Case Study: Data Cleaning in E-Commerce Customer Analytics -- -- 10 Marks

Submitted By-Kajal singh ,UID-24MCI10253

Problem statement

An e-commerce company, XYZ, aims to enhance its customer experience and improve sales through data-driven insights. The company collects vast amounts of data from customer transactions, website interactions, and feedback surveys. However, as the data grows, inconsistencies, duplicates, and errors have emerged, leading to unreliable analytics.

Take dataset from any source, clean it by applying different data cleaning techniques whatever required into that along with the explanation:

GitHub Link for this Case study:
GitHub Link for this Case study:
https://github.com/mannatmahajan5/customers_casestudy32/blob/main/MannatCASE_STUDY.ipynb

Rubrics for evaluation:

1. **Relevant dataset _ 2 marks**

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

#1.Relevant dataset:
df=pd.read_csv('/content/customers-100.csv')
display(df.head())
# Display the first few rows of the dataframe
print(df.head())

# Check the shape of the dataframe
print(df.shape)

# Get summary statistics
print(df.describe())
```

	Index	Customer Id	First Name	Last Name	Company	City	Country	Phone 1	Phone 2	Email	Subscription Date	Website
0	1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard	Chile	229.077.5154	397.884.0519x718	zunigavanessa@smith.info	2020-08-24	http://www.stephenson.com/
1	2	1Ef7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimmychester	Djibouti	5153435776	686-620-1820x944	vmata@colon.com	2021-04-23	http://www.hobbs.com/
2	3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough	Antigua and Barbuda	+1-539-402-0259	(496)978-3969x58947	beckycarr@hogan.com	2020-03-25	http://www.lawrence.com/
3	4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview	Dominican Republic	001-808-617-6467x12895	+1-813-324-8756	stanleyblackwell@benson.org	2020-06-02	http://www.good-lyons.com/
4	5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Priscilla	Slovakia (Slovak Republic)	001-234-203-0635x76146	001-199-446-3860x3486	colinalvarado@miles.net	2021-04-17	https://goodwin-ingram.com/

```
   Index      Customer Id First Name Last Name  \
0      1  DD37Cf93aecA6Dc     Sheryl    Baxter
1      2  1Ef7b82A4CAAD10    Preston    Lozano
2      3  6F94879bDAfE5a6        Roy     Berry
3      4  5Cef8BFA16c5e3c      Linda     Olsen
4      5  053d585Ab6b3159     Joanna    Bender

                         Company               City  \
0                 Rasmussen Group       East Leonard
1                    Vega-Gentry  East Jimmychester
2                   Murillo-Perry      Isabelborough
3  Dominguez, Mcmillan and Donovan         Bensonview
4         Martin, Lang and Andrade     West Priscilla

                     Country                Phone 1                Phone 2  \
0                      Chile            229.077.5154       397.884.0519x718
1                   Djibouti              5153435776       686-620-1820x944
2         Antigua and Barbuda         +1-539-402-0259    (496)978-3969x58947
3          Dominican Republic  001-808-617-6467x12895        +1-813-324-8756
4  Slovakia (Slovak Republic)  001-234-203-0635x76146  001-199-446-3860x3486

                        Email Subscription Date                    Website
0     zunigavanessa@smith.info        2020-08-24   http://www.stephenson.com/
1             vmata@colon.com        2021-04-23        http://www.hobbs.com/
2          beckycarr@hogan.com        2020-03-25     http://www.lawrence.com/
3  stanleyblackwell@benson.org        2020-06-02   http://www.good-lyons.com/
4      colinalvarado@miles.net        2021-04-17  https://goodwin-ingram.com/
(100, 12)
            Index
count  100.000000
mean    50.500000
std     29.011492
min      1.000000
25%     25.750000
50%     50.500000
75%     75.250000
max    100.000000
```

2. Novel dataset - 2 marks

```
#Novel dataset
#Explore basic statistics and data types
print(f"Overview of data: /n{df.info()}")
print(f"Summary statistics: /n{df.describe()}")
print(f"Data types: /n{df.dtypes}")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Index              100 non-null    int64
 1   Customer Id        100 non-null    object
 2   First Name         100 non-null    object
 3   Last Name          100 non-null    object
 4   Company            100 non-null    object
 5   City               100 non-null    object
 6   Country            100 non-null    object
 7   Phone 1            100 non-null    object
 8   Phone 2            100 non-null    object
 9   Email              100 non-null    object
 10  Subscription Date  100 non-null    object
 11  Website            100 non-null    object
dtypes: int64(1), object(11)
memory usage: 9.5+ KB
Overview of data: /nNone
Summary statistics: /n            Index
count  100.000000
mean    50.500000
std     29.011492
min      1.000000
25%     25.750000
50%     50.500000
75%     75.250000
max    100.000000
Data types: /nIndex                  int64
Customer Id          object
First Name           object
Last Name            object
Company              object
City                 object
Country              object
Phone 1              object
Phone 2              object
Email                object
Subscription Date    object
Website              object
dtype: object
```

3. Data cleaning techniques along with the relevant explanation that why these techniques are being applied. -- 3 marks

a) Identifying missing values: The complete Dataset is checked if there is the presence of any null value or not.

b) Checking for the duplicate rows: The dataset is checked for any kind of duplicate row present in the dataset.

c) Replaced Missing values with a central tendency, i.e., Median: One of the suitable method to handle Missing value is to replace it with the central tendencies like mean, median, mode, or standard deviation.

d) Detecting the outliers: Outliers refers to value out of range with respect to the dataset available. We have detected outliers based on their IQR (Inter quartile Range).

e) Replacing the outliers with Median: Outliers detected by the IQR are replaced with the central tendency in order to clean the dataset.

```
#3.Data cleaning in the dataset
#identifying Missing Values in the dataset
print(f"checking for any missing values present:/n{df.isnull().sum()}")
#checking for duplicate rows
print(f"checking for any duplicate rows present:/n{df.duplicated().sum()}")

#remove duplicates
# Remove duplicate rows
df = df.drop_duplicates()

#saved the clean data
df.to_csv('cleaned_customer_100.csv', index=False)
```

```
checking for any missing values present:/nIndex
Customer Id          0
First Name           0
Last Name            0
Company              0
City                 0
Country              0
Phone 1              0
Phone 2              0
Email                0
Subscription Date    0
Website              0
dtype: int64
checking for any duplicate rows present:/n0
```

4. Relevant graphs and tables depicting clean data –3 marks

Relevant graph plotting using matplot library helps the users to properly visualize the datasets with help of different types of graphs showing relations between the different columns.

- Firstly I showed the Cleaned data which was cleaned above using Different techniques.

- Now, I ploted different types of graphs with the clean data present for proper understanding of different patterns of the dataset.

Different types of plots:

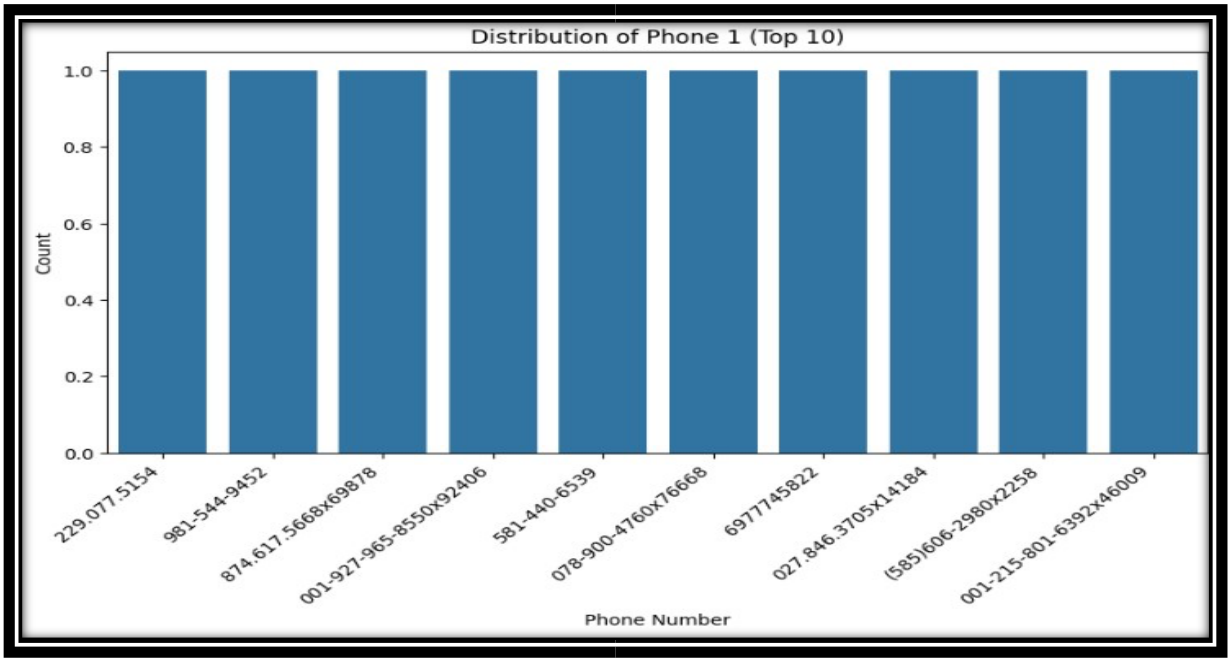
a) Distribution of phone 1 take 10 entries.

b) Relationship between country and company scatter plot take upto only 10 entries.

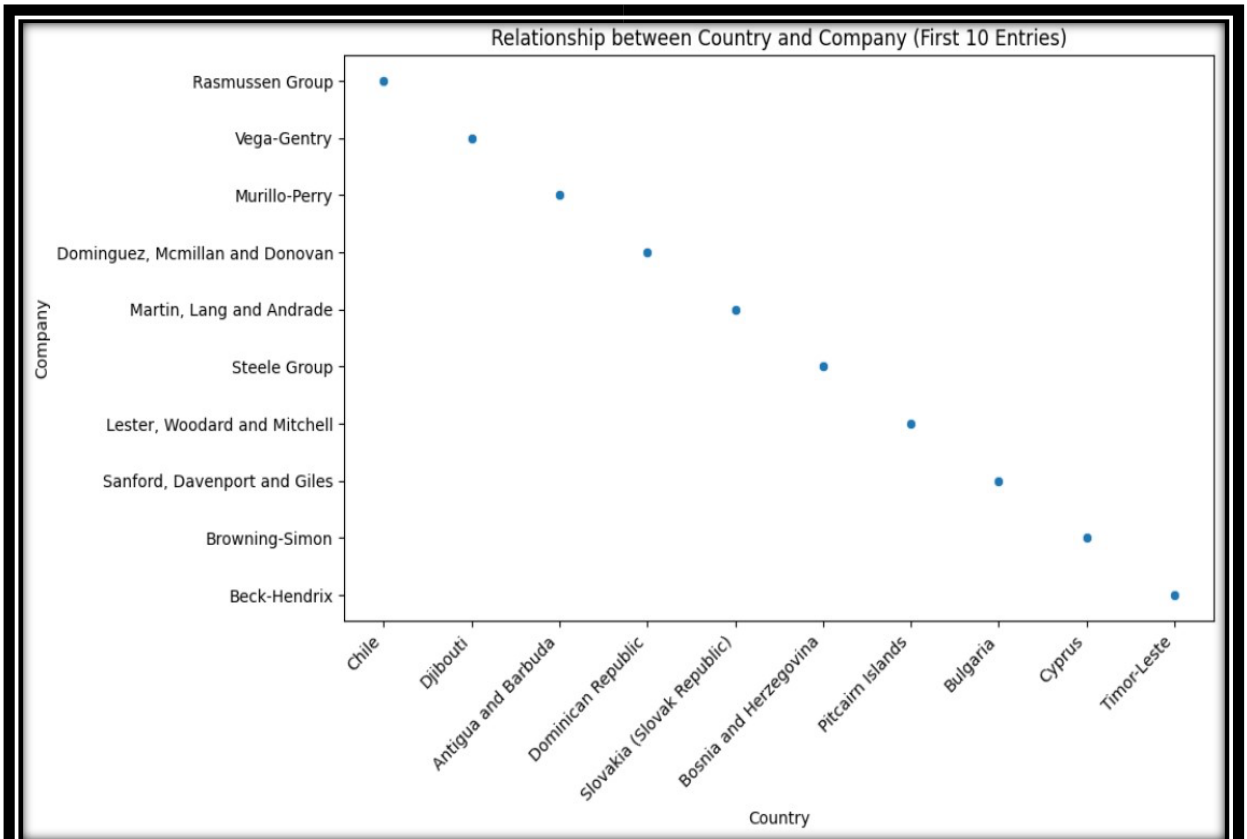
c) Function to create different types of dataset.

d) Top 10 company distribution.

```
# prompt: graph for distribution of phone 1 take 10 entries

phone_counts = df['Phone 1'].value_counts().head(10)
plt.figure(figsize=(10, 5))
sns.barplot(x=phone_counts.index, y=phone_counts.values)
plt.title('Distribution of Phone 1 (Top 10)')
plt.xlabel('Phone Number')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.show()
```

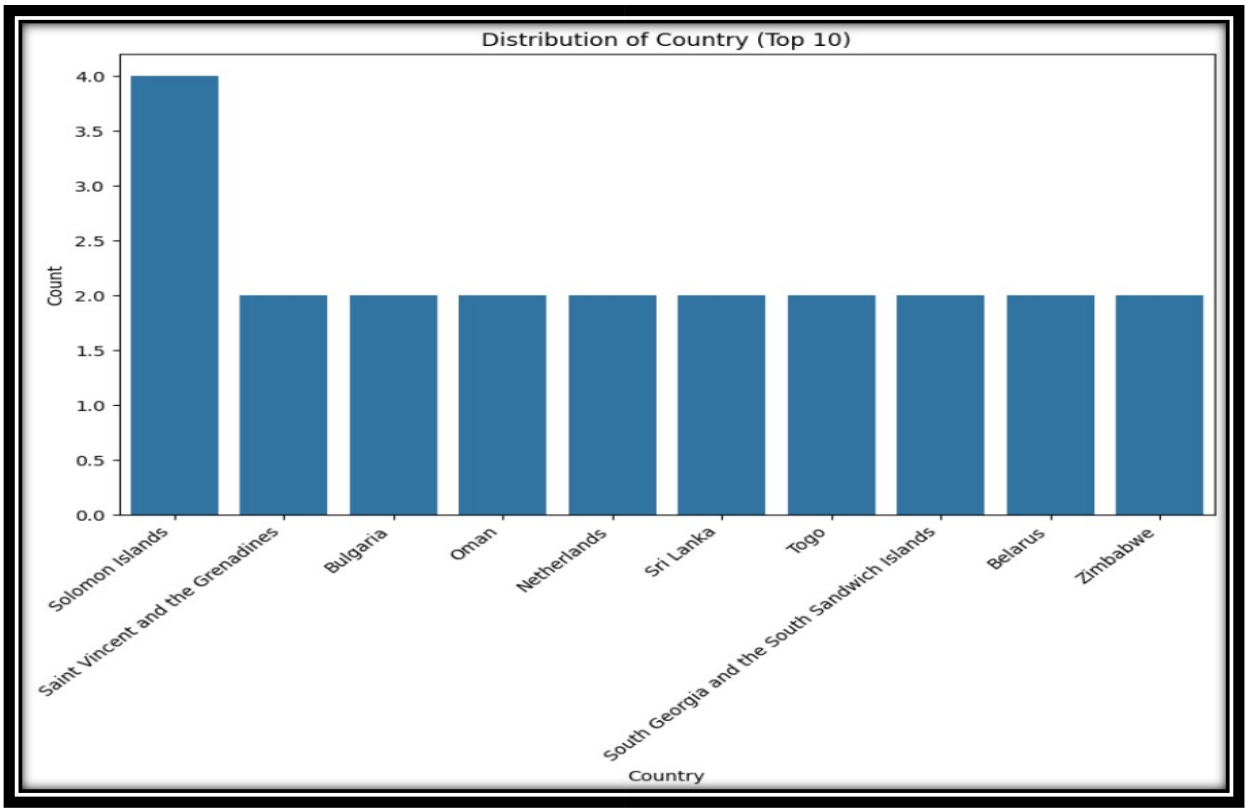


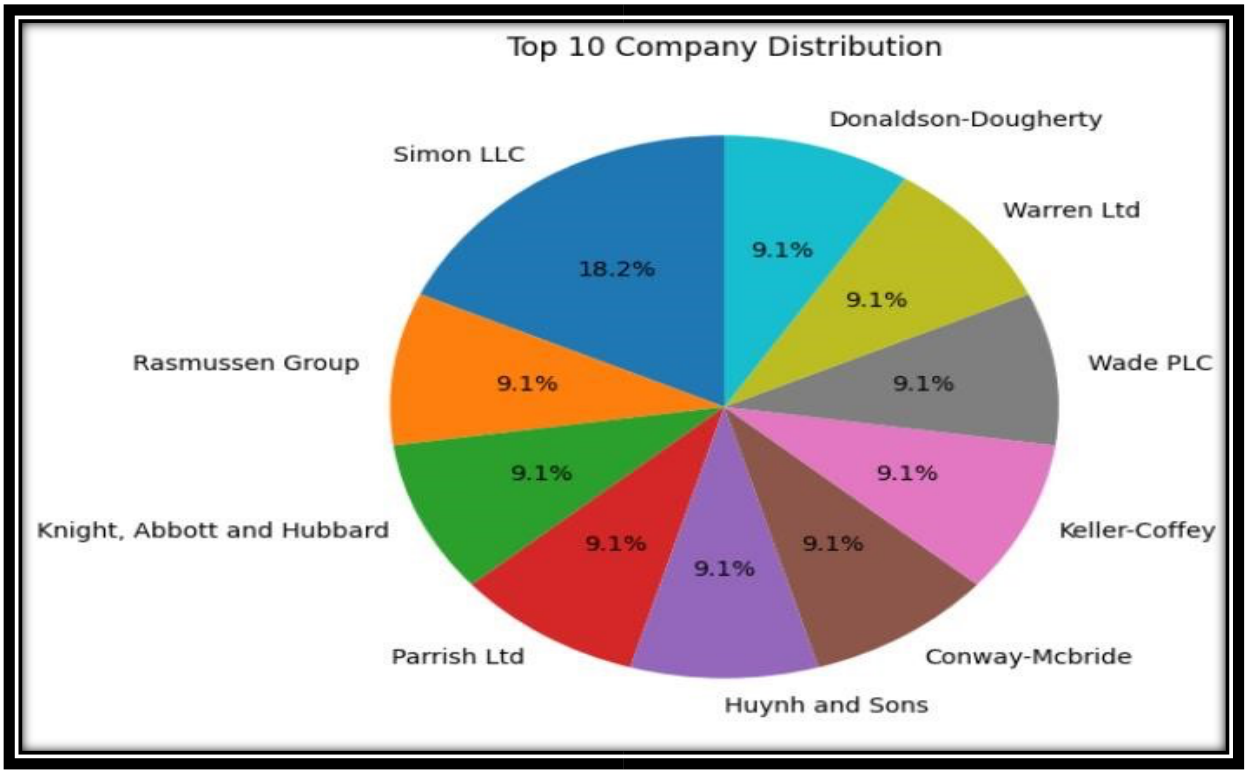
```python
# prompt: relationship between country and company scatter plot take upto only 10 entries

import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('/content/(Mannat)customers-100.csv')

# Assuming you have columns named 'Country' and 'Company'
# Select only the first 10 entries
df_subset = df[['Country', 'Company']].head(10)

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Country', y='Company', data=df_subset)
plt.title('Relationship between Country and Company (First 10 Entries)')
plt.xlabel('Country')
plt.ylabel('Company')
plt.xticks(rotation=45, ha='right')  # Rotate x-axis labels if needed
plt.show()
```

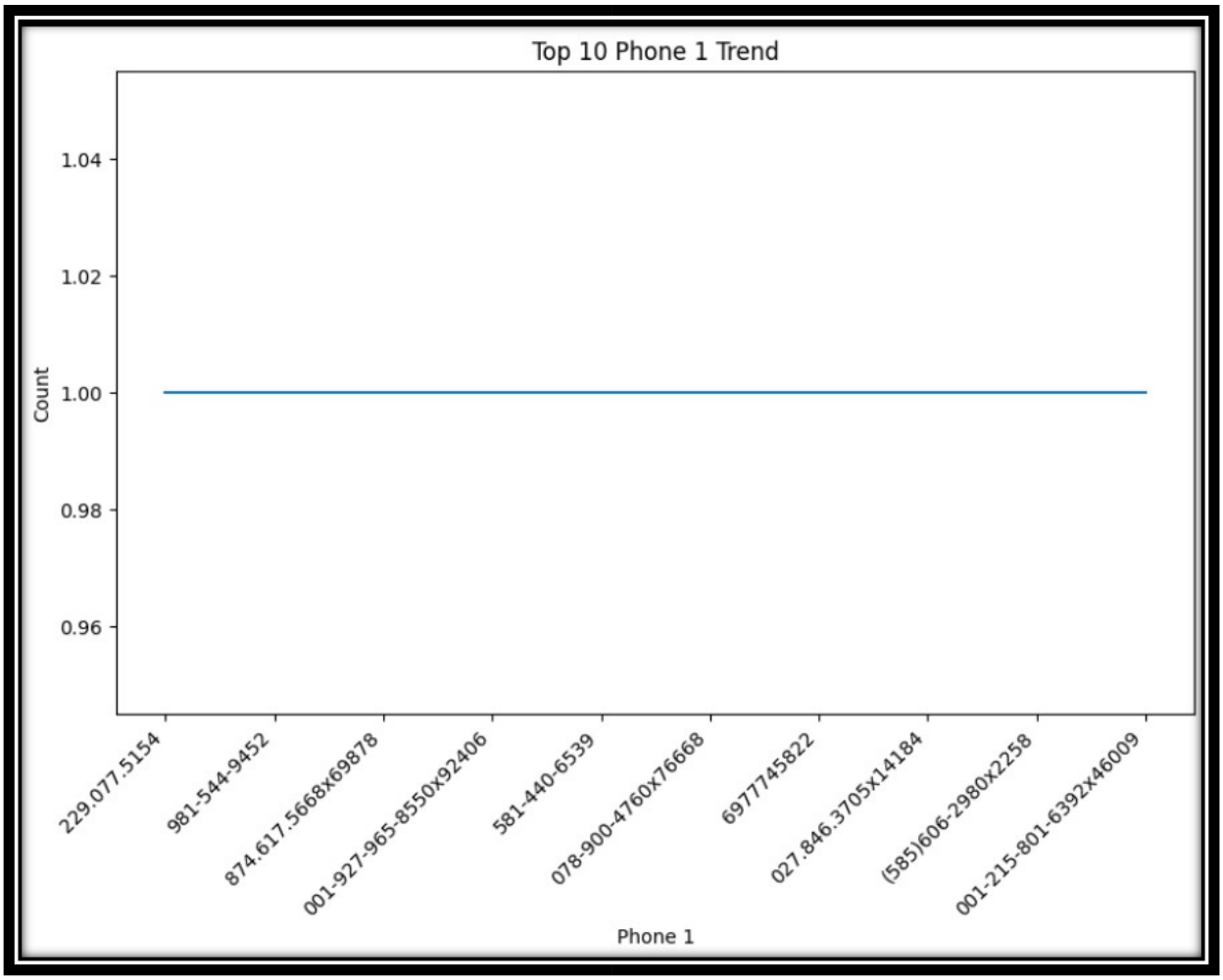


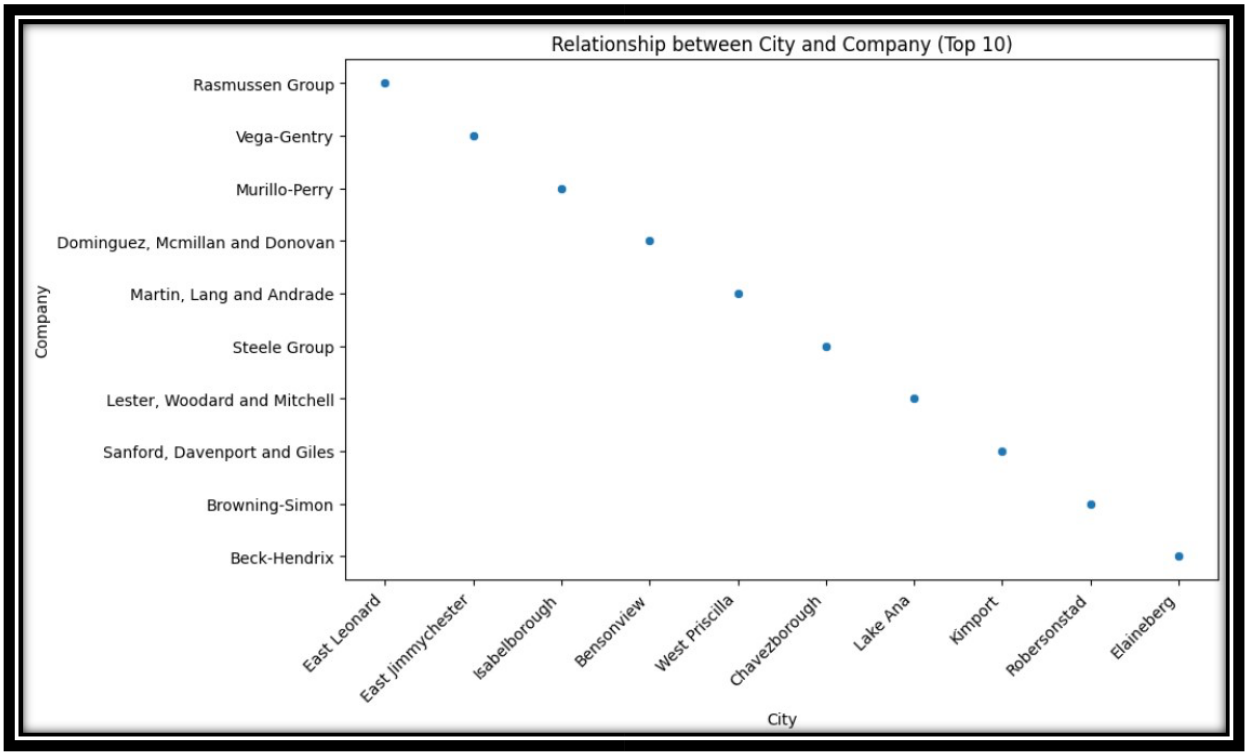
```python
# prompt: create different types of graph on top 10 entries on this dataset for different columns

import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('/content/(Mannat)customers-100.csv')

# Function to create different types of graphs for top 10 entries of a column
def create_graphs(df, column_name, graph_type='bar'):
    """
    Creates different types of graphs for the top 10 entries of a column.

    Args:
        df: Pandas DataFrame.
        column_name: Name of the column to analyze.
        graph_type: Type of graph to create ('bar', 'pie', 'line', 'scatter').
    """

    top_10 = df[column_name].value_counts().head(10)

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

    if graph_type == 'bar':
        sns.barplot(x=top_10.index, y=top_10.values)
        plt.title(f'Distribution of {column_name} (Top 10)')
        plt.xlabel(column_name)
        plt.ylabel('Count')
        plt.xticks(rotation=45, ha='right')
    elif graph_type == 'pie':
        plt.pie(top_10.values, labels=top_10.index, autopct='%1.1f%%', startangle=90)
        plt.title(f'Top 10 {column_name} Distribution')
    elif graph_type == 'line':
        plt.plot(top_10.index, top_10.values)
        plt.title(f'Top 10 {column_name} Trend')
        plt.xlabel(column_name)
        plt.ylabel('Count')
        plt.xticks(rotation=45, ha='right')
    elif graph_type == 'scatter':
        # For scatter plots, we need two columns. Let's use 'Company' as the other column.
        if 'Company' in df.columns:
            df_subset = df[[column_name, 'Company']].head(10)
```

```python
            sns.scatterplot(x=column_name, y='Company', data=df_subset)
            plt.title(f'Relationship between {column_name} and Company (Top 10)')
            plt.xlabel(column_name)
            plt.ylabel('Company')
            plt.xticks(rotation=45, ha='right')

    plt.show()


# Example usage for different columns and graph types:
create_graphs(df, 'Country', graph_type='bar')
create_graphs(df, 'Company', graph_type='pie')
create_graphs(df, 'Phone 1', graph_type='line')
create_graphs(df, 'City', graph_type='scatter')
```


Distribution of Country (Top 10)
Count
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Solomon Islands
Saint Vincent and the Grenadines
Bulgaria
Oman
Netherlands
Sri Lanka
Togo
South Georgia and the South Sandwich Islands
Belarus
Zimbabwe
Country


Top 10 Company Distribution
Donaldson-Dougherty
Warren Ltd
Wade PLC
Keller-Coffey
Conway-Mcbride
Huynh and Sons
Parrish Ltd
Knight, Abbott and Hubbard
Rasmussen Group
Simon LLC
9.1%
9.1%
9.1%
9.1%
9.1%
9.1%
9.1%
9.1%
9.1%
18.2%


Top 10 Phone 1 Trend
Count
1.04
1.02
1.00
0.98
0.96
229.077.5154
981-544-9452
874.617.5668x69878
001-927-965-8550x92406
581-440-6539
078-900-4760x76668
6977745822
027.846.3705x14184
(585)606-2980x2258
001-215-801-6392x46009
Phone 1


Relationship between City and Company (Top 10)
Company
Rasmussen Group
Vega-Gentry
Murillo-Perry
Dominguez, Mcmillan and Donovan
Martin, Lang and Andrade
Steele Group
Lester, Woodard and Mitchell
Sanford, Davenport and Giles
Browning-Simon
Beck-Hendrix
East Leonard
East Jimmychester
Isabelborough
Bensonview
West Priscilla
Chavezborough
Lake Ana
Kimport
Robersonstad
Elaineberg
City