



# complaints

Name: Abdelrhman Mahmoud Ibrahim

Email:body442012@gmail.com

# Introduction:

This data talks about issuance submitted in companies in all the United States, where these complaints are received from the government or the entity responsible for receiving complaints from consumers and then sent to the companies

# Read\_CSV

```
import pandas as pd
import numpy as np
import plotly.express as px

/ 0.5s Python

+ Code + Markdown

df = pd.read_csv('complaints.csv')

/ 42.6s Python
```

df

0.0s

Python

	Date received	Product	Sub-product	Issue	Sub-issue	Consumer complaint narrative	Company public response	Company	State	ZIP code	Tags	Consumer consent provided?	Submitted via	Date sent to company	Company response to consumer	Timely response?	Consumer disputed?	Complaint ID
0	2023-06-13	Credit reporting, credit repair services, or o...	Credit reporting	Improper use of your report	Credit inquiries on your report that you don't...	NaN	NaN	TRANSUNION INTERMEDIATE HOLDINGS, INC.	NY	11213	NaN	NaN	Web	2023-06-13	In progress	Yes	NaN	7112068
1	2023-03-02	Payday loan, title loan, or personal loan	Personal line of credit	Problem when making payments	NaN	NaN	Company has responded to the consumer and the ...	SECURITY SERVICE FEDERAL CREDIT UNION	TX	78214	NaN	NaN	Phone	2023-03-02	Closed with monetary relief	Yes	NaN	6638236
2	2023-03-01	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Information belongs to someone else	NaN	Company has responded to the consumer and the ...	Experian Information Solutions Inc.	PA	19104	NaN	Consent not provided	Web	2023-03-02	Closed with explanation	Yes	NaN	6627419
3	2023-06-08	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Information belongs to someone else	NaN	NaN	Experian Information Solutions Inc.	FL	34203	NaN	NaN	Web	2023-06-08	In progress	Yes	NaN	7082884
4	2023-06-01	Debt collection	Credit card debt	Attempts to collect debt not owed	Debt is not yours	NaN	NaN	CAPITAL ONE FINANCIAL CORPORATION	AZ	85015	NaN	NaN	Web	2023-06-01	In progress	Yes	NaN	7051611
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3745849	2013-01-09	Credit reporting	NaN	Incorrect information on credit report	Account terms	NaN	NaN	Experian Information Solutions Inc.	IL	60071	NaN	NaN	Web	2013-01-09	Closed with explanation	Yes	No	226212
3745850	2013-02-28	Mortgage	Other mortgage	Loan modification, collection, foreclosure	NaN	NaN	NaN	Mr. Cooper Group Inc.	NY	11412	Older American	NaN	Referral	2013-03-01	Closed with explanation	No	Yes	338457

# Solve NULL problems

```
Date received      0.000000
Product            0.000000
Sub-product        0.062814
Issue              0.000000
Sub-issue          0.189718
Consumer complaint narrative 0.638408
Company public response 0.545112
Company            0.000000
State              0.011102
ZIP code           0.008061
Tags               0.892386
Consumer consent provided? 0.234596
Submitted via      0.000000
Date sent to company 0.000000
Company response to consumer 0.000001
Timely response?   0.000000
Consumer disputed? 0.794878
Complaint ID       0.000000
dtype: float64
```

We will delete the columns(Consumer disputed ? ,Consumer consent provided?,  
Tags, Company public response ,Consumer complaint narrative)  
Because the Null percentage is more than 20%

```
df.drop(['Consumer disputed?'], axis = 1, inplace = True )
```

✓ 1.2s

```
df.drop(['Consumer consent provided?'], axis = 1, inplace = True )
```

✓ 0.8s

```
df.drop(['Tags',"Company public response",'Consumer complaint narrative'], axis = 1, inplace = True )
```

✓ 1.0s

# State

Here in this column we have more than 40,000 null we have a number of choices ▶

1- Put the most state in the iteration (mode) ▶

2- Create a new category called Unknown ▶

3-Delete the 40,000 as they did not affect much on the data because their number is more than 3 million ▶

In this situation, the decision depends on going to the person who obtained this data. ▶

```
df.dropna(subset=['State'], inplace=True)
```

✓ 1.2s

# ZIP code

This column depends on the State column and with some analysis I noticed that this column is not used ▶

```
df.drop(['ZIP code'] , axis = 1 , inplace = True )
```

✓ 0.7s

# Company response to consumer

Here I converted the NULL values to X, then we searched for all the rows and used the name of each company and searched by the name of the company for the most common way the company used to respond to customers

[31]	df["Company response to consumer"].fillna("X", inplace=True)													Python
[32]	df[df["Company response to consumer"]=="X"]													Python
...														
		Date received	Product	Sub-product	Issue	Sub-issue	Company	State	ZIP code	Submitted via	Date sent to company	Company response to consumer	Timely response?	Complaint ID
	89541	2023-05-26	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	92860	Web	2023-05-26	x	Yes	7035005
	661122	2021-10-21	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Information belongs to someone else	CITIBANK, N.A.	CA	91201	Web	2021-10-21	x	Yes	4832902
	1218905	2022-07-26	Credit reporting, credit repair services, or o...	Credit reporting	Improper use of your report	Reporting company used your report improperly	Experian Information Solutions Inc.	WV	254XX	Web	2022-07-26	x	Yes	5813036
	1595090	2021-11-30	Credit reporting, credit repair services, or o...	Credit reporting	Improper use of your report	Credit inquiries on your report that you don't...	Experian Information Solutions Inc.	OH	43611	Web	2021-11-30	x	Yes	4960323
	1711116	2018-01-03	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Old information reappears or never goes away	EQUIFAX, INC.	CA	94588	Web	2018-01-03	x	Yes	2770980
[33]	df[df["Company"]=="E Mortgage Capital, Inc."]													Python
...														
		Date received	Product	Sub-product	Issue	Sub-issue	Company	State	ZIP code	Submitted via	Date sent to company	Company response to consumer	Timely response?	Complaint ID
	89541	2023-05-26	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	92860	Web	2023-05-26	x	Yes	7035005
	1009956	2022-03-22	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	90025	Web	2022-03-22	Closed with explanation	Yes	5353204
	1024129	2022-02-26	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	TX	77316	Web	2022-02-26	Closed with explanation	Yes	5265342
	1127747	2021-02-25	Mortgage	Conventional home mortgage	Applying for a mortgage or refinancing an exis...	NaN	E Mortgage Capital, Inc.	WY	820XX	Web	2021-02-25	Closed with explanation	Yes	4166478
	1351170	2022-03-15	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	90025	Web	2022-04-11	Closed with explanation	Yes	5324323



# Company response to consumer

the mode is Closed with explanation

```
[34] a=df[(df["Company"]=="E Mortgage Capital, Inc.")&(df["Company response to consumer"]=="x").index
```

Python

```
[35] df['Company response to consumer'].iloc[a]="Closed with explanation"
```

Python

... `c:\Anaconda\lib\site-packages\pandas\core\indexing.py:1732: SettingWithCopyWarning:`  
A value is trying to be set on a copy of a slice from a DataFrame  
  
See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
`self._setitem_single_block(indexer, value, name)`

```
[36] df[df["Company"]=="E Mortgage Capital, Inc."]
```

Python

...

	Date received	Product	Sub-product	Issue	Sub-issue	Company	State	ZIP code	Submitted via	Date sent to company	Company response to consumer	Timely response?	Complaint ID
89541	2023-05-26	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	92860	Web	2023-05-26	Closed with explanation	Yes	7035005
1009956	2022-03-22	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	90025	Web	2022-03-22	Closed with explanation	Yes	5353204
1024129	2022-02-26	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	TX	77316	Web	2022-02-26	Closed with explanation	Yes	5265342
1127747	2021-02-25	Mortgage	Conventional home mortgage	Applying for a mortgage or refinancing an exis...	NaN	E Mortgage Capital, Inc.	WY	820XX	Web	2021-02-25	Closed with explanation	Yes	4166478
1351170	2022-03-15	Mortgage	Conventional home mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	90025	Web	2022-04-11	Closed with explanation	Yes	5324323
1505419	2021-11-24	Mortgage	Conventional home mortgage	Applying for a mortgage or refinancing an exis...	NaN	E Mortgage Capital, Inc.	CA	92126	Web	2021-11-24	Closed with explanation	Yes	4944180
1634436	2021-08-12	Mortgage	Reverse mortgage	Closing on a mortgage	NaN	E Mortgage Capital, Inc.	CA	90304	Phone	2021-08-12	Closed with explanation	Yes	4626620
1847290	2020-08-03	Mortgage	Conventional home mortgage	Applying for a mortgage or refinancing an exis...	NaN	E Mortgage Capital, Inc.	CA	92602	Web	2020-08-03	Closed with explanation	Yes	3774526

# Sub-issue

# Sub-product

First we will use(fillna) and then we will use apply\_funcation ►

```
df.fillna(0, inplace=True)
```

In Sub-issue we will create a new category and call it "no issue"

```
df['Sub-issue'] = df['Sub-issue'].apply(lambda x: 'no issue' if x == 0 else x)
```

✓ 0.6s

In Sub-product we will create a new category and call it "Unkown"

```
df['Sub-product'] = df['Sub-product'].apply(lambda x: 'Unkown' if x == 0 else x)
```

✓ 0.5s

# Data Type

Modify the data type of the date ►

```
df['Date received'] = pd.to_datetime(df['Date received'])
```

✓ 1.3s

```
df['Date sent to company'] = pd.to_datetime(df['Date sent to company'])
```

✓ 0.8s

Dividing Date received into 3 columns, a column for the day, a column for the month, and another for the year ▶

```
df["day"]=df["Date received"].dt.day
```

✓ 0.4s

```
df["month"]=df["Date received"].dt.month
```

✓ 0.3s

```
df["year"]=df["Date received"].dt.year
```

✓ 0.4s

Delete Date received

```
df.drop(['Date received'] , axis = 1 , inplace = True )
```

✓ 0.5s

Divide the year into 4 seasons ►

```
def map_months(x):  
    if x in [12, 1, 2]:  
        return 'Winter'  
    elif x in [3, 4, 5]:  
        return 'Spring'  
    elif x in [6, 7, 8]:  
        return 'Summer'  
    elif x in [9, 10, 11]:  
        return 'Autumn'  
  
df['Season'] = df['month'].apply(map_months)
```

Create a new column to calculate the difference between receiving the complaint and sending it to the company ▶

```
df['The difference between receiving a Issue and sending it(Days)']=df["Date sent to company"]-df["Date received"]
```

✓ 0.1s

Create Excel files in 2023, 2022, 2021, 2011 and 2012 ▶

```
df_2023=df[df["year"]==2023]
```

✓ 0.1s

```
df_2023.to_csv("2023.csv")
```

✓ 16.1s

```
df_2022=df[df["year"]==2022]
```

✓ 0.2s

```
df_2021=df[df["year"]==2021]
```

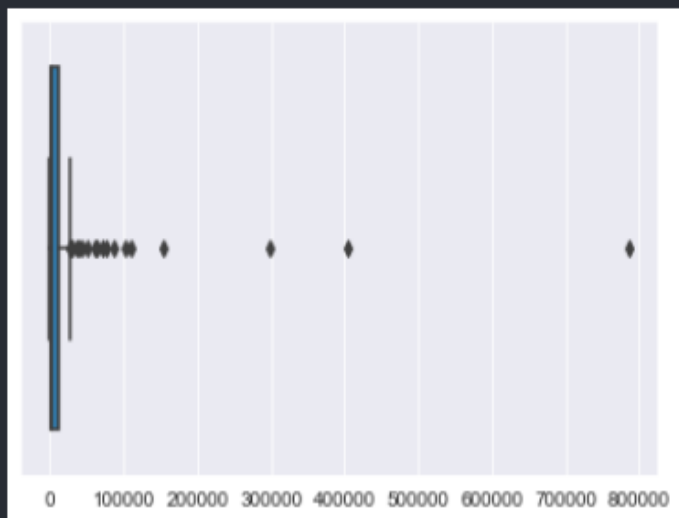
✓ 0.1s

# outliers

```
c=df["Issue"].value_counts()
sns.boxplot(data= df, x= c.values) # there are many outliers
```

✓ 0.3s

AxisSubplot:>



Incorrect information on your report	943207
Problem with a credit reporting company's investigation into an existing problem	522466
Improper use of your report	449231
Attempts to collect debt not owed	170798
Loan modification, collection, foreclosure	111484
...	...
Property was sold	9
Lender damaged or destroyed vehicle	8
Property was damaged or destroyed property	7
Lender sold the property	7
Lender damaged or destroyed property	3

Name: Issue, Length: 165, dtype: int64



In my opinion, there is no “*Outliers*” here because we are talking about the most frequently repeated Issue, and it is noticeable that it has been repeated in a large proportion, as this Issue may be the biggest problem facing consumers ▶

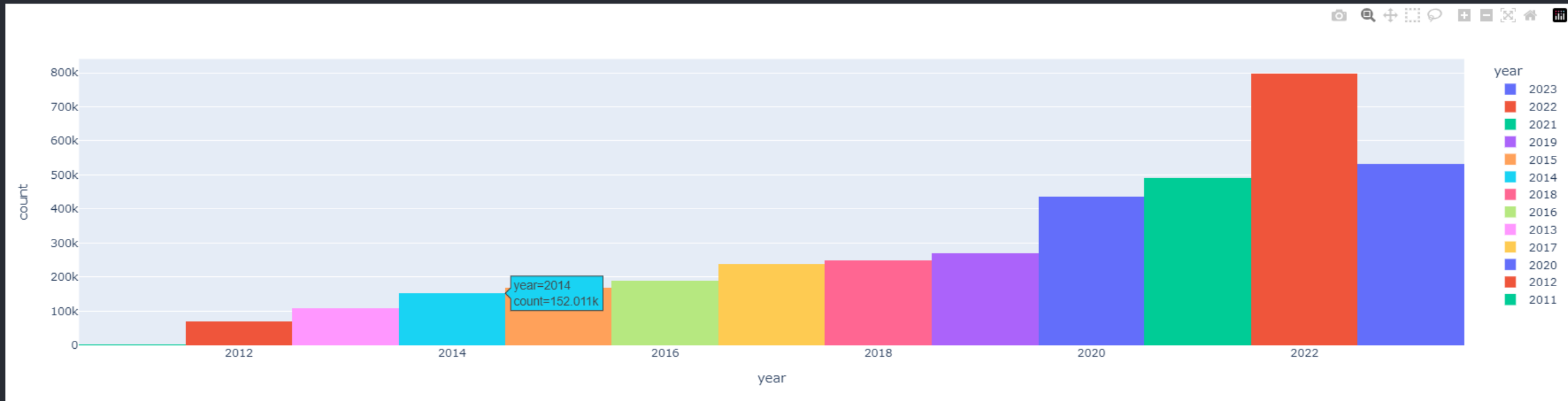


The most year in which Issue were submitted was 2022 ▶

```
px.histogram(df,x="year",hover_data=df.columns,color="year",nbins=13)
```

✓ 14.9s

Python

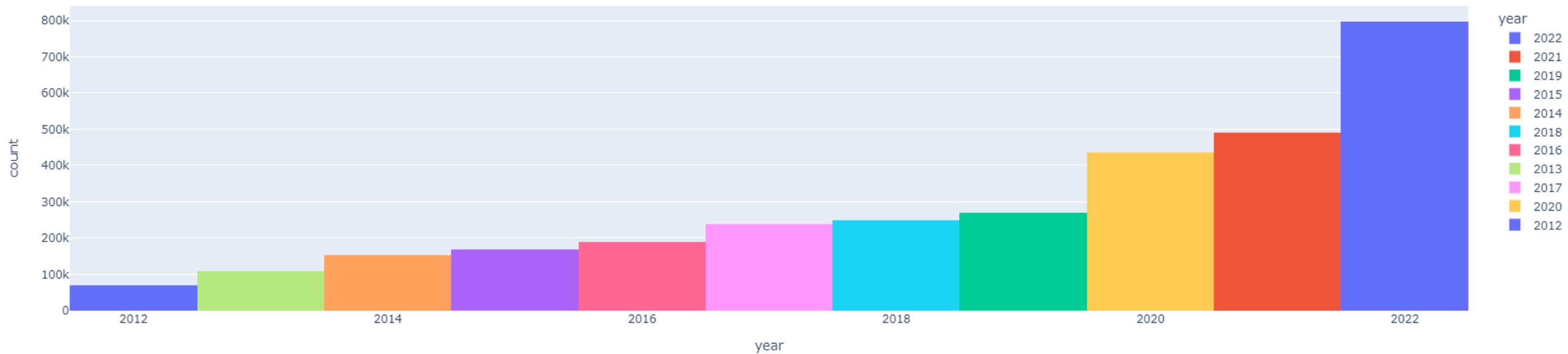
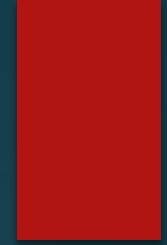


An increase in the number of Issue is observed each year compared to the previous year ▶

Note: In 2023, this data is collected until month 6 only ▶

Note: In 2011 it only contains the month of December ▶

The distribution of complaints over the years after deleting the years 2023 and 2011 because it is incomplete, and I think that this will be effective because I compare the full years to some



It was found that most of the issuance were in the state of California with the number 442833 ▶

```
df["State"].value_counts().head(1)
```

✓ 0.3s

CA	442833
----	--------

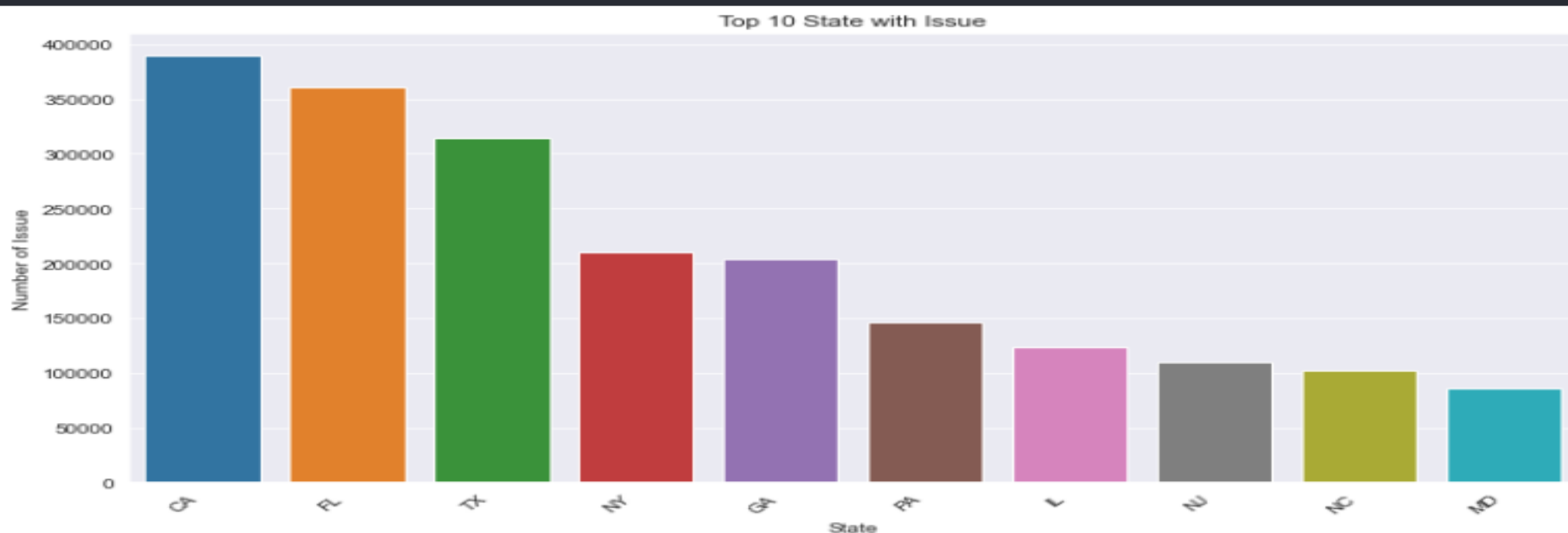
IT WAS FOUND THAT THE COMPANY WITH THE MOST ISSUANCE WAS EQUIFAX, INC. WITH 691029 ▶

```
df["Company"].value_counts()
```

✓ 0.3s

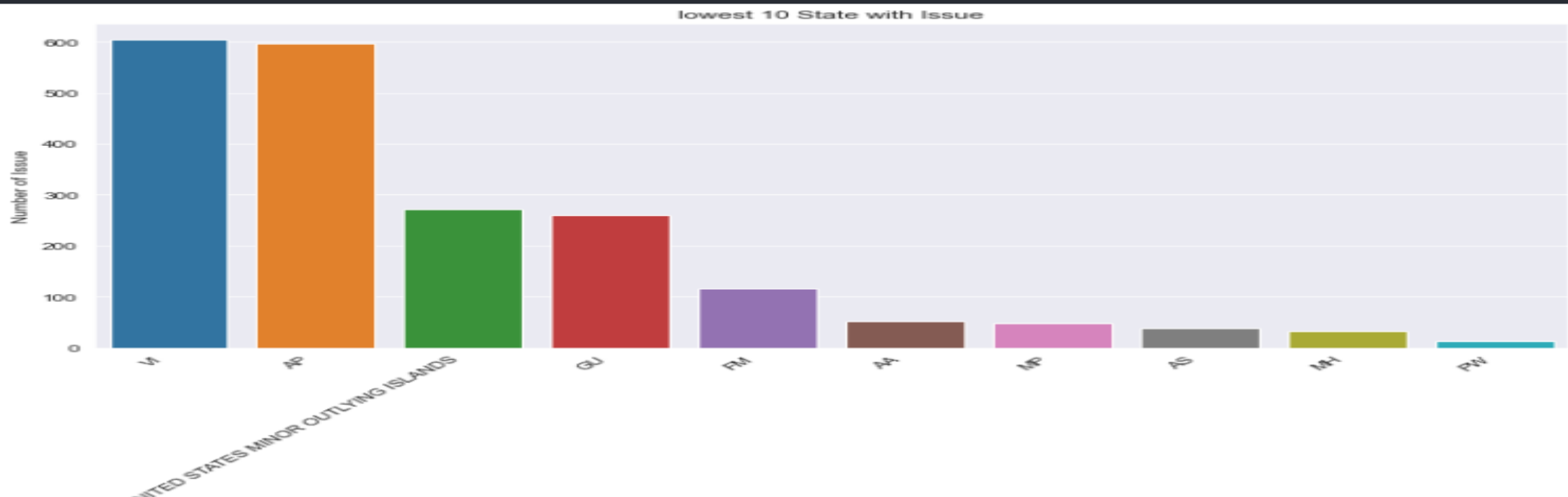
EQUIFAX, INC.	691029
TRANSUNION INTERMEDIATE HOLDINGS, INC.	600448
Experian Information Solutions Inc.	557580
BANK OF AMERICA, NATIONAL ASSOCIATION	124756
WELLS FARGO & COMPANY	113817
...	
Lender Services, Inc	1
Fulkerson Wing, P.C.	1
A1 Solutions Group Inc	1
Venanzi Law Office	1
Dellwo, Roberts & Scanlon, P.S.	1

```
g = df.groupby("State")["Issue"].count().reset_index()
s = g.sort_values("Issue", ascending=False)
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")
plt.figure(figsize=(12, 6))
sns.barplot(x="State", y="Issue", data=s.head(10))
plt.title("Top 10 State with Issue")
plt.xlabel("State")
plt.ylabel("Number of Issue")
plt.xticks(rotation=45, ha='right')
plt.show()
```



# It was found that the least issuance were in the state PW

```
g = df.groupby("State")["Issue"].count().reset_index()
s = g.sort_values("Issue", ascending=False)
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("darkgrid")
plt.figure(figsize=(12, 6))
sns.barplot(x="State", y="Issue", data=s.tail(10))
plt.title("lowest 10 State with Issue")
plt.xlabel("State")
plt.ylabel("Number of Issue")
plt.xticks(rotation=45, ha='right')
plt.show()
```



# Top 10 companies with most issuance

```
grouped = df.groupby("Company")["Issue"].count().reset_index()
```

✓ 1.4s

```
sorted_Company = grouped.sort_values("Issue", ascending=False)
```

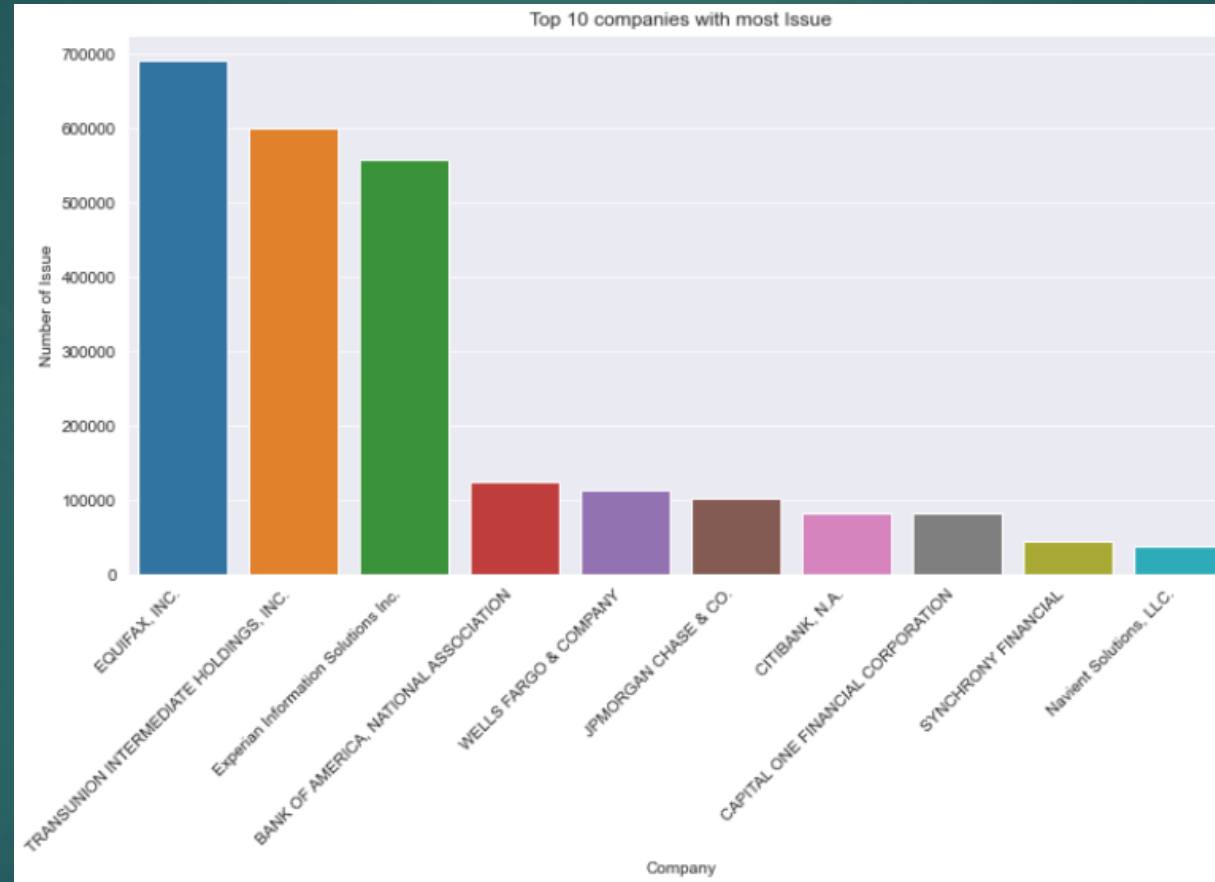
✓ 0.0s

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

```
sns.set_style("darkgrid")
plt.figure(figsize=(12, 6))
sns.barplot(x="Company", y="Issue", data=sorted_Company.head(10))
plt.title("Top 10 companies with most Issue")
plt.xlabel("Company")
plt.ylabel("Number of Issue")
plt.xticks(rotation=45, ha='right')
plt.show()
```

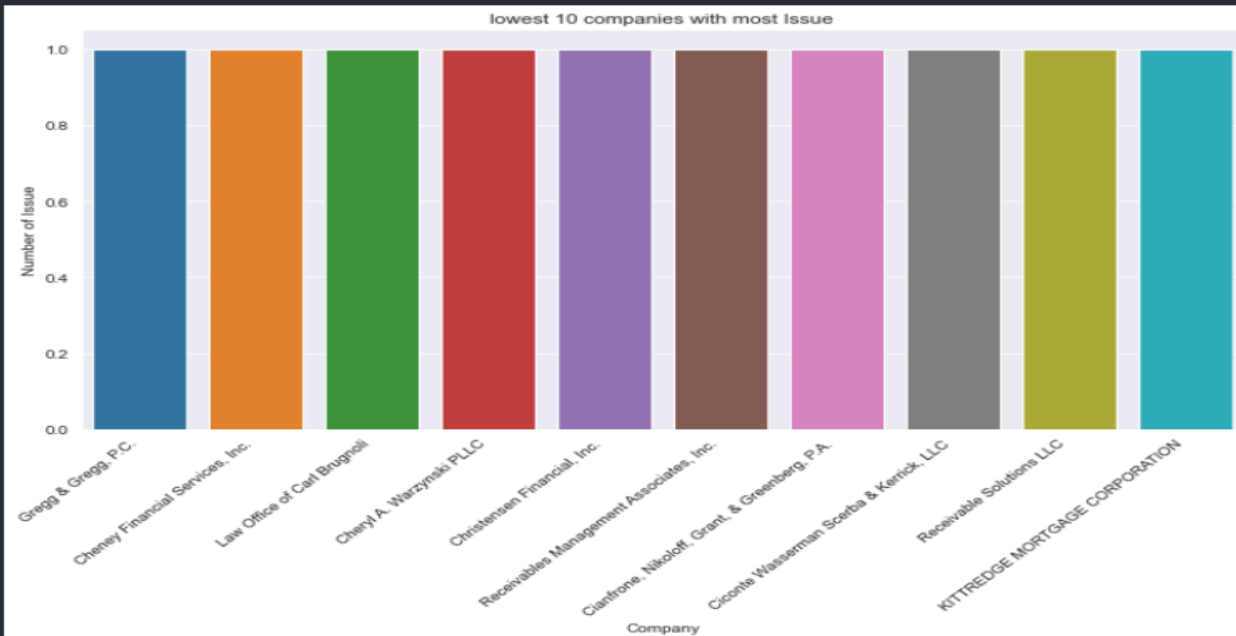
✓ 0.1s

# Top 10 companies with most issuance



# The lowest 10 companies with issuance

```
sns.set_style("darkgrid")
plt.figure(figsize=(12, 6))
sns.barplot(x="Company", y="Issue", data=sorted_Company.tail(10))
plt.title("lowest 10 companies with most Issue")
plt.xlabel("Company")
plt.ylabel("Number of Issue")
plt.xticks(rotation=45, ha='right')
plt.show()
```



Company	Issue
Gregg & Gregg, P.C.	1
Cheney Financial Services, Inc.	1
Law Office of Carl Brugnoli	1
Cheryl A. Warzynski PLLC	1
Christensen Financial, Inc.	1
Receivables Management Associates, Inc.	1
Cianfrone, Nikoloff, Grant, & Greenberg, P.A.	1
Ciconte Wasserman Scerba & Kerrick, LLC	1
Receivable Solutions LLC	1
KITTREDGE MORTGAGE CORPORATION	1



the most Issue submitted “Incorrect information on your report”

```
df.groupby("Issue").count()["Product"].sort_values(ascending=False)
```



✓ 3.6s

Issue	
Incorrect information on your report	943205

the least Issue submitted” Lender damaged or destroyed property”

```
df.groupby("Issue").count()["Product"].sort_values(ascending=True)
```

✓ 3.7s

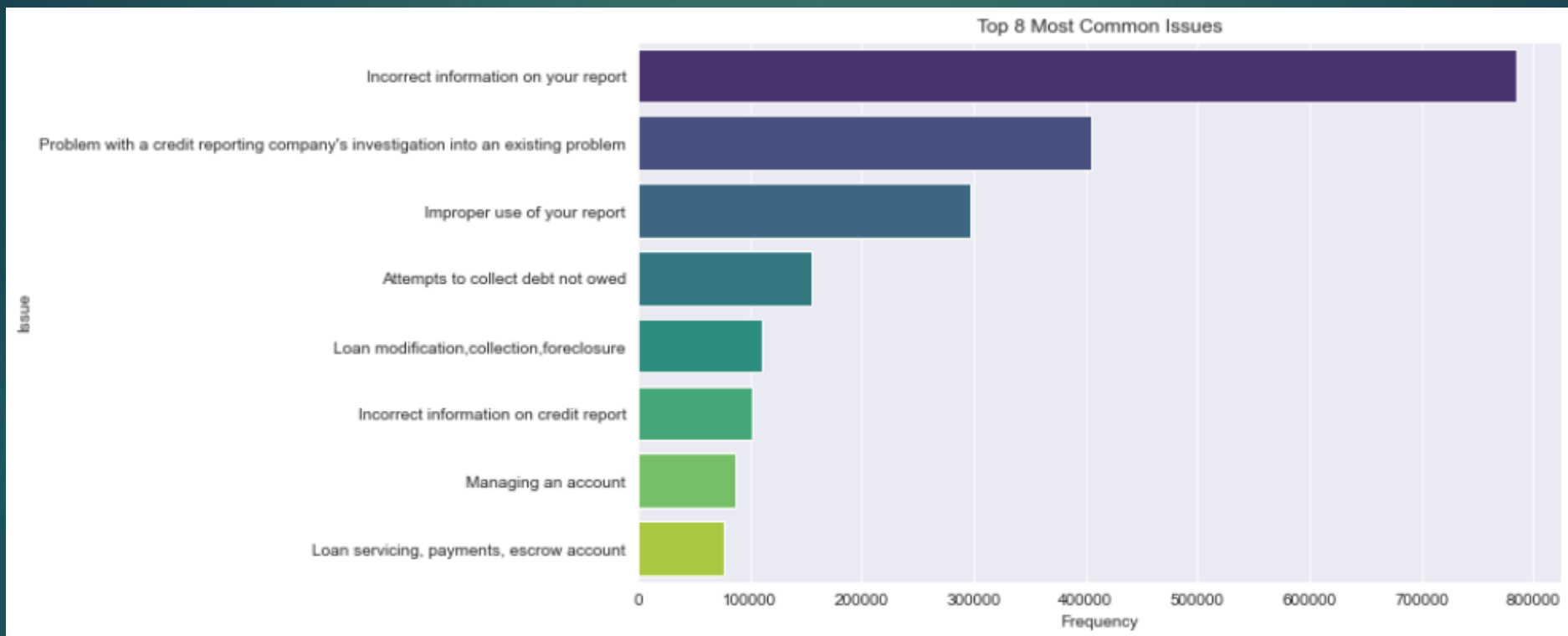
Issue	
Lender damaged or destroyed property	3

Most of the Issues were sent to the company on the same day by 82%. ▶

```
df["The difference between receiving a Issue and sending it(Days)"].value_counts()/df["The difference between receiving a Issue and sending it(Days)"].count()*100
```

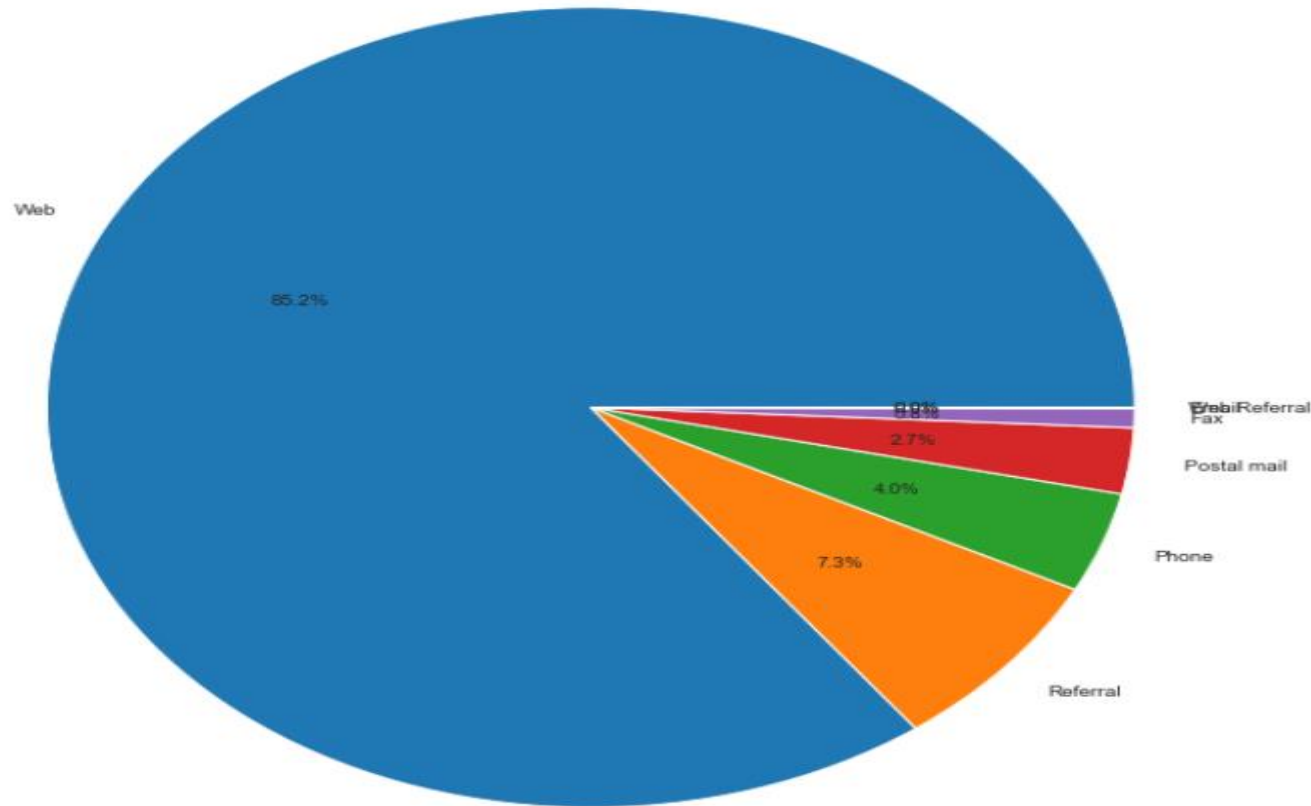
✓ 0.0s

0 days	82.173151
1 days	4.326477



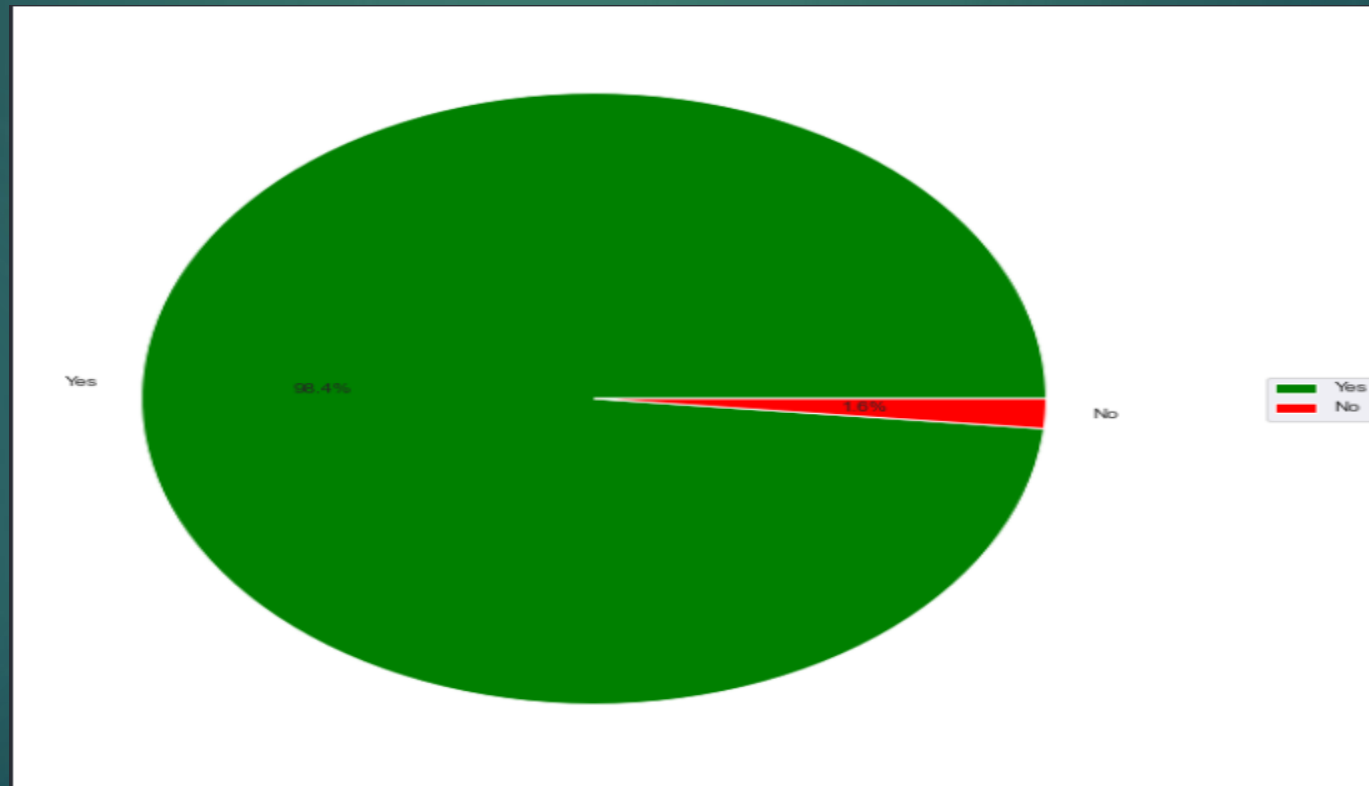
The most common way in which a complaint was submitted was through the web  
By 85.2% ▶

```
plt.figure(figsize=(12,12))
plt.pie(df["Submitted via"].value_counts(),labels=df["Submitted via"].value_counts().index.values,autopct='%1.1f%%')
```

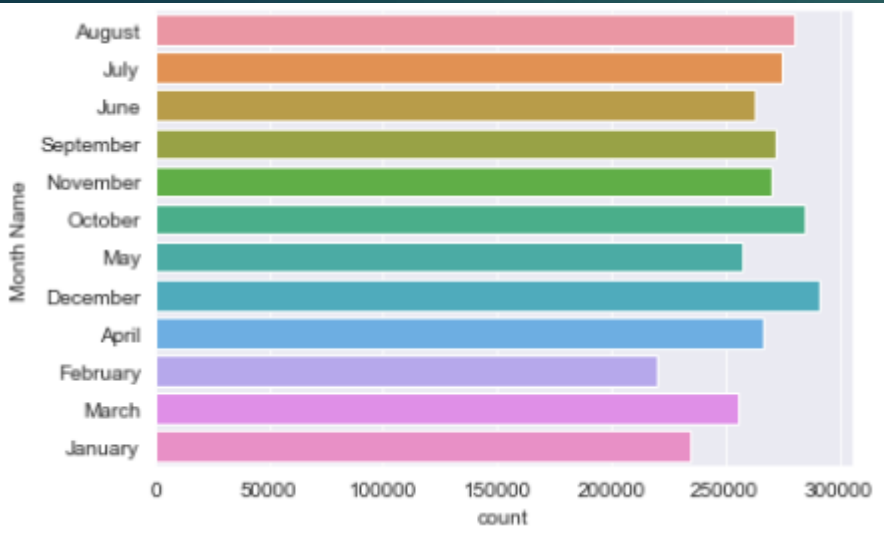


# It was found that the rate of timely response 98.4%

Timely response? ►



December is the month in which Issues are submitted ▶

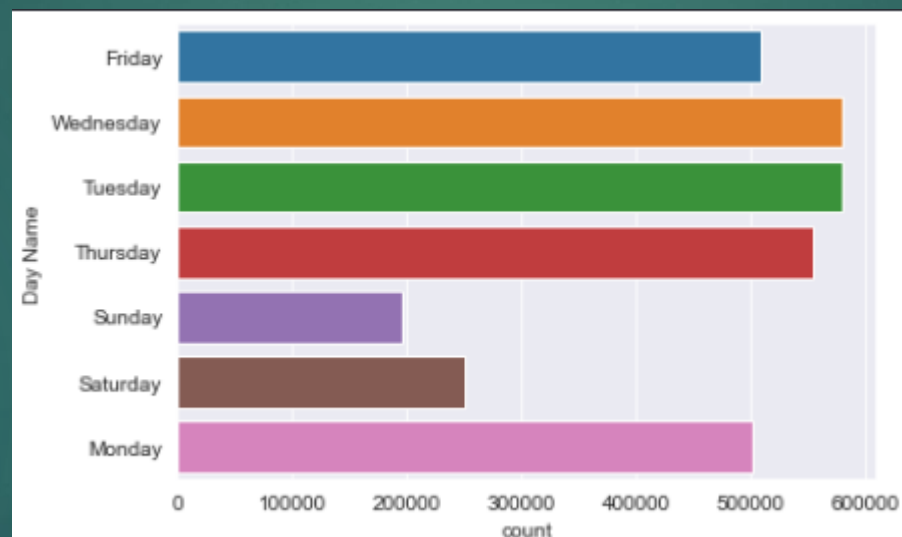


Month Name	Issue
April	266366
August	279959
December	291124
February	220435
January	234727
July	274590
June	262966
March	255809
May	257590
November	270085
October	285232
September	272385

Product	Company
Credit reporting, credit repair services, or o...	1940721
Debt collection	484515
Mortgage	373262
Credit card or prepaid card	187822
Checking or savings account	163494
Credit reporting	139956
Credit card	88476
Bank account or service	84649
Student loan	73701
Money transfer, virtual currency, or money ser...	49075
Vehicle loan or lease	43483
Consumer Loan	31394
Payday loan, title loan, or personal loan	28206
Payday loan	5521
Money transfers	5155
Prepaid card	3776
Other financial service	1038
Virtual currency	17

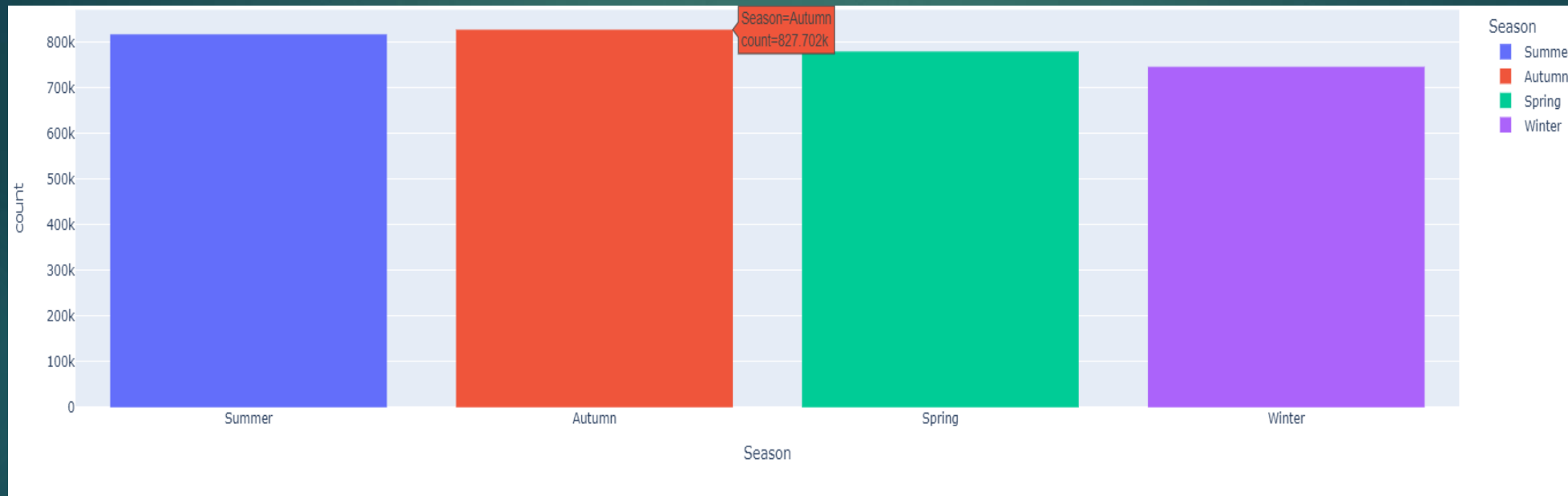
The most product that receives Issues is “Credit reporting, credit repair services, or other personal consumer reports” through 1940721 company

# Issue distribution by day

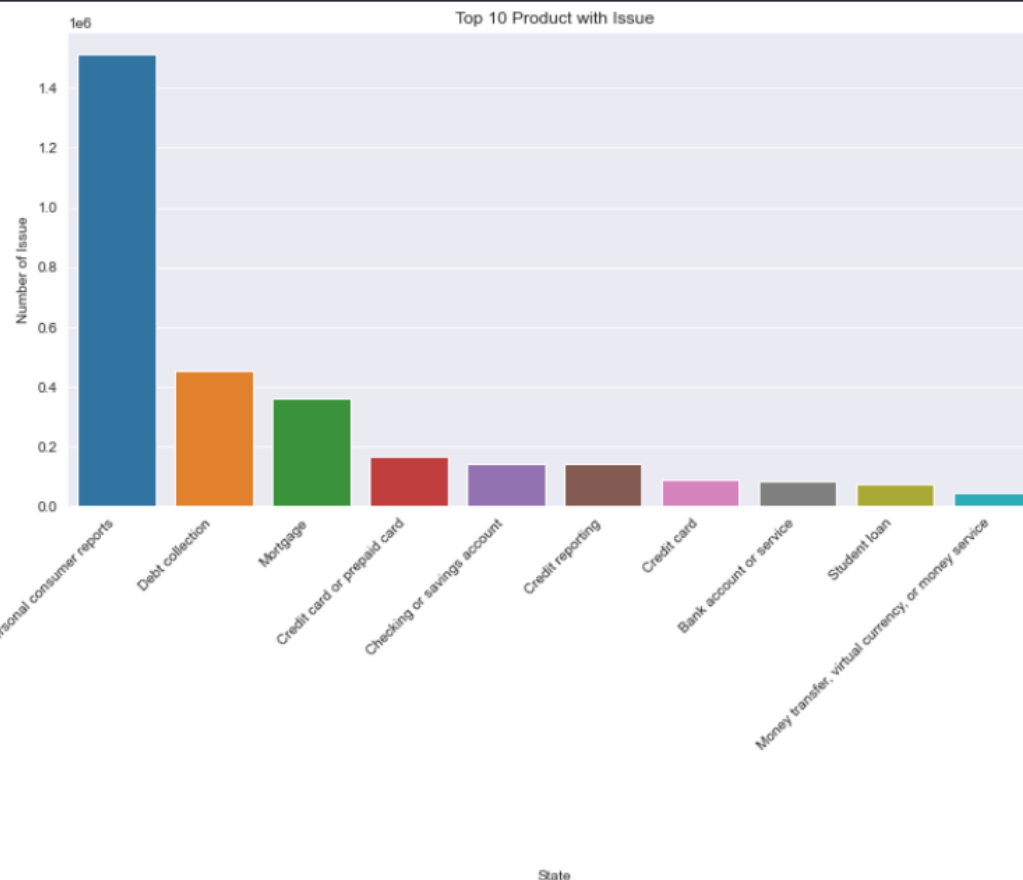


# Issue distribution by Season

## Autumn with Number 827702



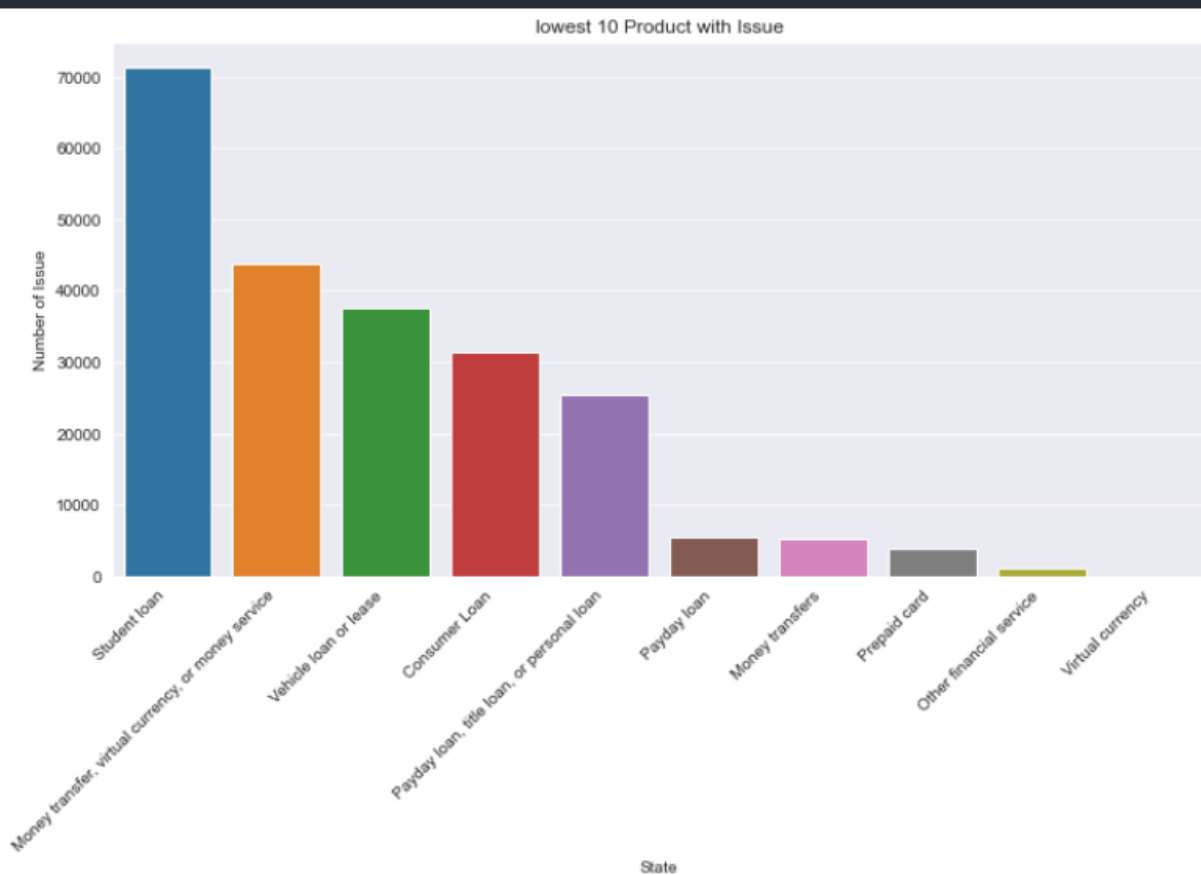
# Top 10 Product with Issue



Product	Issue
Credit reporting, credit repair services, or o...	1511223
Debt collection	455088
Mortgage	361159
Credit card or prepaid card	166472
Checking or savings account	140882
Credit reporting	139956
Credit card	87230
Bank account or service	84649
Student loan	71246
Money transfer, virtual currency, or money ser...	43655



# lowest 10 Product with Issue



Product	Issue
Student loan	71246
Money transfer, virtual currency, or money ser...	43655
Vehicle loan or lease	37523
Consumer Loan	31394
Payday loan, title loan, or personal loan	25284
Payday loan	5521
Money transfers	5155
Prepaid card	3776
Other financial service	1038
Virtual currency	17

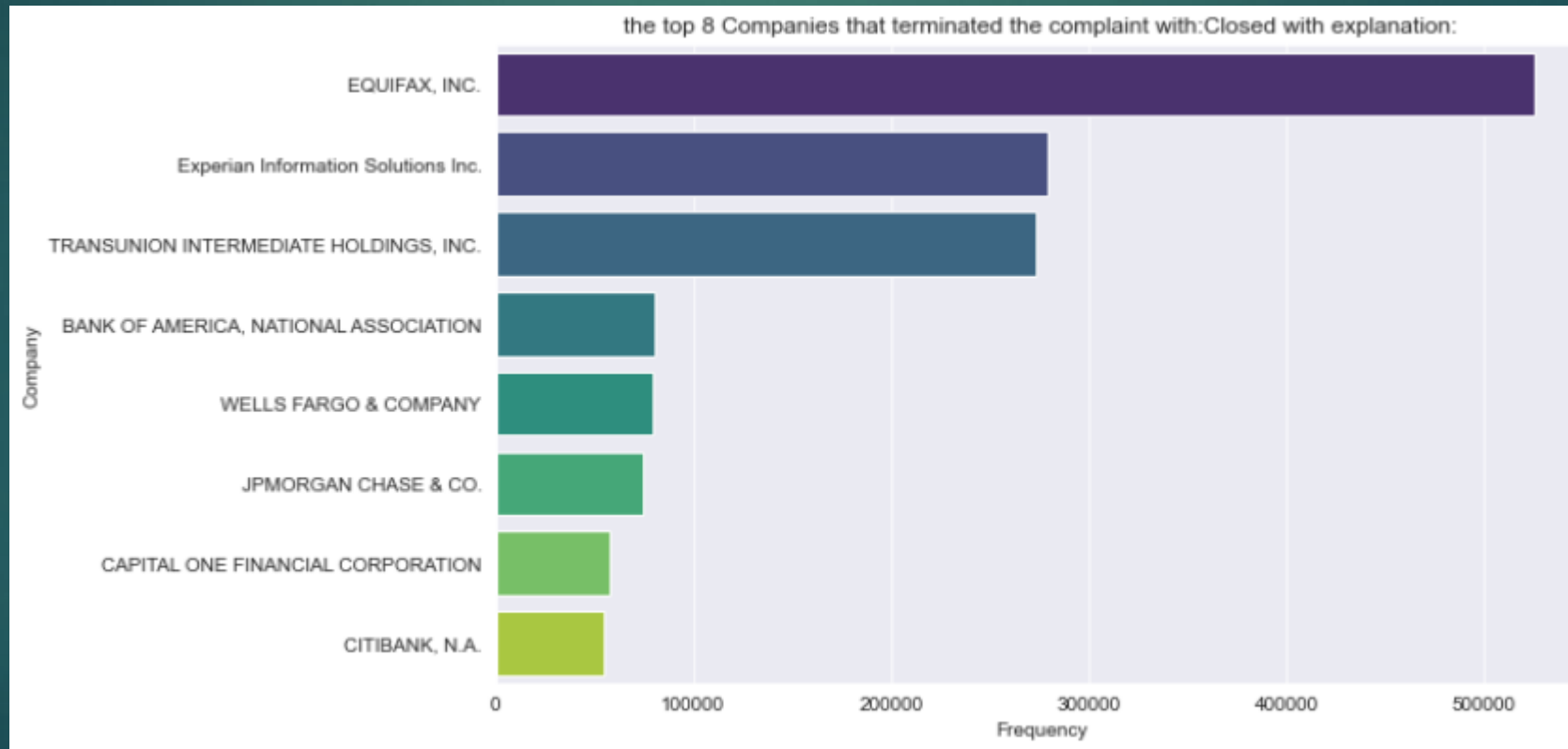
# Company response to consumer

Here we are talking about the compensation that companies provide to consumers ►

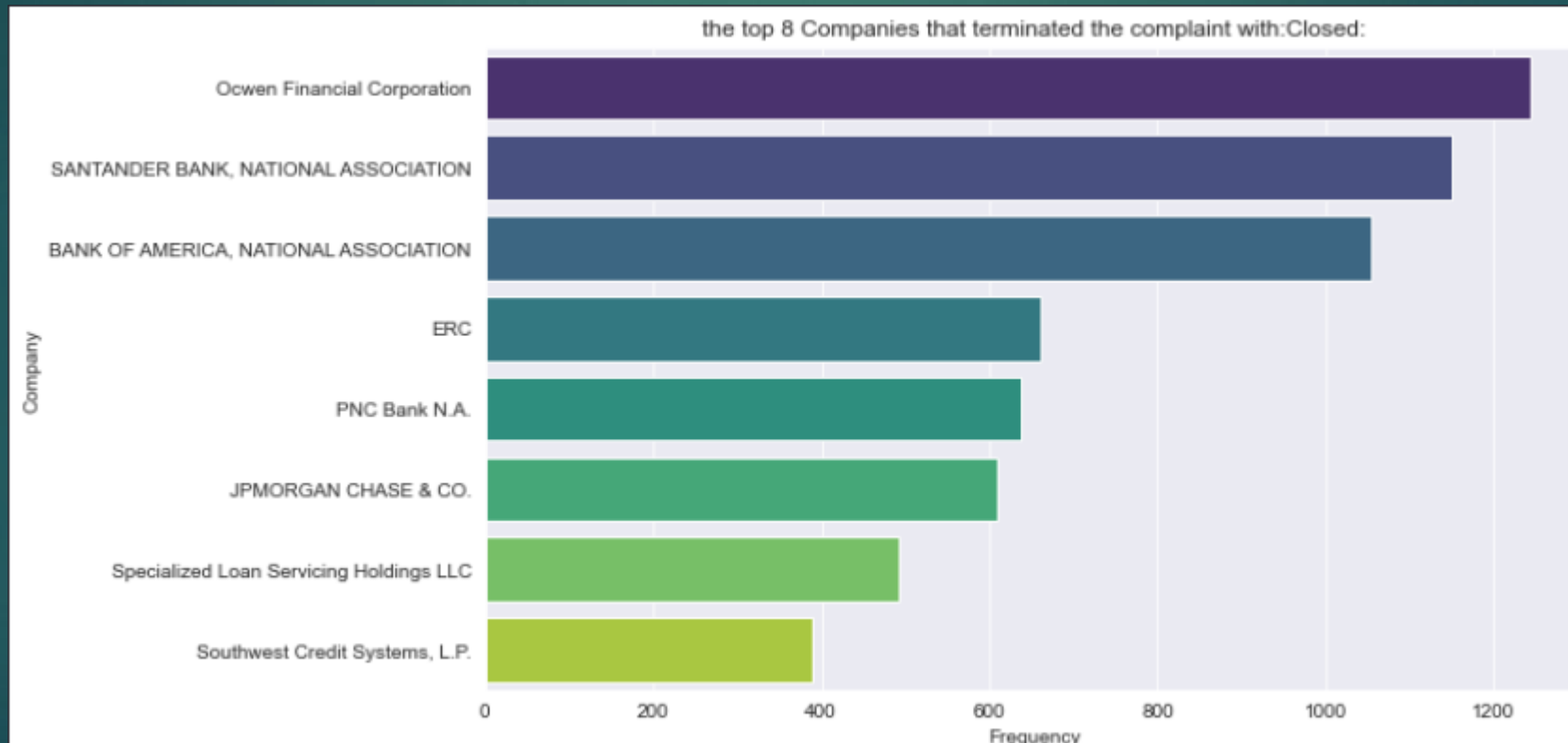
Containing: ►

Company response to consumer	Company
Closed	17425
Closed with explanation	2480303
Closed with monetary relief	123608
Closed with non-monetary relief	520813
Closed with relief	4612
Closed without relief	15964
In progress	2
Untimely response	8541

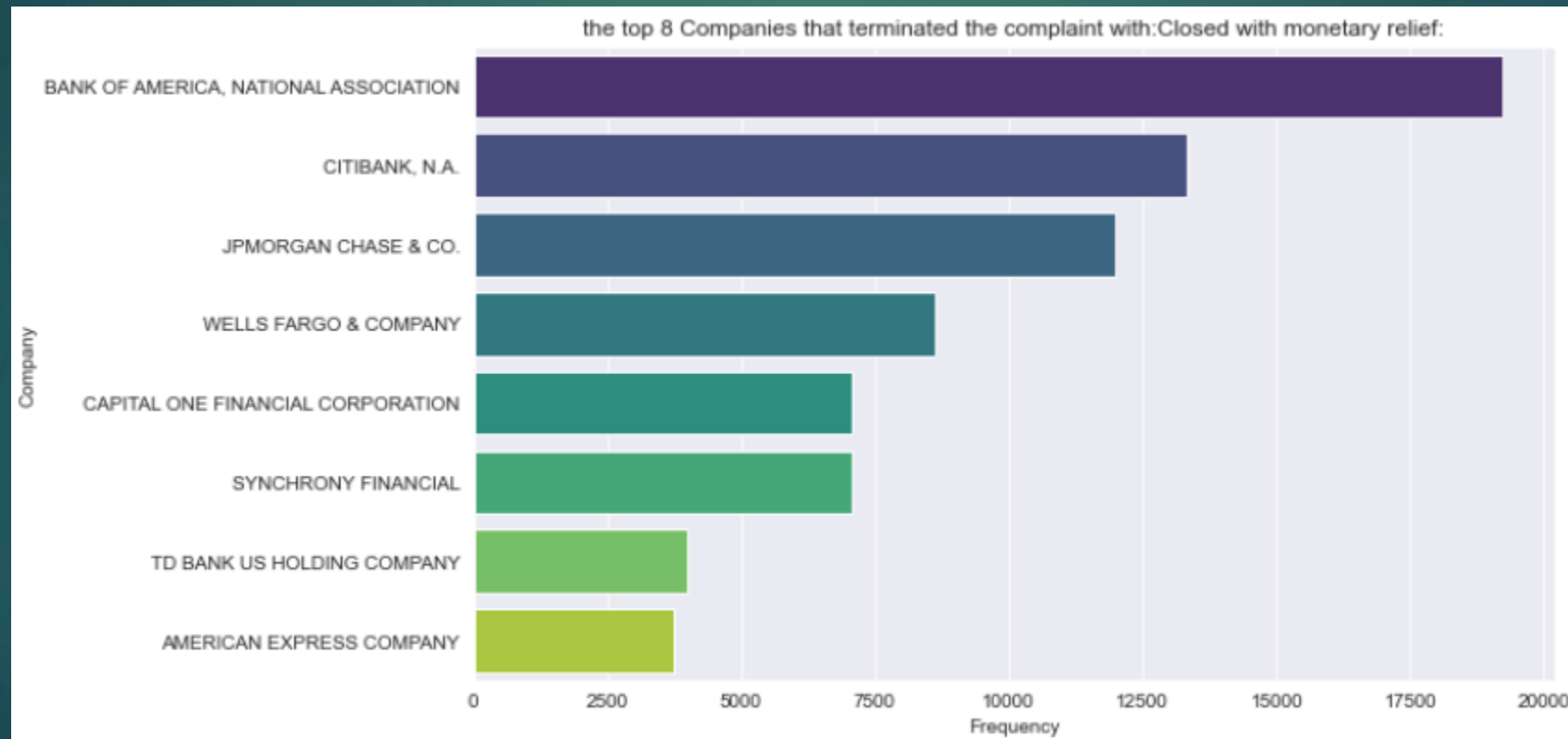
Equifax, Inc. is the top company that closed complaints with explanation, with a count of 525,216. The second top company is Experian Information Solutions Inc. with a count of 279,316, followed by TransUnion Intermediate Holdings, Inc. with a count of 273,640. Bank of America, National Association and Wells Fargo & Company come in fourth and fifth place respectively, with counts of 80,319 and 79,279.



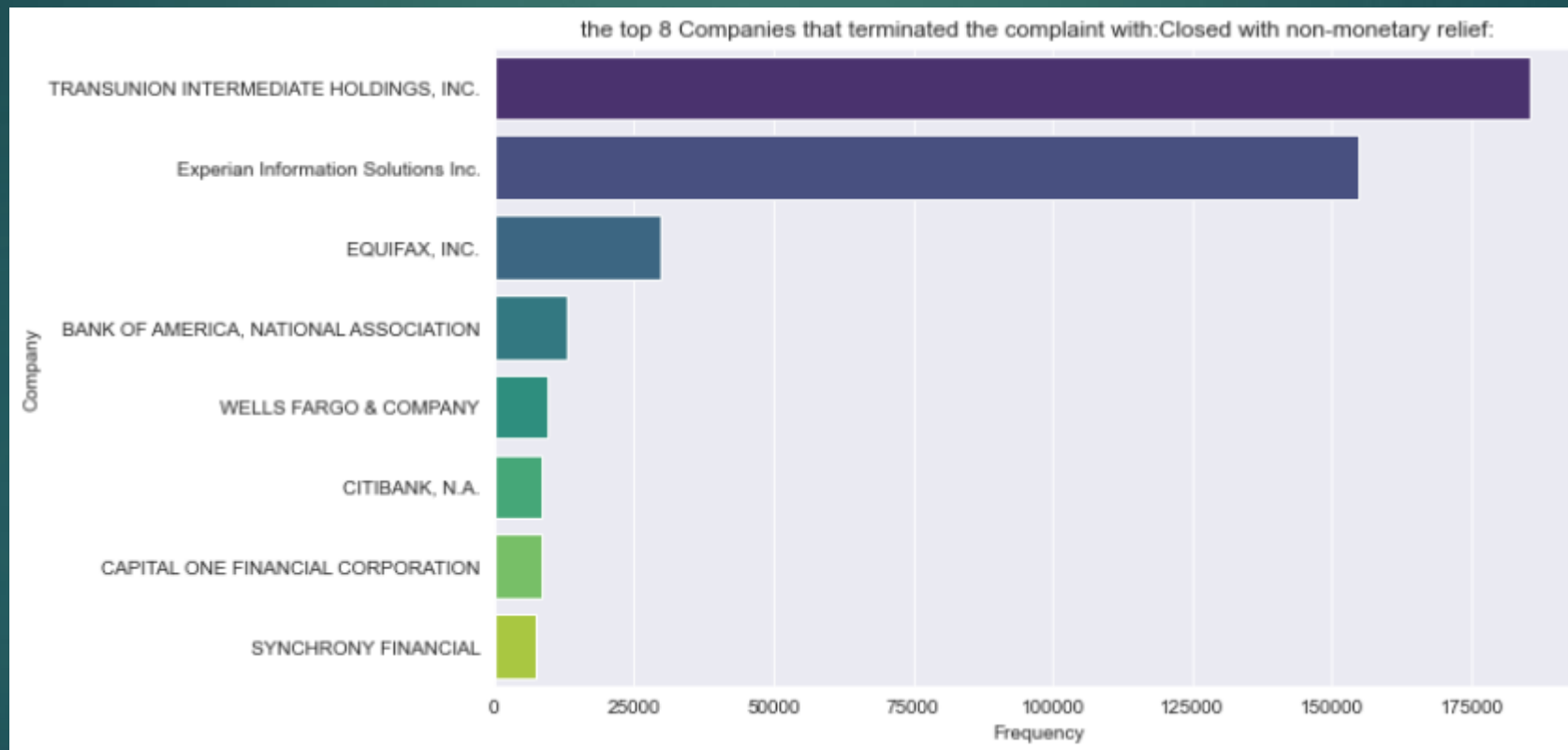
Ocwen Financial Corporation is the top company with closed complaints, with a count of 1,244. The second top company is Santander Bank, National Association, with a count of 1,150, followed by Bank of America, National Association with a count of 1,055. ERC and PNC Bank N.A. come in fourth and fifth place respectively, with counts of 660 and 637.



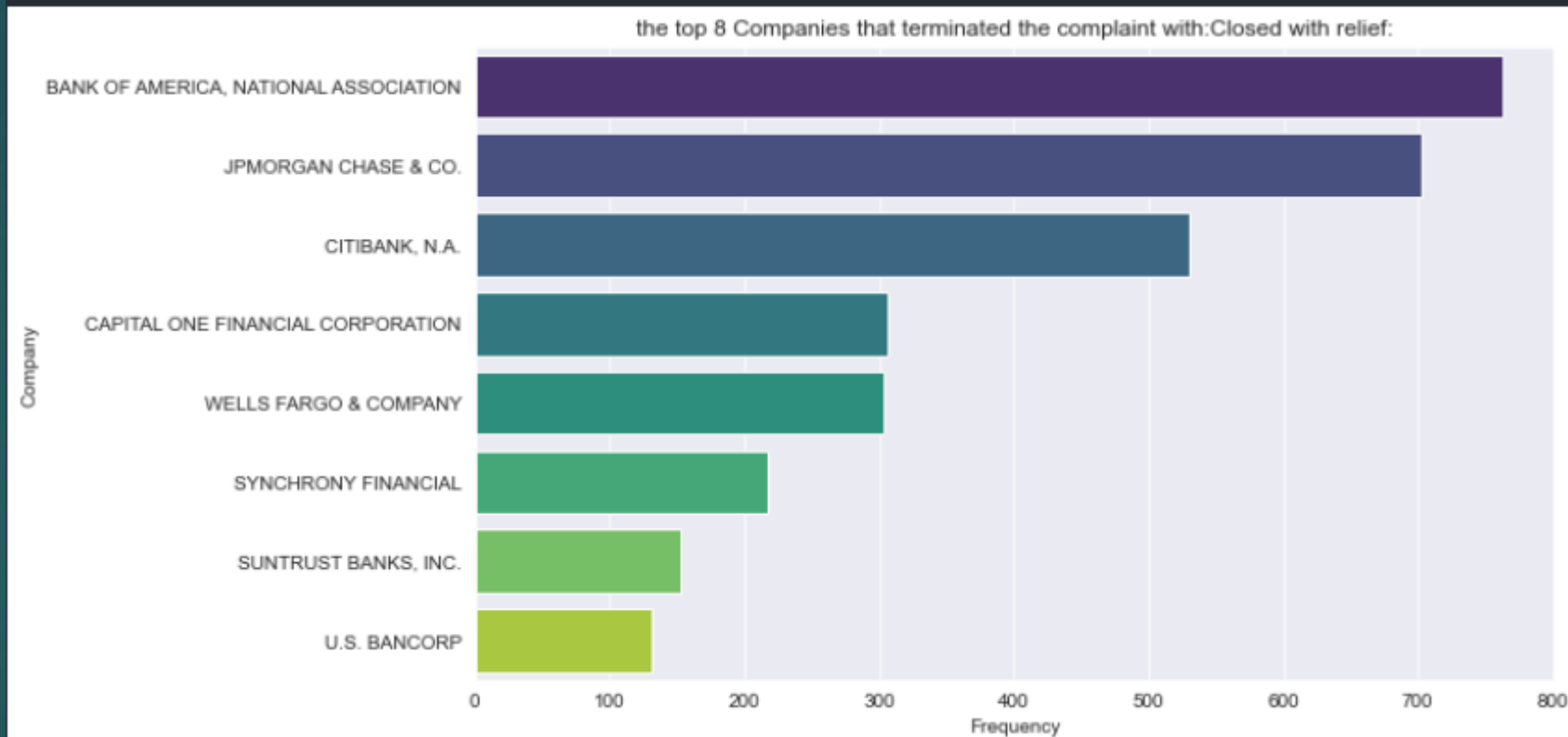
Bank of America, National Association is the top company that closed complaints with monetary relief, with a count of 19,231. The second top company is Citibank, N.A. with a count of 13,312, followed by JPMorgan Chase & Co. with a count of 12,004. Wells Fargo & Company and Capital One Financial Corporation come in fourth and fifth place respectively, with counts of 8,616 and 7,071.



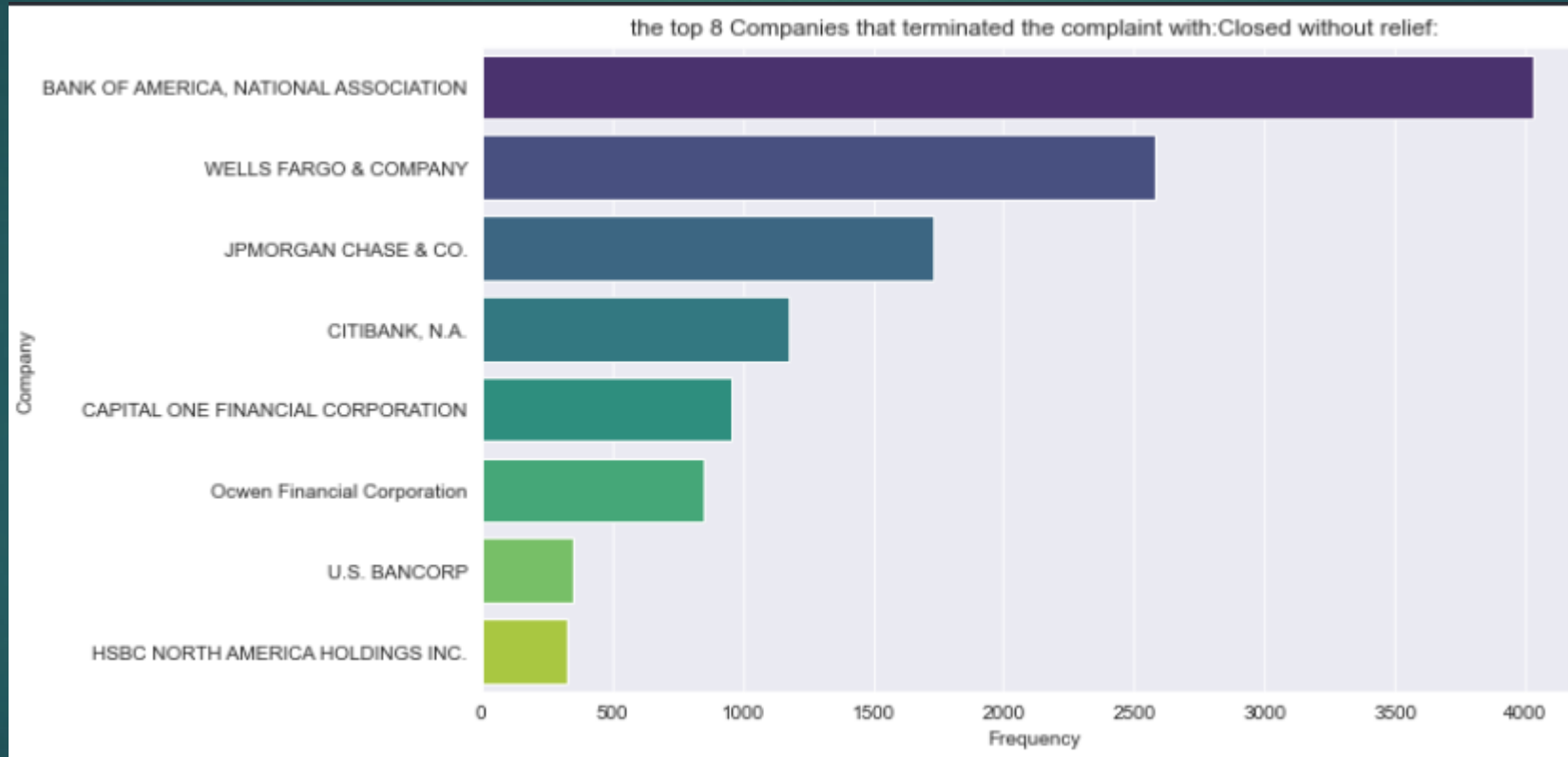
the top company that closed complaints with non-monetary relief is TransUnion Intermediate Holdings, Inc. with a count of 185,442. The second top company is Experian Information Solutions Inc. with a count of 154,672. Equifax, Inc. comes in third place with a count of 29,698. The rest of the companies have much smaller counts.



Bank of America, National Association is the top company that closed complaints with relief, with a count of 763. The second top company is JPMorgan Chase & Co. with a count of 703, followed by Citibank, N.A. with a count of 530. Capital One Financial Corporation and Wells Fargo & Company come in fourth and fifth place respectively, with counts of 306 and 304.



Bank of America, National Association is the top company that closed complaints without relief, with a count of 4,030. The second top company is Wells Fargo & Company with a count of 2,582, followed by JPMorgan Chase & Co. with a count of 1,733. Citibank, N.A. and Capital One Financial Corporation come in fourth and fifth place respectively, with counts of 1,176 and 954.





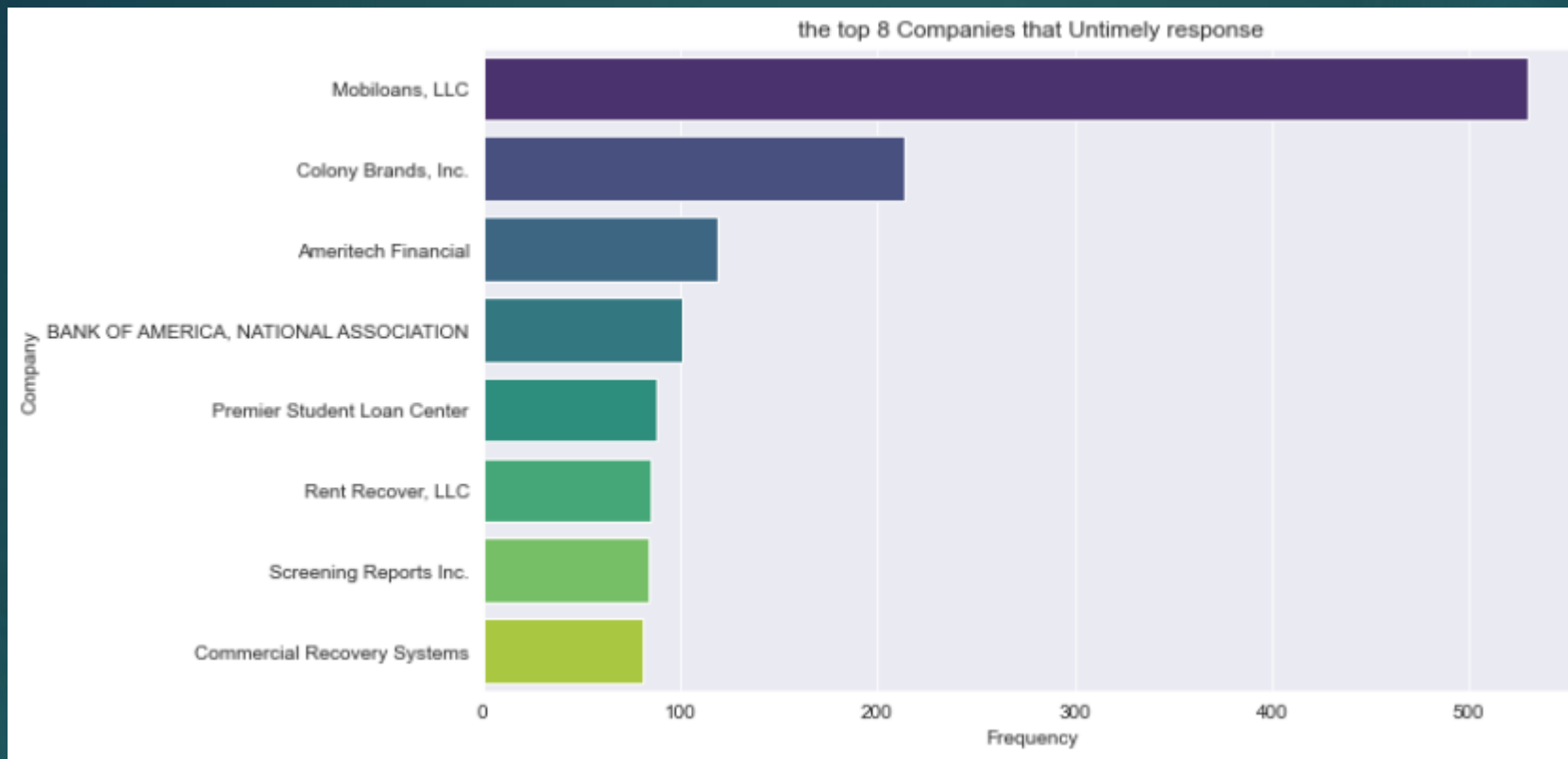
Mobiloans, LLC This company received a total of 530 complaints from consumers who reported an untimely response. ▶

With 214 complaints, Colony Brands, Inc. ranks second in the list of companies with the most untimely response complaints. ▶

Ameritech Financial: This company received 119 complaints from consumers dissatisfied with their response times. ▶

BANK OF AMERICA, NATIONAL ASSOCIATION: Ranked fourth, this company received 101 complaints regarding untimely responses to consumer complaints. ▶

Premier Student Loan Center: With 88 complaints, Premier Student Loan Center rounds out the top five companies with the highest number of untimely response complaints. ▶



there are only two companies that have complaints that are still in progress: CNY Management Group Inc and Home Base Mortgage Group. Each company has one complaint that is still in progress. ►

In the end, we have mentioned in this report a lot of important things and information about this data

We hope this report is valuable and useful to everyone, and we stand ready to provide more details or support in the future.

Thank You!