In [1]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
```

Import data into Python environment.

In [2]:
```
# Load the dataset
df=pd.read_csv(r'D:\AI Engineer Masters\1.Project 3 - Comcast Telecom/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	Status	Filing on Behalf of Someone
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes
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	Open	Yes
4	307175	Comcast not working and no service to boot	26-05-15	26-May-15	1:25:26 PM	Internet	Acworth	Georgia	30101	Solved	No

In [4]:
```
# check if any nulls
df.isnull().sum()
```

Out[4]:
```
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
```

In [5]:
```
df.describe(include='all')
```

Out[5]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone
count	2224	2224	2224	2224	2224	2224	2224	2224	2224.000000	2224	2224
unique	2224	1841	91	91	2190	2	928	43	NaN	4	2
top	361078	Comcast	24-06-15	24-Jun-15	11:59:36 AM	Customer Care Call	Atlanta	Georgia	NaN	Solved	No
freq	1	83	218	218	2	1119	63	288	NaN	973	2021
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	47994.393435	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28885.279427	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1075.000000	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	30056.500000	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	37211.000000	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	77058.750000	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	99223.000000	NaN	NaN

In [6]:
```
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
```

In [7]:
```
df.shape
```

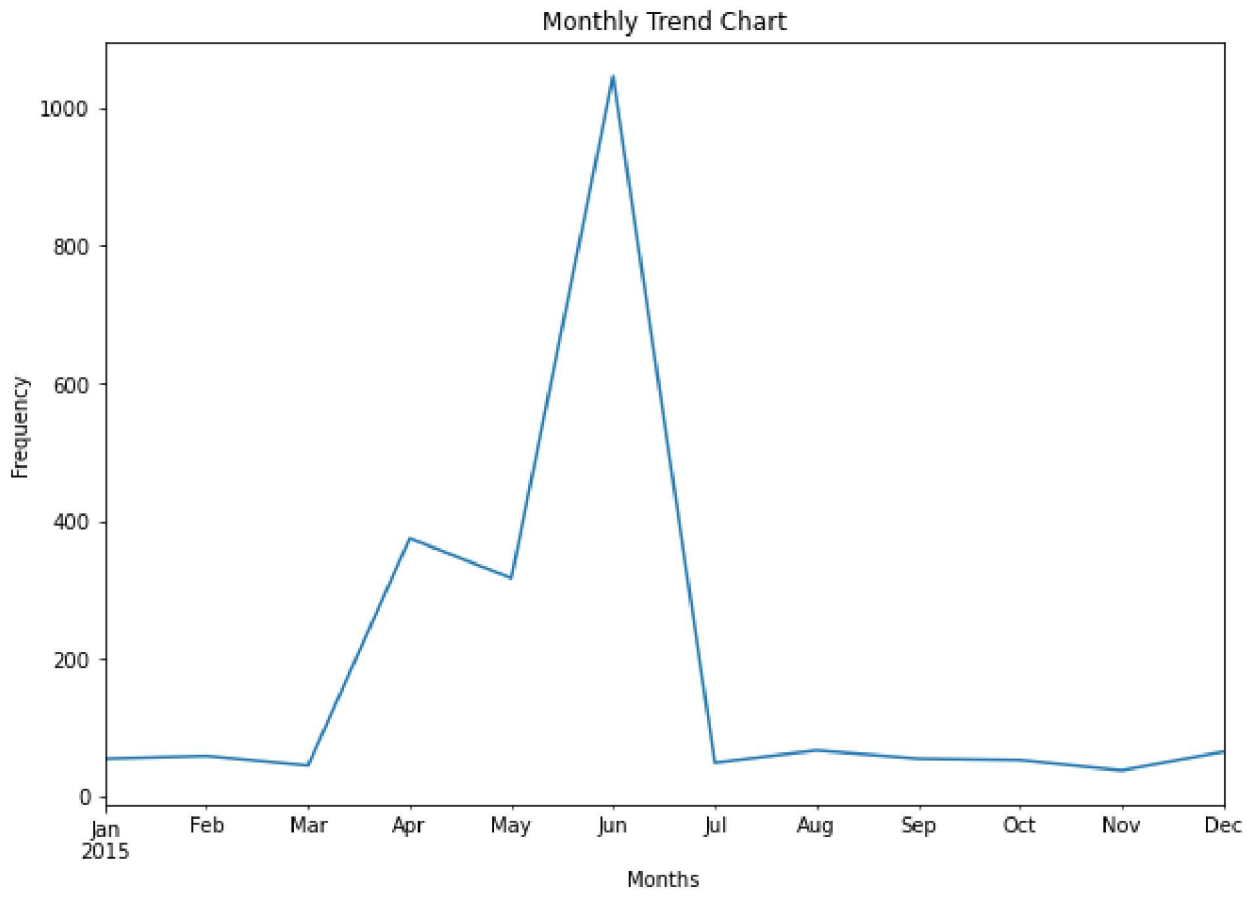
Out[7]:
```
(2224, 11)
```

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

In [8]:
```
# convert field to datetime
df['Date_month_year']=df['Date_month_year'].apply(pd.to_datetime)
```

In [9]:
```
# reset index to datetime
df=df.set_index('Date_month_year')
```

In [10]:
```
#plotting Monthly chart
df.groupby(pd.Grouper(freq='M')).size().plot(figsize=(10,7))
plt.xlabel('Months')
plt.ylabel('Frequency')
plt.title('Monthly Trend Chart')
plt.show()
```



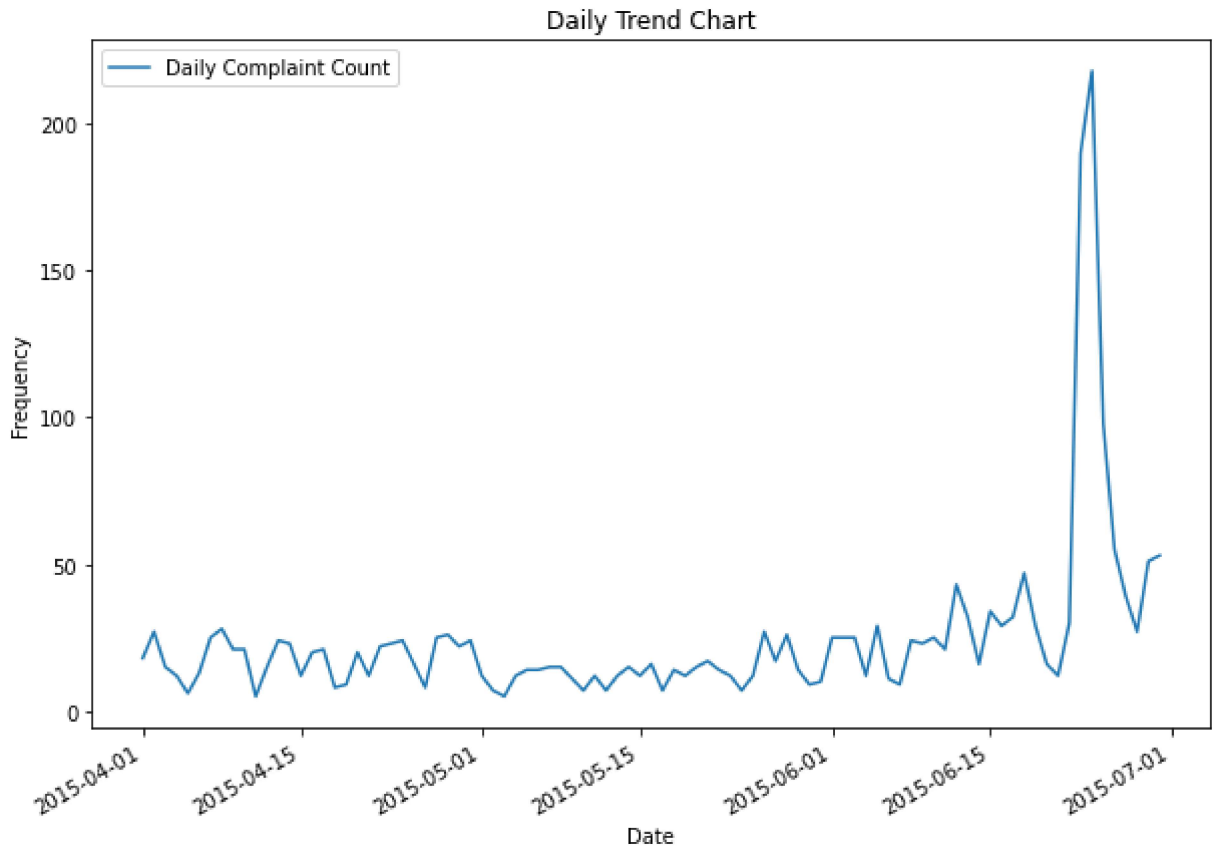
Inferrence : We see most of the complaints in June month

In [11]:
```
daily=df['Date'].value_counts().reset_index(name='Counts')
daily.columns=['Date','Daily Complaint Count']
daily=daily.sort_values('Date')
daily['Date']=daily['Date'].apply(pd.to_datetime)
daily=daily.set_index('Date')
daily.sort_values('Daily Complaint Count',ascending=False).head()
```

Out[11]:

Date	Daily Complaint Count
2015-06-24	218
2015-06-23	190
2015-06-25	98
2015-06-26	55
2015-06-30	53

In [12]:
```
daily.plot(figsize=(10,7))
plt.ylabel('Frequency')
plt.title('Daily Trend Chart')
plt.show()
```



Inferrence : We see a huge day spike in june 24th

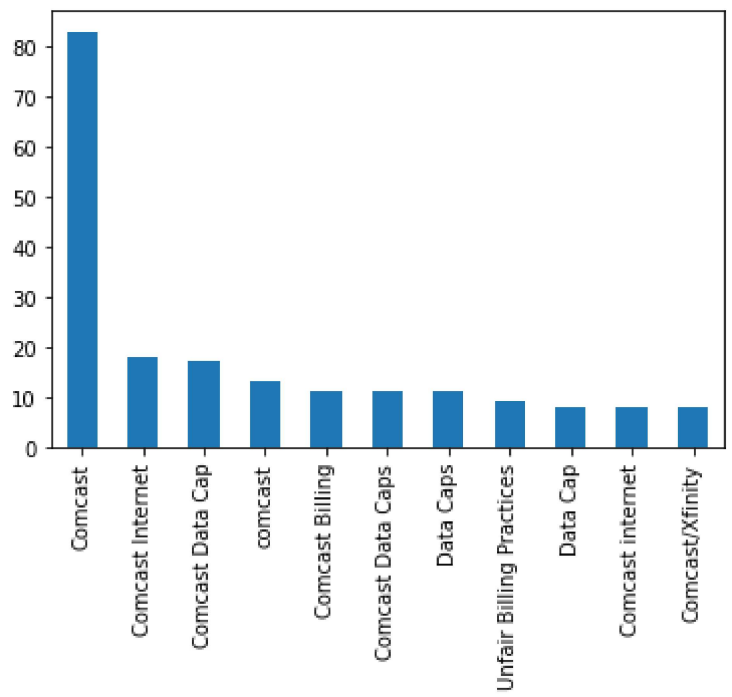
Provide a table with the frequency of complaint types.

In [13]:
```
#Listing top 10 complaint types
df['Customer Complaint'].value_counts().head(10)
```

Out[13]:
```
Comcast                      83
Comcast Internet             18
Comcast Data Cap             17
comcast                      13
Comcast Billing              11
Comcast Data Caps            11
Data Caps                    11
Unfair Billing Practices      9
Data Cap                      8
Comcast internet              8
Name: Customer Complaint, dtype: int64
```

In [14]:
```
df['Customer Complaint'].value_counts()[:11].plot.bar()
```

Out[14]:
```
<AxesSubplot:>
```



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

In [15]:
```
#counting issues related to network
internet_issues1=df[df['Customer Complaint'].str.lower().str.contains('network')].count()
#counting issues related to speed
internet_issues2=df[df['Customer Complaint'].str.lower().str.contains('speed')].count()
#counting issues related to data
internet_issues3=df[df['Customer Complaint'].str.lower().str.contains('data')].count()
#counting issues related to internet
internet_issues4=df[df['Customer Complaint'].str.lower().str.contains('internet')].count()
```

In [16]:
```
#counting issues related to billing
billing_issues1=df[df['Customer Complaint'].str.lower().str.contains('billing')].count()
#counting issues related to charges
billing_issues2=df[df['Customer Complaint'].str.lower().str.contains('charges')].count()
```

In [17]:
```
#counting issues related to service
service_issues1=df[df['Customer Complaint'].str.lower().str.contains('service')].count()
#counting issues related to service
service_issues2=df[df['Customer Complaint'].str.lower().str.contains('customer')].count()
```

In [18]:
```
total_internet_issues=internet_issues1+internet_issues2+internet_issues3+internet_issues4
print('total_internet_issues:',total_internet_issues[0])
total_billing_issues=billing_issues1+billing_issues2
print('total_billing_issues:',total_billing_issues[0])
total_service_issues=service_issues1+service_issues2
print('total_service_issues:',total_service_issues[0])
total_others_issues=df.shape[0]-(total_internet_issues+total_billing_issues+total_service_issues)
print('total_other_issues:',total_others_issues[0])
```

```
total_internet_issues: 945
total_billing_issues: 375
total_service_issues: 584
total_other_issues: 320
```

Inferrence : we see that most of the complaints fall inot internet issues and service issues

- 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 [19]:
```
df['NewStatus']=df['Status'].map({'Open':'open','Pending':'open','Closed':'closed','Solved':'closed'})
df.head()
```

Out[19]:

Date_month_year	Ticket #	Customer Complaint	Date	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	NewStatus
2015-04-22	250635	Comcast Cable Internet Speeds	22-04-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Closed	No	closed
2015-08-04	223441	Payment disappear - service got disconnected	04-08-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Closed	No	closed

Date_month_year	Ticket #	Customer Complaint	Date	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	NewStatus
2015-04-18	242732	Speed and Service	18-04-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Closed	Yes	closed
2015-07-05	277946	Comcast Imposed a New Usage Cap of 300GB that ...	05-07-15	11:59:35 AM	Internet	Acworth	Georgia	30101	Open	Yes	open
2015-05-26	307175	Comcast not working and no service to boot	26-05-15	1:25:26 PM	Internet	Acworth	Georgia	30101	Solved	No	closed

- Provide state wise status of complaints in a stacked bar chart. Use the categorized variable from Q3. Provide insights on:

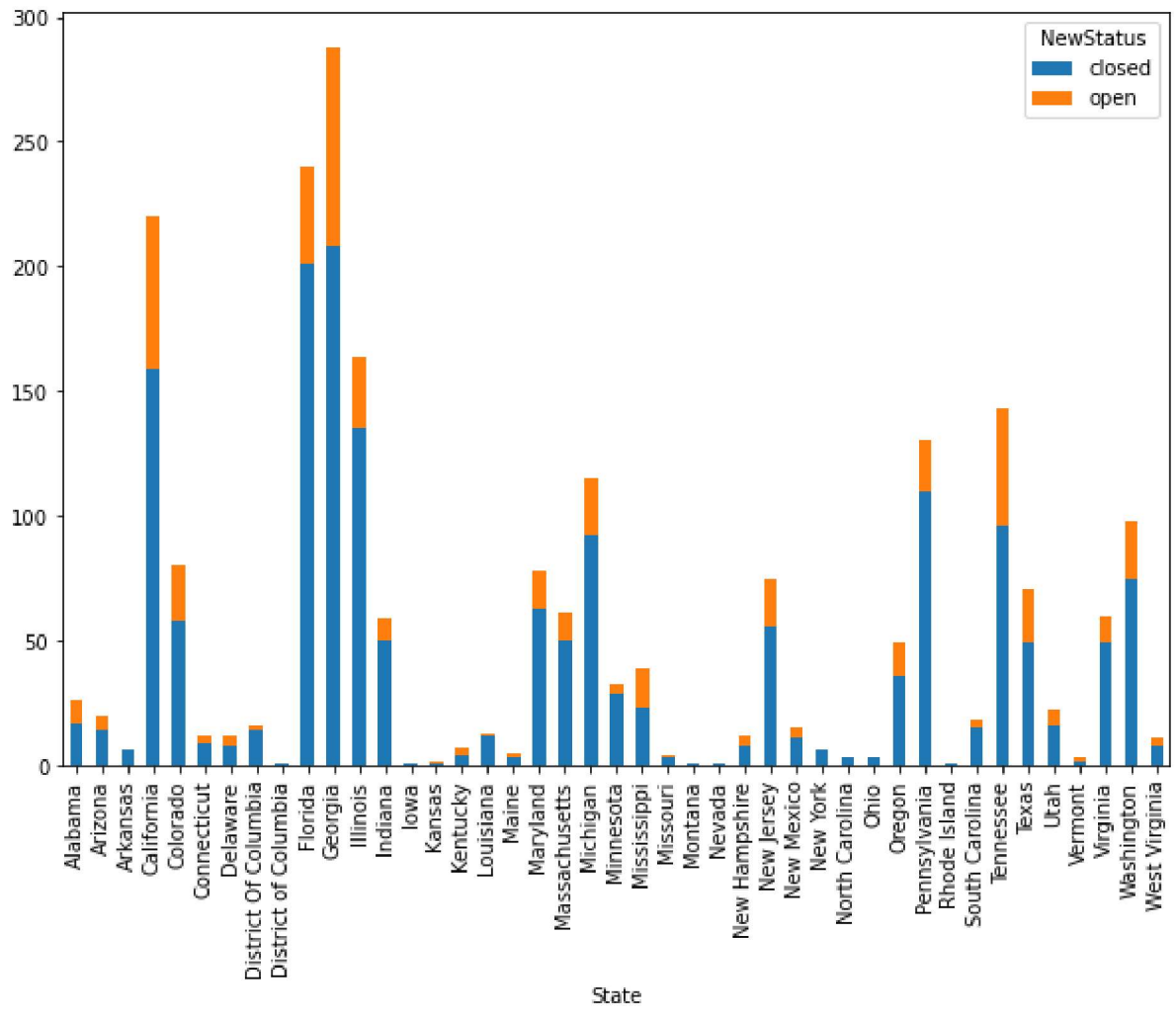
In [20]:
```
state_complain=df.groupby(['State','NewStatus']).size().unstack().fillna(0)
state_complain.head()
```

Out[20]:

State	closed	open
Alabama	17.0	9.0
Arizona	14.0	6.0
Arkansas	6.0	0.0
California	159.0	61.0
Colorado	58.0	22.0

In [21]:
```
state_complain.plot.bar(figsize=(10,7),stacked=True)
```

Out[21]: <AxesSubplot:xlabel='State'>



Which state has the maximum complaints

In [22]:
```
df.groupby('State').size().sort_values(ascending=False).head(1)
```

Out[22]:
```
State
Georgia    288
dtype: int64
```

Inferrence : Georgia has the most complaints

Which state has the highest percentage of unresolved complaints

In [23]:
```
df.NewStatus.value_counts()
```

Out[23]:
```
closed    1707
open       517
Name: NewStatus, dtype: int64
```

In [24]:
```
unresolved_data=df.groupby(['State','NewStatus']).size().unstack().fillna(0)
unresolved_data.head()
```

Out[24]:

State	closed	open
Alabama	17.0	9.0
Arizona	14.0	6.0

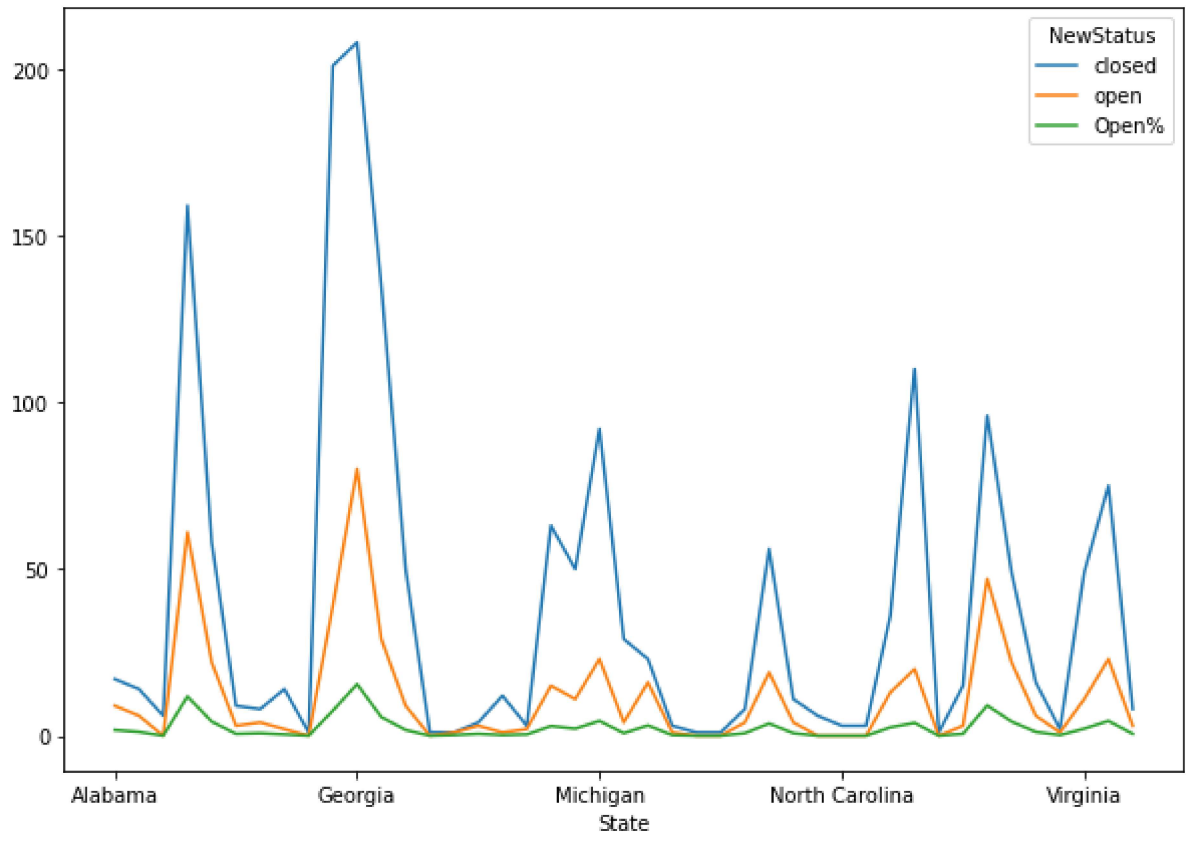
NewStatus	closed	open
State		
Arkansas	6.0	0.0
California	159.0	61.0
Colorado	58.0	22.0

In [25]:
```
unresolved_data['Open%']=unresolved_data.open * 100/(unresolved_data.open.sum())
unresolved_data.sort_values(by='Open%',ascending=False).head()
```

Out[25]:

NewStatus	closed	open	Open%
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

In [26]:
```
unresolved_data.plot(figsize=(10,7))
plt.show()
```



Inferrence : Georgia has the most unresolved complaints

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

In [27]:
```
resolved_data=df.groupby(['Received Via','NewStatus']).size().unstack().fillna(0)
resolved_data.head()
```

Out[27]:

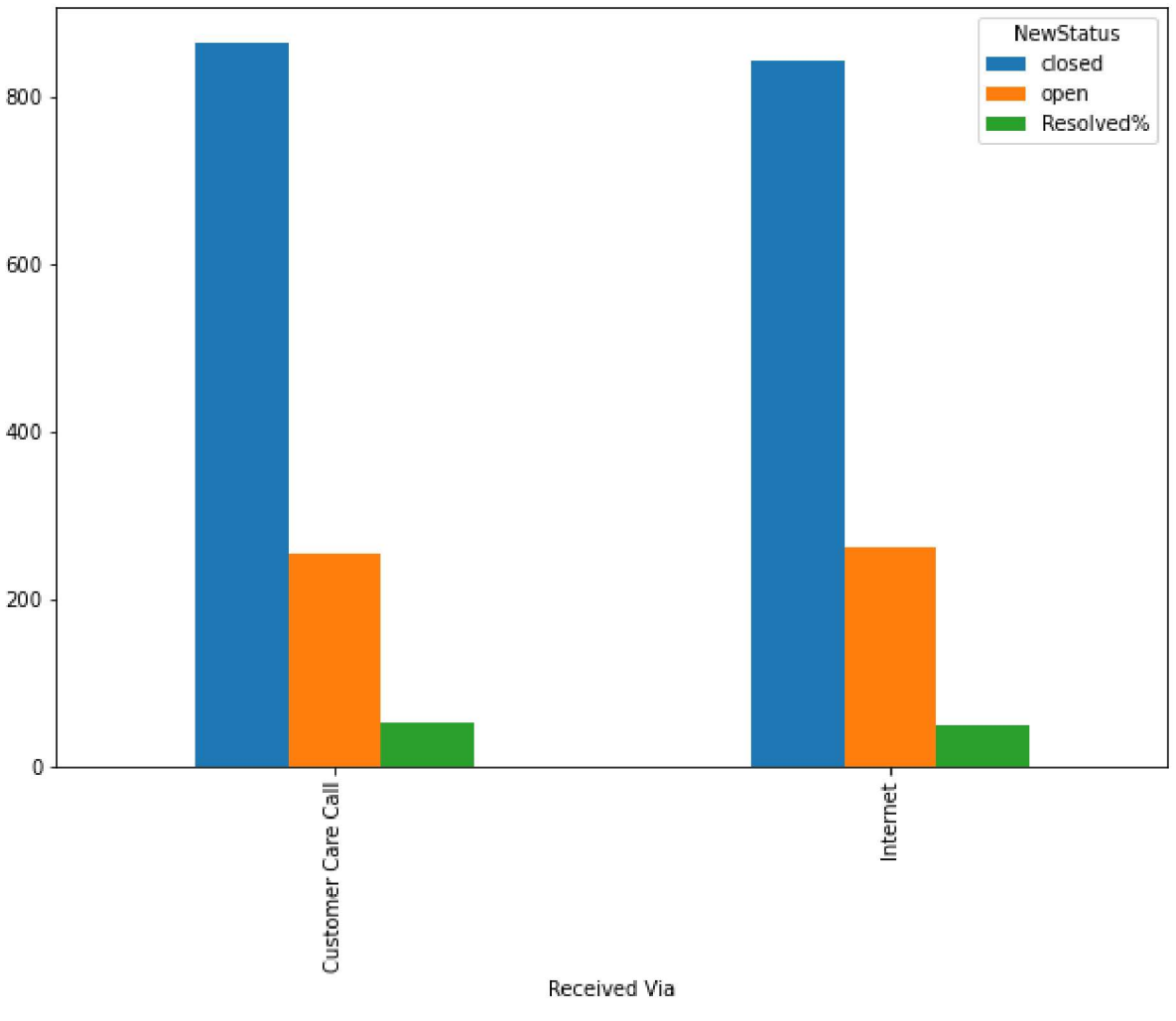
NewStatus	closed	open
Received Via		
Customer Care Call	864	255
Internet	843	262

In [28]:
```
resolved_data['Resolved%']=resolved_data.closed * 100/(resolved_data.closed.sum())
resolved_data.sort_values(by='Resolved%',ascending=False).head()
```

Out[28]:

NewStatus	closed	open	Resolved%
Received Via			
Customer Care Call	864	255	50.615114
Internet	843	262	49.384886

In [29]:
```
resolved_data.plot(kind='bar',figsize=(10,7))
plt.show()
```


NewStatus
closed
open
Resolved%
800
600
400
200
0
Customer Care Call
Internet
Received Via