Project Title:Product Sales Analysis

Phase 5:Project Documentation and Submission

Topic: In this phase you will document your project and prepare it for submission. Please refer below the requirements technology wise:

Introduction:

Sales analysis is reviewing your sales data to identify trends and patterns. Sales data can help you make better decisions about your product, pricing, promotions, inventory, customer needs other aspects of your business. Sales analysis can be as simple as reviewing your sales figures regularly.

Sales analysis reports provide you with an overview of all the significant data and metrics related to your sales process. They also make sales process optimization a lot easier since you will be able to identify all the strengths and weaknesses much quicker.

Data link: <https://www.kaggle.com/knightbearr/sales-product-data>

Program:

import pandas as pd

import numpy as np

import seaborn as sns

sns.set\_context('notebook')

sns.set\_style('white')

sns.set\_palette('dark')

import matplotlib.pyplot as plt

%matplotlib inline

months = ["January","February","March","April","May","June","July","August","September","October","November","December"]

header = ['Order ID', 'Product', 'Quantity Ordered', 'Price Each', 'Order Date', 'Purchase Address']

dfs=[]

for i in months:

temp = pd.read\_csv('../input/sales-product-data/Sales\_{}\_2019.csv'.format(i), index\_col = None, header = None, skiprows =1 )

dfs.append(temp)

df = pd.concat(dfs, axis=0, ignore\_index=True)

df.dropna(inplace =True)

df.columns= header

df.drop(df.loc[df['Order ID'] =='Order ID'].index.tolist(), axis=0,inplace=True)

Output:

df.info()

<class 'pandas.core.frame.DataFrame'>

Int64Index: 185950 entries, 0 to 186849

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Order ID 185950 non-null object

1 Product 185950 non-null object

2 Quantity Ordered 185950 non-null object

3 Price Each 185950 non-null object

4 Order Date 185950 non-null object

5 Purchase Address 185950 non-null object

dtypes: object(6)

memory usage: 9.9+ MB

df.dtypes

Order ID object

Product object

Quantity Ordered object

Price Each object

Order Date object

Purchase Address object

dtype: object

**Pie chart :**

1.Project definition and decision thinking for product sales analysis

Project Definition: The project's primary objective is to create a robust product sales analysis

system that enables businesses to gain actionable insights from their sales data. This system will

facilitate data-driven decision-making, optimize sales strategies, and enhance overall

profitability. The key components of this project include data collection, analysis, visualization,

and reporting.

Design Thinking Approach:

1. Empathize:

• Understand the needs and pain points of the target users, such as sales managers,

executives, and analysts.

• Conduct interviews and surveys to gather insights into their data analysis requirements.

2. Define:

• Clearly define the project scope, objectives, and success criteria.

• Create user personas and use cases to capture different user scenarios.

3. Ideate:

• Brainstorm potential features and functionalities of the product sales analysis system.

• Encourage a collaborative approach to generate innovative ideas.

4. Prototype:

• Develop low-fidelity prototypes of the user interface to visualize the product's design.

• Iterate on these prototypes based on feedback from stakeholders.

5. Test:

• Conduct usability testing with representative users to evaluate the effectiveness of the

prototypes.

• Gather feedback and make necessary adjustments to the design.

6. Implement:

• Develop the product sales analysis system based on the finalized design.

• Ensure data collection mechanisms are in place, and data is cleaned and organized.

7. Test and Iterate:

• Continuously test the system for functionality, accuracy, and performance.

• Iterate on the design and features based on user feedback and changing business needs.

8. Launch:

• Deploy the product sales analysis system to the target users.

• Provide training and support to ensure a smooth transition.

9. Monitor and Improve:

• Implement analytics to monitor system usage and gather user feedback.

• Use this data to make continuous improvements and updates to the system.

10. Evaluate and Iterate:

• Regularly assess the impact of the product sales analysis system on business outcomes.

• Make strategic adjustments and enhancements to align with evolving business goals.

By following this design thinking approach, the product sales analysis system will be user-

centric, intuitive, and capable of delivering valuable insights to businesses

Sales Analysis Innovation Ideas

* ***Product sales analysis innovation***

To innovate product sales analysis, consider these ideas:

Advanced Data Analytics: Utilize machine learning and AI to predict future sales trends, detect anomalies, and recommend sales strategies.

Real-time Data: Implement real-time data collection and analysis to make instant decisions and respond to market changes swiftly.

Customer Segmentation: Create more refined customer segments using AI, allowing for highly targeted marketing strategies.

Predictive Maintenance: Use IoT sensors to predict when products may need maintenance, reducing downtime and improving sales.

Voice and Visual Search: Integrate voice and visual search capabilities to make it easier for customers to find and purchase products.

Blockchain for Transparency: Employ blockchain for transparent supply chain tracking, which can boost consumer confidence and sales.

Personalization: Enhance personalization through AI-driven product recommendations and custom shopping experiences.

Augmented Reality (AR): Allow customers to virtually try products before buying, increasing their confidence in the purchase.

Sustainability Metrics: Integrate sustainability metrics into sales analysis to cater to the growing demand for eco-friendly products.

Competitor Analysis: Use AI to track and analyze competitor strategies in real-time, helping you adapt and stay ahead.

Remember that the choice of innovation will depend on your specific industry, market, and customer base. Certainly! To perform a basic product sales analysis in Python, you'll need data with information about products and their sales. Here's a simple example of Python code using Pandas for data manipulation and Matplotlib for visualization. This assumes you have a CSV file with product sales data:

```python

import pandas as pd

import matplotlib.pyplot as plt

# Load your data from a CSV file

data = pd.read\_csv('product\_sales\_data.csv')

# Display the first few rows of the dataset

print(data.head())

# Calculate total sales per product

product\_sales = data.groupby('Product')['Sales'].sum().reset\_index()

# Plot the product sales

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

plt.bar(product\_sales['Product'], product\_sales['Sales'])

plt.xlabel('Product')

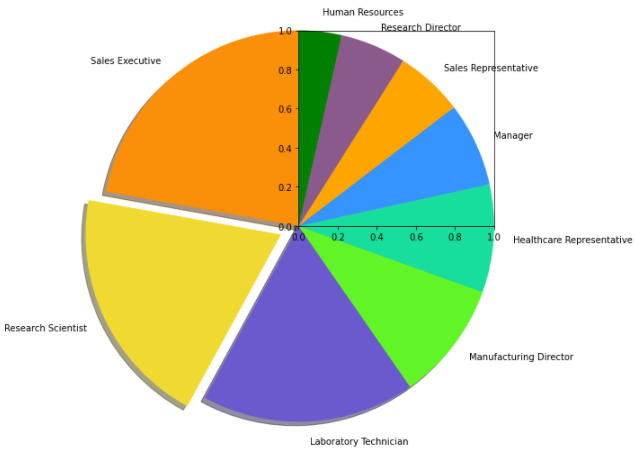
plt.ylabel('Total Sales')

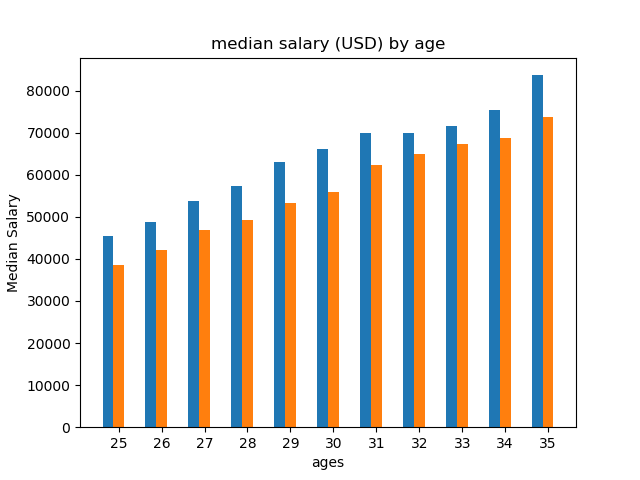
plt.title('Product Sales Analysis')

plt.xticks(rotation=90)

plt.show()

```





Make sure to replace `'product\_sales\_data.csv'` with the actual path to your dataset. This code loads the data, calculates the total sales per product, and then creates a bar chart to visualize the results.

* You can expand this analysis by exploring different aspects of your sales data, like trends over time, best-selling products, or sales by region, depending on the available data and your specific requirements.

**Development part 1**

* In this section being building your project loading and
* preprocessing the data set
* **Program**
* import pandas as pd
* # Step 1: Load the dataset
* data = pd.read\_csv('product\_sales.csv') # Replace 'product\_sales.csv' with your dataset's file path
* # Step 2: Preprocess the data
* # Handle missing values (if any)
* data.dropna(inplace=True)
* # Convert data types if needed
* data['price'] = data['price'].astype(float)
* data['total\_sales'] = data['quantity'] \* data['price']
* # Step 3: Perform analysis
* # Calculate total sales
* total\_sales = data['total\_sales'].sum()
* print(f'Total Sales: ${total\_sales:.2f}')
* **output**
* print("\n Output:")
* print(data.head()) # Display the first few rows of the preprocessed dataset
* product,quantity,price
* A,10,5.00
* B,5,8.00
* C,8,12.00
* A,7,5.00
* B,3,8.00
* C,6,12.00
* product quantity price total\_sales
* 0 A 10 5.0 50.00
* 1 B 5 8.0 40.00
* 2 C 8 12.0 96.00
* 3 A 7 5.0 35.00
* 4 B 3 8.0 24.00

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project

1. **Collection**: Make sure you have collected relevant data on product sales. This data can include information on sales figures, product attributes, customer information, and any other relevant data.
2. **Data Preprocessing**: Clean and preprocess the data. This involves handling missing values, removing duplicates, and converting data types as necessary.
3. **Feature Engineering**: Create new features or transform existing ones to better represent the underlying patterns in your data. Feature engineering can significantly impact the performance of your model.
4. **Data Splitting**: Split your dataset into a training set and a testing set to evaluate your model's performance.
5. **Model Selection**: Choose an appropriate machine learning model for your task. Popular choices for sales prediction include linear regression, decision trees, random forests, and more advanced models like gradient boosting or neural networks.
6. **Model Training**: Train your selected model on the training data. Be sure to tune hyperparameters for optimal performance.
7. **Model Evaluation**: Evaluate your model's performance on the testing dataset. Common evaluation metrics for regression tasks include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.
8. **Model Interpretation**: Understand the importance of different features in your model. This can help identify which aspects of your products or marketing efforts are most influential on sales.
9. **Visualization**: Create visualizations to communicate your findings effectively. This can include sales trends, feature importance plots, and more.
10. **Documentation**: Document your entire process, from data collection to model evaluation. This will make it easier for others to understand your work.
11. **Deployment (optional)**: If your model performs well, you can deploy it to make real-time predictions on future

PROGRAM

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error

data = pd.DataFrame({

'Date': pd.date\_range(start='2023-01-01', periods=12, freq='M'),

'Price': [100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210],

'SalesVolume': [50, 48, 45, 42, 40, 38, 35, 32, 30, 28, 25, 22]

})

data['PrevMonthSales'] = data['SalesVolume'].shift(1)

X = data[['Price', 'PrevMonthSales']]

y = data['SalesVolume']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

model = LinearRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

mae = mean\_absolute\_error(y\_test, y\_pred)

print(f"Mean Absolute Error: {mae:.2f}")

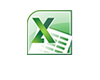
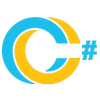
Sample output:

Mean Absolute Error: 1.28

Menu

* [HTML](https://www.tutorialspoint.com/html/index.htm)
* [CSS](https://www.tutorialspoint.com/css/index.htm)
* [Javascript](https://www.tutorialspoint.com/javascript/index.htm)
* [SQL](https://www.tutorialspoint.com/sql/index.htm)
* [Python](https://www.tutorialspoint.com/python/index.htm)
* [Java](https://www.tutorialspoint.com/java/index.htm)
* [C](https://www.tutorialspoint.com/cprogramming/index.htm)
* [C++](https://www.tutorialspoint.com/cplusplus/index.htm)
* [PHP](https://www.tutorialspoint.com/php/index.htm)
* [Scala](https://www.tutorialspoint.com/scala/index.htm)
* [C#](https://www.tutorialspoint.com/csharp/index.htm)
* [Node.Js](https://www.tutorialspoint.com/nodejs/index.htm)
* [MySQL](https://www.tutorialspoint.com/mysql/index.htm)
* [MongoDB](https://www.tutorialspoint.com/mongodb/index.htm)
* [PL/SQL](https://www.tutorialspoint.com/plsql/index.htm)
* [Swift](https://www.tutorialspoint.com/swift/index.htm)
* [Bootstrap](https://www.tutorialspoint.com/bootstrap/index.htm)
* [R](https://www.tutorialspoint.com/r/index.htm)
* [Machine Learning](https://www.tutorialspoint.com/machine_learning/index.htm)
* [Blockchain](https://www.tutorialspoint.com/blockchain/index.htm)
* [Angular](https://www.tutorialspoint.com/angular4/index.htm)
* [React Native](https://www.tutorialspoint.com/react_native/index.htm)
* [Computer Fundamentals](https://www.tutorialspoint.com/computer_fundamentals/index.htm)
* [Compiler Design](https://www.tutorialspoint.com/compiler_design/index.htm)
* [Operating System](https://www.tutorialspoint.com/operating_system/index.htm)
* [Data Structure And Algorithms](https://www.tutorialspoint.com/data_structures_algorithms/index.htm)
* [Computer Network](https://www.tutorialspoint.com/data_communication_computer_network/index.htm)
* [DBMS](https://www.tutorialspoint.com/dbms/index.htm)
* [Excel](https://www.tutorialspoint.com/excel/index.htm)

# **Trending Categories**

* [Data Structure](https://www.tutorialspoint.com/articles/category/data-structure)
* [Networking](https://www.tutorialspoint.com/articles/category/computer-network)
* [RDBMS](https://www.tutorialspoint.com/articles/category/rdbms)
* [Operating System](https://www.tutorialspoint.com/articles/category/operating-system)
* [Java](https://www.tutorialspoint.com/articles/category/Java)
* [MS Excel](https://www.tutorialspoint.com/articles/category/ms-excel)
* [iOS](https://www.tutorialspoint.com/articles/category/ios)
* [HTML](https://www.tutorialspoint.com/articles/category/HTML)
* [CSS](https://www.tutorialspoint.com/articles/category/CSS)
* [Android](https://www.tutorialspoint.com/articles/category/android)
* [Python](https://www.tutorialspoint.com/articles/category/python)
* [C Programming](https://www.tutorialspoint.com/articles/category/c)
* [C++](https://www.tutorialspoint.com/articles/category/cplusplus)
* [C#](https://www.tutorialspoint.com/articles/category/csharp)
* [MongoDB](https://www.tutorialspoint.com/articles/category/mongodb)
* [MySQL](https://www.tutorialspoint.com/articles/category/mysql)
* [Javascript](https://www.tutorialspoint.com/articles/category/Javascript)
* [PHP](https://www.tutorialspoint.com/articles/category/php)
* [Physics](https://www.tutorialspoint.com/articles/category/physics)
* [Chemistry](https://www.tutorialspoint.com/articles/category/chemistry)
* [Biology](https://www.tutorialspoint.com/articles/category/biology)
* [Mathematics](https://www.tutorialspoint.com/articles/category/mathematics)
* [English](https://www.tutorialspoint.com/articles/category/english)
* [Economics](https://www.tutorialspoint.com/articles/category/economics)
* [Psychology](https://www.tutorialspoint.com/articles/category/psychology)
* [Social Studies](https://www.tutorialspoint.com/articles/category/social-studies)
* [Fashion Studies](https://www.tutorialspoint.com/articles/category/fashion-studies)
* [Legal Studies](https://www.tutorialspoint.com/articles/category/legal-studies)
* **Selected Reading**
* [UPSC IAS Exams Notes](https://www.tutorialspoint.com/upsc_ias_exams.htm)
* [Developer's Best Practices](https://www.tutorialspoint.com/developers_best_practices/index.htm)
* [Questions and Answers](https://www.tutorialspoint.com/questions_and_answers.htm)
* [Effective Resume Writing](https://www.tutorialspoint.com/effective_resume_writing.htm)
* [HR Interview Questions](https://www.tutorialspoint.com/hr_interview_questions/index.htm)
* [Computer Glossary](https://www.tutorialspoint.com/computer_glossary.htm)
* [Who is Who](https://www.tutorialspoint.com/computer_whoiswho.htm)

# Active product sales analysis using matplotlib in Python

[Python](https://www.tutorialspoint.com/articles/category/Python)[Server Side Programming](https://www.tutorialspoint.com/articles/category/Server-Side-Programming)[Programming](https://www.tutorialspoint.com/articles/category/Programming)

ADVERTISING

Matplotlib in Python has various functions like read\_csv,sort\_values,group\_by, etc to perform sales data analysis. Every online business which is involved in product sales of any type uses product sales data analysis to increase their sales and know their customers better. Any company which is involved in any type of e-commerce business uses its sales and customer data to identify trends, patterns, and insights that can be used to improve sales and revenue. The sales data can be used to determine which product has the highest traction, which festive season has the highest demand, and many other trends which can help to increase sales.

Python is a popular programming language for data analysis and visualization. Python provides many libraries and tools that can be used to do product sales analysis effectively. In this article, we will use Matplotlib, which is a popular data visualization library in Python to do active product sales analysis.

We will be using sample sales data for active product sales analysis using numpy, pandas and matplotlib. The sample sales data can be found here.

## **StepWise Sales Data Analysis**

### **Data Reading and Processing**

The sample sales data used in this example for analysis has the following columns −

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Order\_Number | Product\_type | Quantity | Price\_Each | Order\_Date | Address |

* Order Number − Unique number for each placed order.
* Product\_Type − Category of the product
* Quantity − quantity of the product ordered
* Price Each − Price per unit
* Order Date − Date and time when the order was placed
* Address − Address to which the product was delivered.

We will have to import pandas and numpy which can be used to read and process the sample sales data. Here is the code to read the data −

The sample sales data can be found on the Kaggle platform [**here**](https://www.kaggle.com/datasets/kyanyoga/sample-sales-data).

### **Example**

import pandas as pd

import numpy as np

import io

from google.colab import files

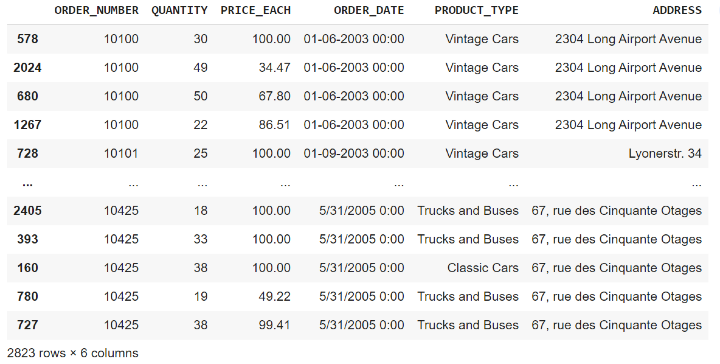
uploaded = files.upload()

# read csv data

Sales\_data = pd.read\_csv(io.BytesIO(uploaded['sample\_sales\_data.csv']), encoding='cp1252')

Sales\_data.sort\_values(by=['ORDER\_NUMBER'])

### **Output**



Once we have read the data, we have to do the processing of the data. The Order Date column needs to be converted to a DateTime object and we can extract month and year from the order date and add a new column for a month, year, and total sales. The code for data cleaning and processing is shown below −

### **Example**

Sales\_data['ORDER\_DATE'] = pd.to\_datetime(Sales\_data['ORDER\_DATE'])

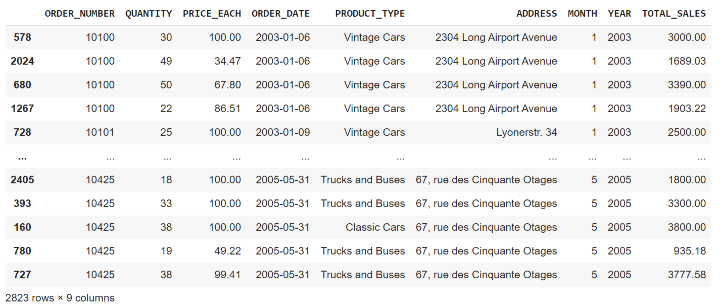
Sales\_data['MONTH'] = Sales\_data['ORDER\_DATE'].dt.month

Sales\_data['YEAR'] = Sales\_data['ORDER\_DATE'].dt.year

Sales\_data['TOTAL\_SALES'] = Sales\_data['QUANTITY'] \* Sales\_data['PRICE\_EACH']

Sales\_data.sort\_values(by=['ORDER\_NUMBER'])

### **Output**



The new column month, year, and toatal\_sales will help us analyze the sales trend over time. Now we can use these columns to plot different plots using the matplotlib library to get some insights from the sample\_sales\_data.

### **Data Visualization**

Till now we have read and processed our data to use it to plot different plots using the matplotlib library in Python. Matplotlib provides line, bar, and scatter plots to visualize the data.

### **Visualization of total sales over time**

To visualize the total sales over time we can plot a line graph using matplotlib.To visualize that we have to −

* Group the data by year and month
* Create a line chart using matplotlib
* Set the title and axis labels
* Display the chart

### **Example**

import matplotlib.pyplot as plt

# Group the data by year and month

sales\_by\_month = Sales\_data.groupby(['YEAR', 'MONTH']).sum()['TOTAL\_SALES'].reset\_index()

# Create a line chart

plt.plot(sales\_by\_month.index, sales\_by\_month.values)

# Set the title and axis labels

plt.title('Total Sales by Month')

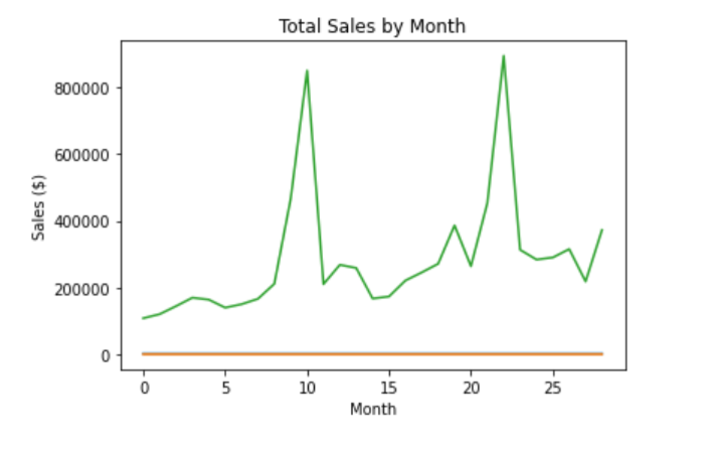
plt.xlabel('Month')

plt.ylabel('Sales ($)')

# Display the chart

plt.show()

### **Output**



### **Visualization of annual revenue over time**

We can visualize the annual revenue for every year and can see which year has the highest revenue and which year has the lowest revenue till now. To do so we have to−

* Group the sales data by year
* Create a bar plot using Seaborn which uses matplotlib underneath
* Set the title and axis labels
* Display the chart

### 

### **Example**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Annual Revenue

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

yearly\_revenue = Sales\_data.groupby(['YEAR'])['TOTAL\_SALES'].sum().reset\_index()

sns.barplot(x="YEAR", y="TOTAL\_SALES", data=yearly\_revenue)

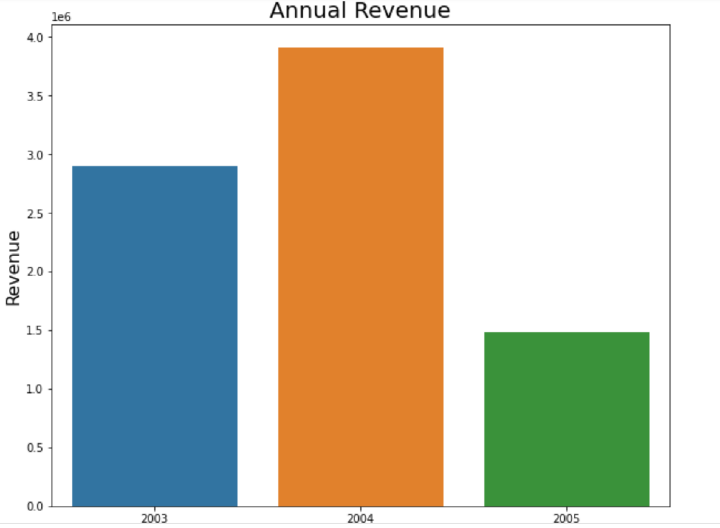
plt.title('Annual Revenue', fontsize = 20)

plt.xlabel('Year', fontsize = 16)

plt.ylabel('Revenue', fontsize = 16)

plt.show()

### **Output**



advantages :

● better profit margins

● flexible products list

● great source of passive income

Disadvantages:

● tight compitive market

● Copy right Of the market

**Conclusion**

Identify The Sales Data To Analyze. To generate significant conclusions that favorably affect yourbottom line, you must analyze the correct type of sales data. Start by setting goals for the areas or products whose revenue growth you

We can analyze and visualize any type of product sales data using matplotlib in Python and get standard data insights that can be used by the company to increase sales. We analyzed the total sales over time and year-wise revenue in the above article using matplotlib, pandas, and numpy in Python.