

Comcast Telecom Consumer Complaint Project

November 6, 2022

1 Name - Samiksha Borade

1.1 Project Title - Comcast Telecom Consumer Complaint

```
[1]: # import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
[2]: import os
```

```
[3]: os.getcwd()
```

```
[3]: 'C:\\Users\\HP'
```

```
[4]: df = pd.read_csv(r"C:
↪\Users\HP\Desktop\comcast_tele_consumer\Comcast_telecom_complaints.csv")
```

```
[5]: df.head()    #getting first rows of dataset
```

```
[5]: Ticket #           Customer Complaint      Date \
0    250635           Comcast Cable Internet Speeds  22-04-15
1    223441      Payment disappear - service got disconnected  04-08-15
2    242732           Speed and Service  18-04-15
3    277946  Comcast Imposed a New Usage Cap of 300GB that ...  05-07-15
4    307175      Comcast not working and no service to boot  26-05-15
```

```
      Date_month_year      Time      Received Via      City      State \
0      22-Apr-15    3:53:50 PM  Customer Care Call  Abingdon  Maryland
1      04-Aug-15    10:22:56 AM      Internet  Acworth  Georgia
2      18-Apr-15    9:55:47 AM      Internet  Acworth  Georgia
3      05-Jul-15    11:59:35 AM      Internet  Acworth  Georgia
4      26-May-15    1:25:26 PM      Internet  Acworth  Georgia
```

```
      Zip code  Status  Filing on Behalf of Someone
0      21009  Closed                No
1      30102  Closed                No
```

2	30101	Closed	Yes
3	30101	Open	Yes
4	30101	Solved	No

```
[6]: df['Customer Complaint'] = df['Customer Complaint'].str.lower()
df.head(2)
```

```
[6]: Ticket #                Customer Complaint      Date \
0    250635                comcast cable internet speeds  22-04-15
1    223441  payment disappear - service got disconnected  04-08-15

Date_month_year      Time      Received Via      City      State \
0      22-Apr-15    3:53:50 PM  Customer Care Call  Abingdon  Maryland
1      04-Aug-15   10:22:56 AM           Internet    Acworth   Georgia

Zip code  Status Filing on Behalf of Someone
0      21009  Closed                        No
1      30102  Closed                        No
```

```
[7]: df['list_cc'] = df['Customer Complaint'].apply(lambda x: x.split(' '))
df.head()
```

```
[7]: Ticket #                Customer Complaint      Date \
0    250635                comcast cable internet speeds  22-04-15
1    223441  payment disappear - service got disconnected  04-08-15
2    242732                speed and service            18-04-15
3    277946  comcast imposed a new usage cap of 300gb that ... 05-07-15
4    307175  comcast not working and no service to boot    26-05-15

Date_month_year      Time      Received Via      City      State \
0      22-Apr-15    3:53:50 PM  Customer Care Call  Abingdon  Maryland
1      04-Aug-15   10:22:56 AM           Internet    Acworth   Georgia
2      18-Apr-15    9:55:47 AM           Internet    Acworth   Georgia
3      05-Jul-15   11:59:35 AM           Internet    Acworth   Georgia
4      26-May-15    1:25:26 PM           Internet    Acworth   Georgia

Zip code  Status Filing on Behalf of Someone \
0      21009  Closed                        No
1      30102  Closed                        No
2      30101  Closed                        Yes
3      30101  Open                          Yes
4      30101  Solved                        No

list_cc
0      [comcast, cable, internet, speeds]
1  [payment, disappear, -, service, got, disconne...
2      [speed, and, service]
```

```
3 [comcast, imposed, a, new, usage, cap, of, 300...
4 [comcast, not, working, and, no, service, to, ...
```

```
[8]: #Convert Date_Full and Date_month_year to Datetime Format
```

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

df['Date_month_year'] = pd.to_datetime(df['Date_month_year'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2224 entries, 0 to 2223
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Ticket #                             2224 non-null   object
1   Customer Complaint                   2224 non-null   object
2   Date                                 2224 non-null   datetime64[ns]
3   Date_month_year                     2224 non-null   datetime64[ns]
4   Time                                 2224 non-null   object
5   Received Via                        2224 non-null   object
6   City                                 2224 non-null   object
7   State                               2224 non-null   object
8   Zip code                            2224 non-null   int64
9   Status                              2224 non-null   object
10  Filing on Behalf of Someone          2224 non-null   object
11  list_cc                             2224 non-null   object
dtypes: datetime64[ns](2), int64(1), object(9)
memory usage: 208.6+ KB
```

```
[9]: df['Date'] = df['Date_month_year'].dt.day
df.head()
```

```
[9]: Ticket # Customer Complaint Date \
0 250635 comcast cable internet speeds 22
1 223441 payment disappear - service got disconnected 4
2 242732 speed and service 18
3 277946 comcast imposed a new usage cap of 300gb that ... 5
4 307175 comcast not working and no service to boot 26

Date_month_year Time Received Via City State \
0 2015-04-22 3:53:50 PM Customer Care Call Abingdon Maryland
1 2015-08-04 10:22:56 AM Internet Acworth Georgia
2 2015-04-18 9:55:47 AM Internet Acworth Georgia
3 2015-07-05 11:59:35 AM Internet Acworth Georgia
4 2015-05-26 1:25:26 PM Internet Acworth Georgia

Zip code Status Filing on Behalf of Someone \
```

0	21009	Closed	No
1	30102	Closed	No
2	30101	Closed	Yes
3	30101	Open	Yes
4	30101	Solved	No

	list_cc
0	[comcast, cable, internet, speeds]
1	[payment, disappear, -, service, got, disconne...
2	[speed, and, service]
3	[comcast, imposed, a, new, usage, cap, of, 300...
4	[comcast, not, working, and, no, service, to, ...

```
[10]: df['month_name'] =pd.to_datetime(df['Date_month_year']).dt.month_name()
df.head()
```

```
[10]: Ticket # Customer Complaint Date \
0 250635 comcast cable internet speeds 22
1 223441 payment disappear - service got disconnected 4
2 242732 speed and service 18
3 277946 comcast imposed a new usage cap of 300gb that ... 5
4 307175 comcast not working and no service to boot 26
```

	Date_month_year	Time	Received Via	City	State	\
0	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland	
1	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia	
2	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia	
3	2015-07-05	11:59:35 AM	Internet	Acworth	Georgia	
4	2015-05-26	1:25:26 PM	Internet	Acworth	Georgia	

	Zip code	Status	Filing on Behalf of Someone	\
0	21009	Closed	No	
1	30102	Closed	No	
2	30101	Closed	Yes	
3	30101	Open	Yes	
4	30101	Solved	No	

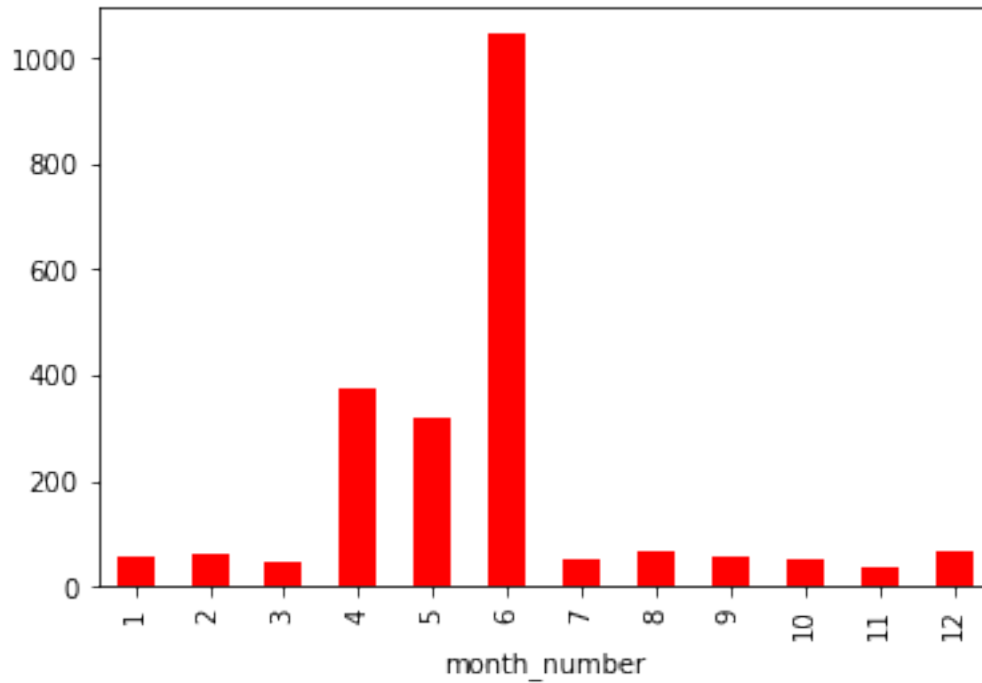
	list_cc	month_name
0	[comcast, cable, internet, speeds]	April
1	[payment, disappear, -, service, got, disconne...	August
2	[speed, and, service]	April
3	[comcast, imposed, a, new, usage, cap, of, 300...	July
4	[comcast, not, working, and, no, service, to, ...	May

```
[11]: # graph using month number

df['month_number']=pd.to_datetime(df['Date_month_year']).dt.month
```

```
df.groupby(['month_number'])['Customer Complaint'].count().plot(kind='bar',color= 'red')

plt.show()
```

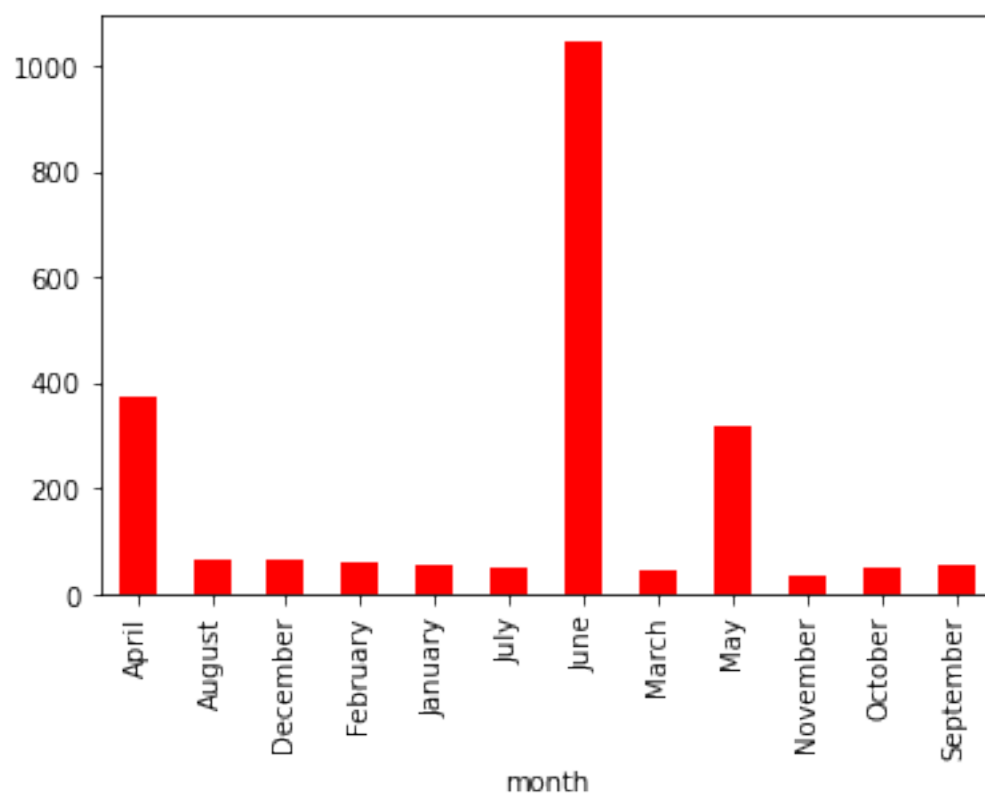
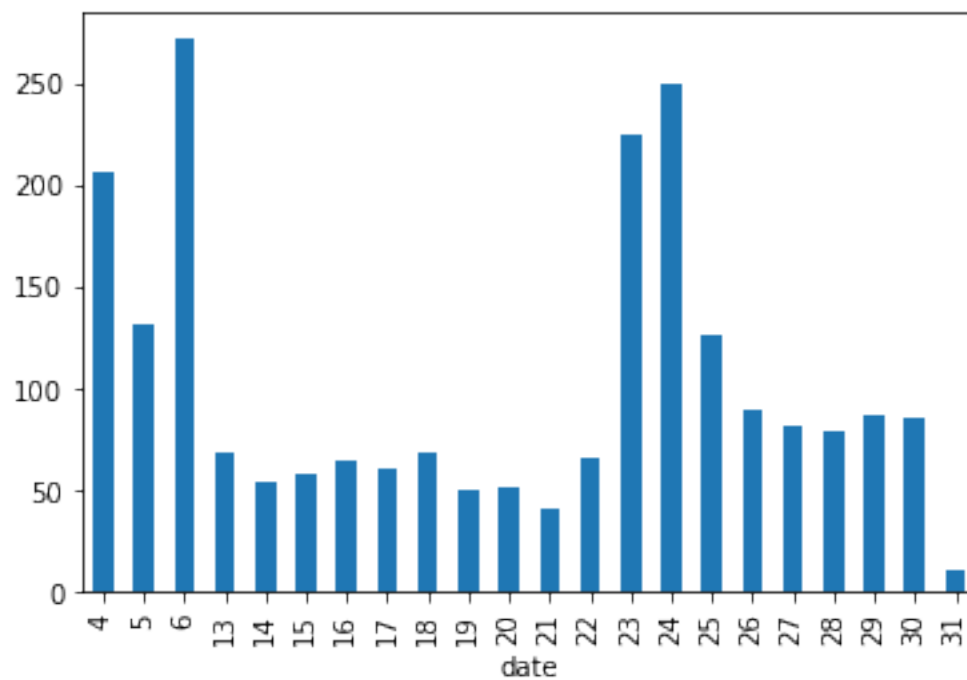


```
[12]: # provide the trend chart for the number of complaint and daily granularity
      ↪ levels.

df['month']=pd.to_datetime(df['Date_month_year']).dt.month_name()
df['date']=pd.to_datetime(df['Date_month_year']).dt.day

df.groupby(['date'])['Customer Complaint'].count().plot(kind='bar')
plt.show()

df.groupby(['month'])['Customer Complaint'].count().plot(kind='bar',color='red')
plt.show()
```



```
[13]: b1 = df.groupby(['month_name', 'date'])[['Customer Complaint']].count().
      ↪reset_index()
      b1
```

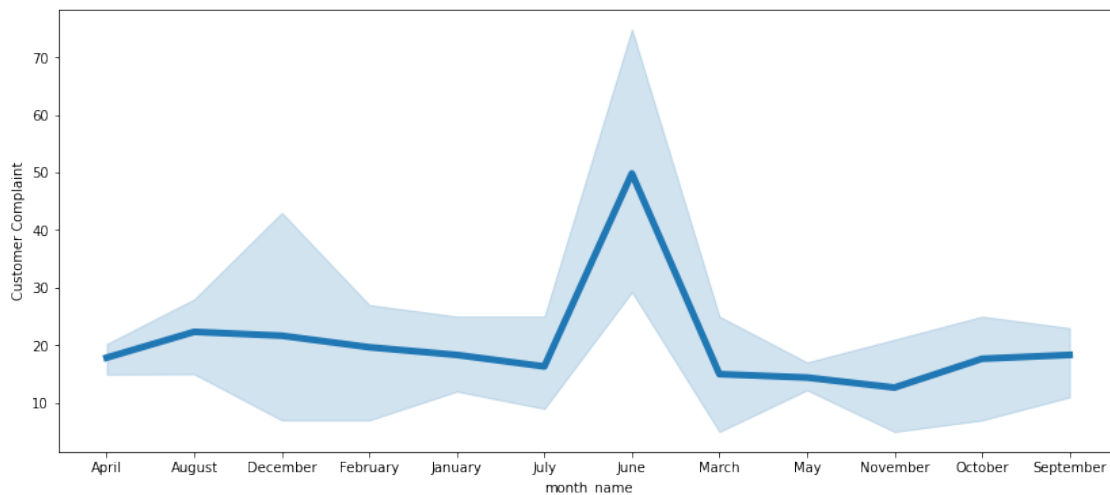
```
[13]:   month_name  date  Customer Complaint
0      April     4             12
1      April     5             12
2      April     6             12
3      April    13             24
4      April    14             23
..      ...    ...
86    October     5              7
87    October     6             25
88  September     4             21
89  September     5             11
90  September     6             23
```

[91 rows x 3 columns]

```
[14]: import seaborn as sns
```

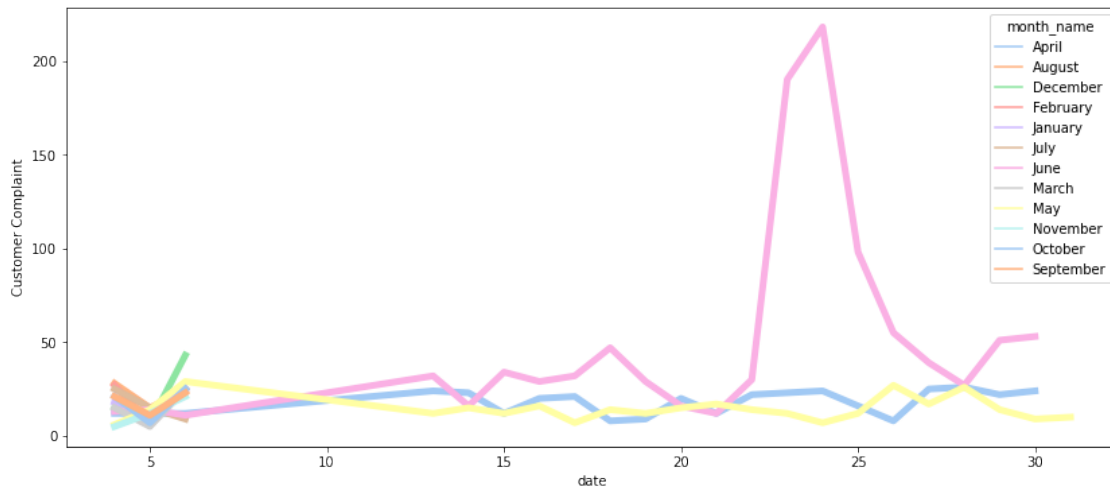
```
[15]: # Provide the trend chart for the number of complaints at monthly granularity
      ↪levels.
      #Increase Graph Size

      plt.figure(figsize=(14,6))           #width, height
      sns.lineplot(x = "month_name",y = "Customer Complaint",data = b1, lw=5)
      plt.show()
```



```
[16]: # Provide the trend chart for the number of complaints at daily granularity
      ↪ levels.
```

```
plt.figure(figsize=(14,6))
sns.lineplot(x = "date", y = 'Customer Complaint',data = b1, palette = '
      ↪pastel', lw=5, hue = 'month_name' )
plt.show()
```



```
[17]: # To get the frequency of complaint types first we have to see all complaint
      ↪types and check for duplicate, case sensitive
      # Incomplete data so that we can make analytics better
      df['Customer Complaint'].value_counts().to_frame().reset_index()
```

```
[17]:
```

	index	Customer Complaint
0	comcast	102
1	comcast data cap	30
2	comcast internet	29
3	comcast data caps	21
4	comcast billing	18
...
1735	monthly data caps	1
1736	comcast/xfinity poor service, fraudulent billi...	1
1737	lost emails/billing	1
1738	improper billing and non resolution of issues	1
1739	comcast, ypsilanti mi internet speed	1

[1740 rows x 2 columns]

```
[22]: df_type.head(10)
```



```
[22]: COMCAST          102
      COMCAST DATA CAP    30
      COMCAST INTERNET    29
      COMCAST DATA CAPS   21
      COMCAST BILLING     18
      COMCAST SERVICE     15
      INTERNET SPEED      15
      UNFAIR BILLING PRACTICES 13
      DATA CAPS          13
      DATA CAP           12
      Name: Customer Complaint, dtype: int64
```

```
[19]: # Better to convert all data into upper case or sentence case so duplicate value
      ↪will shorted
      df_type=df["Customer Complaint"].str.upper().value_counts()
```

```
[20]: # showed only top 25 Complaint Types.I can show clearly in this table that
      ↪COMCAST, COMCAST DATA CAP
      # COMCAST INTERNET are the Highest top 3 complaint types
      df_type.head(25)
```

```
[20]: COMCAST          102
      COMCAST DATA CAP    30
      COMCAST INTERNET    29
      COMCAST DATA CAPS   21
      COMCAST BILLING     18
      COMCAST SERVICE     15
      INTERNET SPEED      15
      UNFAIR BILLING PRACTICES 13
      DATA CAPS          13
      DATA CAP           12
      COMCAST COMPLAINT    11
      COMCAST/XFINITY      11
      COMCAST INTERNET SERVICE 10
      BILLING               9
      BILLING ISSUES        8
      COMCAST CABLE         5
      INTERNET              5
      COMCAST BILLING COMPLAINT 5
      COMCAST ISSUES        5
      COMCAST BILLING PRACTICES 5
      SERVICE ISSUES        5
      SLOW INTERNET         5
      INTERNET SERVICE      5
      COMPLAINT AGAINST COMCAST 5
      COMCAST UNFAIR BILLING PRACTICES 4
      Name: Customer Complaint, dtype: int64
```

```
[23]: # Check how many unique values are under Status Column
```

```
df['Status'].unique()
```

```
[23]: array(['Closed', 'Open', 'Solved', 'Pending'], dtype=object)
```

```
[24]: # which complaint types are maximum i.e., around internet, network issues, or  
      ↪ across any other domains.
```

```
df['Customer Complaint'].value_counts().head(5)
```

```
[24]: comcast                102  
      comcast data cap      30  
      comcast internet      29  
      comcast data caps     21  
      comcast billing       18  
      Name: Customer Complaint, dtype: int64
```

```
[25]: #Create a new categorical variable with value as Open and Closed. Open &  
      ↪ Pending is to be categorized as Open and Closed & Solved is to be  
      ↪ categorized as Closed.
```

```
df['Status']=df['Status'].apply(lambda x: 'Open' if ((x=='Open') |  
      ↪ (x=='Pending')) else 'closed')
```

```
[26]: df_status_by_state = pd.crosstab(df.State,df.Status)
```

```
[27]: df_status_by_state
```

```
[27]: Status          Open  closed  
State  
Alabama             9      17  
Arizona             6      14  
Arkansas            0       6  
California          61     159  
Colorado            22      58  
Connecticut         3       9  
Delaware            4       8  
District Of Columbia 2      14  
District of Columbia 0       1  
Florida             39     201  
Georgia             80     208  
Illinois            29     135  
Indiana             9      50  
Iowa                0       1  
Kansas              1       1  
Kentucky            3       4
```

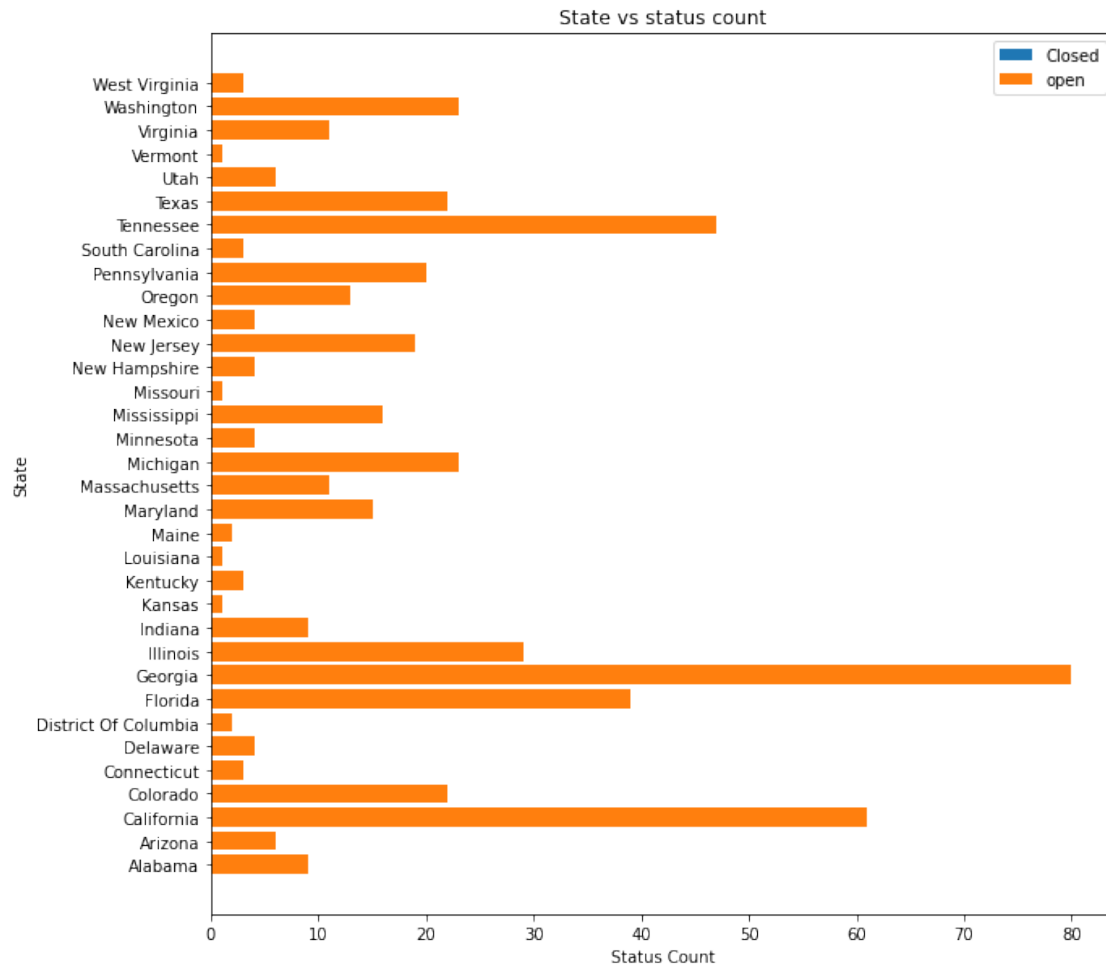
Louisiana	1	12
Maine	2	3
Maryland	15	63
Massachusetts	11	50
Michigan	23	92
Minnesota	4	29
Mississippi	16	23
Missouri	1	3
Montana	0	1
Nevada	0	1
New Hampshire	4	8
New Jersey	19	56
New Mexico	4	11
New York	0	6
North Carolina	0	3
Ohio	0	3
Oregon	13	36
Pennsylvania	20	110
Rhode Island	0	1
South Carolina	3	15
Tennessee	47	96
Texas	22	49
Utah	6	16
Vermont	1	2
Virginia	11	49
Washington	23	75
West Virginia	3	8

```
[28]: # provide status wise status of complaints in a stacked bar chart.use the_
      ↪categorized variable from Q3
```

```
[29]: op=df[df['Status']=='Open'].groupby(['State'])['Status'].count().to_frame().
      ↪reset_index()
      cl=df[df['Status']=='Closed'].groupby(['State'])['Status'].count().to_frame().
      ↪reset_index()

      fig=plt.figure(figsize=(10,10))
      plt.barh(cl.State, cl.Status)
      plt.barh(op.State, op.Status)

      plt.ylabel("State", size=10)
      plt.xlabel("Status Count")
      plt.legend(["Closed","open"])
      plt.title("State vs status count")
      plt.show()
```



[30]: # Which state has the maximum complaints - Georgia has maximum number of complaints

```
df_unresolved_complaints = df[df['Status']=='Open']
```

[31]: df_unresolved_complaint_count = df_unresolved_complaints.State.value_counts()
df_unresolved_complaint_count

```
[31]: Georgia      80
      California   61
      Tennessee   47
      Florida     39
      Illinois    29
      Michigan    23
      Washington  23
      Texas       22
      Colorado    22
```

Pennsylvania	20
New Jersey	19
Mississippi	16
Maryland	15
Oregon	13
Massachusetts	11
Virginia	11
Indiana	9
Alabama	9
Arizona	6
Utah	6
New Hampshire	4
Delaware	4
Minnesota	4
New Mexico	4
Connecticut	3
West Virginia	3
South Carolina	3
Kentucky	3
Maine	2
District Of Columbia	2
Missouri	1
Vermont	1
Kansas	1
Louisiana	1

Name: State, dtype: int64

```
[32]: # finding max number of complaints using max function
```

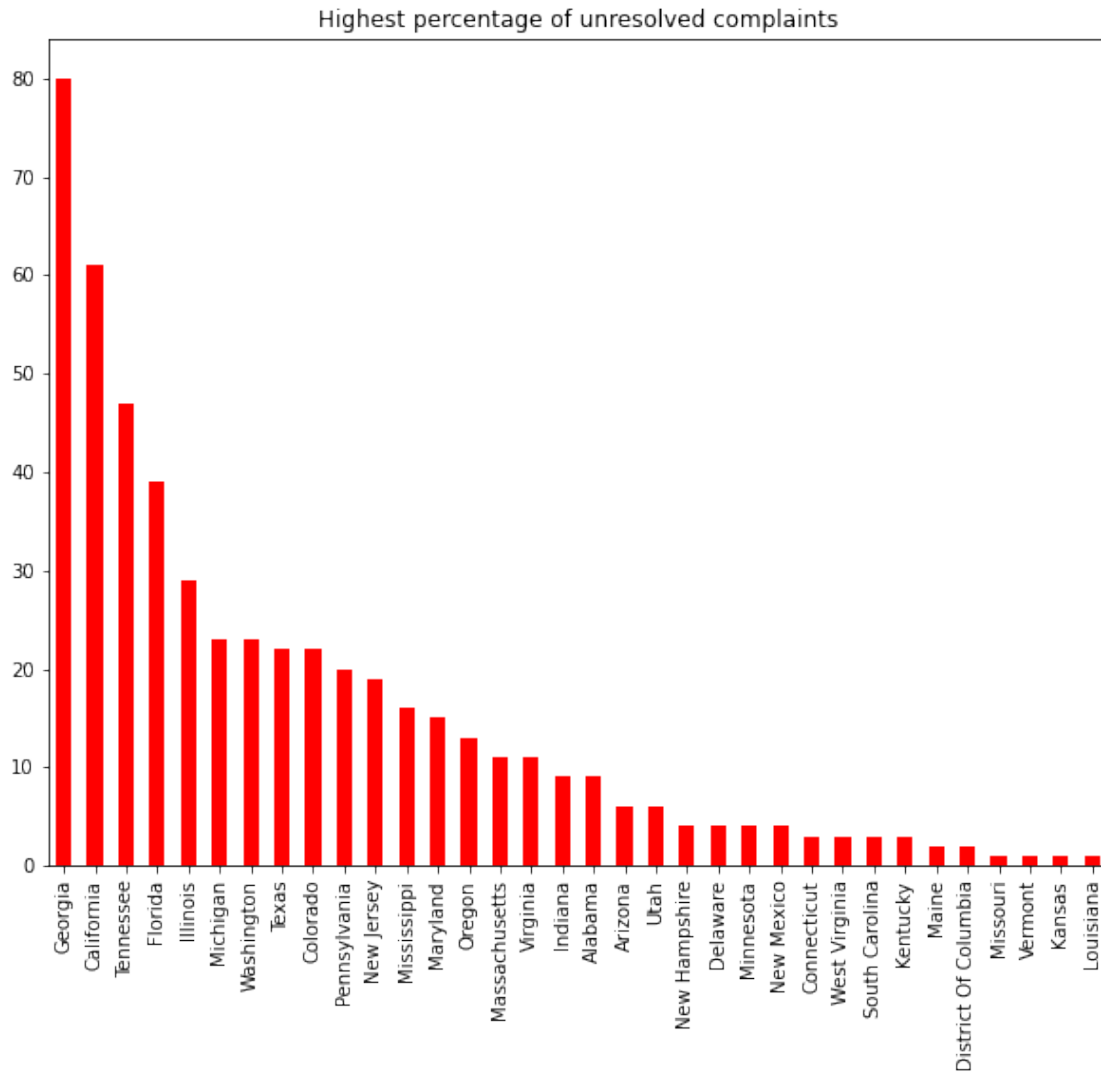
```
np.where(op.Status==op.Status.max())
op.iloc[8]
```

```
[32]: State      Georgia
      Status      80
      Name: 8, dtype: object
```

```
[33]: # Georgia has the Highest Number of unresolved complaints
      # Show this by Bar Graph
```

```
df_unresolved_complaint_count.plot(kind='bar',figsize=(10,8),color="red")
plt.title('Highest percentage of unresolved complaints')
```

```
[33]: Text(0.5, 1.0, 'Highest percentage of unresolved complaints')
```



```
[34]: # Provide the percentage of complaints resolved till date, which were received
      ↪ through the Internet and customer care calls.
```

```
df['Received Via'].unique()
```

```
[34]: array(['Customer Care Call', 'Internet'], dtype=object)
```

```
[38]: df[df['Status']=='Closed'].groupby('Status')['Received Via'].
      ↪ value_counts(normalize=True)*100
```

```
[38]: Series([], Name: Received Via, dtype: float64)
```

```
[36]: df.Status.value_counts()*100
```

```
[36]: closed    170700
      Open      51700
      Name: Status, dtype: int64
```

```
[37]: result=df[df['Received Via'].isin(['Internet','Customer Care Call'])['Status'].
      ↪value_counts(normalize = True).to_frame('Percentage of complaints (Closed vs_
      ↪Open)')
      result*100
```

```
[37]:      Percentage of complaints (Closed vs Open)
      closed                                76.753597
      Open                                  23.246403
```

```
[38]: # Used autopct='%1.1f%%' to show percentage under the pie chart
      myexplode = [0.2, 0]
      plt.title('Complaints Status through the Internet & Customer Care Calls\n')
      df.Status.value_counts().plot(kind='pie',explode = myexplode,autopct='%1.1f%%',
      figsize = (14,6))
```

```
[38]: <AxesSubplot:title={'center':'Complaints Status through the Internet & Customer
      Care Calls\n'}, ylabel='Status'>
```

Complaints Status through the Internet & Customer Care Calls

