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



**Team Members**

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

**Table of Content**

* **Introduction**
* **Objective**
* **Dataset Description**
* **Questions answered in this analysis**
* Insights derived from this analysis
* **Observations**
* **Exploratory Data Analysis**
* Recommendation
* 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.

**Data Source:** Foresight BI Internship Dataset

**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"]

**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\_Target.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 columns**

Forggith\_Sales.head()

Forggith\_Targets.head()

**> \*\*Checked the column names\*\***

Forggith\_Sales.columns

Forggith\_Targets.columns

**Determined the number of rows and columns**

Forggith\_Sales.shape

Forggith\_Targets.shape

**> \*\*Determined the non null count and data types\*\***

Forggith\_Sales.info()

Forggith\_Targets.info()

**Checked for missing values**

Forggith\_Targets.isnull().sum()

Forggith\_Targets.isnull().sum()

Forggith\_Sales.isna().sum()

Forggith\_Targets.isna().sum()

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

**> \*\*Determined if there were Duplicated rows\*\***

Forggith\_Sales.duplicated().sum()

Forggith\_Targets.duplicated().sum()

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

**Determined the summary statistics**

# Summary statistics of the Sales data

Forggith\_Sales.describe()

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

# Summary statistics of Targets data

Forggith\_Targets.describe()

> **\*\*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

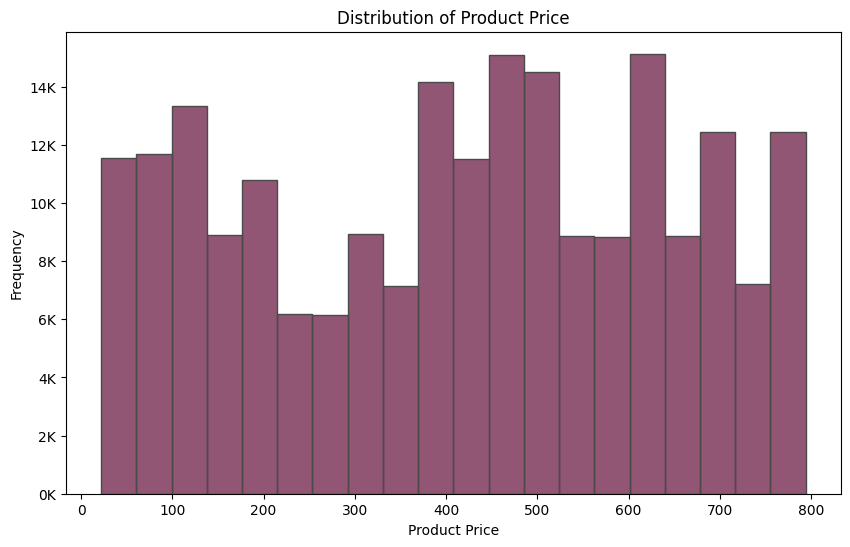
plt.gca().yaxis.set\_major\_formatter(FuncFormatter(format\_thousands))

plt.xlabel('Product Price')

plt.ylabel('Frequency')

plt.title('Distribution of Product Price')

plt.show()



Forggith\_Sales['Quantity'].unique()

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')

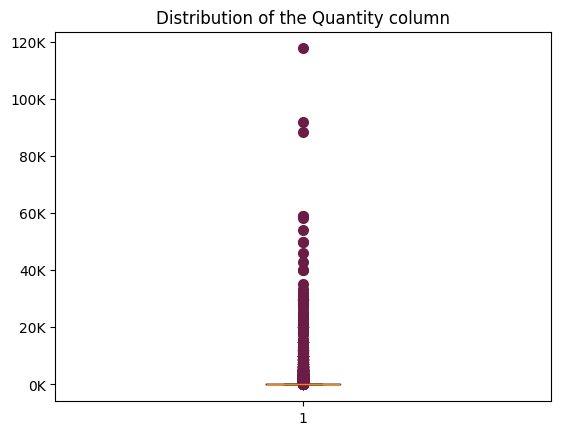
# Define formatter function to display y-axis ticks in thousands with 'K' suffix

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))

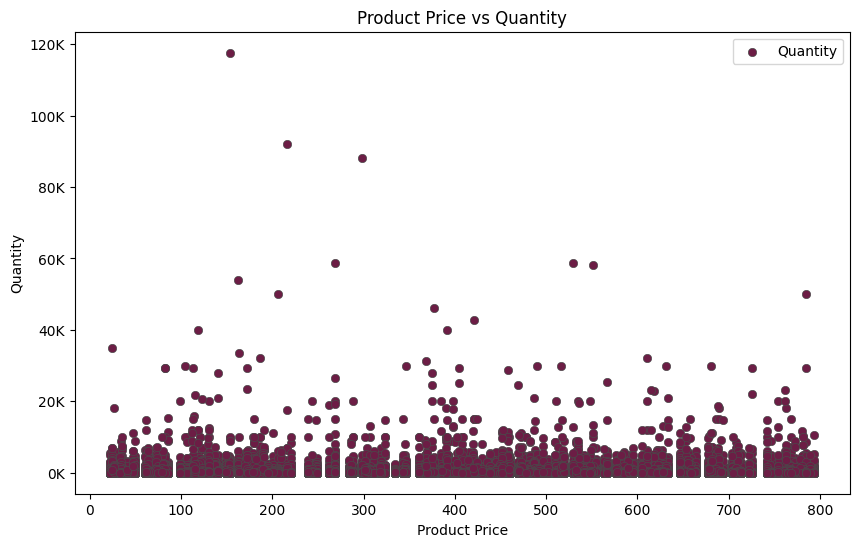
plt.xlabel('Product Price')

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)

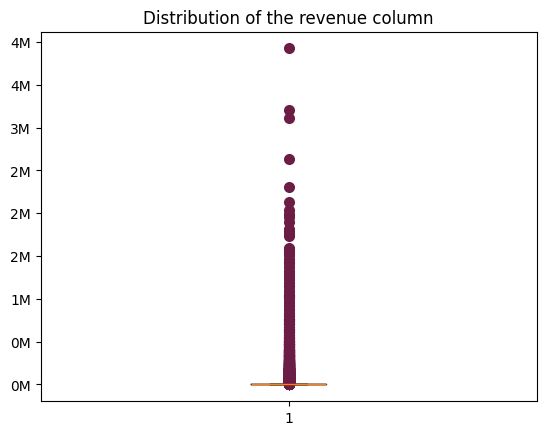
# Create a FuncFormatter object using the custom formatting function

formatter = FuncFormatter(millions\_formatter)

plt.gca().yaxis.set\_major\_formatter(formatter)

plt.gca().yaxis.set\_major\_formatter(formatter)

plt.show()



**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))

1. What is the Total Target from 2022 to 2025?

# Total Target for the yaer 2022 t0 2025?

Total\_Target = int(Forggith\_Targets['Targets'].sum())

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

1. What is the percentage of Revenue Achieved to Target?

Revenue\_Target = int(Total\_Revenue\*100/Total\_Target)

print('The Percentage of Revenue Achieved is {}%'.format(Revenue\_Target))

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

1. 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))

1. What is the Target Volume?

Target\_Volume\_achieved = Forggith\_Targets["Target\_Quantity"].sum()

print('The Target Volume Achieved is {:,}.'.format(Target\_Volume\_achieved))

1. What is the percentage of Revenue Volume Achieved to Target Volume?

RevenueVolume\_TargetVolume = int(Revenue\_Volume\_achieved\*100/Target\_Volume\_achieved)

print('The Percentage of Revenue Volume Achieved to Target Volume is {:,}%.'.format(RevenueVolume\_TargetVolume))

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

1. 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 = Forggith\_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):

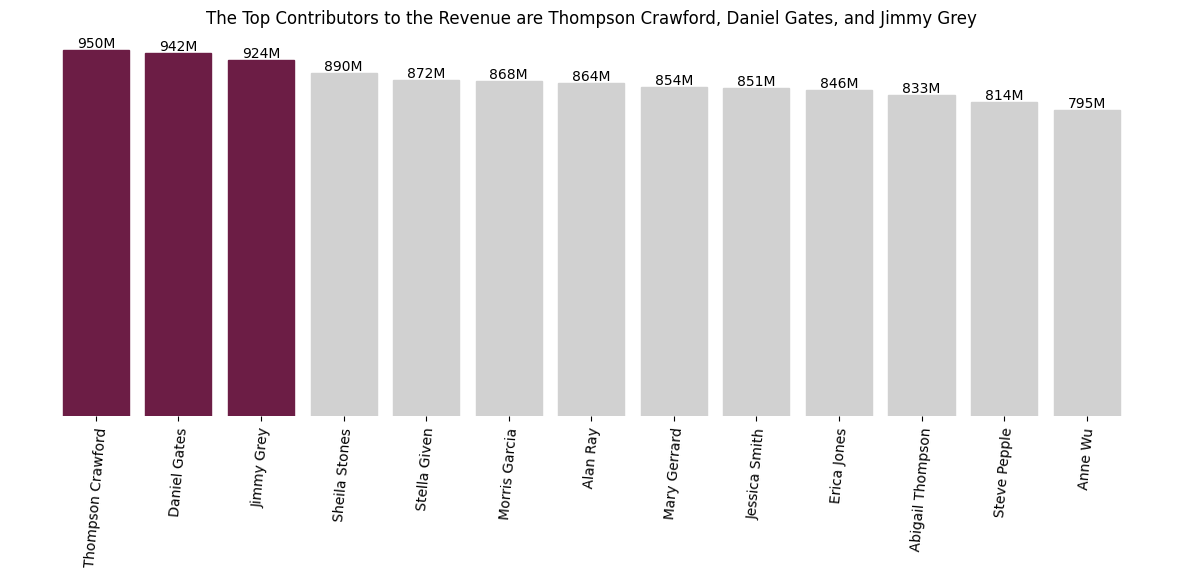
    bar.set\_color(color)

# 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.

1. What is the Target Amount 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\_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)

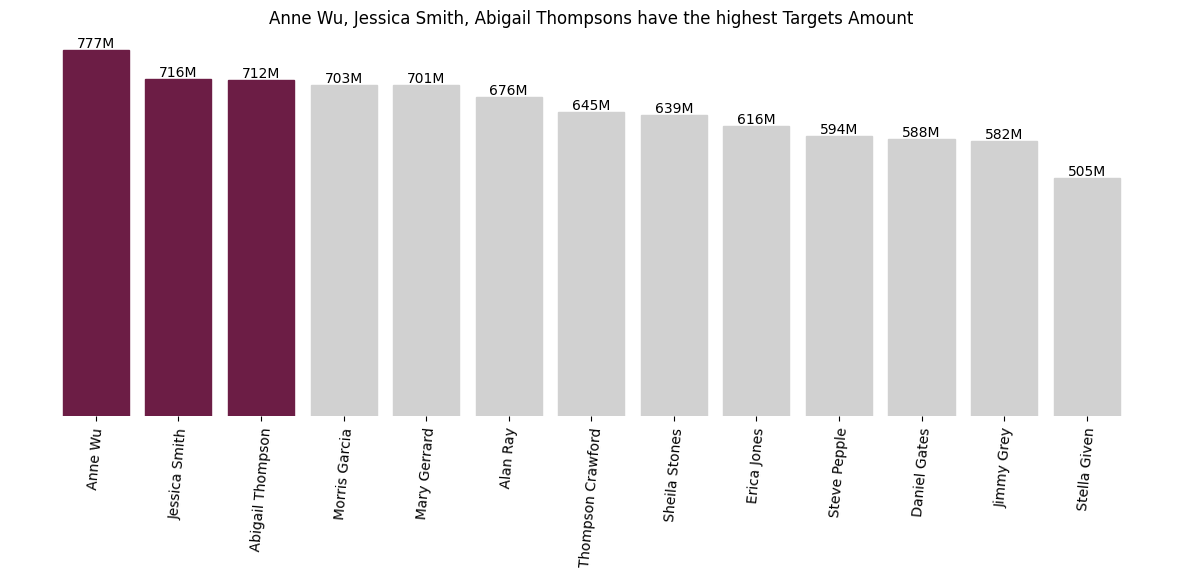
# Hide the y-axis label

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.

1. 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)

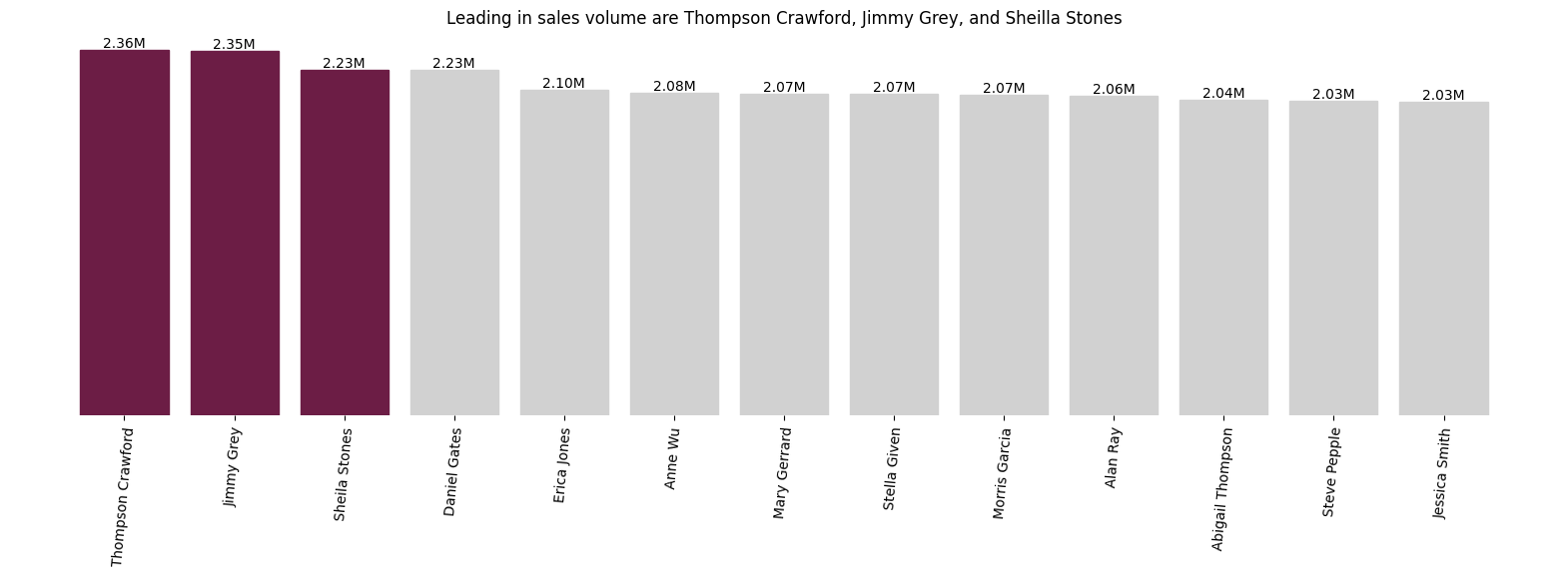
# Hide the y-axis label

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.

1. 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)

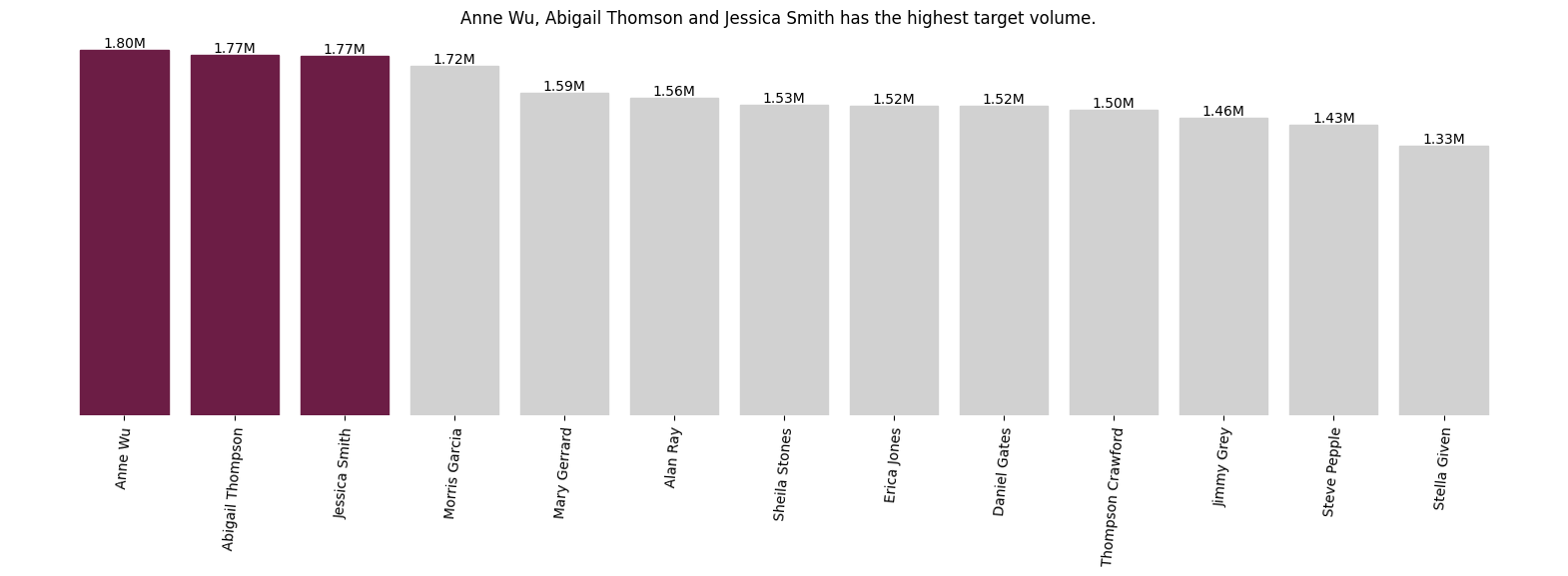
# Hide the y-axis label

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.

1. 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

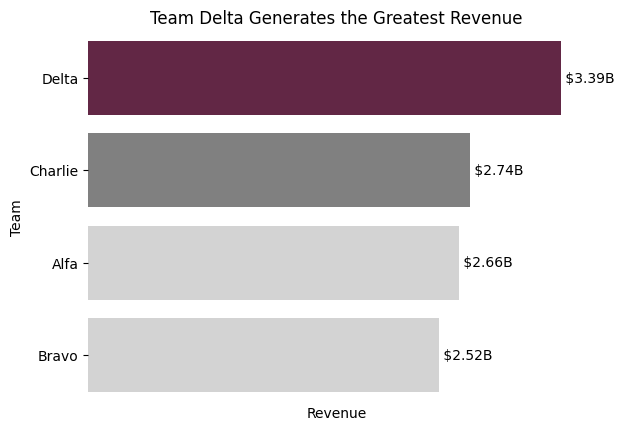
for index, value in enumerate(sales\_team\_sorted['Revenue']):

    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()



1. 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 DeltaTeam 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'])):

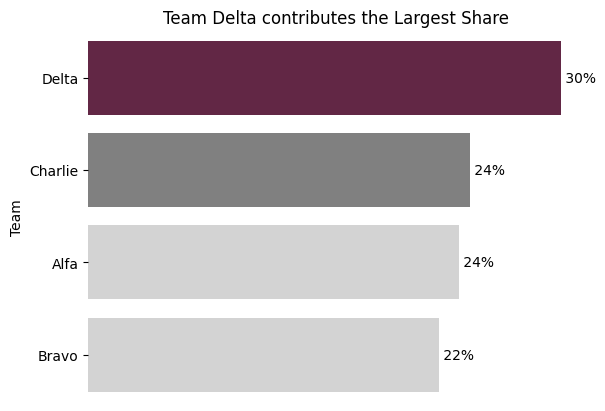
    plt.text(value, index, f' {percentage:.0f}%', ha='left', va='center')

# Remove the x-axis

plt.tick\_params(axis='x', which='both', bottom=False, top=False, labelbottom=False)

plt.xlabel("")

plt.show()



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

1. 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

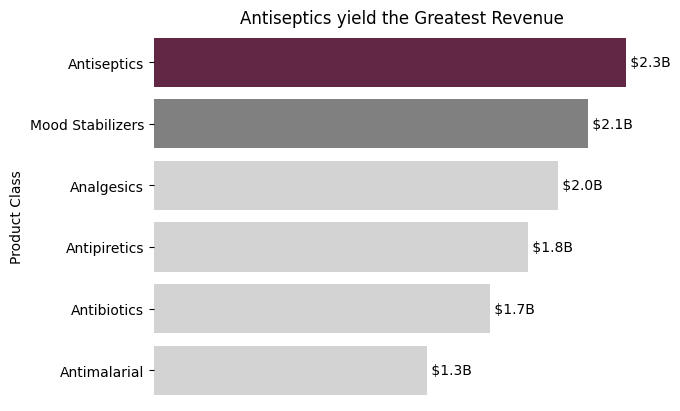
for index, value in enumerate(product\_revenue\_sorted['Revenue']):

    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()



1. 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

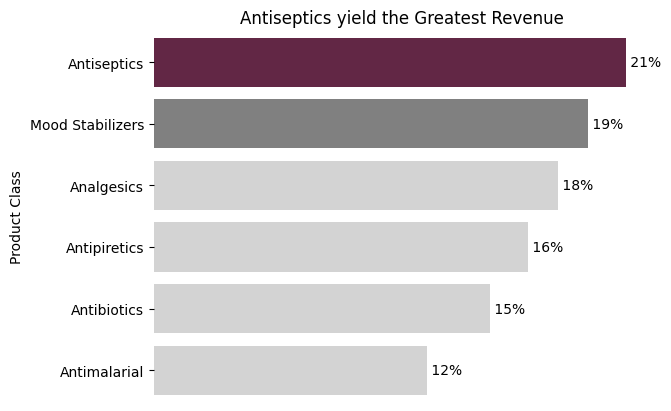
for index, (value, percentage) in enumerate(zip(product\_revenue\_sorted['Revenue'], product\_revenue\_sorted['Percentage'])):

    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()



1. 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

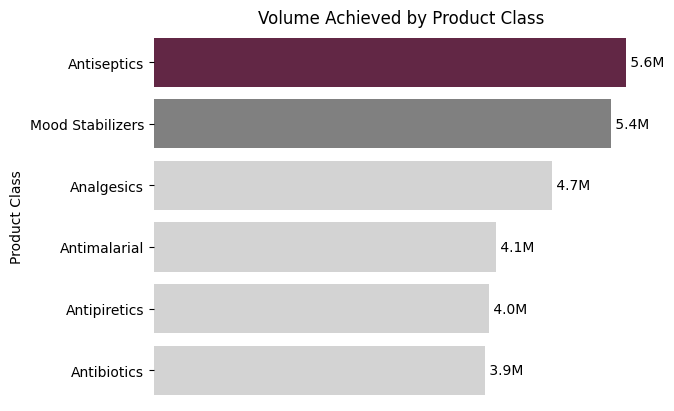
for index, value in enumerate(product\_quantity\_sorted['Quantity']):

    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()



1. 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

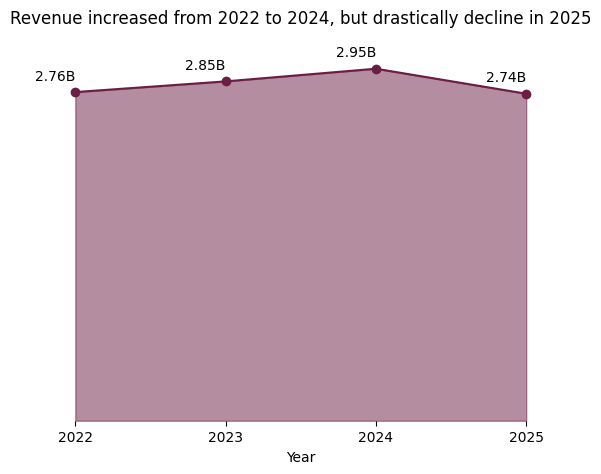
plt.gca().spines['top'].set\_visible(False)

plt.gca().spines['right'].set\_visible(False)

plt.gca().spines['bottom'].set\_visible(False)

plt.gca().spines['left'].set\_visible(False)

plt.show()



> \*\*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()

Team\_Revenue = Forggith\_Sales.groupby(["Year", "Team"]).agg({"Revenue":"sum"}).astype(int)

Team\_Revenue

1. 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):

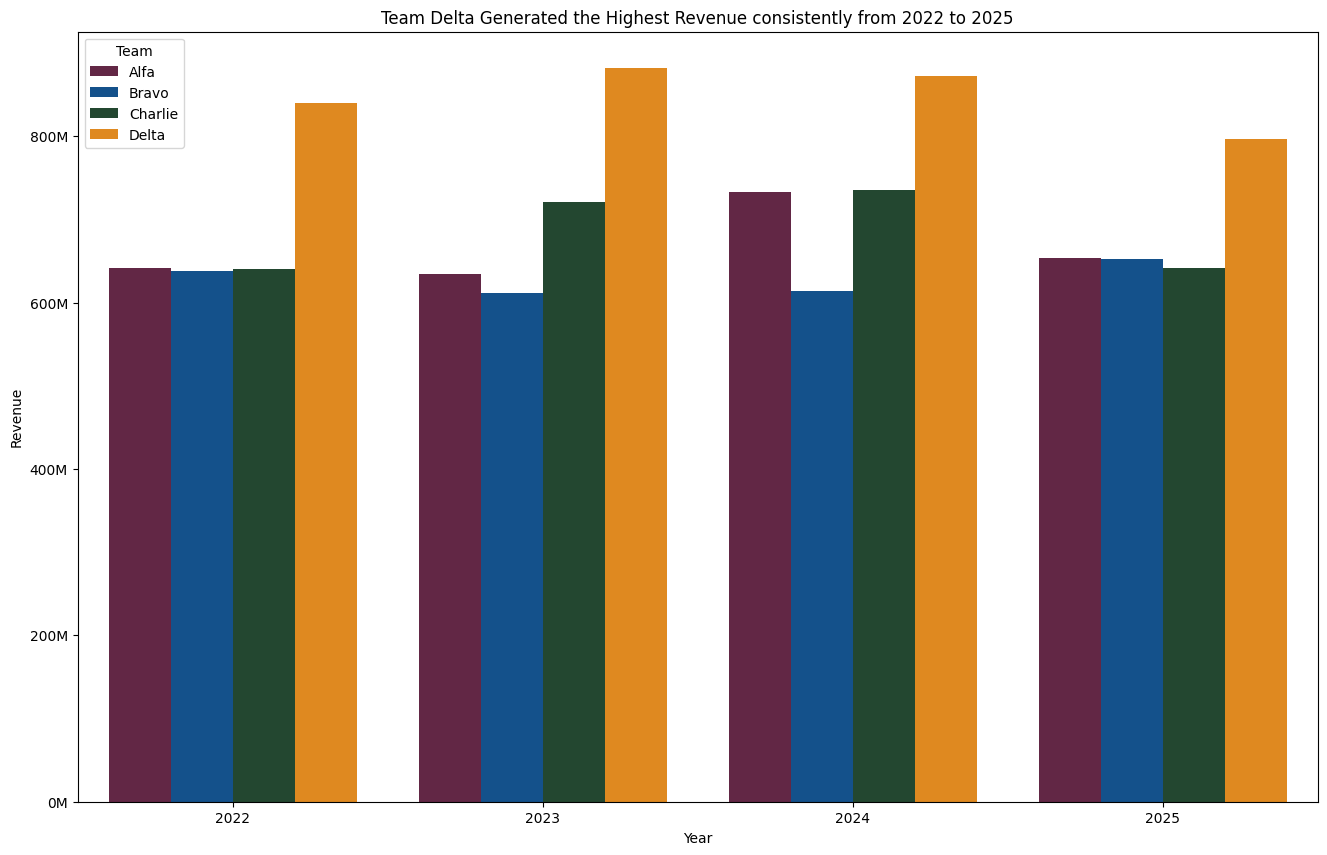
    return f'{x / 10\*\*6:.0f}M'

# 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.

1. 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):

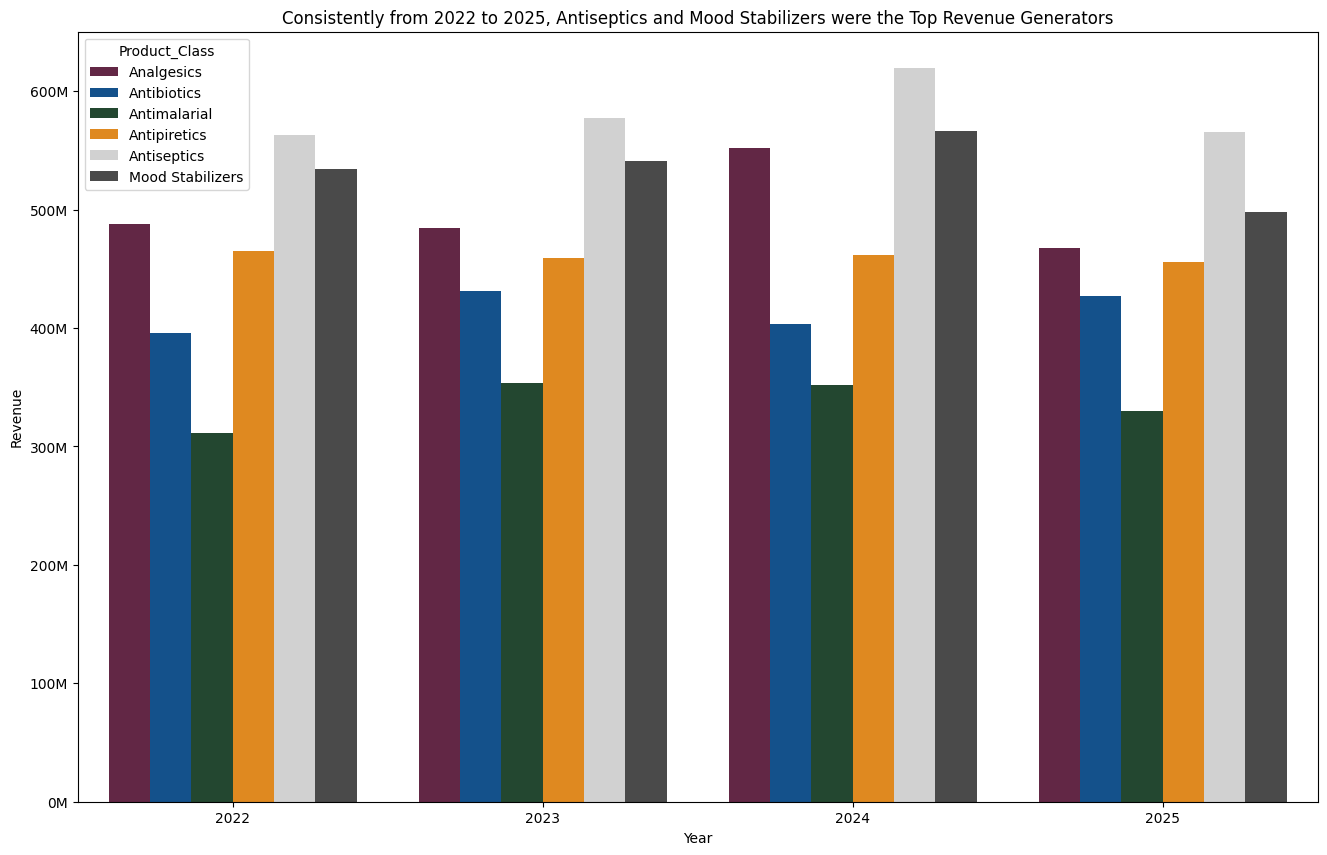
    return f'{x / 10\*\*6:.0f}M'

# 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.

1. 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):

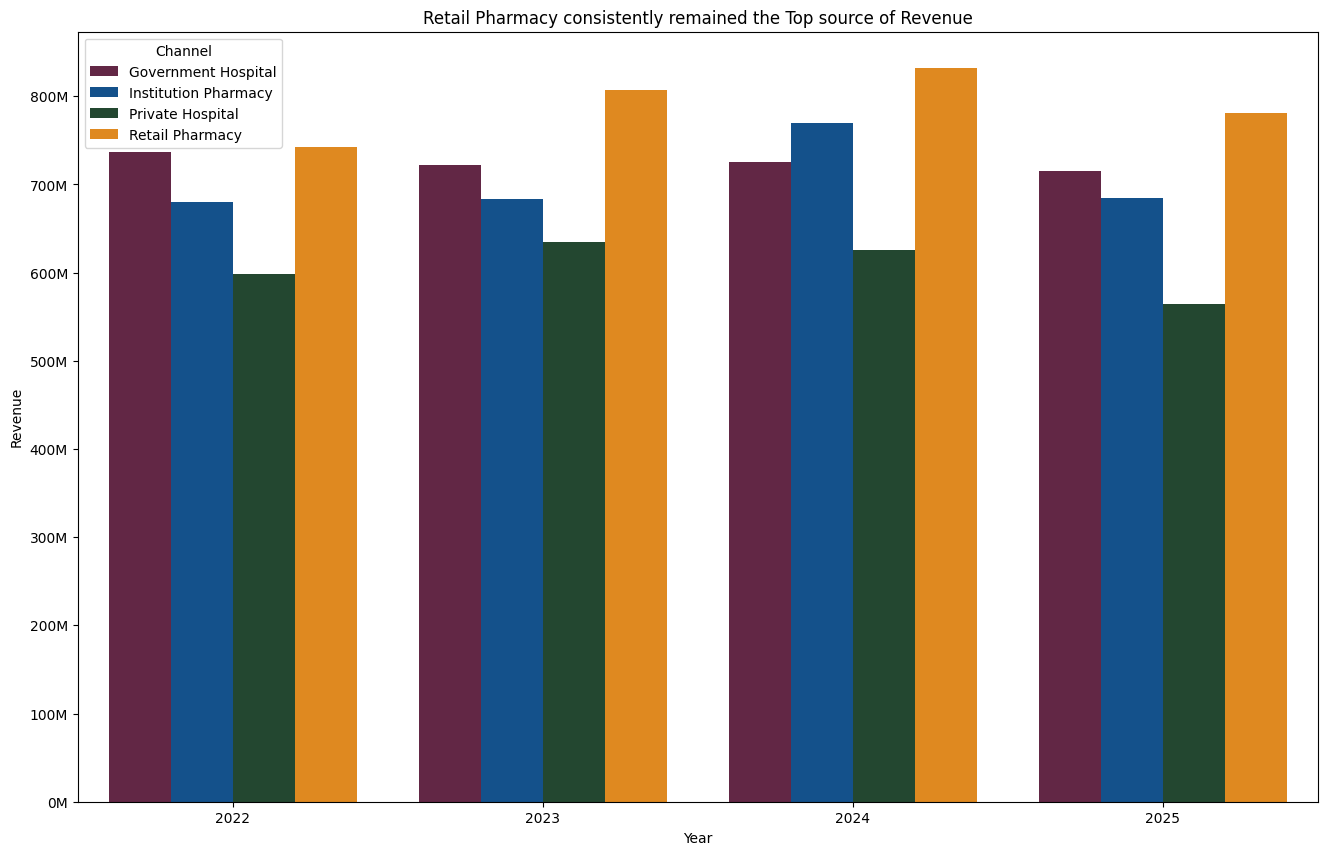
    return '{:.0f}M'.format(x / 10\*\*6)

# 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()

Forggith\_Targets.head(20)

1. 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))

1. 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))

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

1. 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))

**Recommendations**

1. **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.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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.
7. **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.