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

In [ ]:
```
def LLCIS(nums):
    if not nums:
        return 0

    longest_length = 1
    current_length = 1

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

    if current_length > longest_length:
        longest_length = current_length

    return longest_length

# Example usage:
Sample_input1=[1, 3, 5, 4, 7]
Sample_input2=[2, 2, 2, 2, 2]

#User Input code
#S=list(map(int,input().split(",")))


print(LLCIS(Sample_input1))
print(LLCIS(Sample_input2))
```

```
3
1
```

---

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"

In [ ]:
```
def largestNumber(array):
    #If there is only one element in the list, the element itself is the largest
    if len(array)==1:
        return str(array[0])

    #convert a list into a string array.
    for i in range(len(array)):
```

```
        array[i]=str(array[i])


    #find the largest element by swapping technique.
    for i in range(len(array)):
        for j in range(1+i,len(array)):
            if array[j]+array[i]>array[i]+array[j]:
                array[i],array[j]=array[j],array[i]


    #JOIN function in Python
    result=''.join(array)


    #If all elements are 0, answer must be 0
    if(result=='0'*len(result)):
        return '0'
    else:
        return result

# Example usage:
Sample_input1=[10,2]
Sample_input2=[3,30,34,5,9]


#User Input code
#S=list(map(int,input().split(",")))


print(largestNumber(Sample_input1))
print(largestNumber(Sample_input2))
```

```
210
9534330
```

---

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

In [ ]:

```
import json
import csv

Path="C:\\Users\\Hp\\Desktop\\adept ready\\DT A1 sample_json.json"
# Load JSON data from file
with open(Path) as f:
    data = json.load(f)

# Extract "servlet-name" and "servlet-class" from the JSON data
servlets = []
for servlet in data['web-app']['servlet']:
    servlet_name = servlet['servlet-name']
    servlet_class = servlet['servlet-class']
    servlets.append((servlet_name, servlet_class))

# Write to CSV file
with open('servlet_data.csv', 'w', newline='') as csvfile:
    fieldnames = ['servlet-name', 'servlet-class']

    # Create a CSV DictWriter object, specifying the field names
```

```
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

    # Write the header row containing the field names
    writer.writeheader()

    # Iterate over each servlet extracted from the JSON data
    for servlet in servlets:
        writer.writerow({'servlet-name': servlet[0], 'servlet-class': servlet[1]
```

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.

DATA PREPROCESSING

In [ ]:
```
import pandas as pd

Path="C:\\Users\\Hp\\Desktop\\adept ready\\complaints-2024-05-09_02_30.csv"

# Read CSV file into a Pandas DataFrame
df = pd.read_csv(Path)
```

In [ ]:
```
# checking Data types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1552333 entries, 0 to 1552332
Data columns (total 18 columns):
 #   Column                        Non-Null Count    Dtype
---  ------                        --------------    -----
 0   Date received                 1552333 non-null  object
 1   Product                       1552333 non-null  object
 2   Sub-product                   1552331 non-null  object
 3   Issue                         1552331 non-null  object
 4   Sub-issue                     1518925 non-null  object
 5   Consumer complaint narrative  452976 non-null   object
 6   Company public response       773839 non-null   object
 7   Company                       1552333 non-null  object
 8   State                         1548005 non-null  object
 9   ZIP code                      1552281 non-null  object
 10  Tags                          87945 non-null    object
 11  Consumer consent provided?    1265770 non-null  object
 12  Submitted via                 1552333 non-null  object
 13  Date sent to company          1552333 non-null  object
 14  Company response to consumer  1552321 non-null  object
 15  Timely response?              1552333 non-null  object
 16  Consumer disputed?            0 non-null        float64
 17  Complaint ID                  1552333 non-null  int64
dtypes: float64(1), int64(1), object(16)
memory usage: 213.2+ MB
```

In [ ]:
```
# Shape of the DataFrame
df.shape
```

Out[ ]:
```
(1552333, 18)
```

In [ ]:
```
# Display the DataFrame
df.head()
```

Out[ ]:

	Date received	Product	Sub-product	Issue	Sub-issue	Consumer complaint narrative	Company public response	
0	03/29/24	Credit reporting or other personal consumer re...	Credit reporting	Problem with a company's investigation into an...	Their investigation did not fix an error on yo...	NaN	NaN	EQ
1	04/06/24	Credit reporting or other personal consumer re...	Credit reporting	Problem with a company's investigation into an...	Their investigation did not fix an error on yo...	NaN	NaN	Sc
2	03/23/24	Credit reporting or other personal consumer re...	Credit reporting	Improper use of your report	Reporting company used your report improperly	NaN	NaN	Ea S
3	03/07/24	Credit reporting or other personal consumer re...	Credit reporting	Problem with a company's investigation into an...	Their investigation did not fix an error on yo...	I disputed a list of accounts that were not ve...	Company has responded to the consumer and the ...	TR INT
4	03/25/24	Credit reporting or other personal consumer re...	Credit reporting	Problem with a company's investigation into an...	Was not notified of investigation status or re...	I have consistently maintained on-time payment...	Company has responded to the consumer and the ...	TR INT

We have extracted data from the past year, resulting in a dataset comprising 1,552,333 rows and 18 columns.

In [ ]:
```
#Duplicate value identification
duplicate_rows = df[df.duplicated()]
print(len(duplicate_rows))
```

```
0
```

There is no duplicate rows presented in dataset.

In [ ]:
```
#Null value identification
df.isna().sum()
```

Out[ ]:
```
Date received                         0
Product                               0
Sub-product                           2
Issue                                 2
Sub-issue                         33408
Consumer complaint narrative    1099357
Company public response          778494
Company                               0
State                              4328
ZIP code                             52
Tags                            1464388
Consumer consent provided?       286563
Submitted via                         0
Date sent to company                  0
Company response to consumer         12
Timely response?                      0
Consumer disputed?              1552333
Complaint ID                          0
dtype: int64
```

Consumer complaint narrative: 1,099,357 missing values.

Company public response: 778,494 missing values.

Tags: 1,464,388 missing values.

Consumer consent provided?: 286,563 missing values.

Consumer disputed?: 1,552,333 missing values.

These are the few columns has highest count of Nan values.Since we dropping those columns for getting better insights about the data NOTE: Kindly visit the link attached below to know about each columns

https://www.consumerfinance.gov/complaint/data-use/

In [ ]:
```
df.drop(columns=["Consumer complaint narrative","Tags","Consumer consent provide
                 "Company public response","Consumer disputed?"],inplace=True)
```

In [ ]:
```
# Replace null values in specific columns with 'unknown'
columns_to_replace = ["Issue","Sub-product","Sub-issue","State","ZIP code","Comp
df[columns_to_replace] = df[columns_to_replace].fillna('unknown')
```

In [ ]:
```
df.shape
```

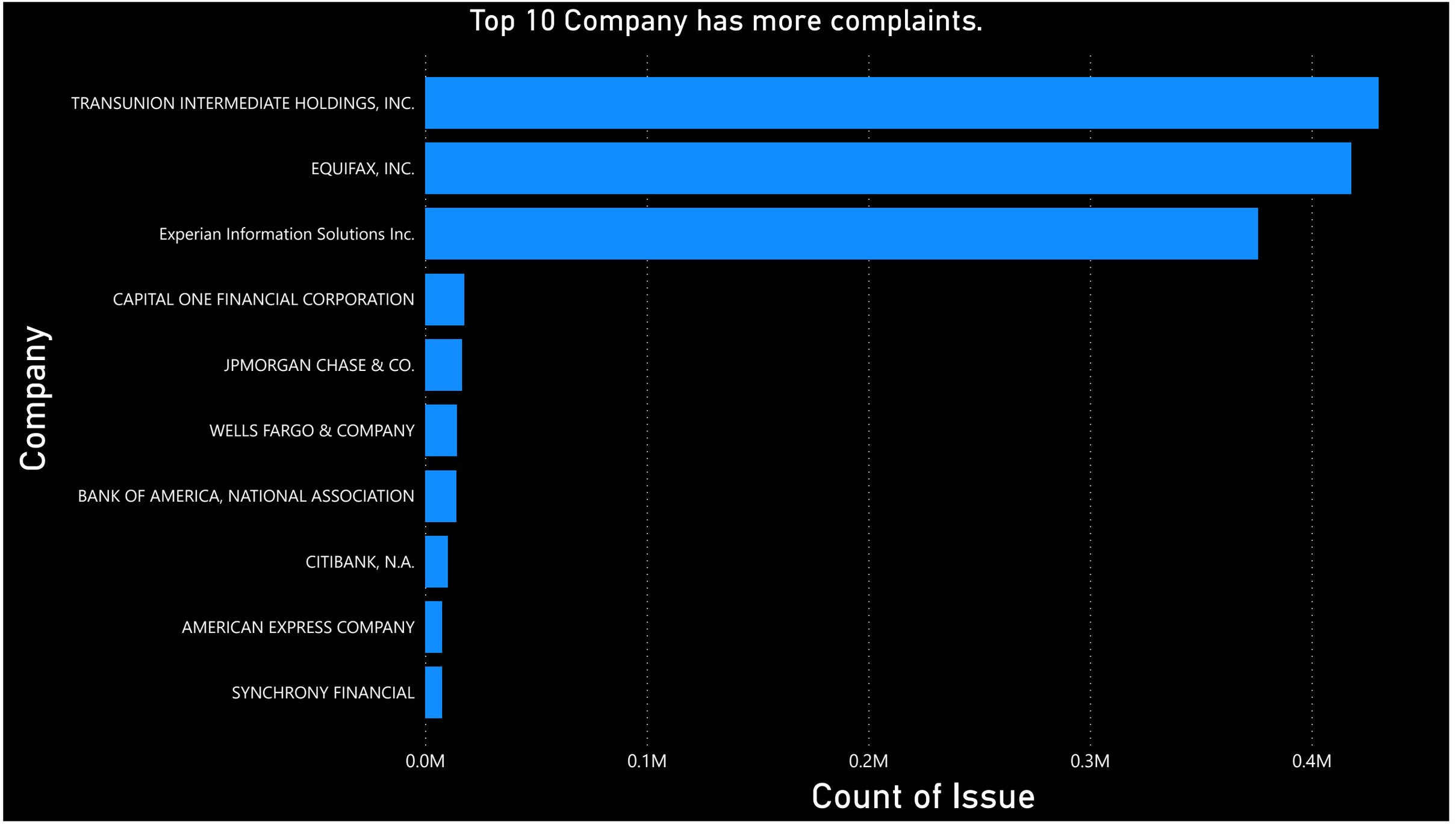
Out[ ]:
```
(1552333, 13)
```

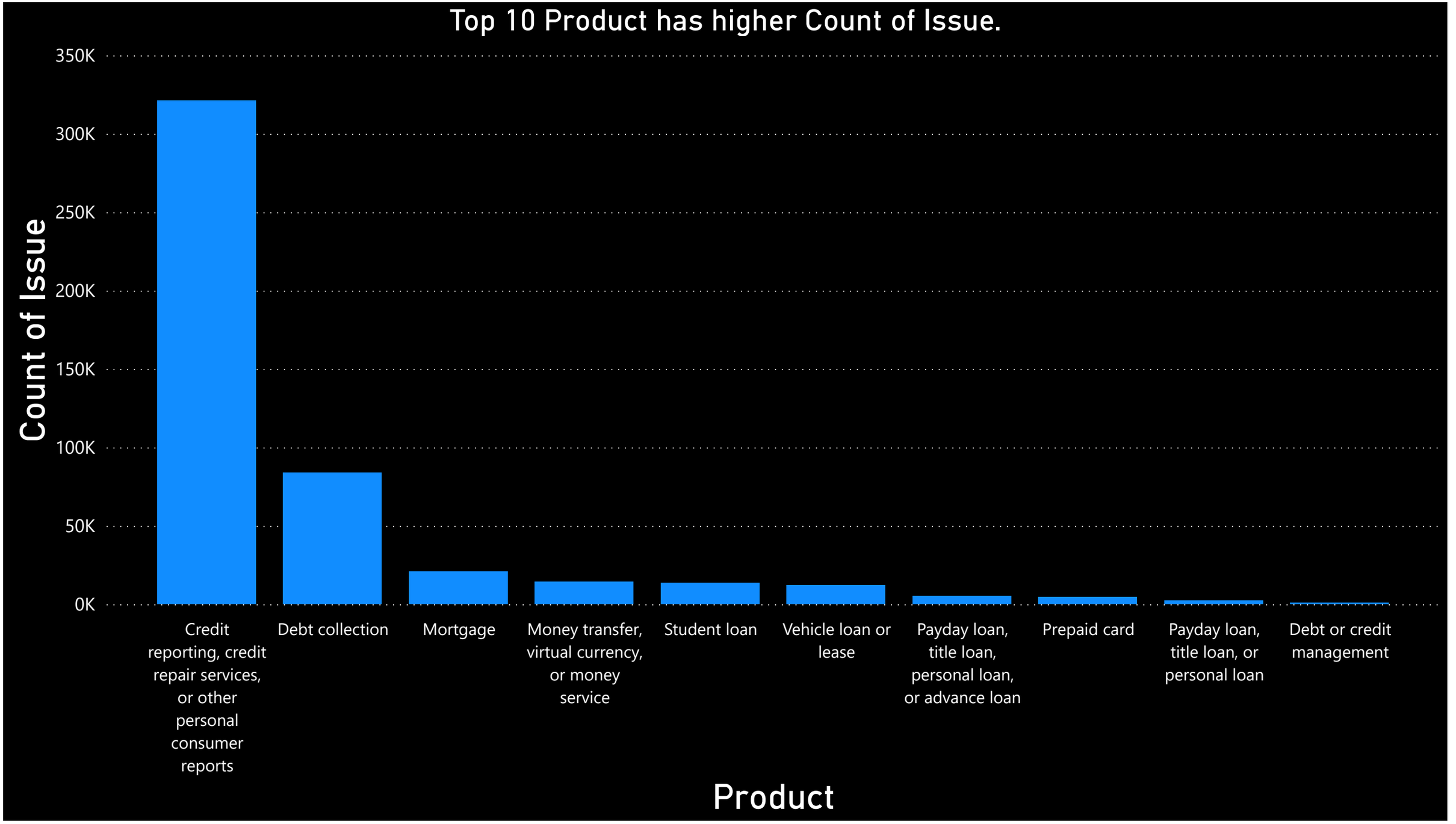
After preprocessing we got dataset of 1,552,333 rows and 13 columns.

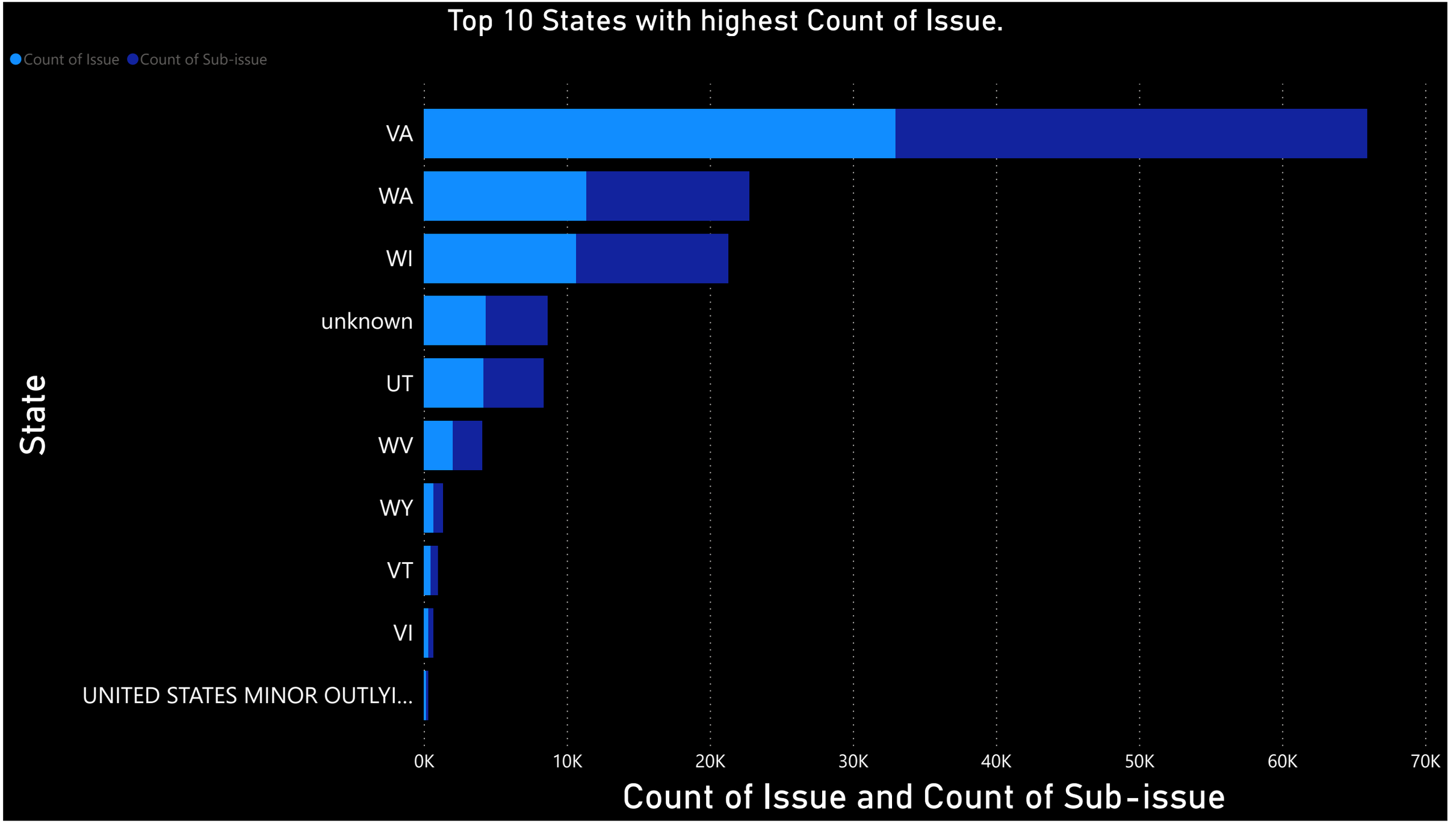
In [ ]:
```
df.isna().sum()
```

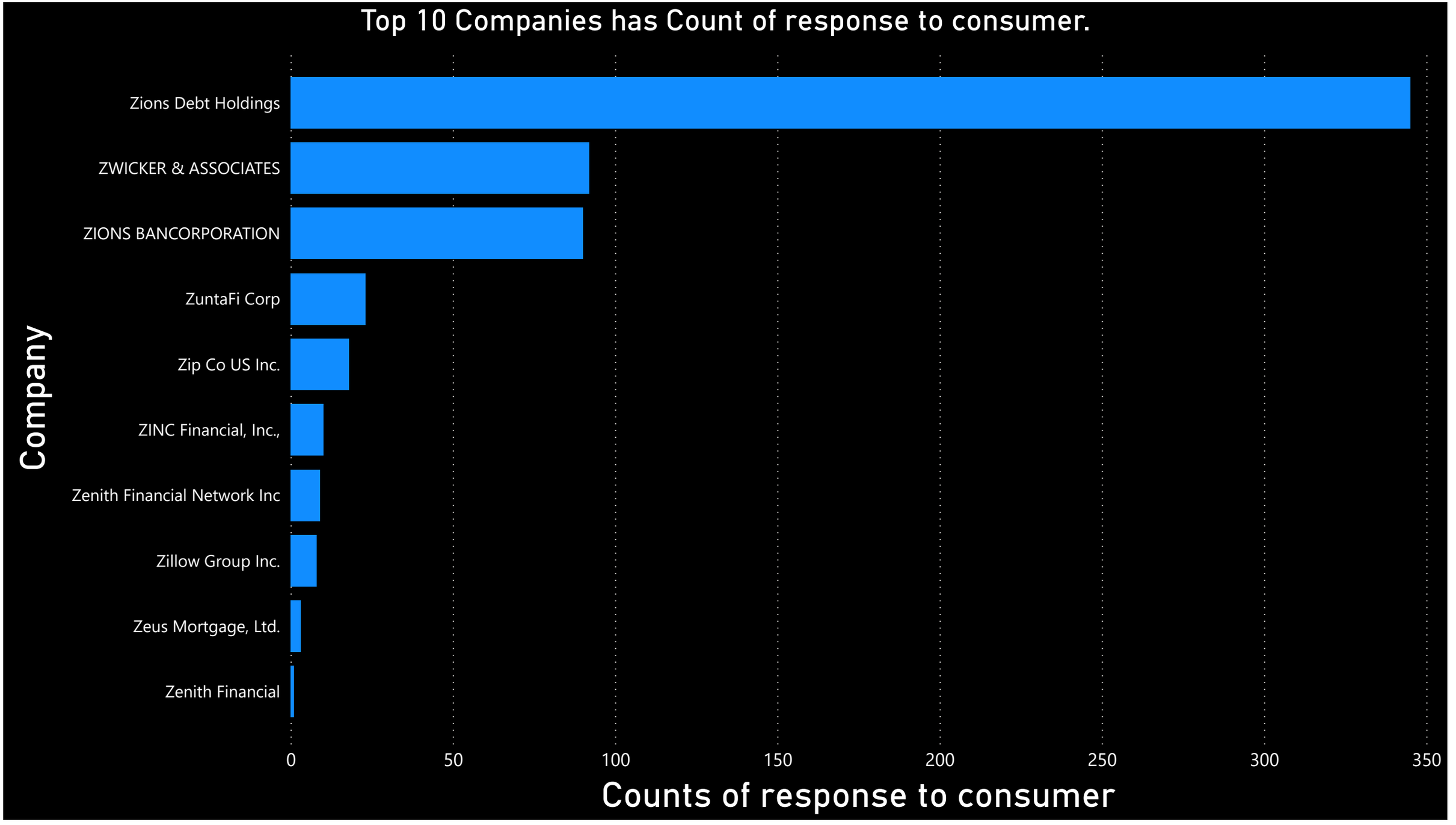
Out[ ]:
```
Date received                   0
Product                         0
Sub-product                     0
Issue                           0
Sub-issue                       0
Company                         0
State                           0
ZIP code                        0
Submitted via                   0
Date sent to company            0
Company response to consumer    0
Timely response?                0
Complaint ID                    0
dtype: int64
```

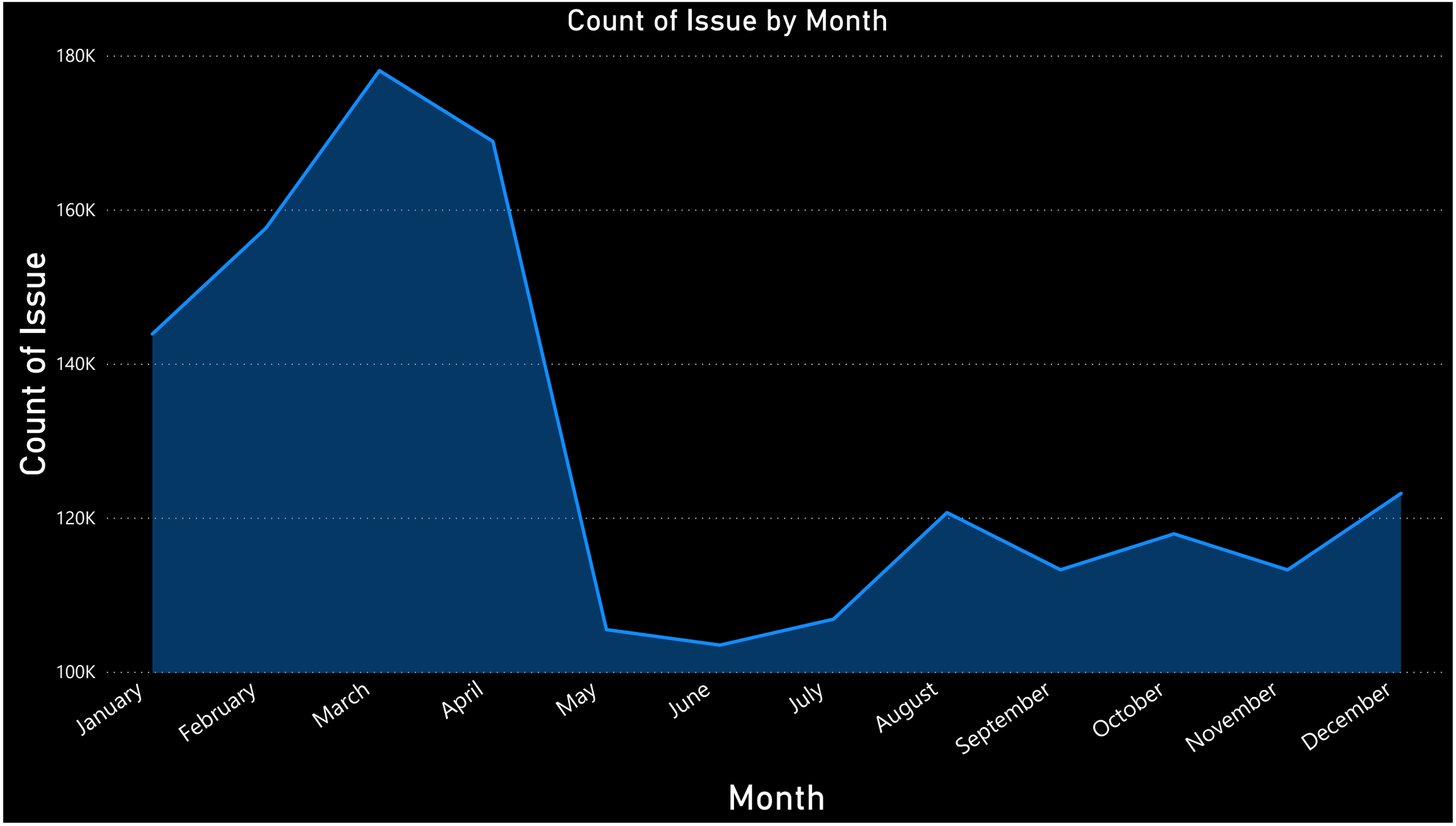
In [ ]:
```
file_path = 'complaints_data.csv'
df.to_csv(file_path, index=False)
```


Top 10 Company has more complaints.
TRANSUNION INTERMEDIATE HOLDINGS, INC.
EQUIFAX, INC.
Experian Information Solutions Inc.
CAPITAL ONE FINANCIAL CORPORATION
JPMORGAN CHASE & CO.
WELLS FARGO & COMPANY
BANK OF AMERICA, NATIONAL ASSOCIATION
CITIBANK, N.A.
AMERICAN EXPRESS COMPANY
SYNCHRONY FINANCIAL
Company
0.0M
0.1M
0.2M
0.3M
0.4M
Count of Issue


Top 10 Product has higher Count of Issue.
350K
300K
250K
200K
150K
100K
50K
0K
Count of Issue
Credit reporting, credit repair services, or other personal consumer reports
Debt collection
Mortgage
Money transfer, virtual currency, or money service
Student loan
Vehicle loan or lease
Payday loan, title loan, personal loan, or advance loan
Prepaid card
Payday loan, title loan, or personal loan
Debt or credit management
Product


Top 10 States with highest Count of Issue.
Count of Issue
Count of Sub-issue
State
VA
WA
WI
unknown
UT
WV
WY
VT
VI
UNITED STATES MINOR OUTLYI...
0K
10K
20K
30K
40K
50K
60K
70K
Count of Issue and Count of Sub-issue


Top 10 Companies has Count of response to consumer.
Company
Zions Debt Holdings
ZWICKER & ASSOCIATES
ZIONS BANCORPORATION
ZuntaFi Corp
Zip Co US Inc.
ZINC Financial, Inc.,
Zenith Financial Network Inc
Zillow Group Inc.
Zeus Mortgage, Ltd.
Zenith Financial
0
50
100
150
200
250
300
350
Counts of response to consumer


Count of Issue by Month
180K
160K
140K
120K
100K
Count of Issue
January
February
March
April
May
June
July
August
September
October
November
December
Month