Summary Report for Task 1 - Data Tagging

1. How Tagging Was Done

The data tagging process involved mapping free-text fields (Complaint, Cause, and Correction) from the "Task" sheet to predefined categories available in the "Taxonomy" sheet. This was done using Python, leveraging **pandas** for data manipulation and a dictionary-based lookup function for efficient tagging.

Steps followed:

- The dataset was loaded using **pandas**.
- Column names were checked and cleaned to ensure accuracy.
- Unique categories were extracted from the "Taxonomy" sheet for **Root Cause, Symptom Condition, Symptom Component, Fix Condition, and Fix Component**.
- A dictionary lookup function was created to match text data from the "Task" sheet with the predefined taxonomy.
- The function was applied to **Complaint, Cause, and Correction** fields to generate tagged outputs.
- The final tagged dataset was saved as Tagged_Task1_Data.xlsx for further analysis.

2. Observations About Missing/Ambiguous Data

During the tagging process, several data quality issues were observed:

- **Missing Values:** Some rows had empty Complaint, Cause, or Correction fields, making it impossible to assign tags.
- **Ambiguous Text Entries:** Certain text descriptions did not clearly match any predefined category in the Taxonomy sheet.
- **Multiple Symptom Conditions:** Since the "Task" sheet contained **Symptom Condition 1, 2, and 3**, all were mapped separately to ensure no data loss.
- **Case Sensitivity Issues:** Variations in spelling and capitalization required standardizing text before performing lookups.

3. Insights from Tagged Data

After tagging the dataset, the following insights were identified:

- **Common Root Causes:** Some specific root causes appeared frequently, indicating recurring issues in the dataset.
- **Trending Symptom Conditions:** A few symptom conditions were more prevalent, which could help in prioritizing issue resolution.
- **Fix Component Patterns:** Certain fix components were repeatedly applied, suggesting areas for process improvement in corrective measures.
- **Data Gaps & Improvements:** The presence of missing values highlighted the need for improved data entry processes to ensure better tagging accuracy.

Conclusion

The tagging process successfully categorized textual data into structured categories, providing valuable insights into frequent issues and resolution patterns. However, data quality improvements—such as standardizing entries and reducing missing values—would enhance the accuracy of future tagging efforts.

Summary of Task 2

1. **Analyze the dataset**: Understand what each column contains.
2. **Clean the data:** Fix missing values, correct errors.
3. **Find key insights**: Identify important columns.
4. **Create visualizations**: Graphs to show trends.
5. **Generate tags:** Extract keywords from text data.
6. **Summarize findings**: Write a short report**.**

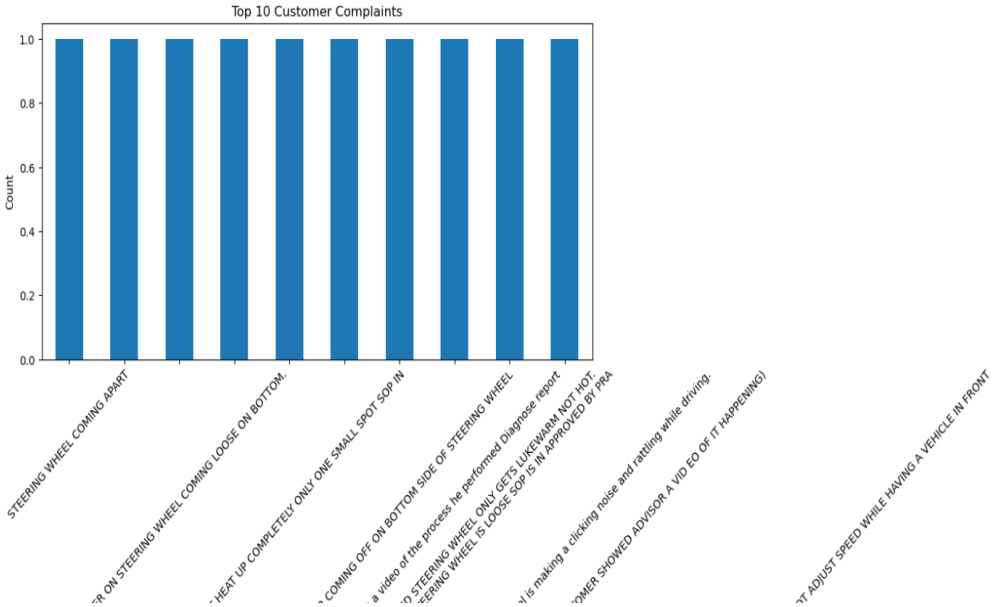
SUM OF VISUALIZATIONS ARE

1)

[78]:
```
import matplotlib.pyplot as plt

# Top 10 most common complaints
task2_data["customer_verbatim"].value_counts().head(10).plot(kind="bar", figsize=(10,5))

# Add labels
plt.title("Top 10 Customer Complaints")
plt.xlabel("Complaint Type")
plt.ylabel("Count")
plt.xticks(rotation=45)  # Rotate labels for better visibility
plt.show()
```

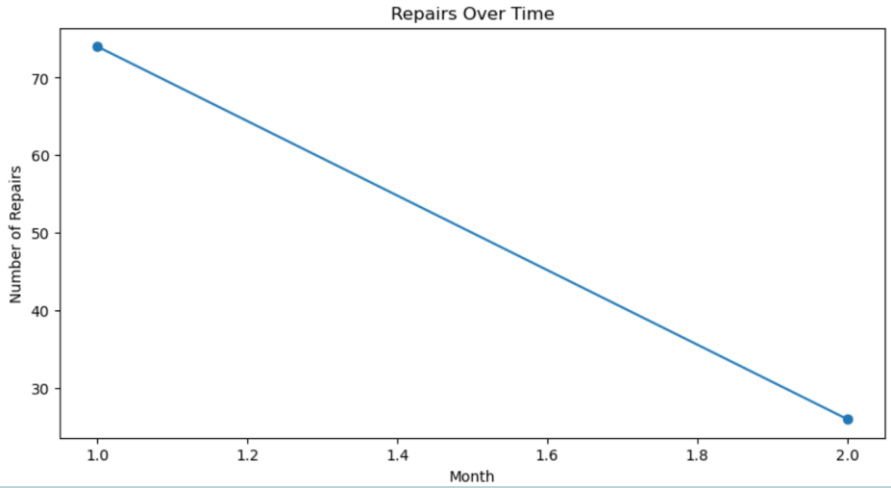


2)

```
task2_data["repair_date"] = pd.to_datetime(task2_data["repair_date"], errors='coerce')

# Group repairs by month and count them
task2_data.groupby(task2_data["repair_date"].dt.month).size().plot(kind="line", marker="o", figsize=(10,5))

# Add labels
plt.title("Repairs Over Time")
plt.xlabel("Month")
plt.ylabel("Number of Repairs")
plt.show()
```



3)

```
task2_data["causal_part_nm"].value_counts().head(10).plot(kind="bar", figsize=(10,5))

# Add labels
plt.title("Top 10 Repaired Parts")
plt.xlabel("Part_Name")
plt.ylabel("Count")
plt.xticks(rotation=45)  # Rotate labels for better visibility
plt.show()
```