

```
In [ ]: import zipfile
import pandas as pd

with zipfile.ZipFile('complaints.csv.zip', 'r') as zip_ref:
    zip_ref.extractall('data_folder')

df = pd.read_csv('data_folder/complaints.csv')
```

```
In [ ]: print(df.info())
print(df.describe())
```

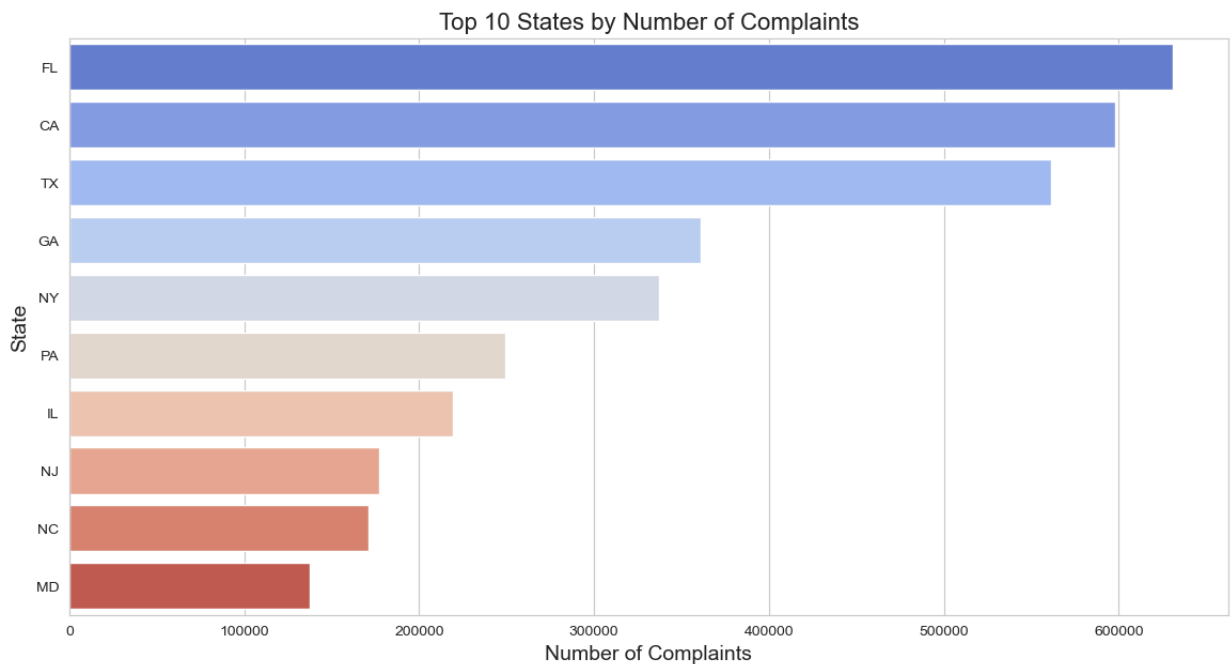
```
In [ ]: print(df.isnull().sum())

df_cleaned = df.dropna()
```

```
In [ ]: print(df['Product'].value_counts())
print(df.groupby('Product')['Sub-product'].value_counts())
```

```
In [ ]: state_counts = df['State'].value_counts().head(10)

plt.figure(figsize=(14, 7))
sns.set_style("whitegrid")
sns.barplot(x=state_counts.values, y=state_counts.index, palette='coolwarm')
plt.title('Top 10 States by Number of Complaints', fontsize=16)
plt.xlabel('Number of Complaints', fontsize=14)
plt.ylabel('State', fontsize=14)
plt.show()
```



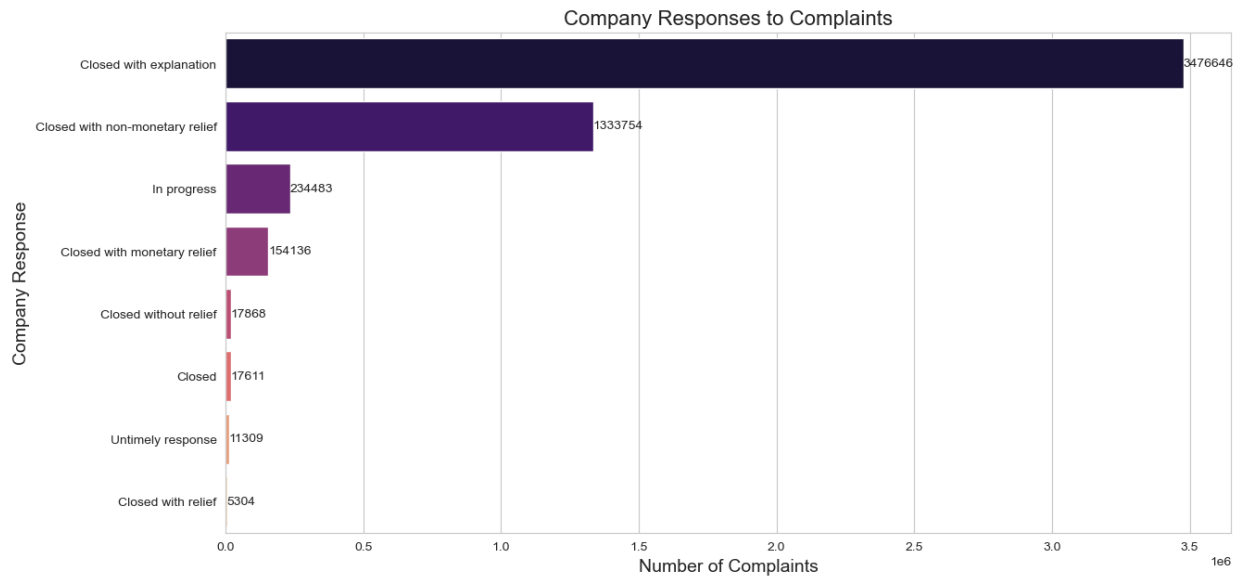
```
In [ ]: response_counts = df['Company response to consumer'].value_counts()

plt.figure(figsize=(14, 7))
sns.set_style("whitegrid")
barplot = sns.barplot(x=response_counts.values, y=response_counts.index, palette='magma')
```

```
plt.title('Company Responses to Complaints', fontsize=16)
plt.xlabel('Number of Complaints', fontsize=14)
plt.ylabel('Company Response', fontsize=14)

for index, value in enumerate(response_counts.values):
    barplot.text(value, index, str(value), ha='left', va='center')

plt.show()
```

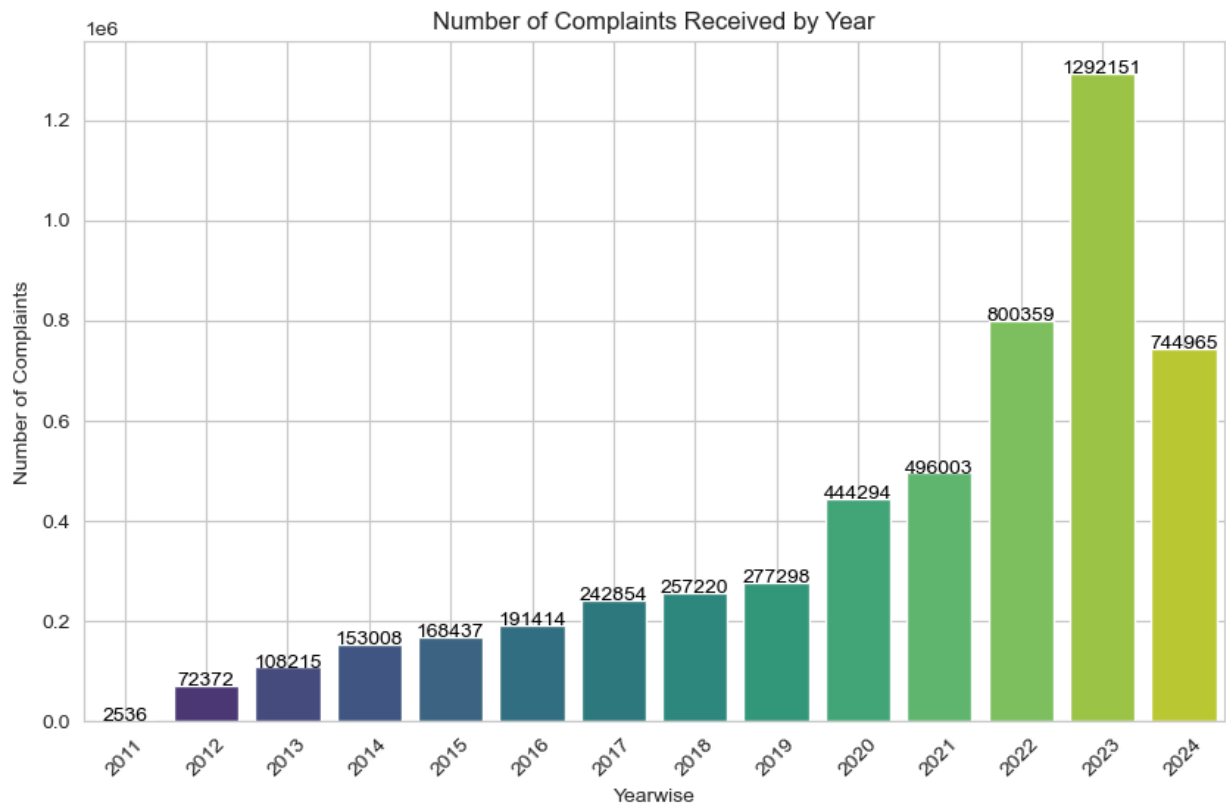


```
In [ ]: import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
barplot = sns.barplot(x='Year', y='Complaint Count', data=complaints_by_year, palette=
plt.title('Number of Complaints Received by Year')
plt.xlabel('Yearwise')
plt.ylabel('Number of Complaints')

for index, row in complaints_by_year.iterrows():
    barplot.text(index, row['Complaint Count'], str(row['Complaint Count']), color='b')

plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

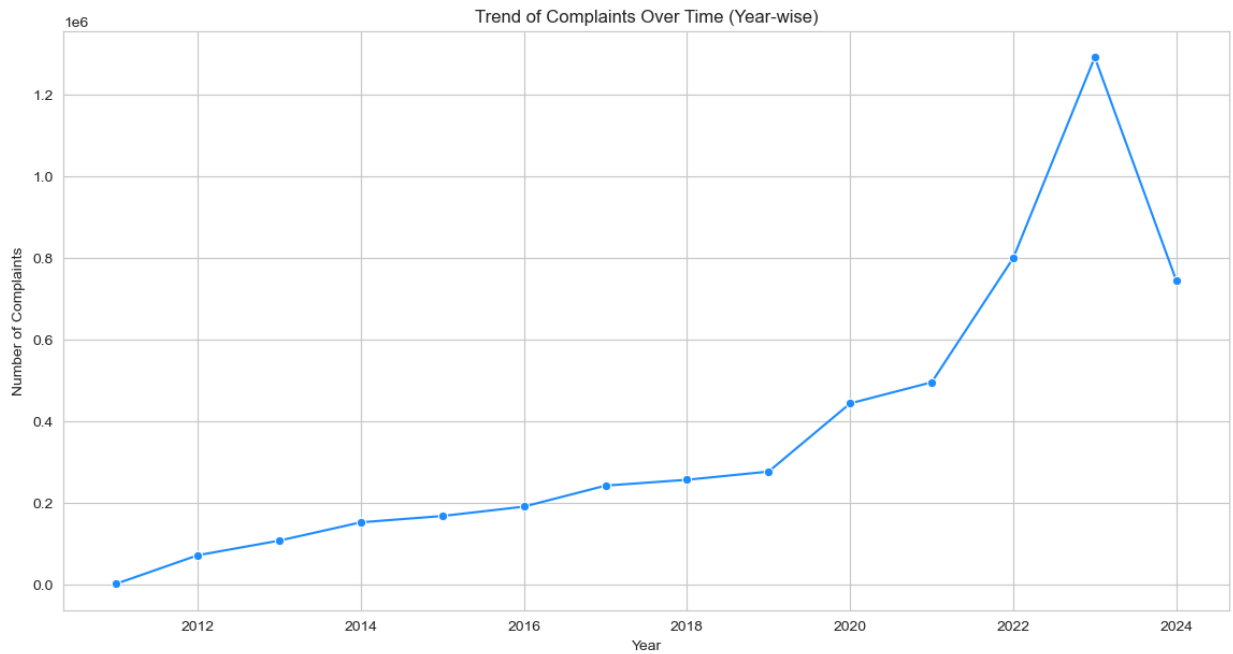


```
In [ ]: import matplotlib.pyplot as plt
import seaborn as sns

df['Year'] = df['Date received'].dt.year

complaints_by_year = df.groupby('Year').size().reset_index(name='Complaint Count')

plt.figure(figsize=(14, 7))
sns.lineplot(x='Year', y='Complaint Count', data=complaints_by_year, marker='o', color='red')
plt.title('Trend of Complaints Over Time (Year-wise)')
plt.xlabel('Year')
plt.ylabel('Number of Complaints')
plt.grid(True)
plt.show()
```

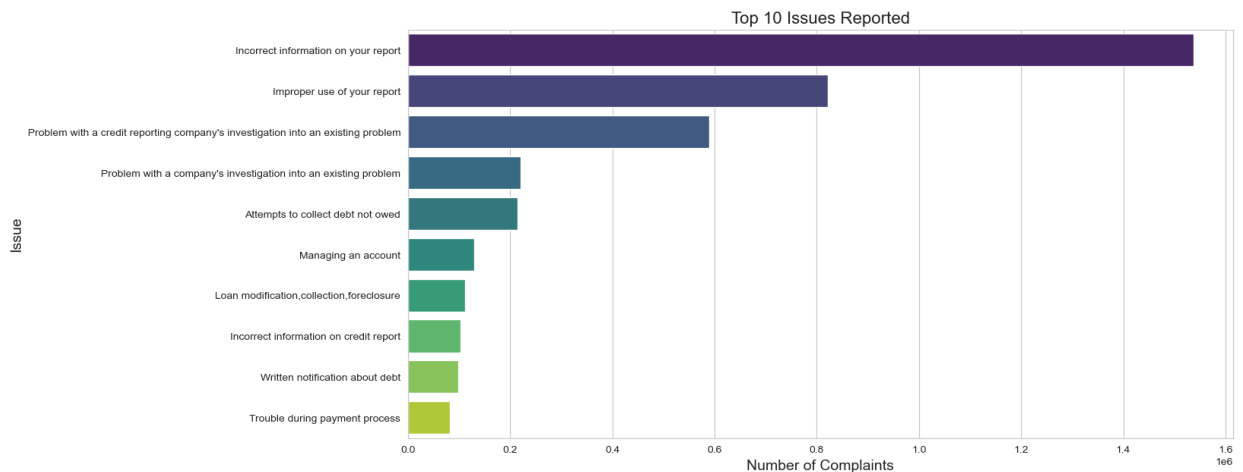


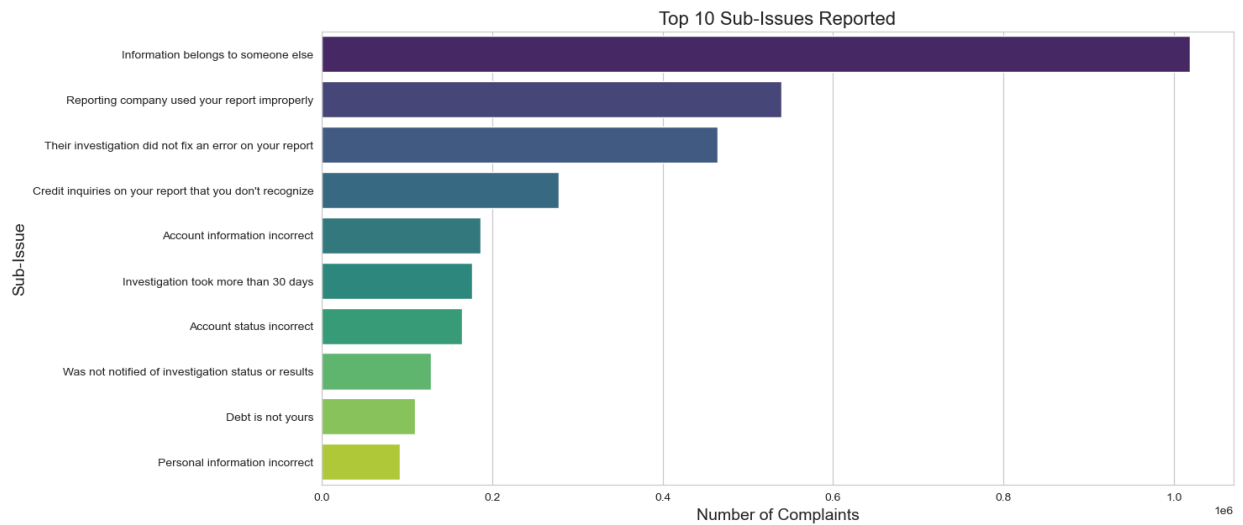
```
In [ ]: # Common Issues
issue_counts = df['Issue'].value_counts().head(10)

plt.figure(figsize=(14, 7))
sns.set_style("whitegrid")
sns.barplot(x=issue_counts.values, y=issue_counts.index, palette='viridis')
plt.title('Top 10 Issues Reported', fontsize=16)
plt.xlabel('Number of Complaints', fontsize=14)
plt.ylabel('Issue', fontsize=14)
plt.show()

# Common Sub-Issues
sub_issue_counts = df['Sub-issue'].value_counts().head(10)

plt.figure(figsize=(14, 7))
sns.set_style("whitegrid")
sns.barplot(x=sub_issue_counts.values, y=sub_issue_counts.index, palette='viridis')
plt.title('Top 10 Sub-Issues Reported', fontsize=16)
plt.xlabel('Number of Complaints', fontsize=14)
plt.ylabel('Sub-Issue', fontsize=14)
plt.show()
```





```
In [ ]: #Problem 2:
def longest_increasing_subsequence(nums):
    if not nums:
        return 0

    max_length = 1
    current_length = 1

    for i in range(1, len(nums)):
        if nums[i] > nums[i - 1]:
            current_length += 1
            max_length = max(max_length, current_length)
        else:
            current_length = 1

    return max_length

#Example
input1 = [1, 3, 5, 4, 7]
input2 = [2, 2, 2, 2, 2]

print("Output for input1:", longest_increasing_subsequence(input1))
print("Output for input2:", longest_increasing_subsequence(input2))
```

Output for input1: 3

Output for input2: 1

```
In [ ]: #Problem 3:
def largest_number(nums):
    nums = sorted(map(str, nums), key=lambda x: x * 3, reverse=True)
    return str(int(''.join(nums)))

#Example

input1 = [10, 2]
input2 = [3, 30, 34, 5, 9]

print("Output for input1:", largest_number(input1))
print("Output for input2:", largest_number(input2))
```

Output for input1: 210

Output for input2: 9534330

```
In [ ]: #Problem 4:
import json
import csv

# Provide the full path to the JSON file in your Downloads folder
file_path = "C:/Users/ajith/Downloads/DT A1 sample_json (1) (1) (3).json"

with open(file_path) as f:
    data = json.load(f)

# Extract servlet information
servlets = data['web-app']['servlet']

# Create a CSV file and write headers
with open('servlet_info.csv', mode='w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['servlet-name', 'servlet-class'])

# Write servlet information to CSV file
for servlet in servlets:
    servlet_name = servlet['servlet-name']
    servlet_class = servlet['servlet-class']
    writer.writerow([servlet_name, servlet_class])

print("CSV file created successfully.")

CSV file created successfully.
```

```
In [ ]:
```