Buriro_Ezekia_Marketing_Data_Analysis

April 16, 2024

1 Marketing Data Analysis

[ ]:

```
# Rreads the data in pandas

import pandas as pd

# Read the data into a Pandas DataFrame
df = pd.read_excel('/DATA_CLEANED.xlsx')

# Print the DataFrame
print(df)
```

```
     Mkoa (Region) Bidhaa (Products)  Idadi Iliyouzwa (Total Sales)  \
0           Kagera            Mchele                            293
1           Kagera            Maziwa                            555
2           Kagera            Mchele                            579
3           Kagera            Mchele                            983
4           Kagera           Mahindi                           1118
…            …               …                            …
1290     Shinyanga            Maziwa                           2487
1291     Shinyanga            Mchele                           2750
1292     Shinyanga            Mchele                           2750
1293     Shinyanga            Mchele                           2750
1294     Shinyanga           Mahindi                           3422

      Gharama (Cost)  Mapato ( Net Income)  Faida (Profit)  Tarehe (Date)
0               2450                717850        128845.5          43831
1               1400                777000        139650.0          43831
2               2450               1418550        254971.5          43831
3               2450               2408350        433135.5          43831
4                500                559000        100545.0          43831
…              …                   …             …            …
1290            1400               3481800        626514.0          44166
1291            2450               6737500       1212382.5          44166
1292            2450               6737500       1212382.5          44166
1293            2450               6737500       1212382.5          44166
1294             500               1711000        307905.0          44166
```

```
[1295 rows x 7 columns]
```

[ ]:

```
df.columns
```

[ ]:

```
Index(['Mkoa (Region)', 'Bidhaa (Products)', 'Idadi Iliyouzwa (Total Sales)',
       'Gharama (Cost)', 'Mapato ( Net Income)', 'Faida (Profit)',
       'Tarehe (Date)'],
      dtype='object')
```

[ ]:

```
# Removes all duplicates in the data.

a = df.drop_duplicates(subset=['Mkoa (Region)', 'Bidhaa (Products)', 'Idadi␣
 ↪Iliyouzwa (Total Sales)',
       'Gharama (Cost)', 'Mapato ( Net Income)', 'Faida (Profit)',
       'Tarehe (Date)'], inplace=True)
print(a)
```

```
None
```

[111]:

```
# df['Bidhaa (Products)'].dtypes
```

[112]:

```
# df['Bidhaa (Products)'].head()
```

[113]:

```
# df['Bidhaa (Products)'] = df['Bidhaa (Products)'].apply(lambda x: pd.
 ↪to_numeric(x, errors='coerce'))
```

[114]:

```
# df['Bidhaa (Products)'] = pd.to_numeric(df['Bidhaa (Products)'])
```

[ ]:

```
# Display the sum and average of 'Gharama (Cost)', 'Idadi Iliyouzwa (Total␣
 ↪Sales)',
#          'Mapato ( Net Income)', separately

# Calculate the sum of 'Gharama (Cost)'
sum_cost = df['Gharama (Cost)'].sum()

# Calculate the sum of 'Idadi Iliyouzwa (Total Sales)'
sum_sales = df['Idadi Iliyouzwa (Total Sales)'].sum()

# Calculate the sum of 'Mapato ( Net Income)'
sum_income = df['Mapato ( Net Income)'].sum()

# Calculate the average of 'Gharama (Cost)'
avg_cost = df['Bidhaa (Products)'].mean()

# Calculate the average of 'Idadi Iliyouzwa (Total Sales)'
avg_sales = df['Idadi Iliyouzwa (Total Sales)'].mean()
```

```
# Calculate the average of 'Mapato ( Net Income)'
avg_income = df['Mapato ( Net Income)'].mean()

# Print the results
print("Sum of 'Gharama (Cost)' is :", sum_cost)
print("Sum of 'Idadi Iliyouzwa (Total Sales)' is:", sum_sales)
print("Sum of 'Mapato ( Net Income)' is:", sum_income)
print("Average of 'Gharama (Cost)' is:", avg_cost)
print("Average of 'Idadi Iliyouzwa (Total Sales)' is:", avg_sales)
print("Average of 'Mapato ( Net Income)' is:", avg_income)
```

```
Sum of 'Gharama (Cost)' is : 2354550
Sum of 'Idadi Iliyouzwa (Total Sales)' is: 1919474
Sum of 'Mapato ( Net Income)' is: 4065377250
Average of 'Gharama (Cost)' is: nan
Average of 'Idadi Iliyouzwa (Total Sales)' is: 1709.2377560106856
Average of 'Mapato ( Net Income)' is: 3620104.407836153
```

[ ]:

```
# Display a pivot table based on  'Idadi Iliyouzwa (Total Sales)',
#         'Gharama (Cost)', 'Mapato ( Net Income)', 'Faida (Profit)',
#         'Mkoa (Region)'

import numpy as np

df_pivot = pd.pivot_table(df,
       values=['Idadi Iliyouzwa (Total Sales)',
       'Gharama (Cost)', 'Mapato ( Net Income)', 'Faida (Profit)'],
       index=['Mkoa (Region)'],
       aggfunc=np.sum)
print(df_pivot)
```

```
               Faida (Profit)  Gharama (Cost)  Idadi Iliyouzwa (Total Sales)  \
Mkoa (Region)
Kagera             95877991.5          308750                         317881
Katavi            155842717.5          556350                         295779
Kigoma            166937467.5          527250                         322406
Mara              137826306.0          423000                         408914
Mwanza             91096105.5          282750                         251542
Shinyanga          83834134.5          256450                         322952

               Mapato ( Net Income)
Mkoa (Region)
Kagera                    532912800
Katavi                    866256500
Kigoma                    927869750
Mara                      766054200
Mwanza                    506325100
Shinyanga                 465958900
```

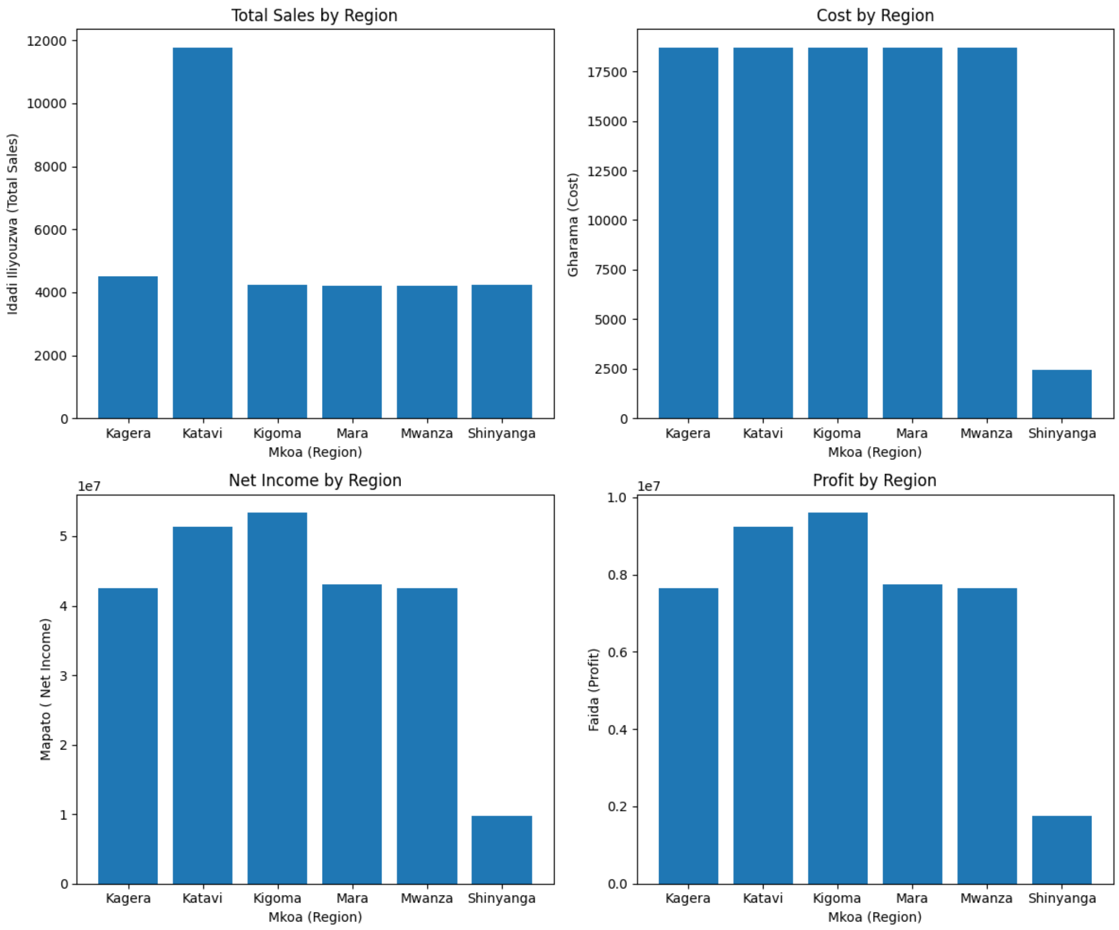
[106]:

```
# # Visualise 'Idadi Iliyouzwa (Total Sales)', 'Gharama (Cost)', 'Mapato ( Net␣
 ↪Income)', 'Faida (Profit)', 'Mkoa (Region)'.

# import matplotlib.pyplot as plt

# # Create a bar chart for 'Idadi Iliyouzwa (Total Sales)'
# plt.bar(df['Mkoa (Region)'], df['Idadi Iliyouzwa (Total Sales)'])
# plt.ylabel('Idadi Iliyouzwa (Total Sales)')
# plt.xlabel('Mkoa (Region)')
# plt.title('Total Sales by Region')
# plt.show()

# # Create a bar chart for 'Gharama (Cost)'
# plt.bar(df['Mkoa (Region)'], df['Gharama (Cost)'])
# plt.ylabel('Gharama (Cost)')
# plt.xlabel('Mkoa (Region)')
# plt.title('Cost by Region')
# plt.show()

# # Create a bar chart for 'Mapato ( Net Income)'
# plt.bar(df['Mkoa (Region)'], df['Mapato ( Net Income)'])
# plt.ylabel('Mapato ( Net Income)')
# plt.xlabel('Mkoa (Region)')
# plt.title('Net Income by Region')
# plt.show()

# # Create a bar chart for 'Faida (Profit)'
# plt.bar(df['Mkoa (Region)'], df['Faida (Profit)'])
# plt.ylabel('Faida (Profit)')
# plt.xlabel('Mkoa (Region)')
# plt.title('Profit by Region')
# plt.show()
```

[110]:

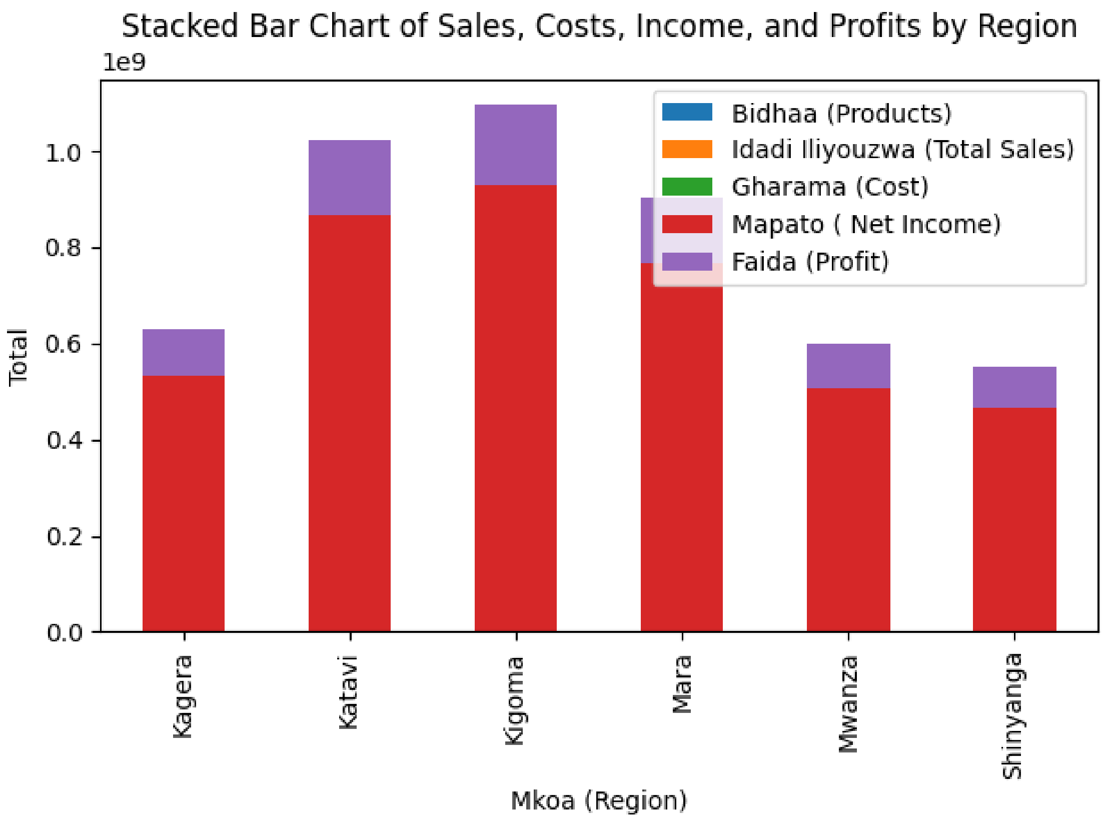
```
# Visualise 'Idadi Iliyouzwa (Total Sales)', 'Gharama (Cost)', 'Mapato ( Net␣
 ↪Income)', 'Faida (Profit)', 'Mkoa (Region)'.

import matplotlib.pyplot as plt

# Create a figure with 2 rows and 2 columns of subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

# Bar chart for 'Idadi Iliyouzwa (Total Sales)'
axes[0, 0].bar(df['Mkoa (Region)'], df['Idadi Iliyouzwa (Total Sales)'])
axes[0, 0].set_ylabel('Idadi Iliyouzwa (Total Sales)')
axes[0, 0].set_xlabel('Mkoa (Region)')
axes[0, 0].set_title('Total Sales by Region')
```

```python
# Bar chart for 'Gharama (Cost)'
axes[0, 1].bar(df['Mkoa (Region)'], df['Gharama (Cost)'])
axes[0, 1].set_ylabel('Gharama (Cost)')
axes[0, 1].set_xlabel('Mkoa (Region)')
axes[0, 1].set_title('Cost by Region')

# Bar chart for 'Mapato ( Net Income)'
axes[1, 0].bar(df['Mkoa (Region)'], df['Mapato ( Net Income)'])
axes[1, 0].set_ylabel('Mapato ( Net Income)')
axes[1, 0].set_xlabel('Mkoa (Region)')
axes[1, 0].set_title('Net Income by Region')

# Bar chart for 'Faida (Profit)'
axes[1, 1].bar(df['Mkoa (Region)'], df['Faida (Profit)'])
axes[1, 1].set_ylabel('Faida (Profit)')
axes[1, 1].set_xlabel('Mkoa (Region)')
axes[1, 1].set_title('Profit by Region')

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```



[ ]:

```
# Stacked Bar Charts: This shows the composition of total sales, costs, net income, and profits for each region or product.
# Create a stacked bar chart for 'Idadi Iliyouzwa (Total Sales)', 'Gharama (Cost)', 'Mapato ( Net Income)', 'Faida (Profit)', 'Mkoa (Region)'.

# Exclude 'Tarehe (Date)' column before grouping
df_stacked = df.drop(columns=['Tarehe (Date)']).groupby('Mkoa (Region)').sum()

# Create the stacked bar chart
df_stacked.plot(kind='bar', stacked=True)
plt.ylabel('Total')
plt.xlabel('Mkoa (Region)')
plt.title('Stacked Bar Chart of Sales, Costs, Income, and Profits by Region')
# Show plot
plt.tight_layout()  # Adjust layout to prevent clipping of labels
plt.show()
```



[ ]:

```
# # Stacked Bar Charts shows the composition of total sales, costs, net income,␣
↪and profits for each region or product
# Each bar represents the total, and segments within the bar represent the␣
↪different components (sales, costs, income, profit).

# # Create a stacked bar chart for sales, costs, income, and profits by region
# df_pivot_stacked = df_pivot.stack().unstack()
# df_pivot_stacked.plot(kind='bar', stacked=True)
# plt.ylabel('Total')
# plt.xlabel('Region')
# plt.title('Sales, Costs, Income, and Profits by Region')
# plt.show()

# # Create a stacked bar chart for sales, costs, income, and profits by product
# df_pivot_product = pd.pivot_table(df,
#         values=['Idadi Iliyouzwa (Total Sales)',
#         'Gharama (Cost)', 'Mapato ( Net Income)', 'Faida (Profit)'],
#         index=['Mkoa (Region)'],
#         aggfunc=np.sum)
# df_pivot_product_stacked = df_pivot_product.stack().unstack()
# df_pivot_product_stacked.plot(kind='bar', stacked=True)
# plt.ylabel('Total')
# plt.xlabel('Product')
# plt.title('Sales, Costs, Income, and Profits by Product')
# plt.tight_layout()  # Adjust layout to prevent clipping of labels
# plt.show()
```

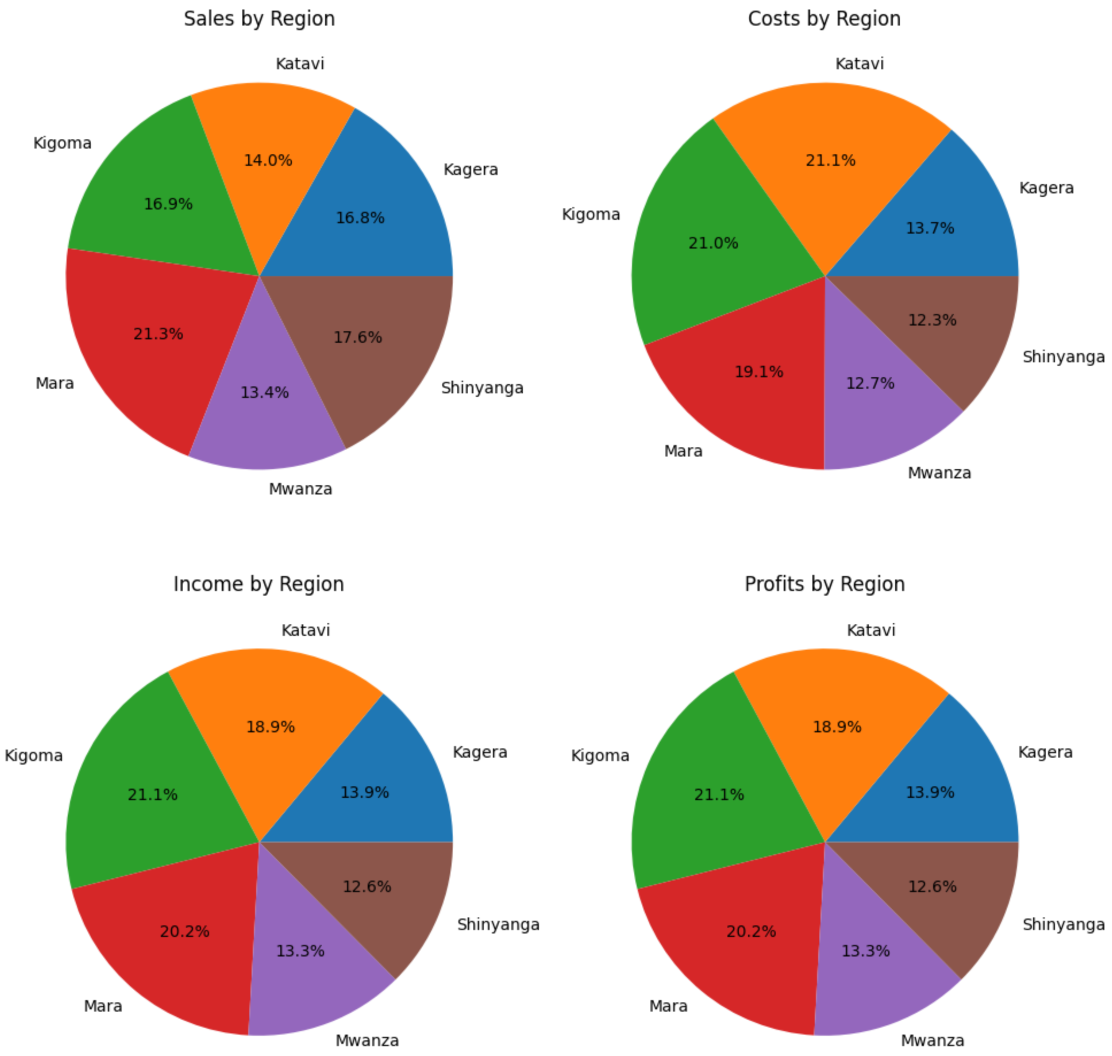
[ ]:

```
# # Pie charts can be used to show the contribution of each region or product␣
 ↪to the total sales, costs, income, or profits.

# # Create a pie chart for sales by region
# labels = df['Mkoa (Region)'].unique()
# sizes = df.groupby('Mkoa (Region)')['Idadi Iliyouzwa (Total Sales)'].sum()
# plt.pie(sizes, labels=labels, autopct='%1.1f%%')
# plt.title('Sales by Region')
# plt.show()

# # Create a pie chart for costs by region
# labels = df['Mkoa (Region)'].unique()
# sizes = df.groupby('Mkoa (Region)')['Gharama (Cost)'].sum()
# plt.pie(sizes, labels=labels, autopct='%1.1f%%')
# plt.title('Costs by Region')
# plt.show()

# # Create a pie chart for income by region
# labels = df['Mkoa (Region)'].unique()
# sizes = df.groupby('Mkoa (Region)')['Mapato ( Net Income)'].sum()
# plt.pie(sizes, labels=labels, autopct='%1.1f%%')
# plt.title('Income by Region')
# plt.show()

# # Create a pie chart for profits by region
# labels = df['Mkoa (Region)'].unique()
# sizes = df.groupby('Mkoa (Region)')['Faida (Profit)'].sum()
# plt.pie(sizes, labels=labels, autopct='%1.1f%%')
# plt.title('Profits by Region')
# plt.show()
```

[116]:

```
# Pie charts can be used to show the contribution of each region or product to␣
 ↪the total sales, costs, income, or profits.

import matplotlib.pyplot as plt

# Create a figure with 2 rows and 2 columns of subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 10))

# Pie chart for sales by region
labels = df['Mkoa (Region)'].unique()
```

```
sizes = df.groupby('Mkoa (Region)')['Idadi Iliyouzwa (Total Sales)'].sum()
axes[0, 0].pie(sizes, labels=labels, autopct='%1.1f%%')
axes[0, 0].set_title('Sales by Region')

# Pie chart for costs by region
sizes = df.groupby('Mkoa (Region)')['Gharama (Cost)'].sum()
axes[0, 1].pie(sizes, labels=labels, autopct='%1.1f%%')
axes[0, 1].set_title('Costs by Region')

# Pie chart for income by region
sizes = df.groupby('Mkoa (Region)')['Mapato ( Net Income)'].sum()
axes[1, 0].pie(sizes, labels=labels, autopct='%1.1f%%')
axes[1, 0].set_title('Income by Region')

# Pie chart for profits by region
sizes = df.groupby('Mkoa (Region)')['Faida (Profit)'].sum()
axes[1, 1].pie(sizes, labels=labels, autopct='%1.1f%%')
axes[1, 1].set_title('Profits by Region')

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```



[ ]:

```

```

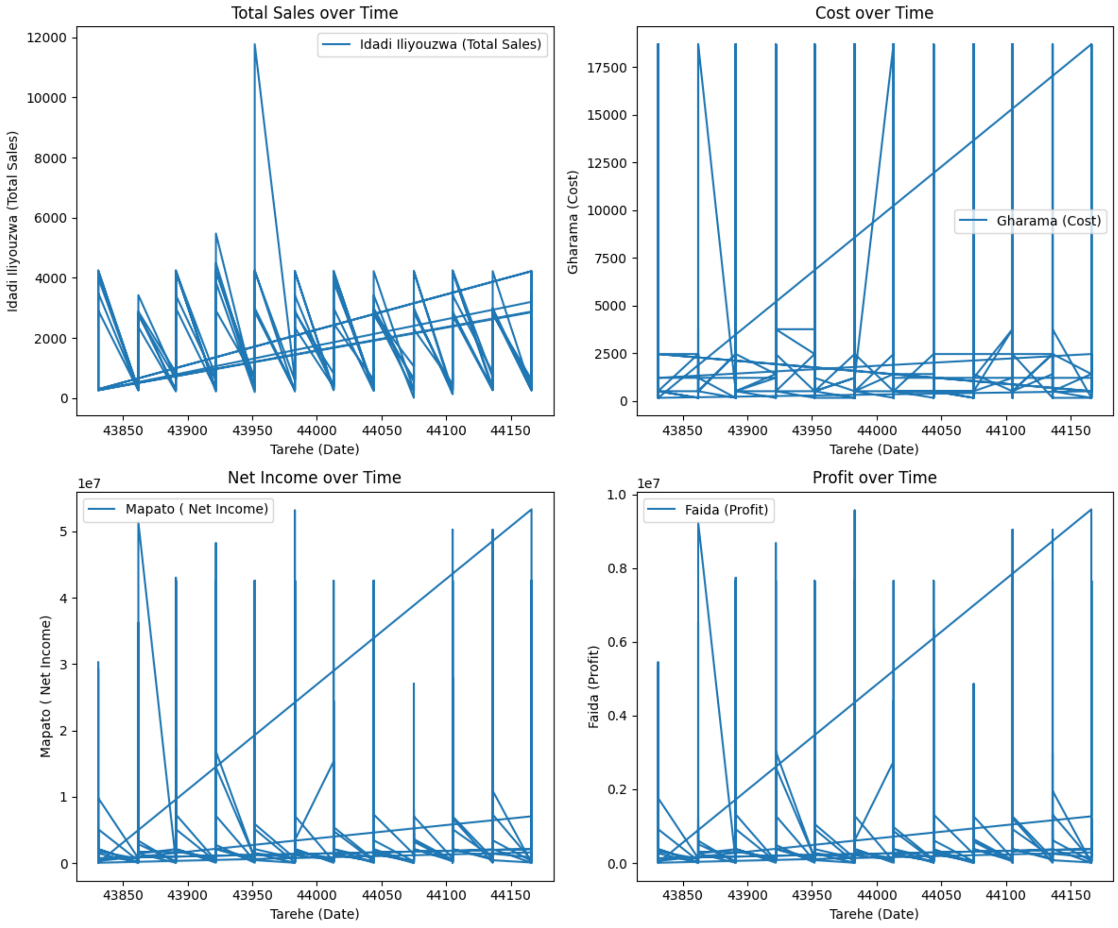
[ ]:

```
# # Line Charts: Line charts is helpful for visualizing trends over time.
# # 'Tarehe (Date)' column is useful for understanding how sales, costs,␣
 ↪income, and profits change over time.

# # Line Chart: This shows how sales, costs, income, and profits change over␣
 ↪time.

# # Create a line chart for 'Idadi Iliyouzwa (Total Sales)'
# df.plot(x='Tarehe (Date)', y='Idadi Iliyouzwa (Total Sales)', kind='line')
# plt.ylabel('Idadi Iliyouzwa (Total Sales)')
# plt.xlabel('Tarehe (Date)')
# plt.title('Total Sales over Time')
# plt.show()

# # Create a line chart for 'Gharama (Cost)'
# df.plot(x='Tarehe (Date)', y='Gharama (Cost)', kind='line')
# plt.ylabel('Gharama (Cost)')
# plt.xlabel('Tarehe (Date)')
# plt.title('Cost over Time')
# plt.show()

# # Create a line chart for 'Mapato ( Net Income)'
# df.plot(x='Tarehe (Date)', y='Mapato ( Net Income)', kind='line')
# plt.ylabel('Mapato ( Net Income)')
# plt.xlabel('Tarehe (Date)')
# plt.title('Net Income over Time')
# plt.show()

# # Create a line chart for 'Faida (Profit)'
# df.plot(x='Tarehe (Date)', y='Faida (Profit)', kind='line')
# plt.ylabel('Faida (Profit)')
# plt.xlabel('Tarehe (Date)')
# plt.title('Profit over Time')
# plt.show()
```

[ ]:

```
# Line Charts: Line charts is helpful for visualizing trends over time.
# # 'Tarehe (Date)' column is useful for understanding how sales, costs,␣
 ↪income, and profits change over time.

import matplotlib.pyplot as plt

# Create a figure with 2 rows and 2 columns of subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

# Line chart for 'Idadi Iliyouzwa (Total Sales)'
df.plot(x='Tarehe (Date)', y='Idadi Iliyouzwa (Total Sales)', kind='line',␣
 ↪ax=axes[0, 0])
axes[0, 0].set_ylabel('Idadi Iliyouzwa (Total Sales)')
axes[0, 0].set_xlabel('Tarehe (Date)')
axes[0, 0].set_title('Total Sales over Time')

# Line chart for 'Gharama (Cost)'
df.plot(x='Tarehe (Date)', y='Gharama (Cost)', kind='line', ax=axes[0, 1])
axes[0, 1].set_ylabel('Gharama (Cost)')
axes[0, 1].set_xlabel('Tarehe (Date)')
axes[0, 1].set_title('Cost over Time')

# Line chart for 'Mapato ( Net Income)'
```

```
df.plot(x='Tarehe (Date)', y='Mapato ( Net Income)', kind='line', ax=axes[1, 0])
axes[1, 0].set_ylabel('Mapato ( Net Income)')
axes[1, 0].set_xlabel('Tarehe (Date)')
axes[1, 0].set_title('Net Income over Time')

# Line chart for 'Faida (Profit)'
df.plot(x='Tarehe (Date)', y='Faida (Profit)', kind='line', ax=axes[1, 1])
axes[1, 1].set_ylabel('Faida (Profit)')
axes[1, 1].set_xlabel('Tarehe (Date)')
axes[1, 1].set_title('Profit over Time')

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```



Total Sales, Net Income, and Profit show a monotonic increasing trend over time.

Cost also behaves the same over time but at a slower rate than total sales, net income, and profit.

Consequently, the graph suggests that the company is experiencing positive growth in sales, net income, and profit.

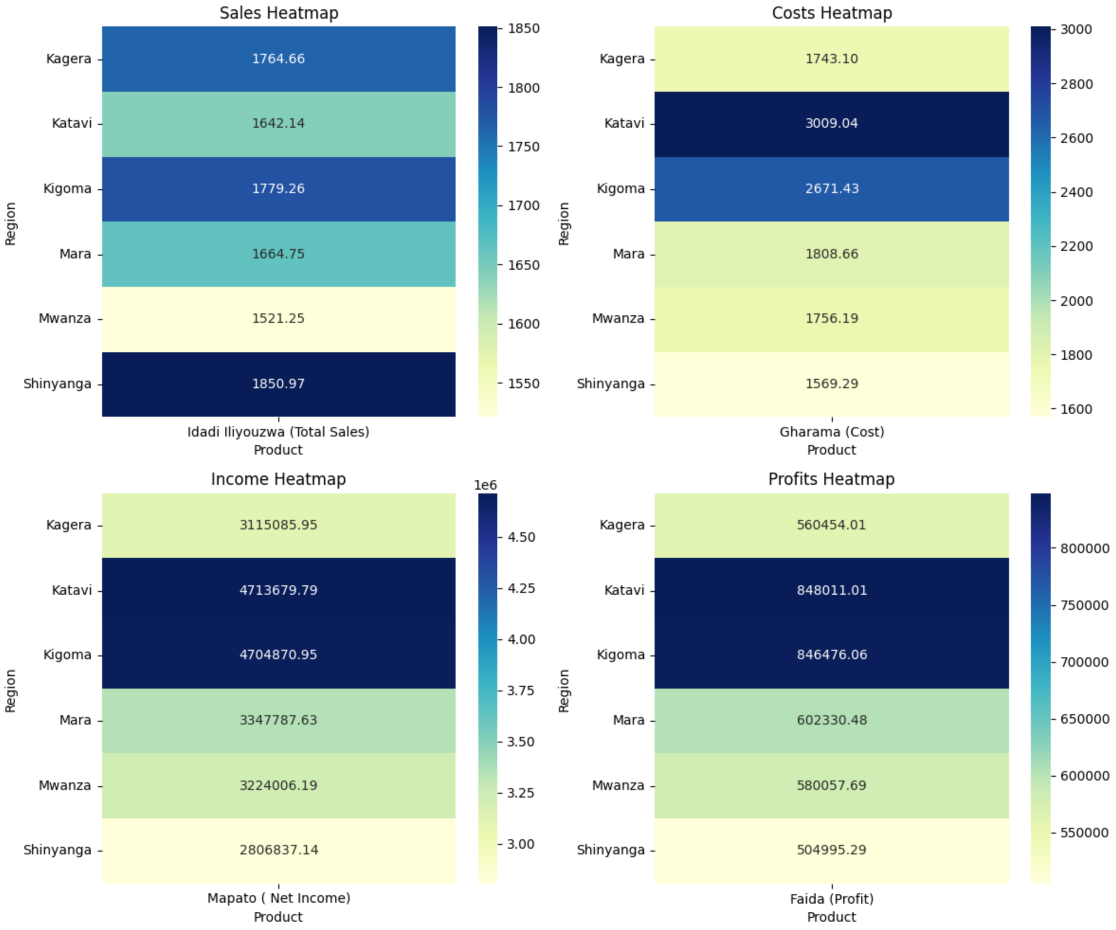
[108]:

```
# # Heatmaps provide a visual representation of the relationship between␣
 ↪regions and products in terms of sales, costs, income, and profits.
# # Each cell in the heatmap represents a combination of region and product,␣
 ↪with color intensity indicating the magnitude of the metric.

# import seaborn as sns

# # Create a heatmap of sales by region and product
# heatmap_sales = df.pivot_table(values='Idadi Iliyouzwa (Total Sales)',␣
 ↪index='Mkoa (Region)') #, columns='Bidhaa (Products)')
# sns.heatmap(heatmap_sales, annot=True, fmt='.2f', cmap='YlGnBu')
# plt.ylabel('Region')
# plt.xlabel('Product')
# plt.title('Sales Heatmap')
# plt.show()

# # Create a heatmap of costs by region and product
# heatmap_costs = df.pivot_table(values='Gharama (Cost)', index='Mkoa (Region)')
# sns.heatmap(heatmap_costs, annot=True, fmt='.2f', cmap='YlGnBu')
# plt.ylabel('Region')
# plt.xlabel('Product')
# plt.title('Costs Heatmap')
# plt.show()

# # Create a heatmap of income by region and product
# heatmap_income = df.pivot_table(values='Mapato ( Net Income)', index='Mkoa␣
 ↪(Region)')
# sns.heatmap(heatmap_income, annot=True, fmt='.2f', cmap='YlGnBu')
# plt.ylabel('Region')
# plt.xlabel('Product')
# plt.title('Income Heatmap')
# plt.show()

# # Create a heatmap of profits by region and product
# heatmap_profits = df.pivot_table(values='Faida (Profit)', index='Mkoa␣
 ↪(Region)')
# sns.heatmap(heatmap_profits, annot=True, fmt='.2f', cmap='YlGnBu')
# plt.ylabel('Region')
# plt.xlabel('Product')
# plt.title('Profits Heatmap')
# plt.show()
```

[107]:

```
# Heatmaps provide a visual representation of the relationship between regions␣
 ↪and products in terms of sales, costs, income, and profits.
# # Each cell in the heatmap represents a combination of region and product,␣
 ↪with color intensity indicating the magnitude of the metric.



import seaborn as sns
import matplotlib.pyplot as plt

# Create a figure with 2 rows and 2 columns of subplots
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 10))

# Heatmap of sales by region and product
heatmap_sales = df.pivot_table(values='Idadi Iliyouzwa (Total Sales)',␣
 ↪index='Mkoa (Region)')
sns.heatmap(heatmap_sales, annot=True, fmt='.2f', cmap='YlGnBu', ax=axes[0, 0])
axes[0, 0].set_ylabel('Region')
axes[0, 0].set_xlabel('Product')
axes[0, 0].set_title('Sales Heatmap')

# Heatmap of costs by region and product
heatmap_costs = df.pivot_table(values='Gharama (Cost)', index='Mkoa (Region)')
sns.heatmap(heatmap_costs, annot=True, fmt='.2f', cmap='YlGnBu', ax=axes[0, 1])
axes[0, 1].set_ylabel('Region')
axes[0, 1].set_xlabel('Product')
axes[0, 1].set_title('Costs Heatmap')

# Heatmap of income by region and product
heatmap_income = df.pivot_table(values='Mapato ( Net Income)', index='Mkoa␣
 ↪(Region)')
sns.heatmap(heatmap_income, annot=True, fmt='.2f', cmap='YlGnBu', ax=axes[1, 0])
axes[1, 0].set_ylabel('Region')
axes[1, 0].set_xlabel('Product')
axes[1, 0].set_title('Income Heatmap')

# Heatmap of profits by region and product
heatmap_profits = df.pivot_table(values='Faida (Profit)', index='Mkoa (Region)')
sns.heatmap(heatmap_profits, annot=True, fmt='.2f', cmap='YlGnBu', ax=axes[1,␣
 ↪1])
axes[1, 1].set_ylabel('Region')
axes[1, 1].set_xlabel('Product')
axes[1, 1].set_title('Profits Heatmap')

# Adjust layout
plt.tight_layout()

# Show plot
```

```
plt.show()
```



Sales Heatmap: This heatmap shows that sales are highest in Shinyanga at TZS 1,866.77 and lowest in Mwanza at TZS 1,543.20.

Costs Heatmap: This heatmap shows that costs are highest in Katavi at TZS 3,073.76 and lowest in Shinyanga at TZS 1,482.37.

Income Heatmap: This heatmap shows that income (presumably net sales after costs) is highest in Kigoma at TZS 5,126,352.21 and lowest in Shinyanga at TZS 2,693,404.40.

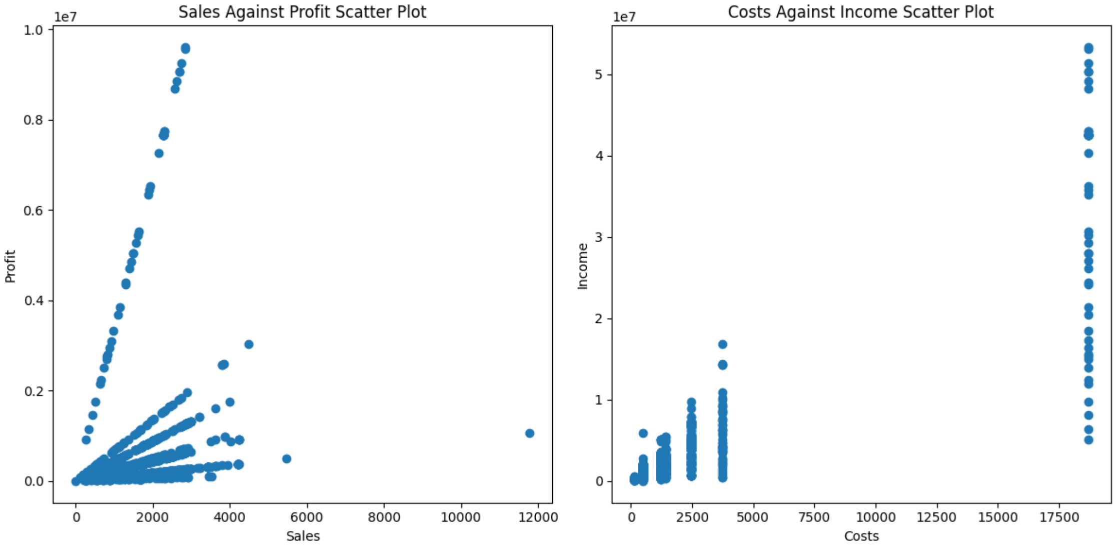
Profits Heatmap: This heatmap shows that profits are highest in Kigoma at TZS 922,306.45 and lowest in Shinyanga at TZS 484,590.37.

Importantly, there seems to be an inconsistency in the data for Shinyanga. Sales are high but costs and profits are the lowest.

```
[ ]: # # Scatter Plots explore the relationships between different metrics (e.g., sales vs. profit)
# # scatter plots can help identify correlations or patterns.

# # Create a scatter plot of sales vs. profit
# plt.scatter(df['Idadi Iliyouzwa (Total Sales)'], df['Faida (Profit)'])
# plt.xlabel('Sales')
# plt.ylabel('Profit')
# plt.title('Sales Against Profit Scatter Plot')
# plt.show()

# # Create a scatter plot of costs vs. income
# plt.scatter(df['Gharama (Cost)'], df['Mapato ( Net Income)'])
# plt.xlabel('Costs')
# plt.ylabel('Income')
# plt.title('Costs gainst Income Scatter Plot')
# plt.show()
```

[109]:

```
# Scatter Plots explore the relationships between different metrics (e.g., sales vs. profit)
# # scatter plots can help identify correlations or patterns.




import matplotlib.pyplot as plt

# Create a figure with 1 row and 2 columns of subplots
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 6))

# Scatter plot of sales vs. profit
axes[0].scatter(df['Idadi Iliyouzwa (Total Sales)'], df['Faida (Profit)'])
axes[0].set_xlabel('Sales')
axes[0].set_ylabel('Profit')
axes[0].set_title('Sales Against Profit Scatter Plot')

# Scatter plot of costs vs. income
axes[1].scatter(df['Gharama (Cost)'], df['Mapato ( Net Income)'])
axes[1].set_xlabel('Costs')
axes[1].set_ylabel('Income')
axes[1].set_title('Costs Against Income Scatter Plot')

# Adjust layout
plt.tight_layout()

# Show plot
plt.show()
```



The first figure shows a positive correlation between sales and profit, which implies that sales increase with profit. However, there is also variability in the data, so it is impossible to say that an increase in sales will always lead to a profit increase.

Moreover, there are a few data points that are outliers. For example, one data point has a very high sales value but a relatively low-profit value. That suggests that this company may have sold lots of products at a low price point or had high costs associated with selling those products.