

**Date:** 10th of February, 2024.

Forggith Banner.jpg

#### **Team Members**

- Zeenat Oyetolu: <https://linkedin.com/in/zeenatoyetolu>
- Bolanle Olaleye: <https://linkedin.com/in/bolanle-olaleye>
- Precious Olonisaye: <https://linkedin.com/in/olonisaye>

**Data Source:** Foresight BI Internship Dataset

## Table of Content

- Introduction
- Objective
- Dataset Description
- Questions answered in this analysis
- Insights derived from this analysis
- Observations
- Exploratory Data Analysis
- Recommendations
- Conclusion

## Introduction

Forggith Pharmaceuticals is a renowned name in the pharmaceutical industry, known for its commitment to excellence, innovation, and patient care. This project aims to analyze sales data and gain insights into the performance of Forggith Pharmaceuticals, a leading pharmaceutical company specializing in the development and distribution of innovative healthcare products. By exploring various aspects of the sales data, we seek to identify trends, patterns, and factors influencing sales performance, ultimately providing valuable insights to support strategic decision-making within the organization.

## Objective

To create a report that will assist in guiding the strategies, tactics, and operations of Forggith Pharmaceuticals.

## Dataset Description

The dataset provided covers sales transactions for the company across various cities over multiple years. It includes details on 18 key attributes for each sales record:

- **Sales ID:** Unique identifier for each sales transaction.
- **Month Year:** Date of the sales transaction, represented as month and year.
- **Sales Rep ID:** Unique identifier for the sales representative responsible for the transaction.
- **Sales\_Rep\_Name:** Name of the sales representative.
- **Manager:** Manager overseeing the sales representative.
- **Team:** Sales team to which the representative belongs.
- **Distributor:** Distributor associated with the transaction.
- **Customer Name:** Name of the customer purchasing the product.
- **Location ID:** Unique identifier for the location of the customer.
- **Subchannel ID:** Unique identifier for the subchannel associated with the transaction.
- **Subchannel Name:** Name of the subchannel.
- **Facility Type:** Type of facility where the transaction occurred.
- **Product ID:** Unique identifier for the product sold.
- **Product Name:** Name of the product.
- **Product Class:** Class of the product.
- **Product Price:** Price of the product.
- **Quantity:** Quantity of the product sold
- **City Name:** Name of the city where the transaction occurred.

## QUESTIONS ANSWERED IN THIS ANALYSIS

1. What is the Total Revenue from 2022 to 2025?
2. What is the Total Target from 2022 to 2025?
3. What is the Percentage of Revenue Achieved to Target?
4. What is the Revenue Volume Achieved?
5. What is the Target Volume?
6. What is the Percentage of Revenue Volume Achieved to Target Volume?
7. What is the Actual Revenue by Sales Representative?
8. What is the Target Amount of Sales Representatives?
9. What is the Actual Volume of Sales Representatives?

10. What is the Target Volume of the Sales Representatives?
11. What is the Actual Revenue Achievement by Sales Team?
12. What is the Percentage Contribution of each Team to the Total Revenue?
13. What is the Revenue Achieved by Product Class?
14. What is the Percentage Contribution of the Product Class?
15. What is the Volume Achievement by Product Class?
16. What is the Revenue Trend from 2022 to 2025?
17. What is the Yearly Revenue Achieved by the Sales Team?
18. What is the Yearly Revenue Achieved by Product Class?
19. What is the Yearly Distribution of Revenue by Channel?
20. What is the Total Revenue Year To Date (January to December 2025)?
21. What is the Total Revenue Same Period Last Year (January 2024 to December 2024)?
22. What is the Total Revenue Previous Year To Date from January 2024 to December 2025?

## INSIGHTS DERIVED FROM THIS ANALYSIS

1. The Target for 2022 to 2025 was reached and surpassed by 33%.
2. The Revenue volume achieved surpassed the Target volume by 35%.
3. The top contributors to the revenue are Thompson Crawford, Daniel Gates, and Jimmy Grey, who excel as sales representatives.
4. The Top contributor to the Volume and Revenue is Thompson Crawford, he generated \$950M and sold 2.36M products. His Target volume was 1.5M and his Target Amount was \$645M.
5. Anne Wu had the highest target of \$777M but he surpassed his target by \$18M.
6. Team Delta makes the greatest contribution to sales, accounting for 30% of the Total Revenue.
7. The antiseptic category of products accounts for the highest proportion of revenue.
8. Revenue increased from 2022 to 2024, but drastically decline in 2025.
9. Throughout the period spanning 2022 to 2025, Retail Pharmacy and Government Hospital maintained their status as the Leading Revenue Contributors.

## Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import seaborn as sns

import warnings
warnings.filterwarnings("ignore", message="use_inf_as_na option is deprecated", category=FutureWarning)

Forggith_colors = ["#6C1D45", "#00529F", "#1D4D2E", "#FF8C00",
"#D1D1D1", "#4A4A4A", "#72B7D2", "#8ABD3B"]
```

**Burgundy #6C1D45**

**Dark Blue #00529F**

**Dark Green #1D4D2E**

**Orange #FF8C00E**

**Light Gray #D1D1D1**

**Dark Gray #4A4A4A**

**Light Blue #72B7D2**

**Light Green #8ABD3B**

## Reading the dataset:

```
# Imported Sales Dataset
Forggith_Sales =pd.read_csv("Forggith_Sales.csv", index_col=0,
encoding="ISO-8859-1")
#Forggith_Sales =pd.read_csv("/kaggle/input/d/zeeenterprise/forggith-
dataset/Forggith_Sales.csv", index_col=0, encoding="ISO-8859-1")

# Imported Target Data
Forggith_Targets = pd.read_csv("Forggith_Target.csv", index_col=0)
#Forggith_Targets =
pd.read_csv("/kaggle/input/d/zeeenterprise/forggith-dataset/Forggith_T
arget.csv", index_col=0)
```

## Data Exploration

```
# Setting display options to avoid scientific notation
pd.set_option('display.float_format', lambda x: '%.1f' % x)
```

**Checked the first few column**

```
Forggith_Sales.head()
```

S/N	Sales ID	Date	Sales_Rep_Name	Manager	Team \
0	20090	1/1/2022	Sheila Stones	Britanny Bold	Delta
1	20091	1/1/2022	Stella Given	Alisha Cordwell	Charlie
2	20092	1/1/2022	Daniel Gates	Alisha Cordwell	Charlie
3	20093	1/1/2022	Mary Gerrard	Britanny Bold	Delta
4	20094	1/1/2022	Mary Gerrard	Britanny Bold	Delta

S/N	Distributor	Customer_Name \
0	Gerlach LLC	Fadel-West Pharmacy
1	Gerlach LLC	Nader-Gaylord Pharmacy
2	Gerlach LLC	McKenzie-Zemlak Pharm
3	Gerlach LLC	Fritsch-Hauck Pharmaceutical Ltd
4	Gerlach LLC	Bernier, Murphy and Rau Pharma Plc

Facility_Type \	Location ID	SubChannelID	Sub_Channel_Name
S/N			
0	Ditzingen-48.8264-9.0667	1	Government Hospital
1	Backnang-48.9464-9.4306	2	Private Hospital
2	Rheinbach-50.6256-6.9491	3	Institution Pharmacy
3	Fürth-49.4783-10.9903	4	Retail Pharmacy
4	Geldern-51.5197-6.3325	1	Government Hospital

\	ProductID	Product_Name	Product_Class	Product_Price
S/N				
0	Exo-Moo	Exotropin Empizine	Mood Stabilizers	785
1	Rob-Ant	Robapril	Antipiretics	453
2	Sec-Ant	Secrelazine Insonamic	Antipiretics	694
3	Meg-Ant	Megenorphine	Antimalarial	402
4	Aga-Moo	Agalsiline	Mood Stabilizers	64

S/N	Quantity	City	Revenue
0	20	Ditzingen	15700
1	10	Backnang	4530

2	25	Rheinbach	17350
3	5	Fürth	2010
4	20	Geldern	1280

```
Forggith_Targets.head()
```

Date \ ProductID	SalesRepID	Sales_Rep_Name	Month	Month_number	Year
Aba-Ant 1/1/2022	SN20038	Morris Garcia	Jan	1	2022
Aba-Ant 1/1/2023	SN20038	Morris Garcia	Jan	1	2023
Aba-Ant 1/1/2024	SN20038	Morris Garcia	Jan	1	2024
Aba-Ant 1/1/2025	SN20038	Morris Garcia	Jan	1	2025
Aba-Ant 2/1/2022	SN20038	Morris Garcia	Feb	2	2022

ProductID	Target_Quantity	Product_Price	Target_Amount
Aba-Ant	168	742	124656
Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208
Aba-Ant	168	742	124656

### Checked the column names

```
Forggith_Sales.columns
```

```
Index(['Sales ID', 'Date', 'Sales_Rep_Name', 'Manager', 'Team',  
      'Distributor',  
      'Customer_Name', 'Location ID', 'SubChannelID',  
      'Sub_Channel_Name',  
      'Facility_Type', 'ProductID', 'Product_Name', 'Product_Class',  
      'Product_Price', 'Quantity', 'City', 'Revenue'],  
      dtype='object')
```

```
Forggith_Targets.columns
```

```
Index(['SalesRepID', 'Sales_Rep_Name', 'Month', 'Month_number',  
      'Year', 'Date',  
      'Target_Quantity', 'Product_Price', 'Target_Amount'],  
      dtype='object')
```

### Determined the number of rows and columns

```
Forggith_Sales.shape
```

```
(213598, 18)
```

```
Forggith_Targets.shape
```

```
(149760, 9)
```

### Determined the non null count and data types

```
Forggith_Sales.info()
```

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

```
Index: 213598 entries, 0 to 157111
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	Sales ID	213598 non-null	int64
1	Date	213598 non-null	object
2	Sales_Rep_Name	213598 non-null	object
3	Manager	213598 non-null	object
4	Team	213598 non-null	object
5	Distributor	213598 non-null	object
6	Customer_Name	213598 non-null	object
7	Location ID	213598 non-null	object
8	SubChannelID	213598 non-null	int64
9	Sub_Channel_Name	213598 non-null	object
10	Facility_Type	213598 non-null	object
11	ProductID	213598 non-null	object
12	Product_Name	213598 non-null	object
13	Product_Class	213598 non-null	object
14	Product_Price	213598 non-null	int64
15	Quantity	213598 non-null	int64
16	City	213598 non-null	object
17	Revenue	213598 non-null	int64

```
dtypes: int64(5), object(13)
```

```
memory usage: 31.0+ MB
```

```
Forggith_Targets.info()
```

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

```
Index: 149760 entries, Aba-Ant to Zyv-Ana
```

```
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	SalesRepID	149760 non-null	object
1	Sales_Rep_Name	149760 non-null	object
2	Month	149760 non-null	object
3	Month_number	149760 non-null	int64
4	Year	149760 non-null	int64
5	Date	149760 non-null	object

```
6   Target_Quantity  149760 non-null  int64
7   Product_Price    149760 non-null  int64
8   Target_Amount    149760 non-null  int64
dtypes: int64(5), object(4)
memory usage: 11.4+ MB
```

### Checked for missing values

```
Forggith_Sales.isnull().sum()
```

```
Sales ID          0
Date              0
Sales_Rep_Name    0
Manager           0
Team              0
Distributor       0
Customer_Name     0
Location_ID       0
SubChannelID      0
Sub_Channel_Name  0
Facility_Type     0
ProductID         0
Product_Name      0
Product_Class     0
Product_Price     0
Quantity          0
City              0
Revenue           0
dtype: int64
```

```
Forggith_Targets.isnull().sum()
```

```
SalesRepID        0
Sales_Rep_Name    0
Month             0
Month_number      0
Year              0
Date              0
Target_Quantity   0
Product_Price     0
Target_Amount     0
dtype: int64
```

```
Forggith_Sales.isna().sum()
```

```
Sales ID          0
Date              0
Sales_Rep_Name    0
Manager           0
Team              0
```



```

Distributor      0
Customer_Name    0
Location_ID      0
SubChannelID     0
Sub_Channel_Name 0
Facility_Type    0
ProductID        0
Product_Name     0
Product_Class    0
Product_Price    0
Quantity         0
City             0
Revenue          0
dtype: int64

Forggith_Targets.isna().sum()

SalesRepID      0
Sales_Rep_Name   0
Month           0
Month_number     0
Year            0
Date            0
Target_Quantity  0
Product_Price    0
Target_Amount    0
dtype: int64

```

- **Outcome:** There are no missing, or null values in the dataset

#### Determined if there were Duplicated rows

```

Forggith_Sales.duplicated().sum()

0

Forggith_Targets.duplicated().sum()

192

```

- **Outcome** There are no duplicates in the dataset.

#### Determined the summary statistics

```

# Summary statistics of the Sales data
Forggith_Sales.describe()

```

	Sales ID	SubChannelID	Product_Price	Quantity	Revenue
count	213598.0	213598.0	213598.0	213598.0	213598.0
mean	126888.5	2.5	412.4	129.8	52920.1
std	61660.6	1.2	225.0	807.5	336729.4
min	20090.0	1.0	22.0	1.0	22.0

25%	73489.2	1.0	195.0	8.0	2105.0
50%	126888.5	3.0	435.0	20.0	6880.0
75%	180287.8	4.0	605.0	69.0	25120.0
max	233687.0	4.0	794.0	117600.0	39250000.0

**Note:** 75% of the Quantities sold fall below 69

*# Summary statistics of Targets data*

Forggith\_Targets.describe()

	Month_number	Year	Target_Quantity	Product_Price
Target_Amount				
count	149760.0	149760.0	149760.0	149760.0
mean	6.5	2023.5	136.9	412.6
std	3.5	1.1	185.6	224.9
min	1.0	2022.0	3.0	22.0
25%	3.8	2022.8	57.0	199.5
50%	6.5	2023.5	91.0	432.5
75%	9.2	2024.2	151.0	606.2
max	12.0	2025.0	4928.0	794.0

#### Noteworthy:

- The average product price is \$412.
- The minimum prize is \$22.
- The maximum prize is \$794.

#### Visualized the distribution of the Numeric Fields

*# Distribution of Product Price field*

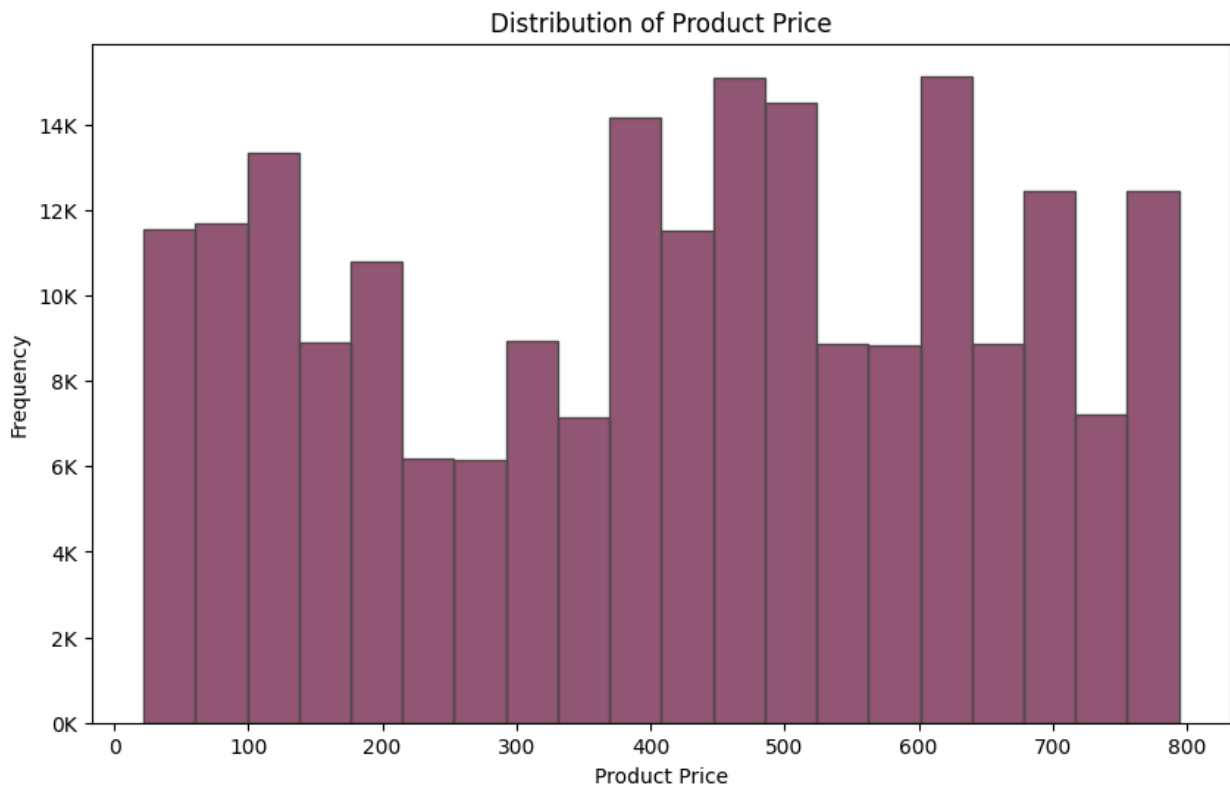
```
plt.figure(figsize=(10, 6))
sns.histplot(Forggith_Sales['Product_Price'], bins=20, color =
'#6C1D45', edgecolor='#4A4A4A')
```

*# Define formatter function to display y-axis ticks in thousands with 'K' suffix*

```
def format_thousands(x, pos):
    return '{:.0f}K'.format(x / 1000)
```

```
# Apply the formatter to the y-axis
plt.gca().yaxis.set_major_formatter(FuncFormatter(format_thousands))

plt.xlabel('Product Price')
plt.ylabel('Frequency')
plt.title('Distribution of Product Price')
plt.show()
```



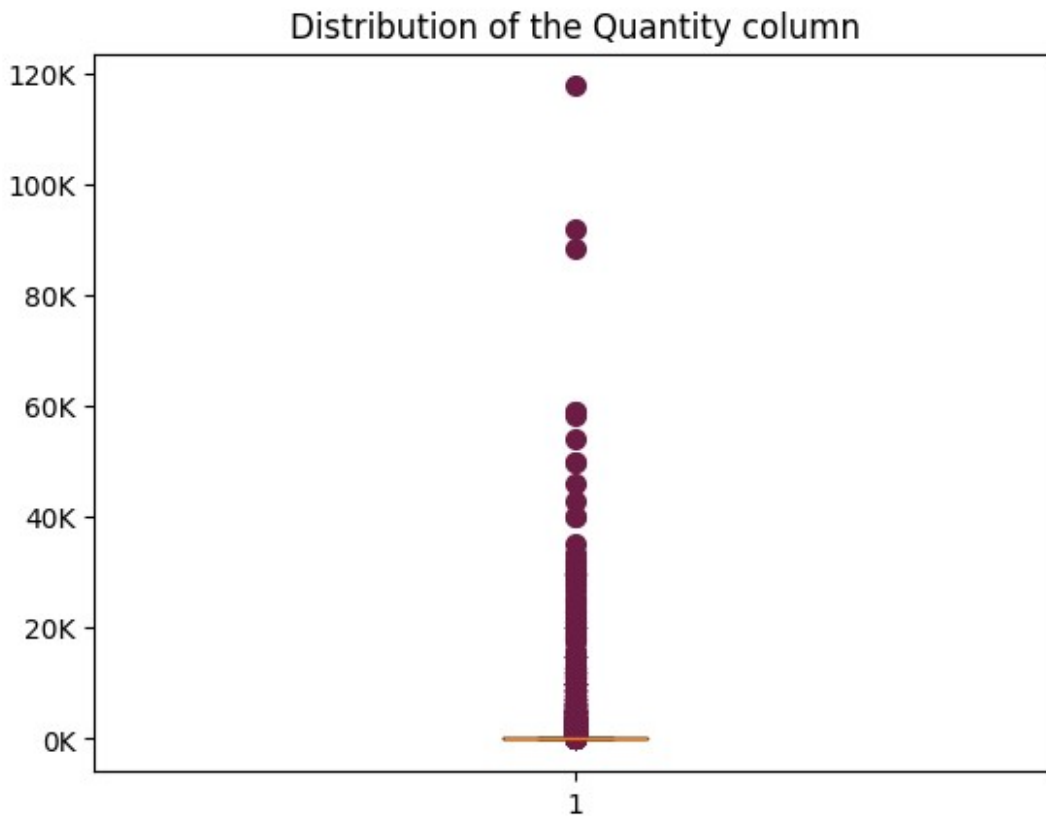
```
Forggith_Sales['Quantity'].unique()
array([ 20, 10, 25, ..., 732, 1821, 919], dtype=int64)
```

**Visualized the relationship between the Quantity and Product Price**

```
# Distribution of the Quantity column
plt.boxplot(Forggith_Sales["Quantity"], flierprops=dict(marker='o',
markerfacecolor='#6C1D45', markersize=8, markeredgecolor='none'))
plt.title('Distribution of the Quantity column')

# Define formatter function to display y-axis ticks in thousands with
'K' suffix
def format_thousands(x, pos):
    return '{:.0f}K'.format(x / 1000)
```

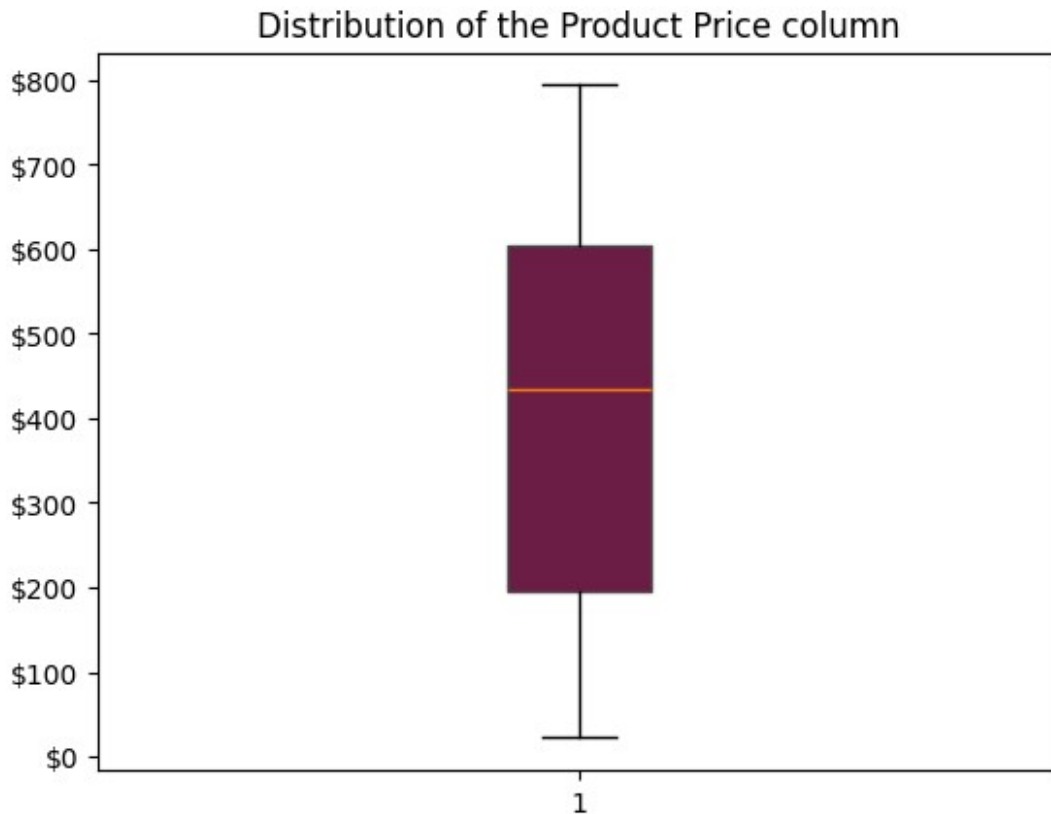
```
# Apply the formatter to the y-axis
plt.gca().yaxis.set_major_formatter(FuncFormatter(format_thousands))
```



```
# Distribution of the Product Price Column
plt.boxplot(Forggith_Sales["Product_Price"],
boxprops=dict(color="#4A4A4A", facecolor="#6C1D45"),
patch_artist=True)
plt.title('Distribution of the Product Price column')

# Define a function to format y-axis ticks with dollar notation
def dollar_formatter(x, pos):
    return '${:.0f}'.format(x)

# Create a FuncFormatter object using the custom formatting function
formatter = FuncFormatter(dollar_formatter)
plt.gca().yaxis.set_major_formatter(formatter)
```



**Note:** There are no outliers in the product price distribution. The dataset is normally distributed.

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Product_Price', y='Quantity', data=Forggith_Sales,
color='#6C1D45', edgecolor='#4A4A4A', label='Quantity')

# Define formatter function to display y-axis ticks in thousands with
'K' suffix
def format_thousands(x, pos):
    return '{:.0f}K'.format(x / 1000)

# Apply the formatter to the y-axis
plt.gca().yaxis.set_major_formatter(FuncFormatter(format_thousands))

plt.xlabel('Product Price')
plt.ylabel('Quantity')
plt.title('Product Price vs Quantity')
plt.legend()
plt.show()
```

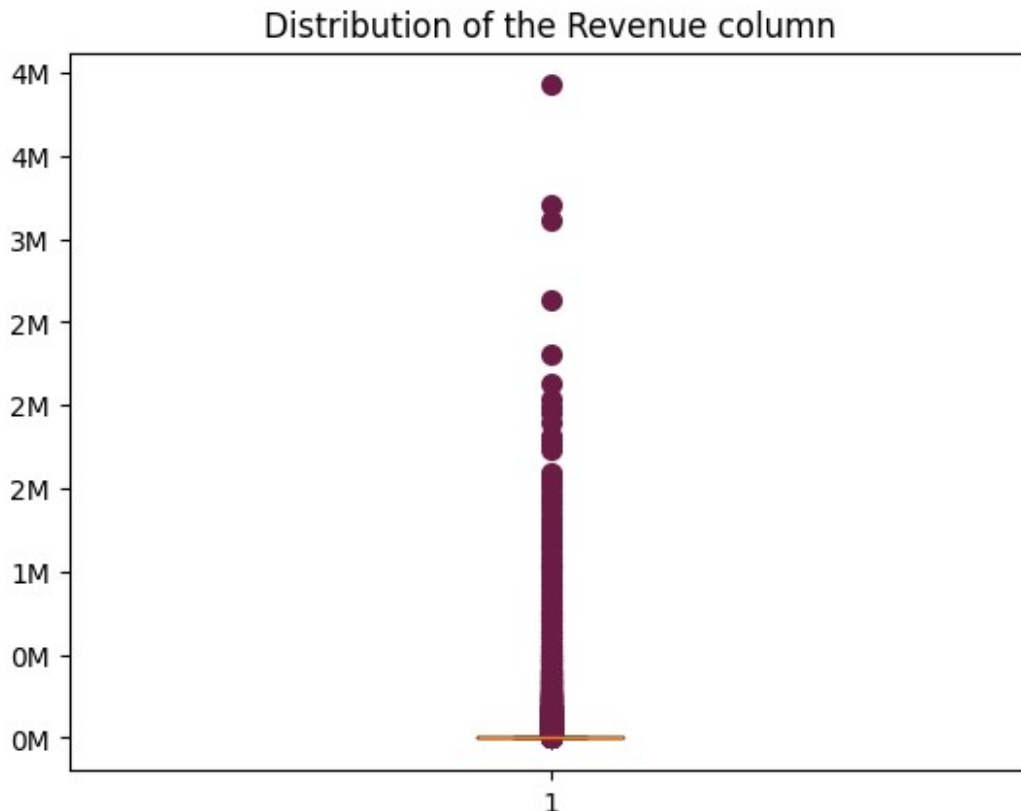


```
# Distribution of the Revenue Column
plt.boxplot(Forggith_Sales["Revenue"], flierprops=dict(marker='o',
markerfacecolor='#6C1D45', markersize=8, markeredgecolor='none'))
plt.title('Distribution of the Revenue column')

# Define a function to format y-axis ticks in billions
def millions_formatter(x, pos):
    return '{:.0f}M'.format(x * 1e-7)

# Create a FuncFormatter object using the custom formatting function
formatter = FuncFormatter(millions_formatter)
plt.gca().yaxis.set_major_formatter(formatter)

plt.gca().yaxis.set_major_formatter(formatter)
```



## EXPLORATORY DATA ANALYSIS

### 1. What is the Total Revenue from 2022 to 2025?

```
# Total Revenue for the yaer 2022 t0 2025?
Total_Revenue = int(Forggith_Sales['Revenue'].sum())
print('Total Revenue Achieved is ${:,}.'.format(Total_Revenue))

Total Revenue Achieved is $11,303,631,675.
```

### 2. What is the Total Target from 2022 to 2025?

```
# Total Target for the yaer 2022 t0 2025?
Total_Target = int(Forggith_Targets['Target_Amount'].sum())

print('Total Target is ${:,}.'.format(Total_Target))

Total Target is $8,453,326,524.
```

### 3. What is the Percentage of Revenue Achieved to Target?

```
Revenue_Target = int(Total_Revenue*100/Total_Target)
print('The Percentage of Revenue Achieved to Target is {:,}
%'.format(Revenue_Target))
```

The Percentage of Revenue Achieved to Target is 133%

**Insight:** The Target for 2022 to 2025 was reached and surpassed by 33%

#### 4. What is the Revenue Volume Achieved?

```
# Total quantity of products sold
Revenue_Volume_achieved = int(Forggith_Sales["Quantity"].sum())
print('The Revenue Volume Achieved is
{:,.}.'.format(Revenue_Volume_achieved))
```

The Revenue Volume Achieved is 27,720,212.

#### 5. What is the Target Volume?

```
# The Total Target Quantity
Target_Volume = Forgith_Targets["Target_Quantity"].sum()
print('The Target Volume is {:,.}.'.format(Target_Volume))
```

The Target Volume is 20,505,144.

#### 6. What is the Percentage of Revenue Volume Achieved to Target Volume?

```
RevenueVolume_TargetVolume =
int(Revenue_Volume_achieved*100/Target_Volume)
print('The Percentage of Revenue Volume Achieved to Target Volume is
{:,.}%.'.format(RevenueVolume_TargetVolume))
```

The Percentage of Revenue Volume Achieved to Target Volume is 135%.

**Insight:** The Revenue volume achieved surpassed Target volume by 35%.

#### 7. What is the Actual Revenue by Sales Representatives?

```
# Define a custom formatter function
def millions_formatter(x, pos):
    return f'{x / 1e6:.0f}M'

# Aggregate the data by sales representative
sales_rep_revenue = Forgith_Sales.groupby('Sales_Rep_Name')
['Revenue'].sum().reset_index()

# Sort the aggregated data by revenue in descending order
sales_rep_revenue_sorted = sales_rep_revenue.sort_values(by='Revenue',
ascending=False)

# Create the plot
plt.figure(figsize=(15,5))
bars = plt.bar(sales_rep_revenue_sorted['Sales_Rep_Name'],
sales_rep_revenue_sorted['Revenue'])
plt.title("The Top Contributors to the Revenue are Thompson Crawford,
```



```

Daniel Gates, and Jimmy Grey")
plt.xticks(rotation=85)

# Apply the custom formatter to the y-axis
formatter = FuncFormatter(millions_formatter)
plt.gca().yaxis.set_major_formatter(formatter)

# Add data labels to the first three bars
for i in range(13):
    plt.text(bars[i].get_x() + bars[i].get_width() / 2,
             bars[i].get_height(),
             f'{sales_rep_revenue_sorted["Revenue"].iloc[i] /
             1e6:.0f}M',
             ha='center', va='bottom')

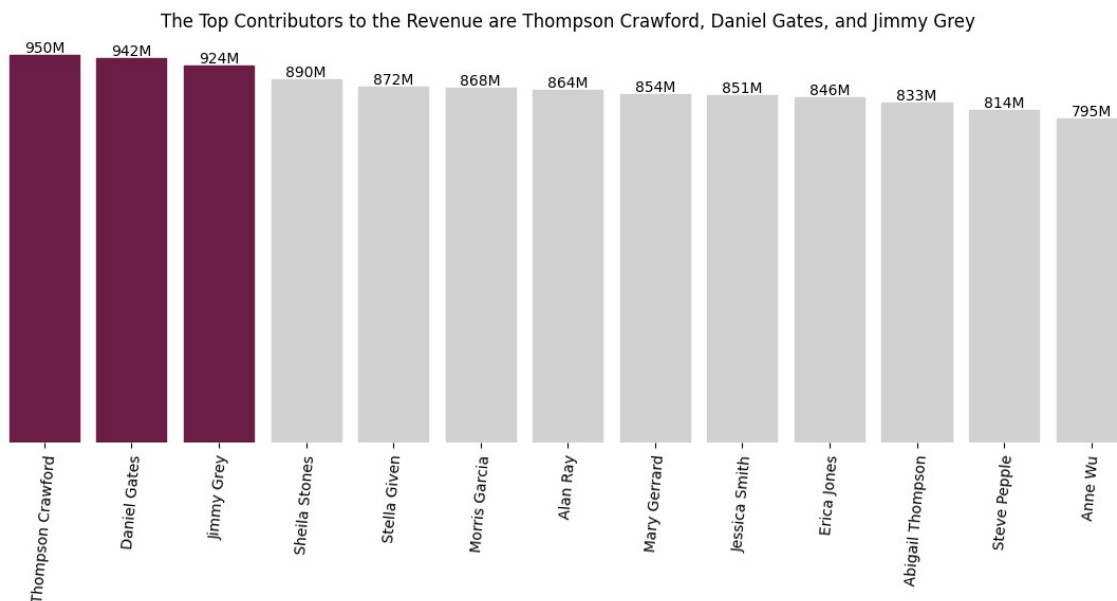
# Set colors for bars
colors = ['#6C1D45' if i < 3 else '#D1D1D1' for i in
range(len(sales_rep_revenue_sorted))]
for bar, color in zip(bars, colors):
    bar.set_color(color)

# Hide the y-axis label
plt.tick_params(axis='y', which='both', left=False, labelleft=False)

# Remove the borders around the chart
plt.box(False)

plt.show()

```



**Insights:** The Top Contributors to the Revenue are Thompson Crawford, Daniel Gates, and Jimmy Grey, who excel as Sales Representatives.

## 8. What is the Target Amount of Sales Representatives?

```
Forggith_Targets.head(5)
```

Date \ ProductID	SalesRepID	Sales_Rep_Name	Month	Month_number	Year
Aba-Ant 1/1/2022	SN20038	Morris Garcia	Jan	1	2022
Aba-Ant 1/1/2023	SN20038	Morris Garcia	Jan	1	2023
Aba-Ant 1/1/2024	SN20038	Morris Garcia	Jan	1	2024
Aba-Ant 1/1/2025	SN20038	Morris Garcia	Jan	1	2025
Aba-Ant 2/1/2022	SN20038	Morris Garcia	Feb	2	2022

ProductID	Target_Quantity	Product_Price	Target_Amount
Aba-Ant	168	742	124656
Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208
Aba-Ant	168	742	124656

```
# Define a custom formatter function
```

```
def millions_formatter(x, pos):  
    return f'{x / 1e6:.0f}M'
```

```
# Aggregate the data by sales representative
```

```
sales_rep_targets = Forggith_Targets.groupby('Sales_Rep_Name')  
['Target_Amount'].sum().reset_index()
```

```
# Sort the aggregated data by targets in descending order
```

```
sales_rep_targets_sorted =  
sales_rep_targets.sort_values(by='Target_Amount', ascending=False)
```

```
# Create the plot
```

```
plt.figure(figsize=(15,5))  
bars = plt.bar(sales_rep_targets_sorted['Sales_Rep_Name'],  
sales_rep_targets_sorted['Target_Amount'])  
plt.title("Anne Wu, Jessica Smith, Abigail Thompsons have the highest  
Targets Amount")  
plt.xticks(rotation=85)
```

```
# Apply the custom formatter to the y-axis
```

```
formatter = FuncFormatter(millions_formatter)  
plt.gca().yaxis.set_major_formatter(formatter)
```

```

# Add data labels to the first three bars
for i in range(13):
    plt.text(bars[i].get_x() + bars[i].get_width() / 2,
             bars[i].get_height(),
             f'{sales_rep_targets_sorted["Target_Amount"].iloc[i] /
             1e6:.0f}M',
             ha='center', va='bottom')

# Set colors for bars
colors = ['#6C1D45' if i < 3 else '#D1D1D1' for i in
range(len(sales_rep_targets_sorted))]
for bar, color in zip(bars, colors):
    bar.set_color(color)

# Hide the y-axis label
plt.tick_params(axis='y', which='both', left=False, labelleft=False)

# Remove the borders around the chart
plt.box(False)

plt.show()

```



**Insights:** Anne Wu, Jessica Smith, Abigail Thompsons have the highest Targets amount.

## 9. What is the Actual Volume by Sales Representatives?

```

#Define a custom formatter function
def millions_formatter(x, pos):
    return f'{x / 1e6:.0f}M'

#Aggregate the data by sales representative

```

```

sales_rep_quantity = Forggith_Sales.groupby('Sales_Rep_Name')
['Quantity'].sum().reset_index()

#Sort the aggregated data by Quantity in descending order
sales_rep_quantity_sorted =
sales_rep_quantity.sort_values(by='Quantity', ascending=False)

# Create a plot
plt.figure(figsize = (20,5))
bars = plt.bar(sales_rep_quantity_sorted['Sales_Rep_Name'],
sales_rep_quantity_sorted['Quantity'])
plt.title("Leading in sales volume are Thompson Crawford, Jimmy Grey,
and Sheilla Stones")
plt.xticks(rotation=85)

# Apply the custom formatter to the y-axis
formatter = FuncFormatter(millions_formatter)
plt.gca().yaxis.set_major_formatter(formatter)

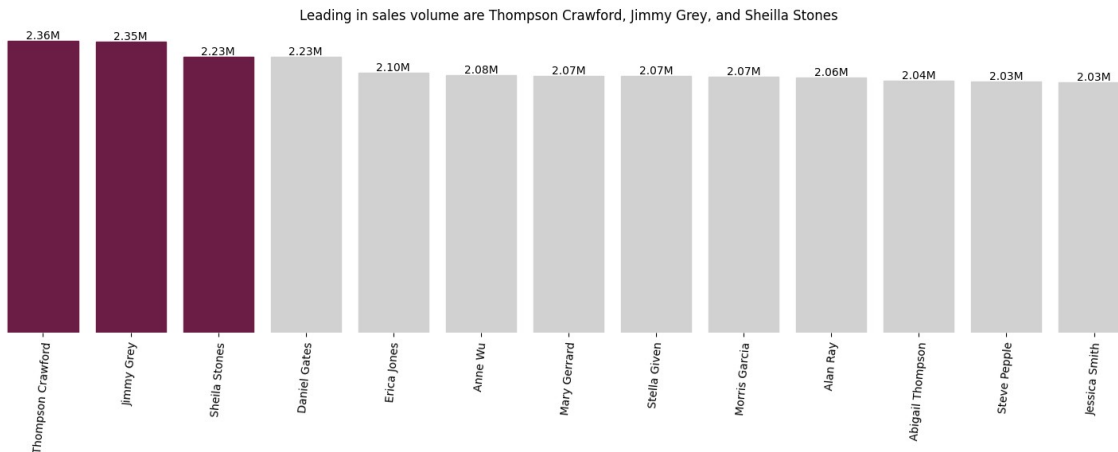
# Add data labels to the first two bars
for i in range(13):
    plt.text(bars[i].get_x()+bars[i].get_width()/2,
bars[i].get_height(),
            f'{sales_rep_quantity_sorted["Quantity"].iloc[i] /
1e6:.2f}M',
            ha='center', va='bottom')

# Set colors for bars
colors = ['#6C1D45' if i < 3 else '#D1D1D1' for i in
range(len(sales_rep_revenue_sorted))]
for bar, color in zip(bars, colors):
    bar.set_color(color)

# Hide the y-axis label
plt.tick_params(axis='y', which='both', left=False, labelleft=False)
# Remove the borders around the chart
plt.box(False)

plt.show()

```



**Insight:** Thompson Crawford, Jimmy Grey, and Sheilla Stones recorded the highest sales volume.

## 10. What is the Target Volume of Sales Representatives?

```
#Define a custom formatter function
def millions_formatter(x, pos):
    return f'{x / 1e6:.0f}M'

#Aggregate the data by sales representative
sales_rep_target_quantity = Forggith_Targets.groupby('Sales_Rep_Name')
['Target_Quantity'].sum().reset_index()

#Sort the aggregated data by Target Quantity in descending order
sales_rep_target_quantity_sorted =
sales_rep_target_quantity.sort_values(by='Target_Quantity',
ascending=False)

# Create a plot
plt.figure(figsize = (20,5))
bars = plt.bar(sales_rep_target_quantity_sorted['Sales_Rep_Name'],
sales_rep_target_quantity_sorted['Target_Quantity'])
plt.title("Anne Wu, Abigail Thomson and Jessica Smith has the highest
target volume. ")
plt.xticks(rotation=85)

# Apply the custom formatter to the y-axis
formatter = FuncFormatter(millions_formatter)
plt.gca().yaxis.set_major_formatter(formatter)

# Add data labels to the first two bars
for i in range(13):
    plt.text(bars[i].get_x()+bars[i].get_width()/2,
bars[i].get_height(),

f'{sales_rep_target_quantity_sorted["Target_Quantity"].iloc[i] /
1e6:.2f}M',
```

```

        ha='center', va='bottom')

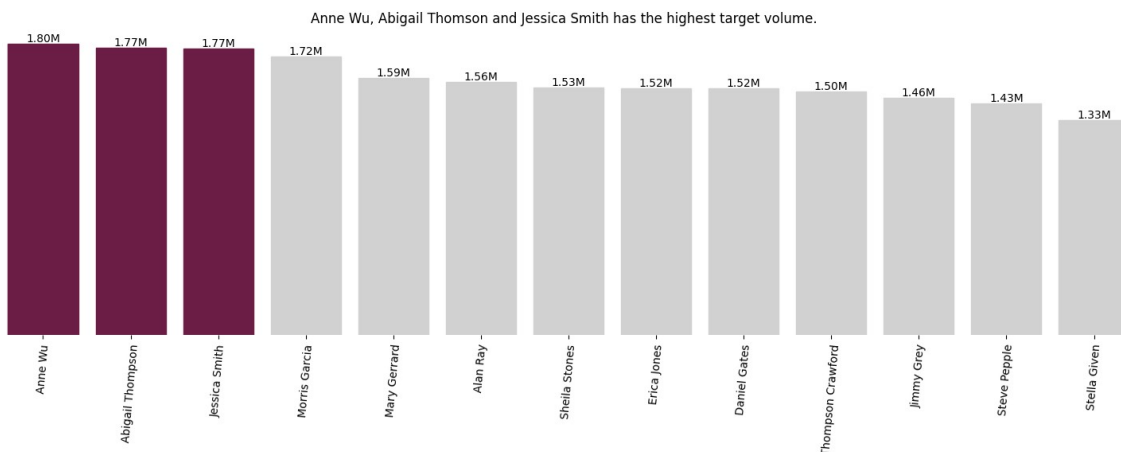
# Set colors for bars
colors = ['#6C1D45' if i < 3 else '#D1D1D1' for i in
range(len(sales_rep_target_quantity_sorted))]
for bar, color in zip(bars, colors):
    bar.set_color(color)

# Hide the y-axis label
plt.tick_params(axis='y', which='both', left=False, labelleft=False)

# Remove the borders around the chart
plt.box(False)

plt.show()

```



**Insight:** Anne Wu, Abigail Thomson and Jessica Smith has the highest target volume.

## 11. What is the Actual Revenue Achievement by Sales Team?

```

# Aggregate the data by Team and sum the revenue
sales_team = Forggith_Sales.groupby('Team')
['Revenue'].sum().reset_index()

# Sort the aggregated data by Revenue in descending order
sales_team_sorted = sales_team.sort_values(by='Revenue',
ascending=False)

# Define colors
main_color = '#6C1D45'
other_color = 'gray'
last_color = 'lightgray'

# Create custom color palette
custom_palette = [main_color] + [other_color] + [last_color] * 4

```

```

# Create count plot with sorted data
sns.barplot(y=sales_team_sorted['Team'],
x=sales_team_sorted['Revenue'], palette=custom_palette)

plt.title('Team Delta Generates the Greatest Revenue')

# Remove the border around the plot
sns.despine(bottom=True, left=True)

# Format ticks in billions
formatter = FuncFormatter(lambda x, _: f'${x/1e9:.2f}B')
plt.gca().xaxis.set_major_formatter(formatter)

# Add data labels to the bars
for index, value in enumerate(sales_team_sorted['Revenue']):
    plt.text(value, index, f' ${value/1e9:.2f}B', ha='left',
va='center')

# Remove the x-axis
plt.tick_params(axis='x', which='both', bottom=False, top=False,
labelbottom=False)

plt.show()

```

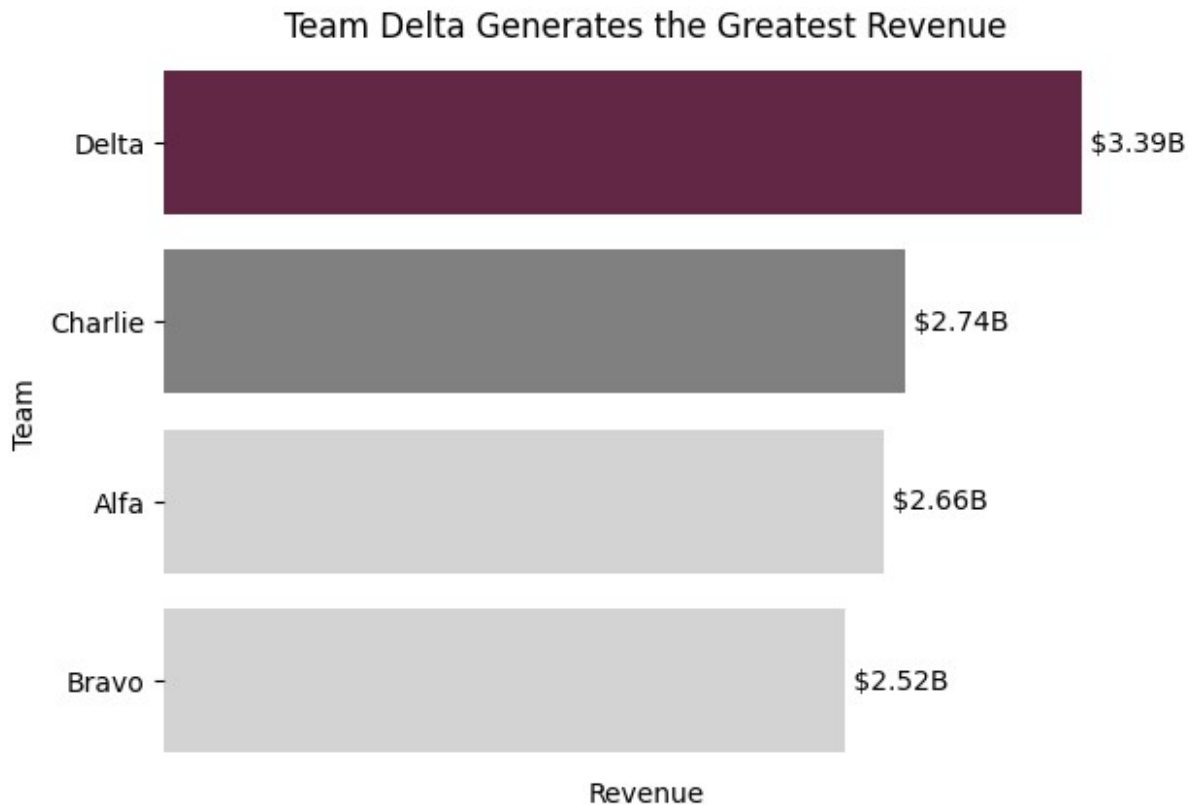
C:\Users\Administrator\AppData\Local\Temp\ipykernel\_13852\3166905123.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```

sns.barplot(y=sales_team_sorted['Team'],
x=sales_team_sorted['Revenue'], palette=custom_palette)
C:\Users\Administrator\AppData\Local\Temp\ipykernel_13852\3166905123.py:16: UserWarning: The palette list has
more values (6) than needed (4), which may not be intended.
sns.barplot(y=sales_team_sorted['Team'],
x=sales_team_sorted['Revenue'], palette=custom_palette)

```



#### 12. What is the Percentage Contribution of each Sales Team to the Total Revenue?

```
# Aggregate the data by Team and sum the revenue
sales_team = Forggith_Sales.groupby('Team')
['Revenue'].sum().reset_index()

# Calculate the total revenue
total_revenue = sales_team['Revenue'].sum()

# Calculate the percentage contribution of each team
sales_team['Percentage'] = sales_team['Revenue'] / total_revenue * 100

# Sort the aggregated data by Revenue in descending order
sales_team_sorted = sales_team.sort_values(by='Revenue',
ascending=False)

# Define colors
main_color = '#6C1D45'
other_color = 'gray'
last_color = 'lightgray'

# Create custom color palette
custom_palette = [main_color] + [other_color] + [last_color] * 4

# Create count plot with sorted data
```



```

ax = sns.barplot(y=sales_team_sorted['Team'],
x=sales_team_sorted['Revenue'], palette=custom_palette)

plt.title('Team Delta contributes the Largest Share')

# Remove the border around the plot
sns.despine(bottom=True, left=True)

# Format ticks in billions
formatter = FuncFormatter(lambda x, _: f'${x/1e9:.0f}B')
plt.gca().xaxis.set_major_formatter(formatter)

# Add data labels to the bars with percentage difference
for index, (value, percentage) in
enumerate(zip(sales_team_sorted['Revenue'],
sales_team_sorted['Percentage'])):
    plt.text(value, index, f' {percentage:.0f}%', ha='left',
va='center')

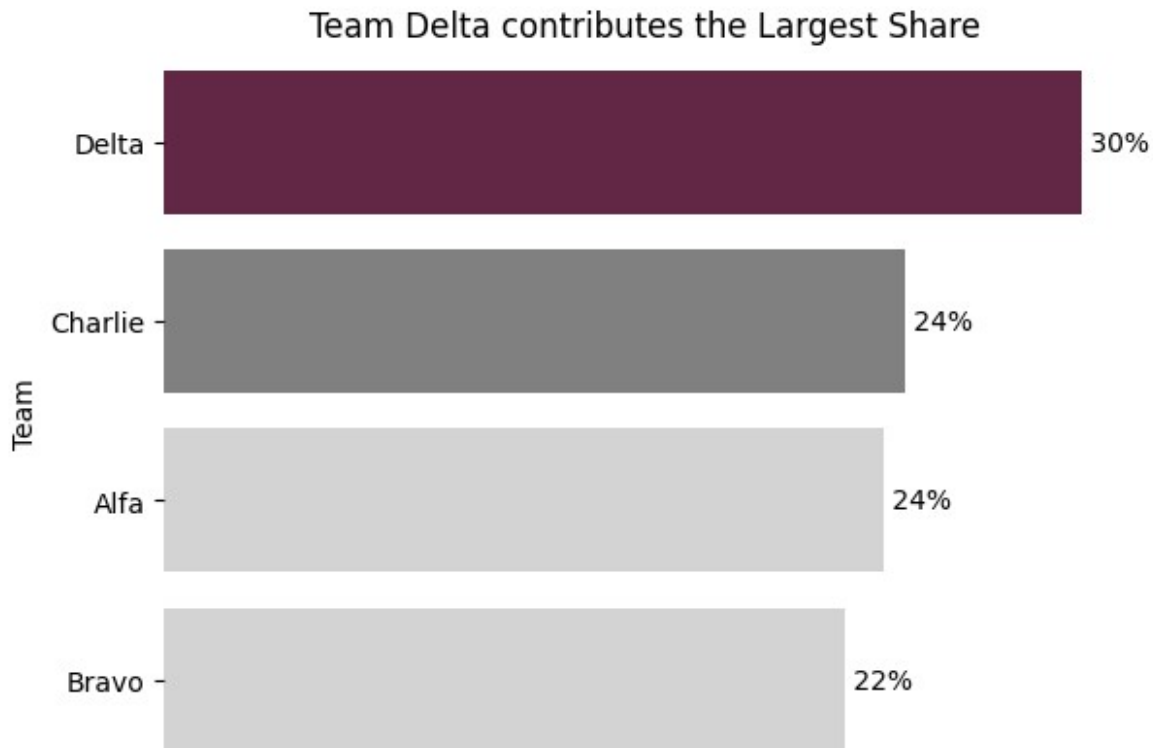
# Remove the x-axis
plt.tick_params(axis='x', which='both', bottom=False, top=False,
labelbottom=False)
plt.xlabel("")
plt.show()

C:\Users\Administrator\AppData\Local\Temp\
ipykernel_13852\3550267030.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

    ax = sns.barplot(y=sales_team_sorted['Team'],
x=sales_team_sorted['Revenue'], palette=custom_palette)
C:\Users\Administrator\AppData\Local\Temp\
ipykernel_13852\3550267030.py:22: UserWarning: The palette list has
more values (6) than needed (4), which may not be intended.
    ax = sns.barplot(y=sales_team_sorted['Team'],
x=sales_team_sorted['Revenue'], palette=custom_palette)

```



**Insight:** Team Delta makes the greatest contribution to sales, accounting for 30% of the Total Revenue.

### 13. What is the Revenue Achieved by Product Class?

```
# Aggregate the data by Product Class and Revenue
product_revenue = Forggith_Sales.groupby('Product_Class')
['Revenue'].sum().reset_index()

# Sort the aggregated product class by Revenue in descending order
product_revenue_sorted = product_revenue.sort_values(by='Revenue',
ascending=False)

# Define colors
main_color = '#6C1D45'
other_color = 'gray'
last_color = 'lightgray'

# Create custom color palette
custom_palette = [main_color] + [other_color] + [last_color] * 4

# Create count plot with sorted data
sns.barplot(y=product_revenue_sorted['Product_Class'],
x=product_revenue_sorted['Revenue'], palette=custom_palette)

plt.ylabel('Product Class')
plt.xlabel('')
```

```

plt.title('Antiseptics yield the Greatest Revenue')

# Remove the border around the plot
sns.despine(bottom=True, left=True)

# Format ticks in billions
formatter = FuncFormatter(lambda x, _: f'${x/1e9:.1f}B')
plt.gca().xaxis.set_major_formatter(formatter)

# Add data labels to the bars
for index, value in enumerate(product_revenue_sorted['Revenue']):
    plt.text(value, index, f' ${value/1e9:.1f}B', ha='left',
va='center')

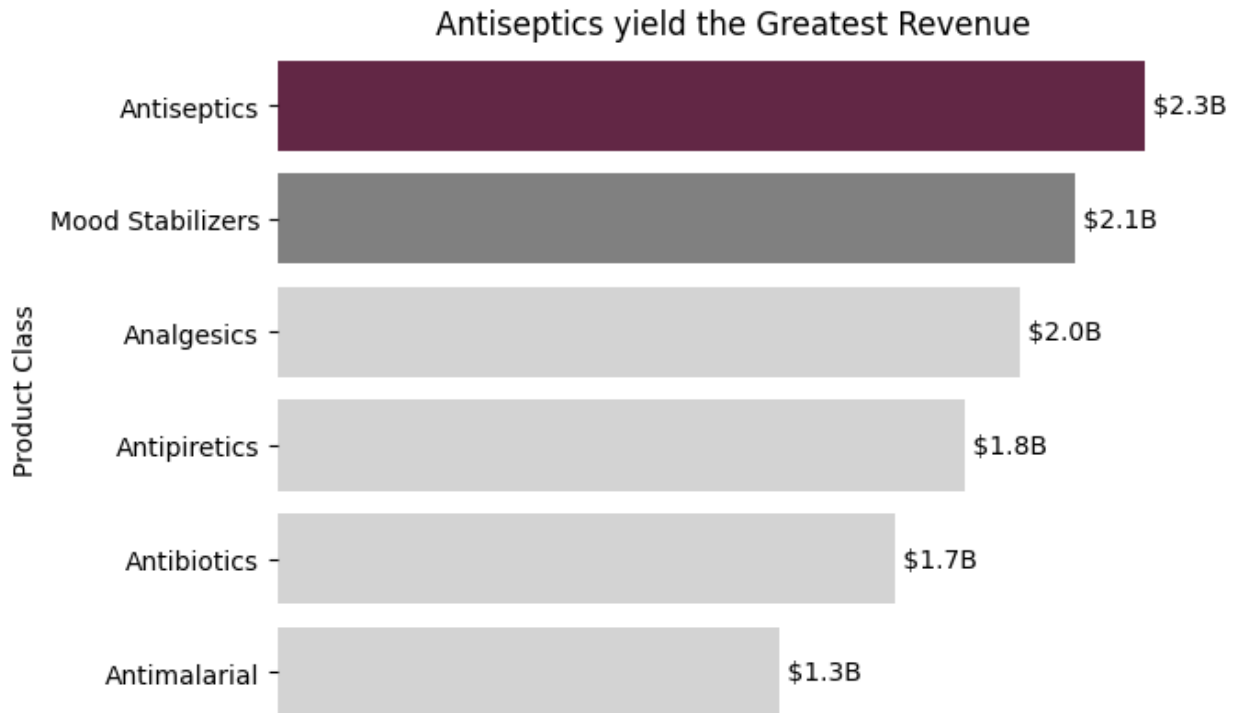
# Remove the x-axis
plt.tick_params(axis='x', which='both', bottom=False, top=False,
labelbottom=False)

plt.show()

C:\Users\Administrator\AppData\Local\Temp\
ipykernel_13852\417716778.py:16: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

sns.barplot(y=product_revenue_sorted['Product_Class'],
x=product_revenue_sorted['Revenue'], palette=custom_palette)

```



#### 14. What is the Percentage Contribution of the Product Class?

```
# Aggregate the data by Product Class and Revenue
product_revenue = Forggith_Sales.groupby('Product_Class')
['Revenue'].sum().reset_index()

# Calculate the total revenue
total_revenue = product_revenue['Revenue'].sum()

# Calculate the percentage contribution of each product
product_revenue['Percentage'] = (product_revenue['Revenue'] /
total_revenue) * 100

# Sort the aggregated product class by Revenue in descending order
product_revenue_sorted = product_revenue.sort_values(by='Revenue',
ascending=False)

# Define colors
main_color = '#6C1D45'
other_color = 'gray'
last_color = 'lightgray'

# Create custom color palette
custom_palette = [main_color] + [other_color] + [last_color] * 4

# Create count plot with sorted data
ax = sns.barplot(y=product_revenue_sorted['Product_Class'],
x=product_revenue_sorted['Revenue'], palette=custom_palette)
```

```

plt.ylabel('Product Class')
plt.xlabel('')
plt.title('Antiseptics yield the Greatest Revenue')

# Remove the border around the plot
sns.despine(bottom=True, left=True)

# Format ticks in billions
formatter = FuncFormatter(lambda x, _: f'${x/1e9:.0f}B')
plt.gca().xaxis.set_major_formatter(formatter)

# Add data labels to the bars with percentage difference
for index, (value, percentage) in
    enumerate(zip(product_revenue_sorted['Revenue'],
        product_revenue_sorted['Percentage'])):
    ax.text(value, index, f' {percentage:.0f}%', ha='left',
        va='center')

# Remove the x-axis
plt.tick_params(axis='x', which='both', bottom=False, top=False,
    labelbottom=False)

plt.show()

```

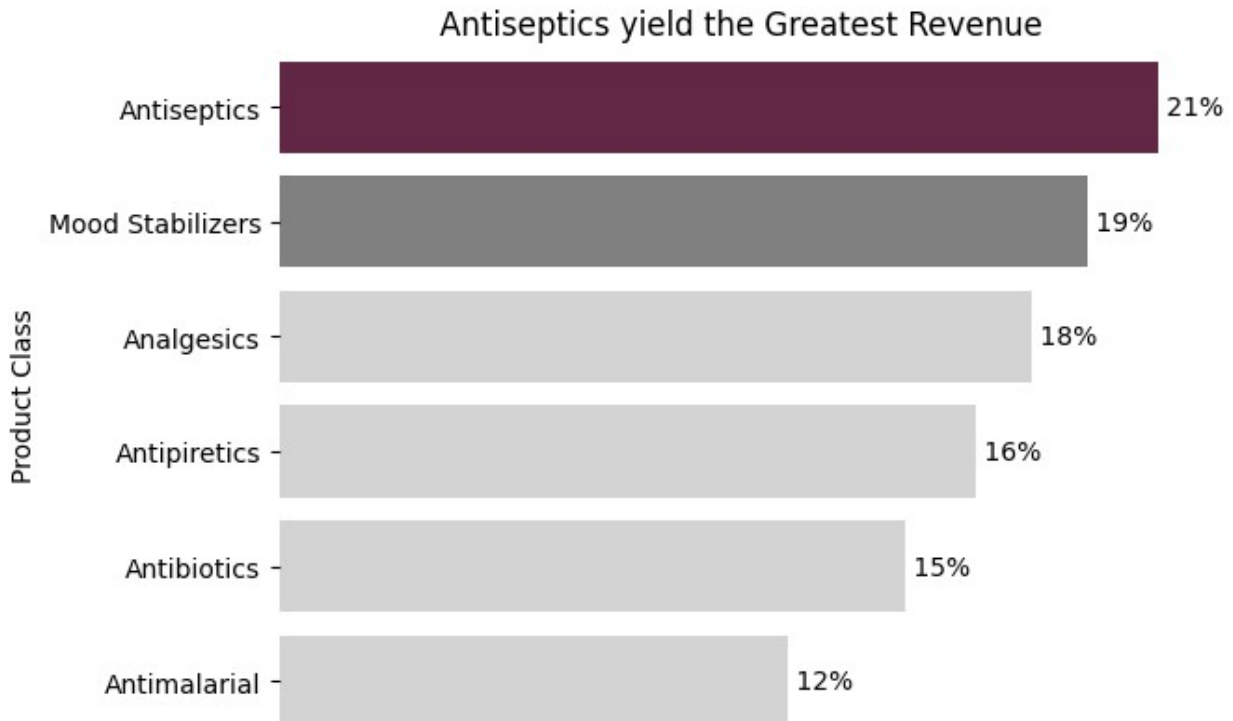
C:\Users\Administrator\AppData\Local\Temp\ipykernel\_13852\1471687384.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```

ax = sns.barplot(y=product_revenue_sorted['Product_Class'],
x=product_revenue_sorted['Revenue'], palette=custom_palette)

```



**Insight:** The antiseptic category of products accounts for the highest proportion of revenue.

#### 15. What is the Volume Achievement by Product Class?

```
# Aggregate the data by Product Class and Quantity
product_quantity = Forggith_Sales.groupby('Product_Class')
['Quantity'].sum().reset_index()

# Sort the aggregated product class by Quantity in descending order
product_quantity_sorted = product_quantity.sort_values(by='Quantity',
ascending=False)

# Define colors
main_color = '#6C1D45'
other_color = 'gray'
last_color = 'lightgray'

# Create custom color palette
custom_palette = [main_color] + [other_color] + [last_color] * 4

# Create count plot with sorted data
sns.barplot(y=product_quantity_sorted['Product_Class'],
x=product_quantity_sorted['Quantity'], palette=custom_palette)

plt.ylabel('Product Class')
plt.xlabel('')
plt.title('Volume Achieved by Product Class')
```

```

# Remove the border around the plot
sns.despine(bottom=True, left=True)

# Format ticks in millions
formatter = FuncFormatter(lambda x, _: f'{x/1e6:.1f}M')
plt.gca().xaxis.set_major_formatter(formatter)

# Add data labels to the bars
for index, value in enumerate(product_quantity_sorted['Quantity']):
    plt.text(value, index, f' {value/1e6:.1f}M', ha='left',
va='center')

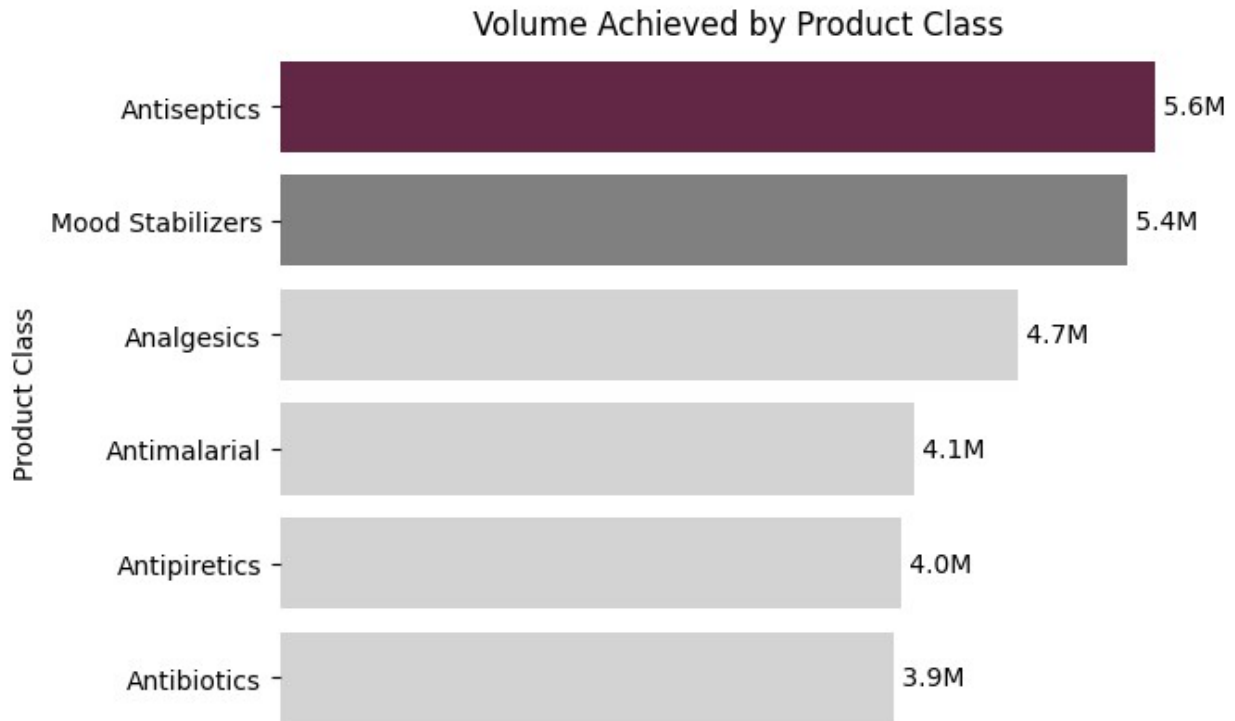
# Remove the x-axis
plt.tick_params(axis='x', which='both', bottom=False, top=False,
labelbottom=False)

plt.show()

C:\Users\Administrator\AppData\Local\Temp\
ipykernel_13852\3678546630.py:16: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

sns.barplot(y=product_quantity_sorted['Product_Class'],
x=product_quantity_sorted['Quantity'], palette=custom_palette)

```



#### 16. What is the Revenue Trend from 2022 to 2025?

```
# Convert "Date" column to datetime type
Forggith_Sales["Date"] = pd.to_datetime(Forggith_Sales["Date"])

Forggith_Sales["Year"] = Forggith_Sales["Date"].dt.strftime("%Y")

# Pre-aggregate data by summing revenues for each year
aggregated_data = Forggith_Sales.groupby("Year")
["Revenue"].sum().reset_index()

# Convert revenue to billions format
aggregated_data["Revenue (Billions)"] = aggregated_data["Revenue"] /
1e9 # dividing by 1 billion

# Plotting
plt.fill_between(aggregated_data["Year"], aggregated_data["Revenue
(Billions)"], color="#6C1D45", alpha=0.5)

# Adding data points
plt.plot(aggregated_data["Year"], aggregated_data["Revenue
(Billions)"], color="#6C1D45", marker='o')

# Adding data label
label_offset = 0.1 # adjust this value to lift the labels more or
less

# Adding data label
```



```

for i, point in aggregated_data.iterrows():
    plt.text(point['Year'], point['Revenue (Billions)'] +
label_offset, f"{point['Revenue (Billions)']:.2f}B", ha='right')

plt.title("Revenue increased from 2022 to 2024, but drastically
decline in 2025", pad=20)
plt.ylabel("") # Empty y-axis label
plt.xlabel("Year")

# Set starting axis to zero
plt.ylim(0)

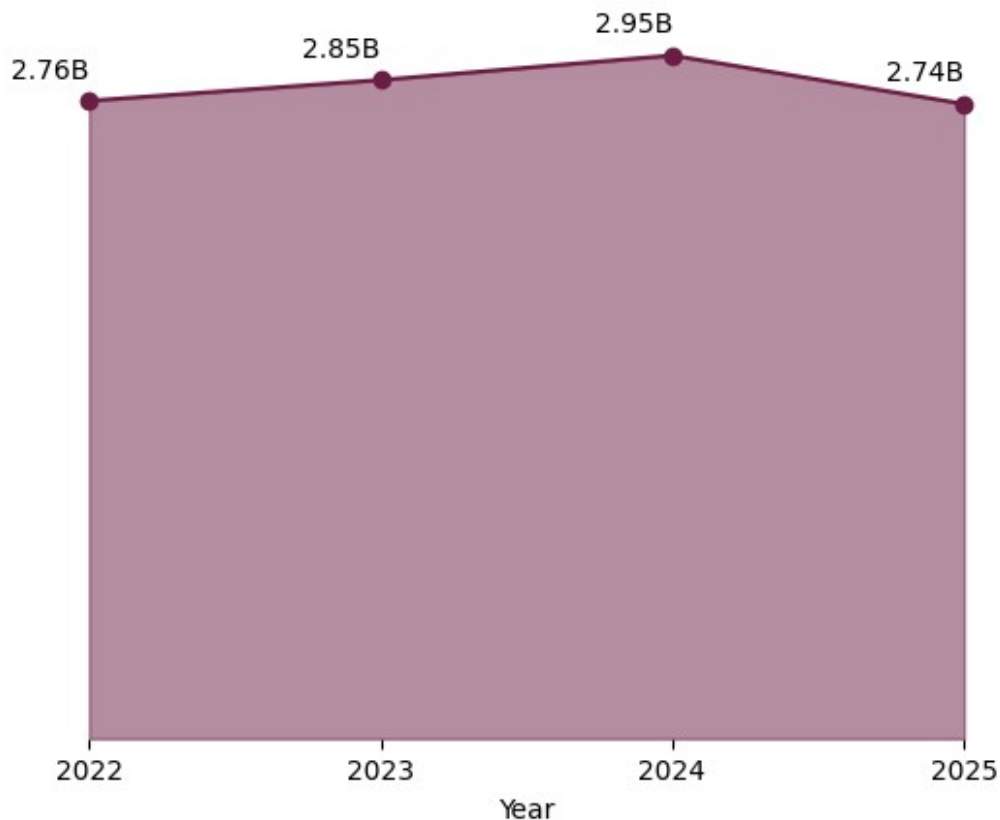
# Remove y-axis ticks
plt.yticks([])

# Remove all spines
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.gca().spines['bottom'].set_visible(False)
plt.gca().spines['left'].set_visible(False)

plt.show()

```

Revenue increased from 2022 to 2024, but drastically decline in 2025



**Insight:** Revenue increased from 2022 to 2024, but drastically decline in 2025.

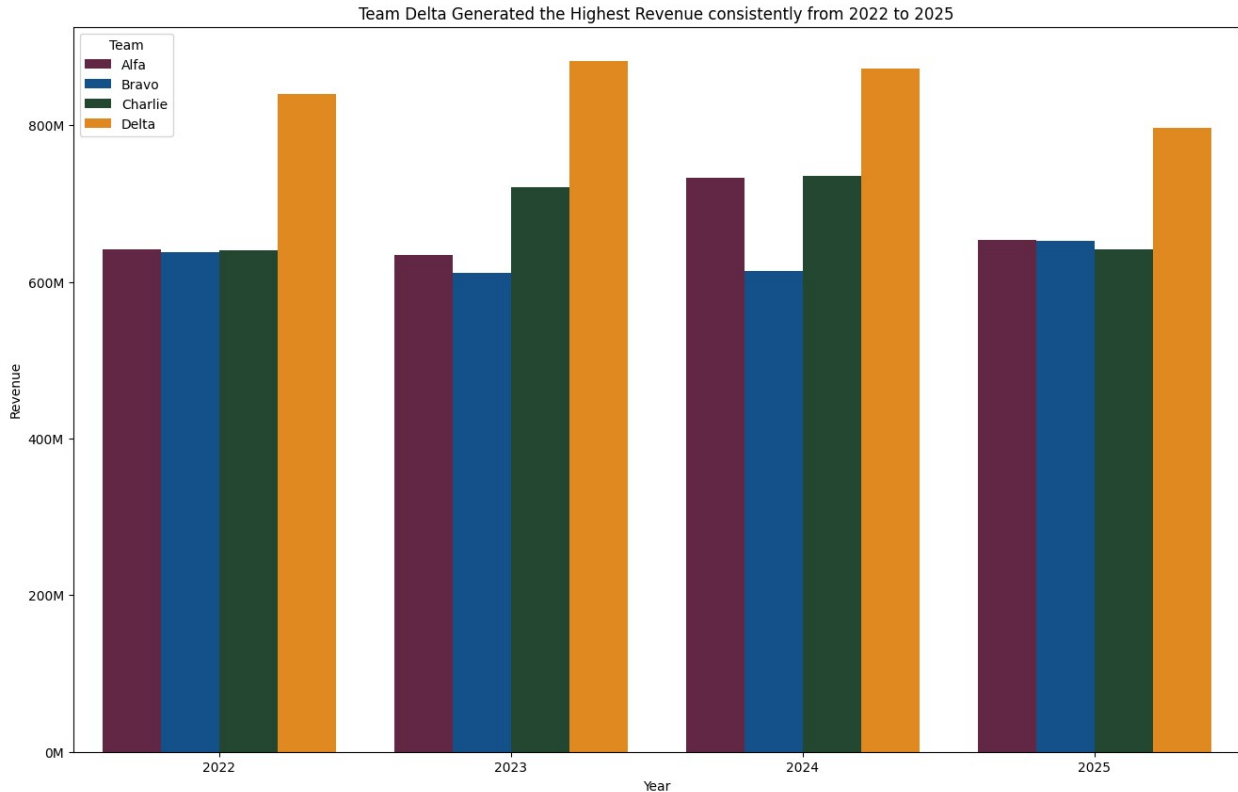
```
Forggith_Sales["Channel"] = Forggith_Sales["Sub_Channel_Name"] + " " +  
Forggith_Sales["Facility_Type"]  
Forggith_Sales["Channel"].head()
```

```
S/N  
0      Government Hospital  
1      Private Hospital  
2      Institution Pharmacy  
3      Retail Pharmacy  
4      Government Hospital  
Name: Channel, dtype: object
```

```
Team_Revenue = Forggith_Sales.groupby(["Year",  
"Team"]).agg({"Revenue": "sum"}).astype(int)  
Team_Revenue
```

### 17. What is the Yearly Revenue Achieved by Sales Team?

```
# Your data and plot setup  
Team_Revenue = Forggith_Sales.groupby(["Year",  
"Team"]).agg({"Revenue": "sum"})  
  
custom_palette = ["#6C1D45", "#00529F", "#1D4D2E", "#FF8C00",  
"#D1D1D1", "#4A4A4A", "#72B7D2", "#8ABD3B"]  
  
plt.figure(figsize=(16, 10))  
sns.set_palette(custom_palette)  
  
# Creating the bar plot  
ax = sns.barplot(data=Team_Revenue.reset_index(), x="Year",  
y="Revenue", hue="Team", estimator=sum)  
  
# Adding title  
plt.title("Team Delta Generated the Highest Revenue consistently from  
2022 to 2025")  
  
# Custom formatting function for y-axis ticks  
def millions_formatter(x, pos):  
    return f'{x / 10**6:.0f}M'  
  
# Applying the custom formatter to the y-axis  
ax.yaxis.set_major_formatter(FuncFormatter(millions_formatter))  
  
# Show plot  
plt.show()
```



**Insight** Team Delta generated the highest Revenue consistently from 2022 to 2025.

### 18. What is the Yearly Revenue Achieved by Product Class?

```
# Extracting year from Date column
Forggith_Sales["Year"] = Forggith_Sales["Date"].dt.strftime("%Y")

# Grouping by Year and Product_Class and aggregating Revenue
Product_Class_Revenue = Forggith_Sales.groupby(["Year",
"Product_Class"]).agg({"Revenue": "sum"}).astype(int)

# Modify legend colors
custom_palette = ["#6C1D45", "#00529F", "#1D4D2E", "#FF8C00",
"#D1D1D1", "#4A4A4A", "#72B7D2", "#8ABD3B"]

# Set the size of the plot
plt.figure(figsize=(16, 10))

# Setting the palette
sns.set_palette(custom_palette)

# Creating the bar plot
ax = sns.barplot(data=Product_Class_Revenue.reset_index(), x="Year",
y="Revenue", hue="Product_Class", estimator=sum)

# Adding title
plt.title("Consistently from 2022 to 2025, Antiseptics and Mood")
```

```

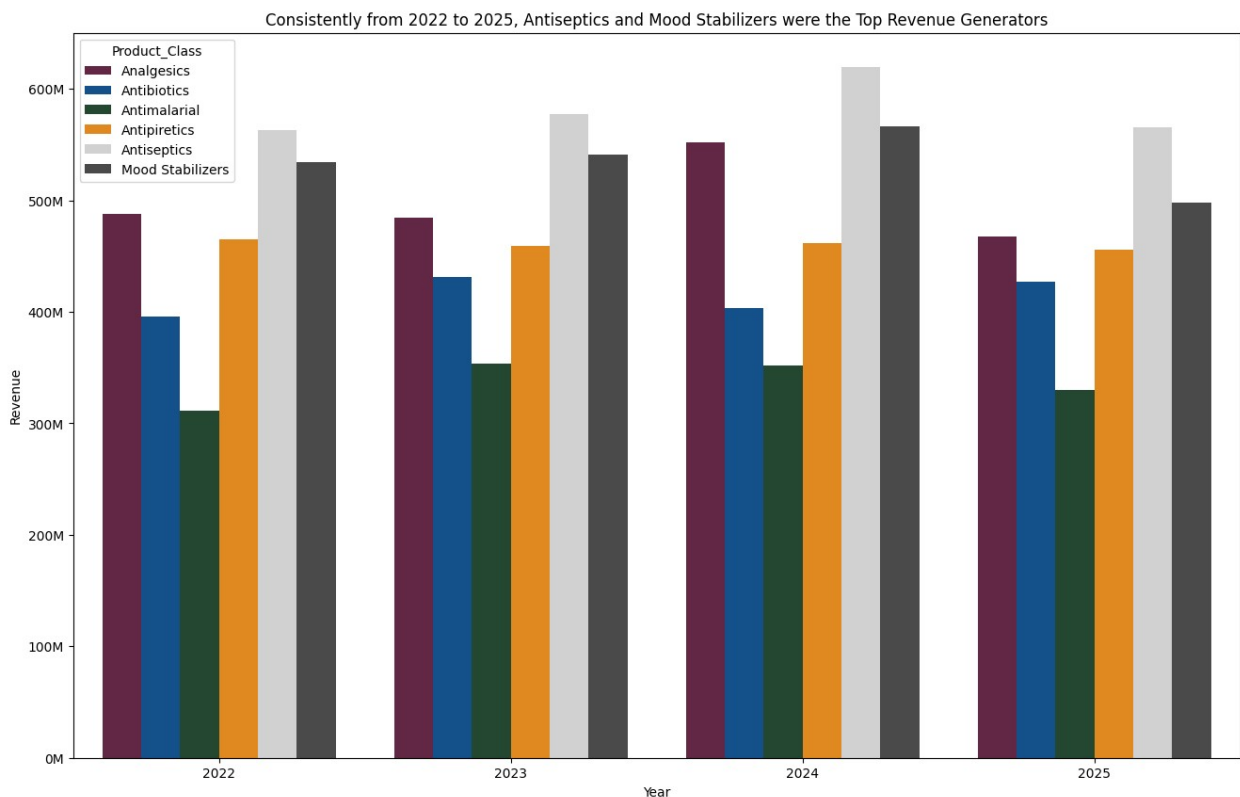
Stabilizers were the Top Revenue Generators")

# Custom formatting function for y-axis ticks
def millions_formatter(x, pos):
    return f'{x / 10**6:.0f}M'

# Applying the custom formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(millions_formatter))

# Show plot
plt.show()

```



**Insight** Consistently from 2022 to 2025, Antiseptics and Mood Stabilizers were the Top Revenue Generators.

### 19. What is the Yearly Revenue by Channel?

```

# Extracting year from Date column
Forggith_Sales["Year"] = Forggith_Sales["Date"].dt.strftime("%Y")

# Creating the Channel column
Forggith_Sales["Channel"] = Forggith_Sales["Sub_Channel_Name"] + " " +
Forggith_Sales["Facility_Type"]

# Grouping by Year and Channel and aggregating Revenue
Channel_Revenue = Forggith_Sales.groupby(["Year",

```

```
"Channel"]).agg({"Revenue": "sum"}).astype(int)

# Modify legend colors
custom_palette = ["#6C1D45", "#00529F", "#1D4D2E", "#FF8C00",
                  "#D1D1D1", "#4A4A4A", "#72B7D2", "#8ABD3B"]

# Set the size of the plot
plt.figure(figsize=(16, 10))

# Setting the palette
sns.set_palette(custom_palette)

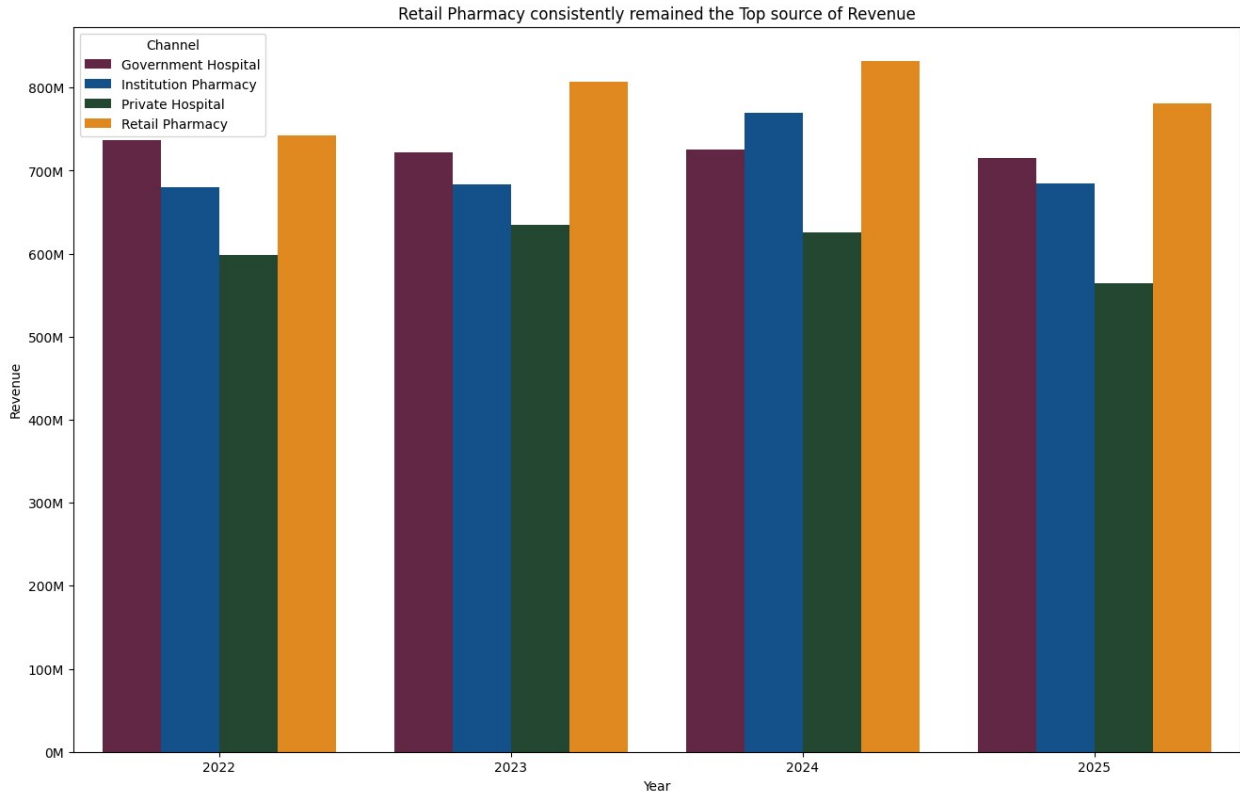
# Creating the bar plot
ax = sns.barplot(data=Channel_Revenue.reset_index(), x="Year",
                 y="Revenue", hue="Channel", estimator=sum)

# Adding title
plt.title("Retail Pharmacy consistently remained the Top source of Revenue")

# Custom formatting function for y-axis ticks
def millions_formatter(x, pos):
    return '{:.0f}M'.format(x / 10**6)

# Applying the custom formatter to the y-axis
ax.yaxis.set_major_formatter(FuncFormatter(millions_formatter))

# Show plot
plt.show()
```



**Insight:** Throughout the period spanning 2022 to 2025, Retail Pharmacy and Government Hospital maintained their status as the Leading Revenue Contributors.

#### Creating a new month column from the Forgith Sales Month Year Column

```
Forggith_Sales["Month"] = Forggith_Sales["Date"].dt.strftime('%B')
# Mapping month numbers to month names
month_mapping = {
    'January': 'Jan', 'February': 'Feb', 'March': 'Mar',
    'April': 'Apr', 'May': 'May', 'June': 'Jun',
    'July': 'Jul', 'August': 'Aug', 'September': 'Sep',
    'October': 'Oct', 'November': 'Nov', 'December': 'Dec'
}

# Applying the mapping to the DataFrame
Forggith_Sales['Month'] = Forggith_Sales['Month'].map(month_mapping)

Forggith_Sales["Month"].unique()

array(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep',
       'Oct', 'Nov', 'Dec'], dtype=object)

Forggith_Targets.head(20)
```

	SalesRepID	Sales_Rep_Name	Month	Month_number	Year
Date \					
ProductID					

Aba-Ant	SN20038	Morris Garcia	Jan	1	2022
1/1/2022					
Aba-Ant	SN20038	Morris Garcia	Jan	1	2023
1/1/2023					
Aba-Ant	SN20038	Morris Garcia	Jan	1	2024
1/1/2024					
Aba-Ant	SN20038	Morris Garcia	Jan	1	2025
1/1/2025					
Aba-Ant	SN20038	Morris Garcia	Feb	2	2022
2/1/2022					
Aba-Ant	SN20038	Morris Garcia	Feb	2	2023
2/1/2023					
Aba-Ant	SN20038	Morris Garcia	Feb	2	2024
2/1/2024					
Aba-Ant	SN20038	Morris Garcia	Feb	2	2025
2/1/2025					
Aba-Ant	SN20038	Morris Garcia	Mar	3	2022
3/1/2022					
Aba-Ant	SN20038	Morris Garcia	Mar	3	2023
3/1/2023					
Aba-Ant	SN20038	Morris Garcia	Mar	3	2024
3/1/2024					
Aba-Ant	SN20038	Morris Garcia	Mar	3	2025
3/1/2025					
Aba-Ant	SN20038	Morris Garcia	Apr	4	2022
4/1/2022					
Aba-Ant	SN20038	Morris Garcia	Apr	4	2023
4/1/2023					
Aba-Ant	SN20038	Morris Garcia	Apr	4	2024
4/1/2024					
Aba-Ant	SN20038	Morris Garcia	Apr	4	2025
4/1/2025					
Aba-Ant	SN20038	Morris Garcia	May	5	2022
5/1/2022					
Aba-Ant	SN20038	Morris Garcia	May	5	2023
5/1/2023					
Aba-Ant	SN20038	Morris Garcia	May	5	2024
5/1/2024					
Aba-Ant	SN20038	Morris Garcia	May	5	2025
5/1/2025					

	Target_Quantity	Product_Price	Target_Amount
ProductID			
Aba-Ant	168	742	124656
Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208
Aba-Ant	168	742	124656

Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208
Aba-Ant	168	742	124656
Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208
Aba-Ant	168	742	124656
Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208
Aba-Ant	168	742	124656
Aba-Ant	185	742	137270
Aba-Ant	204	742	151368
Aba-Ant	224	742	166208

## 20. What is the Total Revenue Year To Date (January to December 2025)?

```
start_date = '2025-01-01'
end_date = '2025-12-01'
date_range_mask = (Forggith_Sales['Date'] >= start_date) &
(Forggith_Sales["Date"]<= end_date)
Forggith_Sales_2025 = Forggith_Sales[date_range_mask]

total_revenue_ytd = int(Forggith_Sales_2025["Revenue"].sum())

print("The Total Revenue for the year 2025 is $
{:,.}.".format(total_revenue_ytd))
```

The Total Revenue for the year 2025 is \$2,744,203,376.

## 21. What is the Total Revenue Same Period Last Year (January 2024 to December 2024)?

```
start_date_sply = '2024-01-01'
end_date_sply = '2024-12-01'
date_range_mask_sply = (Forggith_Sales['Date'] >= start_date_sply) &
(Forggith_Sales["Date"]<= end_date_sply)
Forggith_Sales_2024 = Forggith_Sales[date_range_mask_sply]

total_revenue_sply = int(Forggith_Sales_2024["Revenue"].sum())

print("The Total Revenue for January 2024 to December 2024 is $
{:,.}.".format(total_revenue_sply))
```

The Total Revenue for January 2024 to December 2024 is \$2,953,925,683.

**Insight:** The company made more revenue in the year 2025 compared to 2024.

## 22. What is the Total Revenue Previous Year To Date from January 2024 to December 2025?



```
start_date_PYTD = '2024-01-01'
end_date_PYTD = '2025-12-01'
date_range_mask_PYTD = (Forggith_Sales['Date'] >= start_date_PYTD) &
(Forggith_Sales["Date"]<= end_date_PYTD)
Forggith_Sales_2024_2025 = Forggith_Sales[date_range_mask_PYTD]

total_revenue_pytd = int(Forggith_Sales_2024_2025["Revenue"].sum())

print("The Total Revenue from January 2024 to December 2025 is $
{:,.}.".format(total_revenue_pytd))
```

The Total Revenue from January 2024 to December 2025 is  
\$5,698,129,059.

## Recommendations

- **Recognize and Reward Top Performers:** Given that Thompson Crawford, Daniel Gates, and Jimmy Grey are top contributors to revenue, consider implementing a recognition program to acknowledge their efforts. This could include bonuses, incentives, or public recognition within the organization to motivate them to maintain or even improve their performance.
- **Sales Training and Development:** Since Team Delta makes a significant contribution to sales, investing in their training and development can further enhance their performance. Providing them with advanced sales techniques, product knowledge, and effective communication skills can help them excel even more.
- **Diversification of Revenue Streams:** While the antiseptic category of products is currently the highest revenue generator, explore opportunities to diversify product offerings or expand into new markets to reduce reliance on a single category. This can help mitigate risks associated with fluctuations in demand for specific product categories.
- **Investigate the Decline in 2025 Revenue:** Analyze the factors contributing to the drastic decline in revenue in 2025. It could be due to external factors such as changes in market conditions, competitive pressures, or internal factors such as operational inefficiencies or lack of innovation. Addressing these issues promptly can help prevent further declines in revenue in the future.
- **Review Target Setting Process:** Evaluate the target-setting process to ensure that targets are realistic yet challenging. While surpassing targets can be motivating, consistently setting targets that are too easily achievable may not encourage optimal performance. Adjusting targets based on market conditions and historical performance can lead to more meaningful goals.
- **Strengthen Customer Relationships:** Since Retail Pharmacy and Government Hospital are leading revenue contributors, focus on strengthening relationships with these key customers. This could involve providing personalized service, addressing their specific needs, and offering incentives to encourage repeat business.
- **Long-Term Strategic Planning:** Develop a long-term strategic plan that takes into account market trends, competitive landscape, and potential challenges. This can help

guide decision-making and resource allocation to ensure sustained growth and profitability beyond the current planning period.

## Conclusion

The analysis of the Forggith pharmaceuticals performance data from 2022 to 2025 provides valuable insights into the achievements and areas for improvement within the organization. Despite surpassing the targets for both revenue and volume, there are notable trends and opportunities that require attention to sustain growth and success in the future.