Project_4

January 11, 2023

1 PROJECT: Comcast telecomm Complaints

1.0.1 Description

Comcast is an American global telecommunication company. The firm has been providing terrible customer service. They continue to fall short despite repeated promises to improve. Only last month (October 2016) the authority fined them a $2.3 million, after receiving over 1000 consumer complaints.

1.0.2 Data Dictionary

Ticket #: Ticket number assigned to each complaint
Customer Complaint: Description of complaint
Date: Date of complaint
Time: Time of complaint
Received Via: Mode of communication of the complaint
City: Customer city
State: Customer state
Zipcode: Customer zip
Status: Status of complaint
Filing on behalf of someone

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

Task 1: Import data into Python environment

[2]:
```
df= pd.read_csv('Comcast_telecom_complaints_data.csv')
```

[3]:
```
#Identifying shape of the dataset
df.shape
```

```
[3]: (2224, 11)
```

```
[4]: columns= df.columns
     columns
```

```
[4]: Index(['Ticket #', 'Customer Complaint', 'Date', 'Date_month_year', 'Time',
            'Received Via', 'City', 'State', 'Zip code', 'Status',
            'Filing on Behalf of Someone'],
           dtype='object')
```

```
[5]: df.head()
```

```
[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.tail()
```

```
[6]:      Ticket #                           Customer Complaint      Date  \
     2219   213550                         Service Availability  04-02-15
     2220   318775  Comcast Monthly Billing for Returned Modem  06-02-15
     2221   331188                      complaint about comcast  06-09-15
     2222   360489      Extremely unsatisfied Comcast customer  23-06-15
     2223   363614        Comcast, Ypsilanti MI Internet Speed  24-06-15

          Date_month_year         Time        Received Via        City     State  \
     2219       04-Feb-15   9:13:18 AM  Customer Care Call  Youngstown   Florida
     2220       06-Feb-15   1:24:39 PM  Customer Care Call   Ypsilanti  Michigan
     2221       06-Sep-15   5:28:41 PM            Internet   Ypsilanti  Michigan
     2222       23-Jun-15  11:13:30 PM  Customer Care Call   Ypsilanti  Michigan
```

```
2223       24-Jun-15  10:28:33 PM  Customer Care Call   Ypsilanti  Michigan

      Zip code  Status Filing on Behalf of Someone
2219     32466  Closed                          No
2220     48197  Solved                          No
2221     48197  Solved                          No
2222     48197  Solved                          No
2223     48198    Open                         Yes
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2224 entries, 0 to 2223
Data columns (total 11 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Ticket #                      2224 non-null   object
 1   Customer Complaint            2224 non-null   object
 2   Date                          2224 non-null   object
 3   Date_month_year               2224 non-null   object
 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
dtypes: int64(1), object(10)
memory usage: 191.2+ KB
```

[8]:
```
df.describe()
```

[8]:
```
           Zip code
count   2224.000000
mean   47994.393435
std    28885.279427
min     1075.000000
25%    30056.500000
50%    37211.000000
75%    77058.750000
max    99223.000000
```

[9]:
```
#Identifying unique values of each columns
def print_unique(df):
    for col in df:
        print("Unique Values of column: ",col,"; No. of unique values:␣
 ↪",len(df[col].unique()))
```

```
        print (df[col].value_counts())
        print ("#"*100)
print_unique(df)
```

```
Unique Values of column:  Ticket # ; No. of unique values:  2224
266179    1
326985    1
358157    1
360707    1
281647    1
         ..
342774    1
372108    1
374265    1
362991    1
231199    1
Name: Ticket #, Length: 2224, dtype: int64
################################################################################
####################
Unique Values of column:  Customer Complaint ; No. of unique values:  1841
Comcast                                                                    83
Comcast Internet                                                           18
Comcast Data Cap                                                           17
comcast                                                                    13
Data Caps                                                                  11
                                                                           ..
Comcast - Fraudulent Billing Practices, Unwilling to resolve situation      1
Comcast Internet Speeds                                                     1
Home Shopping Network Emails                                                1
comcast: no service for one month                                           1
Comcast is giving me the ring around and charged me $130                    1
Name: Customer Complaint, Length: 1841, dtype: int64
################################################################################
####################
Unique Values of column:  Date ; No. of unique values:  91
24-06-15    218
23-06-15    190
25-06-15     98
26-06-15     55
30-06-15     53
           …
05-10-15      7
17-05-15      7
04-05-15      6
05-03-15      5
04-11-15      5
Name: Date, Length: 91, dtype: int64
```

```
################################################################################
####################
Unique Values of column:  Date_month_year ; No. of unique values:  91
24-Jun-15    218
23-Jun-15    190
25-Jun-15     98
26-Jun-15     55
30-Jun-15     53
            …
17-May-15      7
24-May-15      7
04-May-15      6
04-Nov-15      5
05-Mar-15      5
Name: Date_month_year, Length: 91, dtype: int64
################################################################################
####################
Unique Values of column:  Time ; No. of unique values:  2190
2:06:03 PM     2
11:59:36 AM    2
9:56:13 PM     2
11:40:30 PM    2
9:50:41 PM     2
              ..
11:14:49 AM    1
3:46:45 PM     1
9:37:12 PM     1
5:54:14 PM     1
6:46:43 PM     1
Name: Time, Length: 2190, dtype: int64
################################################################################
####################
Unique Values of column:  Received Via ; No. of unique values:  2
Customer Care Call    1119
Internet              1105
Name: Received Via, dtype: int64
################################################################################
####################
Unique Values of column:  City ; No. of unique values:  928
Atlanta         63
Chicago         47
Knoxville       36
Houston         33
Jacksonville    31
                ..
Lincolnwood      1
Grovetown        1
Tupelo           1
```

```
Kenmore          1
Woburn           1
Name: City, Length: 928, dtype: int64
################################################################################
####################
Unique Values of column:  State ; No. of unique values:  43
Georgia                 288
Florida                 240
California              220
Illinois                164
Tennessee               143
Pennsylvania            130
Michigan                115
Washington               98
Colorado                 80
Maryland                 78
New Jersey               75
Texas                    71
Massachusetts            61
Virginia                 60
Indiana                  59
Oregon                   49
Mississippi              39
Minnesota                33
Alabama                  26
Utah                     22
Arizona                  20
South Carolina           18
District Of Columbia     16
New Mexico               15
Louisiana                13
Connecticut              12
New Hampshire            12
Delaware                 12
West Virginia            11
Kentucky                  7
New York                  6
Arkansas                  6
Maine                     5
Missouri                  4
Ohio                      3
North Carolina            3
Vermont                   3
Kansas                    2
Iowa                      1
District of Columbia      1
Rhode Island              1
Montana                   1
```

```
Nevada                      1
Name: State, dtype: int64
################################################################################
####################
Unique Values of column:  Zip code ; No. of unique values:  1543
30144    8
30188    8
30022    7
85718    6
37920    6
        ..
33181    1
20895    1
33189    1
10589    1
55303    1
Name: Zip code, Length: 1543, dtype: int64
################################################################################
####################
Unique Values of column:  Status ; No. of unique values:  4
Solved     973
Closed     734
Open       363
Pending    154
Name: Status, dtype: int64
################################################################################
####################
Unique Values of column:  Filing on Behalf of Someone ; No. of unique values:  2
No     2021
Yes     203
Name: Filing on Behalf of Someone, dtype: int64
################################################################################
####################
```

[10]:
```
#Identifying variables with null values
df.isnull().sum()
```

[10]:
```
Ticket #                       0
Customer Complaint             0
Date                           0
Date_month_year                0
Time                           0
Received Via                   0
City                           0
State                          0
Zip code                       0
Status                         0
```

```
Filing on Behalf of Someone    0
dtype: int64
```

———-There are no NULL Values———-

```
[11]: #Doing Backup
      df_bkp= df.copy()
```

Task 2: Provide the trend chart for the number of complaints at monthly and daily granularity levels

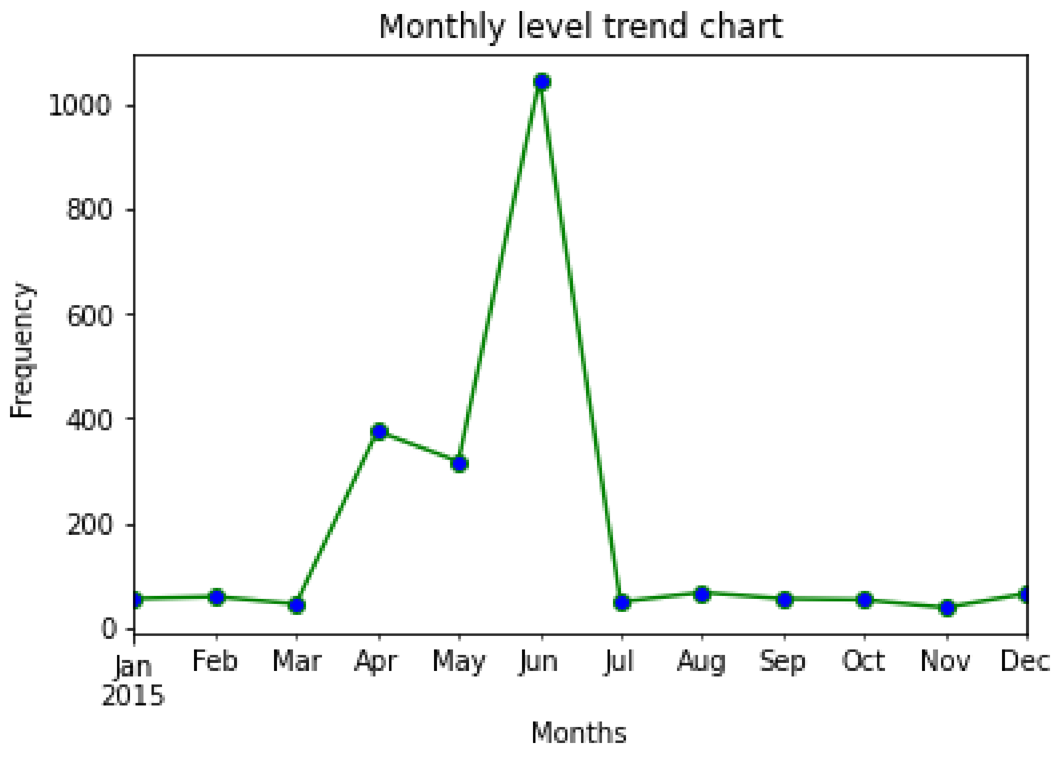
```
[12]: #Converting Dtype of "Date_month_year" from object to datetime
      df['Date_month_year']= pd.to_datetime(df['Date_month_year'])
      #Setting "Date_month_year" as index
      df=df.set_index('Date_month_year')
```

```
[13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2224 entries, 2015-04-22 to 2015-06-24
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Ticket #                     2224 non-null   object
 1   Customer Complaint           2224 non-null   object
 2   Date                         2224 non-null   object
 3   Time                         2224 non-null   object
 4   Received Via                 2224 non-null   object
 5   City                         2224 non-null   object
 6   State                        2224 non-null   object
 7   Zip code                     2224 non-null   int64
 8   Status                       2224 non-null   object
 9   Filing on Behalf of Someone  2224 non-null   object
dtypes: int64(1), object(9)
memory usage: 191.1+ KB
```

```
[14]: #Plotting a Monthly Chart
      months= df.groupby(pd.Grouper(freq="M")).size().plot(color='green',
               marker='o', markerfacecolor='blue')
      plt.xlabel("Months")
      plt.ylabel("Frequency")
      plt.title("Monthly level trend chart")
```

```
[14]: Text(0.5, 1.0, 'Monthly level trend chart')
```



———-Above chart shows that ”June 2015” has maximum Complaints———

```
[15]: #Counting unique values for "Date" column
      df['Date'].value_counts()[:10]
```

```
[15]: 24-06-15    218
      23-06-15    190
      25-06-15     98
      26-06-15     55
      30-06-15     53
      29-06-15     51
      18-06-15     47
      06-12-15     43
      27-06-15     39
      15-06-15     34
      Name: Date, dtype: int64
```

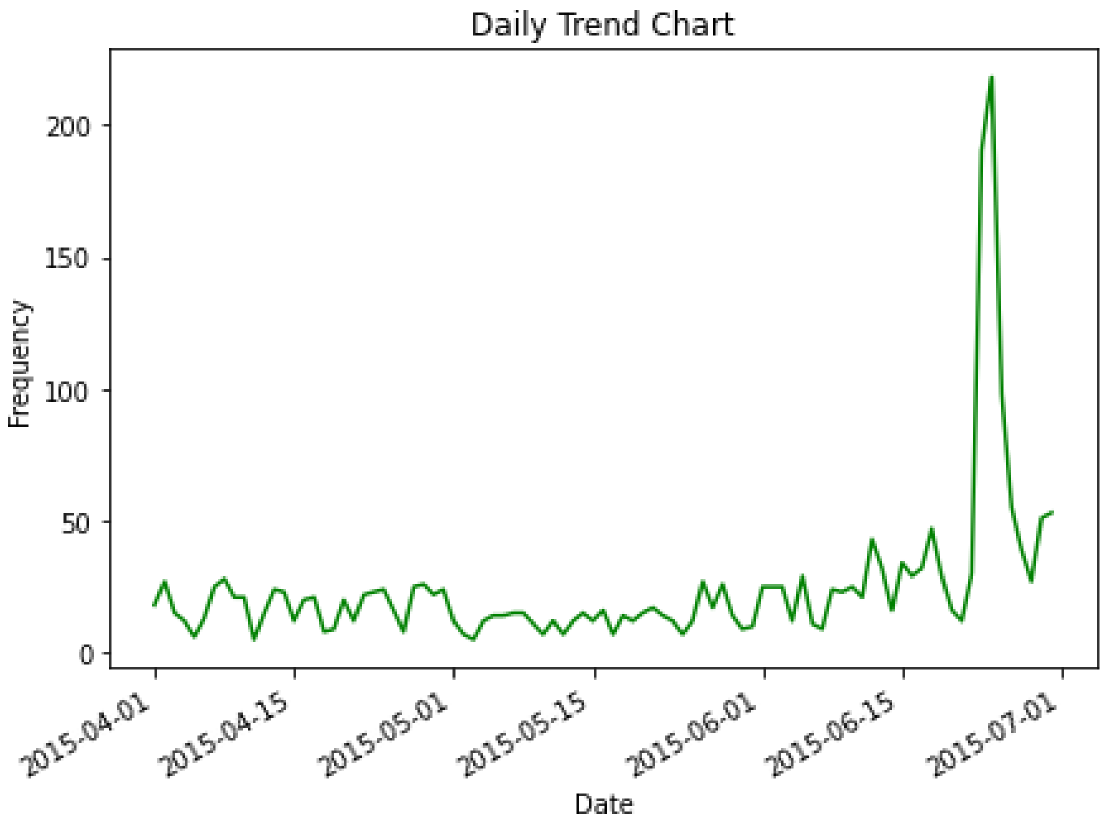
```
[16]: #Converting Dtype of "Date" from object to datetime
      df['Date']= pd.to_datetime(df['Date'])
```

```
[17]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2224 entries, 2015-04-22 to 2015-06-24
Data columns (total 10 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   Time                         2224 non-null   object
 4   Received Via                 2224 non-null   object
 5   City                         2224 non-null   object
 6   State                        2224 non-null   object
 7   Zip code                     2224 non-null   int64
 8   Status                       2224 non-null   object
 9   Filing on Behalf of Someone  2224 non-null   object
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 191.1+ KB
```

[18]:
```
#Sorting the "Date" column
df= df.sort_values("Date")
#Plotting daily chart
df['Date'].value_counts().plot(figsize=(7,5), color='green')
plt.xlabel("Date")
plt.ylabel("Frequency")
plt.title("Daily Trend Chart")
```

[18]:
```
Text(0.5, 1.0, 'Daily Trend Chart')
```

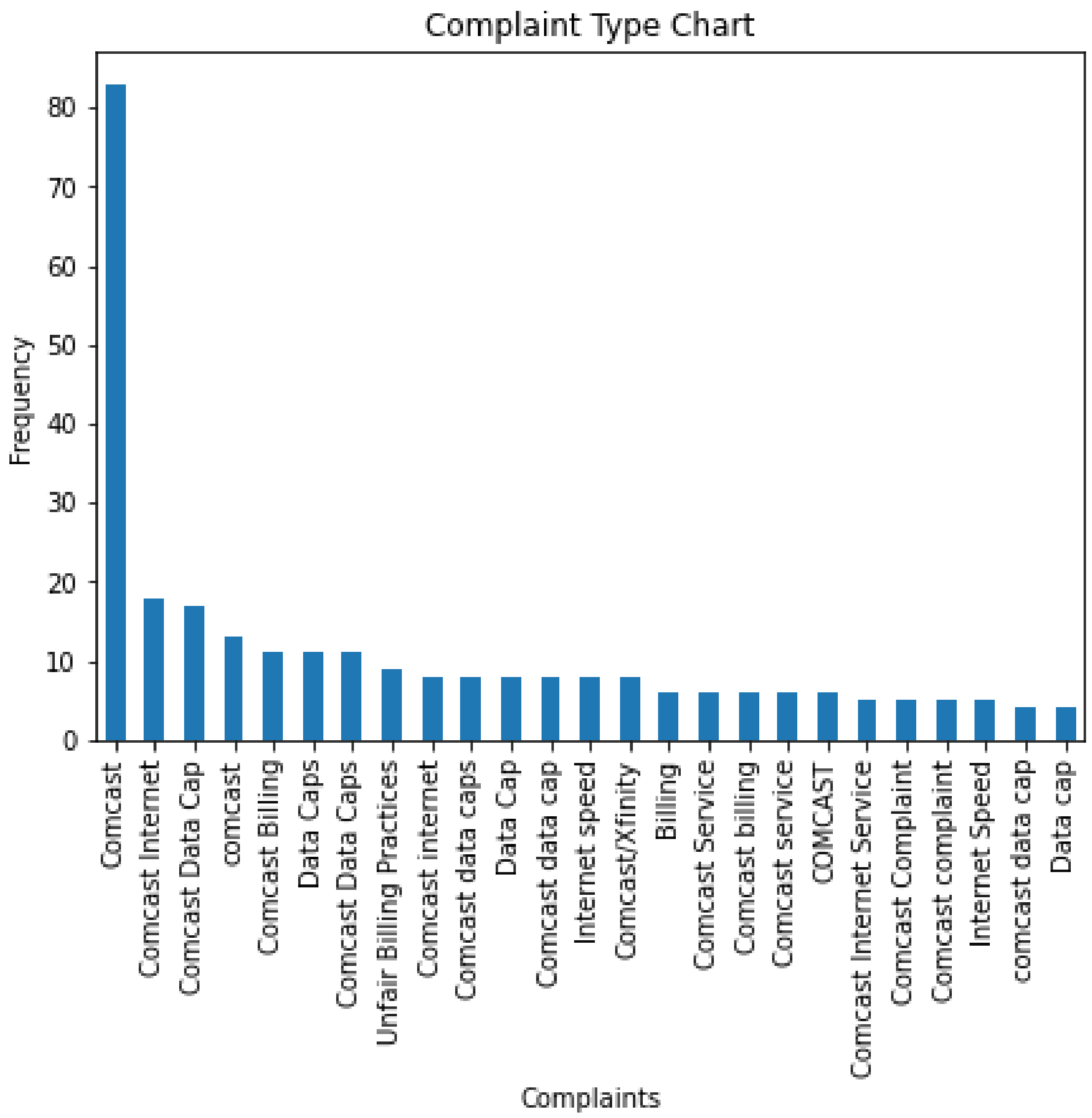


———-Above chart shows that there is maximum complaints between 15th June 2015 and 1st July 2015———-

Task 3: Provide a table with the frequency of complaint types

```
[19]: df['Customer Complaint'].value_counts().head(25).plot(kind='bar', figsize=(7,5))
      plt.xlabel("Complaints")
      plt.ylabel("Frequency")
      plt.title("Complaint Type Chart")
```

```
[19]: Text(0.5, 1.0, 'Complaint Type Chart')
```



————Above chart shows "Comcast" type of complaints are more————-

Task 4: Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

[20]:
```
internet_issues1= df[df['Customer Complaint'].str.contains('network')].count()
internet_issues2= df[df['Customer Complaint'].str.contains('speed')].count()
internet_issues3= df[df['Customer Complaint'].str.contains('data')].count()
internet_issues4= df[df['Customer Complaint'].str.contains('internet')].count()
billing_issues1= df[df['Customer Complaint'].str.contains('bill')].count()
billing_issues2= df[df['Customer Complaint'].str.contains('billing')].count()
billing_issues3= df[df['Customer Complaint'].str.contains('charges')].count()
service_issues1= df[df['Customer Complaint'].str.contains('service')].count()
service_issues2= df[df['Customer Complaint'].str.contains('customer')].count()
```

[21]:
```
total_internet_issues = internet_issues1 + internet_issues2 + internet_issues3␣
↪+ internet_issues4
print(total_internet_issues)
```

```
Ticket #                       374
Customer Complaint             374
Date                           374
Time                           374
Received Via                   374
City                           374
State                          374
Zip code                       374
Status                         374
Filing on Behalf of Someone    374
dtype: int64
```

[22]:
```
total_billing_issues = billing_issues1 + billing_issues2 + billing_issues3
print(total_billing_issues)
```

```
Ticket #                       353
Customer Complaint             353
Date                           353
Time                           353
Received Via                   353
City                           353
State                          353
Zip code                       353
Status                         353
Filing on Behalf of Someone    353
dtype: int64
```

[23]:
```
total_service_issues = service_issues1 + service_issues2
print(total_service_issues)
```

```
Ticket #                       360
Customer Complaint             360
Date                           360
Time                           360
Received Via                   360
City                           360
State                          360
Zip code                       360
Status                         360
Filing on Behalf of Someone    360
dtype: int64
```

[24]:
```
other_issues = 2224- (total_internet_issues + total_billing_issues +␣
↪total_service_issues)
print(other_issues)
```

```
Ticket #                       1137
Customer Complaint             1137
Date                           1137
Time                           1137
Received Via                   1137
City                           1137
State                          1137
Zip code                       1137
Status                         1137
Filing on Behalf of Someone    1137
dtype: int64
```

———-The above analysis shows that other issues are maximum———-

Task 5: 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

[25]:
```
df.Status.unique()
```

[25]:
```
array(['Closed', 'Open', 'Solved', 'Pending'], dtype=object)
```

[26]:
```
df["new_status"] = ["Open" if Status == "Open" or Status == "Pending" else␣
↪"Closed" for Status in df["Status"]]
df=df.drop(["Status"], axis=1)
df
```

[26]:
```
                Ticket #                                 Customer Complaint  \
Date_month_year
2015-01-04        211677                          Comcast refusal of service
2015-01-04        211976   Fraudulent claims reported to collections agency
2015-01-04        211478                                             Comcast
2015-01-04        211904  Unable to get in touch with anyone that has th…
2015-01-04        212381  Comcast speeds as low as 12 MB/s, paying for 1…
…                  …                                                  …
2015-06-30        376328  Comcast Failed to deliver service that was adv…
2015-06-30        375847   Comcast bundles useless services to charge more.
2015-06-30        375249                                       Comcast cable
2015-06-30        375292                         Underhanded sales techniques
2015-06-30        376295                               Slow internet service

                      Date         Time        Received Via             City  \
Date_month_year
2015-01-04      2015-04-01  12:01:06 PM  Customer Care Call            Wayne
```

```
2015-01-04      2015-04-01   1:26:53 PM  Customer Care Call           Atlanta
2015-01-04      2015-04-01  10:47:35 AM            Internet  North Huntingdon
2015-01-04      2015-04-01   1:06:33 PM  Customer Care Call        Huntsville
2015-01-04      2015-04-01   3:10:12 PM  Customer Care Call        Washington
…                        …           …                  …                 …
2015-06-30      2015-06-30  11:24:39 PM            Internet           Houston
2015-06-30      2015-06-30   6:36:52 PM            Internet           Houston
2015-06-30      2015-06-30   3:47:29 PM            Internet      Beach Haven
2015-06-30      2015-06-30   3:59:45 PM            Internet          Lakewood
2015-06-30      2015-06-30  10:57:27 PM            Internet       White House

                       State  Zip code Filing on Behalf of Someone new_status
Date_month_year
2015-01-04      Pennsylvania     19087                          No     Closed
2015-01-04           Georgia     30312                          No     Closed
2015-01-04      Pennsylvania     15642                          No     Closed
2015-01-04           Alabama     35801                          No     Closed
2015-01-04      Pennsylvania     15301                          No     Closed
…                        …           …                          …        …
2015-06-30             Texas     77064                          No       Open
2015-06-30             Texas     77025                          No       Open
2015-06-30        New Jersey      8008                          No       Open
2015-06-30          Colorado     80215                          No       Open
2015-06-30         Tennessee     37188                          No       Open

[2224 rows x 10 columns]
```

Task 6: - Provide state wise status of complaints in a stacked bar chart

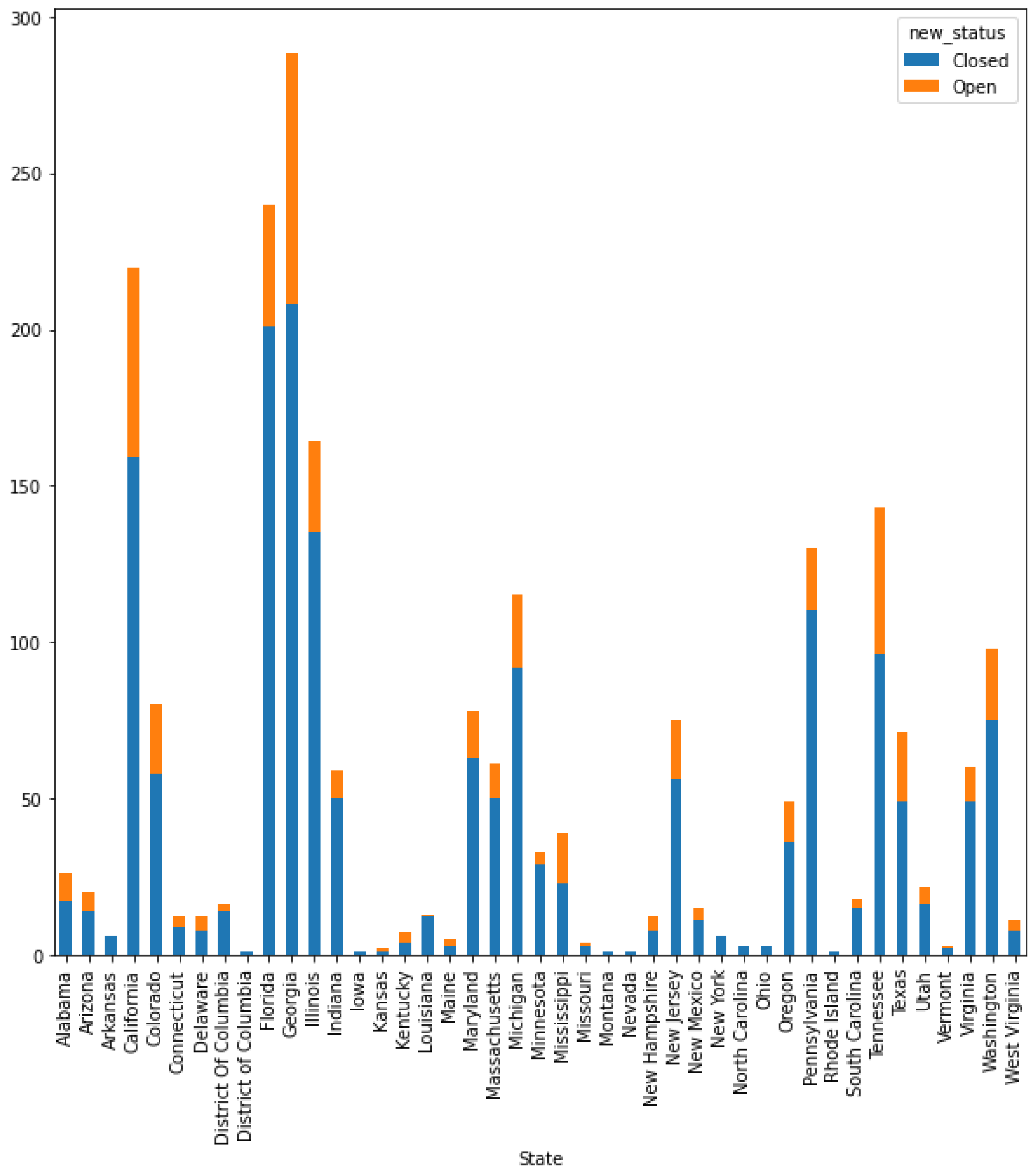
[27]:
```python
statewise_complaints = df.groupby(["State","new_status"]).size().unstack()
print(statewise_complaints)
```

```
new_status            Closed  Open
State
Alabama                 17.0   9.0
Arizona                 14.0   6.0
Arkansas                 6.0   NaN
California             159.0  61.0
Colorado                58.0  22.0
Connecticut              9.0   3.0
Delaware                 8.0   4.0
District Of Columbia    14.0   2.0
District of Columbia     1.0   NaN
Florida                201.0  39.0
Georgia                208.0  80.0
Illinois               135.0  29.0
Indiana                 50.0   9.0
```

```
Iowa                 1.0   NaN
Kansas               1.0   1.0
Kentucky             4.0   3.0
Louisiana           12.0   1.0
Maine                3.0   2.0
Maryland            63.0  15.0
Massachusetts       50.0  11.0
Michigan            92.0  23.0
Minnesota           29.0   4.0
Mississippi         23.0  16.0
Missouri             3.0   1.0
Montana              1.0   NaN
Nevada               1.0   NaN
New Hampshire        8.0   4.0
New Jersey          56.0  19.0
New Mexico          11.0   4.0
New York             6.0   NaN
North Carolina       3.0   NaN
Ohio                 3.0   NaN
Oregon              36.0  13.0
Pennsylvania       110.0  20.0
Rhode Island         1.0   NaN
South Carolina      15.0   3.0
Tennessee           96.0  47.0
Texas               49.0  22.0
Utah                16.0   6.0
Vermont              2.0   1.0
Virginia            49.0  11.0
Washington          75.0  23.0
West Virginia        8.0   3.0
```

[28]:
```
statewise_complaints.plot(kind='bar', figsize=(10,10),stacked=True)
```

[28]:
```
<AxesSubplot:xlabel='State'>
```



Task 7: Which state has the maximum complaints

[29]:
```
df.groupby(['State']).size().sort_values(ascending=False)[:10]
```

[29]:
```
State
Georgia        288
Florida        240
California     220
```

```
Illinois        164
Tennessee       143
Pennsylvania    130
Michigan        115
Washington       98
Colorado         80
Maryland         78
dtype: int64
```

———–Above analysis shows that ”Georgia” has maximum complaints———-

Task 8: Which state has the highest percentage of unresolved complaints

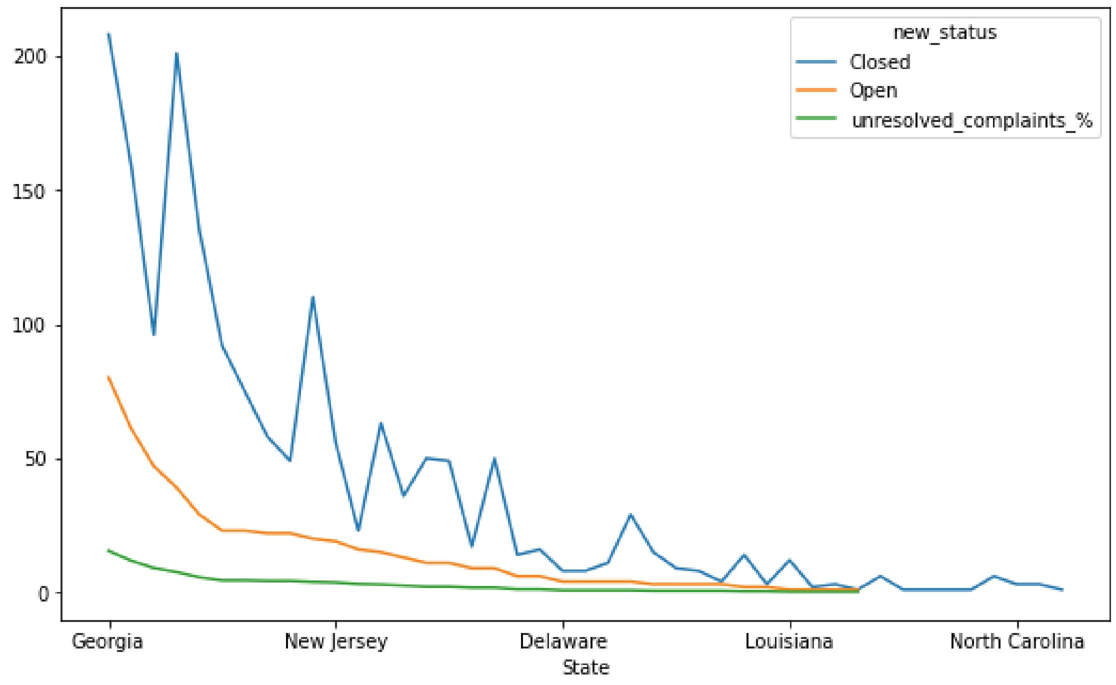
[30]:
```
#Using "Open" Status as it is "Unresolved"
unresolved_complaints = df.groupby(["State","new_status"]).size().unstack().
↪sort_values("Open", ascending= False)
unresolved_complaints['unresolved_complaints_%'] =␣
↪(unresolved_complaints['Open']/unresolved_complaints['Open'].sum())*100
print(unresolved_complaints)
```

```
new_status            Closed  Open  unresolved_complaints_%
State
Georgia                208.0  80.0                15.473888
California             159.0  61.0                11.798839
Tennessee               96.0  47.0                 9.090909
Florida                201.0  39.0                 7.543520
Illinois               135.0  29.0                 5.609284
Michigan                92.0  23.0                 4.448743
Washington              75.0  23.0                 4.448743
Colorado                58.0  22.0                 4.255319
Texas                   49.0  22.0                 4.255319
Pennsylvania           110.0  20.0                 3.868472
New Jersey              56.0  19.0                 3.675048
Mississippi             23.0  16.0                 3.094778
Maryland                63.0  15.0                 2.901354
Oregon                  36.0  13.0                 2.514507
Massachusetts           50.0  11.0                 2.127660
Virginia                49.0  11.0                 2.127660
Alabama                 17.0   9.0                 1.740812
Indiana                 50.0   9.0                 1.740812
Arizona                 14.0   6.0                 1.160542
Utah                    16.0   6.0                 1.160542
Delaware                 8.0   4.0                 0.773694
New Hampshire            8.0   4.0                 0.773694
New Mexico              11.0   4.0                 0.773694
Minnesota               29.0   4.0                 0.773694
South Carolina          15.0   3.0                 0.580271
```

```
Connecticut              9.0   3.0                  0.580271
West Virginia            8.0   3.0                  0.580271
Kentucky                 4.0   3.0                  0.580271
District Of Columbia    14.0   2.0                  0.386847
Maine                    3.0   2.0                  0.386847
Louisiana               12.0   1.0                  0.193424
Vermont                  2.0   1.0                  0.193424
Missouri                 3.0   1.0                  0.193424
Kansas                   1.0   1.0                  0.193424
Arkansas                 6.0   NaN                       NaN
District of Columbia     1.0   NaN                       NaN
Iowa                     1.0   NaN                       NaN
Montana                  1.0   NaN                       NaN
Nevada                   1.0   NaN                       NaN
New York                 6.0   NaN                       NaN
North Carolina           3.0   NaN                       NaN
Ohio                     3.0   NaN                       NaN
Rhode Island             1.0   NaN                       NaN
```

[31]:
```
unresolved_complaints.plot(figsize=(10,6))
```

[31]:
```
<AxesSubplot:xlabel='State'>
```



———-Above analysis shows that ”Georgia” state has the highest percentage of unresolved complaints———-

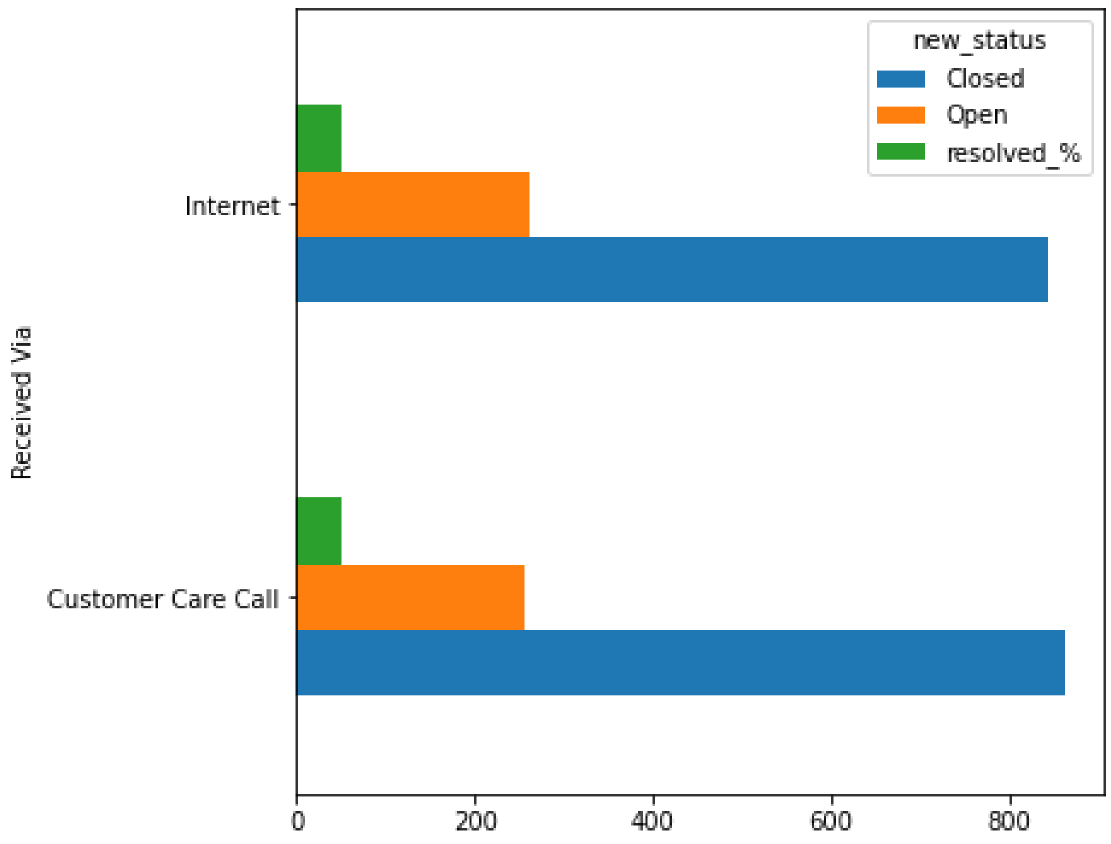
Task 9: Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls

[32]:
```
resolved_data = df.groupby(["Received Via","new_status"]).size().unstack()
resolved_data['resolved_%'] = (resolved_data['Closed']/resolved_data['Closed'].
 ↪sum())*100
resolved_data['resolved_%']
```

[32]:
```
Received Via
Customer Care Call    50.615114
Internet              49.384886
Name: resolved_%, dtype: float64
```

[33]:
```
resolved_data.plot(kind="barh", figsize=(6,6))
```

[33]:
```
<AxesSubplot:ylabel='Received Via'>
```



----------50.6% & 49.38% Complaints resolved till date received through the Internet and customer care calls respectively----------

[ ]:
```
```