

Comcast telecommunication Complaints Project

Importing Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Loading Dataset

In [2]:

```
df = pd.read_csv("Comcast_telecom_complaints_data.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101
3	277946	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	05-Jul-15	11:59:35 AM	Internet	Acworth	Georgia	30101
4	307175	Comcast not working and no service to boot	26-05-15	26-May-15	1:25:26 PM	Internet	Acworth	Georgia	30101

◀ ▶

In [4]:

```
print(df.isnull().sum())
```

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 Nan values present in Dataset

In [5]:

```
df.describe(include='all')
```

Out[5]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Z
count	2224	2224	2224		2224	2224	2224	2224	2224
unique	2224	1841	91		91	2190	2	928	43
top	373309	Comcast	24-06-15	24-Jun-15	12:22:34 AM	Customer Care Call	Atlanta	Georgia	
freq	1	83	218		218	2	1119	63	288
mean	NaN	NaN	NaN		NaN	NaN	NaN	NaN	47994
std	NaN	NaN	NaN		NaN	NaN	NaN	NaN	28885
min	NaN	NaN	NaN		NaN	NaN	NaN	NaN	1075
25%	NaN	NaN	NaN		NaN	NaN	NaN	NaN	30056
50%	NaN	NaN	NaN		NaN	NaN	NaN	NaN	37211
75%	NaN	NaN	NaN		NaN	NaN	NaN	NaN	77058
max	NaN	NaN	NaN		NaN	NaN	NaN	NaN	99223

◀ ▶

In [6]:

```
df.shape
```

Out[6]: (2224, 11)

In [7]:

```
df= df.drop(['Ticket #','Time'], axis=1)
```

In [8]:

```
df.head()
```

	Customer Complaint	Date	Date_month_year	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone
0	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	Customer Care Call	Abingdon	Maryland	21009	Closed	N

	Customer Complaint	Date	Date_month_year	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone
1	Payment disappear - service got disconnected	04-08-15	04-Aug-15	Internet	Acworth	Georgia	30102	Closed	N
2	Speed and Service	18-04-15	18-Apr-15	Internet	Acworth	Georgia	30101	Closed	Ye
3	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	05-Jul-15	Internet	Acworth	Georgia	30101	Open	Ye
4	Comcast not working and no service to boot	26-05-15	26-May-15	Internet	Acworth	Georgia	30101	Solved	N

Task-1 Provide the trend chart for the number of complaints at monthly and daily granularity levels.

In [9]:

```
# Pandas to_datetime() method helps to convert string Date time into Python Date
df['Date_month_year']=df['Date_month_year'].apply(pd.to_datetime)
# Setting 'Date_month_year' as index
df=df.set_index('Date_month_year')
```

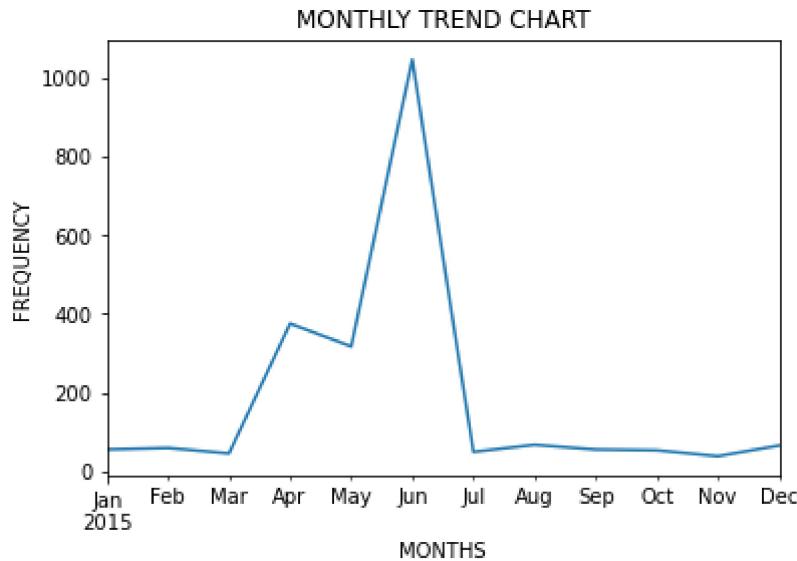
plotting monthly chart

In [10]:

```
# dataframe.groupby() function is splitting the data into groups accoring to freq
months= df.groupby(pd.Grouper(freq="M")).size().plot()
plt.xlabel("MONTHS")
plt.ylabel("FREQUENCY")
plt.title("MONTHLY TREND CHART")
```

Out[10]:

Text(0.5, 1.0, 'MONTHLY TREND CHART')



INSIGHTS:- From the above trend chart, we can clearly see that complaints for the month of June 2015 are maximum

In [11]:

```
#value_counts() function is getting a Series containing counts of unique values from df['Date'].value_counts(dropna=False)[:8]
```

Out[11]:

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

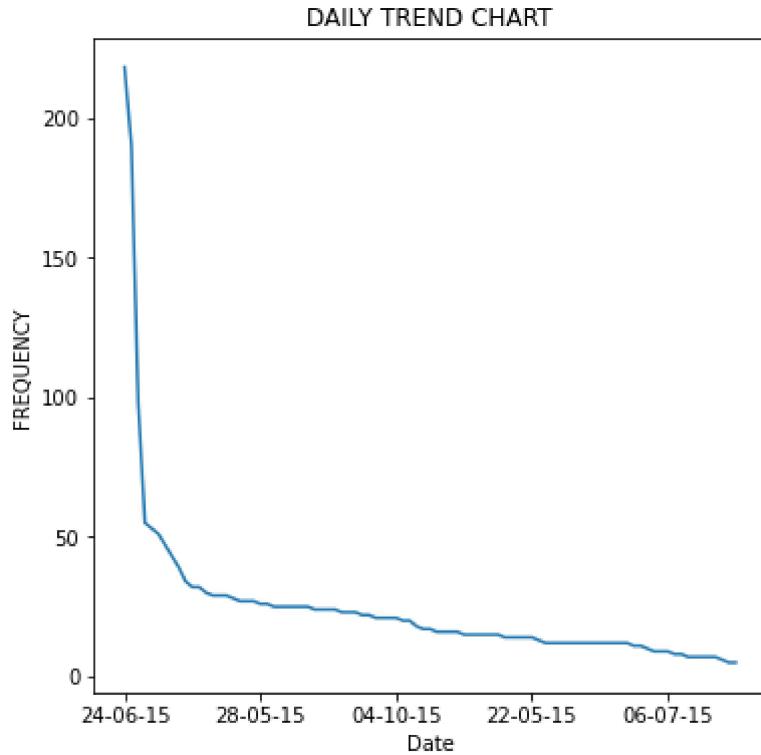
Name: Date, dtype: int64

plotting daily chart

In [12]:

```
df = df.sort_values(by='Date')
plt.figure(figsize=(6,6))
df['Date'].value_counts().plot()
plt.xlabel("Date")
plt.ylabel("FREQUENCY")
plt.title("DAILY TREND CHART")
```

Out[12]: Text(0.5, 1.0, 'DAILY TREND CHART')



In []:

```
df['Customer Complaint'].value_counts(dropna=False)[:9]
```

Out[13]:

Comcast	83
Comcast Internet	18
Comcast Data Cap	17
comcast	13
Comcast Billing	11
Data Caps	11
Comcast Data Caps	11
Unfair Billing Practices	9
Comcast data cap	8

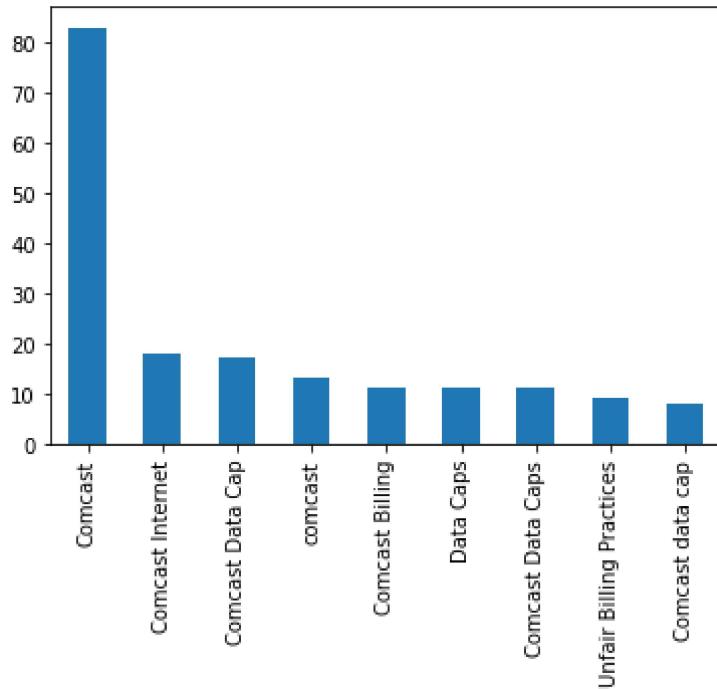
Name: Customer Complaint, dtype: int64

In [14]:

```
df['Customer Complaint'].value_counts(dropna=False)[:9].plot.bar()
```

Out[14]:

```
<AxesSubplot:>
```



Task 3-Which complaint types are maximum i.e., around internet, network issues, or across any other domains.

```
In [15]: internet_issues1=df[df['Customer Complaint'].str.contains("network")].count()

In [16]: internet_issues2=df[df['Customer Complaint'].str.contains("speed")].count()

In [17]: internet_issues3=df[df['Customer Complaint'].str.contains("data")].count()

In [18]: internet_issues4=df[df['Customer Complaint'].str.contains("internet")].count()

In [19]: billing_issues1=df[df['Customer Complaint'].str.contains("bill")].count()

In [20]: billing_issues2=df[df['Customer Complaint'].str.contains("billing")].count()

In [21]: billing_issues3=df[df['Customer Complaint'].str.contains("charges")].count()

In [22]: service_issues1=df[df['Customer Complaint'].str.contains("service")].count()

In [23]: service_issues2=df[df['Customer Complaint'].str.contains("customer")].count()

In [24]: total_internet_issues=internet_issues1+internet_issues2+internet_issues3+internet_issues4
print(total_internet_issues)
```

Customer Complaint	374
Date	374
Received Via	374

```

City           374
State          374
Zip code       374
Status          374
Filing on Behalf of Someone  374
dtype: int64

```

In [25]:

```

total_billing_issues=billing_issues1+billing_issues2+billing_issues3
print(total_billing_issues)

```

```

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

```

In [26]:

```

total_service_issues=service_issues1+service_issues2
print(total_service_issues)

```

```

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

```

In [27]:

```

other_issues=2224-(total_internet_issues+total_billing_issues+total_service_issues)
print(other_issues)

```

```

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

```

INSIGHTS:- From the above analysis we can see that the other issues are maximum.

4.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.

In [28]:

```

df.Status.unique()

```

Out[28]:

```

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

```

In [29]:

```

df["newStatus"] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" for
df= df.drop(['Status'], axis=1)

```

df

Out[29]:

	Customer Complaint	Date	Received Via	City	State	Zip code	Filing on Behalf of Someone
Date_month_year							
2015-01-04	Fraudulent claims reported to collections agency	04-01-15	Customer Care Call	Atlanta	Georgia	30312	No
2015-01-04	Comcast refusal of service	04-01-15	Customer Care Call	Wayne	Pennsylvania	19087	No
2015-01-04	Comcast Cable	04-01-15	Internet	Franklin	Tennessee	37067	No
2015-01-04	Data Overages	04-01-15	Internet	Savannah	Georgia	31406	No
2015-01-04	Comcast	04-01-15	Internet	North Huntingdon	Pennsylvania	15642	No
...
2015-05-31	Comcast	31-05-15	Customer Care Call	Beaverton	Oregon	97006	No
2015-05-31	Comcast of East Windsor NJ Complaint	31-05-15	Internet	East Windsor	New Jersey	8520	No
2015-05-31	n/a (b) (6)	31-05-15	Internet	Loganville	Georgia	30052	No
2015-05-31	Complaint against Comcast for incredibly bad s...	31-05-15	Customer Care Call	Edgewood	Washington	98372	No
2015-05-31	Questionable internet slowdown	31-05-15	Customer Care Call	Peabody	Massachusetts	1960	No

2224 rows × 8 columns

In []:

◀ ▶

5. Which state has the maximum complaints

In [30]:

```
df.groupby(["State"]).size().sort_values(ascending=False)[:5]
```

Out[30]:

State	
Georgia	288
Florida	240
California	220
Illinois	164
Tennessee	143
dtype: int64	

INSIGHTS:- From the above table, we can clearly see that Georgia has maximum complaints.

In []:

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

In [31]:

```
Status_complaints = df.groupby(["State", "newStatus"]).size().unstack()
print(Status_complaints)
```

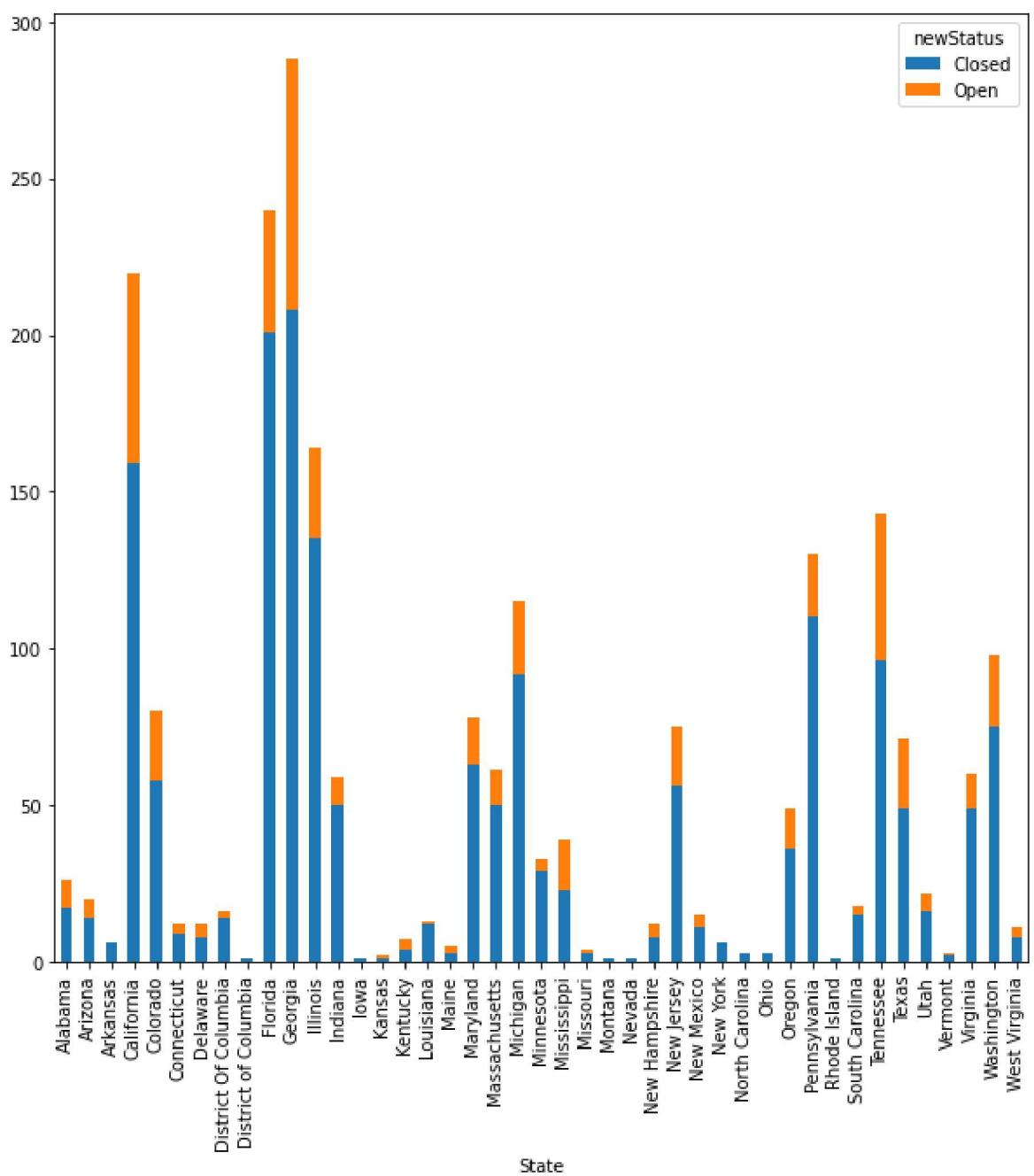
newStatus	Closed	Open
State		
Alabama	17.0	9.0
Arizona	14.0	6.0
Arkansas	6.0	NaN
California	159.0	61.0
Colorado	58.0	22.0
Connecticut	9.0	3.0
Delaware	8.0	4.0
District Of Columbia	14.0	2.0
District of Columbia	1.0	NaN
Florida	201.0	39.0
Georgia	208.0	80.0
Illinois	135.0	29.0
Indiana	50.0	9.0
Iowa	1.0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

In [32]:

```
status_complaints.plot.bar(figsize=(10,10), stacked=True)
```

Out[32]:



INSIGHTS:- From the above chart, we can clearly see that Georgia has maximum complaints.

Task-7 state which has the highest percentage of unresolved complaints

In [33]:

```
print(df['newStatus'].value_counts())
```

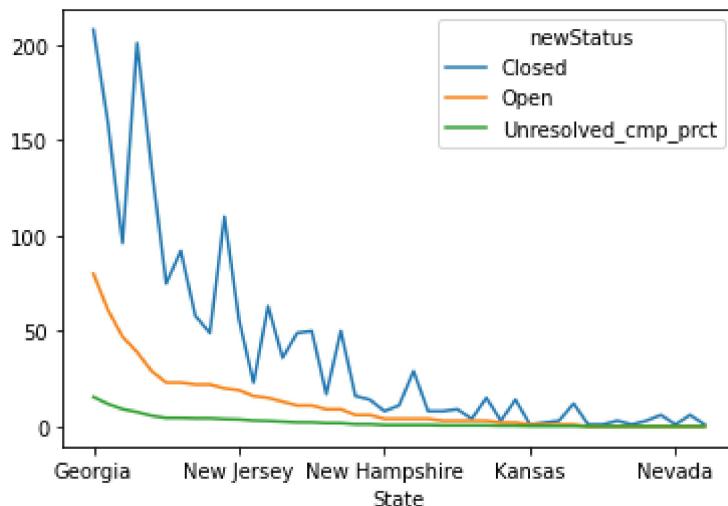
```
Closed    1707
Open      517
Name: newStatus, dtype: int64
```

In [34]:

```
unresolved_data = df.groupby(['State','newStatus']).size().unstack().fillna(0).sort_index()
unresolved_data['Unresolved_cmp_prct'] = unresolved_data['Open']/unresolved_data['Closed']
print(unresolved_data)
unresolved_data.plot()
```

newStatus	Closed	Open	Unresolved_cmp_prct
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
Washington	75.0	23.0	4.448743
Michigan	92.0	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
Virginia	49.0	11.0	2.127660
Massachusetts	50.0	11.0	2.127660
Alabama	17.0	9.0	1.740812
Indiana	50.0	9.0	1.740812
Utah	16.0	6.0	1.160542
Arizona	14.0	6.0	1.160542
New Hampshire	8.0	4.0	0.773694
New Mexico	11.0	4.0	0.773694
Minnesota	29.0	4.0	0.773694
Delaware	8.0	4.0	0.773694
West Virginia	8.0	3.0	0.580271
Connecticut	9.0	3.0	0.580271
Kentucky	4.0	3.0	0.580271
South Carolina	15.0	3.0	0.580271
Maine	3.0	2.0	0.386847
District Of Columbia	14.0	2.0	0.386847
Kansas	1.0	1.0	0.193424
Vermont	2.0	1.0	0.193424
Missouri	3.0	1.0	0.193424
Louisiana	12.0	1.0	0.193424
Montana	1.0	0.0	0.000000
Rhode Island	1.0	0.0	0.000000
Ohio	3.0	0.0	0.000000
District of Columbia	1.0	0.0	0.000000
North Carolina	3.0	0.0	0.000000
New York	6.0	0.0	0.000000
Nevada	1.0	0.0	0.000000
Arkansas	6.0	0.0	0.000000
Iowa	1.0	0.0	0.000000

Out[34]:



INSIGHTS:- From the table generated above we can see that Georgia has maximum unresolved complaints i.e. 80.

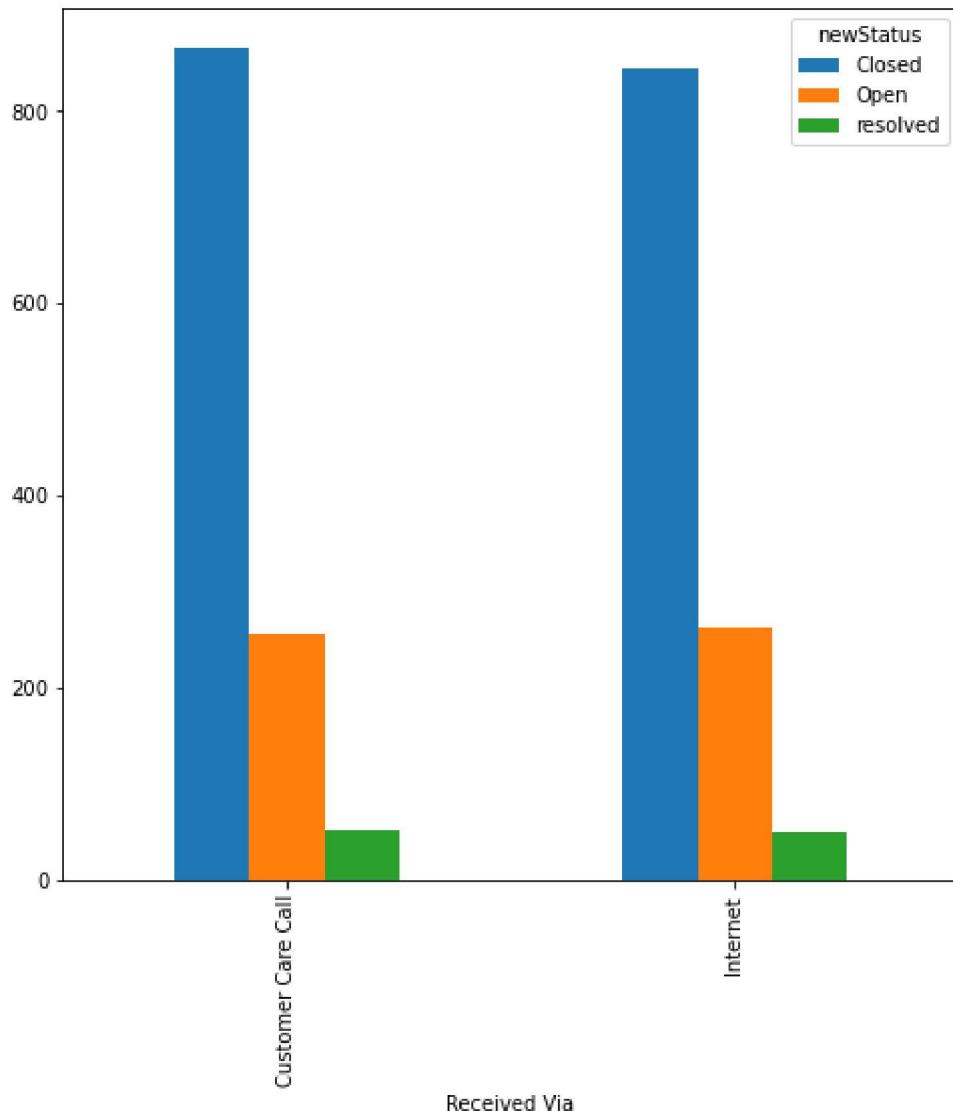
Task-8 Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls

```
In [35]: resolved_data = df.groupby(['Received Via', 'newStatus']).size().unstack().fillna(0)
resolved_data['resolved'] = resolved_data['Closed'] / resolved_data['Closed'].sum()
resolved_data['resolved']
```

```
Out[35]: Received Via
Customer Care Call      50.615114
Internet                 49.384886
Name: resolved, dtype: float64
```

```
In [36]: resolved_data.plot(kind="bar", figsize=(8,8))
```

```
Out[36]: <AxesSubplot:xlabel='Received Via'>
```



INSIGHTS:- From the above pie chart we can clearly see that there are total 50.61% Complaints resolved for Customer Care Call and 49.39% for received via internet.