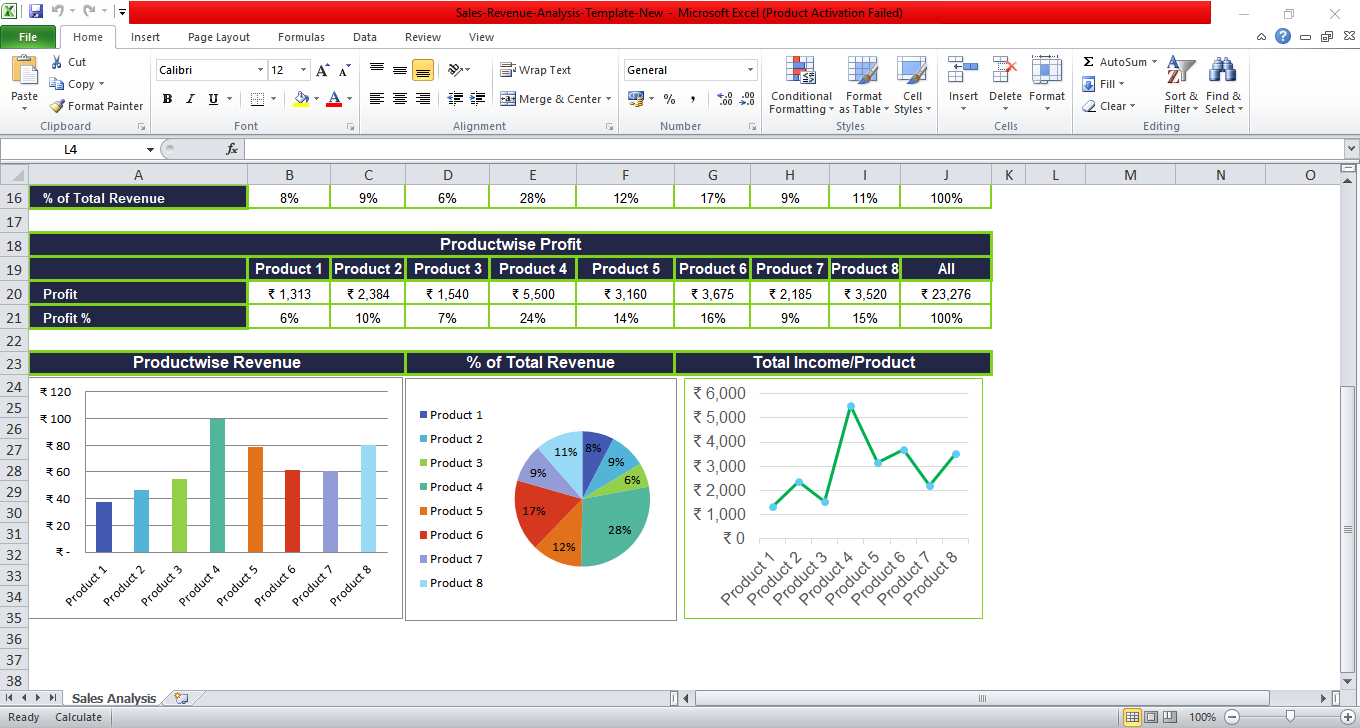
**plt.title('Top Performing Products by Revenue')**

**plt.xlabel('Product')**

**plt.ylabel('Total Revenue')**

**plt.show()**

**You can further extend this analysis to include customer segmentation, market basket analysis, seasonality analysis, and other advanced techniques to derive valuable insights from your sales data. Modify and customize the analysis based on the specific requirements of your dataset and business needs.**

****

## Advantages of Sales Revenue Analysis

* It can be an effective tool for marketing and sales teams for achieving and defining targets.
* This analytics can be useful for new startups, online retail sales, or any other small business to track their sales and profits.
* From such analysis, you get insight to improve in areas where products and services aren’t performing well. This helps to make informed decisions.
* Sales Revenue analysis helps us to determine profitability.
* You can design detailed and feasible plans for the future based on these data.
* Moreover, it helps the business to know where to invest and how to invest.
* Design marketing campaigns and allocate an appropriate budget for these activities.