```
#Load the required libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

#Load the data
df = pd.read_csv('/content/train.csv')
```

```
#View the data
df.head()
```

	category	sub_category	crimeaditionalinfo
0	Online and Social Media Related Crime	Cyber Bullying Stalking Sexting	I had continue received random calls and abusi...
1	Online Financial Fraud	Fraud CallVishing	The above fraudster is continuously messaging ...
2	Online Gambling Betting	Online Gambling Betting	He is acting like a police and demanding for m...
3	Online and Social Media Related Crime	Online Job Fraud	In apna Job I have applied for job interview f...
4	Online Financial Fraud	Fraud CallVishing	I received a call from lady stating that she w...

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
#Basic information

df.info()

#Describe the data

df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 93686 entries, 0 to 93685
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   category            93686 non-null  object
 1   sub_category        87095 non-null  object
 2   crimeaditionalinfo  93665 non-null  object
dtypes: object(3)
memory usage: 2.1+ MB
```

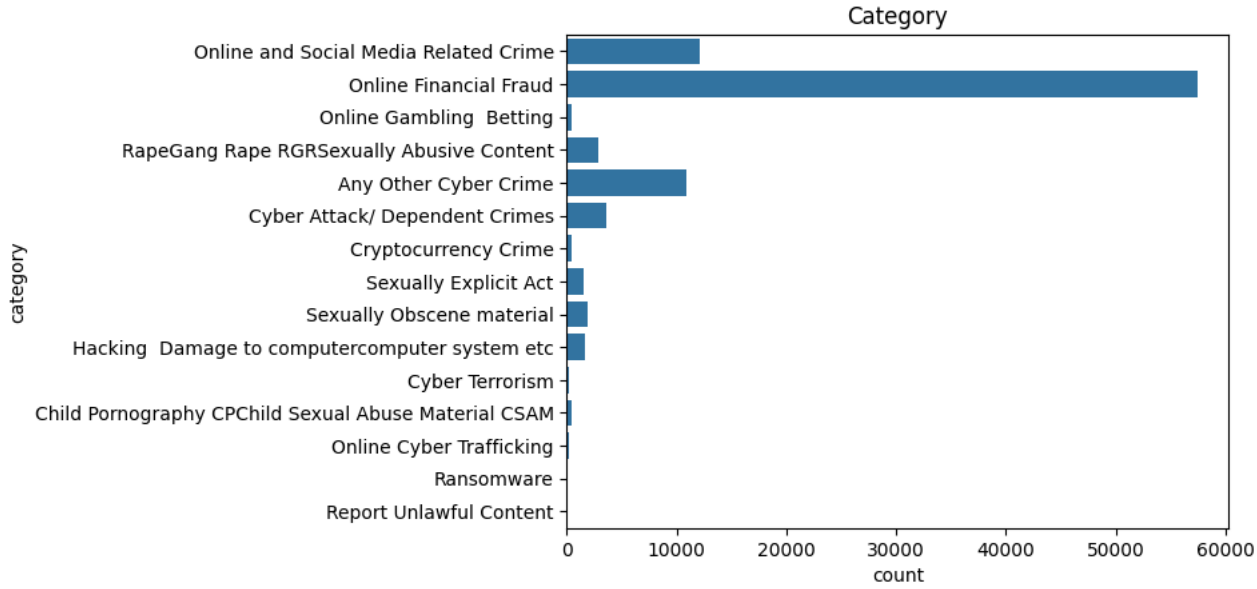
	category	sub_category	crimeaditionalinfo
count	93686	87095	93665
unique	15	35	85013
top	Online Financial Fraud	UPI Related Frauds	Respected Sir\r\n\r\nA very serious matter I w...
freq	57434	26856	2342

```
#Find the duplicates

df.duplicated().sum()
```

```
7803
```

```
#Plot the unique values

sns.countplot(df['category']).set_title('Category')
```

```
Text(0.5, 1.0, 'Category')
```




```
#Find null values

df.isnull().sum()
```

	0
category	0
sub_category	6591
crimeaditionalinfo	21

dtype: int64

```
#Replace null values

df.replace(np.nan,'0',inplace = True)

#Check the changes now
df.isnull().sum()
```

	0
category	0
sub_category	0
crimeaditionalinfo	0

dtype: int64

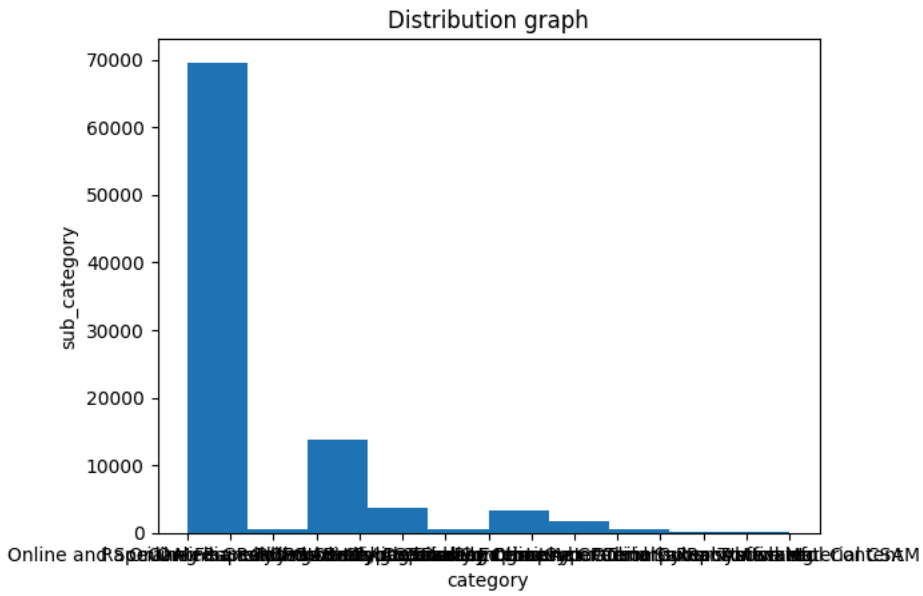
```
#Datatypes

df.dtypes
```

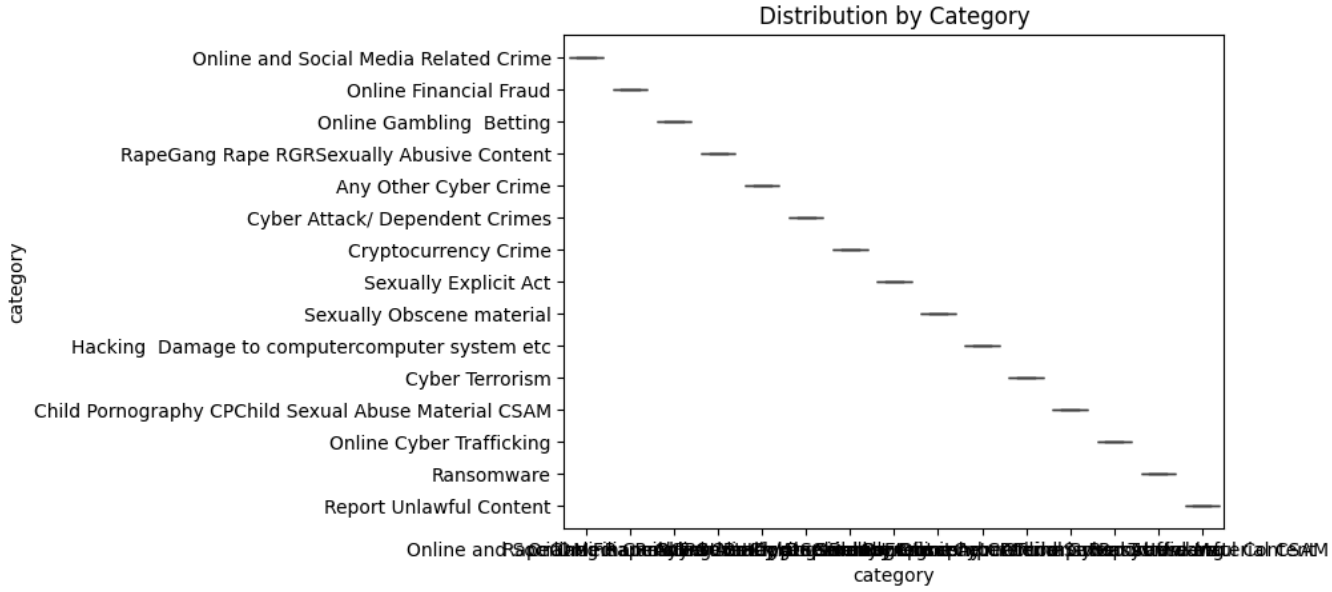
	0
category	object
sub_category	object
crimeaditionalinfo	object

dtype: object

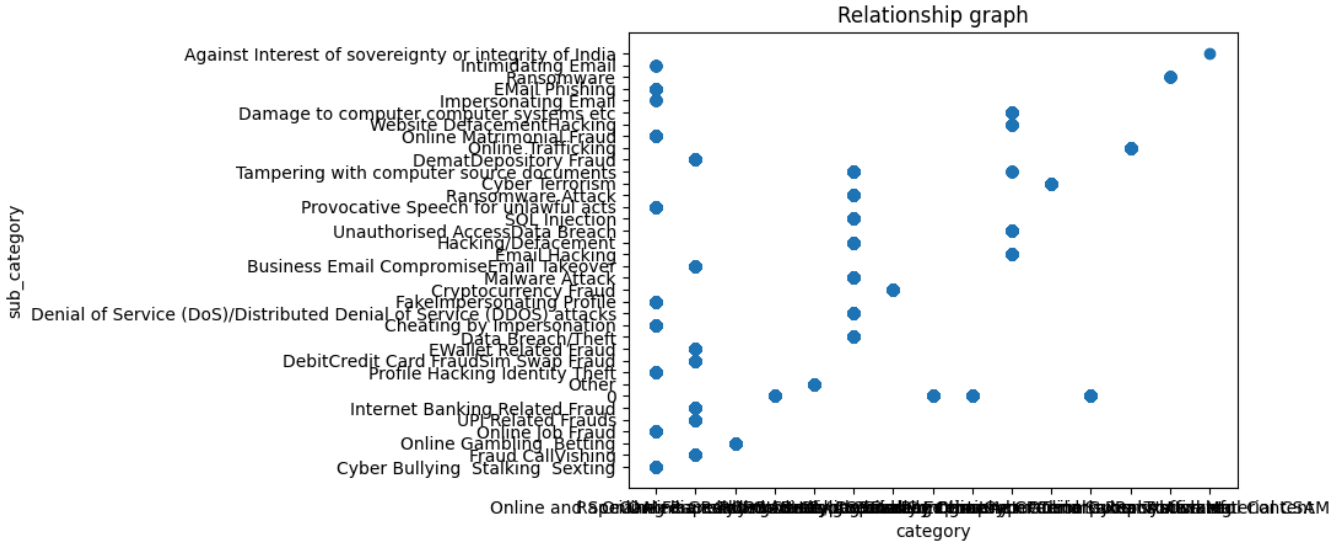
```
import matplotlib.pyplot as plt
plt.hist(df['category'])
plt.title('Distribution graph')
plt.xlabel('category')
plt.ylabel('sub_category')
plt.show()

# Box plot for a numerical column grouped by category
sns.boxplot(x='category', y='category', data=df)
```

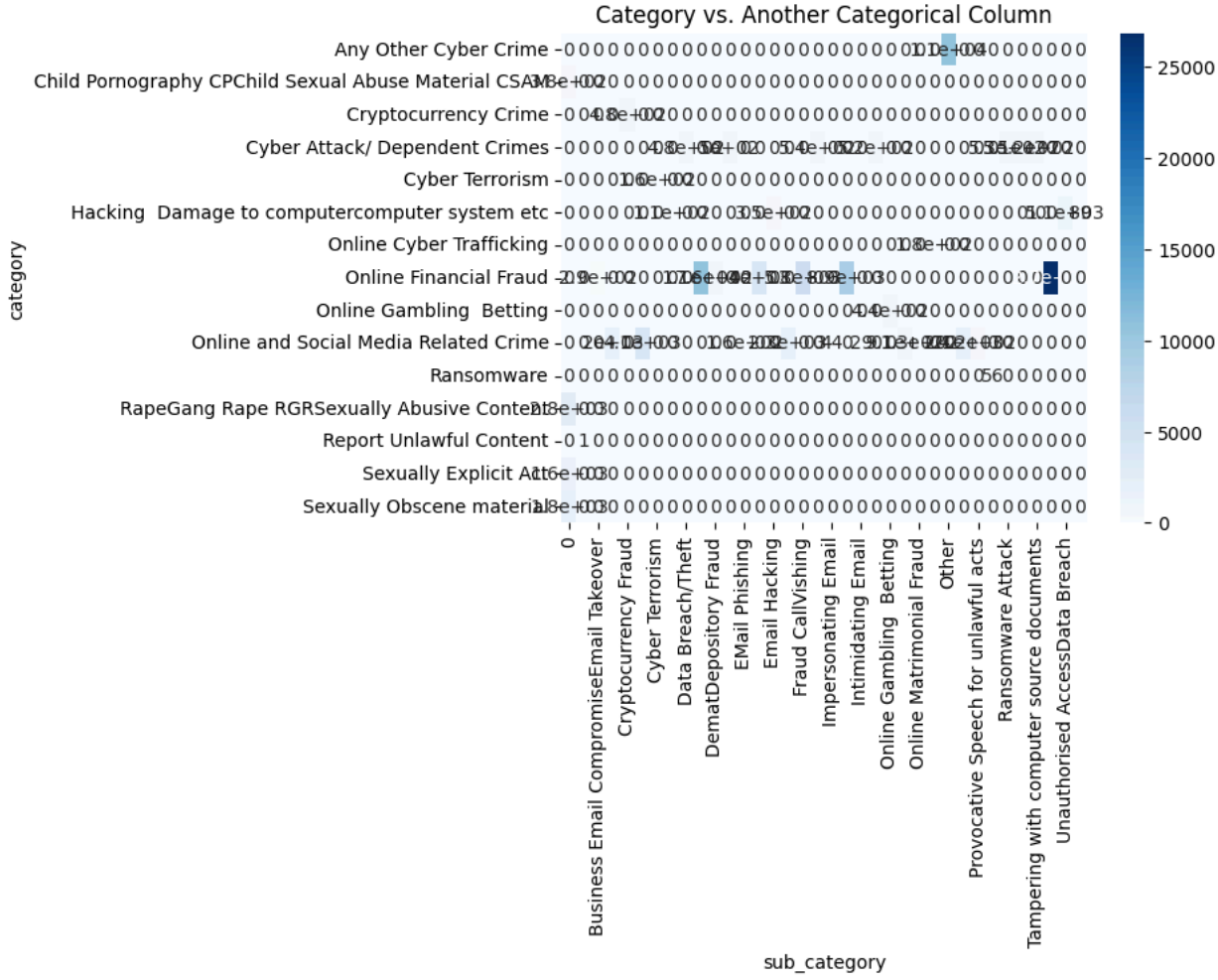
```
plt.title('Distribution by Category')
plt.show()
```



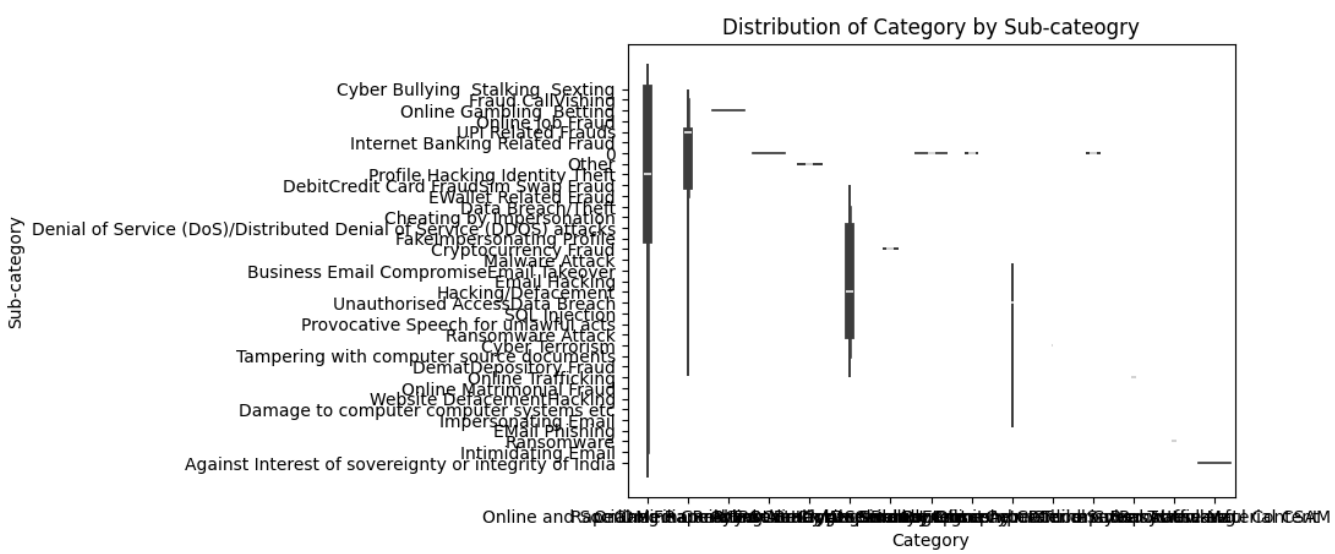



```
plt.scatter(df['category'], df['sub_category'])
plt.title('Relationship graph')
plt.xlabel('category')
plt.ylabel('sub_category')
plt.show()

# Heatmap for categorical features
category_counts = pd.crosstab(df['category'], df['sub_category'])
sns.heatmap(category_counts, annot=True, cmap='Blues')
plt.title('Category vs. Another Categorical Column')
plt.show()
```

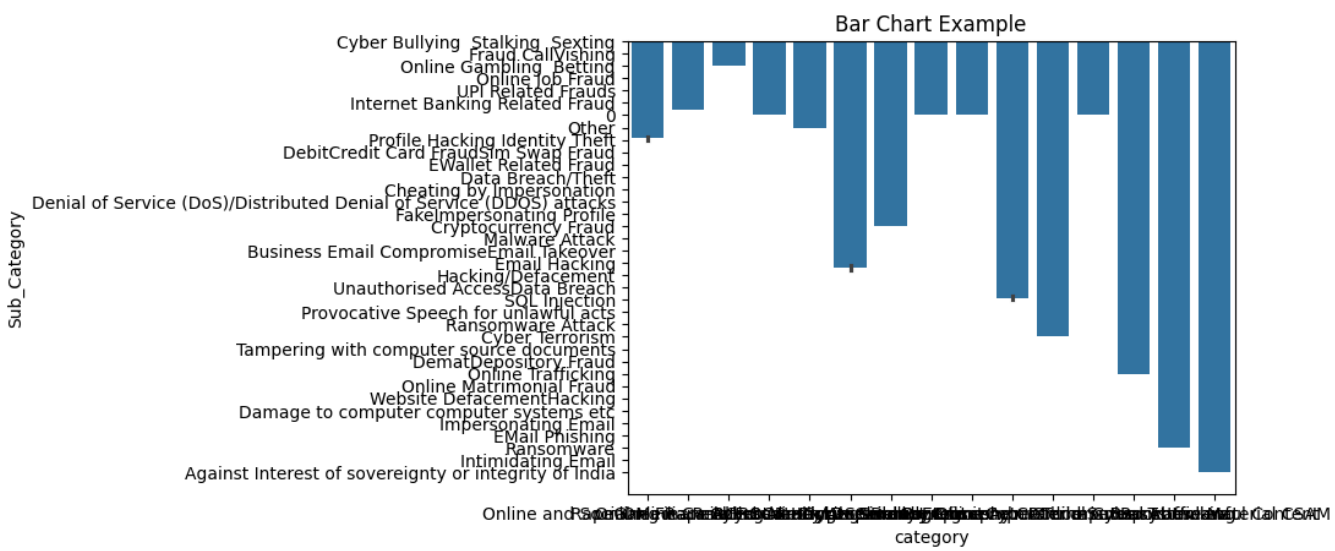




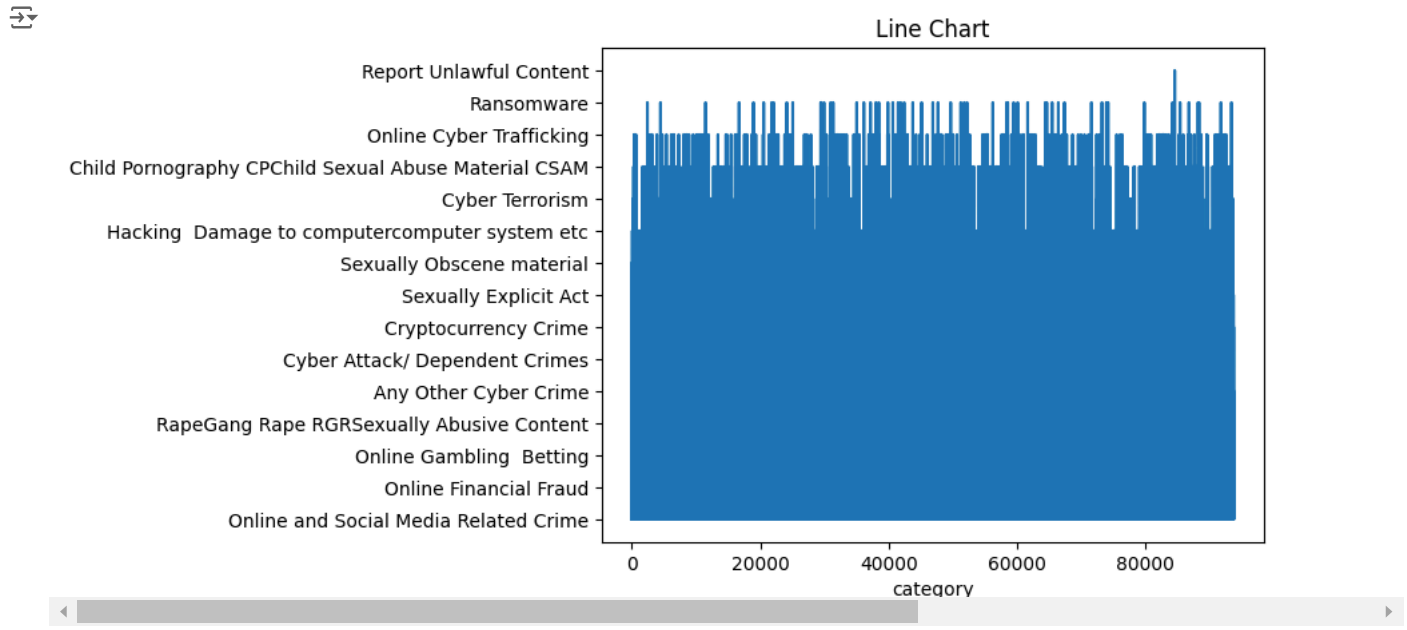

```
sns.violinplot(x='category', y='sub_category', data=df)
plt.title('Distribution of Category by Sub-cateogry')
plt.xlabel('Category')
plt.ylabel('Sub-category')
plt.show()
```



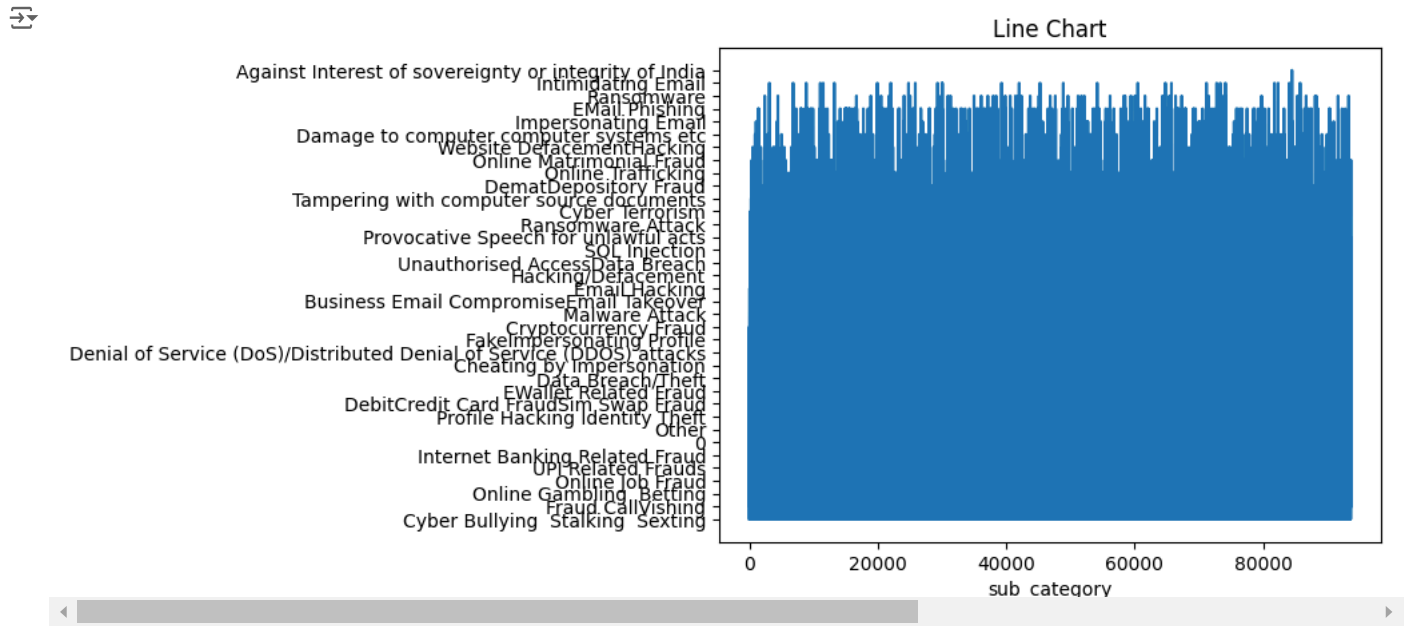

```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming df is your DataFrame and 'category' and 'value' are your columns
sns.barplot(x='category', y='sub_category', data=df)
plt.title('Bar Chart Example')
plt.xlabel('category')
plt.ylabel('Sub_Category')
plt.show()
```



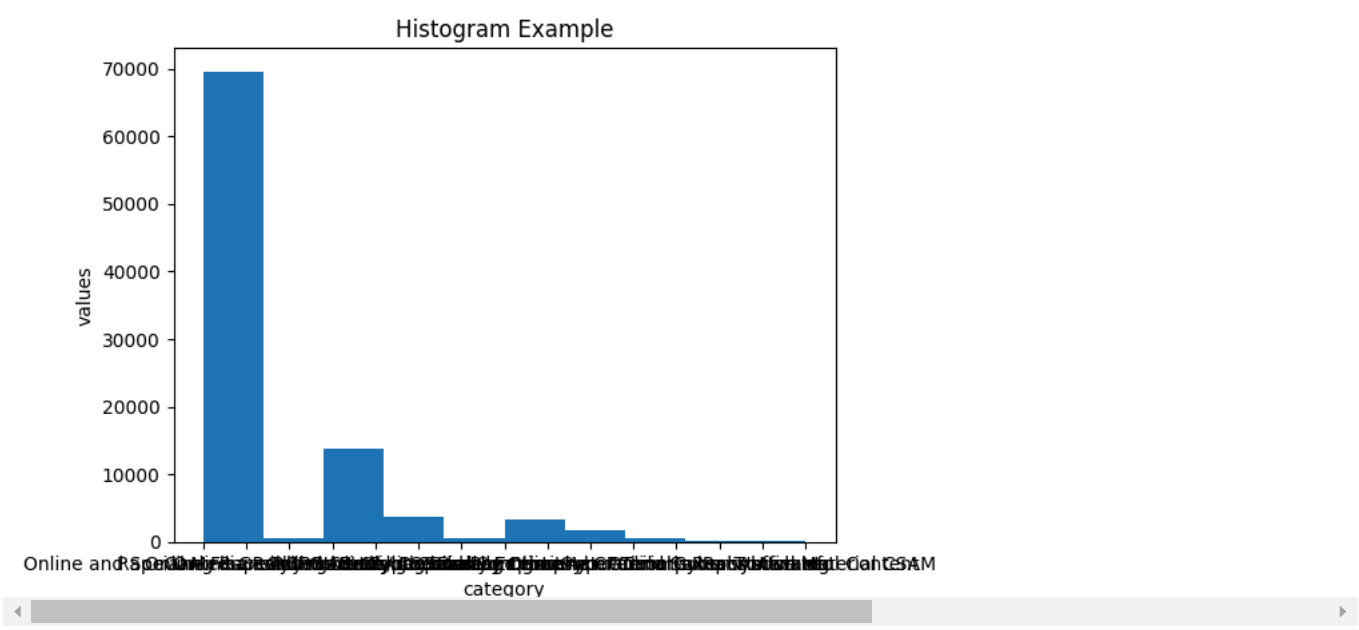

```
import matplotlib.pyplot as plt
plt.plot(df['category'])
plt.title('Line Chart')
plt.xlabel('category')
plt.show()
```



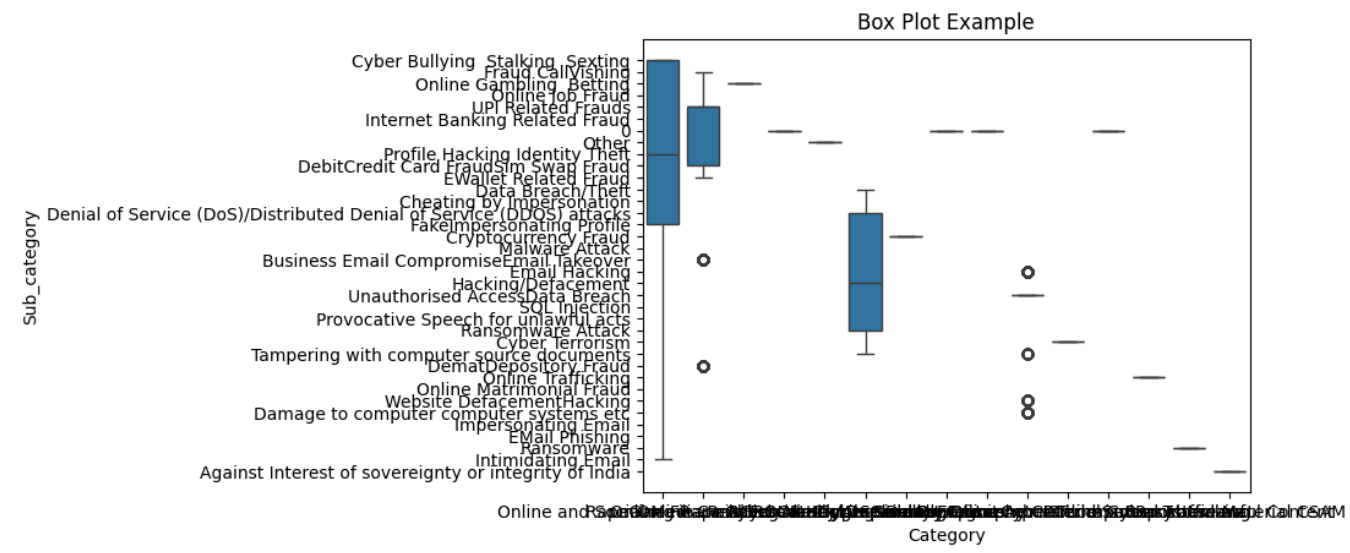

```
import matplotlib.pyplot as plt
plt.plot(df['sub_category'])
plt.title('Line Chart')
plt.xlabel('sub_category')
plt.show()
```



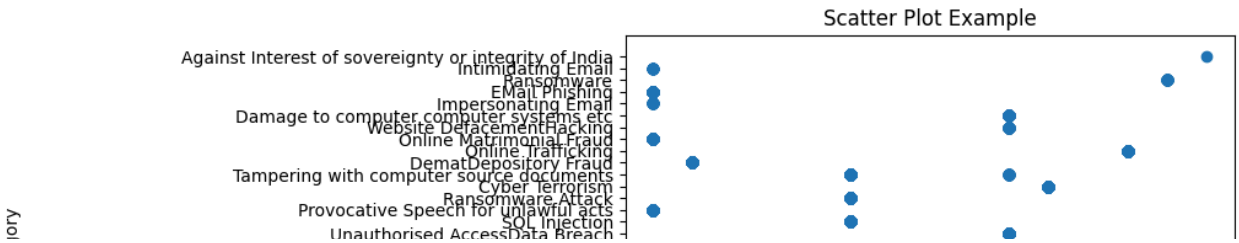

```
import matplotlib.pyplot as plt

plt.hist(df['category'], bins=10)
plt.title('Histogram Example')
plt.xlabel('category')
plt.ylabel('values')
plt.show()
```




```
import seaborn as sns
sns.boxplot(x='category', y='sub_category', data=df)
plt.title('Box Plot Example')
plt.xlabel('Category')
plt.ylabel('Sub_category')
plt.show()
```




```
import matplotlib.pyplot as plt

# Assuming df is your DataFrame and 'x' and 'y' are your columns
plt.scatter(df['category'], df['sub_category'])
plt.title('Scatter Plot Example')
plt.xlabel('category')
plt.ylabel('sub_category')
plt.show()
```




```
import matplotlib.pyplot as plt
import pandas as pd

# Calculate the frequency of each category
category_counts = df['category'].value_counts()

# Get the categories and their corresponding counts
categories = category_counts.index.tolist()
values = category_counts.values.tolist()

# Create the pie chart
plt.pie(values, labels=categories, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Categories')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```