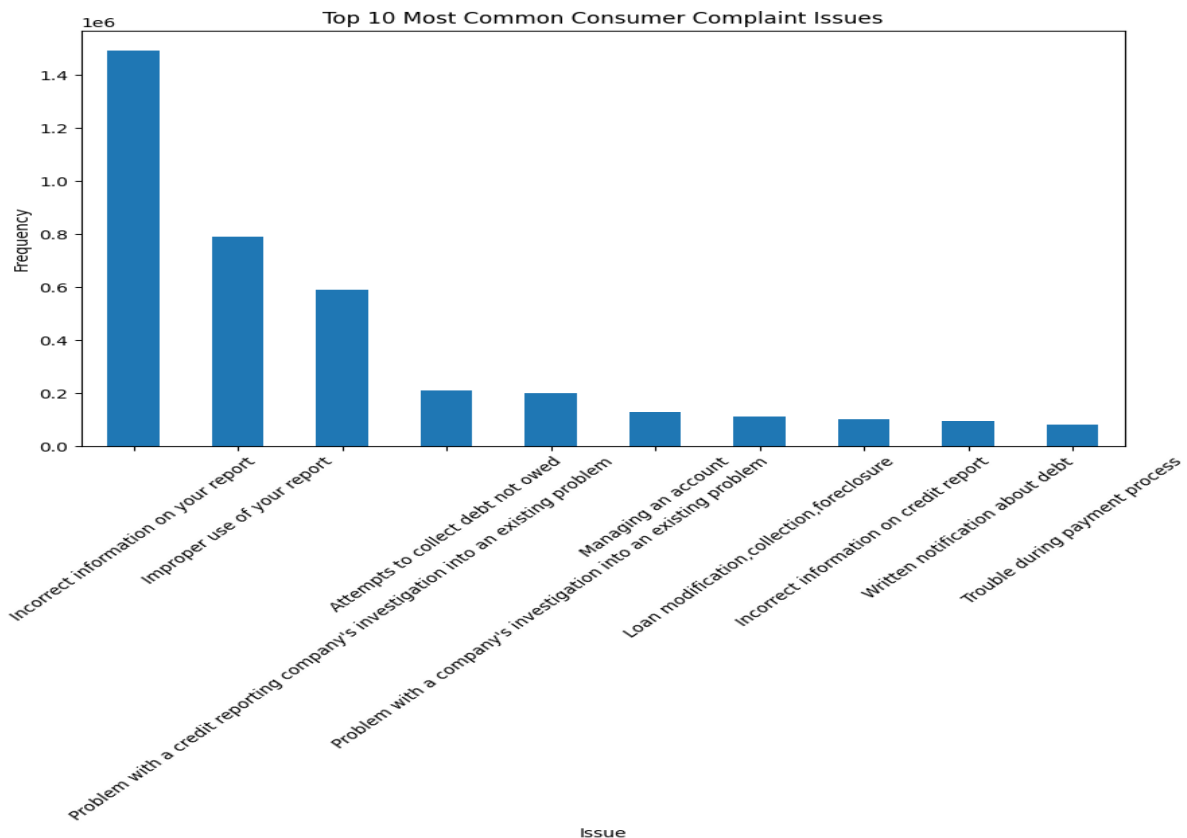


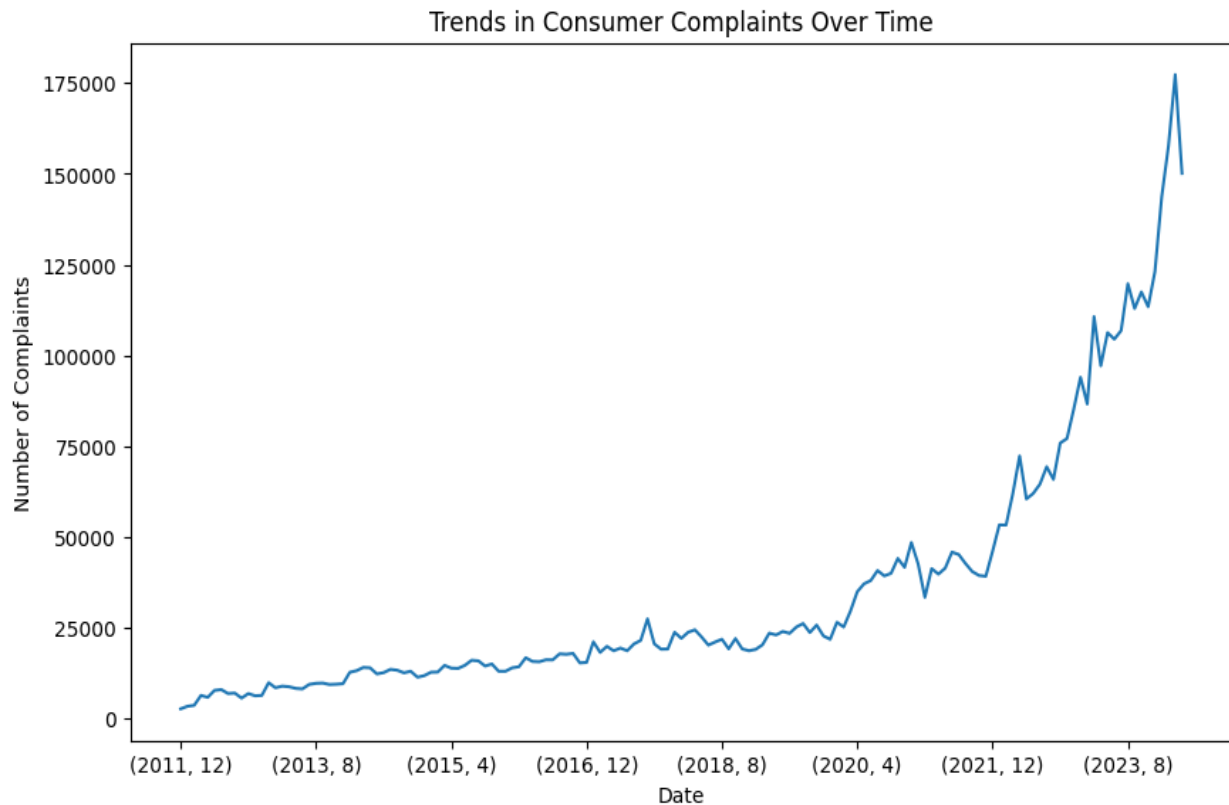
COMPLAINTS ANALYSIS

1. Data set reference link: <https://www.consumerfinance.gov/data-research/consumer-complaints/#download-the-data>
File data source: <https://files.consumerfinance.gov/ccdb/complaints.csv.zip>
Problem statement:
Download the data from the file data source and provide possible data insights.



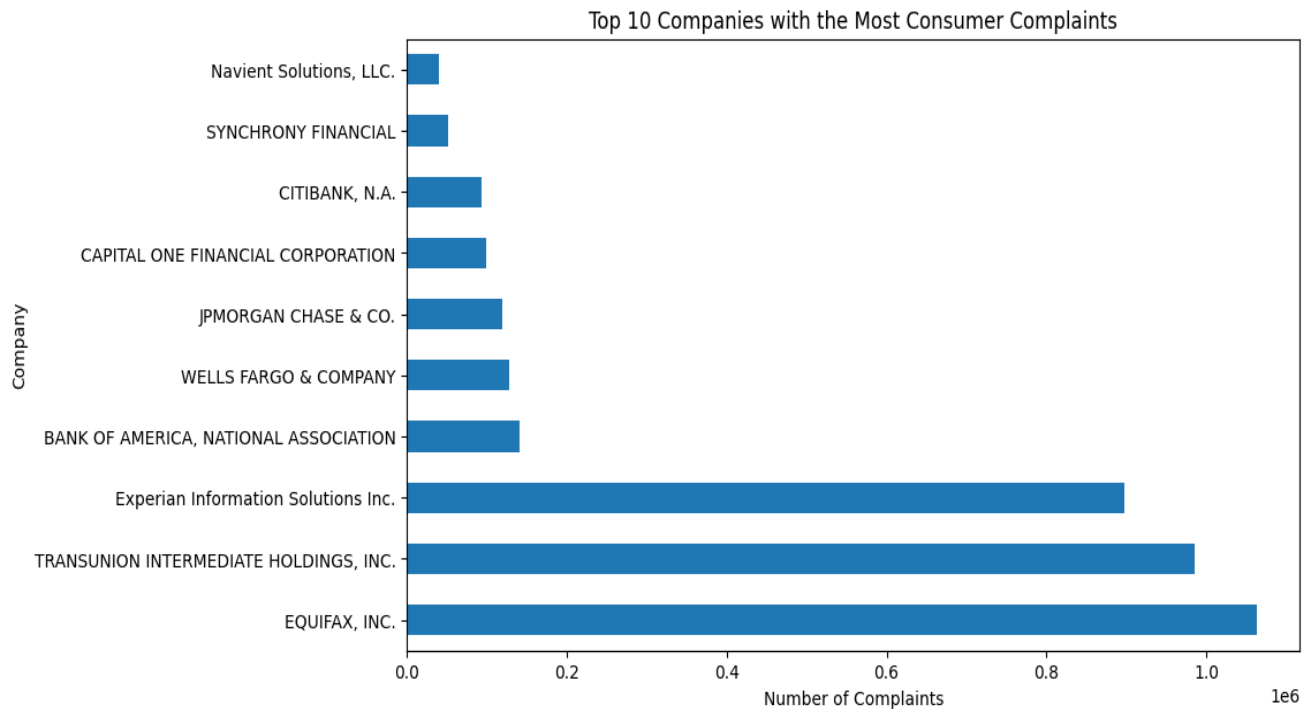
- 1.The bar chart guides resource allocation by revealing top consumer complaints.
- 2.Issue frequency insights inform effective decision-making and resource allocation.
- 3.Identifying high-frequency issues drives quality enhancements for improved satisfaction.
- 4.Issue distribution analysis uncovers process improvement opportunities.
- 5.Proactive resolution fosters trust and loyalty, bolstering customer engagement.

COMPLAINTS ANALYSIS



1. The chart displays how the volume of consumer complaints fluctuates over time, showcasing any discernible patterns or trends.
2. Peaks and troughs in the line reveal potential seasonal variations in complaint frequency, indicating periods of higher complaint rates
3. The line's trajectory signifies the long-term trend in complaint volumes, indicating whether they are rising, falling, or stable over time.
4. Sudden spikes or drops in the line may indicate exceptional events or issues that significantly impact complaint volumes during specific periods.
5. The chart empowers stakeholders to pinpoint peaks in complaint activity, facilitating proactive measures to enhance service quality and address consumer concerns effectively.

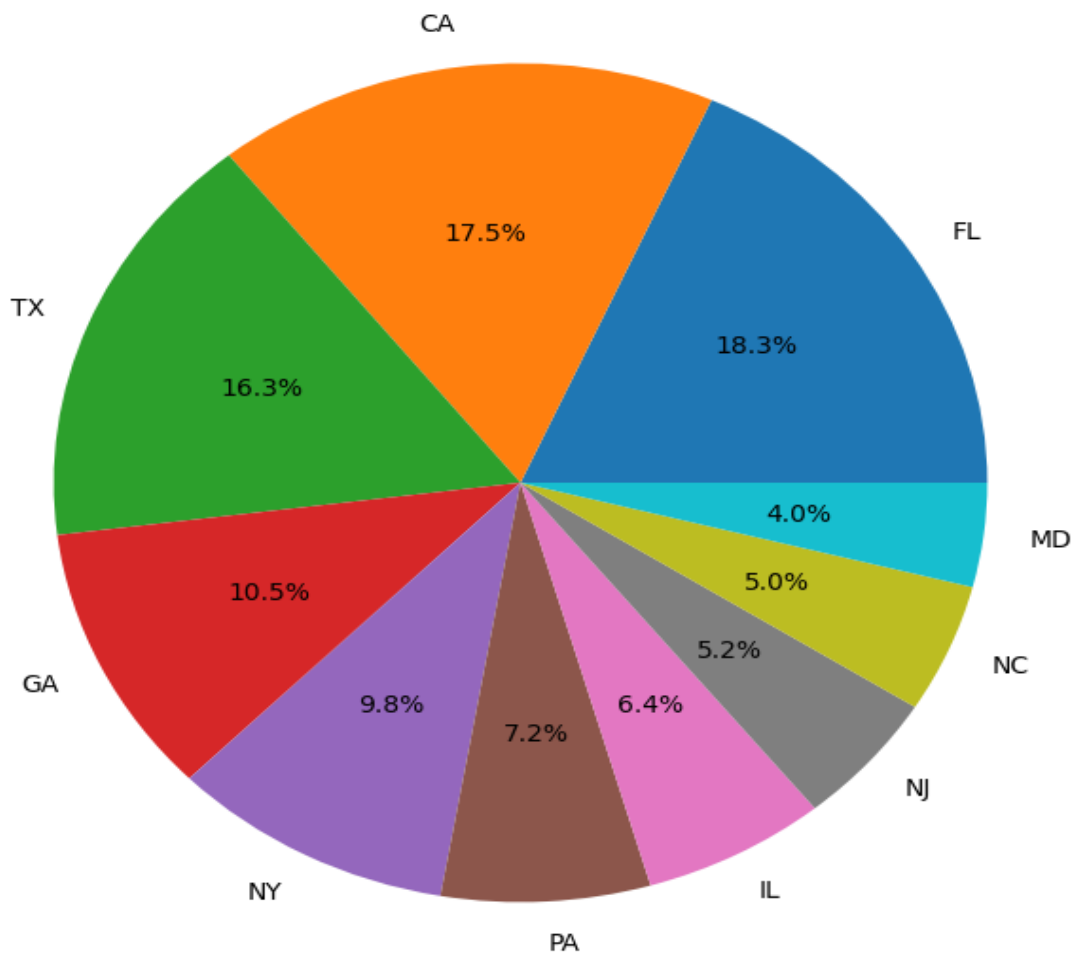
COMPLAINTS ANALYSIS



- 1.The companies are ranked by the number of complaints, with the highest complaint volume at the top.
- 2.Bar length visually represents complaint volume, facilitating quick comparison among companies, with longer bars indicating higher complaint counts.
- 3.The chart enables easy identification of companies with the highest and lowest complaint volumes.
- 4.The chart offers insight into consumer satisfaction and service quality by highlighting companies facing potential issues through higher complaint volumes, indicating potential dissatisfaction.
- 5.The chart facilitates benchmarking for companies, aiding regulatory bodies and consumer advocacy groups in monitoring marketplace issues.

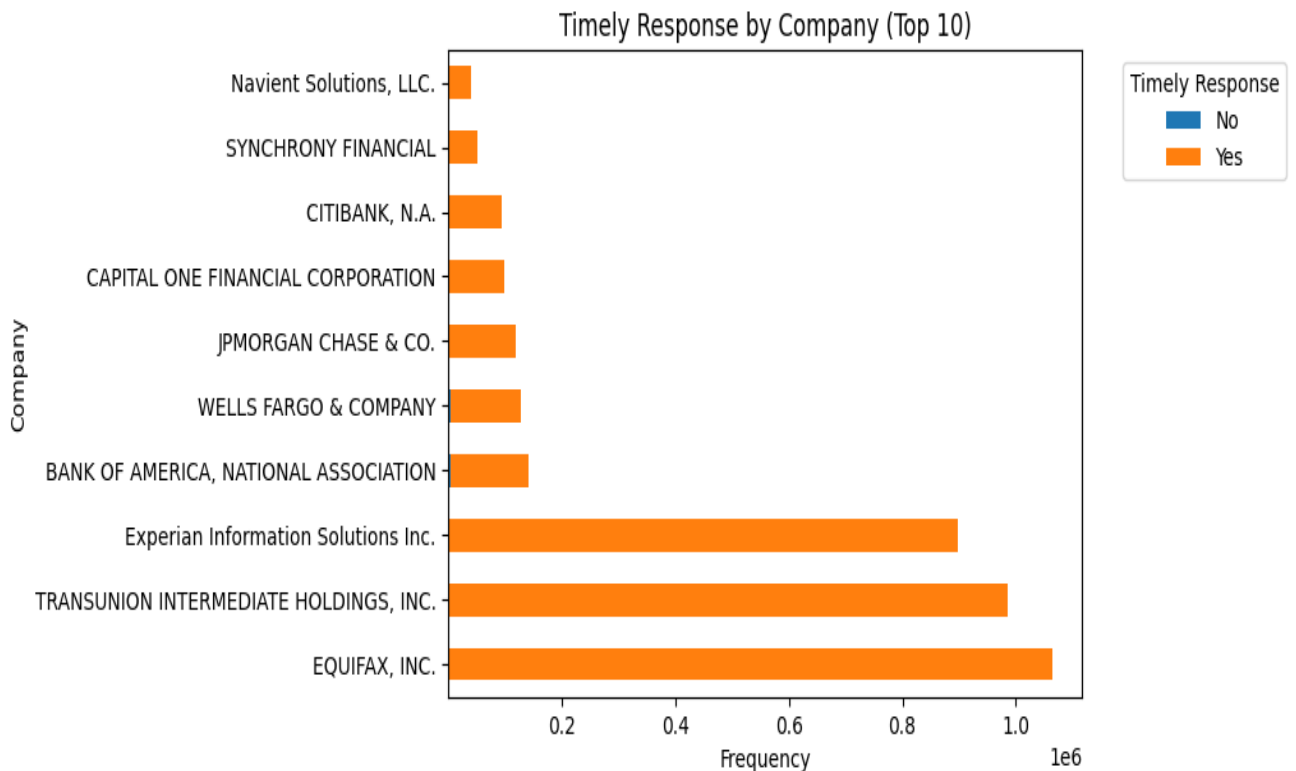
COMPLAINTS ANALYSIS

Top 10 States with the Highest Consumer Complaints



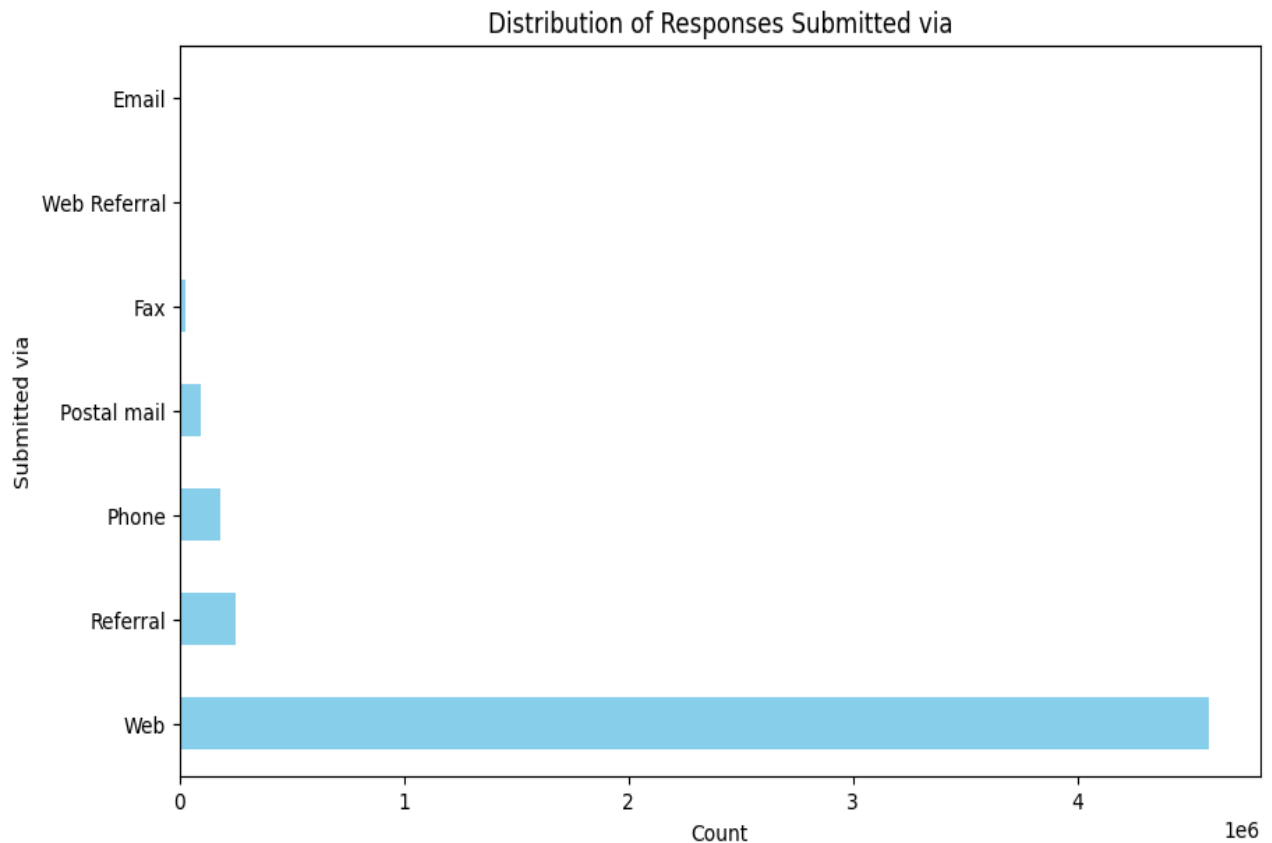
1. Pie chart visualizes top 10 states by consumer complaints, derived from 'State' column's value counts in DataFrame.
2. Pie slices represent complaint proportions per state, formatted with one decimal place using `autopct='%1.1f%%'`.
3. Pie chart visually shows complaint distribution among top 10 states; larger slices indicate higher complaint volumes.
4. Pie chart allows easy comparison of each state's contribution to total complaints, highlighting states with more significant shares in the top10
5. Pie chart aids stakeholders in pinpointing states with highest complaint volumes, facilitating focused attention on pertinent consumer issues.

COMPLAINTS ANALYSIS



- 1.The code groups and reshapes DataFrame by 'Company' and 'Timely response?', calculating group sizes.
- 2.The code computes total responses per company, sorting by this sum to find the top 10 companies.
- 3.The code utilizes Matplotlib to create a horizontal stacked bar chart of timely and untimely responses for the top 10 companies, with specified labels and legend.
- 4.Legend indicates timely vs. untimely responses, positioned outside plot to prevent overlap.

COMPLAINTS ANALYSIS



1. Each bar represents a specific channel through which responses are submitted, allowing for a comparison of the distribution of responses across different channels.
2. The length of the bars indicates the frequency of responses submitted via each channel, with longer bars representing higher submission counts for that particular channel.
3. It's easy to identify which channels are most commonly used by consumers to submit responses, as they are represented by the bars on the y-axis.
4. The chart provides insights into consumers' preferred communication channels for submitting responses, which can inform companies about where to focus their efforts for effective communication with consumers.
5. Companies can use this information to assess the effectiveness of different communication channels and 2.

Given an unsorted array of integers, find the length of the longest continuous increasing subsequence (subarray).

COMPLAINTS ANALYSIS

2. Given an unsorted array of integers, find the length of the longest continuous increasing subsequence (subarray).

Example 1: Input: [1,3,5,4,7] Output: 3

Example 2: Input: [2,2,2,2,2] Output: 1

Example 1

```
a=[1,3,5,4,7]
n=len(a)
substr=[]
length=1
for i in range(0,n-1):
    if a[i]<a[i+1]:
        length+=1
    else:
        substr.append(length)
        length=1
substr.append(length)
print(max(substr))
```

Example 2

```
a=[2,2,2,2,2]
n=len(a)
substr=[]
length=1
for i in range(0,n-1):
    if a[i]<a[i+1]:
        length+=1
    else:
        substr.append(length)
        length=1
substr.append(length)
print(max(substr))
```

COMPLAINTS ANALYSIS

3. Given a list of non negative integers, arrange them such that they form the largest number.

Example 1:

Input: [10,2] Output: "210" Example 2:

Input: [3,30,34,5,9] Output: "9534330"

```
input1 = [10, 2]
input2 = [3, 30, 34, 5, 9]
input1 = [str(num) for num in input1]
input2 = [str(num) for num in input2]
input1.sort(key=lambda x: (x * 4)[:4], reverse=True)
input2.sort(key=lambda x: (x * 4)[:4], reverse=True)
output1 = ''.join(input1)
output2 = ''.join(input2)

print("Output for Example 1:", output1)
print("Output for Example 2:", output2)
```

4. Store all the "servlet-name", and "servlet-class" to a csv file from the attached sample_json.json file using Python.

```
import json
import csv

#local file path directory
pd.read_json(r"C:\Users\Sivashankar murugan\Downloads\DT A1 sample_json.json")

# Read data from JSON file
data = pd.read_json(r"C:\Users\Sivashankar murugan\Downloads\DT A1 sample_json.json")

# Extract "servlet-name" and "servlet-class" from each item in the JSON
servlet_data = [(servlet['servlet-name'], servlet['servlet-class'])
for servlet in data['web-app']['servlet']]

# Write to CSV file
with open('servlet_data.csv', 'w', newline='') as csv_file:
    writer = csv.writer(csv_file)
    # Write headers
    writer.writerow(['servlet-name', 'servlet-class'])
```


COMPLAINTS ANALYSIS

```
# Write data
writer.writerow(servlet_data)

print("CSV file created successfully.")
```