PROJECT 3- Comcast-Telecom-Consumer-Complaints

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. The existing database will serve as a repository of public customer complaints filed against Comcast. It will help to pin down what is wrong with Comcast's customer service.

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

Analysis Task -

To perform these tasks, you can use any of the different Python libraries such as NumPy, SciPy, Pandas, scikit-learn, matplotlib, and BeautifulSoup.

- Import data into Python environment.
- Provide the trend chart for the number of complaints at monthly and daily granularity levels.
- Provide a table with the frequency of complaint types.

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

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

Which state has the maximum complaints Which state has the highest percentage of unresolved complaints

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

Coding part -

Importing the req libraries

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
```

In [53]:

```
%%HTML
<style type="text/css">
    table.dataframe td, table.dataframe th {
        border-style: solid ;
    }
</style>
```

In [ ]:

```
df1 = pd.read_csv('./Comcast_telecom_complaints_data1.csv')  # to read csv file #
df1
```

In [40]:

```
df1.head(5)   ## to see first 5 rows of file
```

Out[40]:

	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	4/8/2015	4-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 ...	5/7/2015	5-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

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

In [36]:

```
df1.isna().sum()
```

Out[36]:

```
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
date_index                     0
dtype: int64
```

In [37]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2224 entries, 2015-01-04 10:43:20 to 2015-12-06 13:18:20
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   object        
 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  date_index                   2224 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(9)
memory usage: 225.9+ KB
```

In [41]:

```
df1.describe()
```

Out[41]:

	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

In [42]:

```
df1_dmy=df1.groupby('Date_month_year')
```

In [43]:

```
df1_dmy.size()
```

Out[43]:

```
Date_month_year
13-Apr-15     24
13-Jun-15     32
13-May-15     12
14-Apr-15     23
14-Jun-15     16
              ..
6-Mar-15      25
6-May-15      29
6-Nov-15      21
6-Oct-15      25
6-Sep-15      23
Length: 91, dtype: int64
```

In [44]:

```
df1.sort_values(['Date_month_year'], axis=0, ascending=True, inplace=True, kind='quicksort', na_position='last')
df1.head()
```

Out[44]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone
1005	231757	monopoly	13-04-15	13-Apr-15	10:44:03 AM	Customer Care Call	Katy	Texas	77493	Closed	No
864	231729	comcast contract problem	13-04-15	13-Apr-15	10:28:49 AM	Internet	Houston	Texas	77049	Closed	No
1668	232972	Internet Speed	13-04-15	13-Apr-15	4:56:22 PM	Internet	Richmond	Indiana	47374	Closed	No
746	231821	Inability to get access to internet through Co...	13-04-15	13-Apr-15	11:10:02 AM	Customer Care Call	Gainesville	Georgia	30506	Closed	Yes
392	231532	Comcast problems	13-04-15	13-Apr-15	12:24:21 AM	Customer Care Call	Chicago	Illinois	60612	Closed	No

In [45]:

```
df1['month'] = pd.DatetimeIndex(df1['Date_month_year']).month
df1.head()
```

Out[45]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	Status	Filing on Behalf of Someone	month
1005	231757	monopoly	13-04-15	13-Apr-15	10:44:03 AM	Customer Care Call	Katy	Texas	77493	Closed	No	4
864	231729	comcast contract problem	13-04-15	13-Apr-15	10:28:49 AM	Internet	Houston	Texas	77049	Closed	No	4
1668	232972	Internet Speed	13-04-15	13-Apr-15	4:56:22 PM	Internet	Richmond	Indiana	47374	Closed	No	4
746	231821	Inability to get access to internet through Co...	13-04-15	13-Apr-15	11:10:02 AM	Customer Care Call	Gainesville	Georgia	30506	Closed	Yes	4
392	231532	Comcast problems	13-04-15	13-Apr-15	12:24:21 AM	Customer Care Call	Chicago	Illinois	60612	Closed	No	4

In [46]:

```
df1_month = df1.groupby('month')
df1_month_size = df1_month.size()
df1_month_size
```

Out[46]:

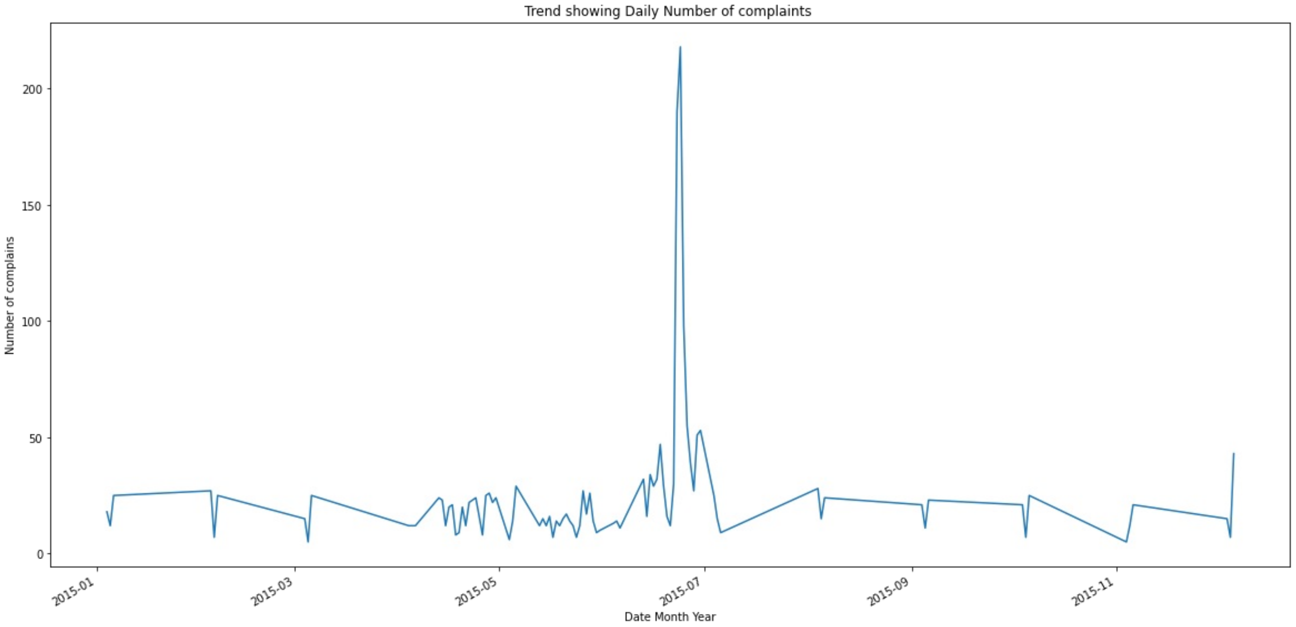
```
month
1       55
2       59
3       45
4      375
5      317
6     1046
7       49
8       67
9       55
10      53
11      38
12      65
dtype: int64
```

In [47]:

```
df1["Date_month_year"] = pd.to_datetime(df1["Date_month_year"])
```

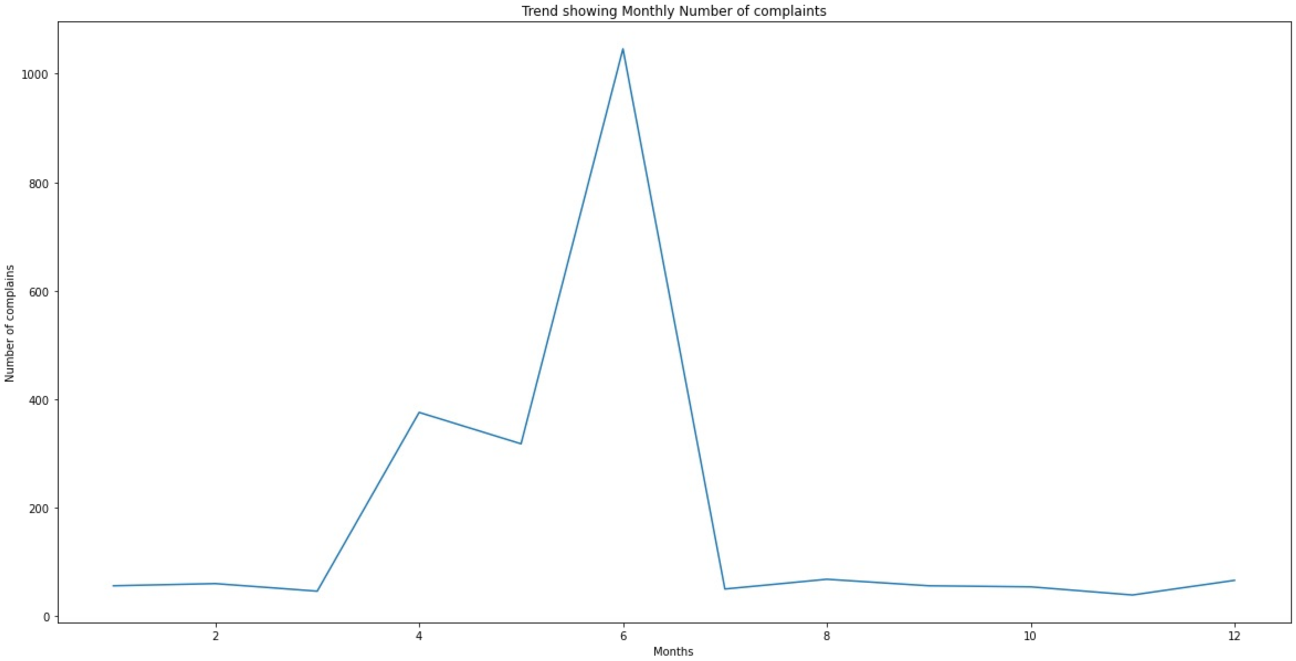
In [48]:

```
plt.figure(figsize=(20,10))
df1['Date_month_year'].value_counts().plot()
plt.title('Trend showing Daily Number of complaints ')
plt.xlabel('Date Month Year')
plt.ylabel('Number of complains')
plt.show()
```



In [49]:

```
plt.figure(figsize=(20,10))
df1_month_size.plot()
plt.title('Trend showing Monthly Number of complaints')
plt.xlabel('Months')
plt.ylabel('Number of complains')
plt.show()
```



Provide a table with the frequency of complaint types

In [9]:

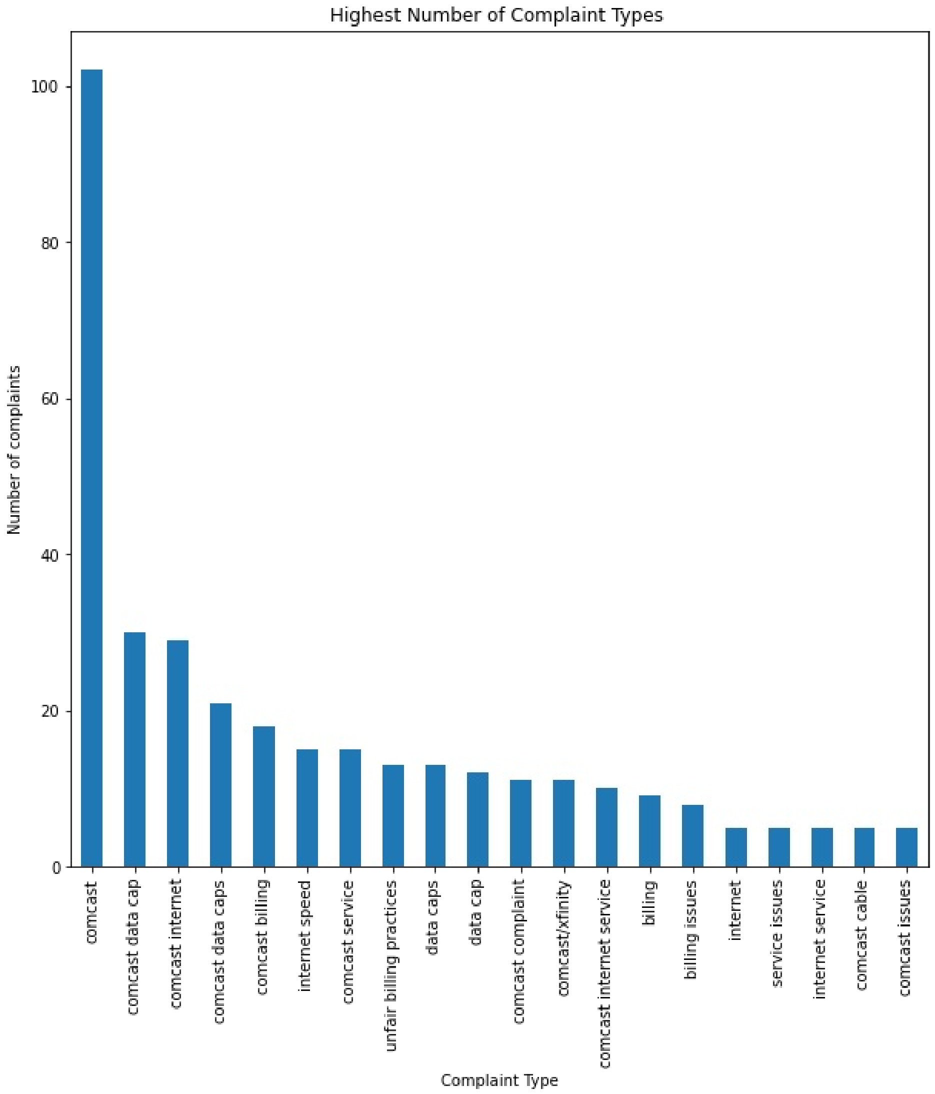
```
complain_type = df1['Customer Complaint'].str.lower().value_counts()
complain_type.head(20)
```

Out[9]:

```
comcast                        102
comcast data cap                30
comcast internet                29
comcast data caps               21
comcast billing                 18
internet speed                  15
comcast service                 15
data caps                       13
unfair billing practices        13
data cap                        12
comcast/xfinity                 11
comcast complaint               11
comcast internet service        10
billing                          9
billing issues                   8
comcast issues                   5
comcast billing complaint        5
comcast billing practices        5
comcast cable                    5
slow internet                    5
Name: Customer Complaint, dtype: int64
```

In [53]:

```
complain_type_head = complain_type.head(20)
plt.figure(figsize=(10,10))
complain_type_head.plot.bar()
plt.title('Highest Number of Complaint Types ')
plt.xlabel('Complaint Type')
plt.ylabel('Number of complaints')
plt.show()
```



Observation - Maximum complaints are from data cap, internet problems and billing issues and services

**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 [12]:

```
df1.groupby('Status').size()
```

Out[12]:

```
Status
Closed     734
Open       363
Pending    154
Solved     973
dtype: int64
```

In [13]:

```
df1['New Status'] = df1['Status']
```

In [14]:

```
df1['New Status'].replace(('Pending', 'Solved'), ('Open', 'Closed'), inplace=True)
df1.groupby('New Status').size()
```

Out[14]:

```
New Status
Closed    1707
Open       517
dtype: int64
```

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

Which state has the maximum complaints

In [52]:

```
df1.groupby('State').size()
```

Out[52]:

```
State
Alabama                  26
Arizona                  20
Arkansas                  6
California              220
Colorado                 80
Connecticut              12
Delaware                 12
District Of Columbia     16
District of Columbia      1
Florida                 240
Georgia                 288
Illinois                164
Indiana                  59
Iowa                      1
Kansas                    2
Kentucky                  7
Louisiana                13
Maine                     5
Maryland                 78
Massachusetts            61
Michigan                115
Minnesota                33
Mississippi              39
Missouri                  4
Montana                   1
Nevada                    1
New Hampshire            12
New Jersey               75
New Mexico               15
New York                  6
North Carolina            3
Ohio                      3
Oregon                   49
Pennsylvania            130
Rhode Island              1
South Carolina           18
Tennessee               143
Texas                    71
Utah                     22
Vermont                   3
Virginia                 60
Washington               98
West Virginia            11
dtype: int64
```

In [16]:

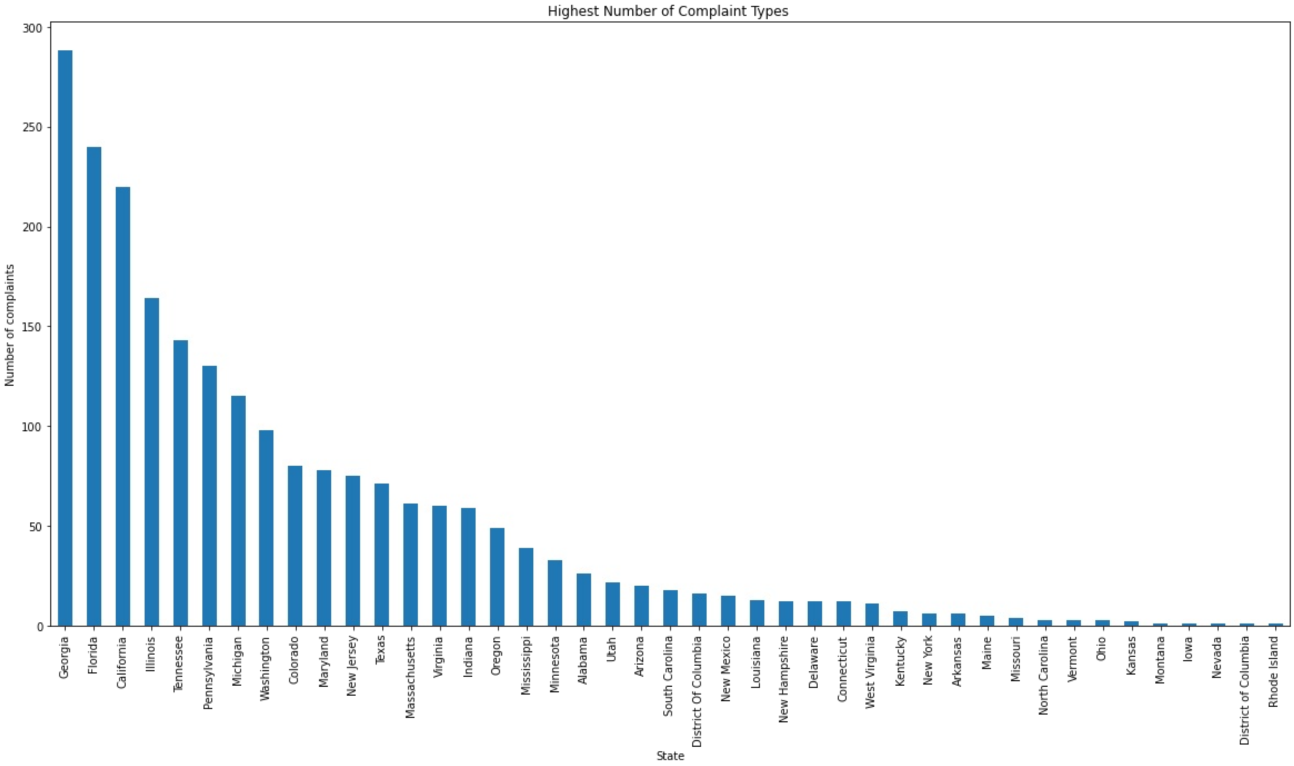
```
df1['State'].value_counts()
```

Out[16]:

```
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
New Hampshire            12
Delaware                 12
Connecticut              12
West Virginia            11
Kentucky                  7
New York                  6
Arkansas                  6
Maine                     5
Missouri                  4
North Carolina            3
Vermont                   3
Ohio                      3
Kansas                    2
Montana                   1
Iowa                      1
Nevada                    1
District of Columbia      1
Rhode Island              1
Name: State, dtype: int64
```

In [23]:

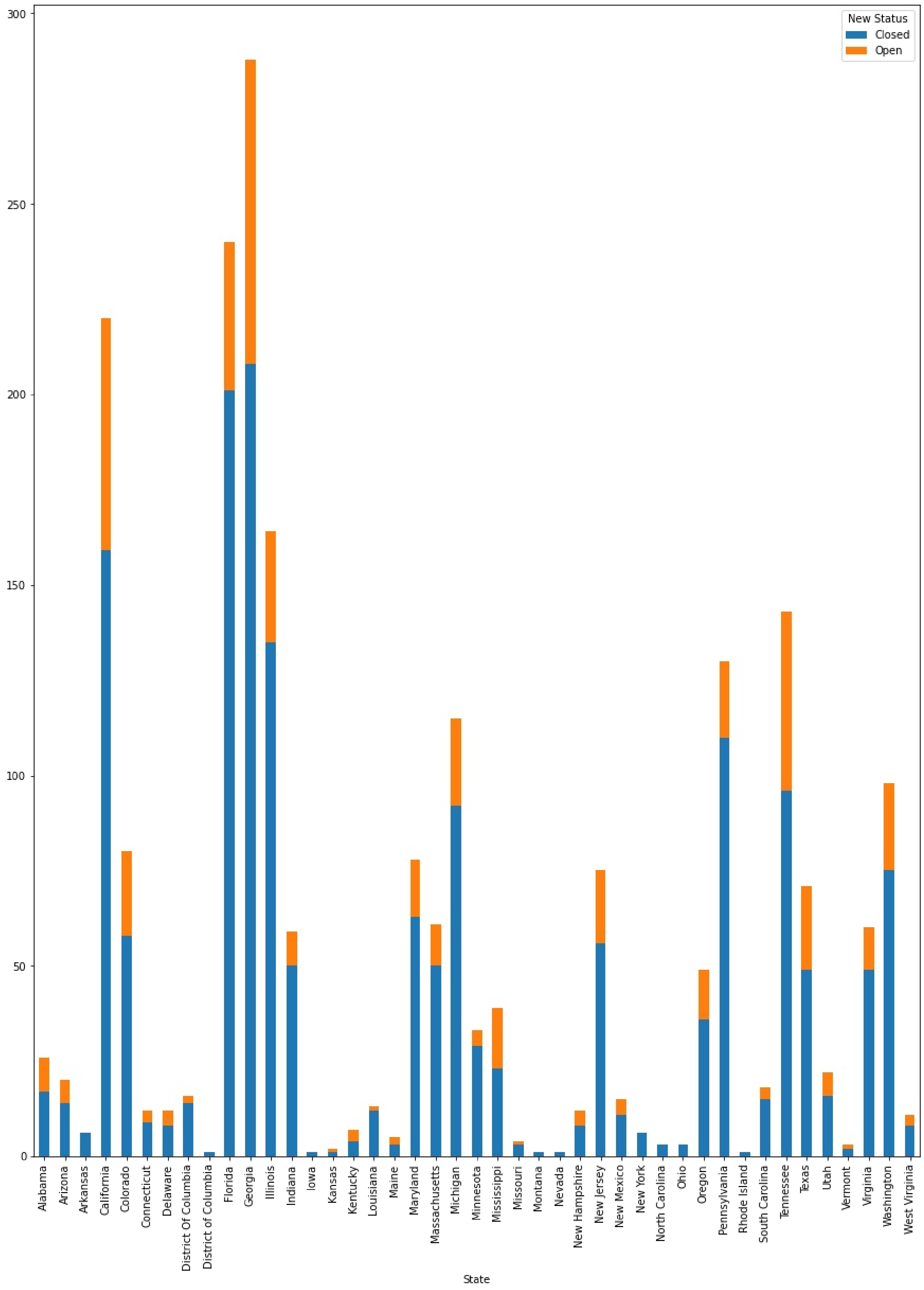
```
plt.figure(figsize=(20,10))
df1['State'].value_counts().plot.bar()
plt.title('Highest Number of Complaint Types ')
plt.xlabel('State')
plt.ylabel('Number of complaints')
plt.show()
```



In [26]:

```
print(df1.groupby(['State', 'New Status']).size().unstack().plot(kind='bar', figsize = (15,20), stacked=True))
```

```
AxesSubplot(0.125,0.125;0.775x0.755)
```



Graph observation - Georgia has the maximum complaints (288)

Which state has the highest percentage of unresolved complaints

In [30]:

```
complaint_by_state = df1.groupby(['State','New Status']).size().unstack().fillna(0)
complaint_by_state
```

Out[30]:

New Status	Closed	Open
State		
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
Connecticut	9.0	3.0
Delaware	8.0	4.0
District Of Columbia	14.0	2.0
District of Columbia	1.0	0.0
Florida	201.0	39.0
Georgia	208.0	80.0
Illinois	135.0	29.0
Indiana	50.0	9.0
Iowa	1.0	0.0
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	0.0
Nevada	1.0	0.0
New Hampshire	8.0	4.0
New Jersey	56.0	19.0
New Mexico	11.0	4.0
New York	6.0	0.0
North Carolina	3.0	0.0
Ohio	3.0	0.0
Oregon	36.0	13.0
Pennsylvania	110.0	20.0
Rhode Island	1.0	0.0
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

Which state has the highest percentage of unresolved complaints

In [32]:

```
complaint_by_state.iloc[0,1]

closed_cont = []
opened_cont = []

for i in range(0, len(complaint_by_state)):
     closed_cont.append(complaint_by_state.iloc[i,0])

for i in range(0, len(complaint_by_state)):
     opened_cont.append(complaint_by_state.iloc[i,1])
```

In [33]:

```
closed_complains = np.array(closed_cont)
opened_complains = np.array(opened_cont)
```

In [34]:

```
percentage_opened = (opened_complains/(opened_complains+closed_complains)) * 100
percentage_closed = (closed_complains/(opened_complains+closed_complains)) * 100
percentage_opened
```

Out[34]:

```
array([34.61538462, 30.        ,  0.        , 27.72727273, 27.5       ,
       25.        , 33.33333333, 12.5       ,  0.        , 16.25      ,
       27.77777778, 17.68292683, 15.25423729,  0.        , 50.        ,
       42.85714286,  7.69230769, 40.        , 19.23076923, 18.03278689,
       20.        , 12.12121212, 41.02564103, 25.        ,  0.        ,
        0.        , 33.33333333, 25.33333333, 26.66666667,  0.        ,
        0.        ,  0.        , 26.53061224, 15.38461538,  0.        ,
       16.66666667, 32.86713287, 30.98591549, 27.27272727, 33.33333333,
       18.33333333, 23.46938776, 27.27272727])
```

In [35]:

```
dictionary = complaint_by_state.to_dict() # converting to dictionary
```

In [36]:

```
c = {} #closed
o = {} #opened

for k, v in dictionary.items(): # k represents keys and v represents values in dictionary
    if k == 'Closed':
        c.update(v)
    else:
        o.update(v)
```

In [37]:

```
# Finding a list with only states names
states = []
for keys in c:
    states.append(keys)
```

In [38]:

```
print("Length of states is {} and length of percentage open list is {}". format(len(states), len(percentage_opened)))
```

```
Length of states is 43 and length of percentage open list is 43
```

In [39]:

```
#fucntion to find the index of the maximum percentage value
def find_max_index(percentage_list):

    result = np.where(percentage_list == np.amax(percentage_list))
    return result[0]
```

In [40]:

```python
#converting items in a list to a single integer
def convert(list_values):
    #convert to string first
    s = [str(i) for i in list_values]
    # Join list items using join()
    res = int("".join(s))
    return res
```

In [41]:

```python
opened_max_index = convert(find_max_index(percentage_opened))
#closed_max_index = find_max_index(percentage_closed) # This is not needed because these are already resolved cases.
                                                     # Finding the max number will give 8 total states.
                                                     # They have ONLY closed cases so the percentage is 100%.
```

In [42]:

```python
print('Highest Unresolved Percentage is',percentage_opened.max(),'% from', states[opened_max_index])
```

```
Highest Unresolved Percentage is 50.0 % from Kansas
```

Kansas has the highest percentage of unresolved complaints. This can be seen from the stacked bar graph as well as the calculation shown above.

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

In [44]:

```python
comcast_received = df1.groupby(['Received Via', 'New Status']).size()
print(comcast_received)
```

```
Received Via       New Status
Customer Care Call  Closed        864
                    Open          255
Internet            Closed        843
                    Open          262
dtype: int64
```

In [46]:

```python
closed = []
opened = []
for counter, value in enumerate(comcast_received):
    if counter % 2 == 0:
        closed.append(value)
    else:
        opened.append(value)
```

In [47]:

```python
closed = np.array(closed)
opened = np.array(opened)
```

In [48]:

```python
percentage_open = (opened/(opened+closed)) * 100
percentage_close = (closed/(opened+closed)) * 100
```

In [49]:

```python
print(percentage_open)
```

```
[22.78820375 23.71040724]
```

Customer Care Call - 22.78% Open

Internet - 23.71% Open

In [50]:

```python
print(percentage_close)
```

```
[77.21179625 76.28959276]
```

Customer Care Call - 22.78% close

Internet - 23.71% close

END of PROJECT

In [ ]: